

Design and Implementation of a Parallel Turbo Decoder for Wireless Communication

Pradeep.G¹, Ragul Raja.A², Saravanan.R³, Senthamil Selvan.M⁴, Poovendran.R⁵

U.G. Student, Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India^{1,2,3,4}

Associate Professor, Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India⁵

ABSTRACT: Wireless sensor network can be considered to be energy constrained wireless scenarios, since the sensors are operated for extended periods of time, while relying on batteries that are small, lightweight and inexpensive. Energy constrained wireless communication application is done with the help of lookup table-log-BCJR (LUT-Log-BCJR) architecture. In our existing system the conventional LUT-Log-BCJR architecture have wasteful designs requiring high chip areas and hence high energy consumptions Energy constrained applications. This motivated our proposed System the LUT log BCJR is designed with Clock gating technique achieves low-complexity energy-efficient architecture, which achieves a low area and hence a low energy consumption, and also achieving a low energy consumption has a higher priority than having a high throughput. we use most fundamental add compare select (ACS) operations and It having low processing steps, so that low transmission energy consumption is required and also reduces the overall energy consumption. Our demonstrated that our implementation has a throughput of 1.03 Mb/s.

KEYWORDS: ASIP, Clock Gating, LUT-Log-BCJR, throughput and Data rate.

I. INTRODUCTION

Turbo codes received a lot of interest from the research community, as they offer a performance unmatched by any other code so far. Turbo codes are also on the verge of finding their way in numerous cutting edge applications like cellular and satellite communications. A turbo decoder is composed of modules that work in an iterative scheme. Different alternatives exist for the algorithm incorporated in these decoder modules.

Recent Application-Specific Integrated Circuit (ASIC) based turbo decoder architectures have been designed for achieving a high transmission throughput, rather than for a low transmission energy. For example, turbo codes have facilitated transmission throughputs in excess of 50 Mbit/s in cellular standards, such as the 3rd Generation Partnership Project 3GPP Long Term Evolution (LTE) and recent ASIC turbo decoder architectures have been designed for throughputs that are in excess of 100 Mbit/s. This has been achieved by employing the Max-Log-BCJR turbo decoding algorithm, which is a low-complexity approximation of the optimal Logarithmic Bahl-Cocke-Jelinek-Raviv (Log-BCJR) algorithm.

The Max-Log-BCJR algorithm appears to lend itself to both high-throughput scenarios, as well as to the above mentioned energy-constrained scenarios. This is because a low turbo decoder energy consumption E_{prb} is implied by Max-Log-BCJR algorithm's low complexity. However, this is achieved at the cost of degrading the coding gain by 0.5 dB compared to the optimal Log-BCJR algorithm, increasing the required transmission energy E_{txb} by 10%. This disadvantage of the Max-Log-BCJR outweighs its attractively low complexity, when optimizing the overall energy consumption $E_b^{tx} + E_b^{prof}$ sensor nodes that are separated by dozens of meters.

This motivates the employment of the Look-Up-Table-Log-BCJR (LUT-Log-BCJR) algorithm in energyconstrained scenarios, since it approximates the optimal Log-BCJR more closely than the Max-Log-

BCJR and therefore does not suffer from the associated coding gain degradation. However, to the best of our knowledge, no LUTLog-BCJR ASICs have been specifically designed for energy-constrained scenarios. Previous LUT-Log-BCJR turbo decoder designs were developed as a part of the on-going drive for higher and higher processing throughputs, although their throughputs have since been eclipsed by the Max-Log-BCJR architectures. This opens the door for a new generation of LUT-Log-BCJR ASICs that exchange processing throughput for energy efficiency.

II. TURBO CODING

A. ENCODER SETUP:

A turbo encoder consists of a parallel concatenation of two relatively low constraint length convolutional codes as depicted. A pseudo random inter leaver π separates these two component codes (ENC1 and ENC2). The size N of this inter leaver determines the code performance, which is better for larger values of N . The turbo code operates as a block code transforming the input bits u_k into output bits c_{k1}, c_{k2}, c_{k3} that are transmitted.

For every new input block, the encoder starts in a known initial state. In order to have a known end state, some extra input bits, called tail bits, are generated. Traditionally this is done only for ENC1. We extend this method and add tail bits after the inter leaver as well, to also force ENC2 in a known end state. We refer to this scheme as full termination. Another option to force both encoders to a known end state is to use precursor bits as in. Both schemes reduce the throughput only very slightly, but improve the decoding performance and allow an interesting optimization of the decoder module.

