

Improved Frequency Domain Architecture for the Digital Block of a Hardware Simulator for MIMO Radio Channels

Bachir Habib, Gheorghe Zaharia, Ghaïs El Zein

Institut d'Electronique et de Télécommunication de Rennes, UMR CNRS 6164

20 av. des Buttes de Coësmes, CS 70839 -35708, Rennes cedex 7, France

bachir.habib@insa-rennes.fr

Abstract—This paper presents a new frequency domain architecture for the digital block of a hardware simulator of MIMO propagation channels. This simulator can be used for UMTS and WLAN applications in indoor and outdoor environments. A hardware simulator must reproduce the behavior of the radio propagation channel, thus making it possible to test “on table” the mobile radio equipments. The advantages are: low cost, short test duration, possibility to ensure the same test conditions in order to compare the performance of various equipments. After the presentation of the general characteristics of the hardware simulator, the new architecture of the digital block is presented and designed on a Xilinx Virtex-IV FPGA, and its accuracy is analyzed.

Keywords—Hardware Simulator, Radio Channel, MIMO, FPGA

I. INTRODUCTION

Universal Mobile Telecommunications System (UMTS) and Wireless Local Area Networks (WLAN) are the mobile and wireless telecommunications systems of third generation and beyond able to offer to general public the high-rate multimedia services. Multiple-Input Multiple-Output (MIMO) systems make use of antenna arrays simultaneously at both transmitter and receiver site to improve the capacity and/or the system performance. However, the transmitted electromagnetic waves interact with the propagation environment. Thus, it is necessary to take into account the main propagation parameters during the design of the future communication systems.

Hardware simulators of mobile radio channel are very useful for the test and verification of wireless communication systems. Their performance can be evaluated and compared in the same test conditions. The channel simulators are rather bulky and costly. They are standalone units that provide the fading signal in the form of analog signal or digital samples [1], [2]. With continuing increases of the field programmable gate (FPGA) capacity, entire baseband systems can be efficiently mapped onto faster FPGAs for more efficient prototyping, testing and verification. Larger and faster FPGAs permit the integration of a channel simulator along with the receiver noise simulator and the signal processing blocks for rapid and cost-effective prototyping and design verification. As shown in [3], the FPGAs provide the greatest design flexibility and the visibility of resource utilization. They are ideal for rapid prototyping and research use such as testbed [4].

At IETR, several architectures of the digital block of a hardware simulator have been studied, in both time and frequency domains [5], [6]. In [7] a new method for determining the parameters of a channel simulator was

developed by fitting the space time-frequency cross-correlation matrix of the simulation model to the estimated matrix of a real-world channel. This solution can be considered only as heuristic method because it shows that the obtained error can be important. Typically, wireless channels are commonly simulated using finite impulse response (FIR) filters, as in [8] and [9]. Nowadays, different approaches have been widely used in filtering such as distributed arithmetic (DA) and canonical signed digits (CSDs). However, with convolution-based architectures, the complexity with MIMO array size becomes impractical. Thus, frequency domain architectures have been presented, as in [6] and [8]. Moreover, a proposed VLSI implementation shows that for higher order MIMO arrays, frequency domain architectures are not only computationally efficient but also highly modular and scalable by design. However, the previous considered frequency domain architectures operate correctly only for signals with a number of samples not exceeding N , where N is the size of the FFT (Fast Fourier Transform) module. Thus, in this paper, we propose a new frequency architecture avoiding this limitation by providing a “ping-pong” system.

The rest of this paper is organized as follows. Section II presents the previous time and frequency domain architectures of the digital block of the hardware simulator in, The new frequency domain architecture is also described. Section III shows the actual realization of the digital block. The prototyping platform is described and simulations carried out give their first results. The accuracy of the new architecture is also analyzed. Lastly, Section IV presents some concluding remarks.

