HARDWARE-IN-THE-LOOP DESIGN VERIFICATION TESTING FOR SOFTWARE-DEFINED RADIO WAVEFORMS

Ying Niu (Raytheon, Fort Wayne, IN; Ying.Niu@raytheon.com);
David Mussmann (Raytheon, Fort Wayne, IN; David_E_Mussmann@raytheon.com);

ABSTRACT

This paper describes a hardware-in-the-loop (HWIL) test environment for a software-defined radio (SDR) wideband radio waveform implemented in an FPGA. This environment executes the radio waveform on a COTS board that fits in a PCI slot in a PC backplane. PC software executes the waveform and compares the implemented waveform performance (BER, PER, packet synchronization, etc) against theoretical results. Graphical user interface permits the user to select functions to be tested and allow the user to swap firmware and software implementations of functions.

Compared to waveform testing in target hardware, this environment expedites design verification testing. The environment enables design verification testing for complex waveforms over the complete range of waveform modes. In this environment, the user can calibrate digital baseband samples or digital modem IF samples for range delay and attenuation, fading, Doppler, and phase rotations. The environment also provides the ability to measure FEC (forward error correction) performance. Additionally, the environment also provides a platform for functional testing outside the target hardware.

1. INTRODUCTION

Software Defined Radios (SDRs) offer greater flexibility and adaptability as they replace standard analog radio components with digital processing hardware and software. However, to implement waveforms in software requires computationally intense algorithms error correction, digital filters and adaptive equalizers. In addition, shorter design cycles require parallel development of hardware and software. To uncover waveform vulnerabilities and correct them rapidly through software updates, we introduce the hardware-in-the-loop (HWIL) waveform test environment.

In our waveform test environment (WTE), the design is validated and verified through simulation and hardware prototyping. End-to-end testing is accomplished at the FPGA level prior to integration with target hardware and software. Changes made at this stage feed back into the waveform development process to improve the implementation before final development of the SDR. This aids the implemented wideband waveform in meeting design requirements and establishes performance baselines.

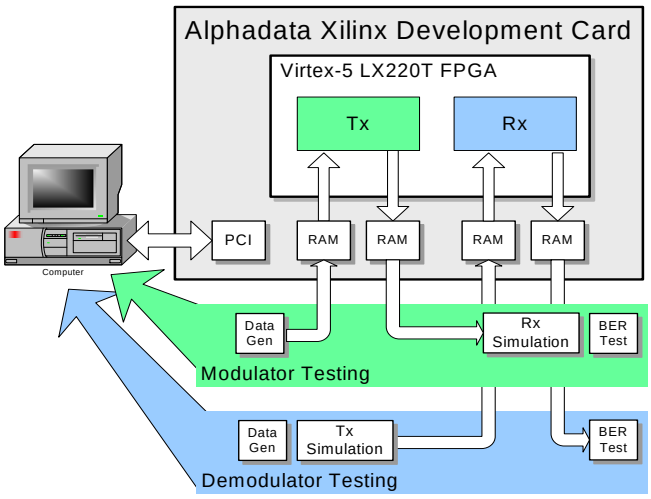

Figure 1: modulator and demodulator testing in the hardware-in-the-loop test environment.

2. TEST GENERATION

This section describes the hardware-in-the-loop test environment for waveforms in software-defined radios. In our case we want to thoroughly test the wideband waveform processing at the FPGA level prior to hardware and software integration.

The test environment is setup using a commercial off the shelf (COTS) FPGA development card plugged through a PCI connection to a computer. A Virtex 5-LX220 FPGA embedded on an AlphaData Xilinx development COTS hardware enables a more efficient system by reducing the hardware development and maintenance time. A custom GUI interface (using AlphaData APIs) from the COTS card permits the user to select the waveform processing in the test environment for transmit or receive path tests and also allows the user to swap firmware and software implementations of the tests. This flexibility provided by the COTS card enabled us to develop specific functional

tests such as the modulator, demodulator and end to end testing.

Figure 1 shows a block diagram of the development card for the modulator and demodulator testing done with the test environment.

For the modulator test, user data from the emulated processor along with status and packet configuration information are stored into four pages of memory in the RAM of the development card. The user executes a packet by writing to a specified register of the FPGA at the WTE GUI interface.

Upon execution, the FPGA processes the transmit path test for the wideband waveform and writes baseband or IF sample results back to the RAM off-chip memory bank. These results processed by the FPGA are the data that would be sent to the DAC interface. For our testing purposes, the baseband or IF samples processed by the FPGA are inputs to the C++ simulation of waveform processing for receive path testing.

