Implementation of a 350 Mbps FPGA-Based Modem

John Glancy (Naval Research Laboratory, Washington, DC, USA); Michael Rupar (Naval Research Laboratory, Washington, DC); Benjamin Egg (Fred Harris & Associates, San Diego, CA)

ABSTRACT

Modern Unmanned Aerial Vehicles (UAVs) have the capability of carrying sensors that produce large amounts of data. A high speed wireless data link is required to transfer this data to other systems for processing. This paper presents an FPGA-based implementation of a QPSK modulator and demodulator to support a high data rate (350 Mbps) relay from a UAV to a ground station. By using high speed ADCs and DACs in conjunction with state of the art FPGAs it is possible to implement all signal processing functions necessary in a modem. We will present the architecture and methodologies used for both the modulator and demodulator and discuss how these algorithms are implemented in an FPGA.

1. INTRODUCTION

The US Navy dependence on Unmanned Autonomous Vehicles (UAVs) has increased over the past five years, and is expected to do so even more, given the new requirements for force protection and securing of borders. The sensors being flown by these UAVs are growing more and more sophisticated, providing imagery and detection for larger ranges with greater resolution. In addition, the high cost of large UAVs such as the Predator motivates the deployment of smaller UAVs to enable more frequent and continuous surveillance.

With this increase in sophistication comes an increase in the data link bandwidth required to relay that information back to the command site where it can be assimilated. The capacity for data compression is often overstated. High resolution video, even if compressed successfully to 4 Mbps, still exceeds the communications capacity of many existing communications relay systems in the fleet over significant ranges. Given the payload restrictions in size, weight and power (SWAP) of these smaller UAVs, the amount of real-time data assimilation and reduction hardware capable of being carried is further constrained.

It is therefore believed that to meet the requirement of persistent surveillance, the best solution is to move as much data off the UAV to the command facility in real time with minimal processing and decision making on-board. To do this requires a communication bandwidth in excess of hundreds of megabits per second (Mbps), and in excess of 1 gigabit per second (Gbps) if that information is relayed between UAVs for greater coverage.

Critical to meeting these requirements is to develop a simplex high data rate modem package that is capable of moving traffic at rates faster than current point-to-point line-of-sight (LOS) communications systems in use. This paper discusses an FPGA-based implementation of a 350 Mbps simplex QPSK modem being developed by engineers at the Naval Research Laboratory (NRL) with support from Fred Harris and Associates.

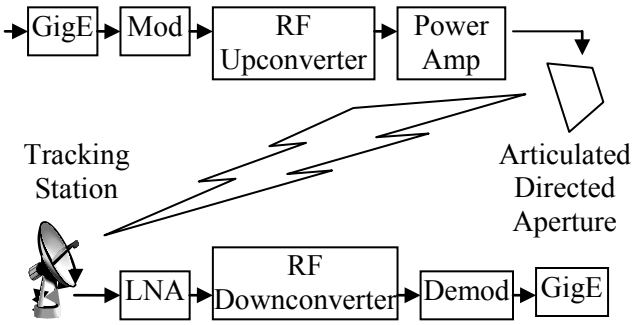

Figure 1: Diagram of UAV communications system being developed by NRL.

2. COMMUNICATIONS SYSTEM SPECIFICATIONS

The modulator and demodulator described in this paper are designed to work in the system specified in Figure 1. In this system the modulator will produce a QPSK signal that is mixed up to a higher frequency for wireless transmission to a land based tracking station. On the ground the signal is mixed back down to an IF, sampled by an ADC and demodulated into the original data stream. The raw data rate of the system is 350 Mbps.

3. HARDWARE PLATFORM

The hardware platform used to implement the modulator and demodulator was the BDR-1 Basic Digital Radio

developed by Fred Harris & Associates. The BDR-1 is a small form factor board (shown in Figure 2) that has two Xilinx Virtex-5 SXT50 FPGAs, a Maxim MAX19692 high speed DAC (2.3Gsps, 12-bit), a Maxim MAX108 high-speed ADC (1.5Gsps, 8-bit), fixed gain amplifiers, variable gain amplifiers, analog filters, and various high speed I/O interfaces. Figure 3 shows a block diagram of the connections between components on the board [1].

