View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Institute of Advanced Engineering and Science

International Journal of Electrical and Computer Engineering (IJECE)
Vol. 7, No. 4, August 2017, pp. 1824~1832
ISSN: 2088-8708, DOI: 10.11591/ijece.v7i4.pp1824-1832

VHDL Design and FPGA Implementation of a High Data Rate Turbo Decoder based on Majority Logic Codes

A. Boudaoud, M. El Haroussi, E. Abdelmounim
ASTI Laboratory, FST Settat, Hassan I University, Settat, Morocco

Article Info

Article history:

Received Jan 6, 2017
Revised Mar 10, 2017
Accepted Mar 27, 2017

Keyword:

Error correcting codes
FPGA implementation
Interleaver
ML-DSC codes
Turbo decoding
VHDL language

ABSTRACT

This paper presents the electronic synthesis, VHDL design and implementation on FPGA of turbo decoders for Difference Set Codes (DSC) decoded by the majority logic (ML). The VHDL design is based on the decoding Equations that we have simplified, in order to reduce the complexity and is implemented on parallel process to increase the data rate. A co-simulation using the Dsp-Builder tool on a platform designed on Matlab/Simulink, allows the measurement of the performance in terms of BER (Bit Error Rate) as well as the decoder validation. These decoders can be a good choice for future digital transmission chains. For example, for the Turbo decoder based on the product code DSC $(21.11)^2$ with a quantization of 5 bits and for one complete iteration, the results show the possibility of integration of our entire turbo decoder on a single chip, with lower latency at 0.23 microseconds and data rate greater than 500 Mb/s.

Copyright © 2017 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Abdelghani Boudaoud,
ASTI Laboratory, FST Settat,
Hassan I University,
B.P: 577, Casablanca Road, Km 3, Settat, Morocco.
Email: boudaoud586@gmail.com

1. INTRODUCTION

The discovery, of Turbo-Codes by C.Berrou [1] in 1993, represents an essential step forward for information transmission systems. Indeed, most terrestrial and satellite transmission standards have also adopted them. Thus NASA uses it in all their space probes since 2003; similarly, the European Space Agency (ESA) lunar probe SMART-1. Turbo-codes are also used as in UMTS, ADSL-2 and in mobile networks 4G-LTE and LTE-Advanced [20].

Initially, the turbo codes were based on convolutional codes concatenated in parallel. R.Pyndiah in 1994 [2] proposed the turbo block codes (TBC), which are an alternative to turbo convolutional codes. These TBC used the decoding weighted inputs and outputs (SISO). The iterative decoding process that we use, follows the model proposed by Pyndiah and built from the "One Step Majority Logic Decodable Codes" (OSMLD) [3] using the soft-out extension threshold decoding classic of Massey [4-9], [14].

The majority decoding uses a linear combination of a reduced set of syndromes represented by the orthogonal Equations.

The current digital systems and channel coding especially decoding, require high data rates. Decoding rate depends on the chosen architecture in the electronic design; the complexity of the circuit (ie its surface) is often considered as a critical parameter. In this context of very high throughput, the operating speed must be maximized while minimizing complexity.

2. CODING AND DECODING OF PRODUCT CODES

2.1. Product Codes Construction

Consider tow systematic block linear codes $C_1(n_1, k_1, d_1)$ and $C_2(n_2, k_2, d_2)$. Where n_i, k_i and d_i are, in order: the code word length, the information symbols number and the Hamming distance (i = 1,2).

Product code $CP=C_2 \otimes C_1$ built based on C_1 and C_2, which has the parameters $(n_1 \times n_2, k_1 \times k_2, d_1 \times d_2)$, is obtained by encoding the $k_2 \times k_1$ information symbols by the C_1 code and $k_2 \times n_1$ symbols by C_2 code, see Figure 1. In this work, we choose $C_1=C_2$ and, DSC(7,3,1), DSC(21,11,2) and DSC(73,45,4) codes.

2.2. Turbo Decoding Principle

A turbo decoder consists of SISO decoders (generally two decoders) and interleavers as shown in Figure 2. The channel symbols are received by line machine, by the first decoder which gives soft priori information, and then the channel information and the extrinsic information are interleaved and supplied column by column, to the input of the second decoder. A complete iteration consists of activating each decoder once. Thus, with more iteration, the decoder converges to the right solution, however it requires more time.

