Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7923066

AN ENHANCED ARCHITECTURE OF TURBO ECODER FOR WIRELESS APPLICATIONS

[1]Mr.P.Balakrishna
Research Scholar, Shri Jagdishprasad Jhabarmal Tibrewala University, Rajasthan

[2]Dr.S.Julian Savari Antony
Associate Professor, Keshav Memorial Institute of Technology, Hyderabad
Email:Padma.balakrishna@gmail.com

ABSTRACT

Numerous channel coding algorithms are being used more frequently in contemporary wireless communication systems to improve throughput and latency. Because they improve the communication's robustness and resistance to noise and other kinds of interference, interleavers are an essential part of a variety of coding schemes. As a result of designers taking a more parallel design approach in response to the constant demand for higher throughputs and shorter delays, parallel versions of these encoding/decoding techniques have emerged.

Keywords: Turbo Codes, QPP, Wireless sensors, LTE.

I.INTRODUCTION

For error correction of Shannon-limit results, two recursive convolutional turbo decoder codes were also introduced in 1993 [1]. Channel and the MAP decoder were transmitted using the parity and systematic bits. Soft values were captured by the MAP decoder and receiver, which calculate log-likelihood ratio reliability. High-speed decoding necessitates a meticulous design for the decoding process, which is an essential function of the interleave. A high rate is the one essential requirement for wireless communication. The high throughput sliding method decreases throughput and increases silicon area for parallel decoders. Previously suitable for the turbo decoder of parallelization, the QPP and contention-free interleavers can now be utilized. Interleaver memory is the issue at hand, resulting in an efficient architecture. Numerous architectures utilized master-slave batcher networks[6]. The deinterleaver is yet another turbo decoding block that performs inverse operations in their original order. When implemented in deinterleaver-free architecture, interleaving in quadratic inverse [4] results in hardware that is more complicated and takes up more space [8]. Additionally, efficient VLSI architecture still lacks efficient deinterleaving architecture. This paper proposes an add-compare-select (ACS) network-based turbo decoder.

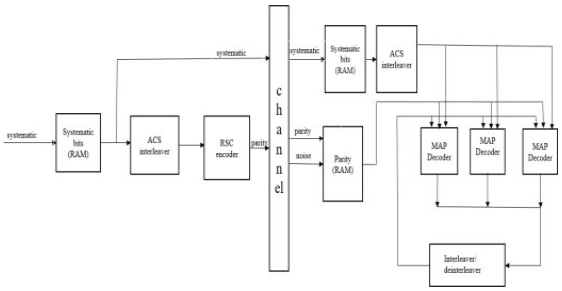

Fig.1. Architecture of the turbo decoder and encoder, despite the addition of some works to locate the QPP

II.ACS INTERLEAVER

A crucial design parameter is the combination of parallel turbo decoders and MAP decoders. After multiplying each block by eight, the suggested architecture can achieve the highest design number, N=8. All blocks were decoded simultaneously by LTE. Parallelism-based ordering is an easy choice for a particular size.

1. Permutation network for Add, Compare, and Select (ACS)

Using a permutation and address generator block interleaver, the values of K LLR were stored column-wise in a folded memory (Fig. 2). Sorted in a descending order, starting at 0, the tiny interleaved rows that were connected to each address in folded memory were shown. With the intention of defining the number of parallelization units as N and the trellis segment as Sand NS=K, each row in the LLR values was designated as non-interleaved and folded memory. Address generators were utilized to calculate the interleaved addresses, which were then taken in permutation order. The creation permutation is unit. The folded memory address was sorted in ascending order from the trellis segment. After the permutation signal is obtained, the adder for the output is sent. for scrambling inputs, and the permutation signals of the selection unit contain N multipliers for Nin inputs. In order to improve outcomes, a VLSI architecture for the network is proposed.

