Design and verification of a testing platform for implementation of software defined radio communications systems

Diseño y verificación de una plataforma de prueba para la implementación de sistemas de comunicación basados en radio definido por software

GARCÍA-DZUL, Rogelio†*, PARRA-MICHEL, Ramón and SÁNCHEZ-VENEGAS, Jorge

CINVESTAV del IPN, Unidad Guadalajara, Laboratorio de Telecomunicaciones

ID 1st Author: *Rogelio, García-Dzul* / **ORC ID:** 0000-0002-2334-3060

ID 1st Coauthor: *Ramón, Parra-Michel* / **ORC ID:** 0000-0003-2327-2482

ID 2nd Coauthor: *Jorge, Sanchez-Venegas* / **ORC ID:** 0000-0001-5342-3400

DOI: 10.35429/JCSI.2019.16.5.14.19 Received July 25, 2019; Accepted September 05, 2019

Abstract

The software-defined radio platforms that currently exist are mainly focused on the development of software communication blocks, leaving aside the hardware development, so the need arises to have a testing platform for hardware blocks which also allows us to interact with a radio frequency stage to perform blocks tests of complete communication systems in the air such as WiFi, Bluethoot, Zig Bee among others. This article describes the design of a platform based on a Field-Programmable Gate Array (FPGA) technology and microprocessor, which allows having multiple blocks and algorithms in hardware and software and interacting between them. In the same way there is a significant number of general purpose interfaces for testing the signals, as well as having specific interfaces which allow us to interconnect with different radio frequency platforms and communication cards with the computer. This allows you to work with different radio technologies according to your needs. A use case is presented in which a Bluethoot transmitter is tested with the use of the platform.

Software defined radio, FPGA platform

Resumen

Las plataformas de radio definido por software que existen actualmente se centran principalmente en el desarrollo de bloques de comunicaciones mediante software dejando de lado el desarrollo de hardware, por lo que surge la necesidad de contar con una plataforma de prueba para bloques de hardware la cual además nos permita la interacción con una etapa de radio frecuencia para realizar pruebas de bloques de sistemas de comunicaciones completos en el aire tales como WiFi, Bluethoot, Zig Bee entre otros. En este artículo se describe el diseño de una plataforma basada en la tecnología de matriz de puertas programables (FPGA, por sus siglas en inglés) y microprocesador, la cual permite tener multiples bloques y algoritmos en hardware y software e interactuar entre estos. De igual manera se cuenta con un número significativo de interfaces de propósito general para prueba de las señales, al igual que tiene interfaces específicas las cuales nos permiten la interconexión con diferentes plataformas de radio frecuencia y tarjetas de comunicación con la computadora. Esto permite trabajar con diferentes tecnologías de radio según sean las necesidades. Se presenta un caso de uso en el cual se prueba un transmisor de Bluethoot con el uso de la plataforma.

Radio definido por software, plataforma FPGA

Citation: GARCÍA-DZUL, Rogelio, PARRA-MICHEL, Ramón and SÁNCHEZ-VENEGAS, Jorge. Design and verification of a testing platform for implementation of software defined radio communications systems. Journal of Computational Systems and ICTs. 2019, 5-16: 14-19.

* Correspondence to Author (email: rjgarcia@gdl.cinvestav.mx)

† Researcher contributing first author.

1. Introduction

Communication systems are made up of different stages such as coding, modulation, demodulation, and decoding. From that base have emerged diferents wireless standards such as WiFi, Bluetooth, ZigBee, LTE and others, each of these include different architectures and algorithms.

The need for implementation and testing of different communication blocks to be able to have specific communication systems such as those mentioned above has given rise to the emergence of new tools and technologies such as software defined radio (SDR).

The SDR technology allows us to make significant changes in the hardware through changes through the software, this gives a significant advantage in terms of costs and ease of operation and implementation of different wireless communication systems.

The implementation of algorithms, modulators, complete standards and in general blocks of communications can be done through software and/or hardware, the latter through hardware description languages such as Verilog.