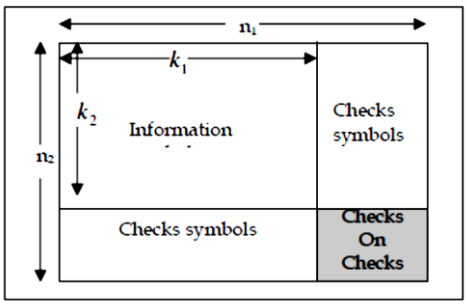

Figure 1. Product code principle

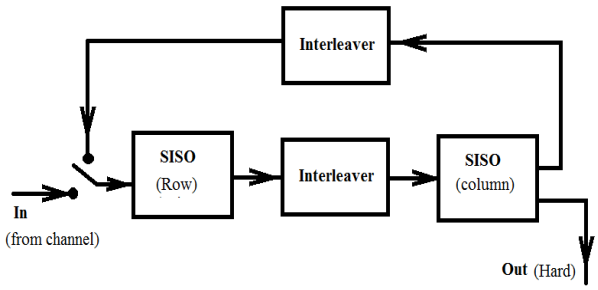

Figure 2. Structure of the turbo decoder

3. THE SISO DECODER

3.1. Threshold Decoding Algorithm

The SISO (Soft In-Soft Out) decoder is the base element of the turbo decoding. We used SISO decoder for which the threshold decoding algorithm [3], [4], [13] is given below:

For each j = n to 1

a. Calculate the terms Bi and ωi with iϵ {1,..,M}
b. Calculate B_0 and w_0
c. Calculate the extrinsic information $W(y_j) = \sum_{i=1}^{M} (1-2B_i)\omega_i$
d. Calculate $LLR(y_j) = \frac{4Es}{N_0} y_j + Wy_j$
e. If $(LLR(y_j)>0)$ ➔ Hard Decision =1
   else ➔ Hard Decision =0

Where:

a. n: Code length;

b. M: number of orthogonal Equations;
c. Bi: the orthogonal Equation on the i^{th} bit, after remove of i^{th} bit;
d. ωi: proportional to the reliability of i^{th} parity Equation,
e. W (y_j): extrinsic information representing the estimated orthogonal Equations on the symbol y_j;
f. LLR (y_i) the decision function on the symbol y_j;

And:

$$\omega i = \ln\left[\frac{1+\prod_{k=1,i\neq k}^{k=ni}\tanh(\frac{L_{ik}}{2})}{1-\prod_{k=1,i\neq k}^{k=ni}\tanh(\frac{L_{ik}}{2})}\right] \quad (1)$$

3.2. Structural Diagram of SISO Decoder

We have designed, in VHDL, a SISO decoder, which can be used in an iterative process. This iterative process we use follows the model proposed by Pyndiah [3]; See Figure 3.

The soft input and respectively the soft output of the q^{th} step (half-iteration) of the iterative decoding are given by:

$$R(q+1) = R + \alpha(q)W(q)$$

$$LLR(q) = 4\frac{E_s}{N_0}R(q) + W(q)$$

Where R represents the lines or columns of the received and quantified word, W(q) the extrinsic information calculated by the previous decoder and α is a coefficient which varies with each iteration. In addition to its threshold output, the decoder has a hard decision output that we used for its validation.

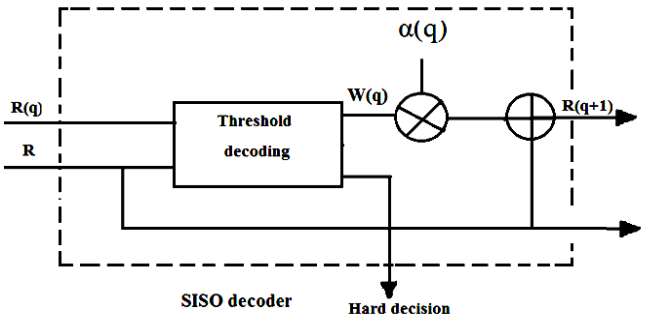

Figure 3. SISO Decoder (elementary cell of turbo decoding)

3.3 FPGA Implementation of the SISO Decoder

To reduce the complexity of the algorithm, we used, to simplify the expression (1), the Equation (2) below proposed by [12] and applied to the majority logic codes by [13], [11], [21].

