International Journal of Advances in Electronics Engineering– IJAEE
Volume 6 : Issue 1 *[ISSN : 2278-215X]*
Publication Date: 18 April, 2016

Hardware implementation of a Turbo Code with 3 Dimensions on FPGA

[Mensouri Mohammed, Aaroud Abdessadek and Ali El Hore]

***Abstract*—Recent wireless communication standards such as 3GPP-LTE, WiMax, DVB-SH, HSPA and LTE / LTE advanced incorporate turbo code for their excellent performance. In this paper, we present a new 3 dimensional turbo decoder including bit error rate (BER) is much better than the 2 dimensional turbo decoder used by LTE / LTE advanced, as is illustrated by simulation. We also address the issue of the implementation of the 3 dimensional turbo decoder on FPGA using the environment QUARTUS II. In this work, we also present the implementation on FPGA of 3 dimensional Turbo encoder using two interleavers QPP (Quadratic Permutation Polynomial) and ARP (Almost Regular Rotation). In decoding scheme, the core of the iterative decoding structure is a soft-input soft-output (SISO) decoder. The MAP algorithm, which is used for SISO decoders, embodies complex mathematical operations such as division, exponential and logarithm calculations. Therefore, MAP algorithm was avoided and the sub-optimal derivatives of this algorithm such as Log-MAP and Max-Log-MAP were preferred for turbo decoder implementations.**

***Keywords*— 3 dimensional turbo encoder, QPP interleaver, ARP interleaver, MAP algorithm, Log max MAP, SISO decoder, 3 dimensional turbo-decoder, FPGA**

I. Introduction

3GPP Long Term Evolution (LTE) [1] is a set of enhancements the 3G system i.e. UMTS (Universal Mobile Telecommunications System) [2]. It is a promising wireless 4G technology. The main advantages of the 3GPP LTE are:

- A high throughput approximately 326.8 Mbps for a system of antennas 4x4 and 172.8 Mbps for a system of antennas 2x2 for each 20MHz of spectrum. In addition, LTE-Advanced [3] the further evolution of LTE, promises to provide up to 1 Gbps peak data rate.
- A low error rate: The digital systems of transmission new generations require very low error rates can be up 10^{-8} dB [4]. And this can be achieved by increasing the minimum Hamming distance. This can be done using either :

- Encoder components with a larger number of states.
- Very appropriate internal permutations.
- Or by increasing the size of the turbo code, i.e. the number of encoder elementals constituting the encoder.

In this work, we are interested in increasing the encoder dimension, i.e. we will pass from a turbo code with 2 dimensions to turbo code with 3 dimensions (TC-3D).

Mensouri Mohammed, Aaroud Abdessadek and Ali El Hore
Department of Computer Science, Faculty of Sciences
Chouaib Doukkali University
El Jadida - Morocco

The schematic of the LTE channel coding is used the Turbo coding [5]. Turbo decoder is usually one of the main blocks of a wireless LTE receiver. Turbo Decoder suffers from high decoding latency due to its iterative decoding process, the forward–backward recursion in the maximum a posteriori (MAP) decoding algorithm and the interleaving/de-interleaving between iterations [6]. Typically, the task of an interleaver is to swap the soft values generated by the decoder Soft Input Soft Output (SISO) and write them in random or pseudo-random positions. As shown in Figure 1, the Turbo coding scheme in the LTE standard is the parallel concatenation of two constituent encoders with 8 states and QPP interleaver [7] i.e. a turbo code 2 with dimensions [5].

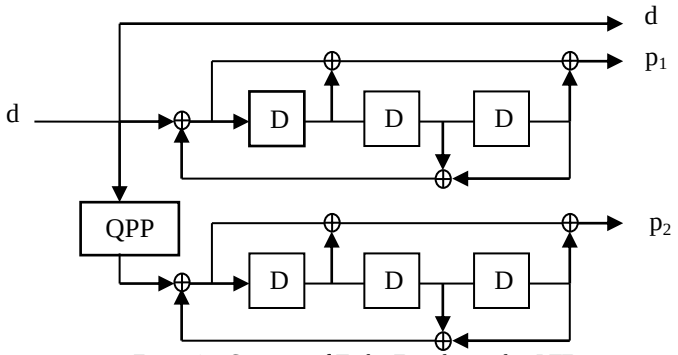

Figure 1. Structure of Turbo Encoder used in LTE

In this paper, we propose new hardware-efficient implementation for LTE coding channel based on a turbo encoder with 3 dimensions as illustrated in Figure 2 [8]. In order to implement an efficient turbo decoder, a suitable decoding algorithm has to be chosen. Turbo codes have been originally implemented with MAP algorithm [9]. However, this algorithm performs complex mathematical operations such as multiplication, division and logarithmic calculations. Therefore, engineers have avoided implementing this complex algorithm and preferred the sub-optimal derivatives of the MAP algorithm such as the Log-MAP and the Max-Log-MAP algorithms which are much simpler to implement but yield worse BER performances [10].