B. DECODER SETUP:

A maximum likelihood turbo decoder would be prohibitively complex. The iterative decoding scheme of figure 1b is a clever solution to this problem. Each decoder module (DEC1 and DEC2) produces an increasingly better correction term (called extrinsic information λ^{extk}) which is appropriately (de)interleaved and used as a priori information (denoted as intrinsic information λ^{intk}) by the next module.

III. EXISTING SYSTEM

In this existing design, we implemented the LUT-Log-BCJR architecture into its most fundamental add compare select (ACS) operations and perform them using a novel low-complexity ACS unit. The demonstration also simulates the energy efficiency, and power reduction as shown in this paper.

The Wireless Sensor Networks (WSNs) can be considered to be energy constrained wireless scenarios since the sensors are operated for extended periods of time, while relying on batteries that are small, lightweight and inexpensive. The Max-Log-BCJR algorithm appears to lend itself to both high-throughput scenarios, as well as to the above-mentioned energy-constrained scenarios. This is because low turbo decoder energy consumption is implied by Max-Log-BCJR algorithm's low complexity. However, this is achieved at the cost of degrading the coding gain by 0.5 dB compared to the optimal Log-BCJR algorithm increasing the required transmission energy by 10%. This disadvantage of the Max-Log-BCJR outweighs its attractively low complexity, when optimizing the overall energy consumption of sensor nodes that are separated by dozens of meters.

A. MODELLING AND IMPROVEMENT PERSPECTIVES:

The IEEE 802.15.4 standard, defines the physical layer (PHY) and medium access control (MAC) sub layer specifications for low-data-rate wireless connectivity with fixed, portable, and moving devices with no battery or very limited battery consumption requirements. In this paper, the energy consumption of a practical Wireless Sensor Network scenario is modelled at the bottom three layers of the traditional networking stack the physical layer, data link layer and the MAC layer. Using these models, the optimization of energy consumption of WSN nodes is achieved using different modulation schemes to maximize the network lifetime. The approach is to design and simulate a wireless sensor network scenario consisting of 12 nodes in 500x500 m area and optimize

the energy consumption (radio energy + circuit energy) of each node using suitable modulation techniques and reduce the Energy Consumption.

B. PARALLEL TURBO-DECODER ASIC FOR 3GPP-LTE:

Turbo-decoding for the 3GPP-LTE (Long Term Evolution) wireless communication standard is among the most challenging tasks in terms of computational complexity and power consumption of corresponding cellular devices. This paper addresses 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. To highlight the effectiveness of our design-approach, we realized a 3.57mm² radix-4-based 8× parallel turbo-decoder ASIC in 0.13 μ m CMOS technology achieving 390 Mb/s. At the more Realistic 100 Mb/s LTE milestone targeted by industry.

IV. PROPOSED SYSTEM

In the proposed system the LUT log BCJR is designed with Clock gating technique which leads to reduction in Power consumption and energy consumption reduction is carried out. The circuit is converted into digital hardware.

The Propose a novel LUT-Log-BCJR architecture for energy-constrained scenarios, which avoids the wastage of energy that is inherent in the conventional architecture. Our philosophy is to redesign the timing of the conventional architecture in a manner that allows its components to be efficiently merged. This produces an architecture comprising only a low number of inherently low-complexity functional units, which are collectively capable of performing the entire LUT-Log-BCJR algorithm. Further wastage is avoided, since the critical paths of our functional units are naturally short and equally-lengthened, eliminating the requirement for additional hardware to manage them.

4.1 LUT-log-BCJR ARCHITECTURE

A number of variants of the LUT-Log-BCJR architecture of have been proposed for further increasing the decoding throughput. For example, employs parallel repetitions of the blocks to “parallel-process” the schedule. Alternatively, employs a radix-4 variant, which processes two sets of or state metrics at a time. In summary, conventional LUT-Log-BCJR architectures achieve high throughputs by employing substantial hardware, which imposes a high chip area and consequently high energy consumption, as quantified later.

