

DIFFERENT FPGA PRODUCTS BASED IMPLEMENTATION OF LTE TURBO CODE

Khadija O. Dheeb ¹, Bayan M. Sabbar ²

^{1,2} College of Information Engineering, Al-Nahrain University, Baghdad, Iraq
{khadija.omran, bayan.mahdi}@coie-nahrain.edu.iq ^{1,2}

Received:2/6/2019, Accepted:22/1/2020

Abstract- In the long- term evolution(LTE) physical layer, using turbo code is considered the core of the error-correcting code. This paper presents an implementation of LTE turbo decoding using the Log- Maximum a posteriori (MAP) algorithm with reduced number of required cycles approximately by 75% based on serial to parallel operation. Additionally, an improvement for this algorithm based on polynomial regression function is done to reduce the implementation complexity. These system implementations, are designed with 40 bit block size of the input using Xilinx System Generator (XSG) to show its applicability in real time using two approaches; Hardware Co-Simulation and HDL Netlist based on three devices, Xilinx Kintex- 7, Spartan- 6 and Artix- 7. From the hardware implementation observation, the system becomes completely real time by the user control using the switches on the board. Also, this system has taken the resources utilization from the devices less than other works.

keywords: Log MAP algorithm (MAP), Xilinx system generator (XSG), LTE turbo decoder, QPP interleaver, FPGA

I. INTRODUCTION

In the modernist world, the communication has become important among people. Thus, the demand for bandwidth from both services, providers and end users is an endless race for telecom system developers in the next generation standards of wireless communication [1], [2]. As a result, LTE has been developed to meet the demands of this time and achieve the goal of global communications mobile broadband [2].The highly LTE performance is due to its very efficient channel coding [1]. One of the most paramount fulfillments in the error- correcting coding at 1993 was the invention of turbo codes [3], which is a one of the practical codes to approach Shannon's channel capacity limit [1]. Turbo encoder of LTE consists of a concatenation of two recursive systematic convolutional (RSC) encoders linked by an interleave[4], [5], This scheme of turbo encoder is a Parallel Concatenated Convolutional Code (PCCC) [1]. Interleaving is a technology widely used in digital communication system. It penetrates the positions of a particular sequence of symbols and arranged in a different chronological order [5]. Suitable interleave in turbo code is quadratic permutation polynomial (QPP) which is present parallelism order flexible for each block size [4]. The turbo decoding functionality operation depend on the concept of soft- in soft- out (SISO) decoder by using the MAP algorithm [6]. BCJR algorithm is turbo decoder algorithm with high throughput [7] proposed by Bahl, Cocke, Jelinek and Raviv, which need great computational power, but have very robust output [1]. To get an equivalent way of computing, the BCJR algorithm has been simplified named Log- MAP and Max-log MAP that operates in logarithmic domain [3]. This paper presents a design and an implementation of turbo decoder using Log- MAP algorithm implemented with reduced number of required cycles approximately by 75%, and improving it based on polynomial regression function with 40 bit block size and minimizing the time of steps processing to a half. This system is implemented in hardware using FPGA device by the Xilinx System Generator (XSG) in Simulink Matlab 2013b and ISE 14.7 to show its applicability in real time using two approaches, Hardware Co- Simulation and HDL Netlist based on

three devices, Xilinx Kintex- 7, Spartan- 6 and Artix- 7. This paper is orderly as follows; description the LTE turbo coder in Section II and the decoder is described in section III. Section IV demonstrates the design and implementation. Finally, Sections V and VI contain the results and conclusions respectively.

II. LTE TURBO CODER OVERVIEW

A. Turbo coder

Turbo encoder consists of two RSC encoders linked by an interleave [4], [5]. The rate of turbo encoder is 1/3. Fig. 1 illustrated turbo encoder structure [8].

B. QPP interleaver

QPP interleave is suitable LTE interleave present parallelism order flexible for each block size. It is suited for the high data rates and it is used to rearrange position bit of the input sequence [4], as shown in the equation below:

$$\Pi(i) = (f_1i + f_2i^2) \bmod k \quad (1)$$

i and K are represent the index and length of the sequence of the input, respectively. $\Pi(i)$ gives the new position of permuted data. Values for f_1 and f_2 are defined differently for each frame length [8].

III. TURBO DECODER OVERVIEW

The iterative decoder which mostly used by turbo codes is formed by a parallel connection of two SISO decoders, as in Fig. 2 [3].