The remainder of this paper is organized as follows fawn: Section II gives a review of the basic principles of turbo codes with 3 dimensions. In Section III, the architecture of elementary SISO decoder is illustrated with their associated decoding algorithm. Section IV describes the architecture of the two interleavers QPP and ARP and some algebraic properties. In Section V, We present hardware implementation of 3 dimensional turbo encoder. In Section VI, We present hardware implementation of 3 dimensional turbo decoder using SISO decoders and many more QPP interleavers and ARP. In Section VII, the implementation results are summarized and compared with existing turbo decoders. We conclude in Section VIII.

II. Fundamentals of 3 dimensional Turbo Code

In order to implement the proposed 3 dimensional turbo encoder and 3dimensional turbo decoder architecture, the fundamentals of turbo code with 3 dimensions are briefly described in this section.

A. *Structure of the TC-3D encoder*

The turbo codes 3with dimensions [11] that will be used throughout this paper is built from a concatenation in parallel the three recursive convolutional systematic encoders identical, reread by two interleavers QPP and ARP (Figure 2). The three encoders have a constraint length $L = M + 1 = 4$ (i.e. 3 D registers). The overall encoder has the code rate $\frac{1}{4}$, so it possesses one input and four outputs. One of the outputs is the uncoded information sequence d, the other outputs are sequences of parities (p_1, p_2, p_3).

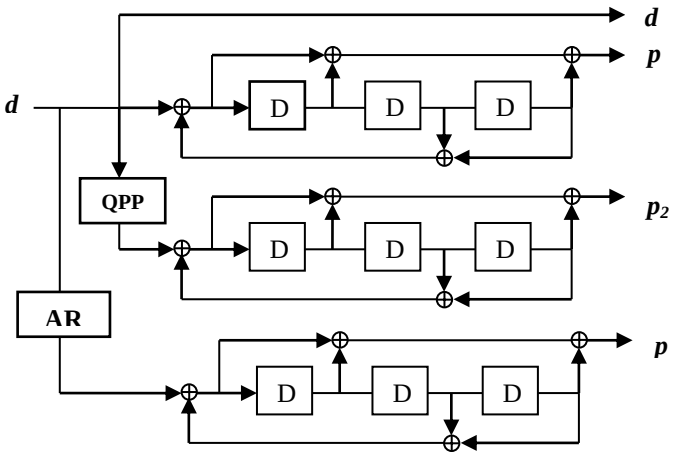

Figure 2. Structure of the TC-3D encoder

The three component encoders encode the same sequence of information in different orders. Each component encoder is the type recursive convolutional with 8 states and polynomials generators:

$$\boldsymbol{G1} = [\mathbf{1\ 1\ 0\ 1}]$$
$$\boldsymbol{G2} = [\mathbf{1\ 0\ 11}]$$

The information bits d are coding initially by the first encoder to provide a first bit of redundancy denoted p_1, those same information bits are interlaced by the interleaver QPP before being coded by the second encoder who delivered the second redundancy denoted p_2 . The information d is then interleaved by ARP interleaver before being coded by the third encoder. The latter issue is a third redundancy denoted p_3.

Finally, the sequence information d of K bits size and sequences of parities p_1 , p_2 and p_3 respective sizes n_1, n_2 and n_3 bits, they are multiplexed to form the code word u of length N bits and transmitted in the channel. In this case, the code rate of the turbo code is $R = \frac{k}{K+n_1+n_2+n_3} = \frac{k}{N}$. Then we can write:

$$u(d, p_1, p_2, p_3) \quad (1)$$

B. *Structure of TC-3D decoder*

After the step of coding, sequence information and three redundancy sequences (i.e. u) are transmitted on noise channel provided to receiver a sequence $r(d', p'_1, p'_2, p'_3)$ of N bits, the relation between u and r to the AWGN channel (Additive White Gaussian Noise) is:

$$r = u + b \quad (2)$$

Or b is a randomized sequence representing the "noise" or "Error" additive.

