Design of Optimized 8-Level Turbo Encoder Using VLSI Architecture for LTE System

AMUTHA PRABHA.N, SYED FARHATHULLAH.S.F, VENKATESH.D

Abstract: Turbo Code is able to closely reach the channel capacity of Shannon limit. It increases the performance in the latest standard in the mobile network technology tree, LTE. A new architecture of Turbo code encoder is based on 3GPP standard. It is developed by implementing optimized 8-level parallel architecture, dual RAM in turbo code internal inter leaver, recursive pair wise matching, and efficient 8-level index generator in turbo code internal inter leaver. The proposed architecture successfully increases the speed of encoder 16 times faster compared to conventional architecture.

Keywords: *Channel Coding, Recursive Pair Wise Matching, Parallel Architecture, 3GPP, Interleaver .*

1 INTRODUCTION

Communication system requiring transmission channels, the transmitted signal which is sent by the transmitter is vulnerable to noises appears on the channel. The performance of the communication system is measured by using data rate, bit error ratio (BER), and packet error rate. Several methodologies have been developed to improve the performance of system near the channel capacity of Shannon limit. Turbo Code is one of the channels coding which is used to reduce received error signals in LTE technology [1] [3].Parallel architecture is adopted to reduce the clock latency of system. This improvement could provide a new perspective in designing channel coding and increase the bit rate of LTE system.

2 TURBO CODE

Turbo codes are finding use in 3G mobile communications and satellite communications as well as other applications where designers seek to achieve reliable information transfer over bandwidth- or latency-constrained communication links in the presence of data-corrupting noise. Turbo codes are nowadays competing with Low Density Parity Check (LDPC) codes, which provide similar performance. The first class of turbo code was the parallel concatenated convolutional code (PCCC). Since the introduction of the original parallel turbo codes, many other classes of turbo code have been discovered, including serial versions and repeat-accumulate codes. Iterative turbo decoding methods have also been applied to more conventional FEC systems, including Reed-Solomon corrected convolutional codes.

2.1 Turbo Encoder

The Turbo Encoder encodes a binary input signal using a parallel concatenated coding scheme. The coding scheme employs two identical convolutional encoders and one internal interleaver [1] [2]. Each constituent encoder is independently terminated by tail bits.

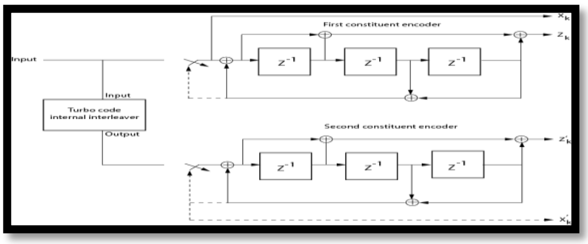

Figure 1.Turbo Encoder

2.2.1 Convolutional Code

Convolutional codes are used extensively in numerous applications in order to achieve reliable data transfer, including digital video, radio, mobile communication, and satellite communication. These codes are often implemented in concatenation with a harddecision code, particularly Reed Solomon. Prior to turbo codes, such constructions were the most efficient, coming closest to the Shannon limit.

2.2.2 Convolutional Encoder

The Convolutional Encoder encodes a sequence of binary input vectors to produce a sequence of binary output vectors. It can process multiple symbols at a time and accept inputs that vary in length during simulation.

2.3 Interleaver

Interleaver rearranges the elements of its input vector without repeating or omitting any elements. The input contains *N* elements, and then the Elements parameter is a column vector of length *N*. The column vector indicates the indices, in order, of the input elements that form the length-*N* output vector; that is, Output (*k*) = Input (Elements (k)) for each integer *k* between 1 and *N*.