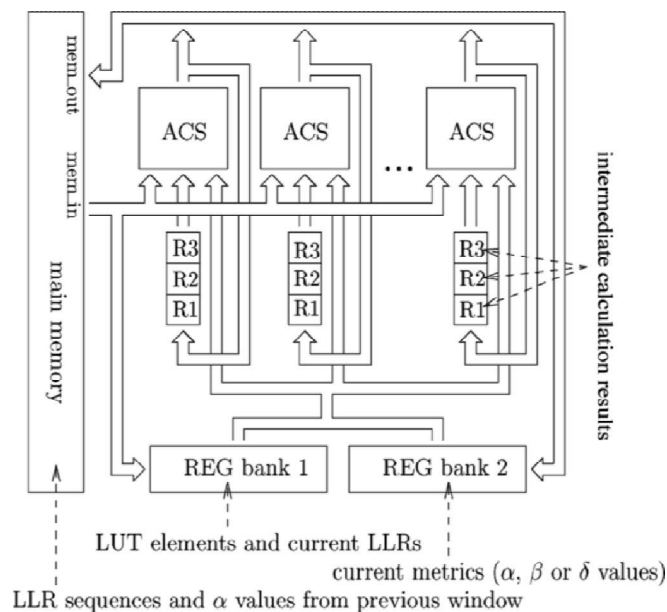
A. LOW ENERGY CONSUMPTION:

This motivates our novel architecture of which is specifically designed to have a minimal hardware complexity and hence a low energy consumption. We validate our architecture in the context of an LTE turbo decoder and demonstrate that it has an order of magnitude lower chip area, hence reducing the energy consumption of the state-of-the-art LUT-Log-BCJR implementation by 71%. Compared to state-of-the-art Max-Log-BCJR are implementations, our approach facilitates a 10% reduction.

We propose a novel LUT-Log-BCJR architecture for energy-constrained scenarios, which avoids the wastage of energy that is inherent in the conventional architecture. Our philosophy is to redesign the timing of the conventional architecture in a manner that allows its components to be efficiently merged. This produces an architecture comprising only a low number of inherently low-complexity functional units, which are collectively capable of performing the entire LUT-Log-BCJR algorithm. Further wastage is avoided, since the critical paths of our functional units are naturally shorted equally-lengthened, eliminating the requirement for additional hardware to manage them. Furthermore, our approach naturally results in a low area and a high clock frequency, which implies a low static energy consumption.

The proposed energy-efficient LUT-Log-BCJR architecture unlike conventional architectures, it does not use separate dedicated hardware for the three recursions. Instead, our architecture implements the entire algorithm using

ACS units in parallel, each of which performs one ACS operation per clock cycle. Furthermore, the proposed architecture employs a twin-level register structure to minimize the highly energy-consuming main-memory access operations. At the first register level, each ACS unit is paired with a set of general purpose registers R1, R2, and R3. These are used to store intermediate results that are required by the same ACS unit in consecutive clock cycles. For example, this allows the four ACS operations equivalent to a calculation to be performed in four consecutive clock cycles using a single ACS unit, as the second register level comprises REG bank 1 and REG bank 2 of Fig. 3, which are used to temporarily store the LUT-Log-BCJR variables between consecutive values of the bit index during the recursions decoding processes.



The REG bank 1 comprises registers for the priority LLRs and dummy registers for the required LUT constants. Meanwhile, the sets of metrics are stored in REG bank 2. The main memory stores all the required a priority LLR sequences and extrinsic LLR sequences during the decoding process and the state metrics from the previous window, which facilitates the processing of the entire LUT-Log-BCJR algorithm. Since the proposed architecture supports a fully parallel arrangement of an arbitrary number of ACS units, it may be readily applied to any LUT Log-BCJR decoder, regardless of the specific convolutional encoder parameters employed.

B. ACS UNIT:

The goal of the add-compare-select unit is to add the branch metric for a trellis edge to the path metric of the present trellis state to create a new path metric for a next state in the next trellis stage. This metric is then compared to the computed path metric from the competitive path to determine the survivor for the next state. The ACS operation must be performed for at most N_{max} present state path metrics for each trellis stage.

We propose the novel low-gate-count ACS unit performs one ACS operation per clock cycle. The control signals of the ACS unit are provided by the operation Code, which can be used to perform the functions listed in Table. Note that the operation code approximates the absolute difference between two operands, as required by (2). Its result is equivalent. In the two's complement operand representation employing fraction bits, this is equivalent to decrementing the binary representation of, which is equivalent to subtracting. Note that a simpler

ACS unit implementations facilitated by this deliberately introduced inaccuracy, which can be trivially cancelled out during the calculation. More specifically, a calculation can be performed with the following four operations, which store intermediate results in the registers.

