



## Fault Tolerance Analysis of Communication System Interleavers: the 802.11a Case Study

P. REYES, P. REVIRIEGO, J. A. MAESTRO AND O. RUANO  
*Universidad Antonio de Nebrija, Madrid, Spain*

*Received: 23 September 2007; Accepted: 18 October 2007*

**Abstract.** The study of Multiple Soft errors on memory modules caused by radiation effects represents an interesting field of current research. The fault tolerance of these devices in radiation environments is traditionally analyzed and increased by means of soft error protection mechanisms as EDAC codes or physical interleaving. As Communication System interleavers are mainly implemented using memories, a similar protection against soft errors to the one used for memory devices could be performed, as a conventional solution, when they are used in critical missions. In this paper, the knowledge of the system is used to apply the communication interleaving pattern as physical interleaving employing the inherent redundancy (added by previous modules of the Communication System) of the data processed by the interleaver as an error correction mechanism. Therefore a similar protection to the conventional solutions is obtained but with a reduced cost.

**Keywords:** single event upsets (SEUs), multiple bit upsets (MBUs), redundancy, fault tolerance, interleaving, soft errors

### 1. Introduction

Soft errors are one of the main concerns when studying the reliability of digital systems as they cause unforeseeable errors in the behavior of the circuit, in the sense that they are induced by transients faults generated by the environment where the system operates (for example a high radiation one), temporary malfunctions of circuit voltage wires, noise in the digital circuit or manufacturing problems that make some nodes more sensitive [1, 2]. As radiation is one of the most significant sources of soft errors in microelectronic circuits and the space environment has higher radiation densities than the terrestrial one, the design of circuits tolerant to space environment hazard effects represents a challenge to the microelectronic industry, and several error mitigation techniques at different levels of design or fabrication process could be performed.

The space radiation particles impact over electronics materials and they ionize atoms through which they propagate [2, 3]. This situation can produce several types of errors on the system operation. Single event effects (SEEs) are the soft errors caused by radiation on microelectronics circuits, and they can be classified in single event transients (SETs) when the ionizing particle flips the value of a logic gate and single event upsets (SEUs) if the value of a storage cell (memory cell, flip flop or latch) is inverted. If the radiation particle induces multiple SEUs or flips several stored values at the same time it is known as multiple bit upset (MBU). Although the probability that several memory cells (very sensitive nodes) are flipped by single events is low, as we move to higher device densities, the probability of MBUs occurrence increases since memory cells are closer to each other and they can be flipped by one single ionizing particle [4, 5]. Therefore, these effects

may cause errors in the system operation that exceed the failure rate specification (the frequency with which a system or component fails) in various application domains, as the space field. In such cases, error mitigation techniques should be added to memories (to deal with SEUs and MBUs), and eventually to combinational logic.

Error detection and correction codes (EDAC codes) that add several redundancy or parity bits to the protected words are the traditional solution used to reduce the number of faults in memory devices. Usually, and because the area, speed and power penalties of this error mitigation technique, the EDAC codes used have single error correction and double error detection capabilities (SEC-DED codes), that means that they can correct only single bit-flips on each protected or coded word (the correction process is performed by the decoder in the memory read operation). In such cases, when memories have to be protected against MBUs a conventional solution consists in using a SEC-DED code (as Hamming one) and a memory with physical interleaving, that adds physical distance between the bits that belong to the same protected word. If this physical distance is higher than the MBU size, the induced bit-flips will be corrected as a set of isolated SEUs on different coded words [5, 6].

If the words of the memory protected with SEC-DED EDAC codes and physical interleaving are not

frequently read, it may happen that new bit flips occur on erroneous words that have not been corrected yet, so several errors could be accumulated in the same protected word with the consequence that the EDAC decoder will not have enough capacity to correct these multiple errors. In this case, additional protection techniques as scrubbing can be included. This protection mechanism consists in reading all the memory words periodically in order to prevent the accumulation of multiple errors in the same coded word [1, 5].

In space applications, communication systems are needed to share information between satellites and terrestrial stations [7] and they can comprise different modules implemented with memories, like for example communication interleavers. These modules will be sensitive nodes to radiation effects and their fault tolerance should be analyzed and, in case of need, enhanced using some of the error mitigation techniques commented, as EDAC codes or physical interleaving. A block diagram of a typical radio communication system (based on the OFDM modulation) is shown in Fig. 1. Other systems will have a similar functionality and therefore the example presented illustrates a generic communication system.

One of the blocks in Fig. 1, commonly implemented using memory modules and known as block interleaver [8], is the interleaver whose function is to ensure that the bits that come from the convolutional

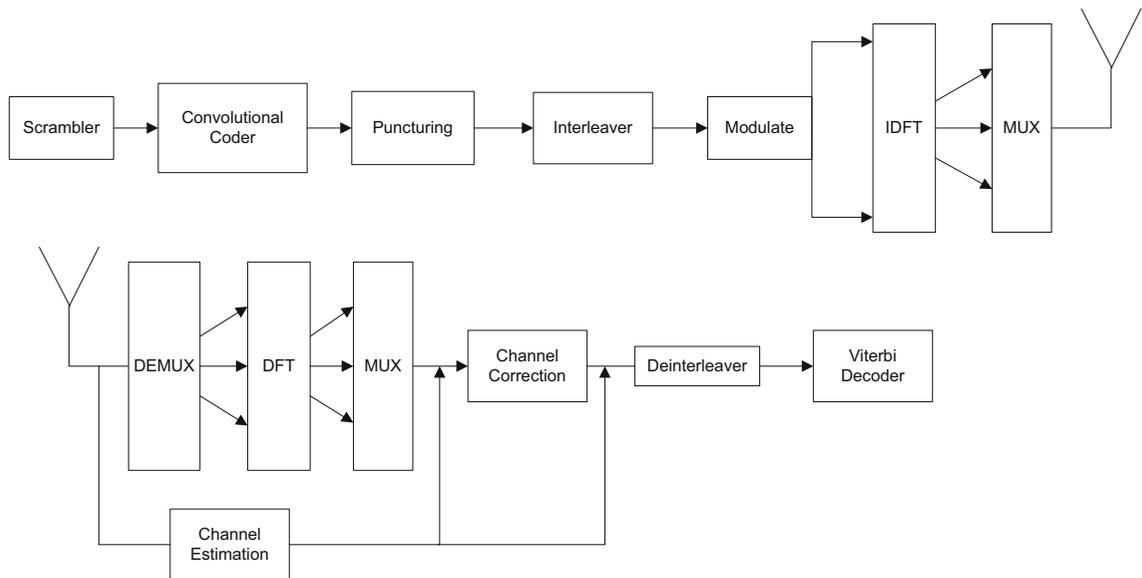


Figure 1. OFDM communication system block diagram.

encoder are interleaved such that consecutive bits at its input are separated at its output, following a specific interleaver pattern. This is done to protect against large bursts of errors on the transmission process as those will be converted by the deinterleaver (in the receiver) into a number of isolated errors. Taking into account that the codes used in most communications systems to correct errors are good at correcting isolated errors but they have a limited ability to deal with bursts of errors. If this communication or system interleaving pattern used to transmit logical contiguous bits out of order is applied in the write operation of the interleaver (a memory block), the bits that come from the convolutional encoder will not be stored sequentially in the memory, so a type of physical interleaving will be applied to the coded data, stored in the block interleaver by means of the communication interleaving pattern. Moreover, as the data interleaved has a certain information redundancy (as a result of the convolutional encoding) we are using the system knowledge ([9] to [12]) in the sense that the characteristics of the convolutional encoder and interleaver modules of the communication system are employed to obtain a protection mechanism against MBUs similar to the combination of physical interleaving with EDAC codes mentioned before.