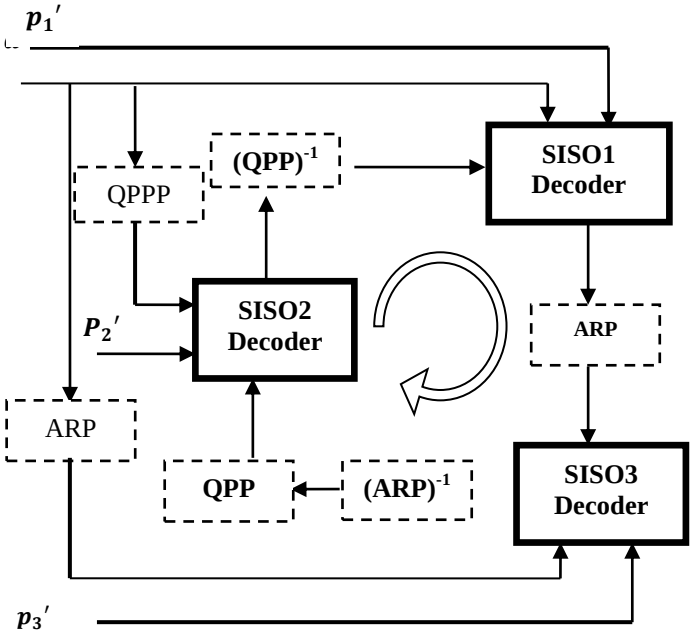

Figure 3. Schema of TC-3D decoder.

The turbo encoder with 3 dimensions can be decoding by using the principle of the turbo code with 2 dimensions. The architecture of the TC-3D decoder is composed of three decoders SISO, four interleavers and two deinterleavers as illustrated in Figure 3. d' is the sequence of received information corresponding to the sequence of transmitted information d. p'_1, p'_2 and p'_3 are the sequences of noisy information associated with redundancies sequences respectively p_1 , p_2 and p_3 . The information exchanged between the elementary decoders is described in the form of a conditional probability of the bit of information transmitted on the observations at the entrance of decoder. In the case of the binary code, the extrinsic information of the bit d_i with $i \in \{1, \dots, K\}$ is formulated like $P(d_i = 1|d')$ or $P(d_i = 0|d')$. The sizes of this quantity are very often considered in a form logarithmic curve such as Logarithms of Report Likelihood LRV:

$$L(d_i) = \ln\left(\frac{P(d_i=1|d')}{P(d_i=0|d')}\right) \quad (3)$$

The extrinsic information produced by a decoder is then considered as a priori information in between the other decoder. The three constitutive decoders carry out a maximum of probability a posteriori (MAP) of decoding on the level of the bit. They use the BCJR algorithm [9] transformed in the logarithmic domain in said Max Log MAP algorithm [10][12], to reduce implementation complexity.

III. Architecture of Decoder SISO Associated With Algorithm Max-Log-Map

A. *Algorithm Max-Log-MAP*

The MAP algorithm, abbreviated Maximum a posteriori, also called the BCJR algorithm [9] is an optimal solution for decoding convolutional codes. It provides the reliability associated each with decoded block. With this intention, the decoder seeks all the information blocks $i = 0,1, \dots, K-1$ in an exhaustive way such as:

$$P(\overline{d}_i/r) \geq P(d_i/r)$$

Decoding according to criterion MAP is difficult to integrate in a circuit into the complexity of calculation, imposed in particular by the exponential, multiplications, and divisions operations. In practice, there exist several sub-optimal algorithms which aim to facilitate the implementation. To bypass the very complex arithmetic operations cited above, one of the effective methods is to transfer the problem in a space logarithmic curve. This modification makes it possible respectively to transform the multiplications, divisions and the exponential operations in additions, subtractions and multiplications. The algorithm Max-Log-MAP was detailed at the algorithmic level in [12]. We make a short recall here and present an associated architecture of it.