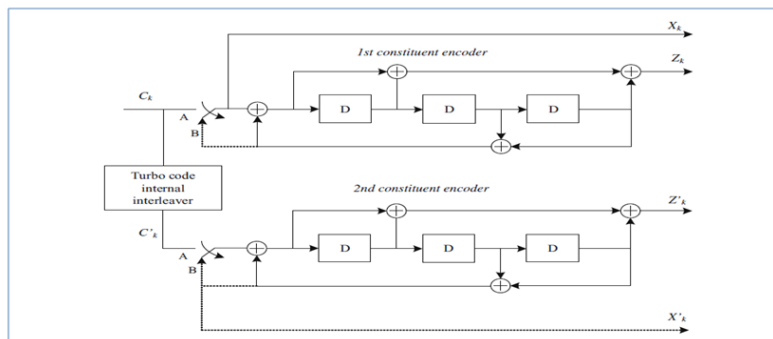


Figure 1: Schematic view of parallel turbo code used in LTE and UMTS [8]. Reproduced by permission of © 3GPP

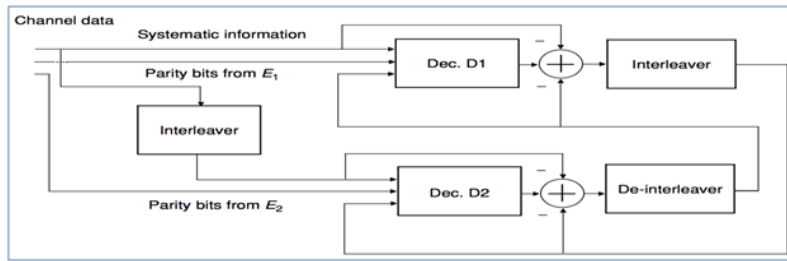


Figure 2: Turbo decoder

The iterative decoder is used to estimate a posteriori probabilities (APPs) $Pr(u_k|y)$. When the value of APPs are known, this makes the optimal decisions on the bits u_k via the algorithm maximum a posteriori (MAP) [9]. In decoder make a decision by comparing log likelihood ratio (LLR) to zero. Thus:

- when sign of the LLR ($LLR \geq 0 \rightarrow 0$), and ($LLR < 0 \rightarrow 1$) [7].

$$L(u_k|y) = \log \frac{APP(i_l = 1)}{APP(i_l = -1)} = \log \frac{\sum_{s^+} P(s', s, y)}{\sum_{s^-} P(s', s, y)} \quad (2)$$

A. Log MAP algorithm

The BCJR algorithm derives the symbol- per- symbol a posteriori LLR $L(u_k|y)$, which leads to the following equation:

$$L(u_k|y) = \log \frac{\sum_{s^+} \alpha_l(s') \gamma_l(s', s) \beta_{l+1}(s')}{\sum_{s^-} \alpha_l(s') \gamma_l(s', s) \beta_{l+1}(s')} \quad (3)$$

Because the complexity of this algorithm in the implementation, a modified version by an article [10] is used which describes a modified of MAP algorithm. These are the most popular modifications of the BCJR algorithm, named log- MAP and Max log -MAP algorithms. The Log-MAP algorithm operates in logarithmic domain using addition and subtraction instead of multiplication and division [3]. The next equation represents the branch metric, forward and backward recursion metric of Log- MAP algorithm in equations 4, 5) and 6

$$\gamma(s', s) = \ln C_k + \frac{u_k L(u_k)}{2} + \frac{L_c}{2} \sum_{l=1}^n X_{kl} Y_{kl} \quad (4)$$

$$\alpha_{l+1}^*(s)_{s' \in s} = \max^* \{ \alpha_l^*(s') + \gamma_l^*(s', s) \} \quad (5)$$

with condition in the initial, $\alpha_0^*(0) = 0, \alpha_0^*(s) = -\infty$ for $s \neq 0$. Similarly,

$$\beta_l^*(s') = \max_{s' \in s}^* \{ \beta_{l+1}^*(s') + \gamma_l^*(s', s) \} \quad (6)$$

$\beta_{L+m}^*(0) = 0, \beta_{L+m}^*(s) = -\infty$ for $s \neq 0$.

Where the \max^* equal:

$$\max^*(z_1, z_2) = \log(e^{z_1} + e^{z_2}) = \max(z_1, z_2) + \log(1 + e^{-|z_1 - z_2|}) \quad (7)$$

And the APP in Equation 3 becomes:

$$L(u_k|y_{(s',s) \in S+}) = \max^* \{ \alpha_i^*(s') \gamma_i^*(s', s) + \beta_{i+1}^*(s) \} - \max_{(s',s) \in S-}^* \{ \alpha_i^*(s') \gamma_i^*(s', s) + \beta_{i+1}^*(s) \} \quad (8)$$

