

Turbo Codes Network-On-Chip-Implementation

P.Ajay, S.Santhosh, S.Sarath

Department of Computer Science and Engineering, Sri Muthukumaran Institute of Technology, Chennai, India.

Abstract – Wireless communication at near-capacity transmission throughputs is facilitated by employing sophisticated Error Correction Codes (ECCs), such as turbo codes. However, real time communication at high transmission throughputs is only possible if the challenge of implementing turbo decoders having equally high processing throughputs can be overcome. This motivates the implementation of turbo decoders using Networks-on-Chip (NoCs), which facilitate flexible and high throughput parallel processing. However, turbo decoders conventionally operate on the basis of the Logarithmic Bahl-Cocke-Jelinek-Raviv (Log-BCJR) algorithm, which has an inherently-serial nature, owing to its data dependencies. This limits the exploitation of the NoC's computing resources, particularly as the size of the NoC is scaled up. Motivated by this, we propose a novel turbo decoder algorithm, which eliminates the data dependencies of the Log-BCJR algorithm and therefore has an inherently-parallel nature. We show that by jointly optimizing the proposed algorithm with the NoC architecture, a significantly improved utility of the available computing resources is achieved. Owing to this, our proposed turbo decoder achieves a factor of up to 2.13 higher processing throughput than a Log-BCJR benchmark.

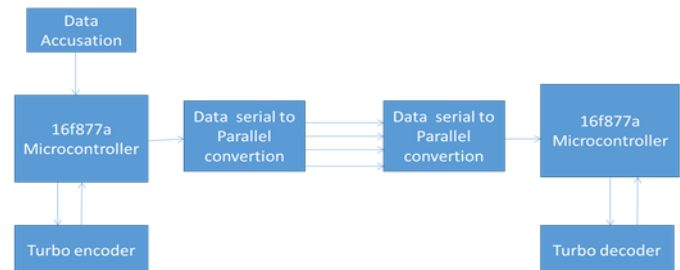
Index Terms – Turbo Codes, Network, Chip, ECC.

1. INTRODUCTION

The problem of IC security is extremely important today, since ICs are involved in all critical aspects of our lives. For economic reasons, nearly all ICs are fabricated by foreign foundries and they include intellectual property (IP) cores supplied by many third-party IP providers. In addition, they rely on outsourced design and test services, and use automation tools from many different vendors. Such a design and manufacturing process provides an adversary with many opportunities to insert Trojanhorse logic to sabotage the mission of an IC used in critical applications. We will show that today there are no reliable methods that can guarantee the pre-deployment detection of Trojan intrusions. This fact has large national-security implications: a Trojan attack can create havoc in the basic civilian infrastructure (electric grid, communication, and banking networks), sabotage critical missions, disable weapon systems, or provide back-door access to otherwise highly-secure systems.

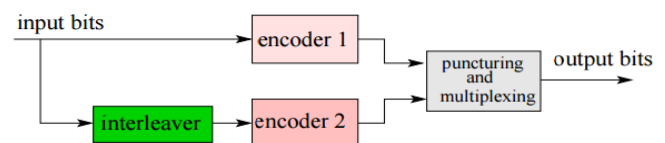
2. RELATED WORK

A turbo based system implemented in the VLSI to emulate the performance of the network on chip system. Logarithmic Bahl-Cocke-Jelinek-Raviv (Log-BCJR) algorithm is used. This limits the exploitation of the NoC's computing resources, particularly as the size of the NoC is scaled up.



2.1. VHDL implementation of a turbo decoder with log-MAP-based iterative decoding

Turbo code is one of the most significant achievements in coding theory during the last decade. By concatenating two simple convolutional codes in parallel, it has been shown that transmission systems employing turbo codes could offer near-capacity performance. More importantly, by employing a suboptimal iterative decoding structure with soft-in/soft-out (SISO) maximum a posteriori-probability (APP) decoding algorithm, the near-capacity performance is achievable at a feasible decoding complexity.