The algorithmic transformation leads the redefinition of the calculations of MAP algorithm by using the following two equations:

$$\log(x+y) = \max(x,y) + \log(1+e^{|y-x|}) \quad (4)$$

$$\log(x+y) \approx \max(x,y) \text{ when } |x-y| \gg 0 \quad (5)$$

The decoding algorithm Max-Log-MAP is organized in the following way:

- To calculate branch metrics $\gamma_i(s',s)$

The branch metric calculation eliminates the exponential:

$$\gamma_i(s',s) = -\frac{1}{2\sigma^2}\|r_i - c(s',s)\|^2 \quad (6)$$

Where $c(s',s)$ is the expected symbol along the branch from state s' to state s.

- To calculate forward state metrics $\alpha_i(s)$.

The forward state metric is calculated in a recursive way starting from following expression:

$$\alpha_i(s) = \max_{s'}(\alpha_{i-1}(s') + \gamma_i(s',s)),\ i = 1, \dots, N-1 \quad (7)$$

With initial conditions

$$\alpha_0(s) = \begin{cases} 0; four\ \ s = 0 \\ -\infty; four\ s \neq 0 \end{cases}$$

The values α are then saved in a dedicated memory

- To calculate backward recursion $\beta_k(s)$

The backward state metric recursion becomes:

$$\beta_{i-1}(s') = max_{(s)}(\beta_i(s) + \gamma_i(s',s)),\ i = N, \dots, 2 \quad (8)$$

With initial conditions

$$\beta_N(s) = \begin{cases} 0; four\ \ s = 0 \\ -\infty; four\ s \neq 0 \end{cases}$$

- Calculating the LRV (Log of Likelihood Ratio) $L(d_i)$.

The log-likelihood ratio calculation becomes:

$$L(d_i) = \max_{\substack{(s',s) \\ d_i=+1}}(\alpha_{i-1}(s') + \gamma_i(s',s) + \beta_i(s))$$

$$- \max_{\substack{(s',s) \\ d_{i=-1}}}(\alpha_{i-1}(s') + \gamma_i(s',s) + \beta_i(s)) \quad (9)$$

- To make the decision $\overline{d}_i$ starting from Log-Likelihood $L(d_i)$:

$$\overline{d}_i = \begin{cases} +1 \text{ si } L(d_i) \geq 0 \\ -1 \text{ si } L(d_i) < 0 \end{cases} \quad (10)$$

The LLR is a convenient measure since it encapsulates both soft and hard bit information in one number. The sign of the number corresponds to the hard decision while the magnitude gives a reliability estimate and to calculate corresponding reliability. The algorithm Max-Log-MAP is simple that MAP algorithm, but at the cost of a degradation of performance of decoding.

B. *Architecture of Decoder SISO*

Decoder SISO is a component elementary of turbo decoder with 3 dimensions. In this section, we present architecture of decoder SISO applying the Max-Log-MAP algorithm. This architecture is broken up into two parts: a control part and a treatment part. The control part makes it possible to synchronize the components while the treatment part applies to the arithmetic operations. Decoder SISO calculates the extrinsic information z and provides a hard decision on the bit considered starting from the entering data d', p'_1, p'_2, p'_3 and z'.

