New Low Area NB-IoT Turbo Encoder Interleaver by sharing resources

Ali Abdelbaky
Digital IC Track
Information Technology Institute
Cairo, Egypt
abdelbaky1992@gmail.com

Hassan Mostafa
Centre for Nanoelectronics and Devices
AUC and Zewail City of Science and Tecnology
New Cairo, Egypt
hmostafa@waterloo.ca

***Abstract*- The interleaver is an important component of turbo coding and has strong impact on its error correction. The design of the NB-IoT Turbo Encoder Interleaver has a big challenge to achieve the required error correction and meet IoT applications low cost and low power constraints. In this paper, a low area interleaver is implemented to achieve the required error correction with 5% area saving by sharing resources. Sharing resources in addition to additional saving power techniques can reduce the interleaver power consumption reasonably.**

I. INTRODUCTION

The variant mobile broadband services have raised the customers' needs and the service operators are forced to direct their efforts towards providing higher capacity and higher data rates for their customers. For this reason, the 3GPP has developed a new standard of wireless communications, the Long-Term Evolution (LTE) [1].The flat, all-inclusive nature of LTE's all-IP architecture makes it ideal for IoT applications. Moreover, LTE provides built-in security along with robust and scalable traffic management capabilities in addition to its spatially wide range [2]. Recently, IoT became a main topic relating to the LTE and makes the 3GPP sets special standard specifications for IoT application. These standard specifications represent a reduced version of LTE with lower data rate for IoT applications rather than the high data rates of LTE. This version is called Narrow Band IoT (NB-IoT) [3].

The LTE radio interface protocol architecture is shown in Fig. 1. The physical layer interfaces the Medium Access Control layer (MAC layer) and the Radio Resources Control layer (RRC layer) [1]. The physical layer offers transport channels to the MAC layer. These channels are characterized by how the information is transferred over the radio interface. The physical layer has to do many functions to provide data transport services to higher layers. These functions require many blocks to be done, as shown in the physical layer block diagram in Fig. 2. One of these functions is the Forward Error Correction (FEC) or channel coding. The FEC function is to make the sent data more

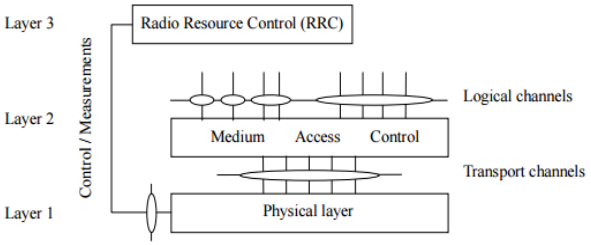

Fig. 1. LTE (E-UTRA) radio interface protocol architecture

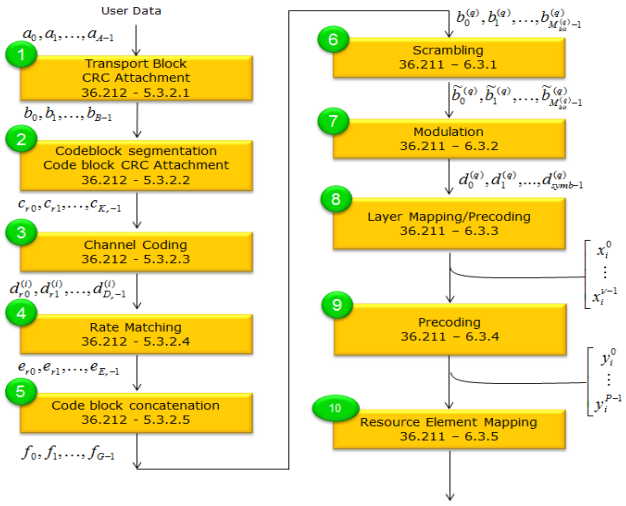

Fig. 2. Physical layer block diagram

immune to errors which can occur due to unreliable and noisy channels. This can be achieved by sending redundant data and the receiver recognizes only the portion of the data that contains no apparent errors. The FEC which is used for LTE communications is the Turbo Coding due to its high performance (Near Shannon channel coding). That means we need a Turbo Encoder in the transmitter and a Turbo Decoder at the receiver. This paper concerns about the Turbo Encoder for User Equipment (UE) regarding that the Turbo Decoder will be implemented in the base station.