The simplified expression of our decoder becomes:

$$\omega i \approx \min_{k=1}^{ni}|L_{ik}| * \prod_{k=1}^{n} sign(L_{ik}) \quad (2)$$

The architecture of the proposed SISO decoder was described in VHDL and implemented on FPGA using Quartus II tool from Altera. We used The EPC4CE115F29C7 FPGA type, containing about 115,000

LEs. We studied the complexity evolution of our decoders for different quantization bit number using C(7,3)⊗C(7,3), C(21,11)⊗C(21,11) and C(73,45)⊗C(73,45) product codes.

The results obtained, after the Quartus II synthesis, are summarized in Table 1. Figure 4 shows a functional simulation example of our SISO decoder for DSC (7,3) code on which it is reported that the outputs are updated at each rising edge of the clock, hence the latency is one clock cycle.

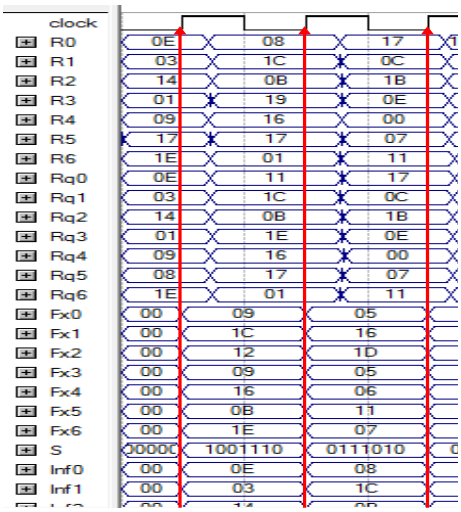

Figure 4. Simulation example of our SISO decoder for the DSC (7,3) code, with 5 bits of quantization.

Table 1. The Obtained Data for Different SISO Decoders

SISO Decoders	Characteristics	Quantization bit number			
		4 Bits	5 Bits	6 Bits	7 Bits
DSC(7,3)	Complexity(LE)	714	994	1141	1302
	Frequency max. (MHz)	47.62	43.71	41.51	41.46
	throughput max.(Gb/s)	0.33	0.31	0.29	0.29
	Latency (ns)	21.00	22.88	24.09	24.12
DSC(21,11)	Complexity(LE)	5334	6496	7749	8863
	Frequency max. (MHz)	24.99	24.30	23.58	23. 34
	throughput max.(Gb/s)	0.52	0.51	0.50	0.49
	Latency (ns)	40.02	41.15	42.41	42.84
DSC(73,45)	Complexity(LE)	52516	63155	78833	90513
	Frequency max. (MHz)	20.02	18.56	16.86	16.12
	throughput max.(Gb/s)	1.46	1.35	1.23	1.18
	Latency (ns)	49.95	53.88	59.31	62.03

4. THE INTERLEAVER

4.1. The Interleaver Principle

An interleaver is a system that receives a sequence of symbols in its input and provides another of the same alphabet to the output in a completely different order. In our case of product codes, the operation of the interleaver is summarized in two stages: receipt of all symbols of the product code matrix n2*n1, and swapping row-column of this matrix.

4.2. FPGA Implementation of the Interleaver

The used interleavers are described in VHDL and implemented on the same FPGA circuit using Quartus II tool for C(7,3)⊗C(7,3), C(21,11)⊗C(21,11) and C(73,45)⊗C(73,45) product codes. The Figure 5 shows the external structure of the interleaver design for the C(7,3)⊗C(7,3) product code.

In Figure 6, which illustrates an example of functional simulation of that interleaver, we note that input lines are recovered as columns on the outputs with latency equals to the length of the base code, seven clock periods in this case.

Table 2 summarizes the readings of the complexity, the maximum frequency and the maximum throughput, for different interleavers depending on the quantization bit number.

