

# Design and Implementation Different Types of Turbo Decoder with Various Parameters

Samir Jasim Mohammed  
College of Engineering  
University of Babylon  
Babylon, Iraq

Ansam Abbas Obaid  
College of Engineering  
University of Babylon  
Babylon, Iraq

## ABSTRACT

This paper presents design and implementation of turbo code, after that many types of decoders are introduced with various many parameters such as (number of iteration, length of code, number of frame, type of decoding techniques, rate, generator polynomial and type of channel) get the Bit Error Rate (BER) for each case, and compare the results. This work in order to study the effect of each parameter on the performance of Turbo Code to specify the parameters that give the optimum performance of this codes. Finally turbo encoder implemented on FPGA device.

## General Terms

MATLAB, Xilinx 14.7, Field Program Gate Array (FPGA), SOVA, log\_MAP, and MAX\_log\_MAP Techniques.

## Keywords

Turbo code, Bit Error Rate (BER), Recursive Systematic Convolutional (RSC) and Log Likelihood Ratio (LLR).

## 1. INTRODUCTION

Turbo codes are presented in 1993. Turbo codes are designed usually from concatenation codes connected in serial concatenated convolutional code (SCCC) or in parallel (PCCC) separated by an interleaver and they have provided for transmission data of achieving close to Shannon limit [1]. Turbo codes chose for the telemetry coding standard for high data transmission in Universal Mobile Telecommunications System (UMTS) for third generation standards of mobile communication, where the states in the trellis is 8 and the constraint length is 4 [2].

## 2. TURBO CODES STRUCTURE

Turbo codes performance is excellent due to the combination of concatenation code, recursive convolutional encoders, pseudo-random interleaver and iterative process of decoders, where the decoders are worked in soft-input and soft-output. For these reasons, turbo codes are involved in several aspects such as: mobile radio, satellite, terrestrial telephony, image processing, etc.[3]. The structure of the encoder and decoder of turbo code introduces in the following parts:

### 2.1 Encoder

The nature of Turbo encoder is using two identical Recursive Systematic Convolutional (RSC) code in parallel and separating by an interleaver. Interleaver is used to minimize the correlation of the outputs of encoders that makes the best results, it's matrix form with rows and columns, relatively to the length of the code. Many types of the interleaver are used in turbo codes but pseudo-random interleaver is consider the best one[4].

Interleaver/deinterleaver are using and they are working a main role in the performance of turbo codes, where interleaver helps to increase the minimum distance of these codes and it breaks the low weight of input sequence by spread out the

burst errors by maps the sequence of bits into another sequence of bits by making the length of the interleaver is very large the superior performance is achieved [5].

The structure of turbo encoder is displayed in Figure (1). According to the block diagram of Turbo encoder the rate is  $R = \frac{1}{3}$  to increase this rate puncturing technique will be used to obtain high rate  $R = \frac{1}{2}$ . Puncturing is operating on the parity bits only but the systematic bits are not punctured [6].

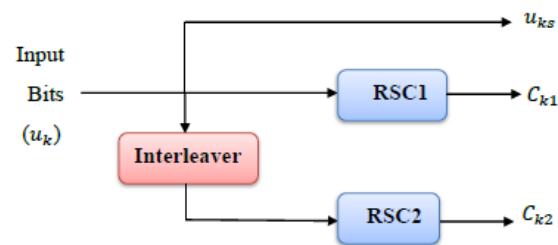


Fig 1: Turbo Encoder block diagram

### 2.2 Decoder

The block diagram of turbo decoder consists from a pair of decoders which cooperatively to improve the estimate the original bits, this diagram is shown in Figure(2).

In the decoding process two decoders connected in an iterative manner, which will allow each of the decoders to receive the sequence corresponding to the systematic, the transmitted parity bits from the encoder and a priori information from the second decoder. Each decoder produces a soft-decision for each bits of the message in logarithmic form called a log likelihood ratio (LLR) [7]. A hard decision is carried out at the second decoder to transfer the last signal into 1's and 0's and compare it with the original message at the end of decoding process [8].

