OQPSK COGNITIVE MODULATOR FULLY FPGA-IMPLEMENTED VIA DYNAMIC PARTIAL RECONFIGURATION AND RAPID PROTOTYPING TOOLS

Raúl Torrego (Communications department: IKERLAN-IK4, Arrasate-Mondragón, Spain; rtorrego@ikerlan.es); Iñaki Val (Communications department: IKERLAN-IK4, Arrasate-Mondragón, Spain; ival@ikerlan.es); Eñaut Muxika (Signal Theory and Communications Area: University of Mondragon, Arrasate-Mondragón, Spain; emuxika@eps.mondragon.edu)

ABSTRACT

Software Defined Radios (SDR) and Cognitive Radios, their evolution provided with the ability of making decisions about their operating behaviour, have emerged as the key technology for implementing wireless communication systems in the future, due to their versatility and to the size and power consumption save they obtain. This paper presents a small form factor OQPSK modulator able to change its IF (intermediate frequency) working frequency depending on the availability of the transmission channel. It is fully implemented on an FPGA using dynamic partial reconfiguration to achieve the frequency change, the internal ICAP configuration port to access FPGA's configuration memory and an embedded uBlaze processor in charge of managing the whole system. Those parts of the modulator requiring digital signal processing have been carried out with Xilinx's rapid prototyping tool System Generator in order to ease their development.

1. INTRODUCTION

The features users demand from communication devices become more and more exigent every day. And not only in those aspects dealing with communication speed, but also in such different characteristics as different communication standard compatibility, battery life, device size or price. In this context, Software Defined Radios (SDR) [1, 2] and evolutions such as Cognitive Radios or Intelligent Radios [3] seem to be the technological answer that will satisfy all these requirements. SDRs are defined as communication systems that can change their characteristics "on-the-fly". Meanwhile, Cognitive Radios add the ability of making decisions about these changes of characteristics, depending on the environment's status.

On the other hand, FPGAs are becoming one of the favourite devices in order to implement signal processing algorithms due to the high performance and flexibility they provide. A characteristic like dynamic partial reconfiguration, which enables an FPGA to change the functionality of part of it while the rest remains working, makes FPGAs an optimum candidate for Cognitive Radio implementation.

Furthermore, with the aim of easing the design of this type of systems, the use of rapid prototyping tools is highly advisable [4]. These tools, on the one hand, allow the designer to program the required algorithms in a graphical way, generating the synthesizable code automatically. On the other hand, functional simulations are feasible in early design steps, which in turn lead to huge time savings.

One of the incoming applications SDR and Cognitive Radios have is their use in wireless industrial communications. Their ability to adapt to the surrounding environment enables the securing of the reliable communications needed in this kind of applications.

Therefore, this paper presents a basic OQPSK cognitive modulator that demonstrates the feasibility of the above presented design framework. OQPSK is the modulator scheme the IEEE 802.15.4 wireless standard uses. Some industrial wireless standards such as WirelessHART are based on it [5]. The cognitive modulator is fully implemented on an FPGA, using dynamic partial reconfiguration to achieve the IF frequency change, while both the modulator and the spectrum sensing algorithm have been designed and tested with Xilinx's "System Generator" rapid prototyping tool. The bitstream management for the partial reconfiguration and the frequency change decision are carried out by an embedded uBlaze soft-processor while the ADC/DAC converter's control is also implemented in hardware in the FPGA.

The paper is organized as follows. Section 2 presents the data processing algorithms implemented with System Generator. In Section 3 dynamic partial reconfiguration and the necessary hardware to achieve it are presented, while Section 4 analyzes the points to be checked for making the code generated by rapid prototyping tools suitable for partial reconfiguration. In Section 5, the hardware where the cognitive modulator has been implemented is shown.

