Vertical Shuffle Scheduling-Based Decoder for Joint MIMO Detection and Channel Decoding

Ali Haroun[1], Hussein Sharafeddin[2], and Ali Al-Ghouwayel[1]
[1]Computer and Communications Engineering Department, International University of Beirut, Lebanon
[2]Faculty of Sciences, Lebanese University, Lebanon

Abstract: *This paper presents a novel architecture of a soft Non-Binary Low Density Parity Check (NB-LDPC) decoder for joint iterative Multiple-Input Multiple-Output (MIMO) receivers. The proposed architecture implements a single variable node processor where the Log Likelihood Ratio (LLR) computation block is removed. It also implements a single Check Node (CN) processor that is composed of six Elementary Check Nodes. The architecture is able to decode the rate R=1/2 with frame length N= 384Low Density Parity Check (LDPC) code using a 64 QAM modulation. To our knowledge, it is the first soft decoder architecture that implements the belief propagation algorithm based on vertical shuffle schedule. Synthesis results show that the proposed architecture consumes 6.476 K slices and runs at a maximum clock frequency of 70 MHz. Taking only the decoding process part alone, 188 clock cycles are required to perform decoding iterations.*

Keywords: *MIMO, Iterative Belief Propagation (BP), Joint Factor Graph, NB-LDPC, Vertical Shuffle Schedule (VSS).*

Received October 7, 2018; accepted January 21, 2019

1. Introduction

Theoretical analysis of Multiple-Input Multiple-Output (MIMO) receivers promises large capacity gain compared to a conventional Single-Input Single-Output (SISO) [12]. This technique stands as an efficient method to increase data rates and reliability by offering higher spectral efficiency thanks to diversity and Spatial Multiplexing (SM) [3]. Associating this novel technique with Forward Error Correction (FEC) codes is necessary to achieve reliable communications. It becomes particularly appealing when the MIMO scheme is concatenated with a Low Density Parity Check (LDPC) code. LDPC codes were originally developed by Gallagher [4] and have been extensively studied during the 1990s. In 1998, Davey and Mackay [2] proposed a version of non-binary Low Density Parity Check (NB-LDPC) codes [2], which was capable of outperforming binary LDPC codes for short and medium frame lengths. These NB-LDPC codes may be conveniently combined with high-order modulation and multiple antenna schemes, which are capable of supporting high data rate transmissions [5].

Despite the benefits of applying an iterative process in MIMO receivers including an NB-LDPC decoder, the costs in terms of computational complexity and latency are still prohibitive [6].

In this paper, a new decoding schedule based on Belief Propagation (BP) is proposed in order to reduce the computational complexity and the latency of the soft NB-LDPC decoder. It offers irrefutable advantages for hardware implementations.

The proposed architecture performs the processing sequentially based on a Vertical Shuffle Schedule (VSS). The principle of this scheduling consists of processing the variable nodes one by one during each inter iteration between the Multiple-Input Multiple-Output-Belief Propagation (MIMO-BP) detector and the NB-LDPC decoder. This sequencing is different from the classic Horizontal Shuffle Schedule (HSS) sequencing, presented in [7, 8], which processes all the parity nodes then all the variable nodes, and so on iteratively. The major benefit of VSS sequencing is to promote convergence within an iterative receiver. A large part of the variable nodes can thus benefit from the updates of previously processed nodes. Another advantage of this type of sequencing is a significant decrease in the execution latency for equivalent performances [7].

The hardware design of such architecture is presented in this work. This architecture allows decoding the rate R=1/2 with frame length N= 384. It is based on Galois Field (GF) (64) using one Variable Node (VN) processor and one Check Node (CN) processor. In order to have good performance and a regular decoding architecture, the parity matrix has a degree of connectivity $d_v=2$ for each variable node and $d_c=4$ for each parity node.

The rest of this paper is organized as follows, section 2 describes the system model of the joint MIMO detection and channel decoding system, section 3gives the top view of the NB-LDPC decoder as well as the proposed architecture of the variable node processor, check node processor constituting the decoder. Timing analysis and synthesis results are

presented in section 4. The conclusion and future work are presented in section 5.

