

Design of a High Frequency FPGA Acoustic Modem for Underwater Communication

Nusrat Nowsheen

School of Engineering and IT
UNSW@ADFA, Northcott Drive
Canberra, ACT 2600 Australia

Craig Benson

School of Engineering and IT
UNSW@ADFA, Northcott Drive
Canberra, ACT 2600 Australia

Michael Frater

School of Engineering and IT
UNSW@ADFA, Northcott Drive
Canberra, ACT 2600 Australia

Abstract—Contemporary underwater acoustic networks use low frequency modems. While these modems can provide long range communication, their low operating frequencies mean that only low channel bandwidth is available, which results in slow data rates. This motivates our development of a high frequency modem which offers the potential for large channel bandwidth, and hence greater link capacity. There is a range-frequency trade-off because absorption becomes very high at high frequency. Our intended operating frequencies from 100 kHz to 1 MHz would only support link ranges perhaps from 1 km down to under 100m, with communication ranges longer than this requiring forwarding over a network. Reconfigurable computing based Field Programmable Gate Arrays (FPGAs) are used to accelerate product development and support evolution of fielded systems. Given the immaturity of the field of underwater communication, a reconfigurable modem is a valuable tool for development and testing modem techniques. We present a design idea to implement an acoustic modem solely in FPGA, whereas most existing modems are implemented as a combination of FPGA and DSP processors. Aside from simple anti-aliasing filters, which could be incorporated in the preamplifier stage, the modem does all of its processing in the digital domain – maximising flexibility. In this work, we describe the initial design and architecture of our software based acoustic modem that avoids the monetary cost or time investment required to design a commercial modem or custom hardware for many applications. Our demodulator is implemented using a Costas loop which performs both suppressed carrier reconstruction and synchronous data detection within the loop. Results from initial implementation are also reported in this paper.

I. INTRODUCTION

Digital communications through radio channels where data is transmitted by means of electromagnetic waves differs from communication in underwater, where acoustic waves are normally used. The propagation characteristics of the acoustic waves differ from radio waves and create a challenging environment for reliable communication [1]. The propagation speed in the underwater acoustic (UWA) channel is low relative to potential source receiver velocities, leading to time-scale changes which are significantly greater than those encountered at radio frequencies. The available bandwidth of the UWA channels is also very limited and depends on both range and frequency [2, 3]. Within this limited bandwidth, the acoustic signals are subject to time-varying multipath [4], which may result in inter-symbol-interference and large Doppler shifts and Doppler spreads relative to radio channels. There are a large number of possible propagation paths and

the individual (different) Doppler effects of each path superimpose at the receiver.

Sophisticated digital modulation methods have been adopted for radio communication systems to support high data transmission. Due to channel properties and bandwidth limitations, accomplishing the same data rate for underwater communication is impractical. Early underwater acoustic telemetry used Amplitude Shift keying (ASK) and Frequency Shift Keying (FSK) for data transmission. ASK performs well when the path is straight and reverberation is low e.g. vertical transmission. In a noisy channel its performance can be improved by an error correction scheme [5] but this decreases the data rate. Most underwater communications use some form of FSK or PSK. FSK is considered appropriate for shallow water long – medium range channels that exhibit rapid phase variation [5]. This technique is still prone to reverberation problems, although incoherent detection eliminates the problem of carrier phase tracking. Orthogonal Frequency Division Multiplexing (OFDM) offers ease of channel equalization in the frequency domain; it is extremely sensitive to any frequency offset that can result from a mismatch between the frequencies of the local oscillators, or from Doppler distortion caused either by the transmitter/receiver motion or by a mismatch between their sampling rates. An OFDM system can only tolerate a frequency offset that is much smaller than the carrier spacing and for this reason frequency synchronization and time-base (Doppler) correction are a critical part of the system [6]. Direct Sequence Spread Spectrum based Code Division Multiple Access (CDMA) can perform well in multipath environments and offers communication at lower Signal-to-noise ratio (SNR), but at the expense of a decreased data rate.