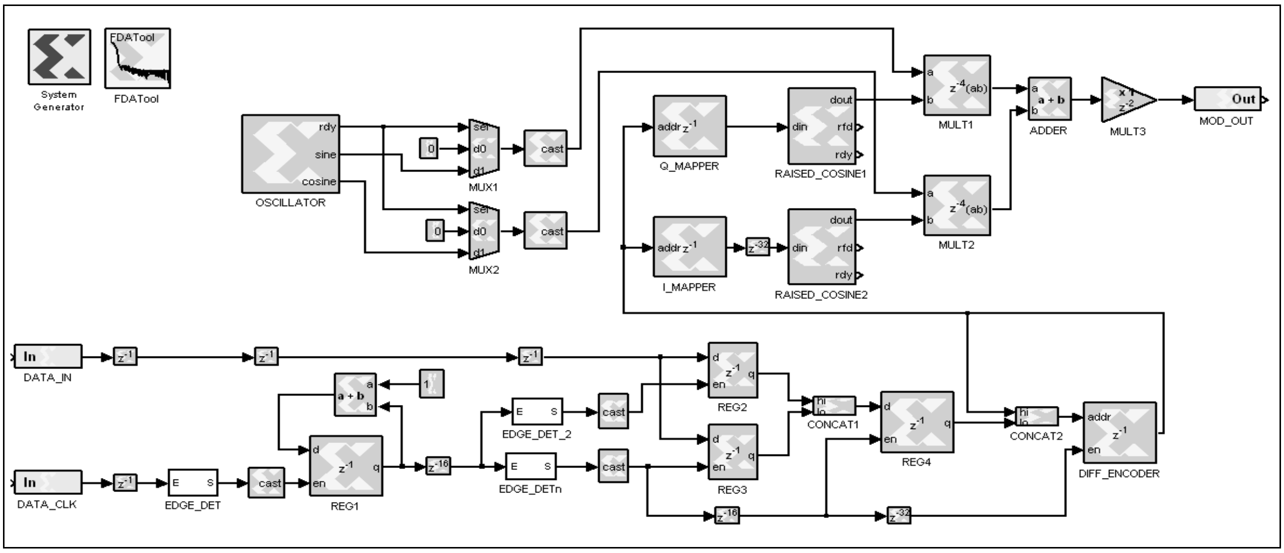


Figure 1: OQPSK Modulator

Measured results are presented in Section 6 while Section 7 contains the conclusion and future work.

2. SYSTEM GENERATOR DESIGN

As stated in the introduction, the use of rapid prototyping tools eases the design of data processing systems [4]. In this design we have used Xilinx's System Generator for this purpose. It is defined as a high-level tool for designing high-performance digital signal processing systems using FPGAs [6]. Working over the Matlab/Simulink design environment, it is possible to exploit all the resources and advantages this environment offers together with System Generator's self-characteristics [7].

This tool mainly offers a graphic interface so that algorithm programming is made interconnecting available predefined blocks. These blocks represent functions or components typically used in signal processing or in the digital logic world, from logic gates to multiplexers, filters or microprocessors. This way, programming is very intuitive and the signals that connect the different blocks represent the real, physical, data paths. Once a design is finished, the tool is able to generate the VHDL code that describes the system, synthesize it, and even, download it directly to the FPGA. Prior to this final implementation, this is, even before the VHDL code is generated, functional simulations can be executed in order to test and tune the algorithm's behaviour. For this purpose, Matlab/Simulink's data sources and analysis tools can be used.

There are two main data processing tasks in the proposed cognitive modulator, the OQPSK modulator itself and the Power Spectral Density (PSD) estimator that scans the transmission channel looking for its availability.

2.1. OQPSK modulator

Figure 1 shows the architecture of the OQPSK modulator designed using System Generator blocks. It works at a 2 Mega bit per second data rate, upconverting it to a reconfigurable Intermediate Frequency (IF) of 5 or 10 MHz depending on the availability of the channel. As will be seen later, this frequency switch is the one carried out by the dynamic partial reconfiguration.