B. Modified of log MAP algorithm

Since Log- MAP algorithm is completed in logarithmic domain. Thus the operation is not easy to be implemented in hardware because the storing the data in look up table and reading which it takes much time. Therefore, a proposing an algorithm to meet the trade off between performance and complexity is needed which is based on article [11]. In article [11], a novel algorithm based on the exploitation of the polynomial regression function is proposed to modify the correction function in the Log- MAP algorithm in equation 7 and replacing it by another function. In terms of performance, the proposed algorithm has the closest performance to the Log- MAP algorithm [11]. Polynomial function used in correction function is as follow:

$$f(x) = \ln(1 + e^{-x}) \approx a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (9)$$

Details of derivation and explanation are shown in [3], and the last equation used to replace in correction function is shown in equation 10.

$$\ln(1 + e^{-x}) \approx 0.0012x^4 - 0.0216x^3 + 0.1539x^2 - 0.5148x + 0.6947 \quad (10)$$

IV. DESIGN AND IMPLEMENTAION

A. Turbo encoder implementaion

Turbo Encoder implementation by XSG in Fig. 3, using QPP interleaver implement by using Black Box component containing the VHDL code Linking with ISE 14.7.

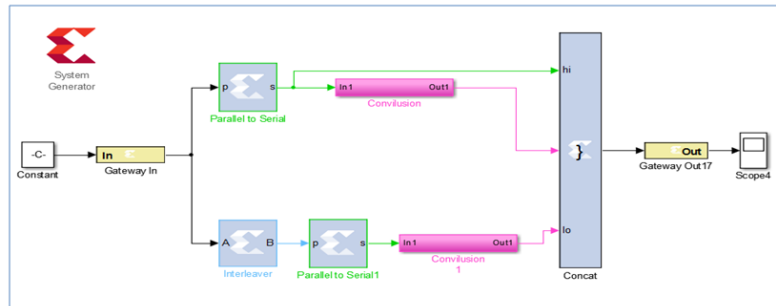


Figure 3: LTE turbo encoder in XSG

B. Turbo decoder implementaion

When receiving concatenated data channel, the time division demultiplexer (TDD) component using to separation concatenation data channel deals with in turbo decoder algorithm as in Fig. 4 same with the procedure in Fig. 2.

Spartan- 6. In addition, in Fig. 12 appear the turbo encoder with Artix- 7 board. The utilization, power and timing of the FPGA boards used in this paper in the above Fig. 11 and 12 in the Table I and adding to it the utilization of previous work in paper [12] to comparison between them.

TABLE I
 RESOURCES UTILIZATION OF IMPLEMENTATION ENCODER

Parameter	Artix- 7	Spartan- 6	Kintex- 7	[12]
Number of slice LUTs	1- 25%	1- 23%	1- 27%	0- 70%
Number of slice registers / flip flop	1- 52%	1- 47%	1- 53%	0- 42%
Number of bonded IOB	5- 12%	4- 12%	2- 12%	40- 89%
Number of BUFG	1- 3%	1- 6%	1- 3%	-
Total time of the turbo encoder	7.274 ns	2.317 ns	1.616 ns	6.30 ns
Total power of turbo encoder	0.102 W	0.044 W	0.159 W	-
Maximum frequency	0.102 W	431.593 MHz	618.812 MHz	-

All three devices implemented using the Xilinx System Generator (XSG) while the [12] implemented using the VHDL code with ISE program simulation. Turbo encoder compose from two RSC and one interleaver, that taken the simple utilization with high speed and low power from the devices 4% as in the Table I. In the article [1], implemented the RSC encoder in XSG which only utilize 4 LUT and 3 registers from the FPGA device. This proves that the most utilization and time delay in the Table I goes into implementation and processing of interleaver. In Table I, the three devices of this paper utilize approximately the same resources. In [12], observe that utilize approximately three times of slice LUTs that are used in this paper. Additionally, using 89 Bonded IOB = 40% from the device while the other devices work in this paper using only 12 Bonded IOB = (2- 5)% from its devices. In general, the implementation complexity of turbo code depend mainly on the number of the input (block size of data input), when increasing the number of input of the system, the RSC encoder is not affected but lead to increase block size of interleaver that lead to increase the implementation complexity of it, therefore; increase using the resources of FPGA device.