The software-defined radio platforms that currently exist focus mainly on the development of communication blocks in software and in the case of having some tools for implementation in hardware such as a FPGA, the end user has limited access to this.

The implementation of blocks in hardware is essential if you want, for example, to have an integrated circuit (IC) as the final producer. This is why there is a need among hardware designers to have a test platform for hardware blocks which provides the user flexibility of operation and interaction between hardware and software, in addition to having general purpose pins for data analysis and in a similar way to perform wireless communication tests.

This paper presents the design of a test platform for the implementation of communication systems in hardware and software and interacting among them.

In the same way, there is a significant number of general purpose interfaces for testing the signals, as well as having specific interfaces that allow us to interconnect with different radio frequency platforms and communication cards with the computer. This allows working with different radio technologies according to the needs of the final user.

In the same way, a case of use of the platform is presented, where it is used to carry out the tests of a Bluetooth transmitter through the implementation of a modulator and a MAC layer of the standard.

2. Design

As mentioned in the previous section, the platform should allow the user flexibility and accessibility to be able to make different configurations in each of the elements, that is why the platform should integrate the sections mentioned below. Figure 1 shows the blocks that integrate the platform, which are described below.

2.1. FPGA

The integration of an FPGA in the platform is of vital importance since it will allow us to implement algorithms and communication blocks through hardware description language. With this element, it will be possible to configure external elements such as the clock generator and the RF section of which will be discussed later. The FPGA included in the platform will be from Intel.

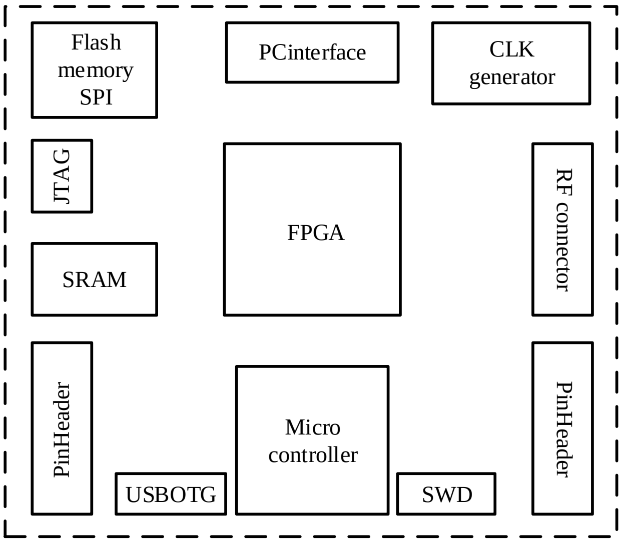


Figure 1 Platform block diagram

To be able to use the FPGA it is necessary to have at least one way of programming it, that is why together with the FPGA two interfaces are included, one of programming through JTAG and the other with programming by means of a programming memory, which allows having a pre-loaded program when the platform is turned on.

To give the user an additional memory to the one contained in the FPGA, an SRAM memory was integrated, which is completely accessible through this same element.

2.2. Microcontroller

An STM microcontroller is included in the platform which will allow us to implement algorithms in programming languages, just as it will have complete communication with the FPGA through general purpose input/output pins; this allows us to have interaction between the algorithms that are implemented in the two different elements of the platform.

In a similar way to the FPGA it is necessary to have a programming interface for the microcontroller, additionally, it has a USB OTG connection, which allows the interaction of the microcontroller with the computer.

2.3. Clock generator

The radio frequency (RF) card to which the platform is adapted requires different clock frequencies for the operation of the transceiver, as well as to be able to make use of different sampling rates according to the need of the user. For this, a clock generator is included in the platform, which, from an input clock, generates 8 independent clock outputs that can be adjusted between a range of 8 KHz to 160 MHz.

To adjust the output frequencies, it is necessary to configure the clock generator, which is done by sending values to the internal registers of the chip through the Inter-Integrated Circuit (I2C) protocol. This configuration can be done through the FPGA or the microcontroller, with the FPGA it is necessary to have a block in hardware that allows to receive the configuration data and send them through the protocol which will be discussed later.

