Bit Error Rate, Power and Area Analysis of Multiple FPGA Implementations of Underwater FSK

Ying Li*, Bridget Benson+, Ryan Kastner+, Xing Zhang*

*Peking University
NO.5 Yiheyuan Road, Haidian District
Beijing 100871, P.R.China
country code: 086
liying~, zhangx~@ime.pku.edu.cn

+University of California, San Diego
La Jolla, CA
USA
country code: 001
b1benson~ ,kastner~@cs.ucsd.edu

ABSTRACT

There has been an increasing interest in creating short-range, low data-rate underwater wireless sensor networks for scientific marine exploration and monitoring. However, the lack of an inexpensive, low-power, underwater acoustic modem is preventing the proliferation of these sensor networks. Thus, we are building an underwater acoustic modem that considers cost and power at every level, from the analog electronics, to the modulation scheme, to the hardware platform. In this paper, we use reconfigurable devices to explore the design space of our modulation scheme – Frequency Shift Keying (FSK) – to select an implementation that provides the lowest power and area without sacrificing reliability. Specifically, we explore the bit error rate, power and area tradeoffs of coherent and non-coherent FSK in response to varying baud rate, signal to noise ratio (SNR), synchronization errors, and Field Programmable Gate Array (FPGA) devices. We determine that although coherent FSK provides better bit error rate in the general case, non-coherent FSK shows similar bit error rate under high SNR and low baud rate parameters and saves nearly 4 times the hardware area over the coherent scheme on the same FPGA devices.

Categories and Subject Descriptors

B.4.1 [**Hardware**]: Data communications devices C.3 [**Special purpose and application-based systems**]: Real-time Systems

General Terms

Design, Experimentation, Reliability

Keywords

FPGA, FSK, coherent detection, non-coherent detection, underwater sensor networks

1. INTRODUCTION

Small, dense underwater sensor networks (UWSNs) have the potential to greatly improve environmental (pollution, coral reef, seismic, ocean current, etc.) and structural (oil platform, pipeline, undersea tunnel, etc.) monitoring which leads to greater understanding of our earth's bodies of water and the increased safety of mankind. These sensor networks are likely to have on the order of 10s to 100s of nodes spaced a relatively small distance apart (up to a few hundred meters) and produce relatively small amounts data (on the order < 1MB per day). Few of these networks currently exist because commercial off-the-shelf (COTS) underwater modems [1-3] (devices that actually receive and transmit data underwater) are not well suited for this application. The COTS modems' power consumption, ranges, and price points are all designed for sparse, long-range, expensive systems rather than small, dense, and cheap sensor-nets [4-6]. Therefore, a new low-cost (to allow for the deployment of 10s to 100s of nodes), low-power (to allow for long deployment) underwater acoustic modem must be designed. [7]

There are many design choices that must be considered when designing a low-cost, low-power underwater acoustic modem including, but not limited to, the implementation of the modulation scheme, the choice of underwater transducers and corresponding analog electronics, interfaces to sensors or higher level networking devices, and the suitable selection of a hardware platform for the implementation [8-18]. Each design choice is a research area in itself, so this paper focuses on the implementation of the modulation scheme – Frequency Shift Keying (FSK) – through the use of reconfigurable hardware.

FSK is a simple modulation scheme that has been widely used in underwater communications over the past two decades due to its resistance to time and frequency spreading of the underwater acoustic channel [14, 19]. Other modulation schemes such as direct sequence spread spectrum (DSSS) [12] and Orthogonal Division Frequency Multiplexing (OFDM) [15-16] are now being considered for higher data rate underwater applications, but the proven robustness of FSK and its simplicity still makes it an attractive modulation scheme for our low-cost, low-power, low-data rate application.

An FSK demodulator can be implemented with a coherent (requiring carrier phase tracking) or non-coherent (not requiring carrier phase tracking) structure and each implementation responds differently to varying levels of signal to noise ratio (SNR), baud rate, and synchronization errors. Selecting an appropriate low cost, power efficient, reliable implementation of an FSK demodulator can be difficult without prior knowledge of how different implementations respond to differing parameters.