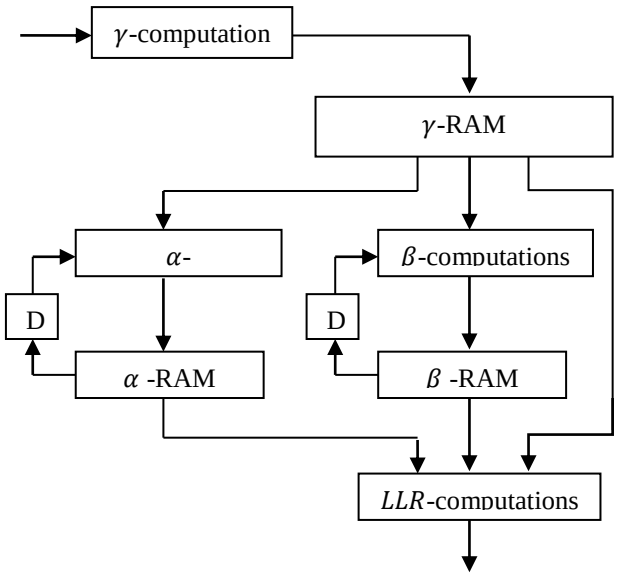

Figure 4. Architecture of decoder SISO applying the Max-Log-MAP algorithm

The block diagram of a typical implementation of Log Max-Log-MAP algorithm is shown in Figure 4. In this figure," γ-computation" represents the branch metrics," α-computations" represents the forward metrics and " β-computations" represents the backward metrics. In this design, the memories for branch and forward metrics are dispensable. After receiving data, " γ -computation" calculates all the branch metrics according to the trellis diagram for all states using equation (6) and stores them in the 'γ-RAM'. The " α-computations" calculates the forward state metrics using above mentioned equation (7) and computes during the forward recursions. The " β -computations" calculates the backward state metrics using above mentioned equation (8) and computes during the backward recursions. The log likelihood ratio is computed by using the "LLR-computations" which uses the equation (9).

IV. Implementation of the Interleavers

Interlacing consists in dispersing the sequence of the data in time. This technique does great favors in digital communications to reduce the effect of fainting. More particularly, this mechanism is very effective to fight against the disturbances of the consecutive symbols. Within the framework of the turbo-codes, the interlacing is a paramount operation. In our case we used two types of the interleaver: ARP interleaver and QPP interleaver.

A. Hardware Implementation of ARP interleaver

The interleaver ARP [13] is given by:

$$\pi(\mathrm{i}) = \big(\mathrm{Pi} + \mathrm{P}_0 + \mathrm{d(i)}\big)\mathrm{mod\ (K)} \qquad (11)$$

where $0 \le \mathrm{i} \le \mathrm{k}-1$ is the sequential index of the binary positions before interlacing, π (i) is the index of bits after interlacing corresponding to position i, K is the size block of the bits information, where P is prime with K. P_0 is a constant of shift and d (i) is a vector of "dither" of length C, where C is a small number (for example 4, 8) called the length of the cycle. For all block sizes, the juxtaposition of d(i) takes the form.

$$\mathrm{d(i)} = \alpha\big(\mathrm{i\ mod(C)}\big) + \mathrm{P}_0\beta(\mathrm{i\ mod(C)}) \qquad (12)$$

Where α (.) and β (.) are vectors each one length C, periodically applied for $0 \le \mathrm{i} \le \mathrm{K}-1$. The two vectors of tramage α (.) and β (.) are composed of multiples length of the cycle C.

For an interlacing ARP, since the size block K must be a multiple of C, certain sizes of block (for example, prime numbers) are not adapted for design ARP interlacing. However, this constraint is acceptable in practice since sizes of blocks (K) which are multiples of whole small numbers (C).

Since (11) is equivalent to a linear permutation with a constant which is function of index i, (11) and its reverse have identical format. That makes it possible to use only one circuit at the same time interlacing and deinterlacing [14].

B. Hardware Implementation of QPP Interleaver

The QPP interleaver can be expressed by a simple mathematical formula. Being given a block information lengthk, the i^{th} position of exit interlacing is specified by the quadratic expression [15][16]:

$$\mathrm{f(i)} = (\mathrm{f_2 i^2 + f_1 i})\mathrm{mod(K)} \qquad (13)$$

Where the parameters $\mathrm{f_1}$ and $\mathrm{f_2}$ are integers and depend on the size of block $0 \le \mathrm{i}, \mathrm{f_1}, \mathrm{f_2} \le \mathrm{k}$. For each bloc k size, a set of different parameters $\mathrm{f_1}$ and $\mathrm{f_2}$ are defined. Based on the algebra analysis in [15], the QPP interleaver is guaranteed to always generate a unique address which greatly simplifies the hardware implementation. The QPP interleaving address can be computed in a recursive manner. Suppose the interleaver starts at $\mathrm{i_0}$, we first pre-compute $\mathrm{f(i_0)}$ as:

$$f(i_0) = (f_2 i_0^2 + f_1 i_0) mod(k) \qquad (14)$$

In the following cycles, as i is incremented by d, f(i + d) is computed recursively as follows:

$$f(i+d) = \big(f(i) + g(i)\big) mod(k) \qquad (15)$$

Where g(i) is defined as:

$$g(i) = (2df_2 i + d^2 f_2 + df_1) mod(k) \qquad (16)$$

Note that g(i) can also be computed in a recursive manner:

$$g(i+d) = (g(i) + 2d^2 f_2) mod(k) \qquad (17)$$

The initial value $\mathrm{g(i_0)}$ needs to be pre-computed as:

$$g(i_0) = (2df_2 i_0 + d^2 f_2 + df_1) mod(k) \qquad (18)$$

The modulo operation in (15) and (18) can be difficult to implement in hardware if the operands are not known in advance. However, by definition we know that both f(i) and g(i) are less than k so calculating (15) and (17) can be realized by additions. In the proposed method, three numbers need to be pre-computed: $(2\mathrm{d}^2\mathrm{f_2})\mathrm{mod(k)}$, $\mathrm{f(i_0)}$, and $\mathrm{g(i_0)}$. Figure 5 shows a hardware architecture to compute the interleaving address f(x), where i starts from $\mathrm{i_0}$ and is incremented by d on every clock cycle..

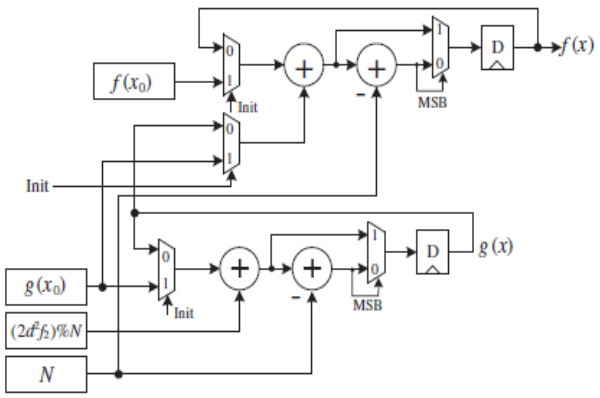

Figure 5. Forward QPP address generator circuit diagram, step size d

V. Implementation of 3 Dimensional Turbo Encoder

In this section, we present the implementation of a turbo encoder with 3 dimensions on FPGA using Quartus II. This turbo encoder consists of three 8-sate identical constituent convolutional codes separated by two interleavers. Figure 6 shows the signals of input / output of the turbo encoder 3 with dimensions:

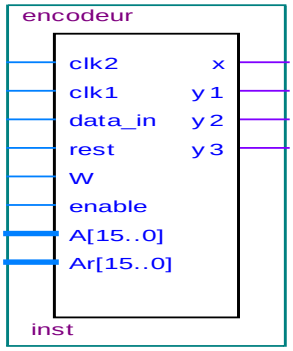

Figure 6. Turbo encoder 3 dimensional block

Clk1,*Clk2*: Clock signals credit in a high state

data_in : input signal of the bits to be coded

rest: Signal to reset the encoder

W : enabling signal of writing on the RAM memory

enable : Signal to enable the encoder

A : address Signal to store data in RAM

Ar : Address signal for reading data from the RAM

x, y1,y2,y3 : output signals of the coded bits

Figure 6 shows the overall architecture of the 3 dimensional turbo encoder. The data generated is then stored in a dedicated memory, denoted M_d. The coding data and the data generation occur in parallel. We therefore need a memory of which the length of the memory is twice as large as that of the frame generated. This memory has two access, dedicated to reading and writing data, which are clocked by two separate clocks *clk1* and *clk2*. The operations of reading and writing are performed in parallel. The control signals are provided by a controller. Reading the memory M_d is controlled by the signal *W*. In addition, the data is read using the respective addresses A_r and *A*. As regards the interleaver associated with encoder, it is equivalent to the turbo decoder.

Table I shows the result of synthesis to the 3 dimensional turbo encoder implemented by Quartus II.

TABLE I. THE RESOURCES USED BY THE TURBO ENCODER WITH 3 DIMENSIONS.

Resources	number of elements
Total logic elements	328
Total combinational function	325
Dedicated logic elements	160
Total pin	42
Total memory bits	65536
Embedded multiplier 9-bitelements	0
Total PLLs	0

VI. Implementation of Turbo Decoder with 3 Dimensions

The architecture of the turbo decoder 3D is connected to various elements: the data memories and the block measures the binary error rate TEB (Figure 7). The process of turbo-decoding treats the data coming from the memories, and transmits extrinsic information to the memory during iterations. In parallel, the TEB is calculated. The memories of received information X, Y_1, Y_2 and Y_3 are regarded as external storages with the decoder.