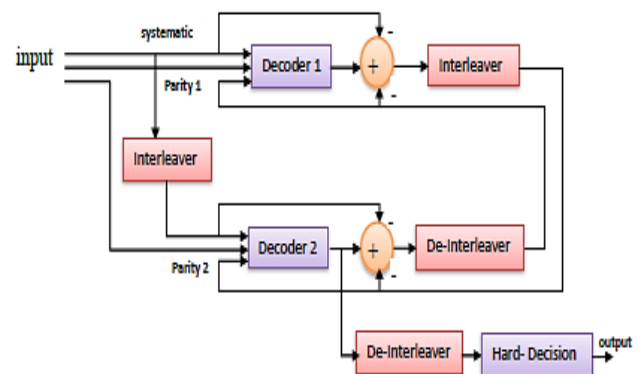


Fig.2: Turbo Decoder block diagram

### 2.3 Implementation of Turbo Decoding Techniques

Turbo decoding techniques are mainly divided into two types: The first one is the soft output Viterbi algorithm (SOVA) and the other is maximum a posteriori probability (MAP) then simplified to Log-MAP and MAX-Log-MAP[9]. In SOVA algorithm chooses a branch with the greater probability in the trellis and discards other branches, but in the MAP algorithm takes all paths in the trellis and it computes all points probabilities [2].

The implementation of the MAP algorithm in hardware is too complex due to numerical representation of probabilities, large numbers of multiplication and division operations and non-linear functions. Therefore, approximations of the MAP algorithms derived in log domain named Log-MAP and Max-Log-MAP [5].

#### 2.3.1 SOVA Algorithm

Soft output Viterbi Algorithm (SOVA) operates similarly to Viterbi Algorithm (VA) but with two essential modifications that allows it to be used as a component decoder for turbo codes. First modify, SOVA is used the path metric that uses the value of a prior probability of each input bits. Second modify, SOVA produces a soft output that indicates the decision accuracy [6].

The SOVA algorithm is summered as follows :

The  $L(u_k/y_k)$  in Equation (1) is represented the soft output of SOVA component decoder and it is decomposed in three terms:

$$L(u_k/y_k) = L(u_k) + L_c y_{ks} + L_e(u_k) \quad (1)$$

Where:  $L_c y_{ks}$  is the value of the weighted received systematic channel,  $L_c$  is denoted as the channel reliability and  $L_e(u_k)$  is the extrinsic value that produced by present SOVA component decoder.

The path metrics can be calculated in Equation(2) by the cross correlation to the received:

$$M(S_k^s) = M(S_{k-1}^s) + \frac{1}{2} u_k L(u_k) + \frac{L_c}{2} \sum_{l=1}^n y_{kl} x_{kl} \quad (2)$$

Where :  $M(S_{k-1}^s)$  the prevised path metric,  $n$  the length of message with parity,  $x_{kl}$  the codewords after the Binary Phase Shift Keying (BPSK) modulation.

The extrinsic information that produced by first SOVA component decoder in Equation (3):

$$L_{e1}(u_k) = L_1(u_k/y_k) - L_1(u_k) - L_c y_{ks} \quad (3)$$

The extrinsic information that produced by second SOVA component decoder as the same as for the first decoder but the  $L_c y_{ks}$  must be interleaved.

The  $L_{e2}(u_k)$  is deinterleaved, then it returns to feed back the first SOVA consider the priori information to the following iteration. This steps repeated for each iteration and finally stop after specified number of iteration, where by increasing the number of iteration better results can be achieved. At the last iteration the soft output LLR can be calculated from the output of the second decoder after de-interleaving and passing through threshold detector.