Following the data path from DATA_IN, where data is introduced serially synchronized with DATA_CLK, to MOD_OUT, where it comes out modulated, 4 main processing steps appear. The first step, which nearly comprises the bottom line of the graph, is the "Data Acquisition" step. Here, serial data and external clock are firstly synchronized with the internal clock in order to avoid metastability and then grouped in two bit symbols to be transmitted later. The second step is the "Differential Encoder". In this step each symbol to be transmitted is encoded depending on the last sent one so that phase ambiguity in the receiver is avoided. In the top line of Figure 1, both the "Oscillator" and the "Modulator" are implemented. The "Oscillator" step generates via a Numerically Controlled Oscillator (NCO) the sine and cosine signals needed for the modulation. Finally, in the "Modulation" step, encoded symbols are mapped to the desired constellation, filtered with a 64-tap, 0.25 roll-off, raised-cosine filter, multiplied with the signals generated by the "Oscillator", added and transmitted.

The frequency switch between 5 and 10 MHz is achieved changing the frequency of the sine and cosine signals generated by the "Oscillator". Therefore, two different setting of the NCO have been implemented, one for each frequency.

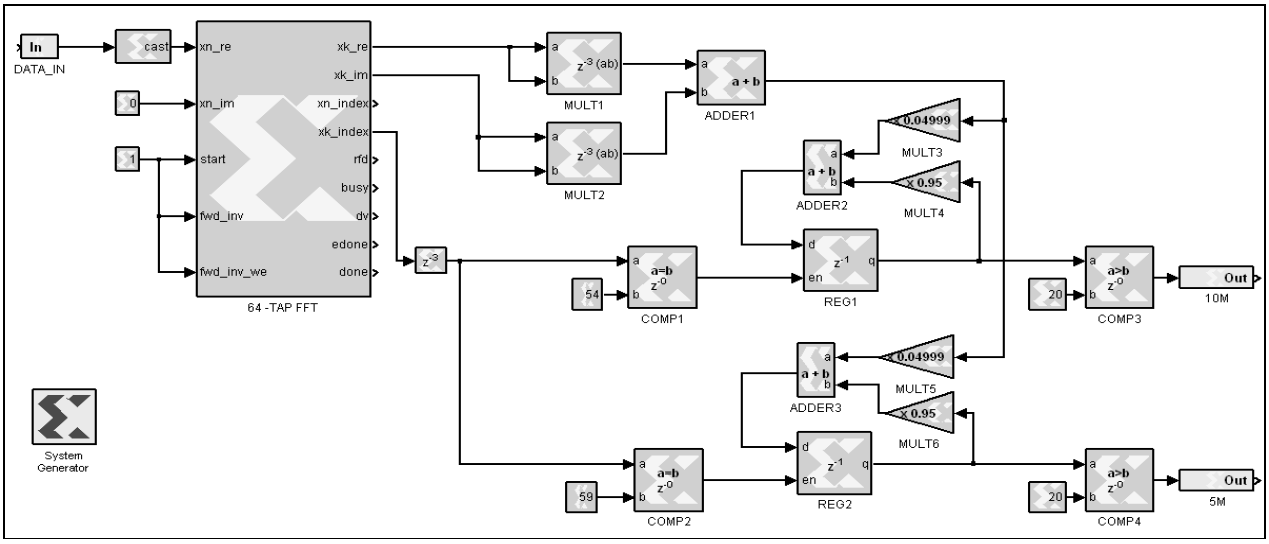


Figure 2: PSD Estimator

2.2. PSD estimator

The second data processing module in the cognitive modulator, the Power Spectral Density (PSD) estimator, can be seen in Figure 2. This module is responsible for determining the availability of the channel to transmit on, based on the power already existing on each frequency. This estimator is implemented in two steps, the “FFT computing” and the “Power Detector”. The “FFT computing” takes the digitalized samples from the Analog to Digital Converter (ADC) working at 61.44 MHz sampling frequency and firstly calculates a 64-tap Fast Fourier Transform (FFT). This operation obtains a real time virtual representation of the available spectrum power distribution on a 64 byte string where each byte covers an approximate bandwidth of 1 MHz. Each byte and therefore each frequency range are refreshed each microsecond and in order to obtain a less noisy representation, the obtained data is averaged. Finally, the “Power Detector” first of all extracts from the averaged string the data points that represent the frequency ranges under observation. FFT block delivers data swapped, being point 63 the one corresponding to DC and point 31 the one corresponding to half the switching frequency [8]. Therefore, in this case, array indexes 54 and 59 are analyzed that correspond to 5 and 10 MHz frequencies. Later it compares the power present on these frequencies with the maximum power threshold the communication can sustain. The results of this comparison point out the availability of each of the channels.

3. PARTIAL RECONFIGURATION: BASICS AND IMPLEMENTATION ARCHITECTURE