2. System Model

A MIMO wireless communication system with two transmits, *(N_t=2)*, and two receive, *(N_r=2)* antennas are considered. The proposed system is depicted in Figure 1. First, the source information is encoded by NB-LDPC codes defined over a Galois Field GF (64). Then, the encoded code words are mapped to 64 Quadrature Amplitude Modulation (QAM) symbols. Note that the same order of GF (64) has been used in the NB-LDPC code and the constellation mapper to get more flexibility into the receiver. After that, the complex symbols are spatially multiplexed onto two antennas and then transmitted over Rayleigh fading MIMO channel. On the receiver side, a MIMO detection based on the BP algorithm followed by an NB-LDPC decoder are considered to compute soft information for all symbols. In order to improve performance, extrinsic information can be exchanged between the detector and the decoder iteratively.

A Joint Factor Graph (JFG) representation of such receiver is given in Figure 2. It consists of two disjoint parts. The upper part is the factor graph representation of the MIMO-BP detector. The lower part corresponds to the factor graph representation of the NB-LDPC decoder. Each couple of received symbol nodes y_{jj} is connected to a couple of candidate symbol node ss_{ii} via *2×2* spatial multiplexing, where $ii \in [1, N_t=2]$ and $jj \in [1, N_r=2]$. At the other end, the ss_{ii} nodes are connected to their associated variable nodes V_{ii} of the NB-LDPC graph.

The connections between the VN and the CN depend on the parity matrix of the NB-LDPC code. In this study, a VSS is applied over the JFG. Indeed this schedule enables a fast iterative process convergence [7]. The main advantage of this type of schedule is its low decoding latency and low Word Error Rate (WER) performances when compared to a HSS [9, 10, 11]. Next, we present the hardware design of such architecture.

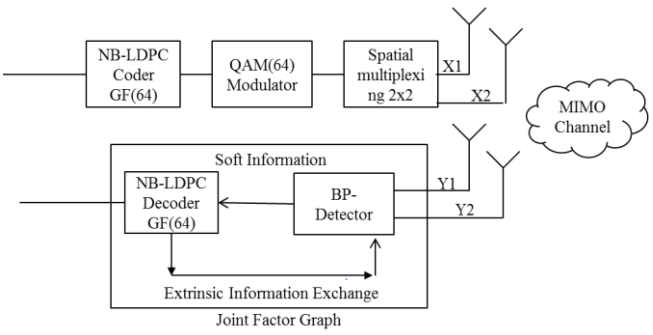

Figure 1. Block diagram of a joint MIMO detection and channel decoding system.

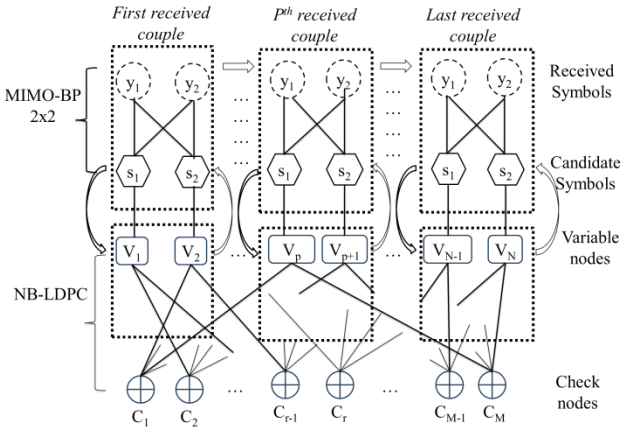

Figure 2. JFG representation of the joint MIMO detection and channel decoding system.

3. Proposed Decoder Architecture

3.1. Global Decoder Architecture

In this section, we briefly present the different elements constituting the decoder; the details will be presented later. Figure 3 shows the external view of the decoder architecture which is composed of one VN and one CN processor respectively. A memorization system consisting of four RAM blocks is implemented to store the extrinsic Log Likelihood Ratio (LLR) obtained during the exchange information between different processors. Two sets of GF multipliers are implemented to perform the GF multiplication and division at the input and the output of the CN respectively.

Initially, the decoder receives the LLR values associated to the variable nodes from the detector. They are stored in the main memory (*Main RAM*) inside the VN processor for future updates. Such processor updates these messages by the incoming LLR values from $RAMC_itoV$, where $i \in [1, d_v=2]$, associated with the connecting CN during the previous iteration. Then, the updated values will be sent to $RAMVtoC_j$ where $j \in [1, d_v=2]$.

