An FPGA Spectrum Sensing Accelerator for Cognitive Radio

George Eichinger (george.eichinger@ll.mit.edu)*, Miriam Leeser (mel@coe.neu.edu)**, Kaushik Chowdhury (krc@ece.neu.edu)**
*MIT Lincoln Laboratory, Boston, MA
**Department of Electrical and Computer Engineering, Northeastern University, Boston, MA

Background

Spectrum scarcity is a serious worldwide concern today. The majority of RF spectrum is already allocated and obtaining new blocks of frequency at affordable costs is often infeasible. Interestingly, while these frequencies are indeed licensed, the licensed users or primary users (PUs) do not continuously transmit in their respective bands [1]. This opens up the possibility of opportunistic re-use of the licensed but vacant spectrum, for which new techniques and tools must be devised. The rapidly emerging field of cognitive radio (CR) is one enabling technology that allows secondary or cognitive radio users to transmit when the spectrum is judged to be available, without interfering with the ongoing transmission of the PUs [2]. In figure 1, we show a representative diagram of five PUs and the time / frequency map of the RF transmissions of these PUs. It can be observed that there are “white spaces” where the PUs are not emitting RF energy, which we call spectrum holes.

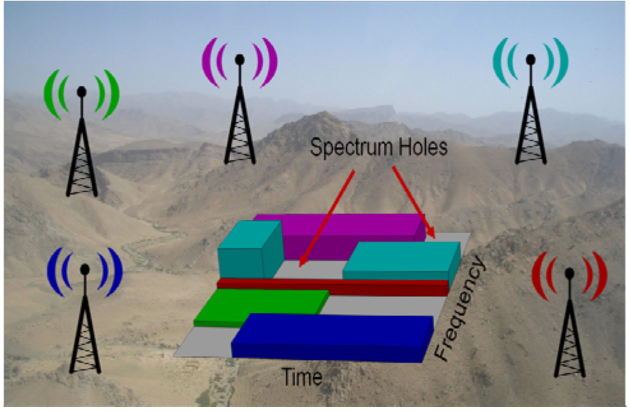

Figure 1: Spectrum holes concept

An interesting and challenging responsibility of a CR is locating and cataloging PUs. The CR has to locate the PUs in the spectrum space so that it can determine free spectrum space for it to operate. Currently, the testbed setup at Northeastern University is composed of Ettus Research Universal Software Radio Peripheral (USRP) family of devices, which gathers millions of sample points of the RF spectrum through its flexible front-end transceiver. The data is sent to a host computer where an FFT is performed to determine the portions of the free spectrum space. This part of the CR process is called *spectrum sensing* because the algorithm senses which frequency bands, and at a finer granularity, specific upper and lower limits of the frequency within these bands that are unusable for CR operation owing to the presence of PUs. A key aim of this project is to migrate the entire spectrum sensing functionality from the host to FPGA hardware directly interfaced with the USRP, so that fast detection of the spectrum holes becomes possible. We will perform all of the analysis in hardware on a Xilinx ML605 FPGA Development Board, which will greatly improve the speed and efficiency of the CR design.

Hardware Overview

The system is shown in figure 2 and consists of a USRP N210 software defined radio, a Xilinx ML605 Development Board and a custom PCB to interface between the two. The basis for the project is the USRP, which is capable of implementing various radio schemes via software on a host computer. The device has a 100 MSPS ADC and a 400 MSPS DAC. The RF portion of the device is implemented separately and there are several that can be utilized depending on what part of the RF spectrum is needed. All of the logic and control of the system is programmed on a Xilinx Spartan 3A-DSP3400 FPGA. This device talks to the host computer over Ethernet and also controls the daughter board. The daughter board that is used in this project is the WBX which can output frequencies anywhere between 50 MHz and 2.2 GHz. The USRP is very capable but if there is one flaw it is the FPGA. The Xilinx Spartan series is a budget FPGA that is not designed to function at high speeds and lacks some advanced FGPA features such as built in DSP blocks. It also has less RAM and DSP blocks that are essential to many different DSP algorithms such as FFTs. Another issue is the size, complexity and timing closure of the baseline firmware. Because the design is complex, it is hard for a user to make simple changes without understanding the rest of the system. Additionally, since the design is on the border of timing stability, minor changes can cause the design to not meet timing closure. Overall modification of the existing firmware is possible but the design is very mature and complex which makes modification difficult.