TABLE II
 RESOURCES UTILIZATION OF IMPLEMENTATION DECODER

Parameter	Kintex- 7	[3]- 4bit	[14]	[13]
Type of decoding algorithm	Improved log- MAP	Max- log- MAP	Max- log- MAP	MAP- BCJR
Number of slice LUTs	3- 6027%	6310	36- 16922%	26 465
Number of slice registers / flip flop	1- 141%	4108	1- 1148%	212 852
Number of bonded IOB	3- 17%	-	13- 33%	-
Number of LUTRAM	1- 3%	30	14- 23%	-
Total number of DSP of turbo code	23- 192%	-	-	-
Total power of turbo code	263.326 W	-	-	-
Maximum frequency	-	270.9 MHz	-	102 MHz

In the turbo decoder is the most impotent branch in the turbo code and have the most complexity of the system. Turbo decoder depend in the iterative processing and the turbo decoding algorithm, in this part real time implementation of all turbo code with improved log- MAP algorithm that are designed in this paper. Real time implementation of the system in the same way of HDL Netlist. with FPGA Kintex- 7 XC7K325t- 2ffg900C. The implementation with 8 bit input and controlled completely from the user by using switch. In Fig. 13 real time implementation of turbo decoder to prove the system is applicable for real time application. Explained that the turbo code depend on the number of bit at the input, and complexity of the Log- MAP algorithm also depending on it. From the Fig. 13 and Table II, showing the resources utilization of FPGA for 8 bit data input. comparison with [3] that are using 4 bit data input, showing although that in this

paper using number of bit twice the number of bit in the [3] and the algorithm using the robust algorithm from the [3], but the resources utilization in our work from the device is less than the [3]. Similarity in the [13] and [14]. The Log-MAP algorithm design of this paper not need to using RAM, all data is available during the processing, while in [3] and [14] using as shown 30 and 23 LUTRAM, respectively, from the devices. Turbo code with improved Log- MAP algorithm that is implemented hardware on FPGA Kintex- 7, that prove the applicable of it for real time application. Also, utilize of resources from the device is less than from previous works [3], [14],[13]. when implementation turbo decoder, there is having tradeoff between the complexity and the BER performance. when needing to minimize device utilization need to decrease number of size block of data input, and decrease the number of iterative in decoder, but these ways lead to degradation BER performance. Finally, can seen in Fig. 14, how the signal transmitted step by step in the system and reconstruct the original message in the receiver side.