As a common implementation for Communication Interleavers is based on memories and these circuits are very sensitive to MBUs occurrence, an analysis of the fault tolerance capability of communication systems (by means of several fault injection campaigns) considering different implementations of the block interleaver is performed in this paper, structured as follows. First, a related work section is presented in order to put in perspective other works based on system knowledge and communication systems testing. A general analysis of several block Interleaver implementations in terms of effectiveness against MBUs is performed in section 3. In section IV the 802.11a communication system is used as a case study, on which simulations are done to illustrate the performance of the different implementations described in section 3. Finally, conclusions of the work and future lines of research are presented.

## 2. Related Work

The specific knowledge of a circuit or application has been used to protect different designs. In the

following some examples covering signal processing systems and processors are given.

Several fault tolerant general purpose processors based on the system knowledge can be seen in [13, 14], where ad-hoc fault tolerant implementations for IBM z990 servers and 8051 microcontroller, respectively, are presented, combining the use of parity bits, EDAC codes and triple modular redundancy (TMR) for control logic, caches and main memories. An implementation study of the fault tolerant Leon-3 processor based on similar error correcting mechanisms is illustrated in [15].

Furthermore, fault tolerant signal processing systems can be designed using the concept of system-knowledge. References [16, 17] propose system knowledge based solutions for the fast Fourier transform (FFT), using different properties of the discrete Fourier transform (DFT) for the concurrent error detection (while the FFT circuit calculates the output), as Parseval's theorem.

The inherent fault tolerance of the signal representation domain is used as system knowledge too, references [18, 19] present a signal representation in sigma-delta domain to enhance the fault tolerance against soft errors of several digital signal processing circuits as FIR filters.

Moreover, in [9] to [11] fault tolerant finite impulse response (FIR) Filters implementations using the knowledge of the system are proposed and compared (in terms of effectiveness against soft errors and area overhead) with the equivalent protected implementations using generic error mitigation techniques, as TMR and Hamming codes. The same authors propose a system knowledge based fault tolerant implementation for adaptive filters like echo cancellers in [12], where the system knowledge consists in using the inherent adaptation logic of the circuit to better correct the effects of soft errors on different structures of the adaptive filter.

In this context, a radiation-hardened high-speed serial data bus for satellite onboard communication is proposed in [20]. In [21], because of the wide use of signal equalizers in satellite communications and the space environment restrictions as low power dissipation, that creates the need of using adaptation algorithms with reduced computational cost, different non-linear functions used for the equalization process of Bussgang algorithms are compared in fault tolerant effectiveness. Note that a similar study of the fault tolerance of different communication interleaver

implementations, considering the whole system where it operates, is presented in the rest of this paper.

### 3. Fault Tolerance Analysis of Different Interleaver Implementations

The inherent redundancy of the bits stored in the communication interleaver, as they are coded with an error correction code prior to interleaving (see Fig. 1), is used as system knowledge to choose the best fault tolerant block interleaver implementation when dealing with MBUs on this module. The idea behind the analysis performed is that the information that comes from the convolutional encoder has enough redundancy to correct, with high probability, isolated errors while its capacity to correct bursts of errors is worse. So the main idea consists in taking advantage of introducing physical distance between contiguous logical bits susceptible to be flipped (bits that have information redundancy), that means to use the interleaving process as a fault tolerant mechanism, either the communication interleaving itself or the addition of physical interleaving. Therefore, the three following points (and its combinations, shown in Table 1) will be analyzed:

- Communication interleaving pattern applied in write or in read memory operations
- Block interleaver memory with physical interleaving
- Extra addition of information redundancy (EDAC codes) to the data that comes from the convolutional encoder stored in the block interleaver

Next, a description of the three possibilities and their combinations is presented.

#### 3.1. Communication Interleaving in Read/Write Memory Operation

In order to transmit the logical data separated in time or frequency, the communication interleaving pattern could be applied before writing the block interleaver memory from where later the data is read sequentially (in order to be transmitted), or the data could be written sequentially and read using the communication interleaving pattern. This decision is not irrelevant and it affects to the fault tolerance of the system, as it will be seen in the following.

Figure 2 tries to illustrate the differences between the two commented options, where signal  $\text{int\_w/int\_r}$  controls if the interleaving pattern is performed in read or in write memory operations [22].

The next example tries to illustrate the differences (in fault tolerance against MBUs) between using the communication interleaving pattern in write or in read operations of the memory module. Let us consider a system that transmits 18 bits and where the system interleaving process performs the following operation:

For the sequence of bits: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

The communication interleaving pattern is: 0 3 6 9 12 15 1 4 7 10 13 16 2 5 8 11 14 17

Where the numbers indicate the correct order of the bits in the sequence (the logical order) and the position in the sequence represents the transmission order. Therefore, after the interleaving process, the first transmitted bit will be the first logical bit (bit 0) but the second transmitted bit will be the fourth logical one (bit 3)...and so on.

If the memory module is read by rows, its width is 3 bits and the communication interleaving is imple-

Table 1. Interleaver implementations considered for fault tolerance optimization.

Physical interleaving	Write/read	Use/No	Case
No physical interleaving	Communication interleaving process in write memory operation	No use of EDAC codes	Case 1
		Use of EDAC codes	Case 2
	Communication interleaving process in read memory operation	No use of EDAC codes	Case 3
		Use of EDAC code	Case 4
Physical interleaving	Communication interleaving process in write memory operation	No use of EDAC codes	Case 5
		Use of EDAC codes	Case 6
	Communication interleaving process in read memory operation	No use of EDAC codes	Case 7
		Use of EDAC codes	Case 8

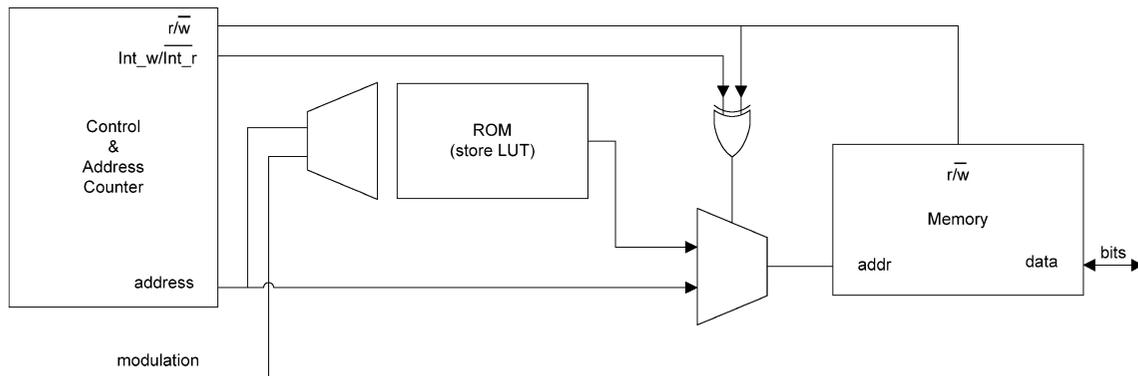


Figure 2. Block interleaver pattern in read or write memory operations.

mented in the memory write operation, the block interleaver will have the bits stored in the following order (see Fig. 3):

However, if the bits are written in the memory in sequence and read using the communication interleaving pattern the interleaver memory content after the write operation would be:

It can be seen in Figs. 3 and 4, in dotted lines, that the logical distances between contiguous physical bits are higher in the case of applying the interleaving pattern in the memory write operation (this difference is more visible when using real interleaving patterns, with higher interleaver depths or block interleaver sizes). Apart from the fixed parameters that have influence on the correction capacity of the convolutional code used (as convolutional encoder memory), it also depends on the logical distance of the bits affected by simultaneous faults (better with higher logical distance). Therefore, if an MBU affects to several information bits stored in the block interleaver, it would be more probable to overcome from MBUs that flip bits more separated, in logical terms.