3 UNOPTIMIZED 8 LEVEL TURBO CODE ENCODER

The parallelization gives a solution to high speed turbo encoder by reducing clock latency.8 level parallelization is used in 3GPP standard, all possible K value could be divided by 8.Encoder can be separated into three parts

1. Non-systematicfeedback convolutional encoder,
2. Systematic outputs sequence, and
3. Tail bits.

3.1 Non-systematic feedback convolutional encoder

Non-systematic feedback convolutional encoder structure is a system model that is represented to simplify the equation derivation[1]. The purpose of algebra manipulation, we use state space representation as mathematical model.

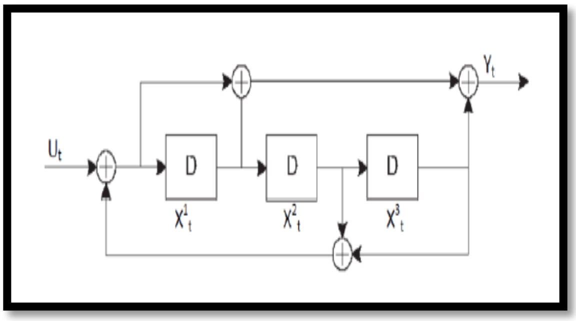

Figure.2 Non-systematic feedback convolutional encoder

The state space representation of Fig. 2 shall than be:

$$M_{t+1} = AM_t + BP_t \qquad (1)$$
$$N_t = CM_t + DP_t$$

$$M_t = \begin{matrix} M_t^1 \\ M_t^2 \\ M_t^3 \end{matrix}$$

$$A = \begin{matrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{matrix} \quad ,$$

$$B = \begin{matrix} 1 \\ 0 \\ 0 \end{matrix}$$

$$C = \begin{matrix} 1 & 1 & 0 \end{matrix} \quad ,$$

$$D = 1$$

3.2 Systematic Output Sequence

As described before in Eq. 1, we denote the more general equation for the relationship between next state and previous state as

$$M_{t+n} = AM_{t+(n-1)} + BP_{t+(n-1),}\{n\epsilon Z^{+}|n > 0\} \qquad (2)$$

To get further relationship between M_{t+n} *and* M_t, then $M_{t+(n-1)}$ can be divided, Eqn 2, can be denoted as

$$M_{t+n} = A^n M_t + A^{n-1}BP_t + A^{n-2}BP_{t+1} + \cdots + BP_{t+(n-1)} \qquad (3)$$

While, for the output variable,Y_t, is described generally as

$$N_{t+n} = CM_{t+(n-1)} + DP_{t+(n-1)} \qquad (4)$$

Hence, using Eq. 3 and 4 and also by substituting the multiplied results between matrices A, B, C, and D, for i=0,1…,$\frac{K}{8}$–1, where t=8i, then we can get 8 level convolutional encoder equation as follows.

$$M^{1}_{(8i)+8} = M^{2}_{(8i)} + M^{3}_{(8i)} + P_{(8i)} + P_{(8i)+3} + P_{(8i)+4} + P_{(8i)+5} + P_{(8i)+7}$$
$$M^{2}_{(8i)+8} = M^{1}_{(8i)} + P_{(8i+2)} + P_{(8i)+3} + P_{(8i)+4} + P_{(8i)+6}$$
$$M^{3}_{(8i)+8} = M^{2}_{(8i)} + P_{(8i+1)} + P_{(8i)+2} + P_{(8i)+3} + P_{(8i)+6}$$