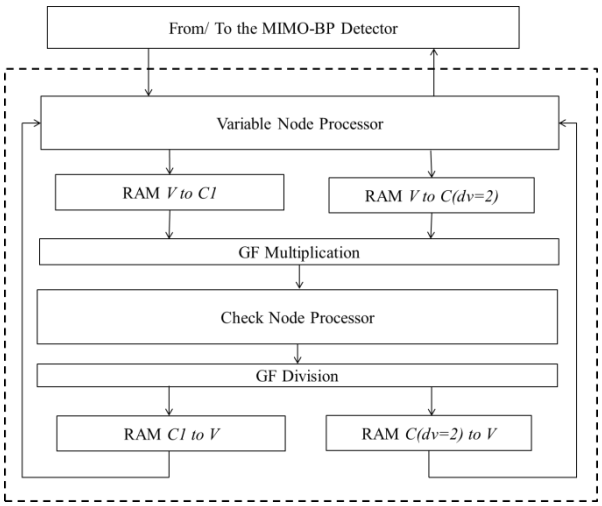

Figure 3. External view of the VSS NB-LDPC Decoder.

Before entering the check node, the LLR values, stored in *RAMVtoC$_j$* are multiplied by non-binary GF symbols. After being processed in the CN, the updated messages are divided by a GF divisor and then stored in *RAMC$_i$toV*. Afterward, the stored messages are re-sent to the corresponding variable node. Finally and after the VN update of all data coming from the CN, the updated LLR values will be transferred to the detector to start a new iteration.

3.2. Decoder Global State Machine

The different machine states handling the decoder execution is described in this section. They depict the different stages a decoder goes through. Note that all states are controlled by a global counter that sets the time of each stage.

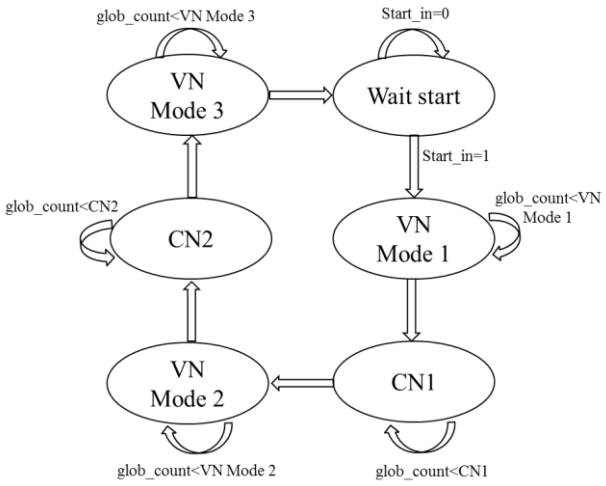

Figure 4. State machine of the decoder.

Figure 4 illustrates the diagram block of the state machine where the six decoder states are shown:

1. *Wait start*: this is the initial state of the decoder where all the signals are initialized. It is controlled by the signal start_in. The decoder waits for the LLR symbols generated by the MIMO-BP detector to start the update of the variable nodes.
2. *VN modes*: as soon as the LLR symbols generated by the detector have arrived, the decoder starts to update the variable nodes by going through different update modes:
   - *VN Mode 1*: during this mode, the values coming from the detector should be updated by the LLR values coming from the first check nodeCN_1. This mode will be followed by updating CN_1.
   - *VN Mode 2*: the values coming from the second CN_2have to be updated by the LLR values stored in the main memory. The update of CN_2 will be the next state that follows this mode.
   - *VN Mode 3*: when the decoding phase is finished, the coming values from the two check nodes, CN_1 and CN_2, should be added and sent to the main memory.
3. *CN modes*: in these states, CN_1 and CN_2 are updated after finishing VN Mode 1 and VN Mode 2 respectively.

After completing *VN Mode 3*, the variable nodes shall send the updated LLR values to the detector to begin a new iteration.

3.3. Variable Node Processor Architecture

The architecture of the VN processor is designed in a way to meet the requirements of the received data from the detector. Figure 5 shows such architecture based on main memory, an adder, several multiplexers, de-multiplexer and a sorter. Two types of memories should be considered in this work. The first is the one that only stores the sixty-four LLR values coming from the detector named Main RAM without the need to store the GF indexes. It is composed of sixty-four cases storing LLR values in ascending order according to their GF indexes from $\alpha^0$to α^{63}.The second type of memory is designed to store the twelve LLR values exchanged within the decoder along with their GF indexes. In order to reduce its complexity as proposed, by Boutillon [1], the number of exchanged messages within the decoder is limited to twelve.