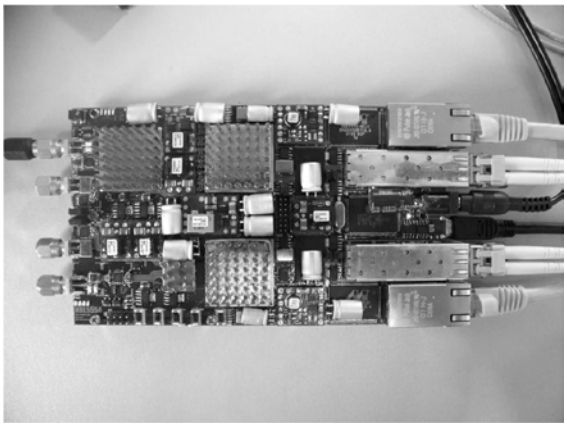

Figure 2: BDR-1, the Basic Digital Radio Platform. IF Ports are to the left, Ethernet, optical, and USB interfaces are to the right. Dimensions are 3.5" x 6.25".

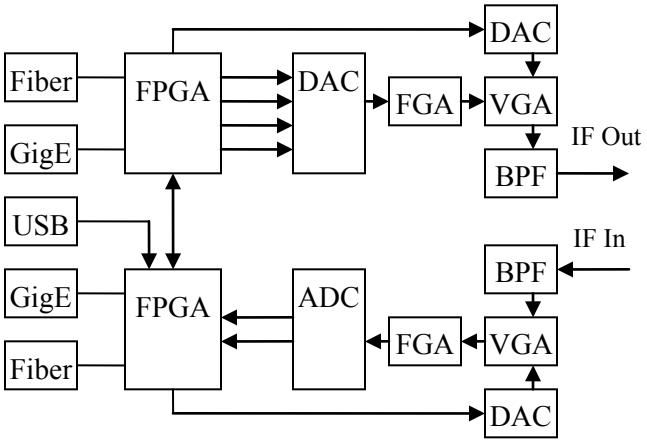

Figure 3: Block diagram of BDR-1 platform.

4. FREQUENCY PLAN

The RF upconverters being used in our UAV communication system require an L-band IF. On the transmit side we are using a sampling rate of 1.75 GHz and selecting the 2nd Nyquist zone image of the signal with a band pass filter. The center frequency was selected to be 4/5 of the sample rate (1.4 GHz) in order to position the signal in the most linear frequency band of the DAC.

On the receive side we chose to undersample the signal at a sampling rate of 1.4 GHz. The center frequency of the signal was chosen to be 3/4 of the sampling rate (1.05 GHz) in order to simplify the digital downconversion process inside the FPGA.

5. TRANSMITTER IMPLEMENTATION

The 350 Mbps data rate implies a 175 MHz symbol rate since we are using QPSK modulation. Therefore, it is necessary to interpolate by a factor of 10 to reach the 1.75 GHz sampling frequency of the DAC. The internal logic of the FPGA is not capable or running at the sampling frequency, so some parallelization is required.

Figure 4 shows the implementation used in the FPGA. A polyphase interpolating filter structure is used for the pulse shaping filters in order to reduce the speed of the internal logic. Each filter operates at 175 MHz. Since we chose to center the output signal at 4/5 of the sample rate, the sinusoids that are multiplied by the I and Q channels repeat every 10 samples. This simplifies the design by allowing us to modulate the filter coefficients instead of the actual data. After the I and Q channels are added together there are 10 parallel data paths.

The 10 parallel data paths go into a FIFO structure which is actually an array of 10 asynchronous FIFOs. Additional logic in the FIFO block selects the next 8 samples from the 10 FIFOs to output onto 8 parallel data paths running at 218.75 MHz. These 8 paths are sent to DDR output buffers on the FPGA. The DAC internally serializes the 4 DDR channels up to the 1.75 GHz sample rate.

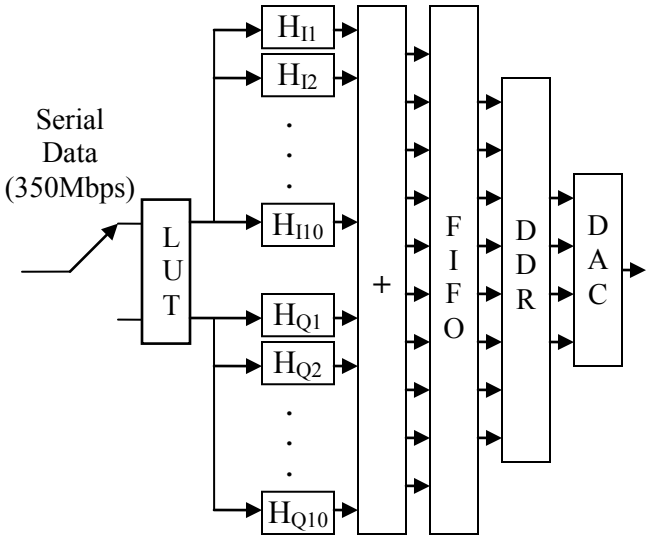

Figure 4: FPGA Implementation of a QPSK modulator

6. RECEIVER IMPLEMENTATION

The block diagram for a basic QPSK demodulator is shown in Figure 5. The following sections will discuss how each of the components is implemented in the FPGA.