Thus, we make use of reconfigurable hardware to test the power and area requirements and bit error rate response of coherent and non-coherent FSK hardware implementations in the presence of varying parameters. Reconfigurable hardware provides an excellent platform for design space exploration of low-power designs as it provides the low-power attributes of a custom hardware solution with the reprogramability of software making the task of finding a power effective solution for an intended application relatively easy. [20]

Thus, the purpose of this paper is to provide a bit error rate, power and area analysis of reconfigurable hardware implementations of FSK to select an appropriate hardware implementation for our low-cost, low-power underwater acoustic modem and to serve as a reference for other researchers on the capabilities of underwater FSK.

The major contributions of this paper are:

- A description of non-coherent and coherent FSK and their corresponding implementations on reconfigurable hardware
- A design exploration of power, area and bit error rate tradeoffs of coherent and non-coherent FSK in response to varying baud rate, SNR, synchronization errors, and FPGA devices
- A selection of the most reasonable FSK implementation for our low-power acoustic modem design based on experiments with real and simulated underwater data

The rest of the paper is organized as follows: Section 2 presents a high level description of our low-cost, low-power, underwater acoustic modem design. Section 3 describes coherent and non-coherent FSK and their corresponding implementations on reconfigurable hardware. Section 4 describes the design exploration of power, area, and bit error rate tradeoffs of coherent and non-coherent FSK in response to varying SNR, baud rate, synchronization errors and FPGA devices. Simulated and real underwater data are used in the exploration. We conclude with a discussion on future directions in Section 5.

2. LOW-COST, POWER EFFICIENT UNDERWATER ACOUSTIC MODEM

In order to make the use of short-range, low-data rate underwater sensor networks for environmental monitoring a reality, we are developing a low-cost, low-power underwater acoustic modem. The design consists of three main components as shown in Figure 1: **1.** The analog front end (dark gray), **2.** a hardware platform (light gray) and **3.** Serial interfaces (white).

The analog front end is responsible for converting electrical signals into sound waves and vice versa (transducer) and for generating the appropriate power level for the received and transmitted signals (analog electronics which include an amplifier, pre-amplifier, and transmit/receive switch). The hardware platform is responsible for control and signal processing, namely performing FSK modulation and demodulation and performing error encoding and decoding. The serial interfaces are responsible for communication with underwater sensors and/or higher level network layers.

Although most power and cost benefits will be gained in the design of the analog front end (as the analog front end is the most power consuming and costly portion of the underwater acoustic modem design), the designer must optimize the implementation at every level, from the analog electronics, to the signal processing scheme and the hardware platform, to achieve a truly low-cost, low-power design.

This paper fits into our overall goal of designing a low-cost, low-power underwater acoustic modem for the short-range, low data rate sensor network as it provides a means to select an implementation of our signaling scheme that matches the goals of our application.

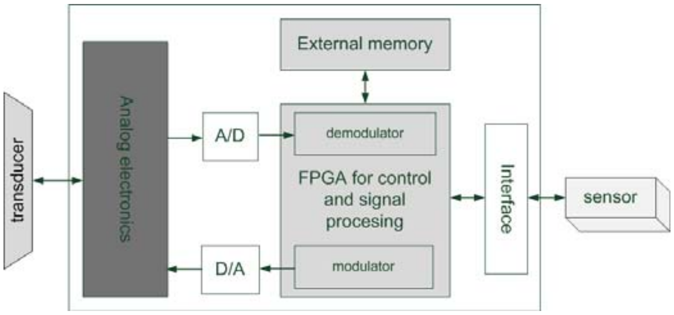

Figure 1. Major components of an underwater acoustic modem: the analog front end (dark gray) the FPGA platform (light gray) and serial interface (black)

3. THE COHERENT AND NON-COHERENT SCHEMES OF FSK DETECTION

In Binary FSK (referred to simply as FSK), the data are transmitted by shifting the frequency of a continuous carrier in a binary manner to one of two discrete frequencies. One frequency is designated as the “mark” frequency and the other as the “space” frequency [21-22]. The mark and space correspond to binary one and zero, respectively. There are two different types of detection schemes for FSK, coherent and non-coherent schemes, indicating the need for carrier phase tracking in the demodulator or not.

