

LOW-POWER PACKET SYNCHRONIZATION SCHEME IMPLEMENTED ON FIELD
PROGRAMMABLE GATE ARRAY

by

CHARLES CARLSON

B.S., Kansas State University, 2013
B.S., Fort Hays State University, 2013

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2015

Approved by:

Major Professor
Dwight D. Day

Copyright

CHARLES CARLSON

2015

Abstract

Synchronization is one of the most critical steps in a wireless communication system. With the system having limited energy resources, low power devices and designs are key aspects of the design process. Digital communication and decoding is discussed along with how synchronization is part of communication. The parameters for wireless communication are outlined and how the system can be simplified in order to reduce power consumption for the network is investigated. The background for the Body Area Network Board which was created for the project, *Biosensor Networks and Telecommunication Subsystems for Long Duration Missions, EVA Suits, and Robotic Precursor Scout Missions*, is discussed along with some synchronization background as well as some previously researched demodulators designed for limited preambles.

With limited-length preambles, oversampling is needed to achieve synchronization. This research investigates what minimum oversampling ratio is needed in a simplified system to still achieve packet synchronization and several synchronization words were compared. The parameters for packet synchronization are outlined as well the impulse noise model used for simulation. For the simulation and the test setup, several oversampling ratios and synchronization words are compared using probability of miss detection and probability of false detection. The oversampling ratio of 16 was shown to be a critical point where increasing the oversampling rate above 16 had diminishing returns. In terms of probability of miss detection, the 7-bit Barker sequence along with the start of frame delimiter for IEEE 802.15.4 had better performance compared to the start of frame delimiter for Ethernet and the sequence 01010111.

Table of Contents

List of Figures	v
List of Tables	vii
Acknowledgements	viii
Dedication	ix
Chapter 1 - Introduction	1
Chapter 2 - Background	3
2.1 Kansas NASA BAN Board	3
2.2 Undersampling	6
2.3 Demodulators	6
2.4 Synchronization	7
Chapter 3 - Synchronization	10
3.1 Synchronization Word Length and Pattern	10
3.2 Oversampling	10
3.2.1 Noise Model	12
3.2.2 Simulation	13
Chapter 4 - Test Results	19
4.1 Test Setup	19
4.1.1 Selection of 1-bit ADC Sampling Frequency	22
4.1.2 Bit Pattern Generation	29
4.1.3 SNR Control	30
4.2 Tables and Plots of Results	31
4.3 Test Results Conclusion	35
Chapter 5 - Summary and Conclusions	37
Chapter 6 - Future Work	38
References	39

List of Figures

Figure 1: Kansas NASA BAN Board	3
Figure 2: Daughter Board with Sensor	4
Figure 3: RFIC Receive Mode Architecture	5
Figure 4: Standard Data Packet Format	7
Figure 5: One-Shot Data Packet Format	8
Figure 6: System overview to achieve synchronization using MLE	11
Figure 7: demodulator bit stream on oscilloscope affected by noise	12
Figure 8: longer data stream, RF signal gen at -77 dBm	13
Figure 9: Packet Structure for Simulations	14
Figure 10: Bit Stream with Impulse Noise	15
Figure 11: 7-bit Barker Sequence as Synchronization Word, Simulation Performance.....	17
Figure 12: 01010111 Sequence as Synchronization Word, Simulation Performance	17
Figure 13: Ethernet SFD as Synchronization Word, Simulation Performance	18
Figure 14: IEEE 802.15.4 SFD as Synchronization Word, Simulation Performance	18
Figure 15: Block diagram of test setup	19
Figure 16: Output of serial data generator on Oscilloscope	20
Figure 17: Visual Representation of Modulated Bit Stream (not to scale).....	21
Figure 18: Libero IDE Design Schematic, FPGA Verilog Design.....	22
Figure 19: Nyquist Folding diagram.....	23
Figure 20: Odd harmonics and possible interference between aliased signals	24
Figure 21: Frequency Phase Detector Operation	25
Figure 22: Up Pulses from PFD with transmission of a one, 25k data rate	26
Figure 23: PFD outputs and averaged output, 10k data rate	26
Figure 24: PFD outputs and averaged output, 20k data rate	27
Figure 25: PFD outputs and averaged output, 25k data rate	27
Figure 26: PFD outputs and averaged output, 30k data rate	27
Figure 27: PFD outputs and averaged output, 40k data rate	28
Figure 28: MATLAB plot with 6.15 MHz sampling frequency of 1-bit ADC.....	29
Figure 29: IF output on SA with RF generator set to -75 dBm	31