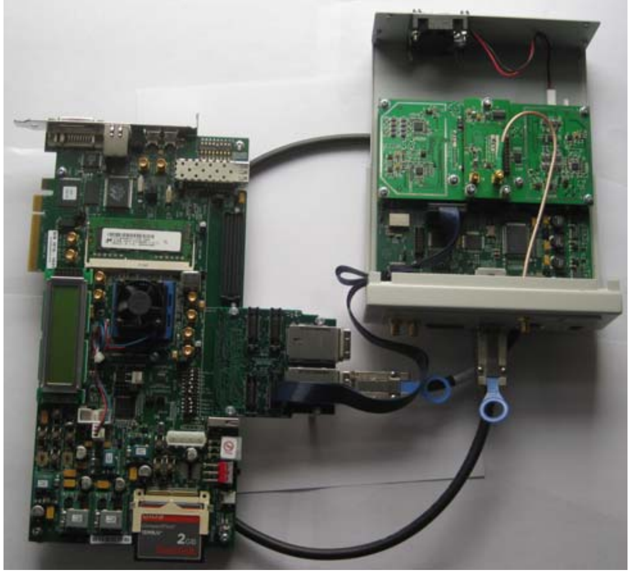

Figure 2: ML605 – Custom PCB – USRP N210

In order to overcome the issues of the USRP we added a second FPGA board to the project to handle the signal

* This work is sponsored by the Department of the Air Force under Air Force Contract FA8721-05-C-0002. The opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

processing requirements. The ML605 has a Xilinx V6 LX240T FPGA which has 6.6x more Block RAM, 4.49x more LUTs and the baseline logic blocks can run ~2.5x faster when compared to the FPGA on the USRP. The ML605 board is a blank slate for the firmware developer and is not cluttered with an existing complex design like the USRP. An added benefit is the time it takes to synthesize designs. For example, the USRP firmware takes a modern computer approximately 45 minutes to fully synthesize and the completed ML605 project takes just 15 minutes.

The physical interface between the ML605 and the USRP is the debug port on the USRP. The manufacturer of the device intended the user to connect this debug port to a logic analyzer and then output various signals for debugging. Because of this, they designed the debug port using a Mictor connector. Mictor connectors are very compact and can handle high speed data. In order to connect to this cable we created the custom PCB shown in figure 2. This PCB is capable of communicating with two USRPs via the Mictor cable and also supports the MIMO port on the USRP. The entire setup is designed for as minimal impact on the USRP as possible. The USRP has pull-ups on the enable lines so that if the ML605 is not connected, the device acts independently. Additionally, the modified USRP firmware forks the ADC data and forwards it to the ML605. The original data path is intact and the USRP can be used with or without the ML605 with no impact on the end user.