There are three modes of operation for the VN processor architecture. The first mode starts after receiving the LLR values from the detector and storing them in the main memory (Main RAM) through *MUX$_1$*. It consists of updating the values coming from the detector by the LLR values coming from the first check node *RAM C$_1$ to V*. Through *MUX$_3$*, the values of this RAM are added according to their GF indexes and then updated by the ones stored in the main memory (through *MUX$_2$*). The updated values should be sorted and sent to *RAM V to C$_2$* via *DeMUX$_1$*.

The second mode consists of updating the values coming from the second CN, *RAM C$_2$ to V*, connected to the variable node processor by the LLR values stored in the main memory through *MUX$_3$* and *MUX$_2$* respectively. The updated values should be sorted and sent to *RAM V to C$_1$* via *DeMUX$_1$*. The last mode is when the decoding phase is finished. The coming values from the two check nodes, *RAM C$_1$ to V* and *RAM C$_2$ to V*, connected to the VN processor should be added and sent to the main memory via *MUX$_1$*. The latter shall send them to the detector to begin a new iteration.

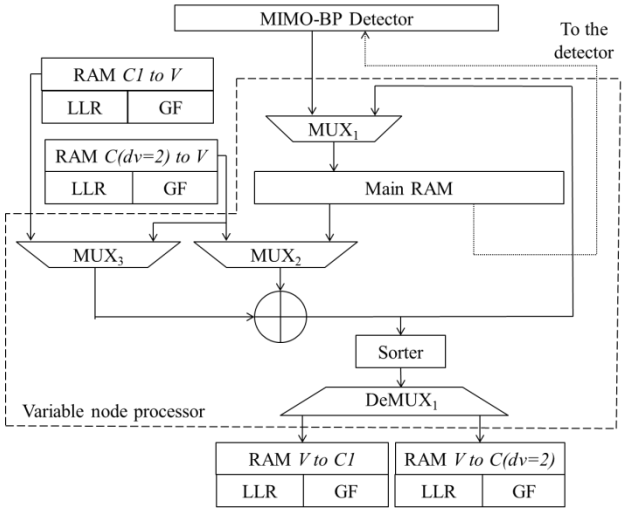

Figure 5. Variable node processor of the NB-LDPC decoder.

The VN update consists in adding the LLR values coming from MUX_2 and MUX_3 using an arithmetic adder. The adder receives ordered lists of messages but generates un-ordered one. Thus, a sorter is needed to order the messages in descending order of LLR values. The sorter architecture implemented in the proposed VN architecture is based on a transcoder circuit connecting the bank of 12 comparators to the 12 registers R_m, $m=0, \ldots, 11$ as shown in Figure 6. Each comparator $Comp_m$ receives two inputs:

1. The new value entering through the input bus *Data_in*.
2. The value stored in its corresponding register R_m.

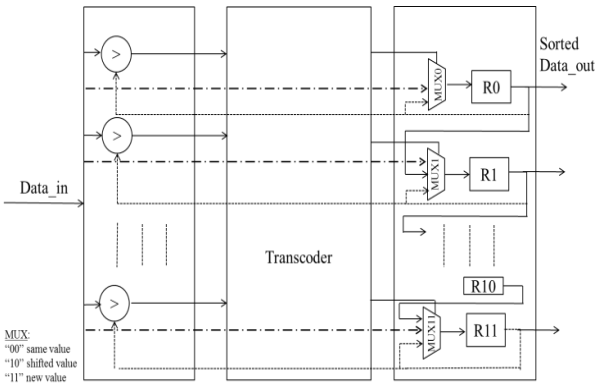

Figure 6. Architecture of the sorter circuit.