Existing descriptions of the underwater acoustic channel tend to describe relatively long paths at relatively low frequencies. The aforementioned channel characteristics are common, and have led to substantial investigation of exotic modulation schemes, such as OFDM. We believe that many of these channel effects will be less pronounced in a high-frequency short-range channel, simplifying the equalisation task and allowing use of simple coherent modulation schemes, such as PSK. Recent studies also show that Phase Shift keying (PSK) achieves a better signal-to-noise ratio (SNR) and thus it is a frequently applied technique for underwater communications [5].

This paper examines in detail the design of a new underwater acoustic modem targeted for high frequency communications. Our FPGA modem is designed to modulate and demodulate a BPSK signal, which has been shown to be suitable for underwater communications. The remainder of this paper is organized as follows. Section II discusses the existing approaches towards underwater modem design, Section III explains our FPGA modem, Section IV contains implementation and verification of the design and finally Section V concludes the paper.

II. RELATED WORKS

Underwater acoustic communication has been investigated by many researchers and the major focus in this area is transmission range, bandwidth utilization and reliability in handling multipath propagation and other channel effects. The diversity in the potential application space for underwater sensor networks has led to different projects with a wide range of design requirements. The high frequency channel is poorly documented, because most existing work applies to longer paths at lower frequencies. Commercially available acoustic modems produce data rates from 100 bps to about 40 Kbps and have an operating range of up to a few km and an operating depth in the range of thousands of meters. Researchers in California Institute for Telecommunications and Information Technology have designed a FSK software modem [7] which can operate only at frequencies below 3 kHz. Researchers at the Woods Hole Oceanographic Institution have described a completely self-contained, autonomous acoustic modem capable of moderate communication rates with low power consumption [8]. This modem uses a single specialized DSP board with on board memory and batteries. Our work is targeted to operate at high frequency with the proper utilization of available underwater bandwidth.

Researchers at UC, Santa Barbara are also developing a hardware acoustic underwater telemetry modem (133 bps data rate) using a DSP board with custom amplifiers, matching networks, and transducers for ecological research applications [9]. Most recent works in underwater development have relied on using the WHOI micro-modem [10], which operates only at low frequencies (<30 kHz). Others use commercial acoustic modems or build custom transceivers for each application - both of which lack flexibility and may require prohibitive monetary or time-investment for many applications. The cost of a single commercial underwater acoustic modem is at least several thousand US dollars. Most of the existing modems for underwater communications are implemented as a combination of FPGA and DSP processor and support specific applications. Both of the works reported in [8] and [9] focus on developing underwater acoustic modems with low cost and specialized affordable hardware, our work aims to reduce the cost even further and make a completely software based acoustic modem with minimal hardware support that can operate in a high frequency (100 kHz to 1MHz) underwater environment.

The BPSK modulation technique described in [11] uses a multiplexer that selects the carrier either in phase or 180° out of phase depending on the condition of binary input data. The authors in [12] have described the FPGA-based digital demodulation of Radio Broadcast Data System information transmitted along FM broadcast but the filter implementation, sinusoidal generation and carrier recovery technique are not discussed in detail. The authors in [13] have discussed a demodulator design for processing satellite data communications. Otherwise there is little published work on FPGA modems. Demodulators require filters to eliminate unwanted noise from the received signal. The authors in [14] present an efficient method for implementing digital filters in FPGAs, that avoid the general approach of multipliers which are not economic in FPGA architecture. In their work, multipliers are replaced by look-up tables and adder-subtractors, which use Bit-Serial Arithmetic. This approach reduces the logic cell utilization in an FPGA but degrades the filter performance. Other approaches towards multiplier-free FPGA filter designs can be seen in [15, 16], which are expensive since they are implemented in hardware.

III. FPGA ACOUSTIC MODEM

This section presents a novel design concept to build a software defined modem in FPGA, providing a flexible platform on which to develop techniques appropriate to high-rate, high-frequency communication.

A. BPSK Modulation

Binary Phase Shift Keying (BPSK) is one of the most efficient binary data modulation techniques in terms of noise immunity per unit bandwidth. Simple BPSK modulation is the process of shifting a carrier phase for one data symbol while not shifting it for another. Figure 1 shows the simplified block diagram of our BPSK modulator which operates by shifting a carrier's phase by 180° for data "one" while not shifting it for data "zero". We have implemented it as a look-up table (LUT) based sinusoid generator, followed by XORing these samples with the binary input data. Because the code (data) and carrier signals are derived from the same reference oscillator they are guaranteed to be synchronised. This may present opportunities for unique oscillator and timing recovery techniques in the demodulator.