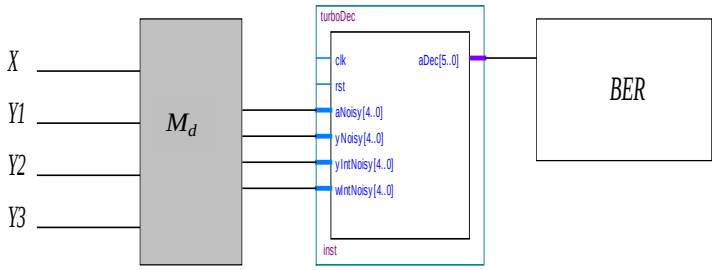

Figure 7. FPGA Architecture of 3dimensional Turbo Decoder

Clk : Clock signals credit in a high state

rst : Signal to reset the encoder

aNoisy: Signal represents the information bits noisy to be decoded

yNoisy: Signal represents the bits of the first parity noisy to be decoded

yIntNoisy : Signal represents the bits of the second parity noisy to be decoded

wIntNoisy : Signal represents the 3rd parity bits noisy to be decoded

aDec: Signal represents the bits decoded by the turbo decoder

A simulation of the decoder is carried out to check the good performance of the decoder by using the tool for simulation Quartus II. Table II presents the synthesis of architecture turbo decoder to 3 dimensions

TABLE II. LES RESSOURCES UTILISEES PAR LE TURBO DECODEUR

Resources	number of elements
Total logic elements	8103
Total combinational function	7751
Dedicated logic elements	1092
Total pin	22
Total memory bits	0
Embedded multiplier 9-bitelements	0
Total PLLs	0

VII. Implementation Results and Comparison

A. *Error-Rate Performance*

Figure 8 shows the bit error rate (BER) performance of the 3 dimensional turbo decoder using Max-log-MAP. The BER performance of the decoder is significantly better than the turbo decoder with 2 dimensions.

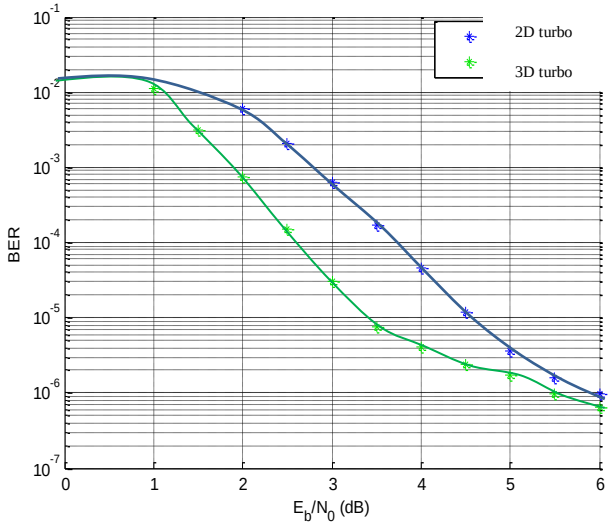

Figure 8. Comparison of BER between 3dimensional turbo decoder and 2 dimensional turbo decoder

B. *Complexity of 3-dimensional turbo decoder*

In this section, we propose the complexity of a turbocode 3 dimensions. Table III compares the hardware complexity of the turbo decoder 2 dimensional and the corresponding turbo decoder 3 dimensional. Table III provides the complexity of the overall hardware dedicated to SISO decoding with the Max-Log-MAP algorithm in terms of addition / compare-select operators. In fact, compared to a classical turbo decoder, the additional complexity of the turbo decoder 3D is mainly due to the implementation of the SISO3 decoder but also to the calculation of the extrinsic information about the encoded parity bits p'_3. The increased complexity was estimated to be less than 10% with respect to classical 2 dimensional turbo code.

TABLE III. COMPLEXITY COMPARISON OF A 3DIMENSINAL TURBO DECODER AND 2DIMENSIONAL TURBO DECODER.

	Add(or subtract)	Compare-select
Branch metrics	$2^{n+1}-2$	
(forward or backward)	2^{v+1}	2^{v}
A posteriori LLRs and hard decision	$2^{v+1}+1$	$2^{v+1}-1$
Extrinsic LLRs for information bits	4	
Extrinsic LLRs for redundancy bits (only for 3D TC)	2	$2^{v+1}-2$
Total 2D TC	$3\times 2^{v+1}+2^{n+2}+1$	$2^{v+1}-2$
Total 3D TC	$3\times 2^{v+1}+2^{n+2}+3$	$3\times(2^{v+1}-2)$