$$N_{(8i)} = M^{1}_{(8i)} + M^{2}_{(8i)} + P_{(8i)}$$
$$N_{(8i)+1} = M^{1}_{(8i)} + M^{2}_{(8i)} + M^{3}_{(8i)} + P_{(8i)} + P_{(8i)+1}$$
$$N_{(8i)+2} = M^{1}_{(8i)} + M^{3}_{(8i)} + P_{(8i)} + P_{(8i)+1} + P_{(8i)+2}$$
$$N_{(8i)+3} = M^{3}_{(8i)} + P_{(8i)} + P_{(8i)+1} + P_{(8i)+2} + P_{(8i)+3}$$
$$N_{(8i)+4} = M^{2}_{(8i)} + P_{(8i)+1} + P_{(8i)+2} + P_{(8i)+3} + P_{(8i)+4}$$
$$Y_{(8i)+5} = M^{1}_{(8i)} + P_{(8i)+2} + P_{(8i)+3} + P_{(8i)+4} + P_{(8i)+5}$$
$$N_{(8i)+6} = M^{2}_{(8i)} + M^{3}_{(8i)} + P_{(8i)} + P_{(8i)+3} + P_{(8i)+4} + P_{(8i)+5} + P_{(8i)+6}$$
$$N_{(8i)+7} = M^{1}_{(8i)} + M^{2}_{(8i)} + P_{(8i)+1} + P_{(8i)+4} + P_{(8i)+5} + P_{(8i)+6} + P_{(8i)+7}$$
$$\text{i} = 0,1\ldots,\tfrac{K}{8}\text{-1} \qquad (5)$$

3.3 Turbo Code Tail Bits

Its support 8 level parallelization of 8 level convolutional encoder, tail bits are implemented using ROM whose address is next eight state.

TABLE 1
Tail Bits Convolutional Encoder

M1 (t+8)	M2 (t+8)	M3 (t+8)	Output Tail Bit Systematic	Output Tail Bit Convolutional Encoder
0	0	0	[0 0 0]	[0 0 0]
0	0	1	[1 0 0]	[1 0 0]
0	1	0	[1 1 0]	[0 1 0]
0	1	1	[0 1 0]	[1 1 0]
1	0	0	[0 1 1]	[1 0 1]
1	0	1	[1 1 1]	[0 0 1]
1	1	0	[1 0 1]	[1 1 1]
1	1	1	[0 0 1]	[0 1 1]

3.4 Turbo Code Internal Interleaver

The main problem of turbo code internal interleaver is index generator denoted as

$$\pi(i) = (f_1 i + f_2 i^2) mod(K) \qquad (6)$$
$$i = 0,1, \ldots . K - 1$$

The simplest approach to implement an interleaver is to store all the interleaving patterns in ROMs. However, this approach becomes impractical for a turbo decoder supporting multiple block sizes and multiple standards [1] [5].Once again to reduce clock latency, by direct manipulating, Eq.6 can be derived into 8 level parallelization into.

$$\pi(8j) = (f_1(8j) + f_2(8j)^2) mod(K)$$
$$\pi(8j+1) = (f_1(8j+1) + f_2(8j+1)^2) mod(K)$$
$$\pi(8j+2) = (f_1(8j+2) + f_2(8j+2)^2) mod(K)$$
$$\pi(8j+3) = (f_1(8j+3) + f_2(8j+3)^2) mod(K)$$
$$\pi(8j+4) = (f_1(8j+4) + f_2(8j+4)^2) mod(K)$$
$$\pi(8j+5) = (f_1(8j+5) + f_2(8j+5)^2) mod(K)$$
$$\pi(8j+6) = (f_1(8j+6) + f_2(8j+6)^2) mod(K)$$
$$\pi(8j+7) = (f_1(8j+7) + f_2(8j+7)^2) mod(K) \qquad (7)$$

4 OPTIMIZED 8-LEVEL TURBO ENCODER

The parallelization presents a solution to design high speed turbo encoder by reducing clock latency. 8 level parallelization is used in 3GPP standard, all possible K value could be divided by 8.Encoder can be separated into two parts :

1. Non-systematic feedback convolutional encoder
2. Internal Inter leaver

4.1Non-systematic Feedback Convolutional Encoder

Number of additions can be minimized by technique called recursive pairwise matching [1] [4]. Basically, this technique finds the most matches operands and stores it in intermediate results. Therefore, we propose algorithm 1 to optimize Eq. 6 by substituting adder with XOR gate because of bit operation as follows.

