Modular FPGA-Based Software Defined Radio for CubeSats

Steven J. Olivieri†, Jim Aarestad‡, L. Howard Pollard‡, Alexander M. Wyglinski†, Craig Kief♯, and R. Scott Erwin∗

†Wireless Innovation Laboratory, Department of Electrical and Computer Engineering,
Worcester Polytechnic Institute, Worcester, MA 01609-2280, USA

‡Department of Electrical and Computer Engineering,
University of New Mexico, Albuquerque, NM 87131-0001, USA

♯Configurable Space Microsystems Innovations & Applications Center, Albuquerque, NM 87106, USA

∗Air Force Research Laboratory – Space Vehicles Directorate,
Kirtland Air Force Base, Albuquerque, NM 87117, USA

Abstract—**In this paper, we present an adaptive digital communication system using field programmable gate array (FPGA) technology. This system adapts the Universal Software Radio Peripheral (USRP) to better suit the space and power limitations of the CubeSat satellite form factor and the Space Plug-and-Play Avionics (SPA) protocol. The result is a highly-adaptive, plug and play software-defined radio (SDR) that is easily incorporated into any CubeSat design.**

I. Introduction

Traditionally, communication systems required specialized hardware to implement their functionality. In order to keep production costs low, these systems contained only the hardware necessary to perform the tasks that they were designed for. As a result, these systems were often difficult to modify and upgrade. Today, digital signal processor (DSP) technology has evolved to the point where many of the encoders, modulators, filters, and decoders used by these communication systems can be implemented in software [1]. Some systems rely on software to perform small tasks while others implement all of their baseband functionality in software. Those systems that implement all of their baseband functionality in software are called software-defined radios (SDRs) [2].

While some SDRs use general-purpose processors or even digital signal processors, many use a technology called field-programmable gate arrays (FPGA). FPGAs consist primary of reconfigurable logic elements and a switch matrix to route signals between them. These devices can be configured to support simple logic operations, such as addition, or more complex systems, such as digital signal filters. In addition, some FPGAs support a technology called dynamic reconfiguration. Dynamic reconfiguration allows a system to swap components as needed, without any reprogramming.

The combination of SDR and FPGAs provides engineers and developers with a method for updating communication systems that are physically difficult to access, such as satellites. Using FPGA technology, one can design a satellite to support multiple digital signal processing blocks (such as modulation or source coding) and to swap them as needed [1], [3]. Additionally, the firmware on an FPGA can be updated remotely to install new blocks and remove unused blocks, enabling a system to support new protocols while in orbit. These features make FPGA-based SDRs an excellent platform for creating radios for satellites.

A wide variety of SDRs are available today, each optimized for different applications. Lyrtech's small form factor (SFF) SDR development platform and the Wireless Open-Access Research Platform (WARP) both provide powerful FPGA and DSP hardware, but come with high costs [4], [5]. Ettus Research LLC produces the Universal Software Radio Peripheral (USRP) and the USRP2. These platforms are available at significantly lower costs, but contain less powerful hardware. Some SDRs are optimized for use in satellites. Vulcan Wireless Inc. has developed two such systems [3]. These systems support numerous S-Band frequencies (2-4 GHz) and work with a variety of communication protocols and encryptions schemes [6]. However, they do not use open source hardware or software. Therefore, researchers and engineers cannot adapt these systems to their needs, nor can they create derivative solutions from them.

In this paper, we present a new SDR platform with the following characteristics:

1) **Reconfigurability**. The system is easily reconfigured to support any number of encoding, modulation, and other signal processing schemes.
2) **Small size**. The system fits the dimensions for a 1U CubeSat ($10 \times 10 \times 10$ cm) and meets all other CubeSat standard requirements [7].
3) **Plug and play**. The system uses the Space Plug-and-Play Avionics (SPA) protocol for plug and play capabilities.
4) **Open source**. The system uses open source hardware and software so that others can adapt it to fir their project goals and so that they can create derivative systems as needed.

978-1-4577-2053-6/12/$31.00 ©2012 IEEE

The rest of this paper is organized as follows: Section II describes the USRP, its companion software GNU Radio, and the CubeSat small form-factor satellite design; Section III details the proposed CubeSat SDR system; Section IV provides synthesis data to confirm the functionality of the new system; and Section V offers directions for future research.

II. Background

The CubeSat SDR system adapts a popular SDR platform known as the Universal Software Radio Peripheral Platform for use in small form-factor satellites known as CubeSats. This section provides some necessary background information on these technologies.