In an M-ary or multiple phase-shift keying (MPSK), there are more than two phases, usually four or eight. Our modulator can produce MPSK by using multiple look-up tables or by shifting the index into the look-up table by an appropriate offset for each of the M symbols.

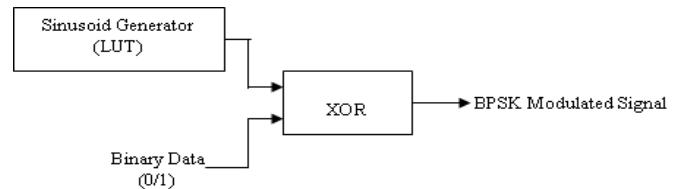


Figure 1. Block Diagram of BPSK Modulator.

B. BPSK Demodulation

Our FPGA demodulator (Costas loop [17]) consists of Multipliers, Filters and a Numerically Controlled Oscillator (NCO) as shown in Figure 2. Carrier recovery during demodulation is done using Costas loop to avoid the phase error problem of squaring loop. The Costas loop relies on feedback concepts related to a Phase Lock Loop (PLL) and performs both phase-coherent carrier reconstruction and synchronous data detection within the loop.

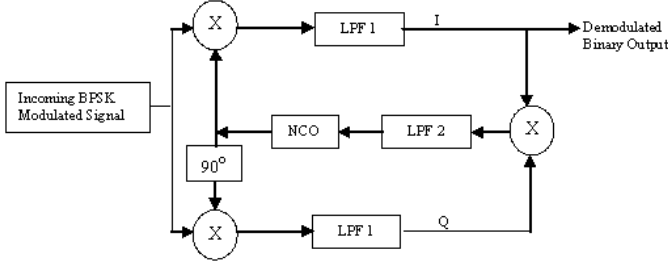


Figure 2. BPSK Demodulation using Costas loop.

C. Carrier Recovery

The mechanism of the Costas loop carrier recovery is to adjust the NCO until it matches the received BPSK signal in phase and frequency. The incoming BPSK modulated signal is multiplied by the local carrier and 90° phase-shifted local carrier. The low-frequency product of a BPSK signal and its coherent carrier is the demodulated information, while the low-frequency component of the BPSK signal and the quadrature of the carrier is zero. The outputs of both demodulators (I and Q) are passed through low-pass filters to remove the high frequency component. The Costas loop is locked when it has adjusted its NCO phase and frequency until the demodulated in-phase I signal is maximum and the demodulated quadrature Q channel is close to zero. In this particular design, the loop is considered locked when the phase error (difference between I and Q channel) is less than a particular threshold value.

As shown in Figure 3, our NCO is implemented as a phase accumulator, which provides an address for two look-up tables containing a sin-wave and cos-wave respectively. The accumulator can handle a fractional phase-value where the non-fractional part is used to address the look-up tables and fractional part only accumulates into the next phase-value. The resolution of the phase accumulator (n bit), and hence frequency resolution can be greater than the number of address bits (k) for the look-up tables. The output of LPF 2 determines the phase error between the BPSK signal and the NCO and is fed back to the NCO to adjust the phase and frequency according to that phase error. The accumulator size determines the frequency resolution, and the size of the look-up tables determines the quality of the produced signal. The current step size value calculated from output of LPF 2 is added with the previous step size stored in phase accumulator of NCO. The step size is digitized slightly based on the output from LPF 2. The phase accumulator sends the upper k bits

($k < n$) of the summation (non-fractional part) to address the look-up tables and sends these n bit to be added with the next step size for getting the next numerical value to address the LUTs. By extending the precision of the phase accumulator beyond the address size of the LUT, our tuneable oscillator (NCO) design can deal with almost infinite frequency resolution which overcomes the limitation of limited frequency support of existing NCO design [18].