Algorithm 1: Non Systematic Feedback Convolutional Encoder

```
**Require:** $M_0^1 = 0\ M_0^2 = 0\ M_0^3 = 0$
**Require:** $\mathrm{i} = 0,1,\ldots.,\mathrm{K}/8 - 1$
**for** $i = 0\ to\ \frac{K}{8} - 1\ \boldsymbol{do}$
    $T_1 \leftarrow M_{(8i)}^1\ xor\ M_{(8i)}^2$
    $T_2 \leftarrow M_{(8i)}^3\ xor\ P_{(8i)}$
    $T_3 \leftarrow\ P_{(8i)+1}\ xor\ P_{(8i)+1}$
    $T_4 \leftarrow M_{(8i)}^1\ xor\ M_{(8i)}^2$
    $T_5 \leftarrow\ P_{(8i)+3}\ xor\ P_{(8i)+4}$
    $V_1 \leftarrow M_{(8i)}^2\ xor\ T_2$
    $V_2 \leftarrow M_{(8i)}^2\ xor\ T_3$
    $V_3 \leftarrow P_{(8i)+1} xor\ T_1$
    $V_4 \leftarrow P_{(8i)+5} xor\ T_5$
    $W_1 \leftarrow\ T_2 xor\ T_3$
    $W_2 \leftarrow\ V_1 xor\ V_4$
    $M_{(8i)+8}^1 \leftarrow W_2\ xor\ P_{(8i)+7}$
    $M_{(8i)+8}^2 \leftarrow T_4\ xor\ T_5 xor\ P_{(8i)+6}$
    $M_{(8i)+8}^3 \leftarrow V_2\ xor\ P_{(8i)+3} xor\ P_{(8i)+5}$
    $N_{(8i)} \leftarrow\ T_1\ xor\ P_{(8i)}$
    $N_{(8i)+1} \leftarrow\ V_3\ xor\ T_2$
    $N_{(8i)+2} \leftarrow\ M_{(8i)}^1\ xor\ W_1$
    $N_{(8i)+3} \leftarrow\ W_1\ xor\ P_{(8i)+3}$
    $N_{(8i)+4} \leftarrow\ V_2\ xor\ T_5$
    $N_{(8i)+5} \leftarrow\ T_4\ xor\ V_4$
    $N_{(8i)+6} \leftarrow\ W_2\ xor\ P_{(8i)+6}$
    $N_{(8i)+7} \leftarrow$
    $V_3\ xor\ P_{(8i)+4} xor\ P_{(8i)+5} xor\ P_{(8i)+6} xor P_{(8i)+7}$
**end for**
**Ensure:** $M_{t+1} \leftarrow A\ M_t + B\ P_t, N_t \leftarrow C\ M_t + D\ P_{t,} t = 0,1,\ldots.,K - 1$
```

4.2 Internal Interleaver

In Eq. 7 shows that the relationship between each level is independent. The idea of proposed algorithm is to make dependencies of next level with the aim of reducing operator arithmetic.

Definition 1: $r = a\ mod\ d$, $\{a \in \mathbb{Z},\ d \in \mathbb{Z}+\}$, then there are unique integers q and r, with $0 \leq r < d$ such that $a = dg + r$.

Definition 2: $r = a\ mod\ d$, then $r = a,\ a,\ d \in \mathbb{Z}+\ a < d\}$

By using two definitions above, we can get the relationship as follows

$$\pi(\mathrm{i} + 1) = (\mathrm{f}_1 + \mathrm{f}_2 + \pi(\mathrm{i}) + 2\mathrm{f}_2\mathrm{i})\mathrm{mod}\ (\mathrm{k}) \quad (8)$$
$$\{f_1, f_2, K \in z^+ | f_1 + f_2 < K\}$$

This equation (Eq. 8) is valid because based on Table I, condition $f1,\ f2,\ K \in \mathbb{Z}+\ f1 + f2 < K\}$ is fulfilled. After that, by further algebra manipulation using 8 level parallelization, we propose algorithm 2 as index generator of turbo code internal inter leaver.