2.4. Interfaces

The platform has input/output pins that are directly connected to the FPGA, these allow us to have a visualization of desired signals, data entry, connection with other devices, as well as the interconnection between two or more test platforms.

One of the fundamental requirements in having an interface for the connection of a commercial RF card, that is why an RF connector was included which is adapted to the Myriad commercial card, which includes an LMS6002D transceiver. This transceiver performs its configuration by writing data in its registers through the Serial Peripheral Interface (SPI) protocol. The SPI block will be implemented within the FPGA and will be discussed later.

3. Development

It is necessary to be able to configure the clock generator and the RF stage, which is why the I2C and SPI blocks were implemented, which allow us to receive the data from the computer and send them to the corresponding configuration registers.

3.1. IP-SPI

The IP-SPI block has connections for a communication interface with the computer, as well as the connections that go to the radio interface which correspond to an SPI connection.

This block will allow us to write the desired configuration in the LMS6002D chip registers in order to obtain, for example, the desired transmission and/or reception frequency, bandwidth, gains of the amplifiers, among other configurations. In the same way, it allows us to read the values of the registers to be able to see the current configurations and perform the necessary calibrations within the transceiver.

The configuration registers of the LMS6002D are divided into 8 logical blocks. The write/read are made through 16 cycles of SPI clock with which the same number of bits is sent, the most significant bit corresponds to a read or writes command indicating it with a 0 or 1.

Of the remaining 15 bits, the 3 more significant is used to indicate which logical block to configure and the next 4 bits indicate a particular address within the block.

The remaining 8 bits are the data to be written in the register chosen in the case of the writings and in the readings only the clock cycles are used to read the register value and send it to the SPI block. Table 1 shows the logical blocks of the LMS6002D.

Address	Description
000:XXXX	Top level
001:XXXX	TX PLL
010:XXXX	RX PLL
011:XXXX	TX LPF
100:XXXX	TX RF
101:XXXX	RX LPF, DACs and ADCs
110:XXXX	RX VGA2
111:XXXX	RX RF

Table 1 Memory map SPI LMS6002D

3.2. IP-I2C

As mentioned above, this block will allow us to configure the Si5351C clock generator of Silabs in order to give the desired frequency to each of the elements of the platform. Similar to the IP-SPI, this block includes a connection to a communication interface with the computer, but with the diference in the connection on the other side of the block corresponds to an I2C interface.

The Si5351C supports read and write operations for its 255 registers, this through the corresponding I2C commands. This IC supports writes and reads of a single record, as well as reading of the record in burst mode, that is, reading or writing in a certain number of consecutive records, since the address auto-increment.

The configurations available for this IC through this block are read the status indicator, set the multiplication and division values to obtain the desired frequency, enable or disable each of the clock outputs, and compensation of the exit phase.

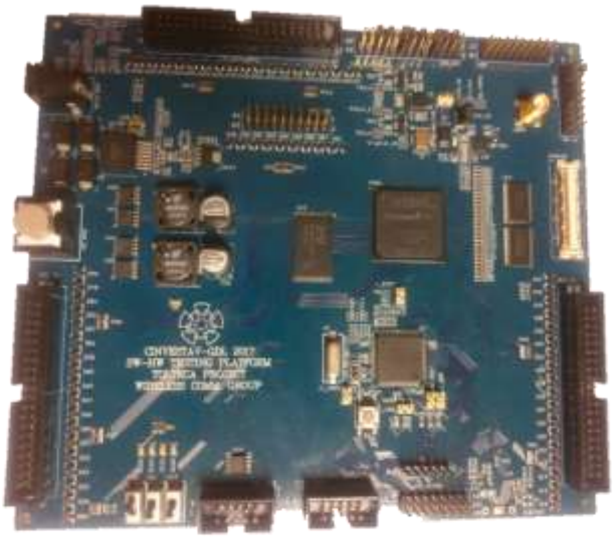


Figure 2 SW-HW Testing platform

Given the connections between the FPGA and the microcontroller, it is important to mention that the configuration of the aforementioned elements is not exclusive of the FPGA since they can be done in the same way by means of the microcontroller.