Usually, the lower the number of inversions of the MBU is the more probable the MBU happens. That means that two simultaneous bit-flips are more probable than 3-bit MBUs. Besides, an event that causes MBUs on a memory module flips the stored value of adjacent memory cells [23] (vertically or/and horizontally). Therefore, it could be analyzed that the fault tolerance of the system will not be the same using the different block interleaver implementations shown in Figs. 3 and 4. In this sense, when one MBU occurs on the implementation of Fig. 3 (interleaving pattern applied in the memory write operation) if it occurs on the two first positions of the fourth row

(see Fig. 3, with bold letter and solid circles), bit 10 and bit 13 will be affected and it will be more probable that the Viterbi decoder (used at the receiver part as a decoder for coded bits with a convolutional encoder, see Fig. 1) may correct the errors using this implementation better than using the other where the communication interleaving pattern is performed in the memory read operation (see Fig. 4, solid circles at same position). All of this thanks to the memory of the used convolutional encoder and the bigger logical distance between contiguous physical bits of the implementation of Fig. 3. It should be seen that the commented case (horizontal MBU) is the worse case that can appear when a 2-bit MBU occurs on the interleaver, for the specific pattern shown in Figs. 3 and 4.

0	3	6
9	12	15
1	4	7
<b>10</b>	<b>13</b>	16
2	5	8
11	14	17

Figure 3. Bits stored in the interleaver, pattern applied in write operation.

0	1	2
3	4	5
6	7	8
9	10	11
12	13	14
15	16	17

Figure 4. Bits stored in the interleaver, pattern applied in read operation.

It must be noticed that, in general, if all the possible MBU patterns [23] are equally probable and bigger block interleavers are considered, the configuration shown in Fig. 3 is more fault tolerant against multiple simultaneous soft errors than the one shown in Fig. 4. In conclusion, applying the communication interleaver pattern in the write memory operation would be better than applying it in the read one. But in this case, the combination of the block interleaver word size and the interleaving pattern itself influences in the fault tolerance results against MBUs (observe the case of block interleaver of 6-bit words) so it should be studied.

Furthermore, in any of the commented cases the fault tolerance will be worse if the used puncturing rate is larger in order to reach higher communication rates, because in this situation not all the outputs bits of the convolutional encoder are transmitted but some of them are eliminated, and the ability to correct errors is lower.

Therefore, when no additional error mitigation techniques are included in the design (as physical interleaving or EDAC codes), it is better to perform the interleaving pattern in the memory write operation (Table 1: case 1) than in the read one (Table 1: case 3).

### 3.2. Memory with Physical Interleaving

If a memory module with physical interleaving is used for the implementation of the communication interleaver the combination of the physical interleav-

ing with the two possible implementations commented in A) should be analyzed (Table 1 cases 5, 6, 7 and 8).

Memories with physical interleaving are commonly used (in combination with some type of SEC-DED EDAC code) to correct MBUs, and it consists in introducing physical distance between all the bits that belong to the same word. So, the main difference between physical and communication interleavers is that while physical interleavers are used to protect the content of the memory against MBUs, the system interleavers are used to protect the data to transmit from large error bursts on the transmission channel using the same idea: store or transmit (respectively) the bits that belong to the same word in a separated manner. In this sense, the communication interleaving pattern has been used in A) as physical interleaving one (case 1) to protect the data stored in the block interleaver against multiple soft errors. In this subsection the combination or not of the two interleaving patterns will be analyzed (cases 5 and 7).

Considering 3-bit words and a memory with physical interleaving patterns of 1 of 3 and 9 bits by row, if we try to write six words (A, B, C, D, E and F) the content of the memory would be the illustrated in Fig. 5.

Words:

$$A : A_{11}A_{12}A_{13}, B : B_{11}B_{12}B_{13}, C : C_{11}C_{12}C_{13},$$

Where:  $A_{11}A_{12}A_{13}$  are the 3 bits of the word  $A$ , for example, and each bit will be written at first row, columns 1, 4 and 7, respectively.

Different rows store different coded words and in the same row, the bits that belong to the same coded word are separated by two columns (physical interleaving pattern of 1 of 3).

If no EDAC codes are used cases 5 and 7 of Table 1 should be analyzed. In case 5, the combination of the two interleaving patterns (communication and physical) is shown on the bits stored in the block interleaver. In case 7, the bits are stored in the block

$A_{11}$	$B_{11}$	$C_{11}$	$A_{12}$	$B_{12}$	$C_{12}$	$A_{13}$	$B_{13}$	$C_{13}$
$D_{11}$	$E_{11}$	$F_{11}$	$D_{12}$	$E_{12}$	$F_{12}$	$D_{13}$	$E_{13}$	$F_{13}$
1	2	3	4	5	6	7	8	9

Figure 5. Physical interleaving.

0	9	1	3	12	4	6	15	7
10	2	11	13	5	14	16	8	17

Figure 6. Physical interleaving and communication interleaving pattern in memory write operation. Bits stored in the interleaver (minimal logical distance of 1).

interleaver by means of the physical interleaving and the communication interleaving is applied in the memory read operation. For the same example used in A) and the physical interleaving pattern shown in Fig. 5 the two commented cases (5 and 7) would be the illustrated in Figs. 6 and 7.

The next considerations can be observed in Figs. 6 and 7. The first one is that there are no conclusions about if the combination of the two interleaving patterns (case 5) enhances or degrades the fault tolerance of the system, because (if no EDAC codes are used) the logical distance between contiguous physical bits (more susceptible to be flipped [23]) depends on the individual characteristics of each interleaving pattern (the logical distance between contiguous bits on the same row is different for the example of case 5 shown in Fig. 6). Moreover, if no combination of the interleaving patterns is performed (case 7), that means that the communication interleaving has to be applied in the memory read operation (see Fig. 7), the minimal logical distance between contiguous bits is always given by the physical interleaving pattern used (for the example with physical interleaving of 1 of 3, the lowest logical distance is 3 as it can be seen in Fig. 7).

In conclusion, in the case of using a memory with physical interleaving and no EDAC codes (cases 5 and 7), applying the communication interleaving in the memory read operation (case 7, where interleaving patterns are not combined in the write operation of the block interleaver) would be a good option, because we do not need to analyze, specifically, any combination of patterns.

### 3.3. Use of EDAC Codes

The third aspect to consider when dealing with MBUs on the Interleaver consists in studying if the use of EDAC codes, as Hamming ones, improves the fault tolerance of the system. In this situation, Hamming codes are employed when physical interleaving is performed to the data, since in this case

MBUs are converted to SEUs, and the Hamming decoder can correct these single errors, increasing the fault tolerance of the system (cases 6 and 8). However, when no physical interleaving is applied (cases 2 and 4), it may be better not to introduce any EDAC code to protect the information bits, because in this situation, the information redundancy is not enough to correct horizontal MBUs (in the case of words written by rows) because flipped bits belong to the same coded word. Furthermore, in the case of MBUs on the parity bits of the same protected word the correction logic could induce errors to the data. However, vertical and diagonal MBUs on Hamming protected words (and no physical interleaving) will be corrected as SEUs because each bit-flip will correspond to different coded words.

### 3.4. Additional Factors and Summary

Additional factors have to be considered when protecting the interleaver to deal with multiple simultaneous soft errors using any of the different commented implementations and studying the fault tolerance of the communication system, such as the SNR of the signal at the receiver or its modulation and puncturing, and the word size of the memory used for the block interleaver implementation.

The main conclusion of this section is that the system or communication interleaver process can be used itself as a protection technique to deal with MBUs as it is a specific type of physical interleaver (case 1 better than case 3 with no extra cost). Here the concept of the system knowledge is utilized, when considering the inherent redundancy of the information that comes from the convolutional encoder and the use of the communication interleaver as a physical one.