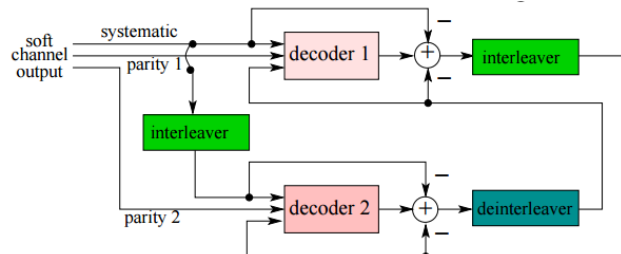


Given the outstanding performance of turbo code, the challenge now is to implement it into various communication systems at affordable decoding complexity using current very large scale integration (VLSI) technologies. In this paper, we first investigated the existing four different turbo decoding algorithms. Comparisons of both their performances and implementation complexities were performed. Log-maximum a posteriori (MAP) -based turbo decoding was found to offer the best performance-complexity compromise. A register-transfer-level (RTL) 12-bit fixed-point turbo decoder based on Log-MAP algorithm was then designed and simulated using VHDL as the hardware description language. The implemented RTL model was verified by comparing its performances with those obtained from a C-language implementation of the same turbo decoder.

2.1.2 VLSI Implementation of a Multi-Mode Turbo/LDPC Decoder Architecture

Flexible and reconfigurable architectures have gained wide popularity in the communications field. In particular,

reconfigurable architectures for the physical layer are an attractive solution not only to switch among different coding modes but also to achieve interoperability. This work concentrates on the design of a reconfigurable architecture for both turbo and LDPC codes decoding.



The novel contributions of this paper are: i) tackling the reconfiguration issue introducing a formal and systematic treatment that, to the best of our knowledge, was not previously addressed and ii) proposing a reconfigurable NoC-based turbo/LDPC decoder architecture and showing that wide flexibility can be achieved with a small complexity overhead. Obtained results show that dynamic switching between most of considered communication standards is possible without pausing the decoding activity. Moreover, post-layout results show that tailoring the proposed architecture to the WiMAX standard leads to an area occupation of 2.75 mm² and a power consumption of 101.5 mW in the worst case.

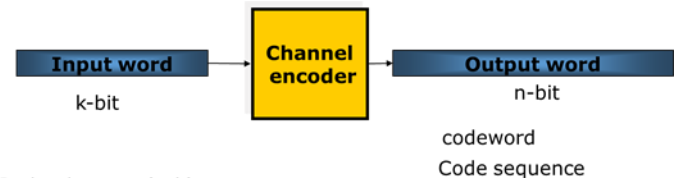
2.1.3 Implementation Trade-Offs of Soft-Input Soft-Output MAP Decoders for Convolutional Codes

Soft-input soft-output (SISO) maximum a-posteriori (MAP) decoders for convolutional codes (CCs) are an integral part of many modern wireless communication systems. Specifically, SISO-MAP decoding forms the basis for turbo decoders, as, e.g., specified for HSDPA or 3GPP-LTE, or for iterative detection and decoding in multiple-input multiple-output wireless systems, such as IEEE 802.11n. In this paper, we investigate the silicon-area, throughput, and energy-efficiency trade-offs associated with SISO-MAP decoders based on the algorithm developed by Bahl, Cocke, Jelinek, and Raviv (BCJR). To this end, we develop radix-2 and radix-4 architectures for high-throughput SISO-MAP decoding of CCs having 4, 8, 16, 32, and 64 states and present corresponding implementation results in 180 nm, 130 nm, and 90 nm CMOS technology. We validate technology-scaling rules and finally demonstrate the use of the presented trade-off analysis by identifying the key design parameters for parallel turbo-decoder implementations.

3. PROPOSED MODELLING

Implements Turbo codes in Wireless sensor network in Embedded platform. Parallel data transmission procedure is implemented to achieve maximum throughput with limited resource.