#### 2.3.2 Log-MAP Algorithm

Although the complexity of Log-MAP is less than of the MAP, but it yet requires a great number of accounts [9]. The expression of LLR for Log-MAP algorithm can be written in Equation(4) [10]:

$$L(\hat{u}_k) = \log \frac{\sum_{u_{k+1}}^{(s',s)} \exp(\alpha_{k-1}^{LM}(s')) \cdot \exp(\gamma_k^{LM}(s',s)) \cdot \exp(\beta_k^{LM}(s))}{\sum_{u_k=-1}^{(s',s)} \exp(\alpha_{k-1}^{LM}(s')) \cdot \exp(\gamma_k^{LM}(s',s)) \cdot \exp(\beta_k^{LM}(s))} \quad (4)$$

Where:  $\alpha_{k-1}^{LM}(s')$  forward recursion,  $\beta_k^{LM}(s)$  backward recursion and  $\gamma_k^{LM}(s',s)$  transitional probability in logarithmic domain.

#### 2.3.3 Max-Log-MAP Algorithm

The Max-Log-MAP is simplified to log-MAP algorithm in mathematic operation , it takes one for each forward and backward recursion out of the trellis and calculates the different metrics numbers [10]. The Max-Log-MAP uses to reduce the complication of the Log-MAP by performing expression for forward and backward recursion, LLR for Max-Log-MAP expressed in Equation(5) [11]:

$$L(\hat{u}_k) = \max_{s',s(u_k=+1)} ([\gamma_k^{LM}(s',s) + \alpha_{k-1}^{MLM}(s) + \beta_k^{MLM}(s')], [\gamma_k^{LM}(s',s) + \alpha_k^{MLM}(s) + \beta_k^{MLM}(s')]) - \max_{s',s(u_k=-1)} ([\gamma_k^{LM}(s',s) + \alpha_{k-1}^{MLM}(s) + \beta_k^{MLM}(s')], [\gamma_k^{LM}(s',s) + \alpha_k^{MLM}(s) + \beta_k^{MLM}(s')]) \quad (5)$$

By applying this approximation the degradation will occur in the BER of Max-Log-MAP compare with Log-MAP.

### 3. FPGA

Field Programmable Gate Array (FPGA)are consisted from an array of logic cells connected by a programmable metal interconnect and an array of I/O cells. The cells and programmable interconnect switches are built as a static RAM (SRAM), so that the current configuration stored in these RAM elements. In order to reconfigure, the memory elements must simply be loaded with new values, thus the hardware can be changed entirely of milliseconds[12].

Here, design and implementation of turbo encoder in FPGA device by using Xilinx program. Two identical RSCs are used and separated between them by an interleaver, all these component worked as a schematic.

### 4. RESULTS AND DISCUSSION

This part presents the results which take out from Simulation of turbo code in MATLAB program and implement turbo encoder by using FPGA device. BPSK modulation and the UMTS interleaver are used for the proposed system.

Many parameters are changing in this simulation to observe the effect each of them in the performance such as: number of iteration, length of code, number of frame, type of decoding technique, rate(puncturing technique), generator polynomial and type of channel. In Table (1), all parameters of turbo code Implementation, get the results for each case and then compared between them. Also the results of implementation the encoder are introduced below:

**Table (1): All Parameters of turbo code Implementation**

1	Number of iterations	1-8
2	Code Lengths	256, 512, 1024, 4096
3	Number of frames	512, 1024
4	Decoding Techniques	SOVA, Log-MAP, MAX-Log-MAP
5	Rate	$R_1 = 1/3, R_2 = 1/2$
6	Generator Polynomials	$g_1(13,15)_8, g_2(7,5)_8, g_3(37,21)_8$
7	Channels	AWGN, Rayleigh fading

In Figure (3), the effect of increasing the number of iteration is illustrated, for example Turbo code performance at log-MAP decoding technique for iteration from 1 to 8 iteration with frame size 256 bit/frame, 1024 frame and rate  $R = 1/2$ .