Figure 30: 7-bit Barker Sequence as Synchronization Word, Test Performance 33
Figure 31: 01010111 Sequence as Synchronization Word, Test Performance..... 33
Figure 32: Ethernet Start of Frame Delimiter as Synchronization Word, Test Performance 34
Figure 33: IEEE 802.15.4 Start of Frame Delimiter as Synchronization Word, Test Performance
..... 34
Figure 34: Oversampling ratio versus number of failed syncs per logic gates 36

List of Tables

Table 1: Simulation Results	16
Table 2: Synchronization words and the next sequence	30
Table 3: RF signal power into BAN board, measured SNR, SNR seen by receiver	31
Table 4: Test Setup Results.....	32
Table 5: Drop in Miss Detection vs Increase in Logic Gates, 7-bit Barker Sequence	35

Acknowledgements

Foremost, I would like to thank my major professor, Dr. Dwight Day, for his guidance and support throughout the research and writing process. His expertise not only helped me expand my knowledge during graduate school, but also helped me to think more deeply about synchronization within wireless communication.

Many thanks to my committee members, Dr. William Kuhn and Dr. Bala Natarajan for their insight while developing, reviewing, and finalizing my thesis.

I also want to acknowledge my research team members, Levi Riley and Chenyu Zheng. Task 3 of the project, *Biosensor Networks and Telecommunication Subsystems for Long Duration Missions, EVA Suits, and Robotic Precursor Scout Mission* was positively impacted by these outstanding individuals, and I'm grateful I had the opportunity to work with such a talented crew.

Most importantly, I would like to thank all my family for their continued support and words of wisdom. I couldn't have made it this far in my academic career without each of them, and I'm thankful they took this journey with me. To all of my friends, thank you for making these years an enjoyable experience.

Dedication

I dedicate this thesis to my parents.

Thank you for instilling a love for learning in me.

Chapter 1 - Introduction

Synchronization is the most critical step for a communication system. Without proper synchronization, devices from cellphones to satellites wouldn't be able to operate. Wireless networks such as Bluetooth or Zigbee aren't exempt and designing a simple/low-power system while still having reliable synchronization is a difficult task.

In order to simplify the synchronization process for a communication system, first consider the channel parameters that need to be estimated by a receiver in order to achieve synchronization. The channel parameters are timing, frequency, phase, and amplitude. Depending on the modulation scheme and choice of RF receiver, certain parameters would not need to be estimated. For Zigbee, which is based on IEEE 802.15.4, 40 bits are used for allowing a receiver to synchronize to a bit stream. This long synchronization header causes the packet to become less efficient and requires more power [1]. This thesis presents a packet synchronization method that will reduce overhead where the only parameter that needs to be determined is the timing parameter thereby simplifying the receiver architecture.

For a digital receiver, there are two sampling methods for timing recovery, synchronized and non-synchronized. Synchronized sampling is achieved with the use of a NCO (numerically controlled oscillator) that makes adjustments to the sampling based on some error signal. A non-synchronized sampling method samples the incoming bit stream based upon a fixed clock and is not locked to the incoming data stream [2]. The synchronized sampling method is typically more computationally expensive than the non-synchronized method, and is therefore, undesirable for a low-power network. In order to avoid the loss of information in the

data stream for a non-synchronized sampling system, oversampling is needed. According to the Nyquist theorem, the sampling rate must satisfy the following condition.

$$\left(\frac{1}{T_s}\right) \geq 2B_x \quad (1)$$

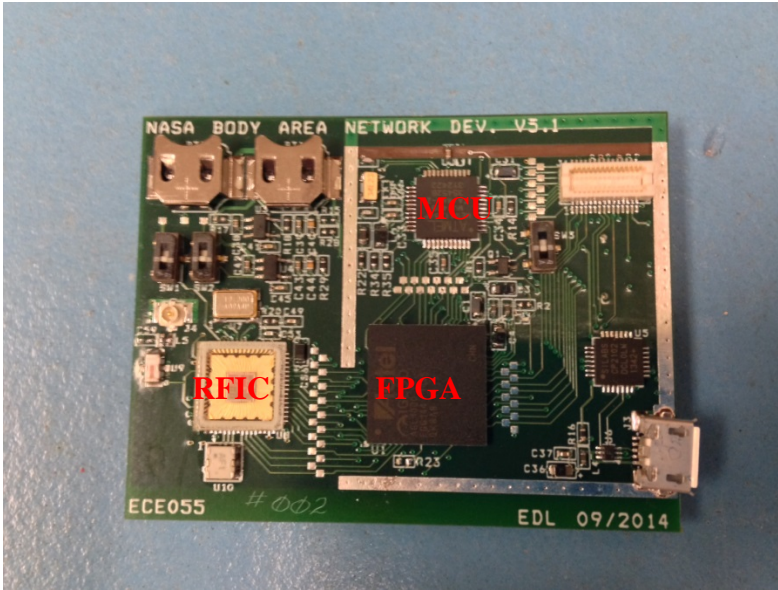
Where B_x is the bandwidth of the incoming signal. Therefore, for the non-synchronized system, a minimum oversampling ratio of two is needed. However, this thesis will investigate the effects of oversampling ratios at rates higher than the minimum needed to reconstruct the signal.