Besides, it has been shown that the fault tolerance of the combination of the communication interleaver and the physical interleaver (case 5) must be analyzed for each specific case, so in the case of using a memory with physical interleaving it would be a

0	3	6	1	4	7	2	5	8
9	12	15	10	13	16	11	14	17

Figure 7. Physical interleaving and communication interleaving pattern in memory read operation. Bits stored on interleaver (minimal logical distance of 3).

good option to apply the communication interleaving pattern in the memory read operation (case 7), what means not to combine the two patterns. Also, if an EDAC code is utilized too it would be better to combine it with the use of physical interleaver (case 8 versus case 4). In this situation, combining the two interleaver patterns and EDAC codes or not combining them (case 6 versus case 8) would have no fault tolerance differences against MBUs on the block interleaver (in the case of physical interleaving greater than the MBU size), because if the physical interleaving is higher than the MBU size the EDAC codes can correct the errors, so it is irrelevant if the bits on the interleaver are stored using the combination or not of the two interleaving patterns.

Note that the conclusions commented for the interleaver at the transmitter side could be, obviously, extrapolated to the deinterleaver used in the receiver side directly, considering that it performs the dual operation. So, for example, if we conclude that it is better to apply the interleaving pattern before writing the memory in the interleaver side, that would mean that it would be better to apply the deinterleaving pattern in the memory read operation at the receiver side. Table 2 illustrates the equivalence between cases described in Table 1 for the interleaver at the transmitter side (second column) and the deinterleaver at the receiver side (third column).

Table 2. Duality between interleaver and deinterleaver.

Description	Interleaver	Deinterleaver
Write operation, no physical interleaving and no EDAC codes	Case 1	Case 3
Write operation, no physical interleaving and EDAC codes	Case 2	Case 4
Read operation, no physical interleaving and no EDAC codes	Case 3	Case 1
Read operation, no physical interleaving and EDAC codes	Case 4	Case 2
Write operation, physical interleaving and no EDAC codes	Case 5	Case 7
Write operation, physical interleaving and EDAC codes	Case 6	Case 8
Read operation, physical interleaving and no EDAC codes	Case 7	Case 5
Read operation, physical interleaving and EDAC codes	Case 8	Case 6

Table 3. Rate-dependent parameters 9Standard 802.11a).

Data rate (Mbps)	Modulation	Coding rate	Coded bits per OFDM symbol ( $N_{CBPSC}$ )	Data bits per OFDM symbol ( $N_{DBPSC}$ )
6	BPSK	1/2	48	24
9	BPSK	3/4	48	36
12	QPSK	1/2	96	48
18	QPSK	3/4	96	72
24	16QAM	1/2	192	96
36	16QAM	3/4	192	144
48	64QAM	2/3	288	192
54	64QAM	3/4	288	216

#### 4. Case Study and Simulation Results

The interleaver module of the 802.11a Wireless communication system has been selected as case study in order to obtain simulation results that prove the main conclusions of the previous section.

The 802.11a communication system can operate at different rates, so different puncturing rates and modulations are defined in the standard [24] as it can be seen in Table 3.

The interleaving pattern of the 802.11a communication system performs two permutations. The first one is usually based on a bit-wise block interleaver with 16 rows and  $N_{CBPSC}/16$  columns and the second one is only applied to QAM modulations type. As it has been commented these permutations can be used themselves to increase the fault tolerance of the whole system against MBUs on the block interleaver, using the communication interleaving as physical one when it is applied in the write operation of the interleaver module. Simulation results related with the addition of physical interleaving enhancement to the block interleaver will be presented too for the case study of the 802.11a communication system.

Due to the characteristics of the information processed by the block interleaver that comes from the convolutional encoder [8, 24, 25] and the more probable MBU patterns [23], the higher the logical distance between contiguous physical bits is the more powerful the Viterbi decoder is to overcome from simultaneous soft errors. Therefore, physical interleaving has to be applied to the bits stored in the block interleaver that come from the convolutional

Table 4. Simulation specifications.

Specifications	
Information packet size (bytes)	100 bytes
Number of packets per simulation	1,000 packets/simulation
Number of simulations performed to average results	10 simulations/result
Block interleaver word size (bits)	Different cases: 3,4,8,12 and 16 bits
MBU sizes (bits)	2, 3 and 4 bits
Fault injection side	Deinterleaver (receiver)
SNR (dB)	Depending on the modulation used
Channel noise	AWGN
Physical interleaving pattern	1 of 4 (when it is applied)

encoder. If this physical distance introduced to contiguous logical bits is made by means of applying the communication interleaving pattern in the memory write operation, the ability to recover from simultaneous soft errors will depend on the specific communication interleaving pattern used (defined in the standard [24], for the 802.11a case study), the word size of the block interleaver memory and the used puncturing rate (that introduces some kind of weakness in the transmitted sequence because of the deletion of some of the output bits of the convolutional encoder). Moreover, other parameters as the SNR of the signal at the receiver must be considered too.

To perform the simulations a whole Matlab description of the 802.11a communication system has been

0	16	32
1	17	33
2	18	34
3	19	35
4	20	36
5	21	37
6	22	38
7	23	39
8	24	40
9	25	41
10	26	42
11	27	43
12	28	44
13	29	45
14	30	46
15	31	47

3-bit words

0	16	32	1
17	33	2	18
34	3	19	35
4	20	36	5
21	37	6	22
38	7	23	39
8	24	40	9
25	41	10	26
42	11	27	43
12	28	44	13
29	45	14	30
46	15	31	47

4-bit words

0	16	32	1	17	33	2	18
34	3	19	35	4	20	36	5
21	37	6	22	38	7	23	39
8	24	40	9	25	41	10	26
42	11	27	43	12	28	44	13
29	45	14	30	46	15	31	47

8-bit words

0	16	32	1	17	33	2	18	34	3	19	35
4	20	36	5	21	37	6	22	38	7	23	39
8	24	40	9	25	41	10	26	42	11	27	43
12	28	44	13	29	45	14	30	46	15	31	47

12-bit words

0	16	32	1	17	33	2	18	34	3	19	35	4	20	36	5
21	37	6	22	38	7	23	39	8	24	40	9	25	41	10	26
42	11	27	43	12	28	44	13	29	45	14	30	46	15	31	47

16-bit words

Figure 8. Different word sizes (3, 4, 8, 12 and 16 bits) for interleaver module of 802.11a and BPSK modulation.

used [8], where an MBU generator with similar patterns to those described in [23] has been developed.

Because of different modulations and puncturing rates are defined in the standard, one swept of the SNR of the signal at the receiver has been implemented for each modulation in order to find the minimum SNR that assures an acceptable packet error rate (PER) at the receiver (in absence of MBUs), similar to the one that the 802.11a systems would operate for each transmission rate or modulation. This has been made in order to see the fault tolerance capacity against MBUs on deinterleaver. Sets of 1000 random packets of 100 bytes for each simulation and MBUs of 2, 3 and 4 bit-flips have been inserted into each packet. Each simulation has been repeated 10 times to obtain an average result.

All the simulations have been performed based on the mentioned specifications that are summarized in Table 4.

It must be noticed that the fault injection has been performed on the deinterleaver module (at the receiver side) because if bit flips are injected on the interleaver (transmitter side) their effects could be modified by the channel when they are sent from the transmitter to the receiver.

#### 4.1. Memory Without Physical Interleaving

The first simulation focuses on cases 1 and 3 (see Table 1), where the only fault tolerant mechanism to deal with MBUs is the Communication interleaving pattern (no EDAC codes and no physical interleaving) applied in the write (case 1) or in the read (case 3) operation of the block interleaver. In this situation, the system knowledge of the convolutional code stored in the interleaver, the interleaving pattern (defined at standard [24]) and the word size of the block interleaver could be combined in order to increase the fault tolerance of the whole system in the case of MBUs on the interleaver, applying the interleaving pattern in the memory write operation (case 1).