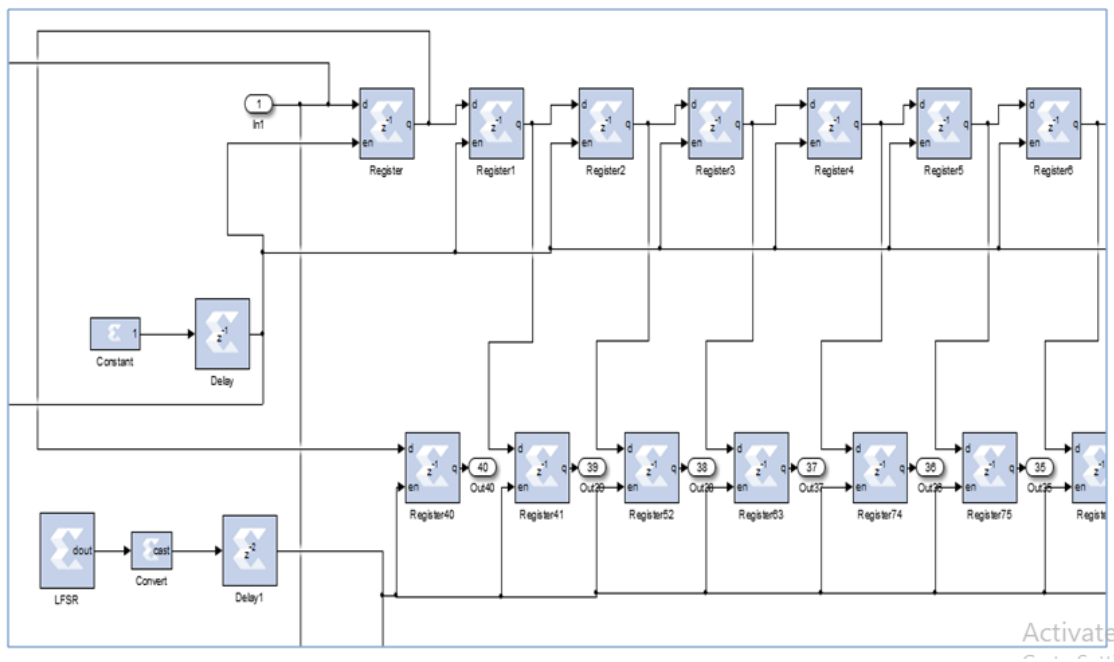


Figure 5: Serial to parallel converter in XSG for 40 bit

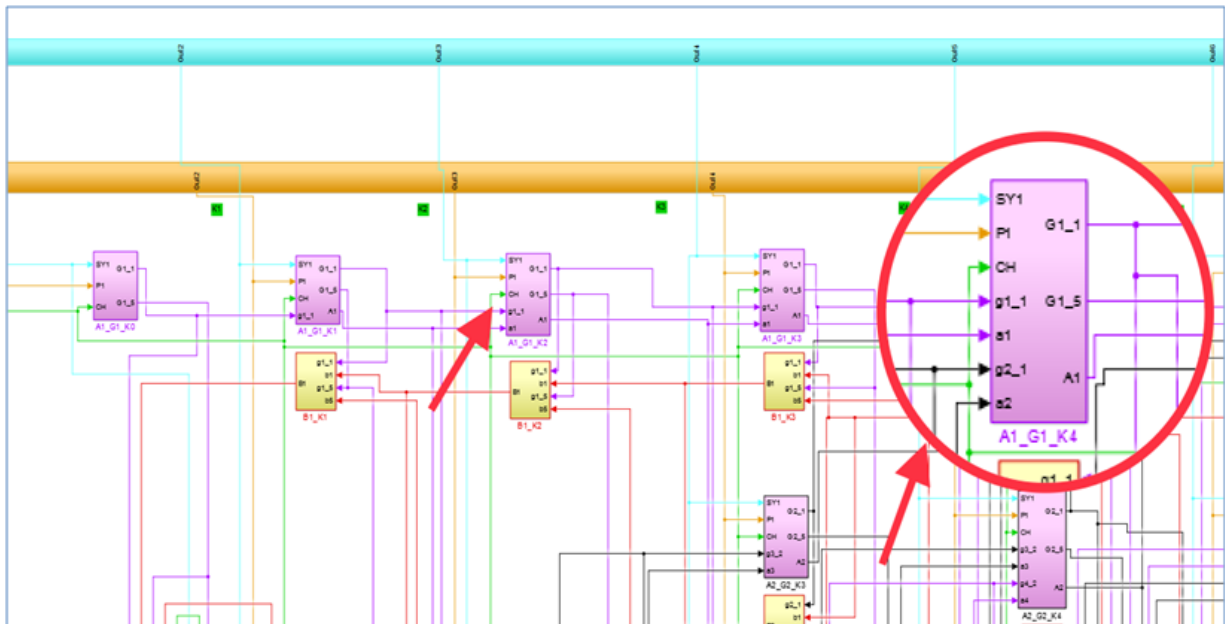


Figure 6: Serial to parallel converter, gamma γ and forward metric subsystems in XSG