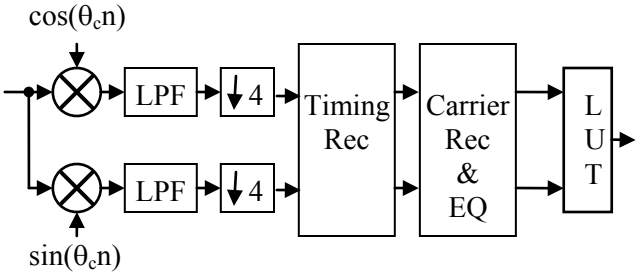

Figure 5: Block diagram for QPSK demodulator implemented in the FPGA.

6.1 Digital Downconverter

The digital downconverter (DDC) is simplified due to the center frequency of the received signal being at 3/4 of the sample rate. As shown in Figure 6, by implementing the lowpass filter of the I channel as a polyphase decimating filter, it is possible to remove half of the coefficients due to the constant values of the cosine signal used to mix the signal down. Since we are clocking the multipliers at such a high speed, there is very limited reuse of resources. This DDC structure results in a significant savings of multiplier resources. A similar simplification is done to the Q channel. Following the downconversion, the I and Q channels are operating at 2 samples/symbol (350 MHz sample rate).

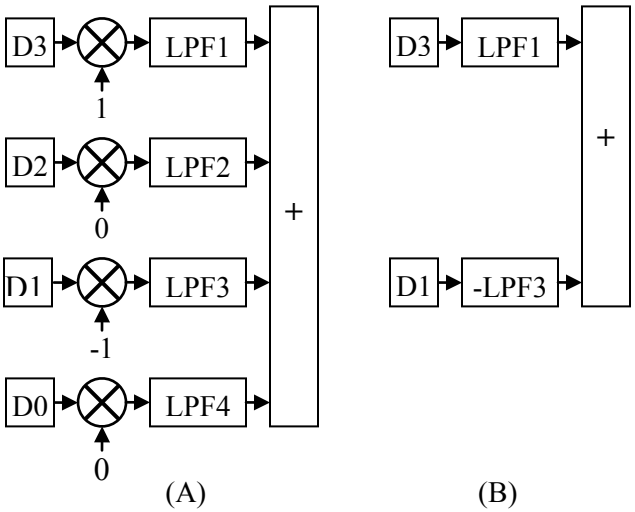

Figure 6: (A) Basic polyphase digital downconverter. (B) Simplified polyphase digital downconverter.

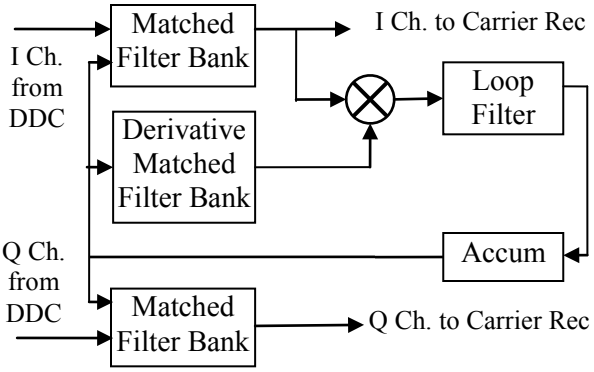

Figure 7: Timing recovery block diagram.

6.2 Timing Recovery

The timing recovery circuit is implemented using a maximum likelihood polyphase matched filter system [3]. Figure 7 shows the block diagram of the timing recovery circuit. This method was chosen due to its quick convergence time. Each filter bank is implemented as a single 12-tap filter with 16 sets of reloadable coefficients stored in memory in order to reduce the total number of multipliers used (Figure 8). The timing error is calculated as the product of the matched filter output with the derivative matched filter output. The error is averaged using a first-order loop filter and then added to an accumulator. The accumulator is used to address the filter bank coefficient memories and select the correct sampling position.