4. Case of use

The blocks used to perform the tests are shown in Figure 3, it is noted that in addition to the Bluetooth block the I2C and SPI configuration blocks are necessary.

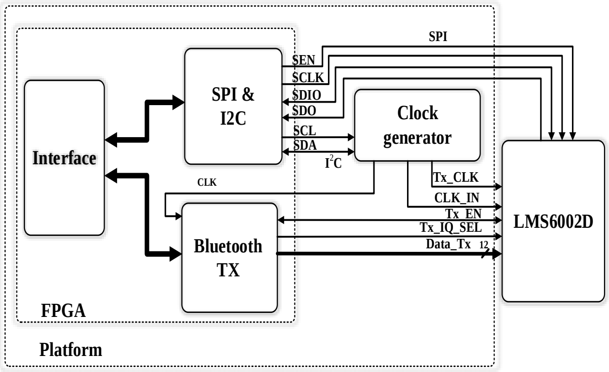


Figure 3 Block diagram for the BLE transmitter tests

A use case is presented, which consists in the use of the platform to test a physical architecture and MAC layer of a reconfigurable Bluetooth transmitter developed by JM Sánchez-Venegas whose detailed architecture can be seen in (Sánchez-Venegas, 2018).

Figure 4 shows the blocks that make up the architecture of Bluetooth transmitter.

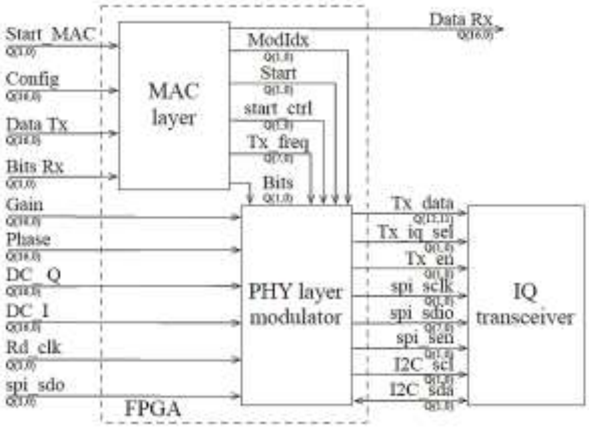


Figure 4 Architecture overview of PHY and MAC layers of all-digital Bluetooth transmitter
Source: (Sánches-Venegas, 2018)

The physical layer of the modulator corresponds to the architecture of a GFSK modulator, which is completely digital and is based on a CPM modulator. This transmitter is capable of generating any CPM scheme with an arbitrary index, but for this example, it is adapting to comply with the Bluetooth 4.0 standard.

The architecture of the MAC layer has two interfaces, the first to communicate with a microprocessor, through which the module is configured and sends the payload of the packets. The second interface is for communication with the modulator/demodulator. The MAC layer is responsible for sending a bit stream of a defined packet to the modulator, which contains the device information such as the MAC address and the device name. Similarly, it receives the bits from the demodulator and processes them. The use of the platform in this test is shows in the Figure 5.

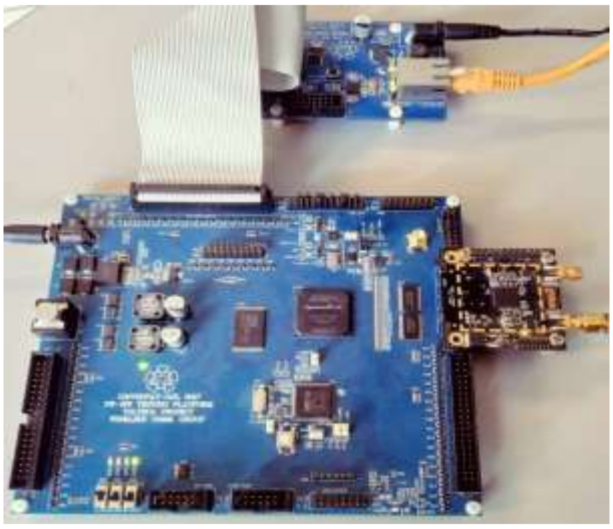
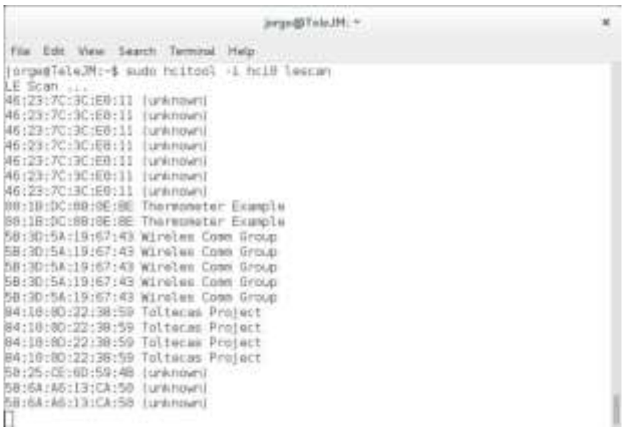