II. HARDWARE SIMULATOR: PRINCIPLE, ARCHITECTURE AND OPERATION

The simulator must reproduce the behavior of a MIMO propagation channel. It operates with radio frequency signals (2 GHz for UMTS and 5 GHz for WLAN). In order to make adjacent channels interference tests for UMTS, it is useful to consider three successive channels, thus 3×5 MHz bandwidth. Therefore, the frequency bandwidth B is 15 MHz for UMTS and 20 MHz for WLAN. The simulator must be able to accept input signals with wide power range, between -50 and 33 dBm, which implies a power control for the simulator inputs.

The design and realization of the RF blocks for UMTS systems were completed in a previous project [6]. The objectives of the PALMYRE II* project mainly concern the

*This work is a part of CPER PALMYRE II Project which is financially supported by “Region Bretagne”

channel model block and the digital block of the MIMO simulator, as shown in Fig. 1.

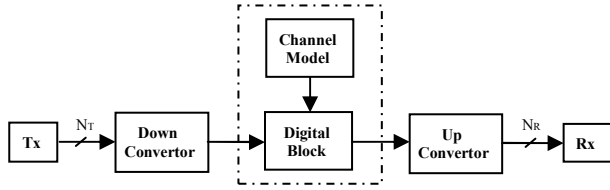


Figure 1. Block diagram of a one-way MIMO channel

A. Channel Model

A MIMO channel is composed of several time variant correlated SISO channels. Fig. 2 illustrates a MIMO channel with $N_T = 2$ transmit antennas and $N_R = 2$ receive antennas.

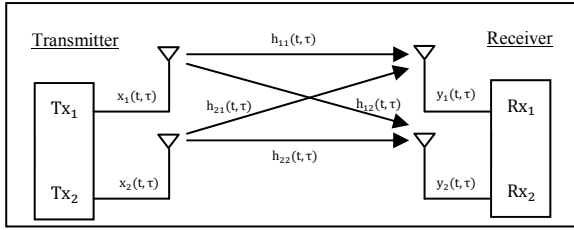


Figure 2. MIMO channel (2 x 2 SISO channels)

For this MIMO channel, the received signal $y_j(t, \tau)$ is:

$$y_j(t, \tau) = x_1(\tau) * h_{1j}(t, \tau) + x_2(\tau) * h_{2j}(t, \tau), \quad j = 1, 2 \quad (1)$$

For a hardware implementation, it is easier to use the Fourier transform to obtain an algebraic product:

$$Y_j(t, f) = X_1(f) \cdot H_{1j}(t, f) + X_2(f) \cdot H_{2j}(t, f), \quad j = 1, 2 \quad (2)$$

For indoor environments, it is convenient to consider (1) because a FIR filter has, in spite of its complexity, a much lower latency. For outdoor environments, the effective duration W_{t_eff} of the impulse response of the channel is wider. Therefore, in order to reduce the complexity and the cost of the digital block, frequency domain architectures can be used.

The channel models used by the simulator can be obtained from measurements by using a time domain MIMO channel sounder designed and realized at the IETR [10]. Different architectures of antenna arrays are available for outdoor and indoor measurements [11]. It is also possible to use well-known standard stochastic models or deterministic ray-tracing models.

B. Digital Block

In order to have a suitable trade-off between complexity and latency, two solutions can be considered: a time domain approach with FIR filters for the indoor environment and a frequency approach with FFT modules for outdoor environments.

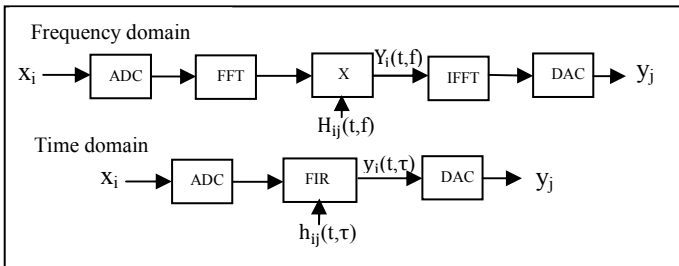


Figure 3. Frequency and time domain architectures of a SISO channel.

Fig. 3 describes simple frequency domain and time domain architectures of the digital block of a SISO channel, which were simulated and tested in [6].