The Interleaver is an important block inside the Turbo Encoder because of its strong effect on the error correction [6].The function of the interleaver is to permute the bits of

the data blocks to avoid burst errors. This permutation will affect the channel coding as will be shown when talking about the Turbo Encoder in Section II. Section III presents the interleaver design issues and block diagram. Section IV presents results and discussion. finally, the conclusion remarks are presented in Section V.

II. TURBO ENCODER

The LTE Turbo Encoder scheme is Parallel Concatenated Convolutional Code (PCCC) with two 8-states constituent encoders and one internal interleaver [3], as shown in Fig. 3, with transfer function

$$G(D) = [1, \frac{1+D^2+D^3}{1+D+D^3}]. \quad (1)$$

The code rate is 1/3 because the Turbo Encoder generates three outputs x_k, z_k and z'_k which are the systematic bit, the parity bit and the interleaved parity bit respectively. The fourth output relates to the termination process which indicates that the encoding process has been terminated by sending specific bits at the end of the encoded data. The interleaver generates permuted (interleaved) bits whose probability of error is fully independent of the probability of error of the systematic and the parity bits. This will increase the error correction of the Turbo Code. Notice that the scheme of constituent encoder is Recursive Convolutional Code (RSC) which consists of flip-flops and XOR gates. As a result, the main obstacle in the design of the Turbo Encoder is the interleaver.

The algorithm of the Turbo Encoder is shown in Fig. 4. In this algorithm, you need two blocks of memory to write/read the systematic and the interleaved bits. This paper presents a proposed algorithm, shown in Fig. 5, which depends on changing the function of the interleaver and sharing resources to save one block of memory.

III. INTERLEAVER DESIGN ISSUES

The interleaver receives the data block as a bit stream. Its function is to store the bits and transmit them again but bits should be interleaved. The index of the interleaved bits is derived from the polynomial

$$\pi(i) = (f_1.i + f_2.i^2) MOD\ K \quad (2)$$

where $\pi(i)$ is the interleaved index and i is the input index (ordered bits index) [3]. K is the transport block size whose max value is 1000 bits [4]. f_1, f_2 values depend on K as summarized in [3] and are implemented logically in the Look Up Table (LUT) as shown in the interleaver block diagram in Fig. 6. The dividend $(f_1.i + f_2.i^2)$ is implemented using three behavioral RTL codes of multipliers. The modulus function, which is responsible for providing reasonable diffusion of the indices, is implemented using successive subtraction [5]. The input index i is generated by the input index counter, which is