A. GNU Radio and the USRP

The Universal Software Radio Peripheral Platform consists of the GNU Radio software, the libusrp software libraries, the USRP radio, and a wide variety of RF cards that attach to the USRP. GNU Radio is a free, open source software project, licensed under the GNU General Public License 3 (GPL3), that provides users with tools to prepare data for wireless transmission and to receive data from wireless channels. While the GNU Radio software could work with any SDR hardware in theory, it is ideally suited for use with the USRP. GNU Radio runs on the Linux and Windows operating systems and utilizes general-purpose processors to perform digital signal processing tasks such as modulation, compression, and channel coding.

GNU Radio exchanges data with libusrp, which forms the communication link between a personal comptuer and the USRP with the USB2 protocol. The USRP is also open source (GPL2). The software and hardware schematics are all available online from Ettus Research [8]. At the heart of the USRP is an FPGA that upconverts outgoing data to an intermediate frequency (IF) band and downconverts incoming data through a series of casdaded integrator-comb (CIC) and halfband filters. Supporting the FPGA are digital-to-analog and analog-to-digital converters, a full-featured USB2 controller, and connectors for the RF daughtercards that implement the RF frontends to the system. Figure 1, adapted from [9], provides a high-level view of the USRP's main components.

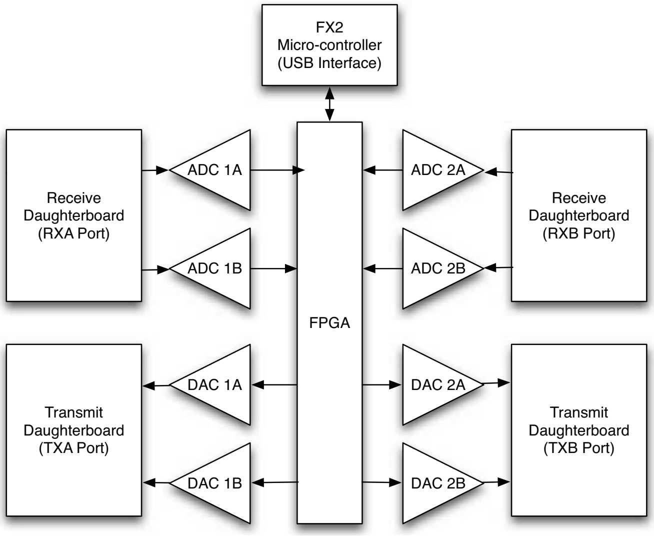
FX2
Micro-controller
(USB Interface)
Receive Daughterboard (RXA Port)
ADC 1A
ADC 1B
FPGA
ADC 2A
ADC 2B
Receive Daughterboard (RXB Port)
Transmit Daughterboard (TXA Port)
DAC 1A
DAC 1B
DAC 2A
DAC 2B
Transmit Daughterboard (TXB Port)

Fig. 1. Block diagram of the USRP. This particular USRP has two Tx boards and two Rx boards, but the system could support four Rx boards instead [9].

B. CubeSats

A CubeSat is a small form-factor satellite conforming to the CubeSat specifications published by California Polytechnic State University [7]. CubeSats are, as their name implies, cube shaped. Each side is 10 cm, giving them a volume of precisely one liter, and the total weight of a CubeSat cannot exceed 1.33 kg. In addition to this 1U configuration, the CubeSat specifications allow for 2U ($10 \times 10 \times 20$ cm) and 3U ($10 \times 10 \times 30$ cm) configurations. The CubeSat SDR was designed to fit into a 1U CubeSat, though it is ideally utilized in a 2U or 3U configuration so that users can the include sensors and other instruments necessary for their satellite's functionality.

The majority of CubeSats are developed at academic institutions in countries all over the world [10]. These CubeSats contain CMOS cameras [11], gamma ray detectors [12], GPS receivers [13], and other scientific instruments and devices. Most of the hardware in these systems consists of is commercially available off-the-shelf (COTS) components, making CubeSats an affordable way to perform experiments in space. However, the majority of CubeSats do not contain interchangeable parts. To counter this problem, the Air Force Research Laboratory's Space Vehicle Directorate (AFRL/RV) developed a system called Space Plug-and-play Avionics (SPA) that introduces the concept of plug and play components to small form-factor satellites. An extension of this system, CubeFlow, applies specifically to CubeSats.