Chapter 2 - Background

2.1 Kansas NASA BAN Board

The Kansas NASA Body Area Network board was used for testing of the design and is shown below in Figure 1.

Figure 1: Kansas NASA BAN Board



The Kansas NASA Body Area Network was designed to be used for the implementation of a biosensor network as part of task 3 for the Biosensor Networks and Telecommunication Subsystems for Long Duration Missions, EVA Suits, and Robotic Precursor Scout Missions project, where task 3's goal was to develop the wireless network and its protocol. The Kansas NASA BAN board consists of three major IC components being a MCU, a FPGA, and an RFIC. When the board is in transmit mode, the MCU (microcontroller) controls a sensor located on the daughter board shown below in Figure 2 and communicates data from the sensor to the FPGA (Field Programmable Gate Array). The FPGA organizes the data into a packet and programs the RFIC (Radio Frequency Integrated Circuit) to transmit the data. When the board is in receive

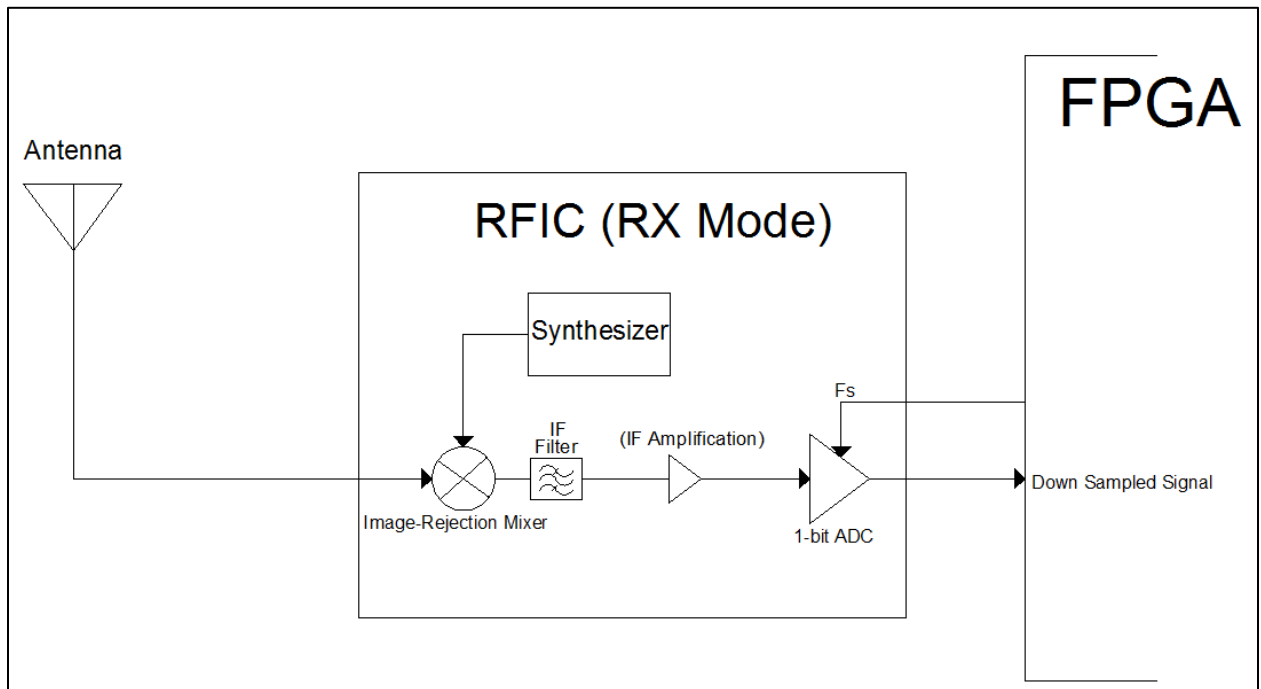
mode, the RFIC receives an incoming signal and sends the signal to the FPGA. The FPGA decodes the signal and sends the decoded data to the MCU.