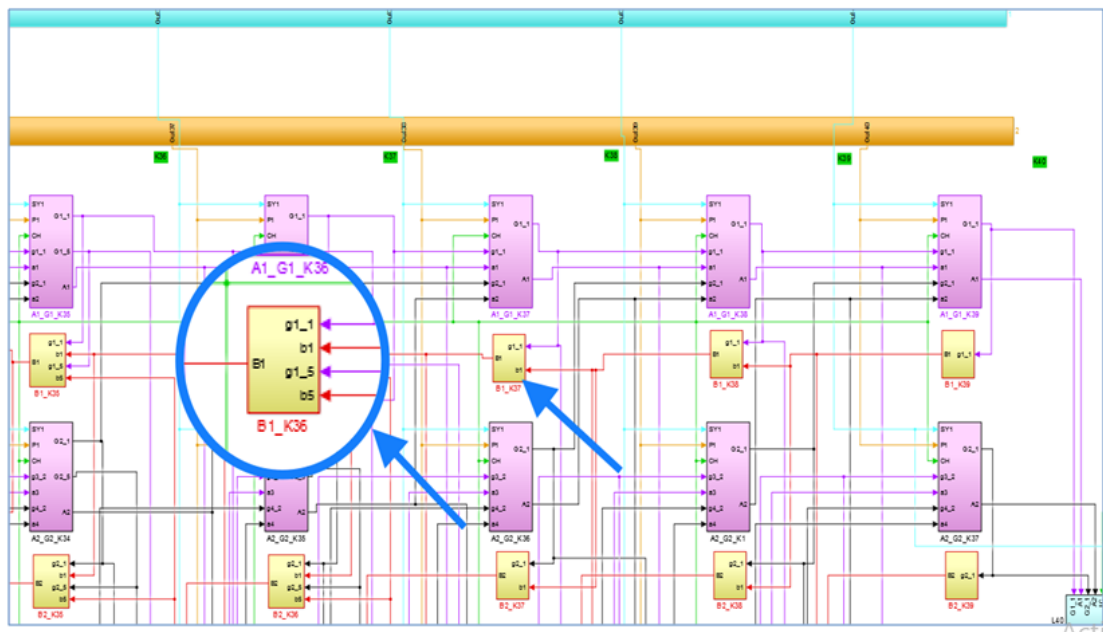


Figure 7: Last Stages of trills diagram, and backward metric subsystem in XSG

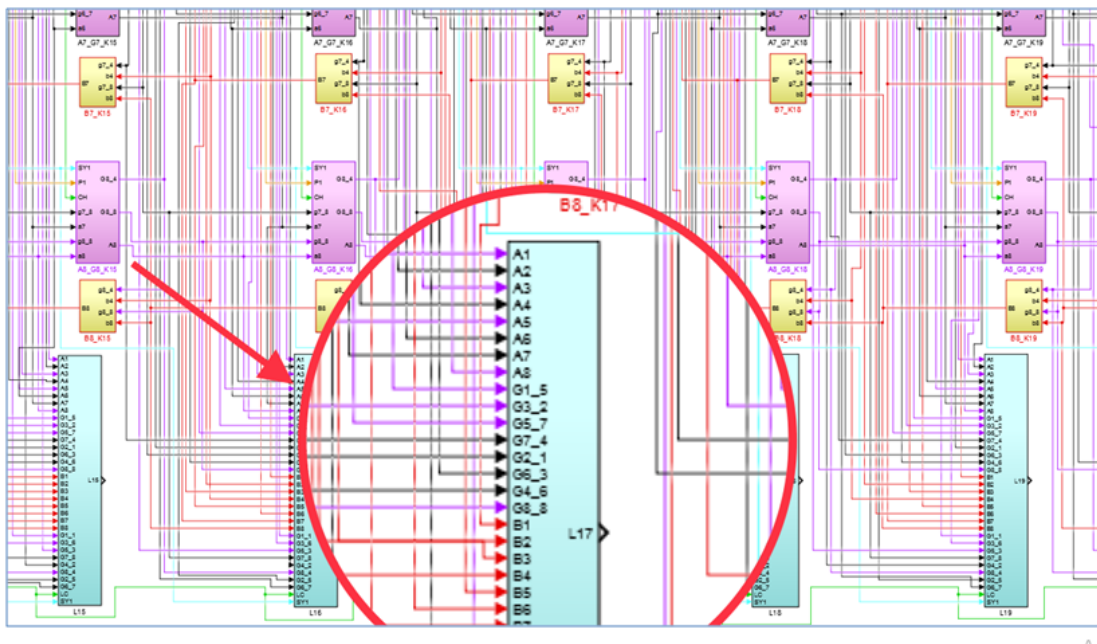


Figure 8: The log likelihood ratio (LLR) $L(u_k|y)$ subsystems in XSG

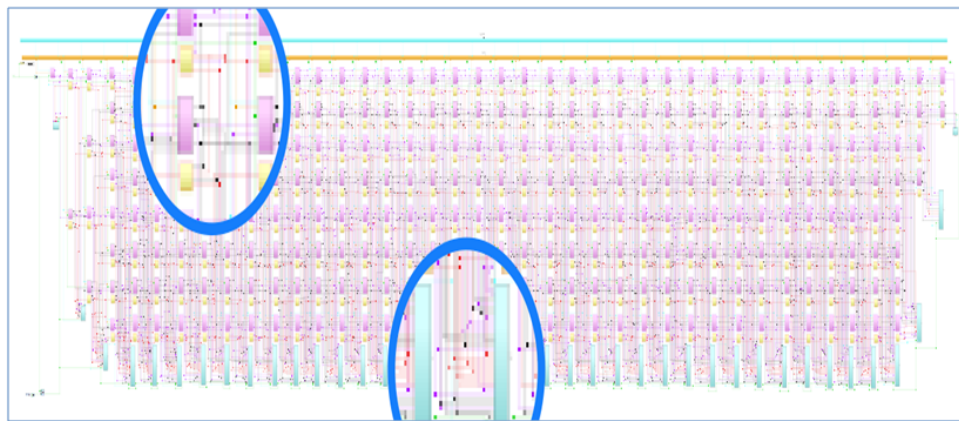


Figure 9: Trills diagram of log MAP algorithm in XSG

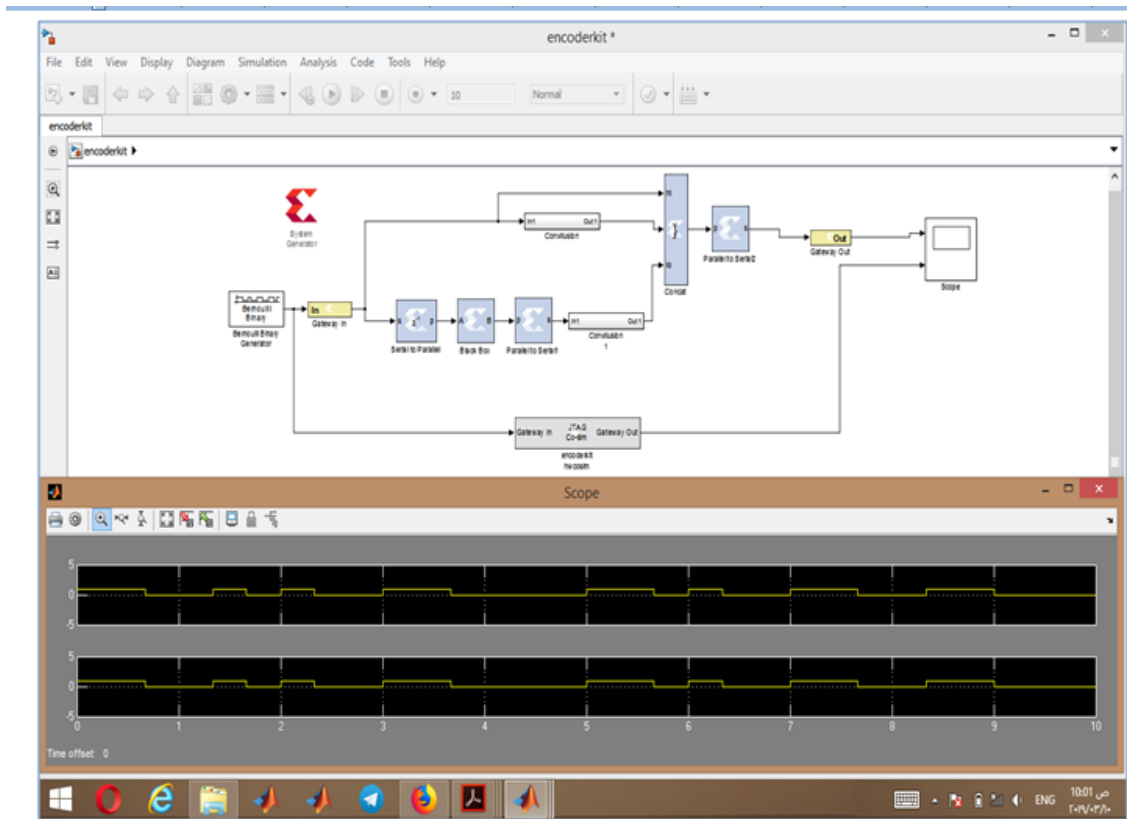


Figure 10: Turbo encoder with JTAG Component and their result in the real time



Figure 11: Real time implementation of turbo encoder using spartan- 6 device



Figure 12: Real time implementation of turbo encoder using artix- 7 device

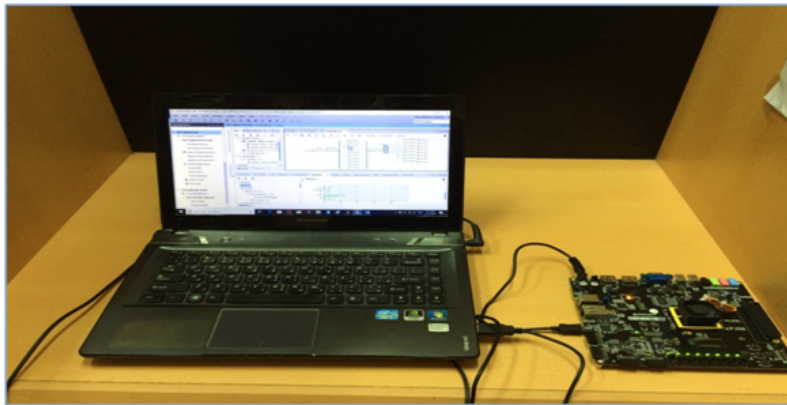


Figure 13: Real time implementation of turbo decoder with kintex- 7 device

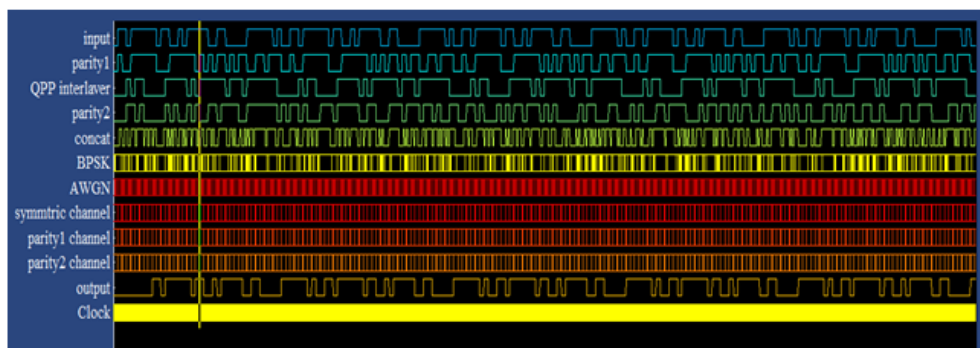


Figure 14: Outputs of all system using wavescope component

VI. CONCLUSIONS

The implementation of turbo code into Xilinx System Generator (XSG) appears that is simple as compared to the use of VHDL and using serial to parallel operation that make data channel available at the same time for all stage of trills diagram of Log- MAP algorithm and reducing the number of required cycles of algorithm by 75%, and make the calculations of its parameters not using any cycle only that are used in the converter. Turbo encoder hardware implementation and proves its applicability in the real time completely controlled by the user, using the switches on the board with Xilinx Kintex- 7, Spartan- 6 and Artix-7 devices by using two methods Hardware Co-Simulation and HDL Netlist. This