SPA is a set of standards that combines common commercial standards, such as USB and Ethernet, with hardware and software extensions to create communications busses appropriate for use in modern, real-time embedded systems [14]. SPA treats each device in a satellite as a "black box". What distinguishes SPA from other attempts at such a system is that each "black box" is self-describing [14]. CubeFlow is the combination of these standards and the power and mechanical requirements for using SPA devices in a CubeSat.

C. Current Issues

The low cost and open nature of GNU Radio and the USRP make them great candidates for the CubeSat SDR system, but they have some drawbacks. The primary limitation is size. The USRP motherboard is much too large to fit into the CubeSat form factor. While better placement of components could substantially reduce the size of the motherboard, changes to the components themselves might be necessary to fit everything into the small confines of a CubeSat.

A secondary limitation is the FPGA used in the USRP and the language in which its firmware was written. The USRP uses an Altera EP1C12 FPGA. Due to the above-mentioned component changes, the CubeSat SDR required a larger FPGA. The firmware on the USRP's FPGA, written in Verilog, uses many Altera-specific macros. The VHSIC Hardware Description Language (VHDL) is prefered for most military and public sector projects. Further, a better design would not use any vendor-specific macros so that it was as portable as possible.

III. Proposed Implementation

The CubeSat SDR system comprises two main components — the FPGA firmware that implements the filtering described in Section II-A and the new motherboard developed by the Configurable Space Microsystems Innovations & Applications Center (COSMIAC) at the University of New Mexico. The authors were not part of the motherboard development process. Thus, this paper briefly describes the resulting hardware and then focuses on the FPGA firmware part of the system.

A. COSMIAC CubeSat Motherboard

The new motherboard is largely similar to the original USRP motherboard, but it contains some significant differences.

- **Reduced Size**. The new motherboard fits the CubeSat form factor.
- **New Components**. NOR flash memory retains the FPGA configuration through power cycles while the AT90 microcontroller and SPA connectors provide plug-and-play functionality. Other components, such as the DDR2 and NAND flash memories, allow the board to be used for non-SDR purposes.
- **Simplified USB2 Controller**. Due to physical constraints, the Cypress FX2 EZ-USB2 controller is replaced with a USB3300 USB2 controller. This controller provides much less functionality, but has a smaller footprint and requires fewer connections.
- **Xilinx FPGA**. A Xilinx Spartan3A-1400 replaces the Altera EP1C12 FPGA present on the original USRP motherboard. This new FPGA provides more configurable logic elements and embedded memories than the original.

Figure 2 [15] is a photograph of the new COSMIAC motherboard (bottom), its power board (back), and an RF daughtercard based on the USRP's XCVR 2450 transceiver (top) sitting atop one side of a 1U CubeSat.

B. Proposed FPGA Firmware

With the new hardware components described in Section III-A, the software in the CubeSat SDR system requires more functionality. Specifically, the loss of the FX2 USB2 controller requires that the FPGA handle all of the data coming into and leaving the system. The FX2 USB2 controller contained a modern 8051 microprocessor, a full USB2 controller with direct memory access (DMA), RS232 controllers, and controllers for the SPI and I2C buses. The USB3300 chip that replaced it on the CubeSat SDR system contains only a partial USB2 controller.

Figure 3 shows a block diagram of the proposed CubeSat SDR system design. At the center of the system is the MicroBlaze softcore microprocessor. This processor acts as a traffic mediator, guiding data between the USB2 port to the various other devices in the system. The SPI and I2C controllers provide users with a means of sending configuration information to the USRP and to the RF daughtercards. One of the RS232 controllers connects to an serial port on the board; the other connects to the AT90, which controls the ASIM for the device. The other components simply control the peripherals that give them their names (e.g. the DDR2 Memory Controller controls the DDR2 memory).