2. Addressgenerator

Based on the following equations, a recursive proposal to generate addresses for the ACS permuting network's inputs is demonstrated.□□i□□$f_1.i+f_2.i^2(modK)$, (1)

□□i+□□□□$f_1.i+f_2.i^2+f_1.$□+2.□.$f_2.i+$□$^2.f_2(modK)$, (2)

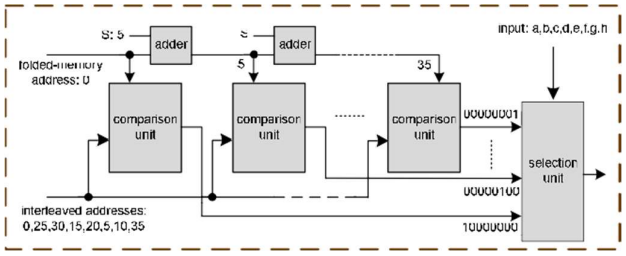

Fig.2. There are seven adders, eight comparison units, and one selection unit for N=8 in the ACS permuting network architecture that has been proposed.

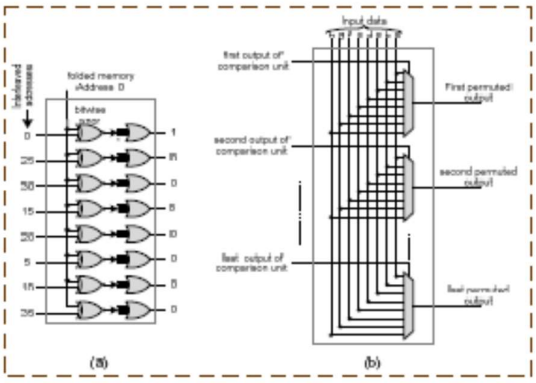

Fig.3. (a)Thecomparisonunit.(b)Theselectionunit

$=\square\square i\square\square\square\square i\square(modK)\square$ $\square\square\square$

$\square\square i\square\square f_1+2.\square.f_2.i+\square^2.f_2(modK),$ (3)

$\square\square i+\square\square=\square\square i\square+2.\square^2.f_2(modK).$ (4)

Multiplicative operations are unnecessary because there are only additions and modulo involved. The interleaved addresses for each row are necessary for the address generator block of the proposed architecture to function. The trellis segment S must be substituted in the recursive formulas in order to accomplish this. These formulas make it possible to use modulo-operating comparing and selecting circuits to create an effective VLSI architecture.

3. Interelaver/Deinterleaver block

One of the turbo decoder's more difficult blocks is the deinterleaver. Modifying the address generator block to generate addresses with an eye toward the QPP inverse is one way to implement the deinterleaving block. The address generator block's hardware complexity rises significantly as a result of the persistent cubic inverse for certain block sizes. As a result, a more straightforward architecture that accommodates all block sizes must be superior. A normal way to deal with carrying out equal super disentangling separates one emphasis into equal parts, with one block expected to perform-interleaving in one half and one more block expected to deinterleave in the other. Figure shows this arrangement in action. 4, and it serves as an example of the procedure.

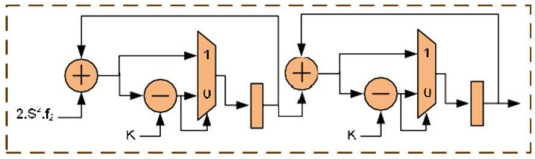

Fig.4. The serial address generator block

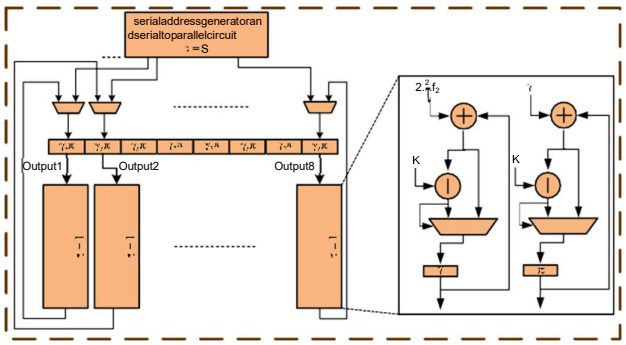

Fig. 5. The parallel address generation block with extended address generator.