Dynamic partial reconfiguration of FPGAs is defined as the possibility of changing the functionality of part of the FPGA while the rest of it continues working [7]. Mainly restricted to SRAM-based FPGAs, this technique takes advantage of the particular structure they have where the downloaded configuration is stored in an internal memory. Rewriting part of this memory on-the-fly, changes the configuration of those elements controlled by this part of the memory without altering the rest of them. On one hand, this allows the static part to manage the reconfiguration of other parts, what is called “Partial self -reconfiguration”. On the other hand, FPGAs are enabled to carry out different functionalities without penalizing its size, reusing a specific area each time a functionality change is needed. Attending to this last possibility, partial reconfiguration appears to be an interesting technology for implementing SDR and Cognitive Radios.

The main disadvantage this technology has is the so called “Reconfiguration time”. While the writing in the configuration memory of the FPGA is happening, the reconfigurable area is disabled. This dead time is dependent on the amount of data to be written on the memory, and therefore on the size of the reconfigurable area, and on the writing speed [9].

In the presented cognitive modulator, the dynamic partial reconfiguration is used in order to change the transmission frequency. As explained in the previous section, this is achieved changing the “Oscillator” of the OQPSK modulator part. However, in order to evaluate the implication of the design size in the reconfigurable time, a second design has been carried out where the whole OQPSK

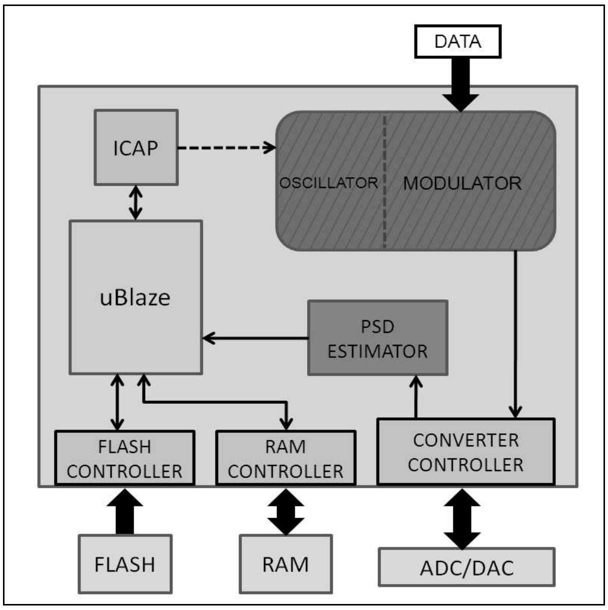


Figure 3: Cognitive Radio architecture

modulator is reconfigured although only the “Oscillator” part changes.

In order to achieve a compact portable implementation where the whole system is implemented on a single FPGA, a self-reconfigurable system has been developed around the already described data processing modules. It consists of a uBlaze processor, an ICAP port (the internal access port to the configuration memory of the FPGA) and the necessary hardware to manage external resources such as memories or ADC/DAC converters. The uBlaze is also connected to the PSD estimator, which gives him the necessary information about channels’ availability. In case a frequency change is needed, the new configuration partial bitstream for the reconfigurable area is downloaded via the ICAP port by uBlaze. These bitstreams are stored on start-up in the FLASH memory, together with the FPGA’s initial configuration bitstream. Once the FPGA is configured for the first time they are copied to the RAM memory which has a faster access time. When working with very small reconfigurable areas and therefore, with small partial bitstreams, they can even be stored in BRAMs so that the access time becomes minimum. In this design, this situation happens when only the oscillator is reconfigured.