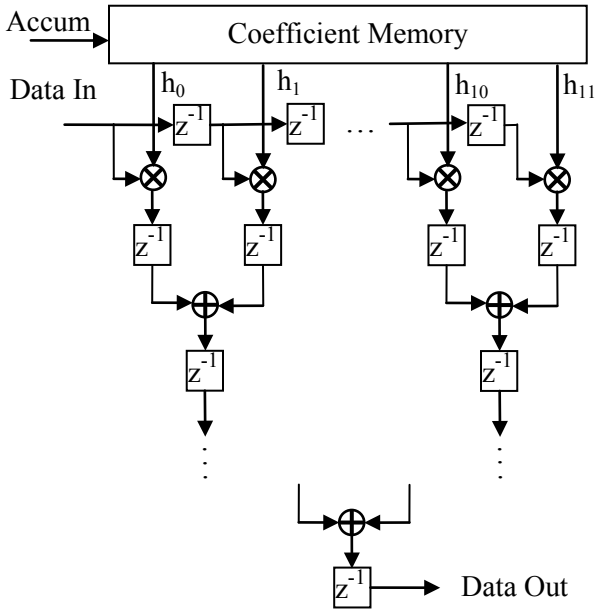

Figure 8: Reloadable coefficient matched filter. There are 16 sets of reloadable coefficients for the 12-tap filter illustrated.

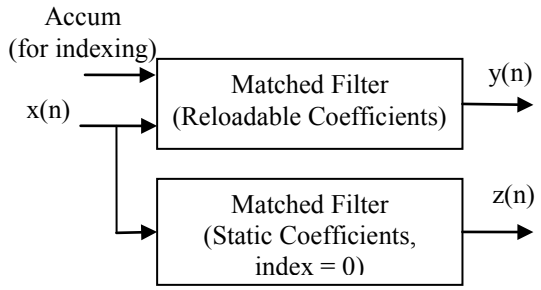

Figure 9: Structure using a reloadable coefficient matched filter and a static coefficient matched filter to avoid losing valid samples on an accumulator rollover.

For every symbol period, one sample is a valid data point and the other is a transition point. The top bit of the accumulator selects whether the first or second sample is valid. Accumulator overflow and underflow conditions must be taken into consideration in order to prevent bits from being lost or duplicated in the data stream.

In the overflow case, both samples within a symbol period are considered invalid. In the underflow case it is necessary to output two valid samples within a single symbol period. This is done by implementing a second matched filter with static coefficients (Figure 9). The coefficients are those of the 0^{th} leg of the filter bank. When an underflow occurs, the two valid samples for that symbol period are the first sample from the static coefficient filter and the second sample from the reloadable coefficient filter bank. Figure 10 illustrates which samples out of the reloadable and static matched filters are valid.

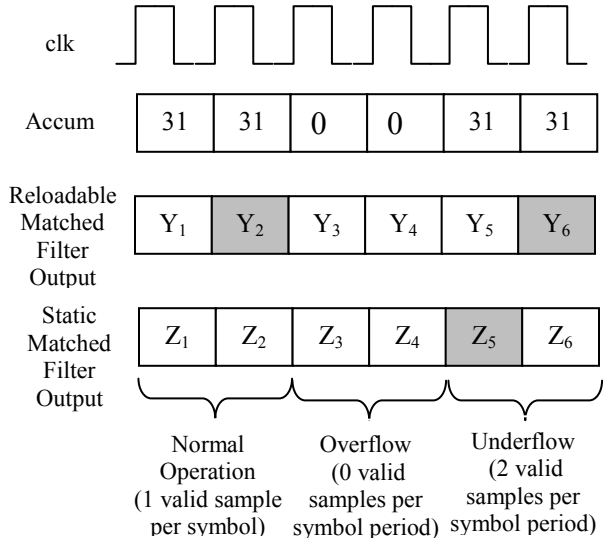

Figure 10: Illustration of which samples are valid under normal, overflow, and underflow conditions. Valid samples are shaded gray.

6.3 Carrier Recovery & Equalization

In QPSK, an estimate of the phase error can be computed as

$$e_{ph} \approx Y(n)sign(X(n)) - X(n)sign(Y(n)) \quad (1)$$

Figure 11 shows the block diagram for the QPSK carrier recovery circuit [5]. Like the timing error, the carrier error is averaged with a first-order loop filter. This value is used to control the frequency of a direct digital synthesizer (DDS) that is used rotate the signal.