The higher the logical distance is the more probable to recover from errors, (characteristic of the convolutional encoder). In Fig. 8 and Table 5, the minimal logical distances between adjacent memory positions for different block interleaver word sizes (3, 4, 8, 12 and 16 bits) and modulations are illustrated.

The analysis of the fault tolerance of the different block interleaver word sizes, illustrated in Table 5,

reveals that a word size of 8 bits could be selected as the optimum value, with the exception of BPSK modulation where the 16-bit word shows a bigger minimal logical distance. In order to prove the commented considerations several fault injections campaigns of 2, 3 and 4-bit MBUs on deinterleaver (deinterleaving pattern applied in the memory read operation, equivalent to the case 1 for interleaver at transmitter side, see Table 2) have been implemented for the all the memory word sizes and modulations illustrated in Table 5. Figure 9 shows some results in average number of erroneous information packets after the Viterbi decoder for 8-bit, 12- and 16-bit word sizes, and a set of the modulations and puncturing rates defined at standard, those that have a puncturing rate of R1/2 (see Table 3).

As it can be observed from Fig. 9 a quasi-optimal word size for the block interleaver (and deinterleaver) can be found for each specific Communication interleaving pattern (in order to obtain a more fault tolerant system against MBUs on interleaver/deinterleaver). In the case of 802.11a, a word size of 8 bits can be selected as the optimal one. So the rest of the simulation results will be obtained using a block interleaver of 8-bit words.

Simulation results in average number of erroneous information packets for all the modulations defined in the 802.11a standard have been obtained and represented in Fig. 10 and Fig. 11, applying the deinterleaving pattern in the memory read operation (solid lines, case 1 of deinterleaver column in Table 2) and in the write one (dotted lines, case 3 of deinterleaver column in Table 2) in order to compare the results. It must be noted that the fault injection has been performed on the deinterleaver module and in such case, applying the deinterleaving pattern in the memory write operation means that the content of the deinterleaver is logically ordered (the opposite than for the case of the interleaver on the transmitter

Table 5. Minimal logical distance of adjacent memory positions.

Modulation	3-bits	4-bits	8-bit	12-bit	16-bits
BPSK	1	1	3	4	5
QPSK	16	16	16	2	13
16QAM	16	16	16	1	16
64QAM	16	16	16	16	1

Modulations defined at 802.11a standard and different word sizes.

## Fault Tolerance Analysis of Communication System Interleavers

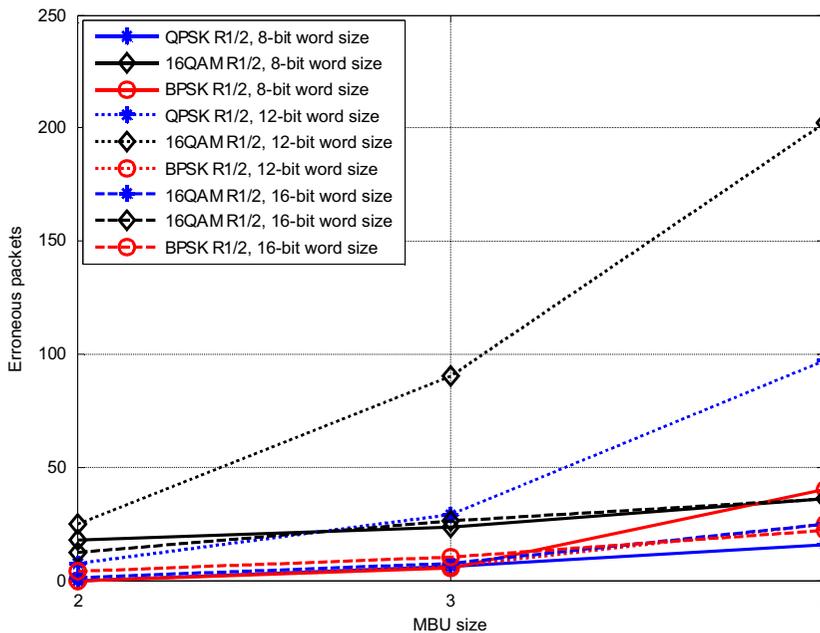


Figure 9. Average number of erroneous packets. Different block interleaver word sizes, modulations and MBU sizes.

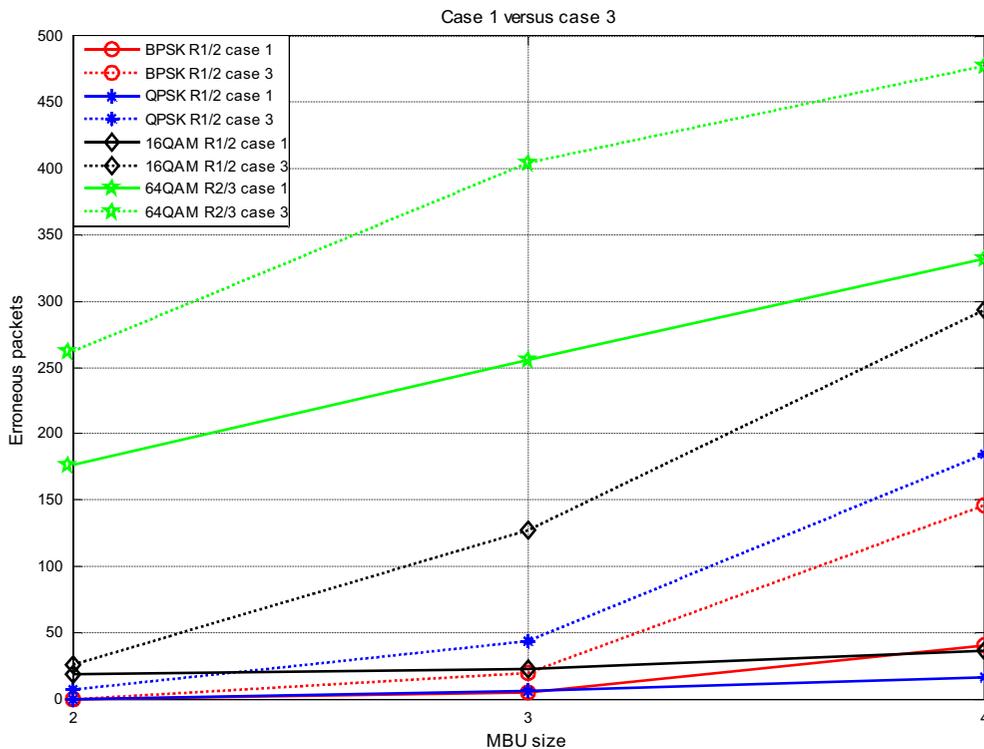


Figure 10. Average number of erroneous information packets for different modulations and deinterleaver pattern in write or in read operation (R1/2, except for 64QAM with R2/3).