Figure 2: Daughter Board with Sensor



With the chosen modulation scheme of BFSK (Binary Frequency Shift Keying), the receiver needs to be able to distinguish between the two transmitted frequencies in order to decode the received signal. In order to simplify the system, only one carrier frequency is used whereas most traditional communication networks use multiple carrier frequencies. Therefore, the carrier frequency will not need to be determined or estimated by the receiver. Figure 3 below shows the receiver architecture of the RFIC and the interface to the FPGA.

Figure 3: RFIC Receive Mode Architecture



The RFIC is a low power/mass/volume UHF radio originally created for mars missions. These characteristics make the RFIC a great choice for low power or energy harvesting systems. The receive chain of the RFIC includes a tuned-RF LNA (low noise amplifier) which is then followed by an image rejection mixer. After the signal has been mixed down to baseband, the signal is sent through an IF filter. The IF filter reduces the bandwidth of the down converted signal down to approximately 300 kHz. The signal is then amplified at the IF frequency of 10.7 MHz then converted to a digital signal by the 1-bit ADC. The sampling frequency for the 1-bit ADC shown above in Figure 3 as F_s , is controlled by the FPGA [3].

The FPGA on the Kansas NASA BAN board is the Actel IGLOO/e. The IGLOO FPGA's are some of the lowest power FPGAs available making them ideal for an energy harvesting network. The IGLOO FPGA's are flash based, which allows them to achieve ultra-low power without losing register data [4]. So why use an FPGA and a microcontroller? FPGA's offer some advantages over microcontrollers for a software defined radio.

FPGA Advantages

- Parallel execution
- Extremely configurable
- Different logic interfaces, I/O configurations
- Space limited, not time limited

Another advantage specific to the IGLOO FPGA's is that only the active portion of the FPGA will be powered and not the entire chip. Currently, the uC is only used to interface to the sensor on the daughter board. In future designs it could be possible to have the FPGA interface to the sensors and eliminate the uC thus making the NASA board more efficient and compact.

2.2 Undersampling

Utilizing a 1-bit ADC, the RFIC is able to reduce computation and have a lower power design without the loss of information. In [5] it was shown that a 1-bit ADC with down sampling does not degrade system performance even if a frequency offset exists compared to a 10-bit ADC without down sampling. Therefore there is no loss in information yet the system can operate at lower frequencies. The FPGA then demodulates the incoming 1-bit data stream.

2.3 Demodulators

In previous work done for Kansas NASA EPSCoR (Experimental Program to Stimulate Competitive Research) project, several demodulators were investigated. A PFD (phase frequency detector) usually incorporated in a PLL (phase lock loop) was shown to be an effective demodulation design. Thus, a simple PFD along with an averaging filter was used to decode the data into a bit stream. At this point, the only channel parameter that needs to be

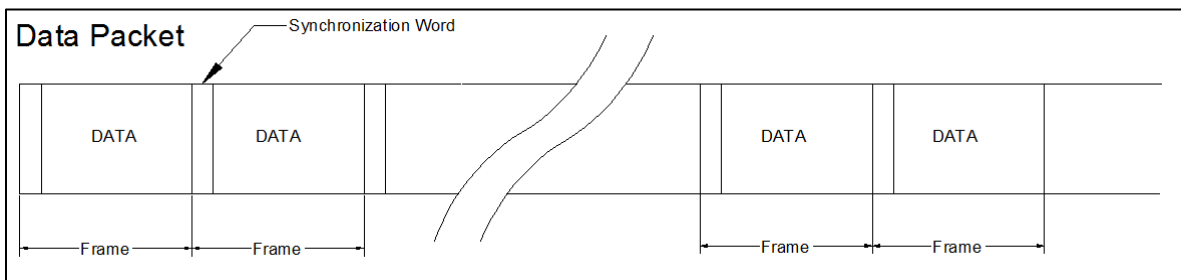
estimated is the timing parameter to achieve synchronization, and more precisely, one-shot packet synchronization.

2.4 Synchronization

James L Massey was one of the first to point out that “one-shot” synchronization where the synchronization word is either preceded by a preamble or nothing is an interesting problem to find the optimum synchronization rule [6]. Since his paper in 1972, several papers have been written that attempt to solve the issue of “one-shot” packet frame synchronization [7]. So what exactly is “one-shot” packet synchronization?