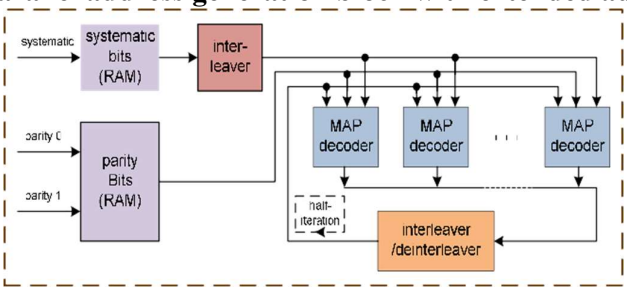

Fig.5. The parallel turbo decoder's parallel architecture

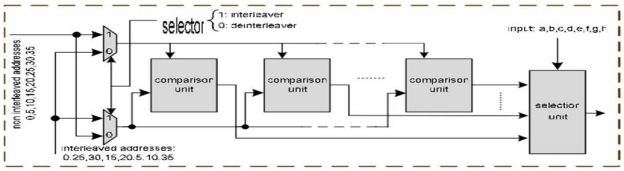

Fig.6.The detailed architecture of the interleaver/deinterleaver block for N = 8.

IMPLEMENTATION RESULTS AND DISCUSSION

The turbo decoder makes use of the parallel-based iterative decoding method to achieve high throughput rates. It accomplishes this by dividing the block in the N MAP processor into subblacks of length S of N.

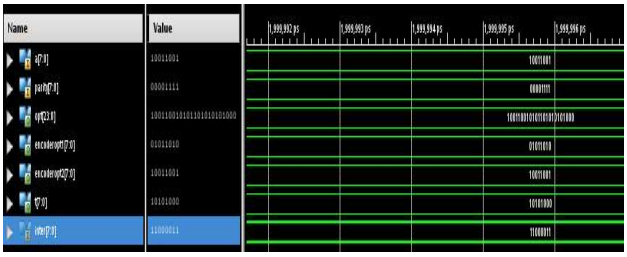

Fig. 7: Turbo Encoder

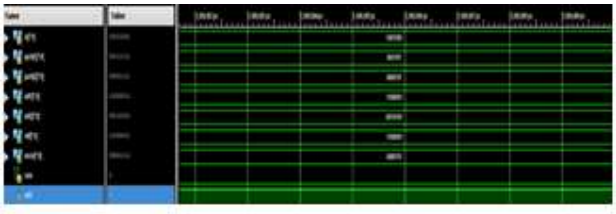

Fig.8: ACS Interleaver/Deinterleaver

Using ACS as the interleaver will eliminate all of the drawbacks of the QPP interleaver, such as its high power consumption. The LTE makes use of the QPPinterleaver, whose inputs and outputs are followed by:

$$x'_i = x_{\square i \square}, i = 0,1,2,\ldots,K-1, \quad (5)$$

Where Kisthecodeblocksizeand$\pi\square i\square\square$is:

$$\pi\square i\square\square f_1.i + f_2.i^2 (modK) \qquad (6)$$

where the parameters f1 and f2 are determined by the block size K. If the following expression is true, the QPP interleaver is contention-free:

$$\lfloor \pi(j+t1S)/S \rfloor \neq \lfloor \pi(j+t2S)/S \rfloor, (7)$$

Where $0 \leq j < S, 0 \leq t1, t2 < N$.

The interleaver from QPP contains vectorizable [7]. The interleaver bottleneck is proposed in [5] [7]. K block sizes in LTE are 40 to 6144. The QPP interleavers also form the inverse, as seen in [8]. As a result, the interleaving algorithm can be used in the deinterleaving process.

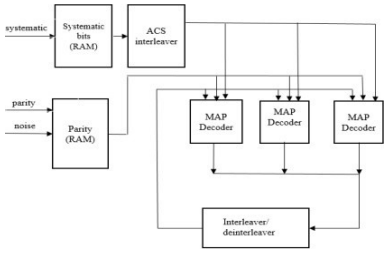

Fig.9:ParallelTurboDecoderArchitecture

ACS Permuting Network

The suggested design and the MSBatcher network architecture [3] were both implemented within the same framework for comparison. TABLE provides a summary of the synthesis's ASIC implementation-based outcomes. The width of the input data (LLR values) and the block size of K=6144 are the outcomes. Another advantage of using the pipeline method with the suggested architecture is a faster data rate. According to the findings, the proposed design uses 38.8% less energy than the current state of the art. Figure 9 depicts the proposed design's layout.