VIII. Conclusion

This paper provides brief discussion on one of the optimal channel coding technique such as turbo code with 3 dimensions and gave brief analysis about the turbo encoder and decoder structure. We have presented implimentation on FPGA for the 3 dimensional encoder and 3 dimensional turbo decoder of LTE/LTE-Advance. As indicated in Section VII, 3 dimensional turbo code is compared with the 2 dimensional turbo code. It is observed that the 3 dimensional turbo code yields a better BER performance than the 2 dimensional turbo code as expected. In addition we compare the implementation complexity of the 3 dimensional turbo decoder with the 2 dimensional turbo decoder on FPGA. The complexity increase was estimated to be less than 10% with respect to classical 2-dimensional turbo code.

References

[1] Evolved Universal Terrestrial Radio Access(EUTRA) and Evolved Universal Terrestrial Radio Access Network (EUTRAN), 3GPP TS 36.300.

[2] General UMTS Architecture, 3GPP TS 23.101 version7.0.0,June2007.

[3] S.Parkvall, E.Dahlman, A.Furuskar, Y.Jading, M.Olsson, S.Wanstedt, K. Zangi, LTE-advanced evolving LTE towards IMT—advanced, in:IEEE Vehicular Technology Conference, September 2008, pp.1–5

[4] C. Berrou, A. Graell i Amat, Y. Ould Cheikh Mouhamedou, C. Douillard, and Y. Saouter, “Adding a Rate-1 Third Dimension to Turbo Codes," in Proc. IEEE Information Theory Workshop ITW ’07, pp. 156-161, 2-6 Sept. 2007.

[5] Multiplexingandchannelcoding,3GPPTS36.212version8.4.0,Septembe r 2008.

[6] C.Berrou,A.Glavieux,P.Thitimajshima, Near shannon limit error-correcting coding anddecoding: turbo-codes, in: IEEE International Conferenceon Communication, May1993,pp.1064–1070.

[7] C. Berrou, Y. Saouter, C. Douillard, S. Kerouedan, M. Jezequel, “Designing good permutations for turbo codes: towards a single model,” in Proc. of ICC 2004, vol. 1, June 2004, pp. 341-345.

[8]] A. Aaroud and M. Mensouri, "Performance Analysis of 3-Dimensional Turbo Codes," International Journal of Computing & Information Technology (IJCIT), vol. 4, no. 1, pp. 17-24, Juin 2012.

[9] Bahl, L. R., Cocke, J., Jelinek, F., Raviv, J., “Optimal Decoding of Linear Codes for Minimizing The Symbol Error Rate”, IEEE Transactions on Information Theory, Vol. 20, pp. 284-287, 1974.

[10] Robertson, P., Hoeher P., “Optimal and Sub-OptimalMaximum a Posteriori Algorithms Suitable for Turbo Decoding”, European Transactions on Telecommunications, Vol. 8, pp.119-125, 1997

[11] M. Mensouri and A. Aaroud, "Weight Distribution and Bounds of Turbo-Code with 3 Dimensions," The SIJ Transactions on Computer Networks & Communication Engineering (CNCE), The Standard International Journals (The SIJ), vol. 1, no. 1, pp. 18-23, April 2013

[12] P. Robertson, P. Hoeher, and E. Villebrun, “Optimal and Sub-Optimal Maximum a Posteriori Algorithms Suitable for Turbo Decoding," European Transactions on Telecommunications (ETT), vol. 8, pp. 119-125, March-April 1997.

[13]] C. Berrou, Y. Saouter, C. Douillard, S. Kerouédan, and M. Jézéquel, “Designing good permutations for turbo codes: Toward a single model,” in Proc. IEEE Int. Conf. Commun., Paris, France, Jun. 2004, pp.341–345.

[14] Ajit Nimbalker, Yufei Blankenship, Brian Classon, T. Keith Blankenship” ARP and QPP Interleavers for LTE Turbo Coding” Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE. April 3 2008.

[15] J. Sun, O.Y. Takeshita, Interleavers for turbo codes using permutation polynomials over integer rings, IEEE Trans. Inform. Theory 51 (January) (2005) 101–119.

[16] Y. Sun and J. R. Cavallaro, "Efficient hardware implementation of a highly-parallel 3GPP LTE/LTE-advance turbo decoder," Integration, the VLSI Journal, vol. 44, no. 4, pp. 305-315, 2011.