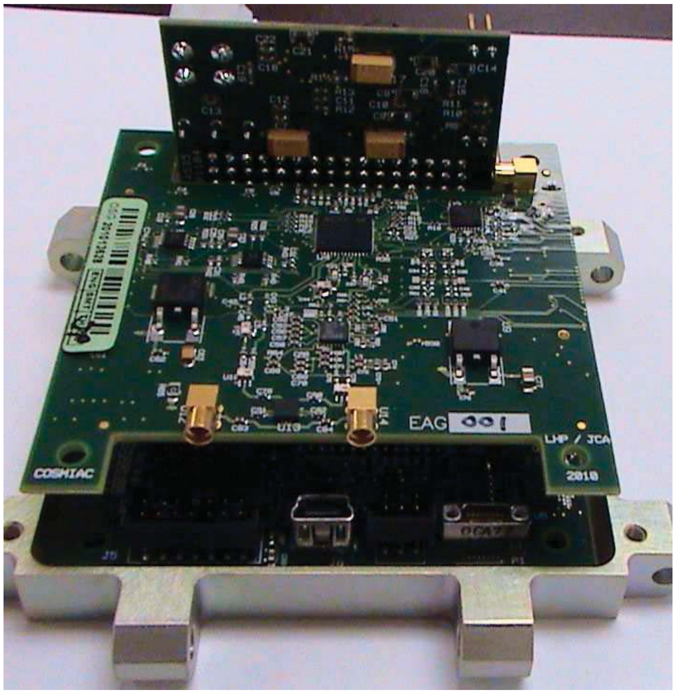

Fig. 2. A photograph of the COSMIAC CubeSat SDR system. The bottom board is the FPGA board while the one on top is the RF daughtercard [15].

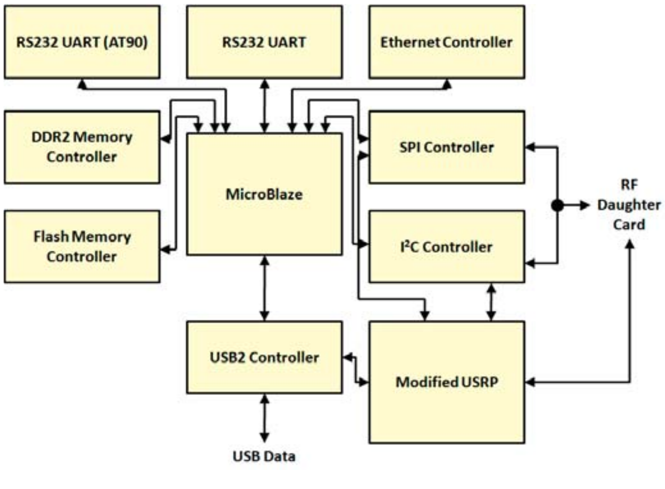

Fig. 3. Block diagram of the proposed CubeSat SDR firmware. The MicroBlaze processor acts as the central control for all of the various communications blocks.

This design has many advantages. First, it uses well-tested components from Xilinx including the DDR2 memory controller, the SPI and I2C controllers, and the MicroBlaze microprocessor. Second, the Embedded Development Kit (EDK) software from Xilinx provides users with a relatively simple way to design MicroBlaze-based systems for custom hardware. Using a MicroBlaze processor to allows one to write much of the control logic in the C programming language. Finally, this system should be compatible with GNU Radio with few modifications since the MicroBlaze and other components can emulate the FX2 USB2 controller with proper configuration.

This proposed system does have two drawbacks. The majority of the components only work with the Xilinx toolset, so the system loses its platform independence. The Altera toolset has similar devices, such as the NIOS II processor, but these are not compatible with the Xilinx toolset. Thus, anyone wish to use the new USRP firmware in a larger system with Altera's tools could not use this proposed system. In addition, the USB2 controller is currently priced at $14,000 [16], a prohibitive cost for smaller organizations. Xilinx does offer free trial licenses, but a full license is necessary to operate the system without the JTAG connection.

C. USRP Variant

As Figure 3 shows, the USRP firmware in the proposed CubeSat SDR system is not the original. Indeed, it has three primary differences. The modified firmware:

- *Works on Xilinx FPGAs*. In fact, this new USRP firmware works on any FPGA large enough to support it, including those from Altera. In addition, there are no proprietary components and so the design should work with any toolset.
- *Is written in VHDL*. The VHSIC Hardware Description Language (VHDL) is a standard for government, military, and other public sector projects. In addition, VHDL is a strongly typed language. As a result, VHDL offer an additional layer of safety in the design phase.
- *Contains extensive documentation*. One of the benefits of open source software is that other parties can modify and extend it to fit their own needs. However, working with another person's code can be very difficult when that code contains little or no documentation. The original USRP firmware contains extensive high-level documentation on various websites, but the actual Verilog contains almost none. The proposed USRP firmware contains extensive documentation in the VHDL files.

This new USRP firmware will provide for easier integration with other government and military applications, will allow future engineers to modify and extend it with ease, and will work on a larger collection of FPGA devices than the original.

IV. Validation

If the two USRP firmwares are logically equivalent, then their hardware usage should be very similar. Table I shows the hardware usage for the USRP firmware. These results were generated with the 64-bit version of Xilinx ISE 12.3 for the Spartan3A-1400FG484-5 FPGA and are the estimated values reported by that software. The Verilog firmware contains some minor modifications from the original to make it compatible with the Xilinx synthesis tools.