If the new value is greater than the value stored in R_m, so the comparator $Comp_m$ generates a signal l_m=1, otherwise l_m=0. Let L denote the vector containing the signals l_m, $m=0, 1, \ldots 11$, i.e., $L=[l_0 l_1 \ \ldots \ l_{11}]$. Each R_m receives its input through a MUX_m and selects one of the three inputs according to its control ll_m generated based on L: (1) *Data_in* if ll_m=11; (2) the content of its neighbour register R_{m-1} if ll_m=10; (3) its current stored value, otherwise. For example, if *Data_in* is greater than R_m for$m>$ 2, and *Data_in*$<R_m$, $m=0, 1, 2$, thus L=[000111111111],R_3 is replaced by *Data_in*, R_m is replaced by R_{m-1} for all m=4, 5, …, 12. R_m, $m=0,1,2$ remain unchanged.

Then, the role of the transcoder is to generate all the control signals ll_m based on the value of L at each clock cycle, where each new input entering the sorter is compared to all R_m, $m=0, 1, \ldots, 12$. In the previous example, the MUX control signals are generated as follows: ll_0=00, ll_1=00,ll_2=00, ll_3=11, and ll_m, $m=4, \ldots 11$=10. The content of the registers is updated accordingly.

This process continues until all the inputs to be sorted have entered the sorter. At the last clock cycle, the registers R_m contains the 12 symbols sorted in descending order.

The overall complexity of this sorter is 12 comparators, 12 shift registers and 12 multiplexers MUX-3-to-1, plus some combinational logic to implement the transcoder. The critical path is the time for one comparator and one MUX-3-to-1.

3.4. Check Node Processor Architecture

The realization of a CN processor with degree of connectivity $d_c = 4$ requires the implementation of six Elementary Check Nodes (ECN) processors organized in three levels where each level implements two ECNs. Note that the internal architecture of an ECN was described in [1].

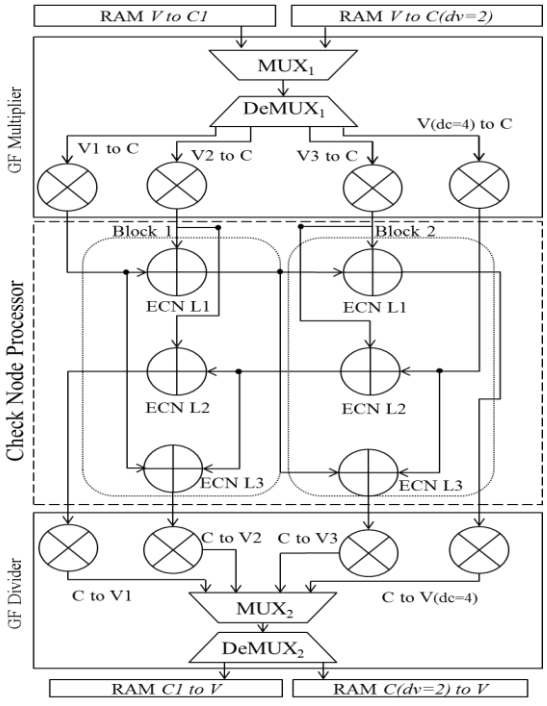

Figure 7. Check Node processor of the NB-LDPC decoder.

Figure 7 shows the architecture of a CN processor that works as follows: From *RAM V to* C_1 or *RAM V to* C_2, the messages V_i *to C*, *i= {0,...,4}* is fed serially. The role of MUX_1 is to choose the information source between *RAM V to* C_1 and *RAM V to* C_2. Similarly, the role of $DeMUX_1$ is to choose the destination among the different V_i *to C*. Before entering the CN processor, the coming messages are multiplied by the GF coefficients of the parity check matrix using four

GF multipliers. Therefore, the two blocks receive the coming messages and the ECN of each block starts its computation when the two input messages are available. Let us consider the execution of ECN L1 of Block1.

This ECN receives serially the messages V_1*to C* and V_2 *to C* and starts the computation immediately. After two clock cycles, ECN L1 generates the first couple of (LLR, GF) that will be sent to the ECN L1 of Block 2. After receiving the message V_3*toC*, ECN L1 of Block 2 will be triggered by the left input connected to the right output coming from the ECN L1 of the Block1. This is the forward step of the well-known Forward-Backward method. Similarly, in the opposite direction, the backward step starts with the computation of the ECN L2 of the Block 2 and the message *C to* V_1 is generated after a latency of four clock cycles. As for the computation of the ECN L3, these ECN start their computation with the arrival of the most delayed input. After processing has completed inside the CN processor, the outgoing messages are divided by the GF coefficients and then sent to *RAM* C_1 *to V* or *RAM* C_2 *to V* through $MUX_2$and $DeMUX_2$.