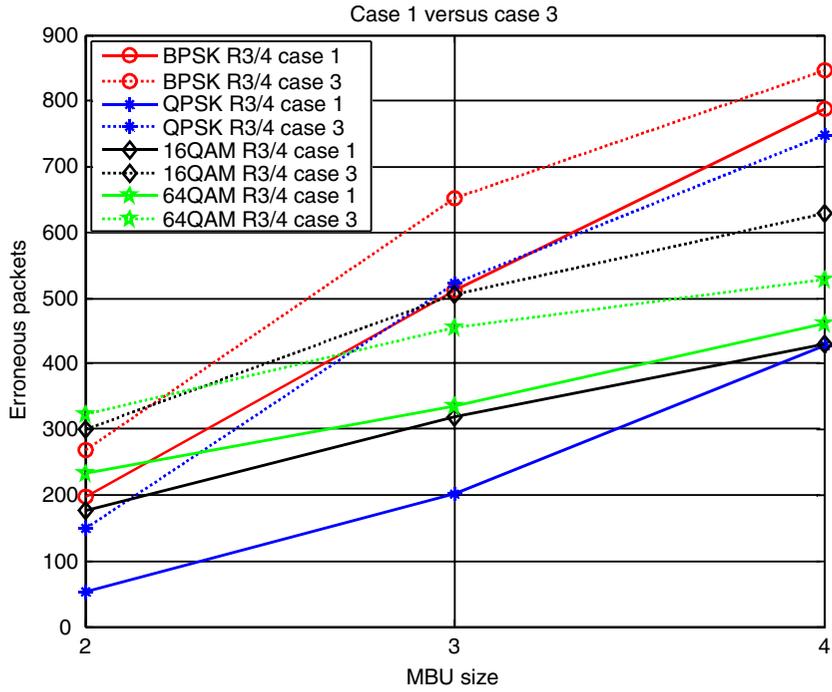


Figure 11. Average number of erroneous information packets, different modulations and deinterleaver pattern in write or in read operation (R3/4).

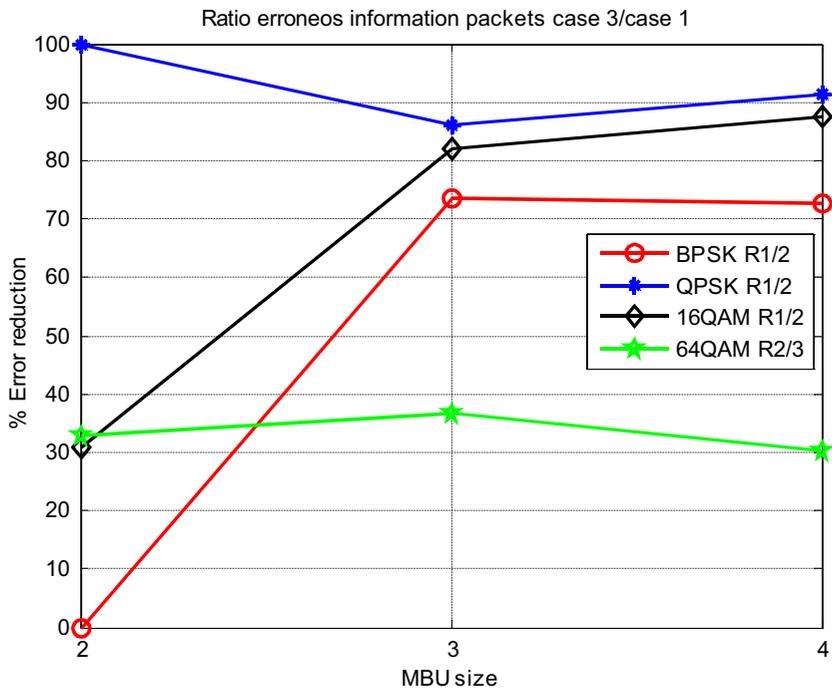


Figure 12. Error reduction percentage of case 1 vs. case 3.

side) and if it is applied in the read operation the content of the deinterleaver is disordered.

Figure 10 illustrates the results for the lowest puncturing rates of all the modulations defined in the standard ( $R=1/2$ , except for the 64-QAM modulation) and shows that the option of applying the deinterleaving pattern in the memory read operation (or the interleaving pattern in the memory write operation, that means case 1 of Table 2) has better results than applying the deinterleaving pattern in the write operation (case 3 of deinterleaver column of Table 2, dotted lines). The higher puncturing rate of the 64-QAM modulation ( $R=2/3$ , see standard specifications [24]) justifies the higher average number of erroneous packets for this modulation due to the deletion of bits in order to transmit data at higher rates.

In Fig. 11 similar results to the ones shown in Fig. 10 are illustrated, but, in this case, they have been obtained for the higher puncturing rate of each of the modulations defined in the 802.11a standard ( $R=3/4$ ; see Table 3).

From Fig. 11, it can be seen that in the case of higher puncturing rates (Figs. 10 and 11 64QAM modulation) the fault tolerance of the system against MBUs on deinterleaver is decreased for every used modulation.

In Figs. 10 and 11 it can be seen that block deinterleaver (or interleaver) implementation of case 1 (solid line, deinterleaver pattern in memory read operation) has better effectiveness results against MBUs than the implementation of case 3 (dotted lines, deinterleaver pattern in memory write operation). That means that when no extra protection mechanisms against multiple simultaneous soft errors are used to protect the block deinterleaver (or interleaver) the communication interleaving pattern can be used itself as a protection technique similar to the use of physical interleaving enhancement. Moreover, in order to see the reduction percentage of erroneous packets of case 1 versus case 3, Fig. 12 has been obtained.

The results illustrated in Fig. 12 have been obtained considering that for each modulation the

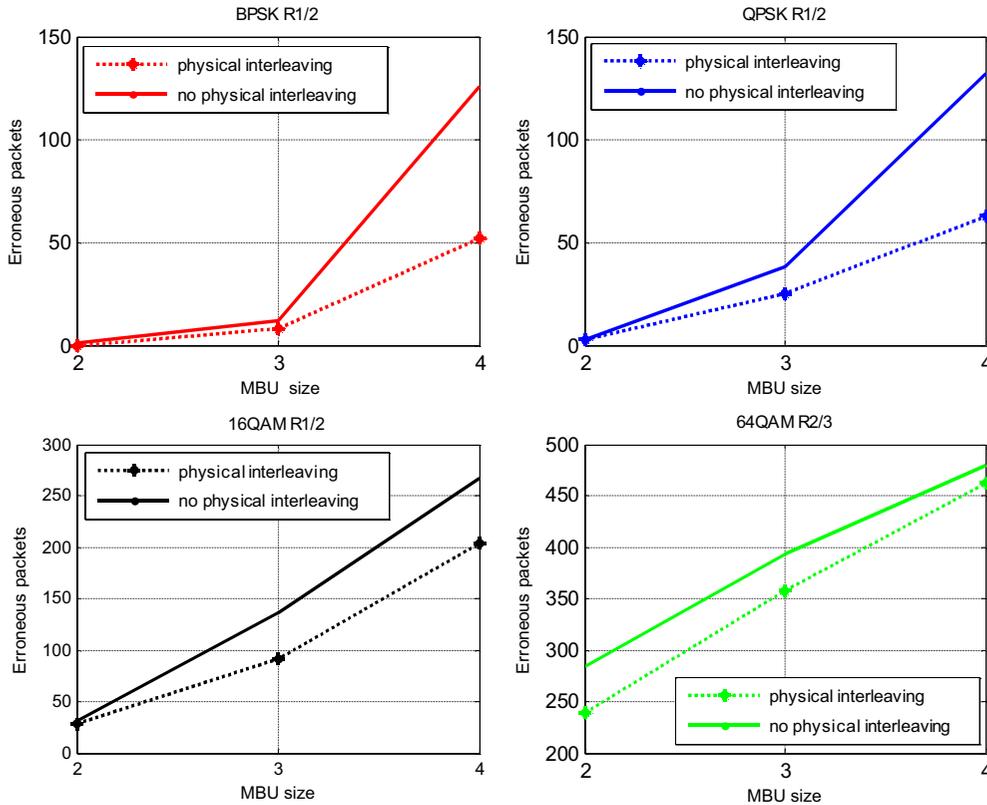


Figure 13. Case 7 (dotted lines) vs. case 3 (solid lines) on the deinterleaver side, distinct transmission rates or modulations.