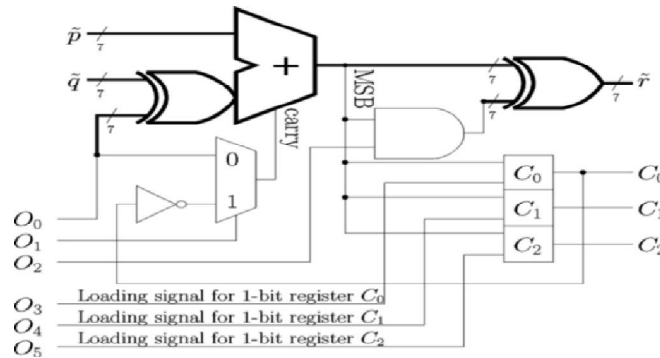


Fig 1 ACS unit.

O	function
000000 ₂	$\tilde{r} = \tilde{p} + \tilde{q}$
100000 ₂	$\tilde{r} = \tilde{p} - \tilde{q}$
101100 ₂	$\tilde{r} = \begin{cases} \tilde{p} - \tilde{q} & \text{if } \tilde{p} \geq \tilde{q} \\ (\tilde{q} - \tilde{p}) - 0.25 & \text{if } \tilde{p} < \tilde{q} \end{cases}$ $C_0 = \begin{cases} 0_2 & \text{if } \tilde{p} \geq \tilde{q} \\ 1_2 & \text{if } \tilde{p} < \tilde{q} \end{cases}$
110010 ₂	$\tilde{r} = \begin{cases} \tilde{p} - \tilde{q} & \text{if } C_0 = 0_2 \\ \tilde{p} - \tilde{q} - 0.25 & \text{if } C_0 = 1_2 \end{cases}$ $C_1 = \begin{cases} 0_2 & \text{if } \tilde{r} \geq 0 \\ 1_2 & \text{if } \tilde{r} < 0 \end{cases}$
110001 ₂	$\tilde{r} = \begin{cases} \tilde{p} - \tilde{q} & \text{if } C_0 = 0_2 \\ \tilde{p} - \tilde{q} - 0.25 & \text{if } C_0 = 1_2 \end{cases}$ $C_2 = \begin{cases} 0_2 & \text{if } \tilde{r} \geq 0 \\ 1_2 & \text{if } \tilde{r} < 0 \end{cases}$

Table Operations of the ACS Unit

C. PROPOSED ENERGY-EFFICIENT LUT-LOG-BCJR ARCHITECTURE:

The proposed energy- efficient LUT-Log-BCJR architecture is unlike conventional architectures, it does not use separate dedicated hardware for the three recursions. Instead, our architecture implements the entire algorithm using ACS units in parallel, each of which performs one ACS operation per clock cycle. Furthermore, the proposed architecture employs a twin-level register structure to minimize the highly energy-consuming main-memory access operations.

D. CONTROLLER DESIGN:

The proposed architecture can be readily applied to any LUT-Log-BCJR decoder, regardless of the corresponding convolution encoder parameters employed. This is achieved by specifically designing a controller for the LUT-Log-BCJR decoder. To exemplify this, we designed a controller for a sliding-window implementation of the LTE turbo code's LUT-Log-BCJR decoder, which corresponds to an encoder having memory elements. Since the proposed architecture employs parallel ACS units, it facilitates the parallel processing of or state metrics at a time. The dynamic energy consumption, the reduced throughput implies increased static energy consumption, particularly in the case of

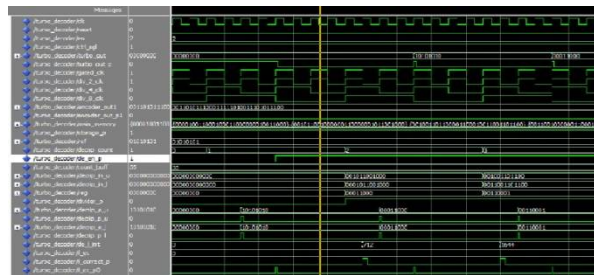
high-density technologies. Furthermore, the lengthening of the critical path implies a greater variety of path lengths, particularly since the backward recursion path is significantly longer than those of the other recursions.