By using the Nyquist-Shannon sampling theorem and the expected performance of the RF/IF filters, the sampling frequency f_s is 40 MHz for UMTS systems and 50 MHz for WLAN systems. However, in order to compare these two architectures, $f_s = 50$ MHz is used for both UMTS and WLAN systems. This choice allows a reasonable low sampling rate and avoids the aliasing problems.

According to the considered propagation environments, Table I summarizes some useful parameters.

TABLE I. SIMULATOR PARAMETERS

	Type	Cell Size	$W_{t_eff}(\mu s)$	N	$W_t(\mu s)$
UMTS (B=15 MHz)	Rural	2-20 km	20	512	10.24
	Urban	0.4-2 km	3.7	128	2.56
	Indoor	20-400 m	0.7	35	0.7
WLAN (B=20 MHz)	Office	40 m	0.39	20	0.4
	Indoor	50-150 m	0.73	37	0.74
	Outdoor	50-150 m	1.16	64	1.28

For these various environments, the size of the FFT blocks is estimated by:

$$N = \frac{W_t}{T_s} = W_t f_s \quad (3)$$

where W_t represents the width of the time window of the channel impulse response. The value N is the closest 2^n value computed for outdoor environments. The number N is imposed by the size of the FFT module and the complexity of the architecture.

C. New Frequency Domain Architecture

This part presents an improved frequency domain architecture for a SISO channel, which can be used in streaming mode, in contrast to the simple frequency domain architecture presented in Fig. 3. First, the error of the simple frequency architecture is presented. Then, the new ‘‘ping-pong’’ frequency architecture is described in details.

In order to test the simple architecture, a continuous Gaussian signal $x(t)$ is considered and long enough to be used in streaming mode:

$$x(t) = x_m e^{-\frac{(t-m_x)^2}{2\sigma_x^2}}, \quad 0 \leq t \leq 3W_t \quad (4)$$

where: $N = 512$, $W_t = N/f_s$, $m_x = 3W_t/2$, $\sigma_x = m_x/3$.

A length of $3W_t$ is sufficient for the test. The FFT block will split the corresponding quantized vector x in three parts (x_1 , x_2 and x_3) of 512 samples each. Applying this signal to the input of a linear system whose impulse response is a continuous signal $h(t)$ also represented on $[0, W_t]$ by a Gaussian signal, we obtain three output vectors y_1 , y_2 and y_3 . To validate the streaming mode, a comparison is made between the concatenation of these three vectors and the theoretical signal $y(t)$, as shown in Fig. 4.

The theoretical result is obtained by the time domain architecture but the frequency domain architecture gives a wrong result. In fact, each partial result y_1 , y_2 and y_3 must have $2N-1$ samples (if x_1 , x_2 , x_3 and h have N samples). An IFFT block gives its result only with N samples. There is a truncation of each partial result y_i . Therefore, the concatenation of these partial results gives a different result.

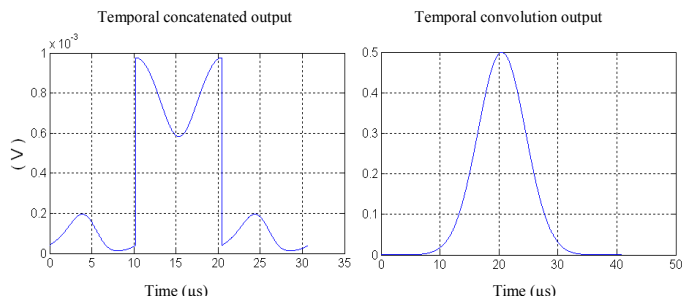


Figure 4. Concatenation of vectors y_1 , y_2 and y_3 , and the theoretical result.

A possible solution is to “complete” each vector x_i with N zeros and to use FFT/IFFT blocks with size two times larger ($2N$). However, to avoid increasing their size, It is convenient to preserve the size of FFT/IFFT blocks and to split the input test vector x into six parts, each one with $N/2 = 256$ samples. We extend each partial vector x_i with a "tail" of 256 zeros, which allow the use of FFT/IFFT blocks with $N = 512$ samples, thus avoiding the truncation of each partial response y_i . Therefore, the new architecture, presented in Fig 5, will operate using two FFT/IFFT blocks of 512 points. Each 256 input samples fed alternately a FFT module due to a switch signal S .