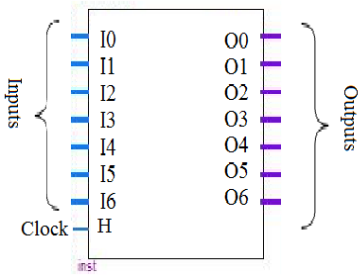

Figure 5. Interleaver structure for DSC(7,3)⊗C(7,3) product code

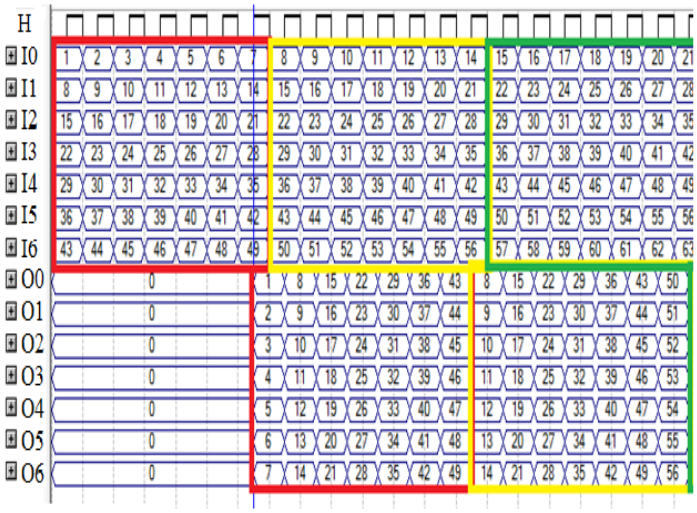

Figure 6. Example of functional simulation of the interleaver for DSC(7,3) code on Quartus II software.

Table 2. Caracteristics of the Intereleavers

Interleaver	Characteristics	Quantization bit number			
		4 Bits	5 Bits	6 Bits	7 Bits
7 To7 Symbols	Complexity(LE)	570	708	850	988
	Frequency max. (MHz)	219.44	218.67	209.16	203.04
	throughput max.(Gb/s)	1.54	1.53	1.46	1.42
	Latency (ns)	31.90	32.01	33.47	34.48
21 To 21 Symbols	Complexity(LE)	4643	5799	6962	8113
	Frequency max. (MHz)	152.00	148.19	139.94	130.77
	throughput max.(Gb/s)	3.19	3.11	2.94	2.75
	Latency (ns)	138.16	141.71	150.06	160.59
73 To 73 Symbols	Complexity(LE)	57388	71867	86014	100302
	Frequency max. (MHz)	84.31	82.22	80.03	76.95
	throughput max.(Gb/s)	6.15	6.00	5.84	5.62
	Latency (ns)	865.85	887.86	912.16	948.67

5. THE TURBO DECODER

5.1. FPGA Implementation of Turbo Decoder

Our turbo decoder consists of two SISO decoder and four interleavers similar to those discussed above, see Figure 7. Figure 8 shows the evolution of the silicon surface (expressed in logic element LE), occupied by the components of the Turbo decoder for C(7,3)⊗C(7,3), C(21,11)⊗C(21,11) and C(73,45)⊗ C(73,45) product codes, with a quantization of 5 bits. Figure 9 illustrates this surface with quantization of 3 to 7 bits for C(21,11)⊗C(21,11) codes.