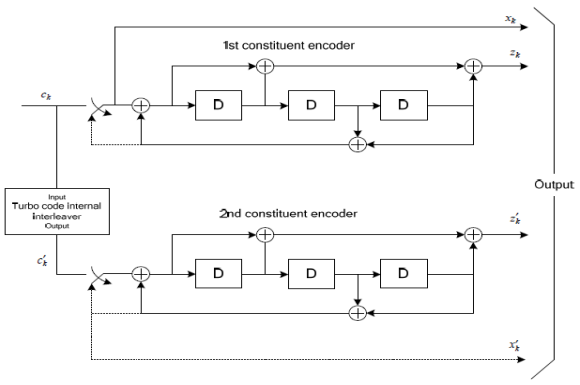

Fig. 3. Turbo Encoder block diagram

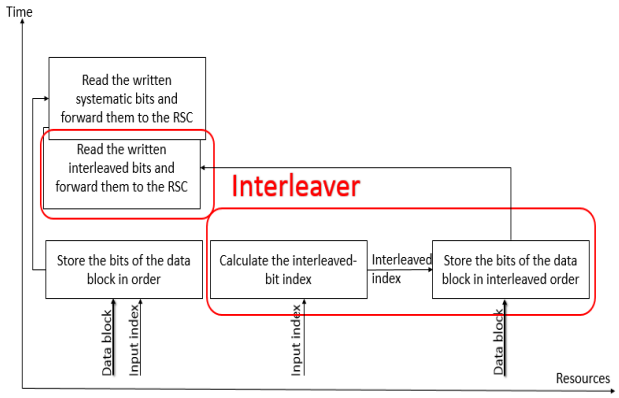

Fig. 4. Turbo Encoder algorithm

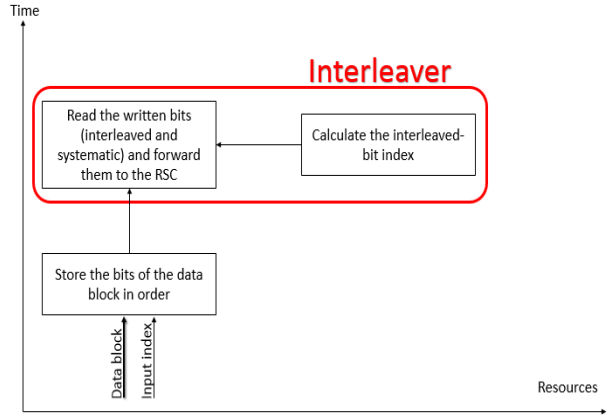

Fig. 5. Proposed Turbo Encoder algorithm

10-bits counter that overflows according to the value of K. The size of the RAM block is 128 bytes. It has a single port for writing bytes and dual port for reading. Because we need to read/write a single bit each clock cycle, there are some additional blocks. The shift register is a 9-bits shifter. The RAM writes the eight LSBs and the extra MSB exists in order not to miss a bit while writing a byte in the RAM.

Also, the Wait Eight Counter is a 3-bits counter which generates the signal "Count-Eight" each eight clock cycles to start writing the byte in the RAM. The addresses of the

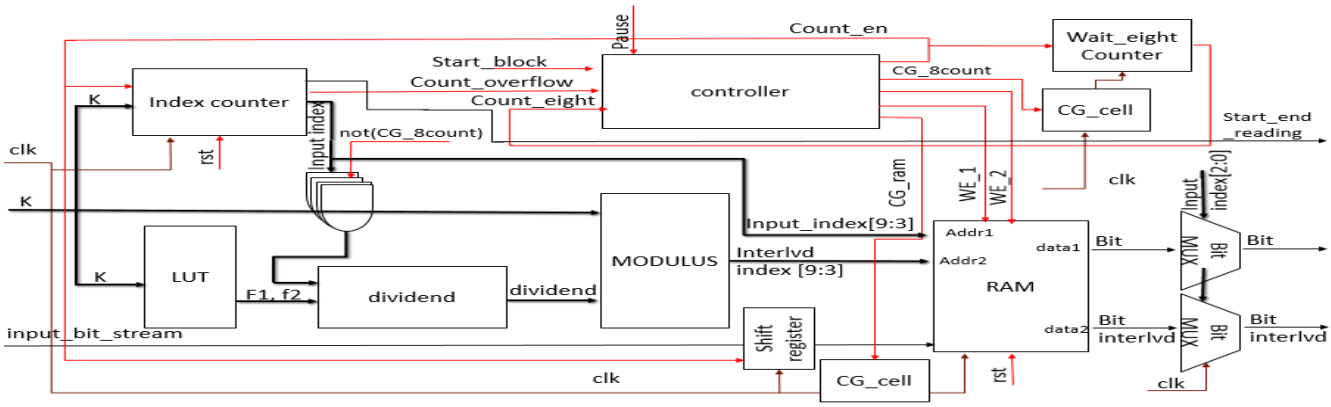

Fig. 6. Interleaver block diagram

RAM are 10-bits addresses. They are divided into two parts, the seven MSBs to read/write bytes and the three LSBs are the selection lines of the 8x1 MUX. This MUX is responsible for choosing which bit of the recent byte will be read. Notice that these three LSBs are delayed one clock cycle to read the right bit.

The Control Unit (CU) is a 6-states mealy FSM as shown in Fig. 7. When the CU senses the start-block pulse, it departs its idle state. The wait-Eight state exists to shift eight bits of the input-bit-stream in the shift register and the Write-Byte state to write the shifted eight bits. When the index counter overflows, it moves to the Read-Bits state. The Read-bits state exists to read the systematic and the interleaved bits from the RAM. When the index counter overflows again, the it moves back to its idle state. Two Pause states to stop reading or writing when the MAC layer asserts the pause signal. The two clock gating cells and the AND gates are for the purpose of saving power. We will discuss how we can save power the next Section. The clock gating cell is implemented as shown in Fig. 8, where the Clock-Gating signal (CG) should be inserted to a negative edge flip-flop to avoid occurrence of glitches and timing violations.