Algorithm 2: Turbo Code Internal Interleaver

```
**Require**: $\gamma \leftarrow f_1 + f_2\ \ \gamma_2 \leftarrow 2f_2\ \pi(0)$
**Require**: $j = 0,1,\ldots.,\frac{k}{8} - 1$
**for** $j = 0\ to\ \frac{k}{8} - 1$ **do**
    $M_1 \leftarrow \gamma_1\ +\ \gamma_2.8j$
    $N_1 \leftarrow\ M_1\ + \gamma_2$
    $N_2 \leftarrow N_1\ + \gamma_2$
    $N_3 \leftarrow N_2\ + \gamma_2$
    $N_4 \leftarrow N_3\ + \gamma_2$
    $N_5 \leftarrow N_4 + \gamma_2$
    $N_6 \leftarrow N_5\ + \gamma_2$
    $N_7 \leftarrow N_6\ + \gamma_2$
    $F_1 \leftarrow\ \pi(8j) + M_1$
    $F_2 \leftarrow\ F_1 + N_1$
    $F_3 \leftarrow\ F_2 + N_2$
    $F_4 \leftarrow\ F_3 + N_3$
    $F_5 \leftarrow\ F_4 + N_4$
    $F_6 \leftarrow\ F_5 + N_5$
    $F_7 \leftarrow\ F_6 + N_6$
    $F_8 \leftarrow\ F_7 + N_7$
```

```
$\pi(8j) \leftarrow F_8\ mod\ (K)$
$\pi(8j+1) \leftarrow F_1\ mod\ (K)$
$\pi(8j+2) \leftarrow F_2\ mod\ (K)$
$\pi(8j+3) \leftarrow F_3\ mod\ (K)$
$\pi(8j+4) \leftarrow F_4\ mod\ (K)$
$\pi(8j+5) \leftarrow F_5\ mod\ (K)$
$\pi(8j+6) \leftarrow F_6\ mod\ (K)$
$\pi(8j+7) \leftarrow F_7\ mod\ (K)$
end for
Ensure: $\pi(i) = (f_1.i + f_1.i^2)mod(K)\ i = 0,1,\dots,K-1$
```

5 SIMULATION AND RESULT

In Proposed Optimized 8-Level Interleaver Turbo Encoder, we have used convolutional encoder and Inter leaver. In Convolutional encoder ,Gateway In and Gateway Out is used for system generator.XOR Gate is used as D-flip flop. Dual RAM is used to encode the input.8 Input is given as an binary format and 8 Output values are taken as the clock signal from the given input .In Inter leaver also Gateway In and Gateway Out is used for system generator. Counters is used to count the values and Black Box is used to store the values.ROM and MUX is used in the inter leaver. In the 8 level Interleaver turbo encoder 8 inputs is given to the convoluional encoder and the encoded output is seen through the scope and then interleaver is given to the convolutional encoder and again encoded output is seen through the scope.