The complete architecture can be seen on Figure 3. The striped area is the one holding the modulator and therefore, the one to be reconfigured. The dashed line across it represents the two reconfiguration possibilities: only the oscillator or the whole modulator reconfiguration. As already explained, both the modulator and the PSD estimator have been designed using rapid prototyping tools. Meanwhile, the uBlaze system, comprised of the uBlaze processor, the ICAP and the memory controllers have been developed with Xilinx’s Platform Studio. Finally, the ADC/DAC converter controller has been implemented by hand in pure VHDL code.

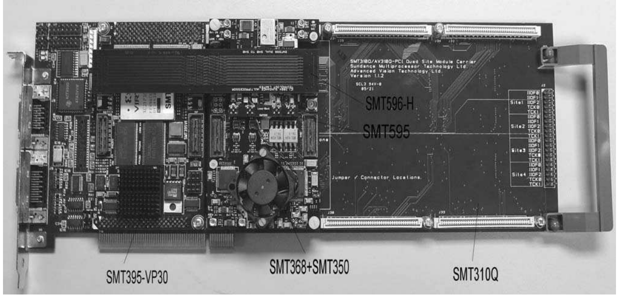


Figure 4: SMT8096

Although the whole system is hardware implemented on the FPGA, there is a small software program running on the uBlaze processor. It is programmed in C and looks after the system’s initialization and partial reconfiguration execution. Once the FPGA is automatically configured on start-up, the initialization task first copies the partial bitstreams to be used before, from FLASH memory to internal BRAM memory o to RAM memory depending on their sizes. Afterward, this task initializes the ICAP access port and starts the reconfiguration loop. This loop reads the information about channel availability from the PSD estimator and evaluates the necessity of a frequency change. In case it is needed, the partial bitstream to be configured is read from the corresponding memory and written to the FPGA internal configuration memory via the ICAP port.

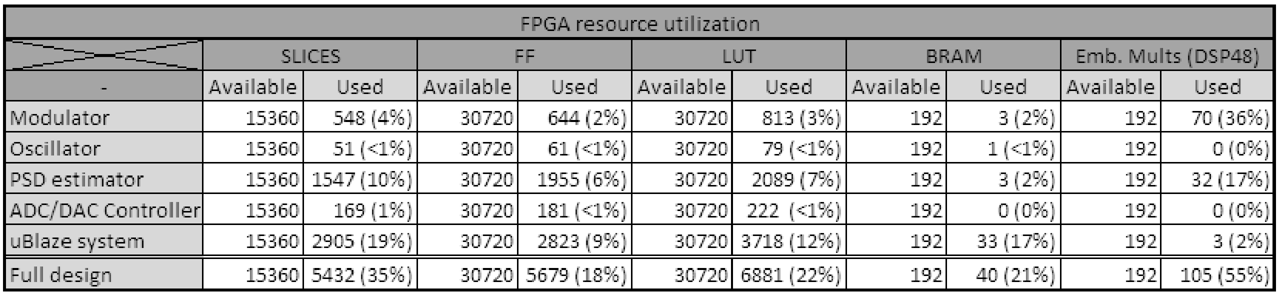
4. PARTIAL RECONFIGURATION IN FPGA RAPID PROTOTYPING TOOLS

Even though both, the rapid prototyping tools and the partial reconfiguration design flow [7], are usually developed by the FPGA manufacturers themselves, their combination is not a standardized procedure. Some precautions must be taken and some steps must be carried out in order to generate suitable VHDL code for partial reconfiguration by the rapid prototyping tools suitable. The whole procedure to be followed is properly explained in [10] but it can be summarized into three main points: extraction of the non-reconfigurable resources to the top entity, maintaining the concordance between the name and ports of the reconfigurable instances and an appropriate management of data exchanges between static and reconfigurable areas.