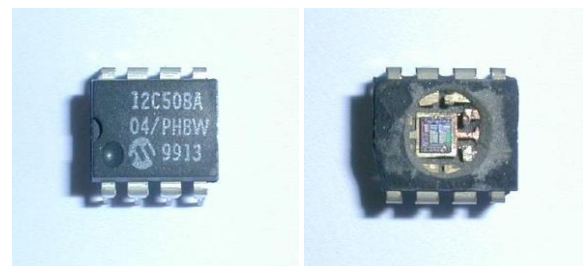
Weak channels between the nodes in the wireless sensor network (WSN) can cause reception of erroneous packets. Retransmission mechanisms are mainly used to tackle the problem of erroneous reception in WSN communication protocols. Weak channels can cause high number of retransmissions in order to deliver a packet correctly, which will consume high energy of both the transmitting and the receiving nodes. Parallel transmission reduces the risk of packet loss.



$$\text{Redundancy} = (n-k)$$

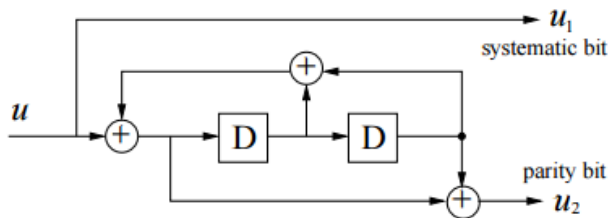
$$\text{Code rate} = k/n$$

We propose our novel NoC-optimized algorithm for the implementation of the LTE turbo decoder. This is achieved by scheduling the computations of the algorithmic blocks described in Section II-D in a manner that is particularly suited to NoC operation. This is possible, since the algorithmic blocks of Section II-D are not bound by a strict scheduling that requires their operation according to forward and backward recursions, as in the conventional Log-BCJR turbo decoding algorithm. In our previous work this property was exploited to achieve fully-parallel operation, in which all algorithmic blocks are operated concurrently, in every clock cycle. More specifically, while introduced the fully-parallel turbo decoding algorithm, considered the implementation of that algorithm in Application Specific Integrated Circuit (ASIC), Field Programmable Gate Array (FPGA) and Graphical Processing Unit (GPU) applications, respectively.



In contrast to this previous work, the novel scheduling proposed in this section is specifically designed for mapping onto an NoC, in which each ASIP performs the operations of Section II-D for different algorithmic blocks in different Clock Cycles (CCs). In particular, the proposed scheduling self-regulates the exchange of information within the NoC, in order to avoid congestion, where the operation of each ASIP in the NoC is adapted in response to the arrival of information delivered by the NoC. More specifically, rather than scheduling the operation of the algorithmic blocks according to strict

forward and backward recursions, the proposed turbo decoder algorithm schedules their operation according to the exchange of information within the NoC. Here, the operation of the algorithmic blocks generates information for delivery over the NoC while the delivery of that information to the connected algorithmic blocks stimulates their operation and so on. In this way, the processing stimulates the networking and the networking stimulates the processing, causing the schedule to grow organically.



As described in Section II-B, the iterative decoding process of the proposed turbo-decoding algorithm may be completed using an NoC, comprising an interconnected network of Intellectual Property (IP) cores. Here, each IP core is responsible for the operation of a different subset of the first N algorithmic blocks. We refer to these subsets as windows, with each window comprising K number of adjacent algorithmic blocks, as shown in Figure 3. We assume that the IP cores are sufficiently powerful or are specifically designed for requiring only a single clock cycle for completing the processing for a single algorithmic block, as detailed in Section II-D. Note however that this implies that multiple algorithmic blocks within the same window cannot be operated concurrently. During the first $(2K-1)$ CCs, the iterative decoding process is initialized by performing a forward recursion and then a backward recursion within each window of the upper row. More specifically, as shown in the forward recursion invokes the operations of Section II-D for the j th algorithmic block in each window of the upper row during the j th clock cycle, where $j = 1, 2, 3, \dots, K-1$. Following this, the backward recursion invokes the operations of Section II-D for the $(2K-j)$ th algorithmic block during the j th clock.