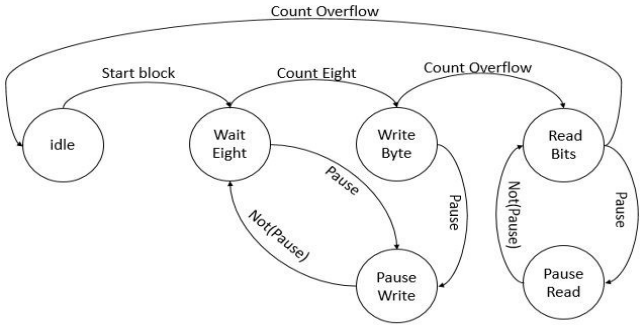

Fig. 7. Mealy finite state machine of the Interleaver

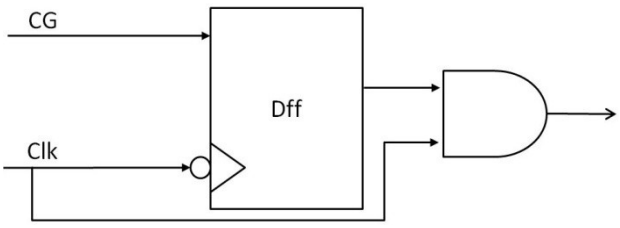

Fig. 8. Clock gating cell

IV. RESULTS AND DISSCUSSION

IoT applications require saving power and cost (area in silicon). As a result, some techniques should be added to the design to achieve these requirements. The original Turbo Encoder needs two blocks of RAM. One for writing/reading the systematic bits in order and the other for writing/reading the interleaved bits in order. In this proposed design, only one block of dual port RAM is used. This block is used to write the systematic bits in order but while reading, the systematic bits are read in order on the first port and out of order (interleaved) on the second port. As a result, one block of 128 bytes RAM is saved. The Modulus block consists of 18 successive subtraction stages which is the minimum guaranteed number of stages [5]. In this proposed design, the original stage was developed by getting rid of the subtractors and sufficing to implement the comparator which contains subtractor inside, as shown in Fig. 9. The division by 2 is, simply, shifting right. The NOR gate are used to sense that one bit is high at least. The original design depends on shifting till find the MSB of the dividend and this will result in adding many extra successive-subtraction stages. The NOR gates can be eliminated and a certain number of stages can be set if the maximum value of the dividend and the maximum value of the divisor K are known. The maximum value of the dividend in the NB-IoT case is a 27-bits number [3] and of K is 10-bits number [4]. As a result, the maximum number of stages will be 18 [5]. K is applied to the first successive subtraction stage then shifted left and applied to the next stage and so on. The K that will be applied to the last stage will represent a 27-bits number. This method will fail if the value of K is small and the dividend is too large but this case will not happen in NB-IoT as the value of the dividend relates to the value of K.

Focusing on saving the dynamic power leads to focusing on

reducing the switching activity of the blocks' ports. The Wait Eight Counter is not needed while the Reading-bits state. So, its clock is gated during this state. Also, the RAM block is not needed in the Wait-Eight state and its clock is gated in this state. In the Writing-Byte and the Wait-Eight states the interleaved index is not needed and, as a result, the applied input index is not needed. So, the applied input index is gated with the inversion of the clock gating signal of the Wait Eight Counter to reduce the switching activity of the interleaved index. Moreover, there is only one signal "Start-end-reading" to determine when the Encoder should start/end reading the bits from the Interleaver instead of two.

The proposed design requires the use of Embedded Dual Port RAM. The Embedded Dual Port RAM is already presented in 130nm technology by many fabs such as INFINEON Inc.