5. HARDWARE PLATFORM

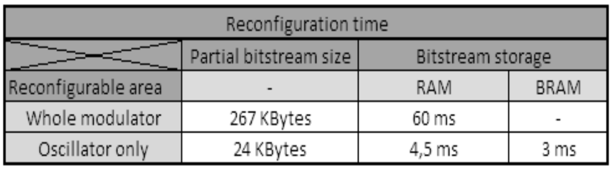
The hardware platform used to implement the OQPSK cognitive modulator was the SMT8096 board [11]. The SMT8096 is a PCI system based on 3 main modules (shown in Figure 4).

Table 1: Resource utilization



FPGA resource utilization										
	SLICES		FF		LUT		BRAM		Emb. Mults (DSP48)	
-	Available	Used	Available	Used	Available	Used	Available	Used	Available	Used
Modulator	15360	548 (4%)	30720	644 (2%)	30720	813 (3%)	192	3 (2%)	192	70 (36%)
Oscillator	15360	51 (<1%)	30720	61 (<1%)	30720	79 (<1%)	192	1 (<1%)	192	0 (0%)
PSD estimator	15360	1547 (10%)	30720	1955 (6%)	30720	2089 (7%)	192	3 (2%)	192	32 (17%)
ADC/DAC Controller	15360	169 (1%)	30720	181 (<1%)	30720	222 (<1%)	192	0 (0%)	192	0 (0%)
uBlaze system	15360	2905 (19%)	30720	2823 (9%)	30720	3718 (12%)	192	33 (17%)	192	3 (2%)
Full design	15360	5432 (35%)	30720	5679 (18%)	30720	6881 (22%)	192	40 (21%)	192	105 (55%)

Table 2: Reconfiguration times



Reconfiguration time			
	Partial bitstream size	Bitstream storage	
Reconfigurable area	-	RAM	BRAM
Whole modulator	267 KBytes	60 ms	-
Oscillator only	24 KBytes	4,5 ms	3 ms

One containing a TI C6416T DSP module (SMT395) combined with a dual high-speed ADC/DAC mezzanine module (SMT350) and an FPGA based module (SMT368), all plugged on a PCI carrier board (SMT310Q). In this project only the FPGA module and the ADC/DAC module have been used.

The FPGA module is based on a Virtex 4-SX (XCV4SX35) FPGA in an FF668 package connected to two banks of 8 Mbytes of high speed ZBTRAM. With the purpose of configuring the FPGA on start-up, and also used in the explained design to store user data, a XCF32 Xilinx Flash PROM is attached to the FPGA.

The board containing the ADC/DAC converters consist of two 14-bit ADCs (TI – ADS5500) sampling at up to 125 MHz, a dual 16-bit DAC (DAC5686) sampling at up to 500 MHz via interpolation and a low-jitter on-board system clock based around the combination of a VCXO and the TI - CDCM7005.

Taking into account the IF and data rate requirements set up for the OQPSK cognitive modulator, and in order to reduce timing issues, both boards have been configured to work with a 61.44 MHz clock. Therefore, also the ADC/DAC converters work at this sampling frequency. The uBlaze system in turn, with his own clock management, works at 100 MHz.

As stated all over the explanation, the output frequency of the cognitive modulator switches between 5 and 10 MHz, this is, intermediate frequency. In order to create a full SDR system working at Radio Frequency (RF) an analog front-end should be attached to the output signal.

6. MEASUREMENTS

Table 1 sums up the resource utilization of each of the parts that make up the cognitive modulator, along with the FPGA occupation of the whole design. Taking the SLICE utilization as an indicator of the size of the design, the presented cognitive modulator occupies around 35% of a general purpose FPGA, demonstrating the small form factor of the designed system in comparison to the potential benefits. Between the data processing functions, the PSD estimator is the main consumer of device resources (10%) due to the FFT function that contains. Overall, the uBlaze system constitutes the biggest part of the design (19%). Being the main function of the uBlaze system the management of the partial reconfiguration, it could be considered an important overhead introduced by it. However, in a more complex system, the uBlaze could be used as a general purpose processor in charge of other software implemented tasks.