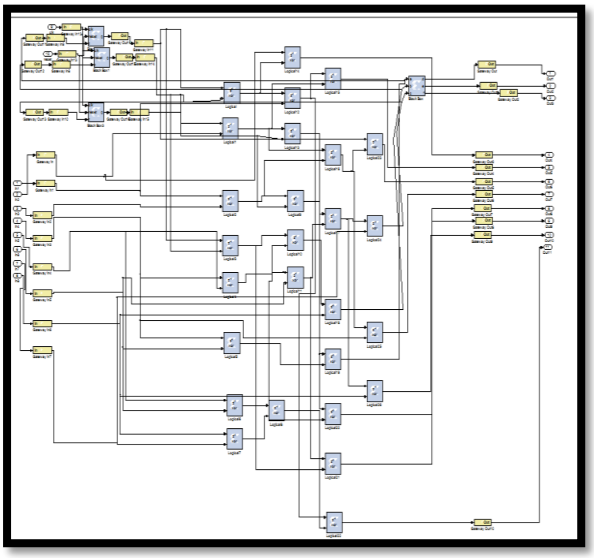

Figure 3. Proposed Convolutional Encoder

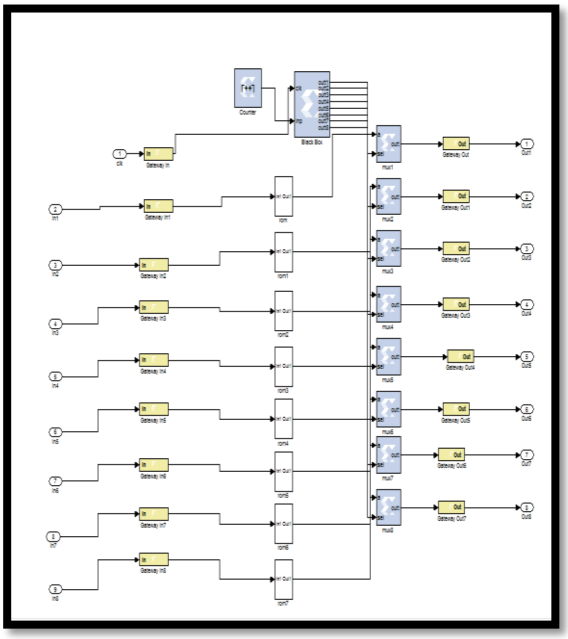

Figure 4. Proposed Interleaver

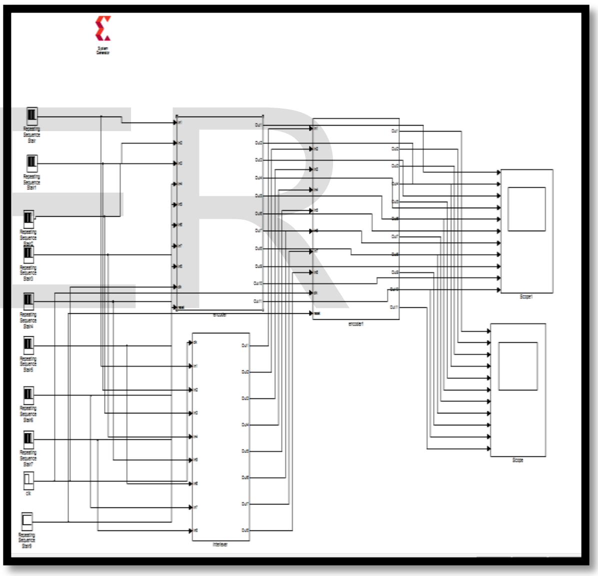

Figure 5. Proposed Optimized 8-level Turbo Encoder

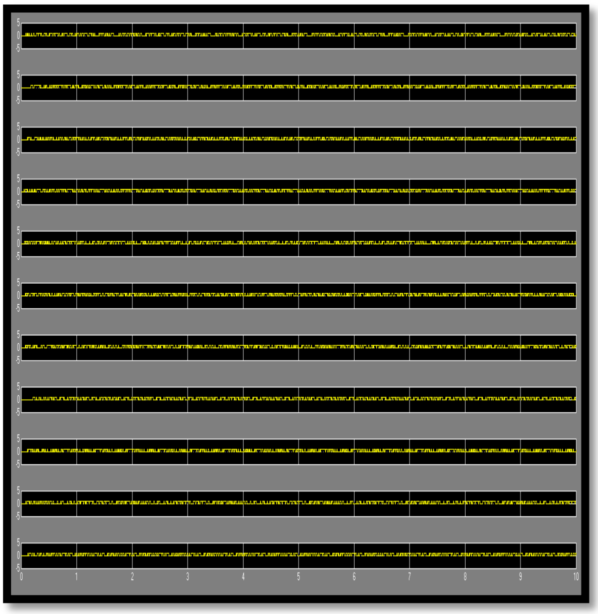

Figure 6. Proposed Optimized 8-level Turbo Encoder Interleaver Output

In the turbo encoder interleaver output, first three clock signals are Dual RAM and the next 8 clock signals are turbo coder interleaver output signals.