3.1 Coherent detection

Coherent demodulation requires channel state information. The classic ‘matched’ filter demodulator is optimal for coherent FSK detection with white Gaussian noise interference as shown in Figure 2 [21].

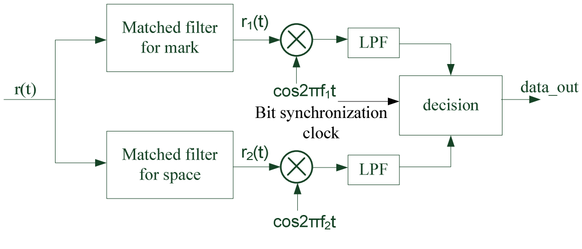

Figure 2. The coherent demodulator with matched filters

In the demodulator, the matched filters will filter out the off band frequencies of received signal r(t) with the center frequencies set as 'mark' and 'space'. The outputs of the matched band pass filters are multiplied by relative coherent carriers generated by a demodulator with the same carrier phase as the input signals. Then low pass filters are used to filter out the double frequency component. The output of low pass filters are sampled and compared by a bit synchronized clock to make a symbol decision. If the output from the mark filter branch is greater than that from the space filter branch, a decision is made that a mark signal was transmitted. On the other hand, if the output from the space filter branch is greater than that from the mark filter branch, a decision is made that a space signal was transmitted.

3.2 Non-coherent detection

The requirement of estimating the carrier phase for the branch signals can make coherent demodulation of FSK signals rather complex. Therefore, the alternative practical option is non-coherent detection. Demodulation of non-coherent FSK can be achieved by several demodulator structures, including envelope detection [22], difference detection and zero-cross detection [23-24]. Note that phase information is not required in any of them.

For envelop detection (shown in Figure 3), the outputs of the mark and space filters are envelope-detected and then compared by a bit synchronized clock pulse to determine which has greater magnitude. If the output from the mark envelope detector is larger than that from the space envelope detector, a decision is made that a mark signal was transmitted. On the other hand, if the output from the space envelope detector is greater than that from the mark envelope detector, a decision is made that a space signal was transmitted.

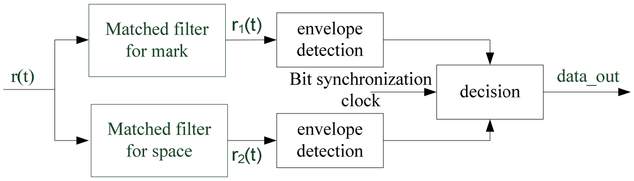

Figure 3. The non-coherent demodulator with envelope detection

The structure of difference detection is shown in Figure 4. The output of a band pass filter multiplies by a delayed version of itself. Then the doubling frequency is filtered out by a low pass filter. When a suitable delay value is selected, the output of the LPF can have a nearly linear relation with the angular frequency of input signal. Therefore, the decision block makes the final decision according to the difference between the angular frequencies of the two carriers.

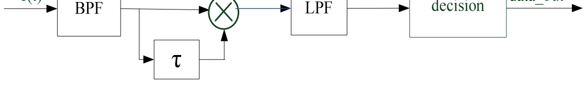

Figure 4. The non-coherent demodulator with difference detection

The flow of zero-cross detection is shown in Figure 5. The zero-cross detection technique is based on counting the zero-crossings of a frequency modulated signal in order to convert the frequency variations into voltage levels. Regions with more frequent zero crossings will have higher voltage levels and regions with fewer zero crossing will have lower voltage levels. The detector selects an offset between these voltages to make the symbol decision. If the voltage is higher than the offset, a decision is made that a mark signal was transmitted. If the voltage is lower than the offset, a decision is made that a space was transmitted.

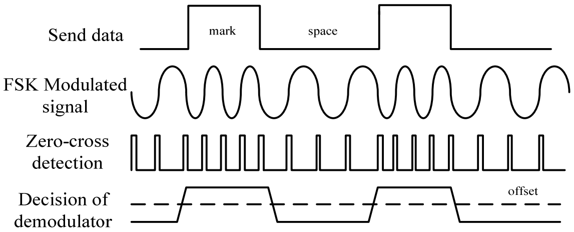

Figure 5. The non-coherent demodulator with zero-cross detection

3.3 Reconfigurable Hardware Implementation of Coherent and Non-Coherent FSK

We implemented the coherent FSK matched filter demodulator and the non-coherent zero-cross FSK demodulator in reconfigurable hardware. The reason we chose the zero-cross demodulator over the other demodulator structures was because earlier work showed it can attain a better bit error rate vs. SNR curve than the other conventional non-coherent demodulator structures [23] and its design was well suited to a hardware solution.