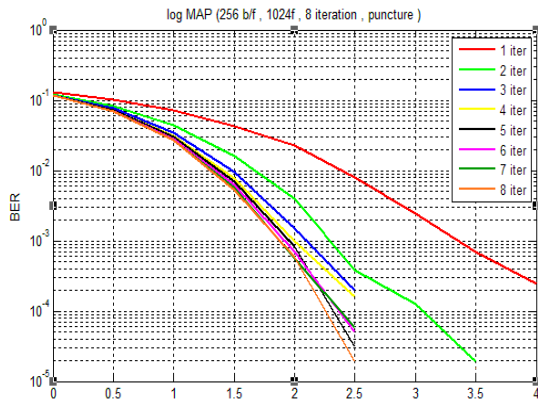


Fig 3: performance of Turbo code at 8 iteration

At any value of  $E_b/N_0$  can be taken and observed the value of BER curves and how it's improved until 6-8 iteration the performance is not much improvement, Table (2) displays the value of BER at  $E_b/N_0 = 2$  dB for each iteration.

Table (2): BER at different number of iteration

$E_b/N_0$ (dB)	iteration	BER
2	1	$2.5 \times 10^{-2}$
2	2	$4 \times 10^{-3}$
2	3	$1.75 \times 10^{-3}$
2	4	$1 \times 10^{-3}$
2	5	$8 \times 10^{-4}$
2	6	$6.5 \times 10^{-4}$
2	7	$5.75 \times 10^{-4}$
2	8	$5.75 \times 10^{-4}$

In Figure (4), The effect of increasing the code length introduced in SOVA technique with lengths 256, 512, 1024 and 4096 bit/frame all them with 1024 frame, 5 iteration and  $R = 1/3$ . Table (3) displays the BER at  $E_b/N_0 = 1.5$  dB for each length.

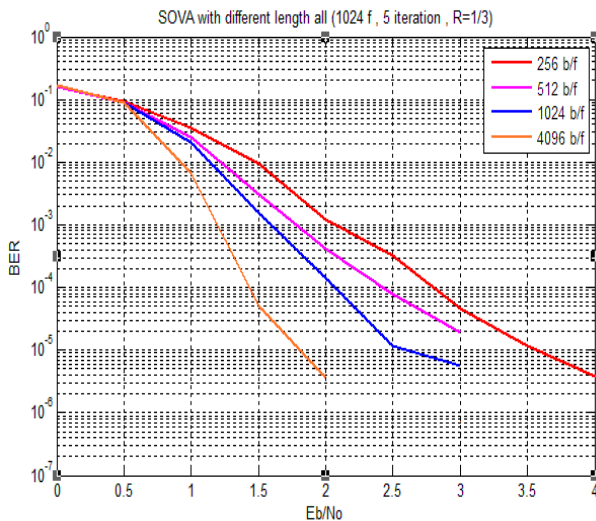


Fig 4 : Performance of Turbo code at different code lengths

Table (3): BER at different code length

$E_b/N_0$ (dB)	Code length	BER
1.5	256	$1 \times 10^{-2}$
1.5	512	$3.25 \times 10^{-3}$
1.5	1024	$1.75 \times 10^{-3}$
1.5	4096	$5.5 \times 10^{-5}$

The effect of increasing number of frame is illustrated in Figure (5), for MAX-log-MAP technique with number of frames 512, 1024 each of them with code length 256 bit/frame, 4 iteration and  $R=1/3$  by increasing the number of frame good BER obtained.