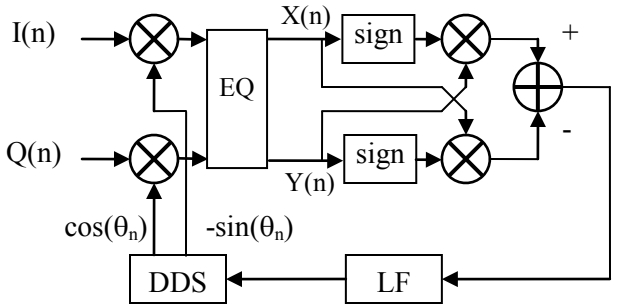

Figure 11: Carrier recovery circuit

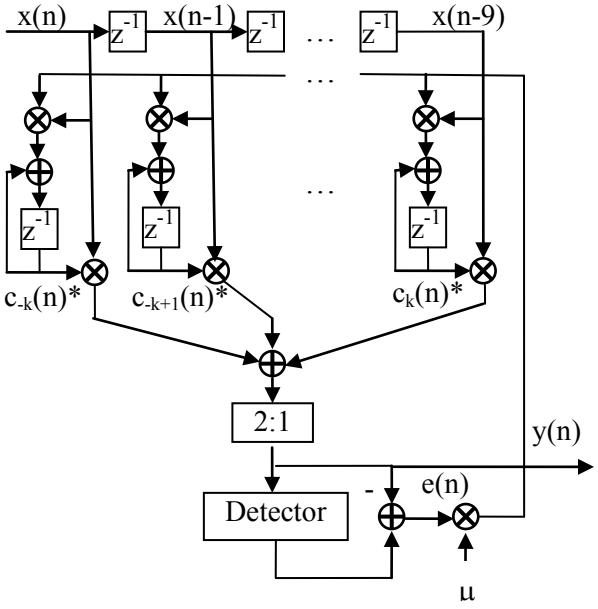

Figure 12: Adaptive FIR Equalizer.

The equalization process is essential to minimize the impact of non-ideal RF analog components and dynamic channel conditions. Due to variations over time of both hardware components and channel conditions, the equalization is adaptive, and utilizes a least mean squared (LMS) algorithm to adjust the equalization filter coefficients

until the error is minimized. For a starting point we chose to implement a complex 10-tap equalizer in the FPGA.

Figure 12 illustrates an adaptive FIR filter structure adopted for the equalizer. The parameter μ is a damping factor which reduces the contribution of each error estimate to slow or speed up the equalizers adaption rate.

It is important to recognize that the equalizer is a complex filter. In other words, it is actually two simultaneous 10-tap FIR filters. Furthermore, since we are operating on complex data, the adaptive equalizer is actually composed of four 10-tap filters. Figure 13 illustrates the 4 paths in the block labeled Complex Equalizer.

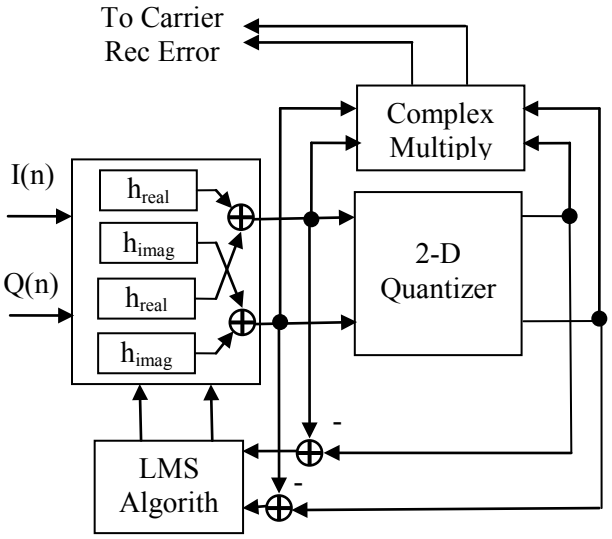

Figure 13: Detailed View of complex signal manipulation.

7. RESULTS

Xilinx XC5VSX50-T-1 FPGAs were used for both the transmitter and the receiver. The resource utilization of both designs is shown Tables 1 and 2. The transmitter implementation is fairly straightforward. The interpolation filters are the main consumer of device resources on the transmit side. On the receive side, the equalizer is the main consumer of device resources. The equalizer uses 8 multipliers per tap. In our case, we are using a 10-tap equalizer, which requires 80 multipliers.