implementation utilize resources from devices less than other works by 2.8 times in LUTs and 7.4 times in IOB. Turbo decoder with improved Log- MAP algorithm has been implemented in hardware with Xilinx Kintex- 7 device, and proves it completely real time implementation. Also, observe the resources utilize from the device less than other works approximately 40 times in Number of Slice Registers / Flip Flop.

REFERENCES

- [1] Jakob L. Buthler, Troels Jessen, Michael Buhl and Rune Simonsen, " Turbo Codes and OFDM Implementation For LTE mobile systems" , Group 11gr850, Aalborg University, May, 2011.
- [2] Dr Houman Zarrinkoub, " Understanding LTE with MATLAB from Mathematical Modeling to Simulation and Prototyping" , John Wiley & Sons, Ltd, 2014.
- [3] Vadim Belov, Sergey Mosin, " FPGA Implementation of LTE Turbo Decoder Using MAX- log MAP Algorithm" , in 2017 IEEE Embedded Computing (MECO), 2017 6th Mediterranean Conference on, 1- 4.
- [4] Book: Stefania Sesia, Issam Toufik and Matthew Baker, " LTE - The UMTS Long Term Evolution, From Theory to Practice Second Edition" , This edition first published 2011 John Wiley & Sons Ltd.
- [5] Book: Jorge Castineira Moreira, Patrick Guy Farrell, " Essentials of Error Control Coding" , Copyright C 2006 John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England.
- [6] Prabhavati D. Bahirgonde and Shantanu K. Dixit, " Low complexity modified constant Log- MAP algorithm for radix- 4 turbo decoder" , In Pervasive Computing (ICPC), 2015 International Conference on, pp. 1- 4. IEEE, Harvard , 2015.