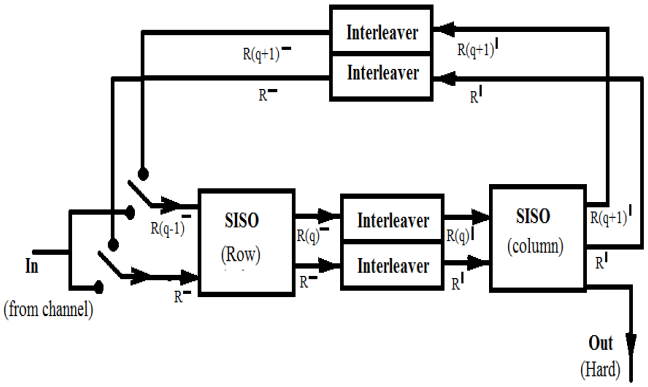

Figure 7. Detailed structure of our Turbo decoder

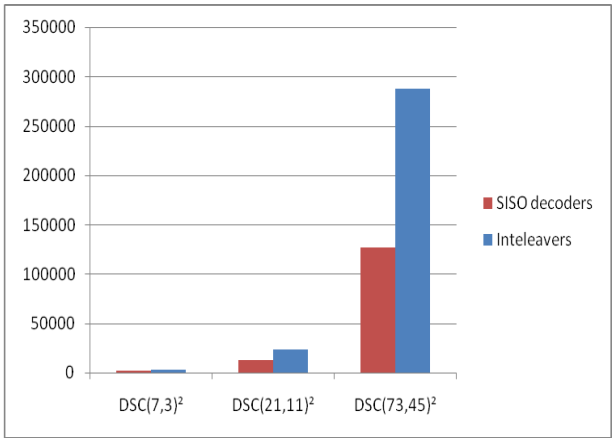

Figure 8. Area occupied by different SISO decoders for quantization of 5 bits

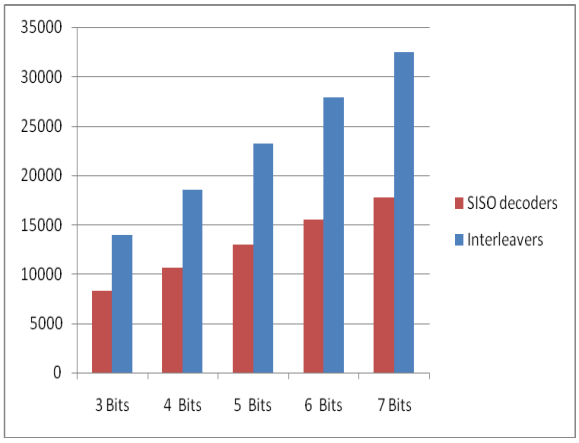

Figure 9. DSC(21,11)² SISO decoders occupied area for different bits of quantization (3 to 7 bits)

5.2. Validation of our Turbo Decoder

After functional simulation on Quartus II software, our design is also validated on a co-simulation chain on Matlab / Simulink and DSP Builder.

The co-simulation chain proposed (Figure 10) consists of the main components of a digital transmission chain: the information to be transmitted is generated in a random manner, coded by a product coder before being modulated on Binary Phase Shift Keying (BPSK) modulation. The channel, AWGN (Add white Gaussian Noise) will insert random errors in the information received. The block called in the figure 11 "Turbo decoder" contains the Turbo decoder to test, functions for the quantization and a data adaptation.

Quantization assigns to each symbol a fixed number of bits, according to the decoder. The loop is closed by a counter of "bit error" in order to measure the BER performance (Bit Error Rate) versus SNR(Signal-to-Noise Ratio). Figure 11 illustrates the performance for the DSC(21,11) turbo decoder at the first iteration in the theoretical case and in VHDL with quantization of 4, 5 and 7 bits.

In Figure 12, we presented the BER performance of the same turbo decoder with a quantization of 5 bits for 1st, 2nd and 3rd iteration

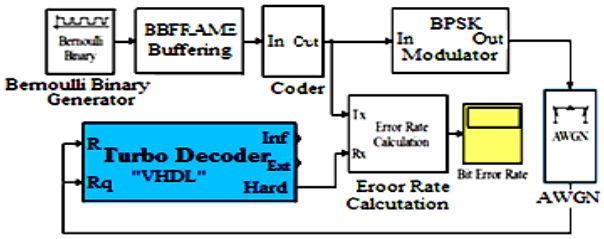

Figure 10: the proposed co-simulation chain for the BER performance measurement of the turbo decoder

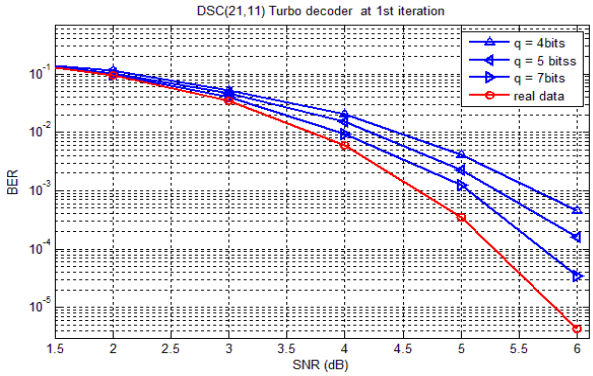

Figure 12. BER performances of Turbo decoder for 1st, 2nd and 3rd iteration

5.3. Performances of our Turbo Decoder

Our turbo decoder designed and implemented on the **EPC4CE115F29C7** FPGA circuit presents the performances summarized in Table 3. It is evident from Table 4, that the proposed design has a good performance compared to other recent architectures; and especially the latency that is nine times lower than the least one in the other designs.