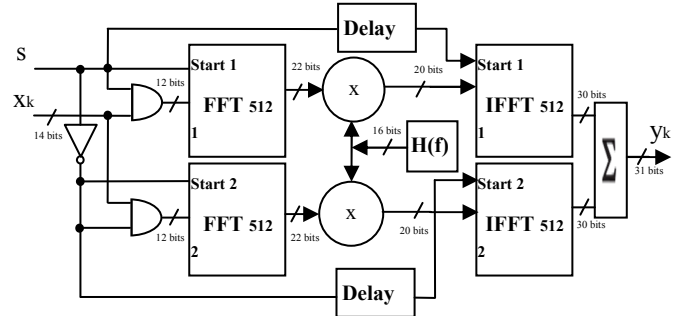


Figure 5. New frequency architecture of a SISO channel.

Each FFT module operates with 12-bit input samples, and has a 12-bit phase factor. The switch signal S , provides alternate use of the two FFT modules. The start input of the FFT modules is active on the rising edge of the switch signal S . The block delay takes into account the processing delay of the FFT modules and the multipliers. Using ModelSim [9], this time is found to be 10.18 μs . Fig. 6 presents the operating principle of the architecture and the result on $4Wt$ of each partial response y_i .

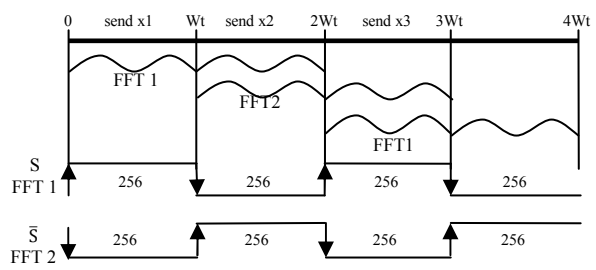


Figure 6. Operating principle of the “ping-pong” architecture

The sliding window truncation block used in [5] and [6] and shown in Fig. 7 is located at the output of the digital adder placed after the IFFT blocks. It is necessary to reduce the number of bits after the sum of the IFFT blocks to 14 bits so that these samples can be accepted by the DAC, while maintaining the highest accuracy.

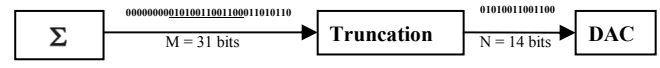


Figure 7. Sliding window truncation, from 34 to 14 bits

III. IMPLEMENTATION

In order to implement the hardware simulator, the adopted solution uses a prototyping platform from Xilinx (XtremeDSP Development Kit-IV for Virtex-4) [12].

A. Description

The XtremeDSP features dual-channel high performance ADCs (AD6645) and DACs (AD9772A) with 14-bit resolution, a user programmable Virtex-4 FPGA, programmable clocks, support for external clock, host interfacing PCI, two banks of ZBT-SRAM, and JTAG interfaces. This development kit is built with a module containing the Virtex-4 SX35 component, selected to correspond to the complexity constraints. It contains a number of arithmetic blocks (DSP blocks) which makes it possible to implement many functions occupying most of the component. This device enables us to implement different time domain or frequency domain architectures and thus to reprogram the component according to the selected (indoor or outdoor) environment.

Several simulations and synthesis are made with Xilinx ISE [12] and ModelSim software [13].

B. Implementation and Results

In the frequency domain, the Fourier transform is realized with the new “ping-pong” architecture with two 512-FFT and two 512-IFFT modules of FPGA. The V4-SX35 utilization summary for this new architecture is given in Table II. The use of FFT/IFFT 512 provides us the test with the worst case for outdoor environments according to Table II.

TABLE II. VIRTEX-4 SX35 UTILIZATION FOR 2 FFTS AND 2 IFFTS IN PING-PONG ARCHITECTURE

Number of slices	13,969 out of 15,360	91%
Number of logic LUTs	20928 out of 30,720	68%
Number of bloc RAM	50 out of 192	26%
Number of DSP48s	150 out of 192	78%
Clock period	7.669 ns	

This architecture uses 91% of slices of the FPGA Virtex-4. Other studies are made in [6] but with a simple architecture using only one FFT with a length of 512, and the simulation results with Virtex-4 (V4-SX35) are presented in Table III.