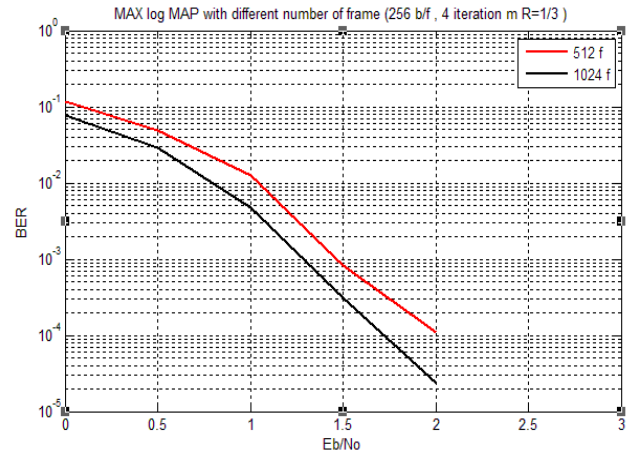


Fig.5: Performance of Turbo code at different number of frame

Figure (6) illustrates the BER of turbo code with various decoding techniques (SOVA, Log-MAP and MAX-Log-MAP) all them at 512 b/f, 1024 f, 4 iteration and Rate  $R = 1/3$ , the performance is improved with increasing  $E_b/N_0$ , as shown log-MAP is better than the others where the gain  $E_b/N_0$  is obtained about 0.5 – 0.7 dB in log-MAP compare with SOVA and 0.1 – 0.2 dB in log-MAP compare with MAX-Log-MAP.

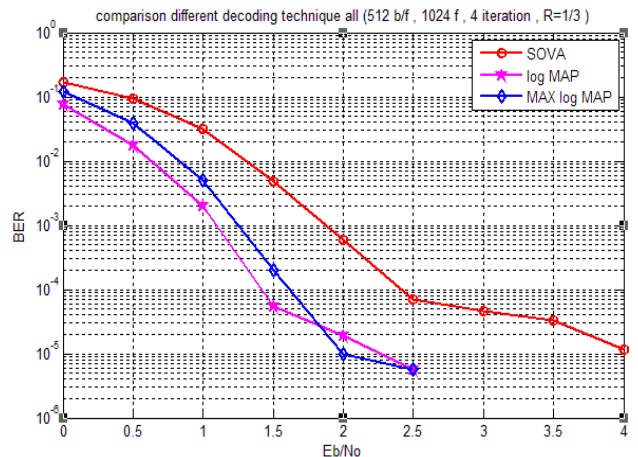


Fig.6 : Performance of Turbo code at different decoding techniques

The Rate of the system can be incersed by using puncturing technique as shown in Figure (7), this introduced for SOVA technique with frame size 1024 b/f, 1024 frame and 5 iteration, degradation will occur about 0.5 dB compare with  $R = 1/2$ .

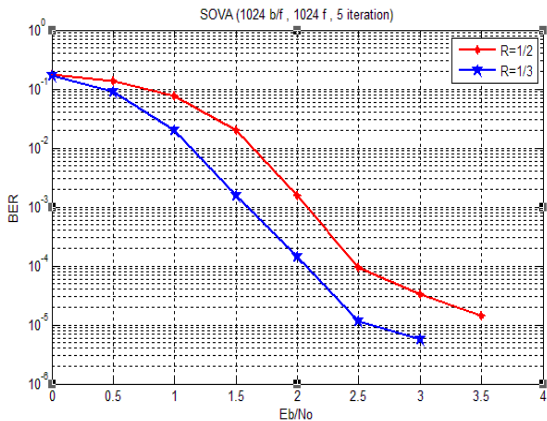


Fig. 7 : Performance of Turbo code at different Rate

The standard generator polynomial of turbo code in UMTS is  $g = (13,15)_8$ , in Figure (8) another generators polynomials applied at the system  $g_1 = (7,5)_8$  and  $g_2 = (37,21)_8$ , in SOVA technique with frame size 256 b/f, 1024f, 5 iteration and  $R=1/3$ .