Similarly, the demodulator testing path begins with software simulation of the transmit path of the wideband waveform. The baseband or IF sample results from the simulation are placed into the memory bank of the RAM in the COTS card. These samples emulate the data that would be received by the FPGA at the ADC interface. The FPGA processes the receive path test for wideband waveform through de-interleaving, down conversion, synchronization, and demodulation algorithms before data is decoded and stored back to the RAM memory bank. Results are extracted from the RAM for further analysis.

In both cases of modulator and demodulator testing, the VHDL is integrated and tested with C++ simulation in order to isolate wideband waveform processing of the FPGA for transmit or receive path tests.

Using the same test environment, Loopback End-to-End Testing can be performed as shown in Figure 2. The loopback end to end testing begins by storing the user data, status, and configuration information passed from the DSP to the RAM memory banks. As with the modulator test, the FPGA conducts waveform processing for transmit path test and writes baseband or IF samples to the memory bank. Adding calibrated noise input to the baseband samples at the memory bank, the FPGA runs the waveform processing for receive path test and writes the results back to the RAM for further bit error rate (BER) and systems analyses.

Different noise calibration of digital baseband samples or digital modem IF samples are introduced for various communication environments. These data calibrations include attenuation, Doppler, propagation range delay, fading, and phase rotations. We analyze the results of end-to-end testing for the different types of noise calibration inputs in order to rigorously test waveform implementation and ensure the waveform meets specified design requirements.

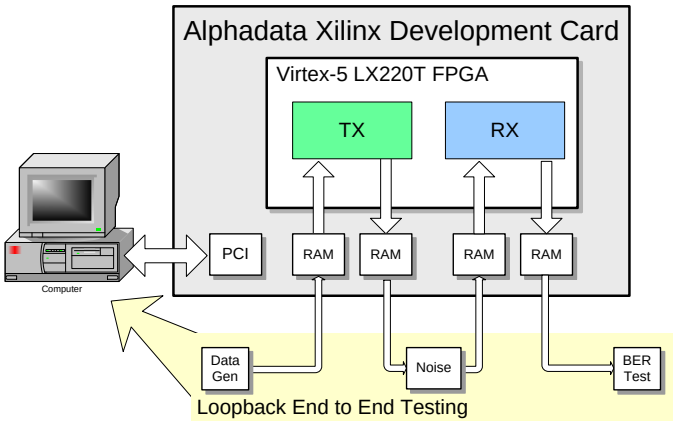

Figure 2: Loopback End-to-End Testing using the waveform test environment

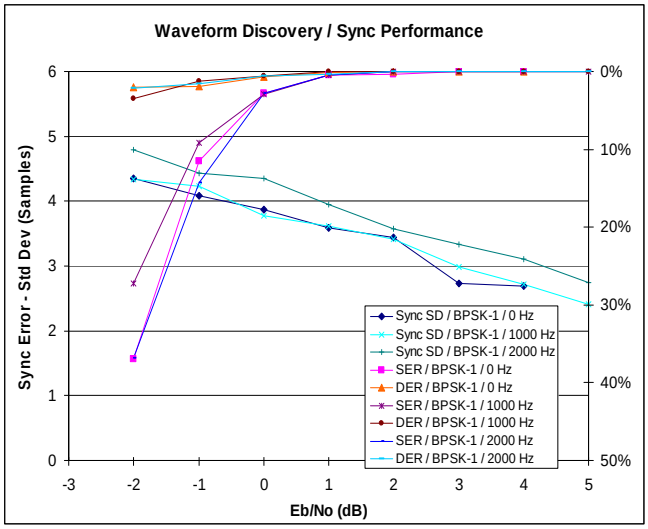

Figure 3: waveform discovery and sync performance

3. WAVEFORM PERFORMANCE AND DATA RESULTS

Output data are stored in the RAM memory bank from end-to-end loopback test of the waveform environment. Plots of the sync, BER, and spectrums were obtained for various packet configuration and slots. The goal of the analysis was to measure the performance of our implemented wideband waveform against simulated and theoretical results. We also conduct thorough testing of the waveform for design verification testing.

In Figure 3, synchronization is analyzed by plotting sync error against signal-noise-ratio for 1-slot BPSK packets. The sync performance follows expected data trends: for increasing signal-to-noise ratio, sync error decreases. Similarly, the packet error rate decreases for increasing signal to noise ratio.