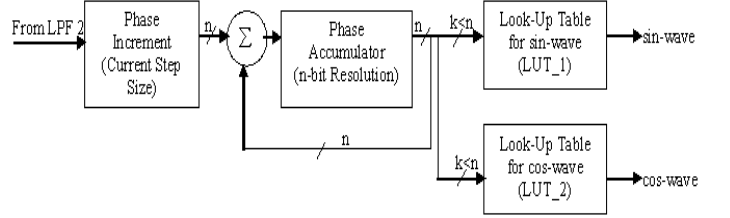


Figure 3. Block Diagram of NCO.

D. Low-pass Filter Discussions

Two identical raised cosine low-pass filters (LPF 1) are used in I and Q channels of the Costas loop to remove high frequency components and to avoid imbalances that will prolong loop settling time. The Raised Cosine filter helps in pulse-shaping in digital modulation due to its ability to minimise inter-symbol interference. As shown in Figure 4, we have implemented these as symmetric FIR filters in the FPGA as a tapped delay line, but instead of using multipliers, we implement the filters weights as a series of right shifts. The filter output is calculated using the following equation:

$$y(m) = b(1) * x(m) + b(2) * x(m-1) + \dots + b(n+1) * x(m-n)$$

where, $y(m)$ is the filter output sample, $x(m)$ is the input sample, $b(i)$ are filter coefficients and n is the filter order.

To design the filter, we have generated filter coefficients (<1) in MATLAB and then quantized the coefficients as a summation of two inverse powers of two. Multiplication by an inverse power of two is simple to implement as a right shift operation in the FPGA. To illustrate, suppose the original coefficient generated from the MATLAB filter design tool for a specific filter tap is 0.78. We would implement this by producing two right bit-shifted versions of the input – in this case a 1-bit and 2-bit shift, and then summing the resultant values. For example, if the current input sample is 128 then the bit shifted values would be 64 and 32, which would produce a filter tap output of 96. If the filters were implemented using a multiplication at full precision the (correct) answer would be 99.84. Implementing the filter elements using this right-shift and add technique uses substantially less FPGA resources than performing multiplication. The decision to add only two elements is a design decision; higher filter coefficient resolution could be obtained by using more right shifted values, provided the FPGA resources were available. In our design we avoid multiplication and quantize the coefficient as $(2^{-x} + 2^{-y})$, where x and y are integers, so the original coefficient 0.78 would be quantized as $(0.75 = 2^{-1} + 2^{-2})$. This avoids the need for an explicit multiplication, which is expensive to perform in an

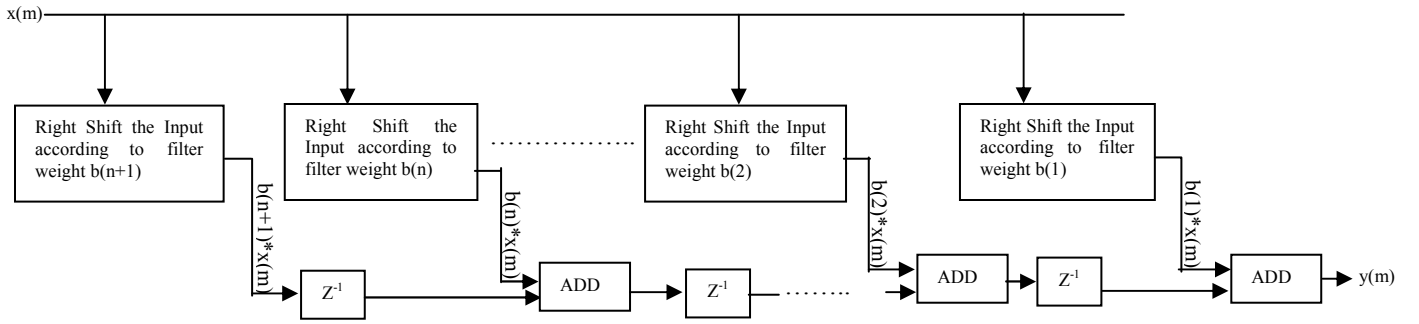


Figure 4. Block Diagram of Filter.

FPGA. The second type of filter (LPF 2) is used to remove spurious components from the feedback signal and is currently being implemented using similar principles.