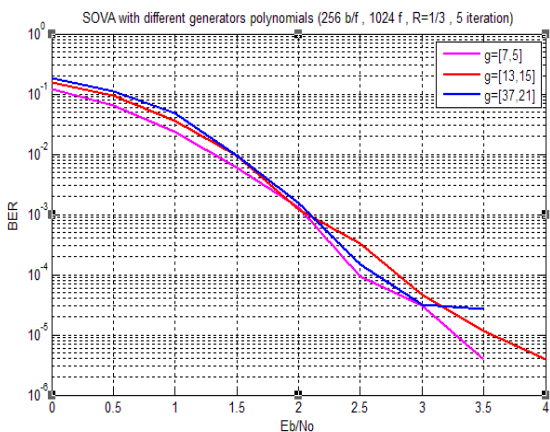


Fig. 8: Performance of Turbo code at Three Generators Polynomials

All Figures introduced above gave the performance of turbo code over AWGN channel. Now, in Figure (9) present the BER of turbo code over Rayleigh fading channel and compare it with AWGN channel. For an AWGN channel the fading amplitude  $a=1$  whereas for a Rayleigh fading channel  $a$  is a random variable. This is introduced for log-MAP with the code length 512b/f, 1024f, 3 iteration and  $R = 1/3$ .

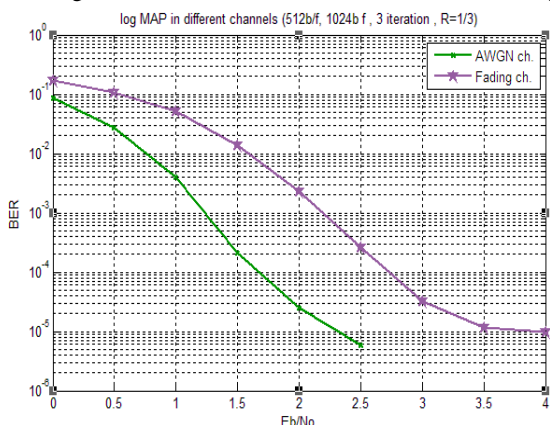


Fig.9 : Performance of Turbo code at two channels (Rayleigh fading channel and AWGN)

In Figures (10,11,12) show the results of implementation

Turbo encoder on FPGA.

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	19	11,776	1%	
Number of 4 input LUTs	12	11,776	1%	
Number of occupied Slices	14	5,888	1%	
Number of Slices containing only related logic	14	14	100%	
Number of Slices containing unrelated logic	0	14	0%	
Total Number of 4 input LUTs	12	11,776	1%	
Number of bonded IOBs	5	372	1%	
Number of BUFGMUXs	1	24	4%	
Average Fanout of Non-Clock Nets	2.56			