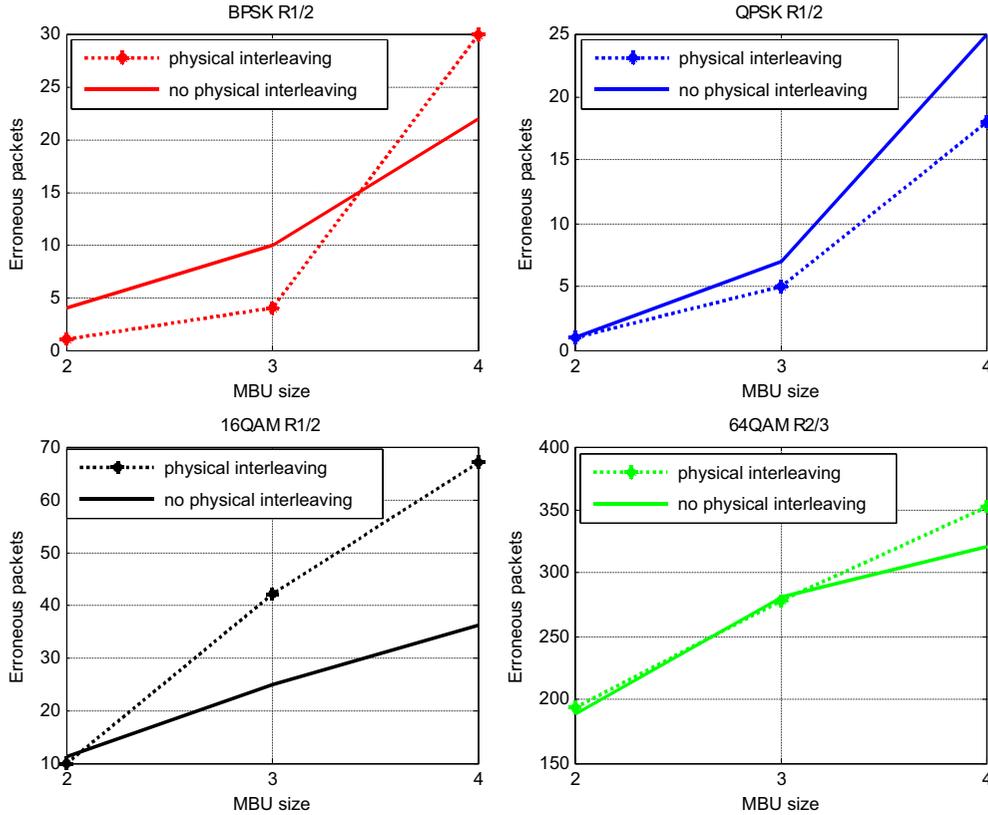


Figure 14. Case 5 (dotted lines) vs. case 1 (solid lines) on deinterleaver side, distinct transmission rates or modulations.

erroneous packet results for case 3 represent the 100% of errors and the difference between erroneous information packets of case 3 and case 1, in percentage, corresponds to the error reduction for each case. As it can be analyzed, except for the case of BPSK and 2-bit MBU size where no error reduction is observed (due to the reduced interleaver depth), the minimum error reduction percentage observed is higher than 30% for the case of 64-QAM modulation, that has a puncturing rate of  $R2/3$ .

#### 4.2. Memory with physical interleaving (Cases 5 to 8)

Simulation results, taking into account that the block interleaver can be implemented using a memory with physical interleaving, are shown in Figs. 13 and 14, where the combination or not of physical and communication interleaving patterns are shown and compared with the results obtained when no physical interleaving is applied, for different modulations. The results have been obtained considering the same

specifications shown in Table 4 and a set of the modulations and rates shown in Table 3 (defined in the 802.11a standard) for a physical interleaving pattern of 1 of 4 (that means higher than the maximum MBU size injected).

Figure 13 shows the comparison between using or not a memory with physical interleaving when the deinterleaver pattern is applied in the memory write operation (case 7 versus case 3 of deinterleaver column

Table 6. Effectiveness analysis of different Interleaver implementations.

Cases	Fault tolerance	Extra cost
Case 1	Good	No extra cost
Case 3	Bad	No extra cost
Cases 2 and 4	Depending on the case	Average
Case 5	Depending on the case	Low
Case 7	Good	Low
Cases 6 and 8	Excellent	High

in Table 2). In this case, the use of physical interleaver (case 7, dotted lines) increases the fault tolerance of the system with respect to its absence (case 3, solid lines), reducing the average number of erroneous information packets after the Viterbi decoder, as it can be observed from the plots of each modulation. These results are independent of the communication interleaving pattern, that means that they only depends on the physical interleaving pattern applied, and they could be better if EDAC codes are applied too, because in this case, if the physical pattern is larger than the MBU size, the correction logic of the decoder could correct the error before the data comes to the Viterbi decoder. As it can be seen in the figure, the use of physical interleaving enhances the fault tolerance of the system what is more visible as we move to higher MBU sizes (for the case of modulations with puncturing rates of  $R=1/2$ , BPSK, QPSK and 16QAM). However, and due to the puncturing applied to the information bits transmitted (deletion of some of the data bits before transmitting them) if a puncturing rate different of  $R=1/2$  (that means no deletion of any data bit) is used, the commented enhancement of the fault tolerance when the MBU size increases is not observed (see fourth graphic of Fig. 13, 64QAM modulation and  $R=2/3$ ).

However, when the communication interleaving pattern is applied in the memory read operation of the deinterleaver block (that means that the content of the deinterleaver is disordered based on the communication interleaving pattern) it would be better not to use a memory with physical interleaving, because in such situation, the fault tolerance of the whole communication system against MBUs on deinterleaver (or interleaver) should be studied for each specific combination of patterns. Figure 14 shows the results for the commented case, where it can be seen if the combination of the two interleaving patterns (dotted lines, case 5 of deinterleaver column in Table 2) is better or worse than the no combination (solid lines, case 1 of deinterleaver column in Table 2) depending on the specific case and communication pattern (compare results for QAM and PSK modulations).

Therefore, the main conclusions commented in section 3, for a general case, have been validated for the case study of the 802.11a communication system.

Table 6 summarizes the effectiveness results of the different implementations and combinations analyzed and an estimation of the extra cost of each case is illustrated. To this end, we have considered that the

extra cost of physical interleaving is lower than the extra cost of EDAC codes inclusion. So low extra cost is obtained when physical interleaving is added to the memory, an average cost overhead when Hamming or any EDAC code is selected and, in case of combining Hamming codes and physical interleaving, a higher cost is introduced in the final design.

## 5. Conclusions and Future Work

In this paper different communication block interleaver implementations have been analyzed and compared in terms of effectiveness results against multiple soft errors. The main idea behind the implementations compared consists in using the system knowledge of the communication system (in this case, the knowledge that the data processed by the interleaver comes from a convolutional encoder) in order to use the communication interleaving pattern itself as physical interleaving pattern (usually employed to protect memories against soft errors). Therefore, the main conclusions obtained are the next ones:

- The communication interleaving pattern can be used to protect the block interleaver (and deinterleaver) memory content against multiple soft errors.
- The word size of the block interleaver memory has influence in the fault tolerance effectiveness results, and a specific study for each communication system must be made in order to find the best word size.
- The combination of the communication interleaving pattern with a physical one in the content of the block interleaver (and deinterleaver) is not a good option because the fault tolerance against MBUs depends on each specific combination.

As future work, other communication system modules will be analyzed or protected against SEUs and MBUs using the concept of system knowledge. In particular, a GNSS signal processor is being analyzed in order to protect it with similar fault tolerance to TMR or EDAC codes but with lower area cost.

## References

1. C. W. Slayman, "Cache and Memory Error Detection, Correction and Reduction Techniques for Terrestrial Servers and Workstations", IEEE Transactions on Device and Material Reliability, vol. 5, no. 3, September 2005, pp. 397–404.