Our coherent matched filter demodulator uses matched Finite Impulse Response (FIR) filters designed using the Matlab filter design automation tool and implemented with only add and shift operations. Each filter has 30 orders of 16 bit fixed point coefficients and a pass band of 200 Hz centered by the mark or space frequency. No embedded multiplier is used in our coherent demodulator in order to save design space. We used the Matlab filter HDL coder to implement our low pass filters, which also have 30 orders of 16 bit fixed point coefficients. The decision module includes two accumulators in order to add the output of low pass filters in a symbol period, and a comparator to select the greater one of the accumulators to make the decision.

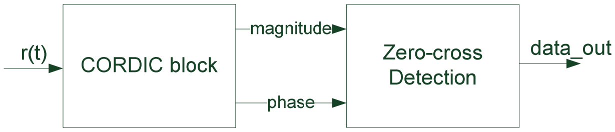

Figure 6. Zero-cross demodulator structure implemented on FPGA

Our non-coherent zero cross detector uses a COordinate Rotation DIgital Computer (CORDIC) algorithm,[25] which can perform the conversion between rectangular (sine and cosine) and polar (phase and amplitude) by several mathematic iterations. [24] The function of zero cross detector can be easily designed using polar objects. The CORDIC block calculates out the phase and amplitude information of input modulated sine type waveforms, which also produces simple phase-axis-crossing in the phase diagram. Since the phase varies from –л to л , when there is a phase transition from positive to negative or negative to positive,

a zero-crossing is counted. The zero- crossing count is accumulated and compared in one symbol period to the offset (as described in section 3.2) to make the final decision.
Theoretically, coherent detection offers a lower average bit error rate than non-coherent detection. The average bit error for coherent (1) and non-coherent (2) FSK detection in an Additive White Gaussian Noise (AWGN) channel when SNR>>1 are:

$$P_e = \frac{1}{\sqrt{2\pi r}} e^{-\frac{r}{2}} \quad (1)$$

$$P_e = \frac{1}{2} e^{-\frac{r}{2}} \quad (2)$$

where r is the ratio of signal and noise by power spectral density.

However, how different really are the bit error rates in a practical implementation? Does coherent or non-coherent FSK offer more power or area savings than the other without sacrificing reliability? Does coherent or non-coherent FSK perform better in the presence of synchronization errors or low SNR? We explore the answers to these questions in the next section.

4. BER, POWER and AREA ANALYSIS OF FSK FPGA IMPLEMENTATIONS

In this section we describe a series of experiments we conducted to explore the power, area, and bit error rate tradeoffs of our coherent and non-coherent FSK implementations in response to varying baud rate, SNR, synchronization errors, and FPGA devices. Our experiments make use of one 'real' underwater data set and two 'simulated' data sets to provide enough data for reasonable calculations of bit error rate.

The 'real' underwater dataset consists of the raw received signal of a 400 symbol sequence sent from the underwater transmit transducer to the underwater receive transducer in our underwater lab bench setup. It is important to test our FSK implementations with real underwater data as the underwater environment can have unpredictable affects on the signal. Though we wished to collect a larger amount of raw received data to process with our FSK implementations, 400 symbols was the maximum amount of raw data our data acquisition board could store.

Thus we simulated 10000 symbols of raw received data with varying levels of AWGN in Matlab to provide a test data set with enough symbols for a reasonable calculation of bit error rate and simulated the same 400 symbols of raw received data to provide a test data set that compares how well our simulated received data matches our real received data.