Table 3. Summary of DSC (21, 11) Turbo Decoder Performances With a Quantization of 5 Bits.

Iteration	Path*	Throughput max. (Mb/s)	Latency min. (µs)	Complexity (LE)	Occupation (%)	BER at SNR=3dB
1	2S+I	510	0.224	24 590	21.38%	$4.72\ 10^{-2}$
2	4S+3I	255	0.590	36 188	31.46%	$9.91\ 10^{-4}$
3	6S+5I	170	0.956	36 188	31.46%	$5.62\ 10^{-5}$

(*) Information path: **S** for SISO decoder and **I** for Interleaver

Table 4. Comparaison With Others Turbo Decoder Design

	This work		[18]	[15]	[16]	[17]
	2nd iter.	3rd iter.	2010	2005	2003	2013
Number of Iterations	2	3	5	6	4	NA
Throughput (Mb/s)	255	170	930	758	75.6	346
Latency(µs)	0.59	0.956	5.5	10.5	5.35	NA
Frame length	441	441	5120	5120	432	NA
BER at SNR=3dB	$9.9\ 10^{-4}$	$5.6\ 10^{-5}$	$4\ 10^{-6}$	NA	$4\ 10^{-7}$	$2\ 10^{-3}$
Modulation	BPSK	BPSK	BPSK	NA	NA	BPSK
Channel type	AWGN	AWGN	AWGN	NA	NA	AWGN
Code rate	0.274	0.274	NA	NA	0.33	0.33

6. CONCLUSION

The approach of the VHDL design using a parallel mode and the FPGA implementation of the proposed turbo decoder has allowed us a high data rate, low complexity and very low latency. This will justify its use in future communication channels.

Use of a FPGA circuit with better "Speedgrad" could further increase data rate and reduce latency. In addition, a review of the interleaver architecture is necessary to reduce more its complexity.

REFERENCES

[1] C Berrou, A. Glavieux, P. Thitimajshima, *"Near Shannon Limit Error Correcting Coding and Decoding: Turbo Codes"*, IEEE International Conference on Communication ICC93, vol. 2/3, May 1993.

[2] R. Pyndiah, A. Glavieux, A. Picart, S. Jacq, "Near Optimum Decoding of Product Codes", Globecom'94, San Fransisco,1994.

[3] M. Belkasmi, M. Lahmer, F. Ayoub, "*Iterative Threshold Decoding of Product Codes Constructed from Majority Logic Decodable Codes"*, ICCTA conf, Damascus,Syria, pp. 2376- 2381, April 2006.

[4] J. L. Massey, "Threshold Decoding", M.I.T. Press, Cambridge, Massachusetts, 1963.

[5] E. J.. Weldon Jr, "Difference-Set Cyclic Codes", *Bell System Technical Journal*, Vol.45, pp. 1045_1055, sep 1966.

[6] R. L. Graham, F. J. MacWilliams, "On the Number of Information Symbols in Difference-Set Cyclic Codes", *Bell System Technical Journal*, vol. 45, pp. 1057-1070, Sep 1966.

[7] J. Singer, "A Theorem in Finite Projective Geometry and Some Applications to Numbers Theory", *AMS Trans.*, vol. 43, pp. 377-385, 1938.

[8] F. J. MacWilliams, "A Table of Primitive Binary Idempotents of odd Length n, 7 ≤n ≤511", *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 118–123, Jan. 1979.

[9] F. J. MacWilliams, J. Seery, "The weight Distributions of some minimal Cyclic Codes," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 796–806, Nov. 1981

[10] C. Clarck, B. Cain, "Error-Correction Coding for Digital Communications," Plenum Press, 1983.