TABLE III. VIRTEX-4 SX35 UTILIZATION FOR 1 FFTS 512

Number of slices	5996 out of 15,360	39%
Number of logic LUTs	9306 out of 30,720	30%
Number of bloc RAM	29 out of 192	15%
Number of DSP48s	56 out of 192	29%
Clock period	6,8 ns	

The FFT block latency is measured between the time when the data entered in the FFT block and the time when the result is provided, which is roughly a latency of 45 μs , while it is 46 μs for the Ping-Pong architecture. In order to determine the accuracy of the digital block, Gaussian signals are used: $x(t)$ is the input signal, $h(t)$ stands for the channel impulse response and $y(t)$ is the output signal.

If $[-V_m, V_m]$ is the full scale of the converters, $x(t)$ and $h(t)$ are:

$$x(t) = x_m e^{-\frac{(t-m_x)^2}{2\sigma_x^2}} \quad (5)$$

$$h(t) = h_m e^{-\frac{(t-m_h)^2}{2\sigma_h^2}} \quad (6)$$

where $x_m = V_m/2$. The value of x_m must be chosen neither too small in order to obtain a good accuracy of the digitized signal, nor too big in order to avoid over-range values. The parameters of the test signals are chosen in order to obtain the output signal:

$$y(t) = y_m e^{-\frac{(t-m_y)^2}{2\sigma_y^2}} \quad (7)$$

where: $y_m = \frac{V_m}{2}$, $m_y = \frac{W_t}{2}$, $\sigma_y = \frac{m_y}{4}$

Moreover, in order to obtain $y_m = x_m = V_m/2$, h_m is determined by:

$$y_m = x_m \cdot h_m \cdot \frac{\sigma_h \cdot \sigma_x}{\sigma_y} \cdot \sqrt{2\pi} \quad (8)$$

where: $\sigma_y^2 = \sigma_x^2 + \sigma_h^2$

The relative error ε is determined for each sample of the output by:

$$\varepsilon = \frac{Y_{\text{Xilinx}} - Y_{\text{theoretical}}}{Y_{\text{theoretical}}} \cdot 100 [\%] \quad (9)$$

Therefore, the Signal-to-Noise Ratio (SNR) is given by:

$$\text{SNR} = 20 \cdot \log_{10} \left| \frac{Y_{\text{theoretical}}}{Y_{\text{Xilinx}} - Y_{\text{theoretical}}} \right| [\text{dB}] \quad (10)$$

Fig. 8 shows the theoretic and Xilinx signals at the output with the relative error of the computed signal.

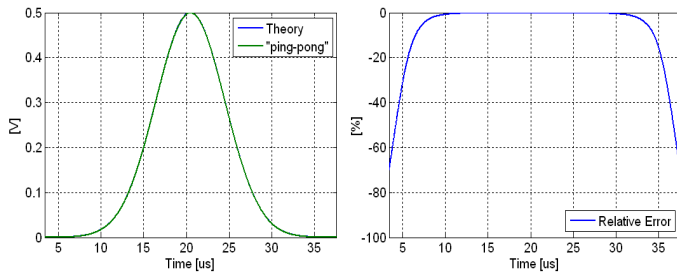


Figure 8. The theoretic and Xilinx output signals, and the relative error

The error is high only for small values of the output signal because the signal Gaussian test is close to 0.

The global values of the relative error ε and of the SNR of the output signal before and after the final truncation are:

$$\varepsilon = \frac{\|e\|}{\|y\|} \times 100 [\%] \quad (11)$$

$$\text{SNR} = 20 \cdot \log_{10} \frac{\|y\|}{\|e\|} [\text{dB}] \quad (12)$$

where y is the theoretical output signal, y_c is the computed signal (with or without truncation) and $e = y_c - y$. For a given digital signal $x = [x_1, x_2, \dots, x_N]$, $\|x\|$ is:

$$\|x\| = \sqrt{\frac{1}{N} \sum_{k=1}^N x_k^2} \quad (13)$$

Table IV shows the global value of the relative error of the Xilinx signal and the SNR between.