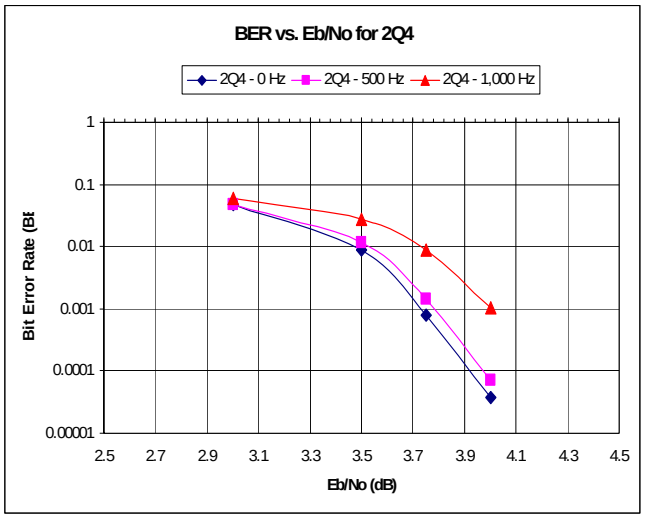

Figure 4: BER vs. signal-to-noise ratio for three frequencies errors

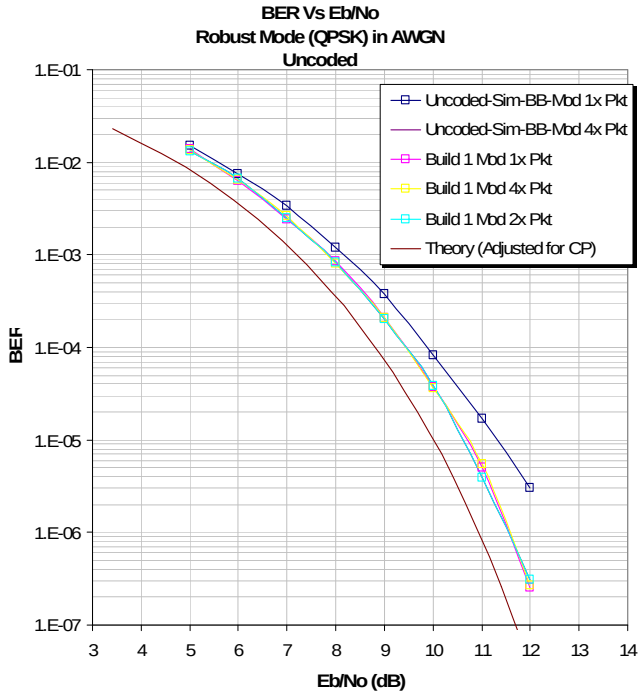

Figure 5: BER vs. signal-to-noise ratio.

BER versus signal-to-noise ratio performance is evaluated for 1-slot BPSK and 2-slot QPSK packets. Figure 4 shows an example of a 2-slot QPSK 4-segment packet BER performance for three frequency errors. The effect of Doppler on bit error rate is evident. Figure 5 also shows the BER performance to compare the FPGA waveform processing against the simulated waveform and theory.

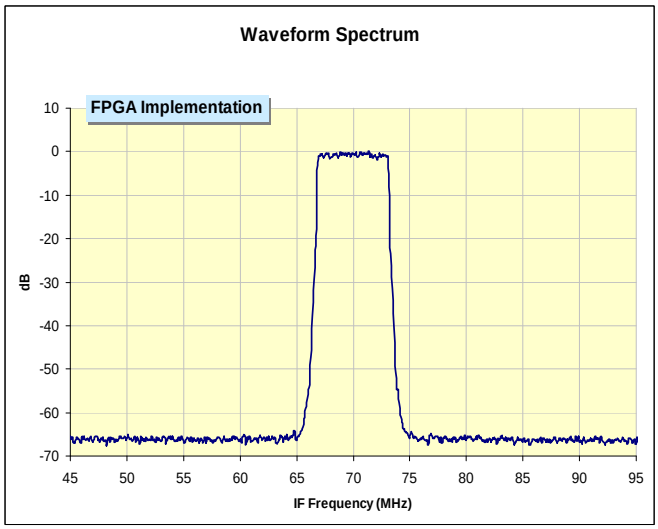

Figure 6: waveform IF spectrum for 20 MHz BW.

To validate a wideband waveform was accurately generated in the waveform test environment, we plotted the waveform spectrum generated by the FPGA at the IF for 20 MHz BW, as depicted in Fig 6.

4. CONCLUSIONS

We have introduced a hardware-in-the-loop waveform test environment that uses a commercial off the shelf PCI card. Flexibility of the COTs hardware and the GUI application allow for specific functional testing such as modulator, demodulator and end-to-end testing of the FPGA waveform processing. This waveform test environment enables our goals of shorter development cycles, product flexibility, adaptability, and decreased costs.

10. REFERENCES

[1] T.Young, "Physical Layer Waveform Implementation," *Raytheon FCS Comms*, 2002.

[2] M.S. Gudaitis, "Practical Considerations for a Waveform Development Environment", MILCOM, Vol 01, pp 190-194, 2001