[11] M. Elharoussi, M. Belkasmi, "VHDL Design and FPGA Implementation of a Fully Parallel Architecture for Iterative Decoder of Majority Logic Codes for High Data Rate Applications", *Journal of Wireless Networking and Communications (JWNC)*, 2(4), pp. 35-42, 2012

[12] J. Hagenauer, E. Offer, L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes", *IEEE Trans. Inform. Theory*, Mars 1996, Vol. 42, pp. 429-446.

[13] A.Boudaoud, E.Abdelmounim, A. Barazzouk, J. Zbitou, M.Belkasmi, "FPGA Implementation of HIHO and SIHO Decoders for DSC Codes", *IEEE explorer*, pp. 1461 – 1464, April 2014

[14] L. D. Rudolph, "Geometric Configuration and Majority Logic Decodable Codes", PhD thesis, University of Oklahoma, Norman, 1964.

[15] G. Prescher, T. Gemmeke, T.G. Noll, *"A parametrizable Lowpower High-throughput Turbo-decoder,"* in IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP), volume 5, pages 25-28, Mar. 2005.

[16] B. Bougard, A. Giulietti, V. Derudder, J. Willem, S. Dupont, F. Catthoor, L. Hollevoet, L. V. der Perre, H. D. Man, R. Lauwereins, *"A scalable 8.7 nj/bit 75.6 Mb/s parallel concatenated convolutional (turbo-)codec,"* in IEEE Int. Solid-State Circuit Conf., pp. 152–484, Feb. 2003.

[17] Rahul Shrestha, Roy P Paily, *"Design and Implementation of a High Speed MAP Decoder Architecture for Turbo Decoding"*, 26th IEEE International Conference on VLSI Design and the 12th IEEE International Conference on Embedded Systems, pp. 86-91, 2013.

[18] S. M. Karim, I. Chakrabarti, "An Improved Low-power High-throughput log-MAP Turbo Decoder*", IEEE Trans. Consum. Electron.*, vol. 56, no. 2, pp. 450-457, 2010

[19] A. Z. Jidin, T. Sutikno, "FPGA Implementation of Low-Area Square Root Calculator", *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, 2015; 13(4): 1145-1152.

[20] Yi Bo-nian, "Turbo Code Design and Implementation of High-Speed Parallel Decoder", *TELKOMNIKA*, Vol. 11, no. 4, April 2013, pp. 2116~2123

[21] M. Elharoussi, M. Belkasmi, "Design and FPGA Implementation of a Fully Parallel Architecture for Turbo decoding of majority logic codes For High data rate Applications", *International Journal of Computing and Information Technology (IJCIT)*, Vol. 3, N_2, pp. 81-95, 2011

BIOGRAPHIES OF AUTHORS

Abdelghani BOUDAOUD was born in Azilal Morocco in 1973. He is graduated from High Normal School of Technical Education (ENSET) in 1996 at Mohammedia Morocco. In 2003, he had the diploma of aggregation in electrical engineering at ENSET of Rabat Morocco. In 2011 he obtained the Master in «Laser Instrumentation and Optoelectronic Components" at the Faculty of Science and Technology (FST) Hassan I University, Settat Morocco. Currently, he is PhD student in "System Analysis and Information Technology Laboratory "(ATSI) in FST, Hassan I University, Settat, Morocco.

Mustapha ELHAROUSSI was born in Azilal Morocco in 1974; he received his PhD in Error Correcting Codes from Mohammed V University Morocco in 2013. In 2014 he joined, as Professor, applied Physics department of FST, Hassan I University, Settat, Morocco. His current research interests include hardware design of algorithms applied to information processing and control of industrial systems at System Analysis and Information Technology Laboratory at Hassan I University, Settat, Morocco.

Elhassane ABDELMOUNIM was born in Oued-Zem Morocco in 1965; he received his PhD in Spectral analysis from Limoges University France in 1994. In 1996, he joined, as Professor, applied physics department of FST, Hassan I University, Settat, Morocco. His current research interests include digital signal processing and machine learning. He is currently coordinator of a Bachelor of Science in electrical engineering and he is researcher at System Analysis and Information Technology Laboratory at Hassan I University, Settat, Morocco.