6 CONCLUSION

Turbo code encoder and algorithm of encoder is proposed to increase the hardware implementation performances of system. The proposed algorithm employs parallel processing architecure to improve the clock latency and reduce the size of encoder. 8-level parallel algorithm is chosen to support all block size possibilities in encoder. Double RAM is also used in the turbo code internal interleaver in order to reduce clock latency. To optimize the proposed algorithm and architecture, we implement recursive pairwise matching which significantly reduce the number of arithmetic operation. Based on the simulation results, proposed algorithm and architecture has successfully increase the speed of encoder by 16 times and reduce the size of encoder by 50%. From this result, it could be concluded that the proposed algorithm and architecture of turbo code encoder have met standard specification of LTE with satisfactory performances enhancement.

REFRENCES

[1] Ardimas Andi Purwita, Arnaud Setio, Trio Adiono Optimized 8-Level Turbo Encoder Algorithm and VLSI Architecture for LTE. 2011 International Conference on Electrical Engineering and Informatics 17-19 July 2011, Bandung, Indonesia

[2] Sophia Antipolis, Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (R elease 9) : ETSI, June, 2010

[3] C. Bruno, L. Angel, S. Stefania, R. Cornelius, Constantinos B. Papadias,3GPP LTE and LTE Advanced, EURASIP Journal on Wireless Communications and Networking, pp.1-3, 2009.

[4]M.Potkonjak, M.Srivastava, and A.Chandrakasan,, Multiple constant multiplications: Efficient and versatile framework and algorithms for exploring common subexpression elimination, , IEEE Trans. Computer-Aided Design, vol. 15, pp. 151-165, Feb. 1996.

[5] Yang Sun , Yuming Zhu , Manish Goel , Joseph R. Cavallaro, Configurable and scalable high throughput turbo decoder architecture for multiple 4G wireless standards, ,Proceedings of the 2008 International Conference on Application-Specific Systems, Architectures and Processors,p.209-214, July 02-04, 2008

[6] Codes,Jinghu Chen, Member, IEEE, Ajay Dholakia, Senior Member, IEEE, Evangelos Eleftheriou, Fellow, IEEE, Marc P. C. Fossorier, Senior Member and Xiao-Yu Hu, Member, IEEE, Reduced-Complexity Decoding of LDPC IEEE Transactions on Communications, Vol. 53, No. 8, August 2005

[7] Seok-Jun Lee, Member, IEEE, Naresh R. Shanbhag, Senior Member, IEEE, and Andrew C. Singer, Senior Member, IEEE ,Area-Efficient High-Throughput MAP Decoder Architectures , IEEE Transactions on Very Large Scale Integration (VLSI) Systems.

[8] Sun, Y., Zhu, Y., Goel, M., & Cavallaro, J. R. (2008).Configurable and scalable high throughput turbo decoder architecture for multiple 4G wireless standards. In *IEEE* International conference on application-specific systems, architectures and processors (ASAP).

- Author name

Prof.Amutha Prabha.N

School of Electrical Engineering

VIT University

Vellore-Tamil Nadu (INDIA)

E-mail: amuthaprabha@vit.ac.in

- Co-Author's name

1. Syed Farhathullah.S.F

B.Tech-Electronics and Instrumentation Engineering

VIT University

Vellore-Tamil Nadu (INDIA)

E-mail: syedfarhathullah@gmail.com

2.Venkatesh.D

B.Tech-Electronics and Instrumentation Engineering

School of Electrical Engineering

VIT University

Vellore-Tamil Nadu (INDIA)

E-mail: dvenkat.93@gmail.com