IV. IMPLEMENTATION AND VERIFICATION

The FPGA modem design is being implemented on the Altera's Cyclone III FPGA starter kit [19] and a Terasic ADC/DAC mezzanine card. The FPGA board master oscillator operates at 50 MHz. Each design block is built and validated separately. The experimental result of each of the individual blocks is obtained at the output of DAC in the THDB-ADA daughter board on Cyclone III starter Kit with HSMC interface and over a custom digitized USB interfaces.

A. Modulator Output

The result obtained from our BPSK modulator is shown in Figure 5. The modulator output is observed and measured using a Digital Oscilloscope. The figure clearly shows that the phase change in the output occurs when there is a change in the binary input data (0 to 1 or 1 to 0). In this example, the data rate and carrier frequency are set to the same value to produce a compact diagram.

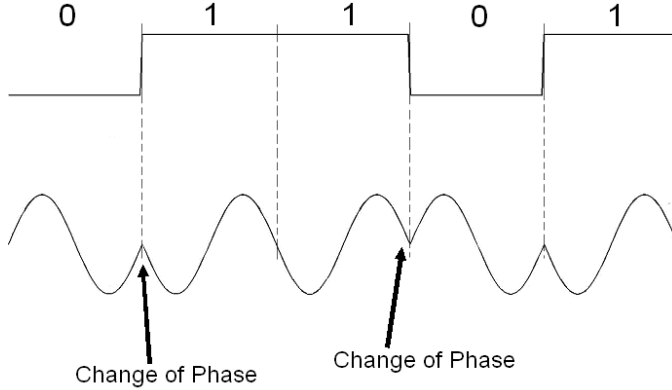


Figure 5. BPSK Modulation.

B. NCO Output

The NCO output frequency depends on the phase increment value calculated from the output of loop filter (LPF 2). For this particular experiment, we have maintained a look-up table for our NCO with 64 samples. Figure 6(a) shows the sinusoid output of NCO with frequency 390.6 kHz when the step size (phase increment) of NCO is 0.5 and the phase accumulator

scans 128 samples (each sample in the LUT is used per cycle twice) from the LUT. The FPGA clock operates at speed of 50 MHz. Dividing 50MHz by 128 produces the desired frequency of ~390 kHz. Figure 6(b) shows the sinusoid output of NCO with frequency 588.2 kHz when the current step size is 0.75 and scans 85 samples from the LUT. Dividing 50MHz by 85 produces the desired frequency of about 588 kHz. NCO accuracy and stability are driven from the crystal on the FPGA board.

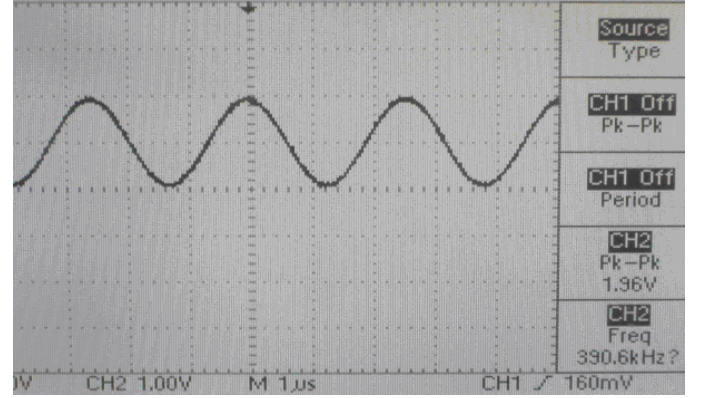


Figure 6(a). NCO Output with 0.5 Phase Increment.

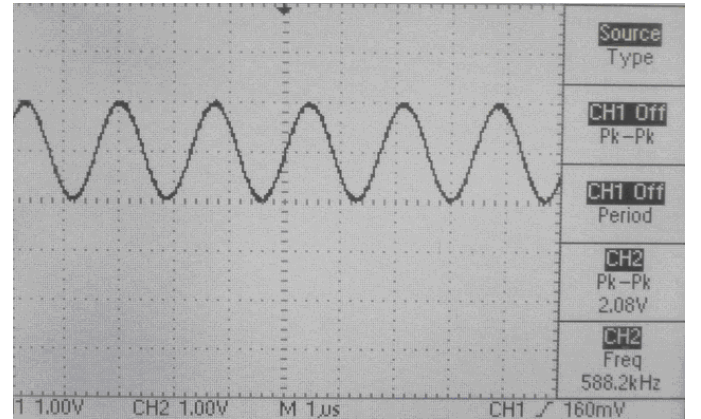


Figure 6(b). NCO Output with 0.75 Phase Increment.