All of our test data sets employ a 1kHz space frequency and 2kHz mark frequency and provide perfect symbol synchronization with an SNR of 15dB (the ambient noise level) unless otherwise noted. The non-coherent and coherent demodulator implementations make use of a 1MHz sample clock. In both of the two demodulators, the detected sequence was compared with the transmitted binary sequence to calculate the bit error rate. If there is no bit error in a test, a minimum possible error value is set as 1/#symbols transmitted (i.e. 1/400 or 0.0025 for the 400 symbols tests). We use Modelsim and Xilinx ISE as our simulation and implementation tools and use the Xilinx power estimator for our power estimations.

4.1 Bit Error Rate vs. BPS

The first test we performed looks at the bit error rate response of coherent and non-coherent FSK in response to varying baud rate. We select baud rates suitable for our low-data rate modem, from 80 bits per second (bps) to 200 bps. Figure 7 shows the results of the experiment for the 400 symbol 'real' and 400 symbol 'simulated' data sets for both coherent and non-coherent detection. The dashed lines are for the 'real' data test and the solid lines are for the 'simulated' data test.

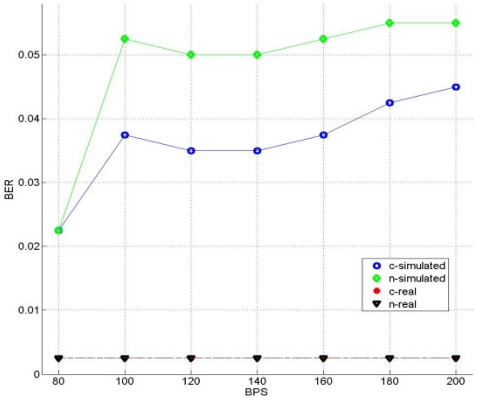

Figure 7. BER vs. BPS results for 400 symbol test sets

The results indicate that our 'real' signal achieves better BER performance than the simulated signal suggesting the sample size was too small to capture accurate results.
Therefore, we use the 10000 simulated symbols test to observe more convincing results. Figure 8 shows the result of the experiment for the simulated data sets for coherent and non-coherent detection. The dashed lines are for the 10000 symbol test and the solid lines are for the 400 symbol test. Because the 10000 symbol data set provides lower BER than the 400 symbol data set, the perfect detection for the real signal seems plausible.

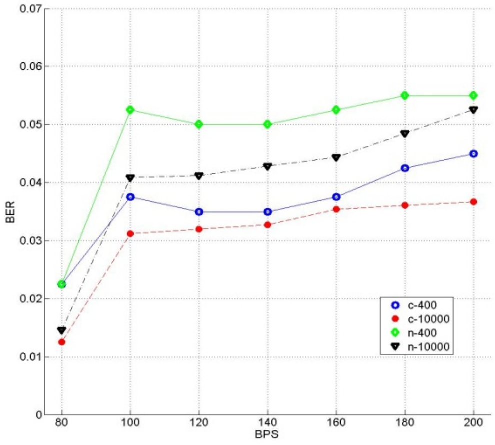

Figure8. BER vs. BPS results for simulated test sets

Figures 7 and 8 both show the error rate increased with rising BPS and coherent detection provides lower bit error rate than non-coherent detection in the general case. However, for 80 bps, the bit error rate between the coherent and non-coherent schemes is insignificant.

4.2 Bit Error Rate vs. SNR

Our second experiment uses our simulated data to analyze the response of non-coherent and coherent FSK to varying SNR. We applied different levels of AWGN to the simulated signals to achieve SNRs of 15dB, 18dB, and 20dB. Figure 9 shows the results of this experiment. The dashed lines are for the 10000 symbol test and the solid lines are for the 400 symbol test.

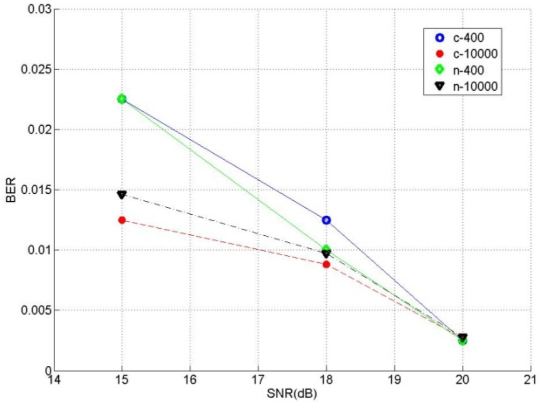

Figure9. BER vs. SNR results for simulated data

The results show that the error rate reduces significantly with rising SNR and the difference between coherent and non-coherent detection becomes insignificant. However, for lower SNR, coherent detection generally performs better than non-coherent detection.