TABLE I. Interleaver/Deinterleaver Block Synthesis Results.

		Interleaver/Deinterleaver	**Address-generator**
ASIC(180nm)	**Power(mW)**	13.3	8.2
	Throughput(Gbps)	14.98	69.26
	Area(um^2)	57604	37725
	Frequency(MHz)	336	336
	Criticalpath(ns)	2.19	2.19
	Energyefficiency(pJ/bit)	0.69	0.013
Virtex-6(xcvlx240t)	**Throughput(Gbps)**	18.58	80.74
	Frequency(MHz)	406	391
	#ofSliceRegisters	479(<<1%)	203(<<1%)
	#ofSliceLUTs	858(<<1%)	473(<<1%)

TABLE II displays the synthesis results for the XC6VLX240T- 1FFG1156 FPGA implementation and ASIC implementation. A separate architecture that reversed the interleaving network to perform deinterleaving was thought to exist alongside the interleaver, as stated in [7].

Power consumption is reduced by 44.3% when the proposed interleaver/deinterleaver architecture is implemented. Additionally, the Xilinx ML605 evaluation kit was used to test the proposed scheme, and MATLAB was used to compare and verify the platform's outputs with expected outcomes.

CONCLUSION

Utilizing a novel architecture and an ACS permuting network, this paper makes interleaving and deinterleaving for turbo decoding suggestions. Pipelining for high throughputs is simple thanks to the proposed ACS permuting network's space- and power-efficient design. As a direct consequence of the proposed solution, a brand-new architecture was developed that is capable of both interleaving and deinterleaving. This architecture eliminates the need for additional hardware for deinterleaving and shifts its complexity to the address generator by computing inverted addresses. It is possible to use any block size with the suggested method, which results in an efficient design.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," *In Proc. IEEE Int. Conf. on Commun.*, vol. 2, pp. 1064 –1070, May 1993.

[2] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.

[3] C. Studer, C. Benkeder, S. Belfani, Q. Huang, "Design and implementation of a parallel Turbo-Decoder ASIC for 3GPP-LTE," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 8–17, Jan. 2011.

[4] A. Nimbalker, Y. Blankenship, B. Classon, and T. K. Blankenship, "ARP and QPP interleavers for LTE turbo coding," *In Proc. IEEE WCNC*, Las Vegas, NV, USA, pp. 1032–1037, Mar. 2008.

[5] 3GPP TS 36.211 v9.1.0 Release 9, March 2010.

[6] J. Ryu and O. Y. Takeshita, "On quadratic inverses for quadratic permutation polynomials over integer rings," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1254–1260, Mar. 2006.

[7] S. Wang, J. Ma, G. He, Z. Mao, "Generalized interleaving network based on configurable QPP architecture for parallel decoder," *IEEE Workshop on Signal Processing Systems*, pp. 204–209, Oct. 2011.

[8] C. Ma, P. Lin, "Efficient implementation of quadratic permutation polynomial interleaver in turbo codes," *Int. Conf. on Wireless Commun. & Signal processing*, pp. 1–5, Nov. 2009.

[9] Claude Berrou. *Error-correction coding method with at least two systematic convolutional codings in parallel, corresponding iterative decoding method, decoding module and decoder*. 1991. URL: https://patents.google.com/patent/US5446747A/en.

[10] C. Berrou, A. Glavieux, and P. Thitimajshima. "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1". In: *Proceedings of ICC '93 - IEEE International Conference on Communications*. Vol. 2. IEEE, pp. 1064–1070. ISBN: 0-7803-0950-2. DOI: 10.1109/ICC.1993. 397441. URL: http://ieeexplore.ieee.org/document/397441/.

[11] Fred Daneshgaran and Marina Mondin. "Design of interleavers for turbo codes: iterative

interleaver growth algorithms of polynomial complexity". In: *IEEE Transactions on Information Theory* (1999). ISSN: 00189448. DOI: 10.1109/18.782105.

[12] Hamid R. Sadjadpour *et al.* "Interleaver design for turbo codes". In: *IEEE Journal on Selected Areas in Communications* (2001). ISSN: 07338716. DOI: 10.1109/49.924867.