In Table 2 a summary of the measured reconfiguration times can be seen. The minimum necessary time needed to achieve the frequency change reaches the 3ms. This time corresponds to the reconfiguration of only the oscillator and with the partial bitstream stored in the internal BRAMs. The same bitstream stored in the external ZBTRAM connected to the FPGA, needs 4.5ms to be downloaded, this is an increase of 50%. Therefore, the partial bitstream management is an important fact when trying to reduce the configuration time. When the whole modulator is reconfigured, the reconfiguration time rises up to 60ms. In this case the partial bitstream can only be stored in the external RAM.

Analysing the reconfiguration speeds, the measured results give a writing speed of 8 MBps with the partial bitstreams stored in BRAM and 4.5 MBps if they are stored in RAM. These speeds are far away from the theoretical maximum speed achievable by the ICAP port of 400 MBps. This is due to the suboptimal implementation of the standard ICAP controller available with the uBlaze system. Designing an optimized controller [9] reconfiguration times below the millisecond for the whole modulator reconfiguration could be achieved.

7. CONCLUSIONS

This paper has presented a cognitive OQPSK modulator fully implemented over an FPGA using dynamic partial reconfiguration. The data processing algorithms have been designed using Xilinx's System Generator rapid prototyping tool, taking advantage of the benefits and resources it offers. The use of a uBlaze embedded processor and the ICAP internal access port makes it possible an implementation in a single device and very fast reconfiguration times.

Although the modulator's characteristics may seem quite basic, the main contribution of this paper is the Cognitive Radio implementation over FPGA partial reconfiguration and rapid prototyping tools, and not the radio features itself. Therefore, the presented implementation is suitable as a proof-of-concept of this design framework.

According to the measurements made, the reconfiguration time is the main drawback the dynamic partial reconfiguration introduces when applied to SDR based systems. Therefore, future work around this subject will consider facing this problem with an optimized implementation of the ICAP controller and with a design methodology that selects the minimum reconfigurable parts in order to improve the overall performance.

8. REFERENCES

[1] J. Mitola, "Software radios-survey, critical evaluation and future directions," in *National Telesystems Conference, 1992. NTC-92.*, 1992, pp. 13/15-13/23.

[2] J. Mitola, "The software radio architecture," *Communications Magazine, IEEE,* vol. 33, pp. 26-38, 1995.

[3] http://www.wirelessinnovation.org/page/Defining_CR_and_DSA

[4] D. Haessig, J. Hwang, S. Gallagher, and M. Uhm, "A case study of Xilinx System Generator design flow for rapid development of SDR waveforms," in *SDR Forum Technical Conference, Orange County*, 2005.

[5] HART Field Communication Protocol Specification, Revision 7.0, 2007.

[6] (XILINX), "System Generator for DSP User Guide," 10.1.1 ed, 2008.

[7] (XILINX), "Early Access Partial Reconfiguration User Guide For ISE 9.2.04i," 2008.

[8] (XILINX), "System Generator for DSP Reference Guide," 10.1.3 ed, 2008.

[9] M. Liu, W. Kuehn, Z. Lu, and A. Jantsch, "Run-time partial reconfiguration speed investigation and architectural design space exploration," in *FPL 09: 19th International Conference on Field Programmable Logic and Applications*, Prague, 2009, pp. 498-502.

[10] R. Torrego, I. Val, E. Muxika, and A. Berrizbeitia, "Partial reconfiguration in FPGA rapid prototyping tools," in *IP-Embedded Systems Conference*, Grenoble (France), 2009.

[11] (SUNDANCE), "SMT8096 User Manual Version 1.2," 2005.