C. Raised Cosine Filter Output

Figure 7(a) shows the frequency response of LPF 1 (raised cosine filter) with the coefficients generated from MATLAB. Figure 7(b) shows the frequency response of LPF 1 (raised cosine filter) with the quantized coefficients that we selected

to use in the FPGA to implement multiplication-free operation in filter. We have implemented the filter in MATLAB with multiplication using the original filter coefficients as well as our quantized coefficients. The impulse responses are shown in Figure 8(a) and Figure 8(b) respectively. We have also implemented the multiplication-free filter on the FPGA with the quantized coefficients and obtained the impulse response as shown in Figure 8(c). Our result shows that the multiplication-free filters produce almost similar frequency and impulse response as filters implemented with multiplication but avoid the multiplication cost.

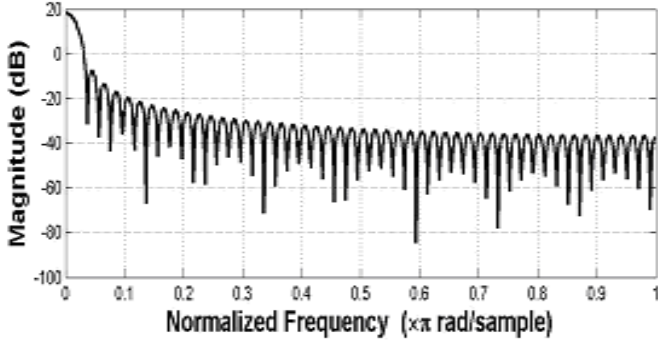


Figure 7(a). Frequency Response of Raised Cosine Filter with Original MATLAB Coefficients.

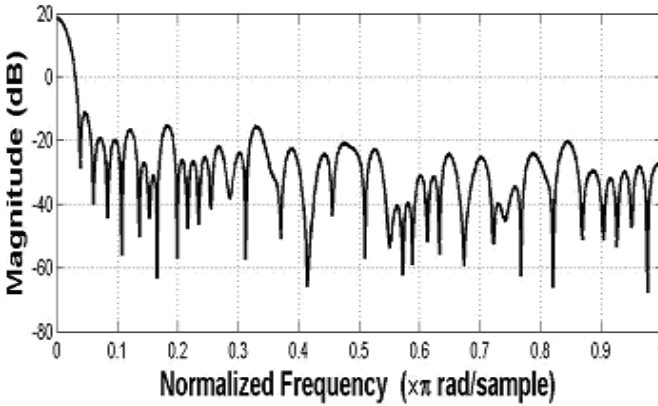


Figure 7(b). Frequency Response of Raised Cosine Filter with Quantized FPGA Coefficients.

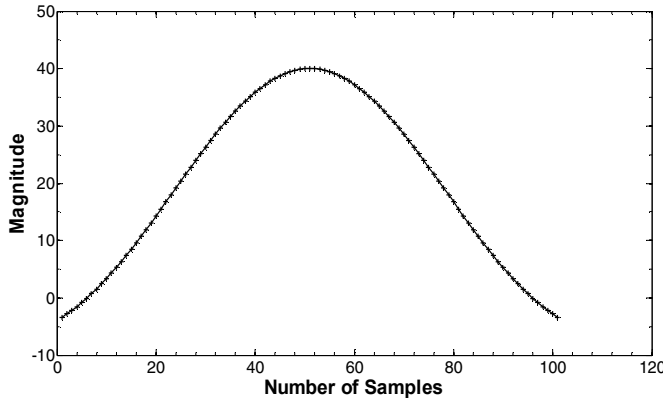


Figure 8(a). Impulse Response with Original Filter Coefficients.