The Interleaver is compared with a MATLAB model. The RTL and the model are stimulated with the same test vectors and systematic and interleaved outputs of both are compared. The critical block in the design, the Modulus block, was verified on MODELSIM using test vectors generated by MATLAB covering almost all the cases of K. The Design was synthesized by DESIGN COMPLIER using TSMC130nm technology. The timing verification is achieved using tough timing constraints. The input delay, the output delay, and the maximum transition is 45% of the clock period and the maximum load capacitance is 100f F. Moreover, the area is optimized. The operating frequency is 8MHz. If pipelining flip-flops are added after the output of the dividend, the operating frequency can be 25MHz but with one clock cycle latency. The authors investigated that this Interleaver represents about 98 % of the estimated total area of the Turbo Encoder. The total area is 61478 μm^2 and the estimated area of the proposed Interleaver is 56877μm^2. The combinational area is large (about 89% of the total area) due to the tough timing constraints and the existence of the three behavioral multipliers and the LUT. The estimated area of the 128 bytes RAM is about 3200 μm^2 (5% of the total area). That means the proposed algorithm reduces the cost in silicon by saving about 5% of the total area. Assuming that the previously presented power saving techniques are implemented in the two algorithms, the two algorithms will consume the same power, approximately. The comparison between the two algorithms is shown in the chart in Fig. 10.

V. CONCLUSION

In this paper, a proposed algorithm of the Turbo Encoder for NB-IoT is presented. This algorithm depends on changing the function of the interleaver and on sharing resources. Some techniques are applied in the interleaver design to reduce the switching activity of the sub-blocks' ports to

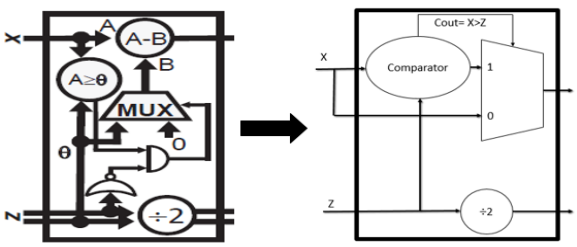

Fig. 9. Original to proposed successive subtraction stage

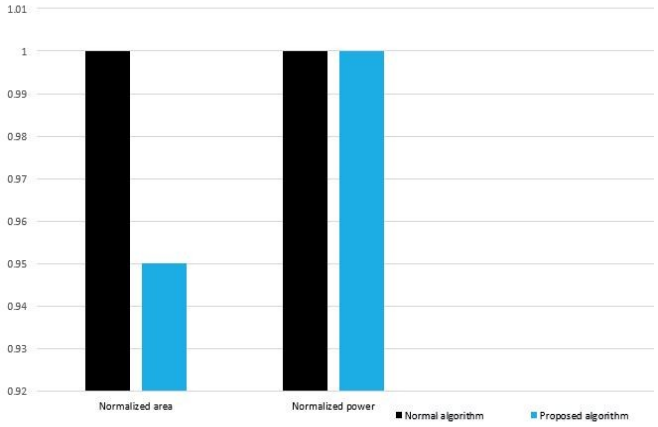

Fig. 10. Comparison between the Two algorithms

reduce the dynamic power. The proposed interleaver is optimized to overcome the challenge of the NB-IoT, it achieves the required performance, the required error correction, and can save power and cost (area).

ACKNOWLEDGMENT

This research was partially funded by ONE Lab at Cairo University, ITI, ITIDA, NTRA, NSERC, Zewail City of Science and Technology, AUC, the STDF, Intel, Mentor Graphics, SRC, ASRT and MCIT. Big thanks to Dr. Ahmed Shalaby, Assistant Professor at E-JUST, And the Digital Design Team at Siware Systems for their support.

REFERENCES

[1] 3GPP TS36.201," LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); LTE Physical layer; General description", v13.3.0, April 2017.

[2] Kimberly Tassin, "LTE PLAYS A KEY ROLE IN THE INTERNET OF THINGS", FierceWireless eBrief, July 2016.

[3] 3GPP TS36.212, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding", v14.2.0, April 2017.

[4] Y.-P. Eric Wang, Xingqin Lin, Ansuman Adhikary, Asbjörn Grövlen, Yutao Sui, Yufei Blankenship, Johan Bergman, and Hazhir S. Razaghi, "A Primer on 3GPP Narrowband Internet of Things (NB-IoT)", IEEE communications magazine, March 2017.

[5] J. T. Butler and T. Sasao, "Fast Hardware Computation of x mod z", Parallel and Distrubuted Processing Workshop and Phd Forum (IPDPSW), 2011 IEEE International Symposium on, May 2011.

[6] Ching Lung Chi, "Quadratic Permutation Polynomial Interleavers for LTE Turbo Coding", International Journal of Future Computer and Communication, vol. 2, no. 4, August 2013.