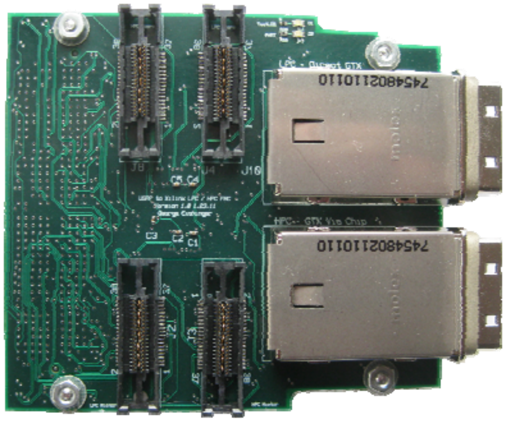

Figure 2: Custom PCB

System Design

Now that the hardware is in place to communicate between the USRP and the ML605, it is time to discuss the algorithm. In this project the ML605 is being used as a spectrum sensing accelerator. In the original algorithm the ADC data was sent over Ethernet to the host computer where it would be processed, FFTed, thresholded and then they would switch frequencies. This method is too slow and did not allow for real time operation. In this algorithm we added the ML605 as an accelerator. A block diagram of the system is shown in figure 3. The data from the USRP ADC is forked and sent to the ML605. The data then goes through a clock crossing FIFO to align itself to the FPGA. The data then flows through a streaming FFT. The outputs of the FFT are compared to a threshold value and each bin is determined to either be above a threshold, ‘1’, or below a threshold, ‘0’. These data are then stored in a memory based on its bin number. Additionally, it is possible to perform multiple FFTs of the same frequency data and average them for better results. You can also perform a finer grain FFT than required and average over multiple bins for a higher probability of detection. The entire process can be controlled and reconfigured from the host computer. This includes the length of the FFT, the averaging, the thresholding value and the scaling.

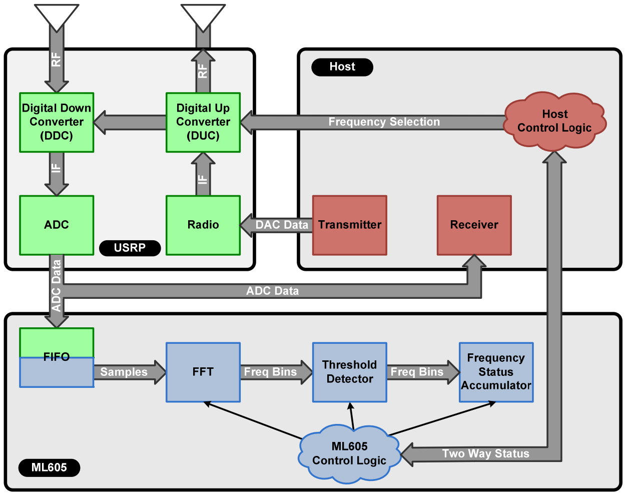

Figure 3: Flow diagram of system design

When integrated into the full system, the host computer will set all of the desired parameters before starting its CR functionality. It will then command the USRP to a specific frequency. Then it will tell the ML605 the threshold value for that frequency range. The ML605 will immediately start performing the spectrum sensing algorithm with the parameters given earlier. Once done, it will respond to the Host with one or more words with each bit representing whether a PU was found in that region or not. The host can then change the frequency to perform more sensing or move on to the transmit portion of the algorithm.

Future Work

Currently, the system is commanded by the host to perform a spectrum sensing function to improve the speed of the algorithm, after appropriately tuning the front-end to a specific frequency. The next step will be to control the changing of the frequencies automatically from the ML605. This will dramatically speed up the algorithm and will allow more bandwidth to be analyzed. Additionally, an FFT is a basic way to perform spectrum sensing and we plan to implement more advanced methods such as wavelet matching detection. Finally, one could use this as a platform for research other than spectrum sensing. This platform is a robust front end combined with powerful hardware and software resources for a relatively low cost. Additional MAC functionality of a CR could be integrated in hardware, leading to novel “split- MAC” designs, with time critical functionalities of the MAC performed on the FPGA, and policy decisions undertaken by the host.

References

[1] J. Yang, “Spatial channel characterization for cognitive radios,” MS Thesis, UC Berkeley, 2004.

[2] I. F. Akyildiz, W-Y. Lee, and K. R. Chowdhury, “CRAHNs: Cognitive Radio Ad Hoc Networks,” *Ad Hoc Networks Journal*, (Elsevier), vol.7, no.5, pp. 810-836, July 2009.