4. Timing Diagram and Synthesis Results

In this section, the timing diagram and the synthesis results of the proposed architecture are presented. We modelled our proposed architecture in Very high speed integrated circuit Hardware Description Language (VHDL). Model SIM and ISE-XILINX were the tools of choice for simulation and synthesis.

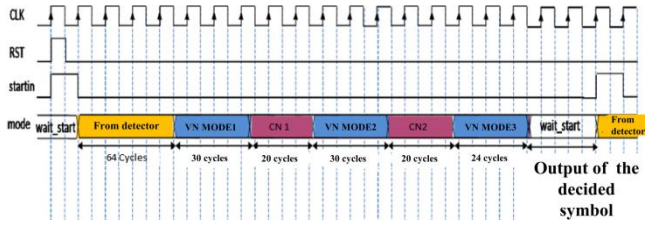

Figure 8. Overall timing diagram of the proposed architecture.

Figure 8 shows the timing diagram of the VSS proposed architecture to process one symbol. The first stage starts when the data symbols are loaded from the MIMO-BP detector; it takes 64 cycles. During this stage, the data is filled into the main memory (*Main RAM*) of the VN processor. After that, the VN processor starts working in three modes. The first is when it updates the values coming from the detector by the LLR values coming from the first *CN RAM* C_1 *to V*; this stage takes 30 cycles. After that, the CN processor starts processing the incoming values from the VN processor which takes 20 cycles before returning the updated messages. The VN processor then starts the second mode by updating the values coming from the detector by the LLR values from the second check node *RAM* C_2 *to V*. This stage takes also 30 cycles. Once more, the CN processor starts processing the incoming values from the VN processor which takes also 20 cycles before sending the updated messages to VN.

Finally, the VN processor starts the third and last mode when the coming values from the two CNs, *RAM* C_1 *to V* and *RAM* C_2 *to V*, are added and sent to the main memory for sending them later to the detector to begin a new iteration. This stage takes 24 cycles. The overall timing of processing one symbol thus takes 188 clock cycles.

Table I shows the frequency and the synthesis results of our proposed VSS decoder compared to the DAVINCI decoder [1]. Note that the synthesis study is conducted on the FPGA device Virtex 4, XC4VLX200 using ISE-XILINX. By analysing this table we notice that the proposed architecture has a better frequency with fewer resources compared to the DAVINCI decoder. It can reach up to 70 MHz clock rate vs. 58 MHz and consumes 6476 slices vs. 10025 slices for the DAVINCI decoder.

Table 1. Synthesis results of the proposed architecture on Virtex 4, XC4VLX200 device.

	Proposed Architecture	**DaVinci decoder in [1]**
VN Processor	776 Slices	1175 Slices
	98 MHz	78 MHz
CN Processor	5700 Slices	8550 Slices
	69 MHz	69 MHz
Total	6476 Slices	10025 Slices
	70 MHz	58 MHz

5. Conclusions

In this paper, a novel architecture based on VSS schedule is presented. The proposed architecture describes a single VN processor connected to a single CN processor. The architectural exploration is driven by the limitation of the hardware complexity. A new schedule of the NB-LDPC decoder has been considered to decrease the latency and the computational complexity without impacting the error rate performance. Based on the results of the timing diagram and the synthesis analysis, we can conclude that our proposed architecture consumes 35% less resources with increasing 21% of frequency compared with the state-of-the-art DAVINCI decoder. Besides, a maximum operating frequency of 70 MHz can be reached on the target device. To the best of our knowledge, this is the first VSS architecture dedicated to a NB-LDPC decoder. These optimistic results are a first step towards an efficient hardware implementation of optimized decoder architecture.

References

[1] Boutillon E., Conde-Canencia L., and Al-Ghouwayel A., "Design of a GF(64)-LDPC Decoder Based on the EMS Algorithm," *IEEE Transactions on Circuits and Systems I: Regular papers*, vol. 60, no. 10, pp. 2644-2656, 2013.