4.3 Bit Error Rate vs. Synchronization Errors

Our third experiment looks at the bit error rate response of coherent and non-coherent FSK in response to varying synchronization errors in the 'real' and 'simulated' data sets. In order to introduce synchronization errors, we modify the data sets to start $^1/_8$ through the symbol and $^1/_4$way through the symbol representing being $^1/_8$ and $^1/_4$ out of synch respectively. Figure 10 shows the results of this experiment for the 400 symbol 'real' and 400 symbol 'simulated' data sets for both coherent and non-coherent detection.

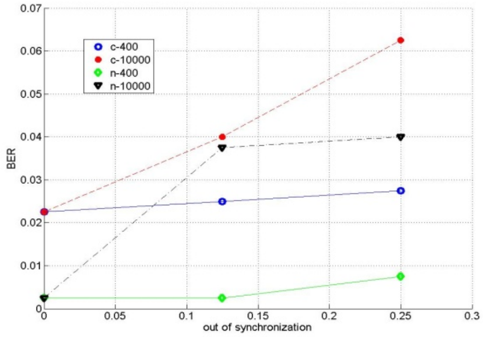

Figure 10. BER vs. Synch Error results for 400 symbol test sets

The results indicate that as expected, the error rate rose with increasing synchronization error in all cases. The 'real' signal again shows better BER than the corresponding 'simulated' signal likely due to the small sample size.

Figure 11 shows the results of this experiment for the simulated data sets for both coherent and non-coherent detection. The results also indicate that the error rate rose with increasing synchronization error in all cases. Both graphs show that the coherent scheme is more resistant to synchronization error as the bit error rate for the coherent scheme is lower than that of the non-coherent scheme for the out of synch simulations.

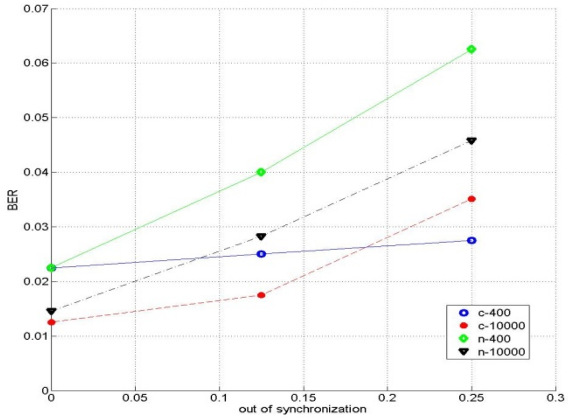

Figure 11.BER vs. Synch Error results for simulated test sets

From all the experiments above, we observe that the coherent scheme performs better than the non-coherent scheme in response to varying baud rate, SNR, and synchronization error in the general case. However, for synchronized, low baud rate, and high SNR, the difference in BER between the coherent and non-coherent scheme is insignificant. We observe that the 10000 symbol case matches more closely to our real signal test, suggesting the bit error rates reported for the 10000 symbol case may be close to the actual bit error rate in the underwater environment.

4.4 Non-coherent and Coherent Power and Area Requirements

Our final experiment looks at the power and area requirements of the coherent and non-coherent FSK when implemented on a variety of Xilinx FPGAs, ranging from the large Virtex IV device to the smallest Spartan-3 device. We select these devices because they are widely used in reconfigurable hardware applications and can provide a reference to select the lowest power device that fits the design. The area is represented as a percentage of the area used on the chip and the power is represented in Watts in Table 1.

Since the clock rate was very low in our tests (only 1MHz), the dynamic power consumptions for these two schemes were too small to tell the significant difference. As a result, the whole power was mainly determined by quiescent power, which was nearly the same per device. However, the area requirement for the coherent scheme is more than five times larger than that of the non-coherent scheme and cannot even fit on the smallest Spartan device. Thus where power is concerned, the non-coherent scheme offers the lower power solution because it can fit on a lower power device.