Figure 5 Transmitter testing

To carry out the tests, a commercial Bluetooth dongle receiver was used, which is configured in the listening mode of the Bluetooth LE advertising packages; to make use of this device it is necessary to have a computer with Linux operating system to be able to execute the corresponding commands. Figure 6 shows the results of the tests performed.



```
jorge@TeleJM:~$ sudo hcitool -i hci0 lescan
LE Scan ...
46:23:7C:3C:E0:11 (unknown)
46:23:7C:3C:E0:11 (unknown)
46:23:7C:3C:E0:11 (unknown)
46:23:7C:3C:E0:11 (unknown)
46:23:7C:3C:E0:11 (unknown)
46:23:7C:3C:E0:11 (unknown)
46:23:7C:3C:E0:11 (unknown)
88:1B:DC:88:8E:8E Thermometer Example
88:1B:DC:88:8E:8E Thermometer Example
58:3D:5A:19:67:43 Wireles Comm Group
58:3D:5A:19:67:43 Wireles Comm Group
58:3D:5A:19:67:43 Wireles Comm Group
58:3D:5A:19:67:43 Wireles Comm Group
58:3D:5A:19:67:43 Wireles Comm Group
84:18:8D:22:38:59 Toltecas Project
84:18:8D:22:38:59 Toltecas Project
84:18:8D:22:38:59 Toltecas Project
84:18:8D:22:38:59 Toltecas Project
58:25:CE:6D:59:4B (unknown)
58:6A:A6:13:CA:58 (unknown)
58:6A:A6:13:CA:58 (unknown)
```

Figure 6 Decoded advertising LE 1M PHY packet on commercial Bluetooth dongle.
Source: (Sánches-Venegas, 2018)

5. Conclusions

In this work a platform was developed that allows the easy prototyping and testing of blocks of hardware and software oriented to wireless communication systems and communications in general.

Being able to make the configurations of the elements that make up the platform through the FPGA or the microcontroller gives the user versatility and likewise the opportunity to have available the full capacity in available logic elements of the FPGA, which is a problem in the SDR platforms from the market.

Another advantage with respect to commercial SDR platforms is the possibility of being able to carry out tests with different transceivers given the connection with the platform, which allows you to test your own RF card designs.

6. Acknowledgements

The present work was supported by ANR-CONACYT TOLTECA project No. 273562.

7. References

Burns, P. (2003). Software Defined Radio for 3G. Artech House, Inc.

Harada, H., & Prasad, R. (2002). Simulation and software radio for mobile communications. Artech House.

Mazover, V. K. (2015). Fundamentos de comunicaciones digitales. Limusa.

Microsystems, L. (2019). LMS6002D Datasheet. Retrieved from Lime Microsystems: www.limemicro.com

Sánchez-Venegas, J. M., Ramírez-Pérez, A., Jaramillo-Ramírez, R., & Parra-Michel, R. (2018). An all-digital physical and MAC layer architectutes for a reconfigurable Bluetooth transmitter. 2018 IEEE 9th Latin American Symposium on Circuits & Systems (LASCAS).

Silabs, S. (2019). Silicon Labs. Retrieved from https://www.silabs.com/