TABLE IV. COMPARISON OF THE GLOBAL VALUES OF THE RELATIVE ERROR AND THE SNR

Comparison	Error (%)	SNR (dB)
Output without truncation	0.0647	63.78
Output with sliding truncation	0.0707	63.01
Output with brutal truncation	0.0782	62.13

For input signals with less than 512 samples, it is suitable to use the simple frequency architecture with one FFT with a length of 512. It takes smaller percentage of slices occupation in the FPGA and a smaller latency. However, in the general case, for practical applications, for input signals with more than 512 samples, we must use the new "ping-pong" architecture.

IV. CONCLUSION

In this paper, a new frequency domain architecture was proposed and analyzed. This new architecture can accept long input signals. This architecture was tested with Gaussian signals. The accuracy and the latency of this new architecture have been determined. However, the huge percentage of used slices suggests the use of more performing FPGAs or more economic architectures.

This work will be continued to complete the design of the architecture of the hardware simulator by minimizing the latency of the digital block and the number of slices used for a SISO channel. Other time domain architectures are under test. More measurement campaigns will be carried out for various types of environments (indoor, outdoor) and for both UMTS and WLAN frequency bands. The final objective of these measurements is to obtain realistic and reliable impulse responses of the MIMO channel in order to supply the digital block of the hardware simulator.

REFERENCES

- [1] Wireless Channel Emulator, Spirent Communications, 2006.
- [2] Baseband Fading Simulator ABFS, Reduced costs through baseband simulation, Rohde & Schwarz, 1999.
- [3] P. Murphy, F. Lou, A. Sabharwal, P. Frantz, "An FPGA Based Rapid Prototyping Platform for MIMO Systems", *Asilomar Conf. on Signals, Systems and Computers*, ACSSC, Vol. 1, pp. 900-904, 9-12 Nov. 2003.
- [4] P. Murphy, F. Lou, J. P. Frantz, "A hardware testbed for the implementation and evaluation of MIMO algorithms", *Conf. on Mobile and Wireless Communications Networks*, Singapore, 27-29 Oct. 2003.
- [5] S. Picol, G. Zaharia, D. Houzet and G. El Zein, "Hardware simulator for MIMO radio channels: Design and features of the digital block", *Proc. of IEEE VTC-Fall 2008*, 21-24 Sept. 2008, Calgary, Canada.
- [6] S. Picol-Williamson, G. Zaharia, G. El Zein, "Partie numérique du simulateur matériel pour canal de propagation MIMO", *Edition universitaires européennes*, Sarrebruck, Allemagne, 2010
- [7] D. Umansky, M. Patzold, "Design of Measurement-Based Stochastic Wideband MIMO Channel Simulators", *IEEE Globecom 2009*, Honolulu, Hawaii, 30 Nov. - 4 Dec. 2009
- [8] H. Eslami, S.V. Tran and A.M. Eltawil, "Design and implementation of a scalable channel Emulator for wideband MIMO systems", *IEEE Trans. on Vehicular Technology*, vol. 58, no. 9, Nov. 2009, pp. 4698-4708.
- [9] S. F. Fard, A. Alimohammad, B. Cockburn, C. Schlegel, "A single FPGA filter-based multipath fading emulator", *IEEE Globecom 2009*.
- [10] G. El Zein, R. Cosquer, J. Guillet, H. Farhat, F. Sagnard, "Characterization and modeling of the MIMO propagation channel: an overview", *European Conference on Wireless Tech.*, Paris, 3-4 Oct. 2005.
- [11] H. Farhat, R. Cosquer, G. El Zein, "On MIMO channel characterization for future wireless communication systems", *4G Mobile & Wireless Comm. Tech.*, River Publishers, Aalborg, Denmark, 2008, pp. 225-233.
- [12] www.xilinx.com.
- [13] ModelSim - Advanced Simulation and Debugging, <http://model.com>.