[2] Davey M. and Mackay D., "Low Density Parity

Check Codes over GF (q)," *IEEE Communication letters*, vol. 2, no. 6, pp. 165-167, 1998.
[3] Foschini G., and Gans M., "On Limits of Wireless Communications in a Fading Environment when Using Multiple Antennas," *Wireless Personal Communication*, vol. 6, no. 6, pp. 311-335, 1998.
[4] Gallager R., "Low Density Parity Check Codes," *IRE Transaction on Information Theory*, vol. 8, no. 1, pp. 21-28, 1962.
[5] Hanzo L., Liew T., and Yeap B., *Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Fading Channels*, John Wiley and Sons, 2002.
[6] Haroun A., AbdelNour C., Arzel M. and Jego C., "Low Complexity Soft Detection of QAM Demapper for aMIMO System," *IEEE Communication letters*, vol. 20, no. 4, pp. 732-735, 2016.
[7] Haroun A., AbdelNour C., Arzel M., and Jego C., "Symbol-based BP Detection for MIMO Systems associated with Non-Binary LDPC Codes*," in Proceedings of Wireless Communication Network Conference*, Istanbul, pp. 212-217, 2014.
[8] Haroun A., AbdelNour C., Arzel M., and Jego C., "Low-Complexity LDPC-coded Iterative MIMO Receiver Based on Belief Propagation Algorithm for Detection," *in Proceedings of International Symposium on Turbo Codes and Iterative Information Process*, Bremen, pp. 213-217, 2014.
[9] Haroun A., AbdelNour C., Arzel M. and Jego C., "Low-Complexity Layered BP-based Detection and Decoding for a NB-LDPC Coded MIMO System," *in Proceedings of International Conference on Communication*, Sydney, pp. 5107-5112, 2014.
[10] Haroun A., Nasr R.,and Al-Ghouwayel A., "On the Implementation of Vertical Shuffle Scheduling Decoder for Joint MIMO Detection and Channel Decoding System*," in Proceedings of International Arab Conference on Information Technology*, Werdanye, 2018.
[11] Haroun A., Rashid A., and Haidar J., "Implementation of a Belief Propagation Detector for an Iterative MIMO Receiver," *in Proceedings of International Arab Conference on Information Technology*, Werdanye, 2018.
[12] Telatar E., "Capacity of Multi-Antenna Gaussian Channels," *Transactions on Emerging Telecommunications Technologies*, vol. 10, no. 6, pp. 585-595, 1999.

Ali Haroun received his Bachelor's degree in Physics-Electronics in 2008 from the Lebanese University, Beirut, and his M.S. Degree in Signal Telecom Image Speech in 2010 from the Doctoral School of Sciences and Technology (Lebanese University), Beirut. He received his PhD in Sciences and Technology of Information and Communications in 2014 from Telecom Bretagne (Institute Mines Telecom), Brest, France. Then, he was a post-doctoral fellow at INP, Bordeaux, France. His research activities concernedChannel detection of MIMO systems, Channel Coding /Decoding of NB-LDPC and Analog to digital converter. He is a member of reviewing boards of a number of IEEE journals and Conferences. In 2017, he joined the International University of Beirut-BIU, Beirut, where he is currently an Associate Professor in the Computer and Communication Engineering Department.

Hussein Sharafeddin received his B.E. degree in computer and communications engineering in 1990 from the American University of Beirut. He received his M.S. degree in computer engineering from the Ohio State University, Columbus in 1995. Dr.Sharafeddinspent more than 10 years working for IBM in New York. He received his PhD degree in computer and systems engineering in 2009 together with a master's in applied mathematics from RPI, Troy, NY. His research focused on 3D image analysis. Dr.Sharafeddin is currently with the Faculty of Sciences at the Lebanese University, Lebanon.

Ali Al Ghouwayel received his B.E. degree in Physics-Electronics in 2002 from the Lebanese University, Beirut, his M.S. Degree in Telecommunication in 2004 from the National School of Engineering of Brest, France, and his PhD in Signal Processing and Telecommunications in 2008 from the High School of Electricity and University of Rennes 1, Rennes, France. In 2008, He joined the Lab-STICC laboratory in Lorient, France and worked as a Post-Doctoral researcher on the European Project DAVINCI. In 2010, He joined the Lebanese International University, Beirut, where he is currently an Associate Professor in the Computer and Communication Engineering Department. His current research interests include Study and Optimization Algorithm-Architecture for Hardware Implementation of Non-Binary LDPC decoders, Design of Reconfigurable Ultra-Throughput RateArchitectures for 5G Systems, and ultra-low power transmission techniques for future IoT system.