Fig.10: Device Utilization Summary of Encoder

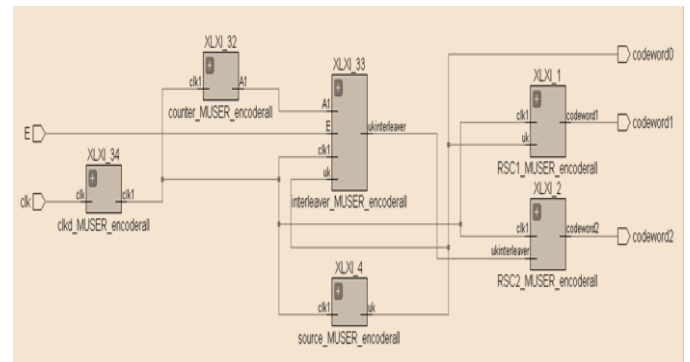


Fig.11: RTL Schematic of encoder

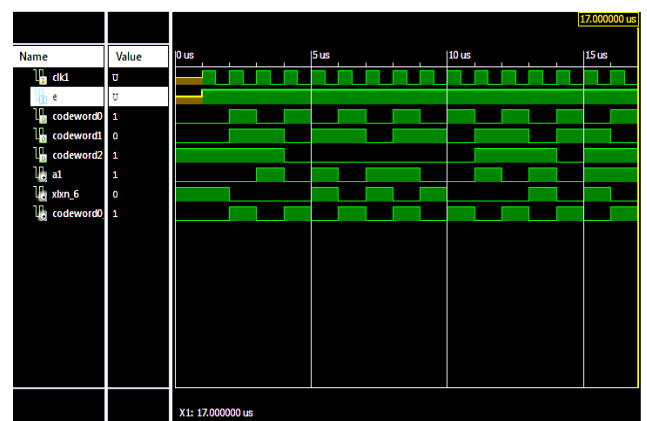


Fig.12: Turbo encoder Simulation in FPGA

## 5. CONCLUSION

Here, the design and implementation of turbo codes are presented as a simulation in MATLAB R2016b program and the encoder is presented on FPGA device type spartan3 and spartan3A by using Xilinx program 14.7. The BPSK modulation and the UMTS interleaver are assumed for the proposed system. After that many types of decoders were done with change many parameters, from this work and the results can conclude that when the number of iteration increased the BER decrease for the same  $E_b/N_0$  and the performance is improving. The increasing of code length and number of frame will improve the performance, different code length applied (256,512,1024 and 4096) and number of frame (512 and 1024) obtained the results for each case and compared between them. Many types of decoding techniques applied (SOVA, Log-MAP and MAX-Log-MAP) by using log-MAP technique the results of BER is best than the SOVA and MAX-Log-MAP for the same conditions of the system(512 b/f, 1024 f, 4 iteration and  $R = 1/3$ ) also the gain  $E_b/N_0$  is obtained about 0.5 – 0.7 dB in log-MAP compare

with SOVA and 0.1 – 0.2 dB in log-MAP compare with MAX-Log-MAP. Puncturing technique is used to increase the rate of the system but the BER is degrade about 0.5 dB by increasing the rate. Also changing the generator polynomial of the system and compare with the standard generator polynomial  $g = [13,15]$ , where two addition generators polynomials are used  $g_1(7,5)_8$ ,  $g_2(37,21)_8$ . Finally two types of channel used in this simulation (AWGN and Rayleigh fading channel) obtained the results for each one and compared between them when using AWGN channel the performance is best than the fading channel.

## 6. REFERENCES

- [1] Anum I., 2012 "Software implementation and performance of UMTS Turbo decoder", M.Sc. Thesis, Tampere University.
- [2] Jagadeesh K. and Chaitali C., 2004, "Design and Implementation of Low-Energy Turbo Decoders", IEEE Transactions On Very Large Scale Integration (VLSI) Systems, VOL. 12. NO. 9.
- [3] Rupinder K. and Sarpreet S.,2013, "Techniques for Turbo Decoding Using Parallel Processing, Comparative Analysis", Volume 3, Issue 4.
- [4] Christian B., Andreas B., Teo C. and Qiuting H., 2009, "Design and Optimization of an HSDPA Turbo Decoder ASIC", Journal Of Solid-State Circuits, VOL.X,NO. X.
- [5] Hamid R. S., Neil J. A. S., Masoud S. and Gabriele N.,2001,"Interleaver Design for Turbo Codes," IEEE Journal on Selected Area In Communications, vol.19, No.5.
- [6] Todd K. Moon, Wiley,2005, "Error Correction Coding: Mathematical Methods and Algorithms".
- [7] Jorge. C. Moreira and Patrich G. Farrell Wiley, 2006, "Essential of Error Control Coding".
- [8] Ibrahim S. R. and Mehmet Y., 2005,"Implementation of a turbo codes test bed in the Simulink environment", International Symposium on Signal Processing and Its Applications (pp. 847-850), Piscataway.
- [9] Yi Bo-nian, 2013, "Turbo Code Design and Implementation of High-Speed Parallel Decoder", Telkommika, Vol. 11, No. 4, April 2013,pp. 2116-2123.
- [10] ANG, LAY H. and LIM, WEE G.,2009, "SOVA Based LTE Turbo Decoders",Master Thesis, Lund University.
- [11] Eid Mohamed A. A., 2013,"Design and Implementation for multi-standard Turbo Decoder using a reconfigurable ASIP", Master's Thesis, The Cairo University.
- [12] JEAN-PIERRE D., GE'RY JEAN A. B. and GUSTAVO D. S., A JOHN WILEY & SONS, INC., PUBLICATION 2006," SYNTHESIS OF ARITHMETIC CIRCUITS, FPGA, ASIC, and Embedded Systems".