Table 1. Power and area implementation results for different FPGA devices

	Coherent		**Non-coherent**	
Devices	Power (W)	Area (%)	Power (W)	Area (%)
Virtex IV xc4vsx35	0.431	5%	0.43	1%
Spartan 3 xc3s1000	0.101	12%	0.10067	2%
Spartan 3 xc3s400	0.062	27%	0.06238	5%
Spartan 3 xc3s200	0.043	51%	0.04347	10%
Spartan 3 xc3s50	0.029	129%	0.02960	27%

Therefore, the series of experiments we conducted to explore the power, area, and bit error rate tradeoffs of our coherent and non-coherent FSK implementations in response to varying baud rate, SNR, synchronization errors, and FPGA devices clearly indicate that the coherent scheme offers better bit error rate performance than the non-coherent scheme in the general case, but comes at a cost of 5x the hardware space. The large area of the coherent scheme perhaps could be reduced by using smaller filters (e.g. filters implemented with a pipeline scheme) but this area reduction would come at a cost of additional timing delays and increased complexity for clock control. The non-coherent scheme performs just as well as the coherent scheme for the synchronized, low baud rate, high SNR situation and thus provides the better solution if these conditions can be met in the application.

5. CONCLUSIONS AND FUTURE WORK

This paper describes our work on analyzing FSK demodulator FPGA implementations for a low-cost, low-power underwater acoustic modem design. Two types of detection structures were implemented with different schemes: conventional matched-filter coherent detection and CORDIC non-coherent zero-crossing detection. We conducted a series of experiments with real and simulated underwater data to explore the power, area, and bit error rate tradeoffs of our coherent and non-coherent FSK implementations in response to varying baud rate, SNR, synchronization errors, and FPGA devices. The experiments clearly indicate that the coherent scheme offers better bit error rate performance than the non-coherent scheme in the general case, but comes at a cost of 5x the hardware space. The non-coherent scheme performs just as well as the coherent scheme for the synchronized, low baud rate, high SNR situation and thus provides the better solution if these conditions can be met in the application because the design can fit into a lower power device without sacrificing reliability.

This work is only a small piece of our entire low-cost, low-power underwater acoustic modem design. We are concurrently designing the other parts of the modem described in section 2 including the analog front end, other aspects of the control and signal processing scheme (including a digital up convertor and digital down converter, a symbol synchronization block and error coding), and the serial interfaces. Thus, the power and area results reported for different devices will help us select a suitable device for our entire design. Our hope is that by analyzing the power consumption at every design level while keeping costs in mind, we will be able to achieve a low-cost, low-power acoustic modem design that will make the proliferation of underwater sensor networks a reality.

6. ACKNOWLEDGMENTS

This material is based upon work partially supported by the China Scholarship Council and partially supported under National Science Foundation Grant #0816419 and National Science Foundation Graduate Research Fellowship.

7. REFERENCES

[1] LinkQuest, Inc. Underwater acoustic modems. http://www.link-quest.com/html/uwm_hr.pdf

[2] DSPCOMM, Underwater wireless modem.

[3] Benthos, Inc. Fast and reliable access to undersea data. http://www.benthos.com/pdf/Modems/ModemBrochure.pdf

[4] Sozer E. M. and Stojanovic M., Proakis J G. 2000. “Underwater Acoustic Networks”, IEEE Journal of oceanic engineering, Vol. 25, No. 1, (January 2000)

[5] Akyildiz I. F., Pompili D. and Melodia T. 2004. “Challenges for Efficient Communication in Underwater Acoustic Sensor Networks”, ACM Sigbed Review, Vol. 1, No. 2, (July 2004)

[6] Heidemann J., Li Y, Syed A., Wills J. and Ye W., "Research Challenges and Applications for Underwater Sensor Networking", Proceedings of the IEEE Wireless Communications and Networking Conference. April 2006.