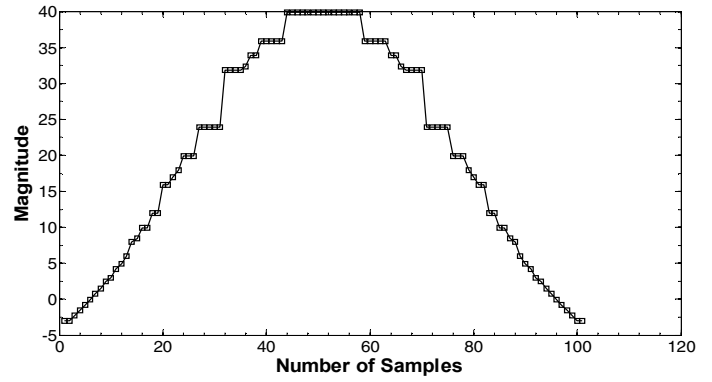


Figure 8(b). Impulse Response with Quantized Filter Coefficients.

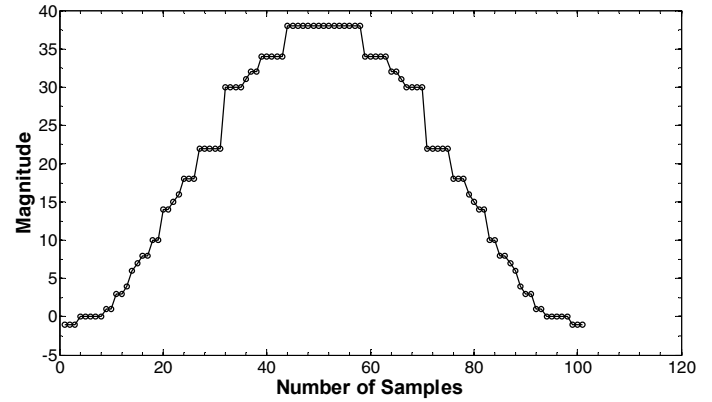


Figure 8(c). Impulse Response of Multiplication-free Filter with Quantized Coefficients.

The impulse response shown in figure 8(b) does not exactly match with figure 8(a) due to quantization error of the filter coefficients. Figure 8(c) produces impulse response of multiplication-free filter implemented as a series of right-shifting operations and is different from 8(b) since right-shifting operation causes losing some precisions at the output samples. If Figure 8(b) is found to be important then rounding rather than the truncation could be used when right-shifting. Figure 8(d) shows the impulse responses of MATLAB implemented filter using original coefficients, quantized coefficients and FPGA implemented multiplication-free filter with quantized coefficients together. It is clearly evident from the figures that our multiplication-free filter produces correct impulse response with very little loss of precision.

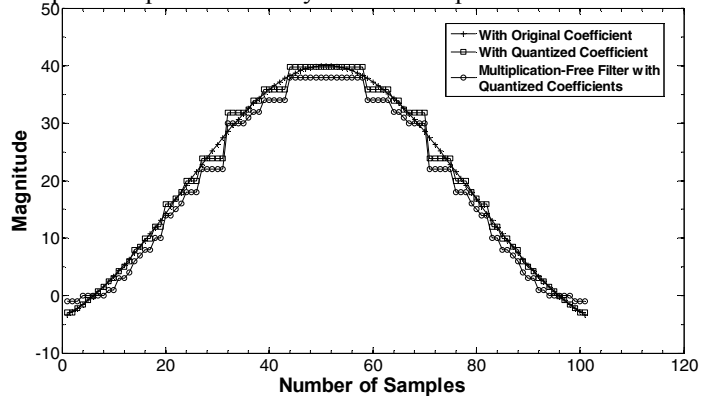


Figure 8(d). Figures 8(a), 8(b) and 8(c) superimposed.

V. CONCLUSION

This paper describes our work on designing and developing a high frequency FPGA acoustic modem for underwater communications. Our software based FPGA modem should reduce hardware cost. In addition, the filter implementation using shift and add avoids multiplication which consumes considerable resources on an FPGA. The capability offered by the reconfigurable FPGA platform is a promising choice for more robust signal processing or communication system. We have presented the details of our design with preliminary test results. However, the entire work is still in progress. Tests of the modem with real underwater communication are planned for near future. Adding features to make demodulator more tolerant of channel effects and to support a wide range of data rate and modulation schemes is a future goal.