Table 1: Device Utilization

	BRAM	Mults	Slices	LUT/FF
TX	4	120	1020	1956/2064
RX	12	175	3340	4191/10374

Table 2: Percentage of Device Used

	BRAM	Mults	Slices	LUT/FF
TX	3%	41%	12%	5%/6%
RX	9%	60%	40%	12%/31%

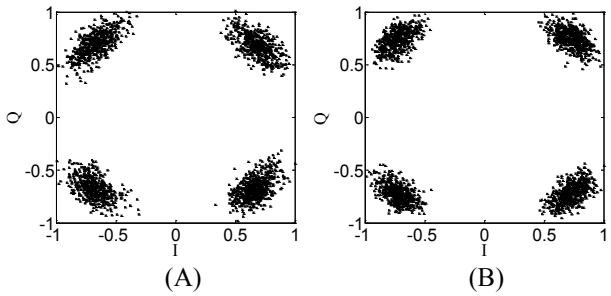

Figure 14: Short range received constellations. (A) Carrier recovery output with equalizer off. (B) Carrier recovery output with equalizer on.

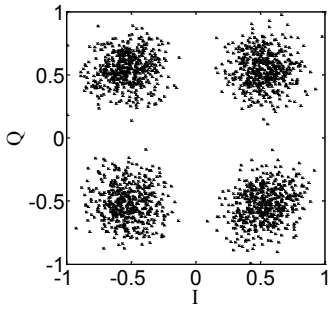

Figure 15: Long range received constellation (equalizer on).

It was relatively easy to meet timing with the transmit design. The design runs at a maximum of 232 MHz (only 218.75 MHz is required) with very little effort. On the other hand, the receive design required a great deal of manual placing of components in the FPGA. The maximum clock rate for the receive design is 360 MHz (350 MHz is required). Route delays in the loops for the timing recovery and carrier recovery circuits are probably the root cause of this difficulty. An easy way to reduce the amount of effort required to route this design would be to migrate to a faster speed grade FPGA.

Two free-space tests of the modem were conducted in which the modulator and demodulator were positioned on stationary mounts on the ground. The first test was a short range test on a runway over a 100 yard distance. Figure 14 shows plots of the received constellations and the effect that the equalizer has. With the equalizer turned off the timing and carrier recovery loops are able to lock and produce a fairly clean constellation. The constellation points tighten up further with the equalizer turned on. This shows that the equalizer is able to remove some analog distortions that have been introduced into the system.

The second test was a long range test from the Naval Research Laboratory in Washington, DC to a nearby hilltop over a distance of 3 miles. This is a more difficult channel to transmit over due to nearby buildings, trees, and water which can create multipath reflections. Figure 15 shows a plot of the received constellation with the equalizer turned

on. Over the longer distance the demodulator was still able to produce a constellation with fairly distinct points. The system was unable to lock with the equalizer turned off. This illustrates the effectiveness of the equalizer in our system.

Table 3: Free-space Packet Error Results

Test Distance	BER
100 yards	5.7e-3
3 miles	5.0e-3

Table 3 shows the bit error rates (BER) in both free-space tests. No forward error correction (FEC) routines were used in either test. In each case, the packet size was 1024 kB and 100,000 packets were analyzed. The results show that the BER remained fairly consistent across the two tests. The addition FEC would reduce the BER to more acceptable levels.

8. CONCLUSION

By using the flexible architecture of FPGAs in combination with high-speed DACs and ADCs, it is possible to implement a high-speed modem that could potentially be used on UAVs. With this design we were able to successfully transmit and receive a 350-Mbps data stream over a distance of 3 miles. These free-space tests also show that the adaptive FIR equalizer performs well and successfully cleans up the receive constellation. Further development will included the addition of FEC to reduce the BER. The implementation of higher order PSK or higher order QAM will also be investigated in order to increase the overall data-rate even further.

9. REFERENCES

[1] Fred Harris & Associates, "BDR-I User Manual."

[2] S. Haykin, *Communications Systems*. (John Wiley & Sons Inc, New York. 2001).

[3] C. Dick, B. Egg, F. Harris, "Architecture and Simulation of Timing Synchronization Circuits for the FPGA Implementation of Narrowband Waveforms," *Proceeding of the SDR 06 Technical Conference and Product Exposition*. 2006.

[4] L. Litwin, "Matched Filtering and Timing Recovery in Digital Receivers,"http://rfdesign.com/mag/ radio_matched_filtering_timing, *RFDesign.com*, September 2001.

[5] C. Dick, F. Harris, and M. Rice, "FPGA Implementation of Carrier Synchronization for QAM Receivers," *Journal of VLSI Signal Processing*. vol. 36, Issue 1, January 2004, pp.57-71.