[7] Proakis J. G., Sozer E. M., Rice J. A. and Stojanovic M. 2001. “Shallow Water Acoustic Networks”, IEEE Communications Magazine, (November 2001)

[8] Vasilescu I., Kotay K., Rus D., Dunbabin M. and Corke P. 2005. “Data Collection Storage, and Retrieval with an Underwater Sensor Network”, In Proc. of Sensys’05, San Diego, CA, (2005)

[9] Wills J., Ye W. and Heidemann J. 2006. “Low-Power Acoustic Modem for Dense Underwater Sensor Networks”, In Proc. of International Workshop on Underwater Networks (WUWNet),(September 2006)

[10] Jurdak R., Lopes C. V. and Baldi P. 2006. “Software Acoustic Modems for Short Range mote-based Underwater Sensor Network”, OCEANS 2006-Asia Pacific, (May 2006)

[11] Sozer E. M. and Stojanovic M. 2008. “Reconfigurable Acoustic Modem for Underwater Sensor Networks”, In Proc. of International Workshop on Underwater Networks (WUWNet), (September 2008)

[12] Iltis R. A., Lee H., Kastner R., Doonan D., Fu T., Moore R. and Chin M. 2005. “An Underwater Acoustic Telemetry

Modem for Eco-Sensing", In Proc. of MTS/IEEE Oceans'05, (September 2005)

[13] Utility Acoustic Modem available: auvlab.mit.edu

[14] Freitag L., Grund M., Singh S., Partan J., Koski P., and Ball K. 2005. "The WHOI Micro-Modem: An acoustic communications and navigation system for multiple platforms", In Proc. of OCEANS Conference, (September 2005)

[15] Yan H., Zhou S., Shi Z., and Li B. 2007. "A DSP implementation of OFDM acoustic modem", In Proc. of the ACM International Workshop onUnderWater Networks (WUWNet), (Montreal, Quebec, Canada, September 14, 2007)

[16] Kang T. and Litis R. A. 2008. "Matching Pursuits Channel Estimation for an Underwater Acoustic OFDM Modem" In Proc. of the 2008 IEEE International Conference on Acoustics, Speech, and Signal Processing Special Session on Physcial Layer Challenges in Underwater Acoustic Communications, (Las Vegas, Nevada, April 2008)

[17] Zhong K, Quek S. S., Koh T. A., Tan B. A. 2007. "A Real-Time Coded OFDM Acoustic Modem in Very Shallow Underwater Communications", OCEANS 2006 - Asia Pacific , (pp.1-5, 16-19 May 2007)

[18] Nasri N, Hnia H. B., Kachouri A., Abdellaoui M., Samet M. 2006. "Modulation/demodulation techniques with FPGA's architecture to improve OFDM wireless underwater communication transceiver", International Conference on Design and Test of Integrated Systems in Nanoscale Technology(DTIS 2006), (pp. 400-403, Sept. 5-7, 2006)

[19] Kilfoyle D. B., Baggeroer A. B. 2000. "The State of the Art in Underwater Acoustic Telemetry", IEEE Journal of Oceanic Engineering, Vol. 25, No.1, (January 2000)

[20] Benson B., Irturk A., Cho J. U. and Kastner R. 2008. "Survey of Hardware Platforms for an Energy Efficient Implementation of Matching Pursuits Algorithm for Shallow Water Networks", In Proc. of International Workshop on Underwater Networks (WUWNet), (September 2008)

[21] Shen Y. H., Li S. G., Yi X. X. and Qian Z. P. 2000. "Optimal Coherent Detection of FSK Signals With Imbalance Amplitudes in the Presence of Colored Gaussian Noise", Journal of electronics Bol. 17, No. 4, (October , 2000)

[22] Watson B. "FSK: Signals and Demodulation", The communications Edge TM

[23] Ismailoglu N., Yalcin T. 1999. "Low-power Design of a Digital FM Demodulator based on Zero-cross Detection at IF", VTC 99, Amsterdam, Netherlands, (September 1999)

[24] Cui X. X., Yu D. S., Sheng S. M., Cui X. L. 2006. "A CORDIC Demodulator Platform for Digital-IF Receiver", 8th International Conference on Solid-State and Integrated Circuit Technology, ICSICT'06, (October 2006)

[25] Andraka R. 1998. "A Survey of CORDIC Algorithms for FPGA Based Computers ", In Proc. of the 1998 ACM/SIGDA sixth international symposium on Field programmable gate arrays, (February 1998)