2. D. Schrimpf, D.M. Fleetwood, "Radiation effects and soft errors in integrated circuits and electronic devices", Eds., World Scientific Publishing, Singapore 2004.
3. J. E. Mazur, "An Overview of the Space Radiation Environment", The Aerospace Corporation Magazine of Advances in Aerospace Technology, vol. 4, no. 2, Summer 2003.
4. R. Baumann, "Soft Errors in Advanced Computer Systems" IEEE Des. Test Comput., vol. 22, no. 3, May 2005, pp. 258–2618.
5. M. Nicolaidis, "Design for Soft Error Mitigation", IEEE Transactions on Device and Material Reliability, vol. 5, no. 3, September 2005, pp. 405–418.
6. W. Heidergott, "SEU Tolerant Device, Circuit and Processor Design" DAC 2005, June 13–17, 2005, pp. 5–10, Anaheim, CA, USA. Copyright 2005 ACM 1-59593-058-2/05/0006.
7. E. C. Regla, V. K. Konangi, M. A. Seibert, "Protocols for inter-satellite communication in a formation flying system" 20th AIAA International Communication Satellite Systems Conference and Exhibit, 12–15 May 2002, Montreal, Quebec, Canada.
8. J. Heiskala, J. Terry, "OFDM Wireless LANs: A Theoretical and Practical Guide." SAMS.
9. P. Reyes, P. Reviriego, J.A. Maestro, O. Ruano, "A new Protection Technique for Finite Impulse Response (FIR) Filters in the Presence of Soft Errors", IEEE International Symposium on Industrial Electronics, ISIE 2007, June 4–7, 2007, pp. 3328–3333, Vigo (Spain).
10. P. Reyes, P. Reviriego, J.A. Maestro, O. Ruano, "New Protection Techniques against SEUs for Moving Average Filters in a Radiation Environment", IEEE Trans. Nucl. Sci., vol. 54, no.4, August 2007, pp. 957–964.
11. P. Reyes, P. Reviriego, O. Ruano, J.A. Maestro, "Efficient Structures for the Implementation of Moving Average Filters in the Presence of SEUs. using System Knowledge", the 2006 Radiation Effects on Components and Systems Workshop, RADECS 06, September 27–31th, Athens, (Greece).
12. P. Reviriego, P. Reyes, J.A. Maestro, O. Ruano, "System Knowledge-Based Techniques against SEUs for Adaptive Filters", the 2007 Radiation Effects on Components and Systems Workshop, RADECS 07, September 10–14, 2007 Deauville (France).
13. P. J. Meaney, S.B. Swaney, P.N. Sanda, L. Spainhower, "IBM z990 Soft Error Detection and Recovery", IEEE Transactions on Device and Materials Reliability, vol.5, no. 3, September 2005, pp. 419–427.
14. F. G. Lima, S. Rezugui, E. Cota, L. Carro, M. Lubaszewski, R. Velazco, R. Reis. "Designing and Testing a Radiation Hardened 8051-like Micro-controller", 13th Symposium on Integrated Circuits and Systems Design, 18–24 September, 2000, pp. 250–260 (SBCCI'00).
15. Z. Stamenković, C. Wolf, G. Schoof, J. Gaisler, "An Implementation Study on Fault Tolerant LEON-3 Processor System" D & R Industry Articles, <http://www.us.design-reuse.com/articles/article15502.html>.
16. A. Reddy, P. Banarjee, "Algorithm Based fault detection for signal processing applications", IEEE Trans. Comput., vol. 39, no. 10, October 1990, pp. 1304–1308.
17. B. Shim, N. R. Shanbhag, S.Lee, "Energy-Efficient Soft Error-Tolerant Digital Signal Processing", IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 14, no. 4, April 2006, pp. 336–348.
18. E. Schüler, L. Carro, "Reliable digital circuits design using Sigma-Delta modulated signals" 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 3–5 October 2005, pp. 314–324.
19. E. Schüler, D. S. Farenzena, L. Carro, "Evaluating Sigma-Delta modulated signals to develop fault-tolerant circuits", Eleventh IEEE European Test Symposium, 21–24 May 2006, pp. 137–144 (ETS'06).
20. K. D. Wolfram, H. J. Bloom, "New Radiation- Hardened High-Speed Serial Data Bus For Satellite Onboard Communications" IEEE International Symposium on Geoscience and Remote Sensing, 2004. IGARSS 04. vol. 1, 20–24 September 2004.
21. J.B. Destro-Filho, D. W. Matolak, "Effects of Single Event Upsets on Satellite Communications: Issues for Blind Equalizer Design", 6th European Conference on Radiation and Its Effects on Components and Systems, 10–14 September 2001.
22. E. Tell, D. Liu, "A Hardware Architecture for a multi block interleaver", IEEE International Conference on Circuits and Systems for Communications, June 30–July 2, 2004, Moscow, Russia.
23. D. Radaelli, H. Puchner, S. Wong, S. Daniel, "Investigation of Multi-Bit Upsets in a 150 nm Technology SRAM Device", IEEE Trans. Nucl. Sci., vol. 52, no. 6, December 2005, pp. 2433–2437.
24. "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications High-speed Physical Layer in the 5 GHz Band" LAN/MAN Standards Committee of the IEEE Computer Society. Adopted by the ISO/IEC and redesignated as ISO/IEC 8802-11:1999/Amd 1:2000(E).
25. T.H. Meng, B. McFarland, D. Su and J. Thomson, "Design and Implementation of an All-CMOS 802.11a Wireless LAN Chipset", IEEE Communications Magazine, Topics in Circuits for Communications, vol. 41, no. 8, August 2003, pp. 160–168.



**Pilar Reyes** holds a M.Sc. in Telecommunication Engineering (2004) from Universidad de Sevilla, as well as a degree in Industrial Engineering (2000) from Universidad de Córdoba. She has been a researcher with Anafocus, working on the design an implementation of a Vision System on Chip. Currently, she is a full-time researcher at Universidad Antonio de Nebrija, as well as a Ph.D. candidate.

## Fault Tolerance Analysis of Communication System Interleavers



**Pedro Reviriego** received the M.Sc. and Ph.D. degrees (Honors) from Technical University of Madrid in 1994 and 1997, both in Telecommunications Engineering. From 1997 to 2000 he was an R&D engineer at Teldat working on router implementation. In 2000 he joined Massana to work on the development of transceivers. During 2003 he was a Visiting professor at University Carlos III. From 2004 to 2007 he was Distinguished Member of Technical Staff with LSI Corporation working on the development of Ethernet transceivers. He is currently with Universidad Antonio de Nebrija. His research interests are fault tolerant systems, performance evaluation of communication networks and the design of physical layer communication devices. He has authored numerous papers in international conferences and journals. He has also participated in the IEEE 802.3 standardization for 10GBaseT.



**Juan Antonio Maestro** holds a M.Sc. degree in Physics (1994) and a Ph.D. degree in Computer Science (1999) from

Universidad Complutense de Madrid. He has served both as a lecturer and researcher at several universities, as Universidad Complutense de Madrid, UNED (Open University), Saint Louis University and Universidad Antonio de Nebrija, where he currently manages the Computer Architecture and Technology Group. His current activities are oriented to the Space field, where several projects on reliability and radiation protection, as well as collaborations with the European Space Agency. He is the author of numerous technical publications, both in journals and international conferences. Besides from this, he has worked for several multinational companies, managing IT projects as a PMP, and organizing support departments. His areas of interest include High Level Synthesis and co-Synthesis, Signal Processing and Real-Time systems, Fault-tolerance and Reliability.



**Oscar Ruano** holds a M.Sc. in Computer Engineering (2005) from Universidad Antonio de Nebrija. He has worked with different multinational companies in the IT consultancy field, as Accenture. Currently, he is a full-time researcher at Universidad Antonio de Nebrija, as well as a Ph.D. candidate.