TABLE I
Hardware utilization for the USRP firmware.

Logic Type	VHDL		Verilog	
	No. Used	% used	No. Used	% used
Slices	3197	28	3,164	28
Flip Flops	4,068	18	4068	18
4-Input LUTs	5,467	24	5,360	23
I/O Buffers	173	46	173	46
Block RAMs	10	31	10	31
18×18 Multipliers	2	6	2	6
Clock Buffers	3	12	3	12

As Table I shows, the hardware utilization for both versions of the firmware is nearly identical. The VHDL implementation uses 107 more 4-input LUTs than the Verilog version, a difference of less than 2difference is most probably a result of the synthesis tools reading VHDL and Verilog in slightly different ways. The difference in the number of slices used is explained by the difference in the number of 4-input LUTs used. All other hardware utilization is identical for both versions of the firmware, providing evidence that the two are logically equivalent.

V. Conclusion

This paper presented a flexible, plug and play software-defined radio system for CubeSat form-factor satellites. Based on the USRP, this new CubeSat SDR provides CubeSat engineers with an easy to use SDR that is compatible with the GNU Radio software and the Space Plug-and-Play Avionics (SPA) protocol.

Future implementations would do well to eliminate the dependency on GNU Radio. This would allow satellites to use the CubeSat SDR system without including a general purpose processor system. In addition, migration to the Spartan6 FPGA would improve power consumption and make more logic elements and memories available for addition features. Since some Spartan6 FPGAs were designed as drop-in replacements for the Spartan3 devices, this is a potentially simple way to greatly improve the system with little effort.

Acknowledgement

The authors would like to thank the generous support provided by the Air Force Research Laboratory – Space Vehicles Directorate at Kirtland AFB via FA9453-08-2-0259.

References

[1] W. I. Forum, "What is Software Defined Radio." Available from http://data.memberclicks.com/site/sdf/SoftwareDefinedRadio.pdf, April 2011.

[2] G. Youngblood, "A Software-Defined Radio for the Masses, Part 1," *QEX*, vol. Jul/Aug, pp. 13–21, 2002. Available from http://www.arrl.org/files/file/Technology/tis/info/pdf/020708qex013.pdf.

[3] "Vulcan Wireless SDR Products." Available from http://www.vulcanwireless.com/products/, April 2011.

[4] "Mouser — Lyrtech SFF SDR Development Platform." Availabe from http://www.mouser.com/Lyrtech/, April 2011.

[5] “Wireless Open-Access Research Platform.” Available from http://warp.rice.edu/index.php, April 2011.

[6] K. Lynaugh and M. Davis, “Software Defined Radios for Small Satellites.” Presented at the ReSpace/MAPLD 2010 Conference in Albuquerque, New Mexico, November 2010.

[7] California Polytechnic State University, *CubeSat Design Specification Rev. 12*, 2009. Available from http://www.cubesat.org/images/developers/cds_rev12.pdf.

[8] “USRP and Daughtercard Schematics.” Available from http://code.ettus.com/redmine/ettus/projects/public/documents, April 2011.

[9] A. M. Wyglinski, M. Nekovee, and T. Hou, *Cognitive Radio Communications and Networks: Principles and Practice*. Academic Press, 2009.

[10] L. David, “Cubesats: Tiny Spacecraft, Huge Payoffs.” Available from http://www.space.com/308-cubesats-tiny-spacecraft-huge-payoffs.html, 2004.

[11] “University of Tokyo CubeSat Mission.” Available from http://www.space.t.u-tokyo.ac.jp/cubesat/mission/index-e.html, April 2011.

[12] “AAUSAT II CubeSat.” Available from http://aausatii.space.aau.dk/eng/, April 2011.

[13] “CanX-1 CubeSat.” Available from http://www.utias-sfl.net/nanosatellites/CanX1/, April 2011.

[14] J. Lyke, D. Fronterhouse, S. Cannon, D. Lanza, and W. T. Byers, “Space Plug-and-Play Avionics,” in *3rd Responsive Space Conference*, (Los Angeles, CA), AIAA, April 2005.

[15] “COSMIAC CubeSat SDR Photograph.” Available from http://www.cosmiac.org/images/CCRBwSDR.JPG, April 2011.

[16] “Xilinx USB 2.0 Device.” Available from http://www.xilinx.com/products/intellectual-property/xps_usb2_device.htm, April 2011.