This in turn implies that a greater fraction of the static energy consumption can be considered to be wasted, by giving short data paths more time to settle than necessary. In summary, efforts to slow down the conventional LUT-Log-BCJR architecture result in energy wastage, which cannot be avoided without completely redesigning the architecture.

V. CLOCK GATING

The disadvantage of the clock gating enable signal is generated by user software. If the IP can generate the enable signal automatically by itself, the user software will be no need. Now, we analyse the IP model first. Any IP core (Except combinational circuit) can be modelled as an Finite State Machine (FSM) which includes several states: Idle, Ready, Run and so on, as shown in the dashed box of Fig. Each circle is a state and each arrow shows transition from a state to another. The state and the transition will be mapped to the sequential circuit and the combinational circuit respectively by synthesis. When an IP core finishes the work, it enters the idle state (IS) and stay there until it accepts another request from the system bus. We call each of those states except IS working state (WS). Hence, all states in an IP core are classified to two classes: IS and WS. When an IP core stays in the IS for some cycles, it does not need the clock.

VI. RESULT



TABLE

PARAMETER	EXISTING	PROPOSED
AREA	7581um	6912um
POWER	88mW	14mW

VII. CONCLUSION

We demonstrated that upon aiming for a high throughput, conventional LUT-Log-BCJR architectures may have wasteful designs requiring high chip areas and hence high energy consumptions. However, in energy-constrained applications, achieving a low energy consumption has a higher priority than having a high throughput. This motivated our low-complexity energy-efficient architecture, which achieves a low area and hence a low energy consumption by decomposing the LUT-Log-BCJR algorithm into its most fundamental ACS operations. In addition, the proposed architecture may be readily reconfigured for different turbo codes or decoding algorithms. We validated the architecture by implementing an LTE turbo decoder, which was found, in Table III, to have an order-of-magnitude lower area than conventional LUT-Log-BCJR decoder implementations and an approximately 71% lower energy consumption of 0.4 n J/bit/iteration. Compared to state of the art Max-Log-BCJR implementations, our approach facilitates a 10% reduction in the overall energy

consumption at transmission ranges above 58 m. Furthermore, we demonstrated that our implementation has a throughput of 1.03 Mb/s, which is appropriate for energy-constrained applications, such as in environmental monitoring WSNs.

REFERENCES

1. J. Zhang and M. P. C. Fossorier, "Shuffled iterative decoding," *IEEE Trans. Commun.*, vol. 53, no. 2, pp. 209–213, Feb. 2014.
2. C.-C. Wong and H.-C. Chang, "Reconfigurable turbo decoder with parallel architecture for 3GPP LTE system," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 7, pp. 566–570, Jul. 2013.
3. C.-H. Lin, C.-Y. Chen, and A.-Y. Wu, "Area-efficient scalable MAP processor design for high-throughput multistandard convolutional turbo decoding," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 2, pp. 305–318, Feb. 2013.
4. G. Wang, H. Shen, Sun, J. R. Cavallaro, A. Vosoughi, and Y. Guo, "Parallel interleaver design for a high throughput HSPA +/LTE multistandard turbo decoder," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 5, pp. 1376–1389, May 2014.
5. J. S. Kunze, E. Matus, G. Fettweis, and T. Kobori, "A 'multi-user' approach towards a channel decoder for convolutional, turbo and LDPC codes," in *Proc. IEEE Workshop Signal Process. Syst. (SIPS)*, Oct. 2015, pp. 386–391.
6. C. Condo, M. Martina, and G. Masera, "VLSI implementation of a multimode turbo/LDPC decoder architecture," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 6, pp. 1441–1454, Jun. 2013.
7. J. V. Lapotre et al., "Optimizations for an efficient reconfiguration of an ASIP-based turbo decoder," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2013, pp. 493–496.
8. P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 2, Jun. 2014, pp. 1009–1013.
9. C. Schurgers, F. Catthoor, and M. Engels, "Memory optimization of MAP turbo decoder algorithms," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 2, pp. 305–312, Apr. 2015.
10. O. Muller, A. Baghdadi, and M. Jézéquel, "Parallelism efficiency in convolutional turbo decoding," *EURASIP J. Adv. Signal Process.*, vol. 2010, p. 927920, Dec. 2015.