REFERENCES

- [1] E. M. Sozer, J. G. Proakis, M. Stojanovic, J. A. Rice, A. M. Hatch, "Direct Sequence Spread Spectrum Based Modem for Underwater Acoustic Communication and Channel Measurements", *Oceans'99 MTS/IEEE*.
- [2] L. Berkhovskikh and Y. Lysanov, "Fundamentals of Ocean Acoustics", Newyork: Springer, 1982.
- [3] A. Quazi and W. Konard, "Underwater Acoustic Communications", *IEEE Commun. Mag.*, pp: 24-29, Mar, 1982.
- [4] M. Stojanovic, "Wiley Encyclopaedia of Telecommunications", chapter Acoustic (underwater) Communications. JohnWiley & Sons, Inc., 2003.
- [5] R. S. H. Istepanian, M. Stojanovic, "Underwater Acoustic Digital Signal Processing and Communication Systems", ISBN 0-7923-7304-9, 2002.
- [6] M. Stojanovic, "Low Complexity OFDM Detector for Underwater Acoustic Channels", *OCEANS 2006*.
- [7] R. Jurdak, C. V. Lopes, and P. Baldi, "Software acoustic modems for short range mote-based underwater sensor networks," *Oceans 2006 - Asia Pacific*, Vols 1 and 2, pp. 292-298 985, 2006.
- [8] M. Johnson, J. Preisig, L. Freitag, "FSK and PSK Performance of the Utility Acoustic Modem", *IEEE OCEANS*, 1999.
- [9] R. A. Iltis, H. Lee, R. Kastner, D. Doonan, T. Fu, R. Moore and M. Chin, "An Underwater Acoustic Telemetry Modem for Eco-Sensing," In *proc. MTS/IEEE Oceans'05*, September 2005.
- [10] L. Freitag, M. Grund, J. Partan, S. Singh, P. Koski, and K. Ball, "The WHOI Micro-Modem: An Acoustic Communications and Navigation system for Multiple Platforms," *Oceans 2005*, Sept. 2005.
- [11] J. S. Ruque, D. I. Ruiz and C. E. Carrion, "Simulation and Implementation of BPSK Modulation on a FPGA Xilinx Spartan 3 xcs200-4ftp256 using Simulink and the System Generator blockset for DSP/FPGA, URL: http://www.ece.unm.edu/xup/docs/collaboration/BPSK_Modulator.pdf.
- [12] P. I. Martos & J. L. Bonadero, "FPGA-Based Digital Demodulation", *XII Reunión de Trabajo en Procesamiento de la Información y Control*, 16 al 18 de octubre de 2007.
- [13] K. R. Nataraj, Dr S. Ramachandran and Dr B. S. Nagabushan, "Development of Algorithm for Demodulator for Processing Satellite Data Communication", *IJCSNS International Journal of Computer Science and Network Security*, vol. 9, no. 6, June 2009.
- [14] M. Arroyuelo, J. Arroyuelo and A. Gross, "FPGA-Based Digital Filters using Bit-Serial Arithmetic", *CACIC 2007*.
- [15] M. Shajaan, K. Nielsen and J. A. Sorensen, "Time-Area Efficient Multiplier-Free Filter Architectures for FPGA Implementation", *1995 International Conference on Acoustics, Speech, and Signal Processing*, Vols 1-5, pp. 3251-3254 3662, 1995.
- [16] S. Shanthala and S. Y. Kulkarni, "High Speed and Low Power FPGA Implementation of FIR Filter for DSP Applications", *European Journal of Scientific Research*, ISSN 1450-216X vol. 31, no. 1 (2009), pp. 19-28.
- [17] J. Feigin, "Practical Costas Loop Design", *RF Design*, pp. 20-36. January 2002.
- [18] B. K. Upadhyaya, "Design and Development of VHDL Based Tuneable Sine Wave Generator Using DDS Technique: Extension to FPGA Implementation", *NCC 2009*, January 16-18, IIT Guwahati.
- [19] ALTERA Cyclone III Design Guidelines, AN-4666-1.2, November 2008, URL: <http://www.altera.com/literature/an/an466.pdf>.