4. RESULTS AND DISCUSSIONS

Monte Carlo simulations were conducted to evaluate the performance of the proposed algorithm using different BCH codes as code components. The encoded bits are BPSK modulated and serially transmitted through an AWGN channel. The simulation results as a function of E_b/N_0 are presented. The BER for the standard HIHO TPC is presented for 1, 2, and 5 iterations. No noticeable gain is achieved by increasing the number of iterations beyond 5 which is typical for HIHO TPCs [8]. The results obtained for a BCH(31, 21, 5)₂, with the reliability threshold $\gamma = 1$. As depicted in this figure, the proposed non-sequential decoding provides an extra 0.5 dB of coding gain at BER of 10^{-6} compared to the standard HIHO

decoder using 5 iterations. The performance improvement observed at high values of E_b/N_0 is more significant since the reliability of the received sequences with a single error increases when P_{ch} decreases as depicted in Fig. 1. In addition, this figure shows the BER of a TPC using 4 SISO decoding iterations with and without input signal quantization. These results show that the extra coding gain offered by the SISO decoder is about 2.5 dB without quantization, while it drops to 1.5 and 1.1 dB with 3 and 2-bits quantization, respectively. The 1-bit quantization case is also considered in Fig. 3 where the SISO decoder soft input signal is replaced with a two-levels signal [16]. It can be observed from Fig. 3 that the TPC SISO decoder with 1-bit quantization slightly outperforms the proposed HIHO decoder for BERs less than 3.5×10^{-4} . For smaller BERs, the proposed HIHO TPC decoder outperforms the SISO decoder due to the error floor that appears at low BERs. The error floor effect becomes negligible for codes with long codeword lengths and high data rates. In such cases, the SISO decoder using 1-bit quantization slightly outperforms the proposed decoder for the entire range of E_b/N_0 . However, it should be noted that the SISO decoders computational complexity is almost fixed regardless the number of quantization levels. Therefore, the computational complexity of the proposed decoder is substantially smaller than the SISO decoder with 1-bit quantization. The BERs for a TPC constructed using BCH(63, 36, 11)₂ are presented in Fig. 4 using $\gamma = 4$. The obtained simulation results suggest that the extra coding gain achieved by the new decoding algorithm decreases as the codeword length increases, which is consistent with the results presented.

5. CONCLUSION

A new non-sequential decoding algorithm is proposed to improve the performance of iterative TPCs using HIHO decoding. The new algorithm utilizes the reliability information embedded in the received code components to avoid error amplification. Simulations results show that the proposed algorithm offers a substantial improvement over traditional sequential decoding with negligible additional complexity. Furthermore, the proposed HIHO decoder renders it self as an efficient alternative for SISO decoders in certain practical environments since its computational complexity is substantially smaller than that of the SISO decoders and the coding gain loss is about 1.5 dB.

REFERENCES

- [1] M. Brejza, L. Li, R. Maunder, B. Al-Hashimi, C. Berrou, and L. Hanzo, "20 Years of Turbo Coding and Energy-Aware Design Guidelines for Energy-Constrained Wireless Applications," IEEE Commun. Surv. Tutorials, vol. PP, no. 99, pp. 1-1, jun 2015. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7131434>
- [2] P. Hailles, L. Xu, R. Maunder, B. Al-Hashimi, and L. Hanzo, "A Survey of FPGA-based LDPC Decoders," IEEE Commun. Surv. Tutorials, no. c, pp. 1-1, 2015. [Online]. Available:

- <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7360870>
- [3] A. Abbasfar, D. Divsalar, and Kung Yao, "Accumulate repeat accumulate coded modulation," in IEEE MILCOM 2004. Mil. Commun. Conf. 2004., vol. 1. IEEE, 2004, pp. 169–174. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1493264>
- [4] ETSI, LTE; Evolved Universal Terrestrial Radio Access (EUTRA); Multiplexing and Channel Coding, ETSI Std. v11.1.0, feb 2013.
- [5] L. Benini and G. De Micheli, "Networks on chips: a new SoC paradigm," IEEE Computer, vol. 35, no. 1, pp. 70–78, 2002.
- [6] H. Moussa, "On-chip communication network for flexible multiprocessor turbo decoding," in Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on, 2008, pp. 1–6.
- [7] M. Martina and G. Masera, "Turbo noc: A framework for the design of network-on-chip-based turbo decoder architectures," Circuits and Systems I: Regular Papers, IEEE Transactions on, vol. 57, no. 10, pp. 2776–2789, 2010.