- [7] A. Nimbalkar, T. K. Blankenship, B. Classon, T. E. Fuja, and D. J. Costello, " Contention- free interleavers for high- throughput turbo decoding" , IEEE Trans. Commun. , vol. 56, no. 8, pp. 1258- 1267, Aug. 2008.
- [8] LTE, Evolved Universal Terrestrial Radio Access (E- UTRA), " Multiplexing and channel coding" , 3GPP TS 36.212 version 10.0.0 Release 10.
- [9] Cristian Anghel, Cristian Stanciu and Constantin Paleologu, " LTE Turbo Decoding Parallel Architecture with Single Interleaver Implemented on FPGA" , Circuits, Systems, and Signal Processing, 1- 21, 2016.
- [10] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, " Optimal Decoding of Linear Codes for minimizing symbol error rate" , IEEE Transactions on Information Theory, vol. IT- 20(2), pp. 284- 287, March, 1974.
- [11] D.H. Nguyen and H. Nguyen, " An improved log- MAP algorithm based on polynomial regression function for LTE turbo decoding" , in 2015 IEEE International Conference on Communication Workshop (ICCW, IEEE, 2015, pp. 2163- 2167.
- [12] Palle Prasanth Kumar, K V Gowreesrinivas and P Samundiswary, " Design And Analysis Of Turbo Encoder Using Xilinx ISE" , in 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2016.
- [13] L. F. Gonzalez Perez, L. C. Yllescas Calderon, and R. Parra Michel, " Parallel and configurable turbo decoder implementation for 3GPP- LTE" , in Proc. Int. Conf. Reconfigurable Comput. FPGAs, Dec. , 2013, pp. 1- 6.
- [14] S. Mishra, H. Shukla, and S. Madhekar, " Implementation of Turbo Decoder Using MAX-LOG- MAP Algorithm in VHDL" , pp. 1- 6, 2015.