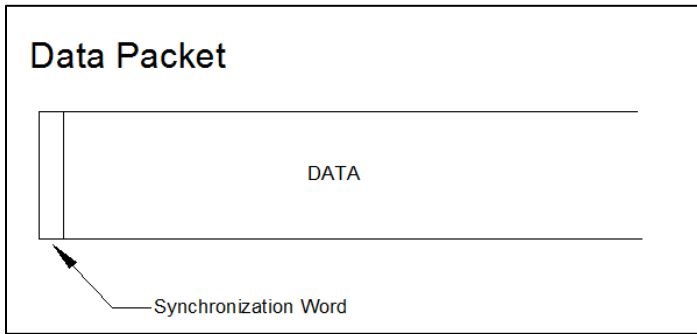
Frame synchronization is the process of using unique bits, a synchronization word, to synchronize the receiver to each frame within the incoming data stream. This is a critical issue for any communication system. As an example, in video transmission a packet of data is made up of several frames and these frames must be distinguishable from one another. Therefore, unique frame synchronization words are used to separate the frames. For MPEG-4 video encoding, resynchronization words are placed at bit intervals [8]. Figure 4 below shows a data packet with several frames of data.

Figure 4: Standard Data Packet Format



For the case of “one-shot” packet synchronization, the synchronization word is at the start of the data packet and is either preceded by a preamble or nothing. The entire data packet is treated as a single frame. Figure 5 below shows the case of “one-shot” packet synchronization.

Figure 5: One-Shot Data Packet Format



In order for the communication system to receive the incoming data stream, it is critical for the receiver to detect and sync to the synchronization word, otherwise, the entire data packet will be lost.

In most wireless network systems, the frame synchronization word, or start of frame delimiter, is preceded by a preamble at the beginning of the packet. The preamble is a sequence of bits that the receiver uses for carrier and bit synchronization. Carrier synchronization is typically acquired by the RF front end of the receiver. A reference carrier is generated at the receiver with a phase that is close to that of the received signal. A baseband signal is produced at the receiver through coherent demodulation of the incoming signal with the created reference. Bit synchronization is the process of aligning the receiver clock with that of the bit stream sequence. The main difference between bit/carrier synchronization and packet synchronization is that the issue of packet synchronization is usually solved with aid of a specific message format, synchronization word, which is inserted into the bit stream specifically for synchronization purposes whereas estimating bit and carrier synchronization does not always involve a special design of the data packet [9]. As mentioned above, a conventional receiver does not take advantage of the known preamble for synchronization purposes. In [10] a receiver design was used that utilized the known bit sequence of the preamble for packet synchronization. This type of process is known as two stage frame synchronization. The first stage tries to synchronize with

the known bit sequence of the preamble, and the second stage, conventional frame synchronization method, takes over once the preamble bit stream has been located. In essence, this two stage frame synchronization method could be viewed as simple packet synchronization with an extended synchronization word where the beginning of the synchronization word is also used for bit/carrier synchronization. The method in [10] was shown to provide 1.5 to 2 dB improvement than the traditional method.

A similar type of frame synchronization was investigated in [11]. Frame synchronization was achieved at the receiver through multiple frame observations. It was shown in [11] that using a multiple-frame decision method improved frame synchronization, however, having more than double frames did not result in system performance.

This thesis presents a packet synchronization process where only one synchronization word is used to achieve packet synchronization. As was mentioned above, in order to simplify the system, only one carrier frequency is used where in a typical communication system multiple carrier frequencies are used. For instance, IEEE standard 802.15.4 can operate at three different frequency bands, 868 MHz, 915 MHz, or 2450 MHz. Therefore, a preamble is currently not needed to achieve carrier synchronization, however, the RFIC has the capabilities to achieve carrier synchronization and this will be covered more in the future work section. By oversampling the bit stream, the bit rate and packet synchronization can both be estimated from the one synchronization word.

Chapter 3 - Synchronization

3.1 Synchronization Word Length and Pattern

There are two parameters to consider for one-shot packet synchronization, the length and bit pattern of the synchronization word. To make the data packet efficient, the length of the synchronization word should be kept as short as possible to reduce overhead. The synchronization word should also be chosen as to maximize the likelihood that the detector will find the sync word and to minimize the chance that the receiver will have a false detection of the sync word. In other words, the sync word should be chosen to maximize the probability of detection and minimize the probability of a false detection.

The study of binary sequences up to length 40 were studied in [12]. In [12] it is noted that Barker codes are optimal sequences for the use of sync words since Barker codes have been shown to have small sidelobes in their autocorrelation functions. In [7] several sync words including the 7-bit Barker sequence along with the 13-bit Neuman-Hofman sequence were used to test frame synchronization for asynchronous packet transmission. For the purpose of this thesis, the 7-bit Barker sequence was used along with the sequences, 01010111, the start of frame delimiter for Ethernet, and the start of frame delimiter for IEEE 802.15.4.

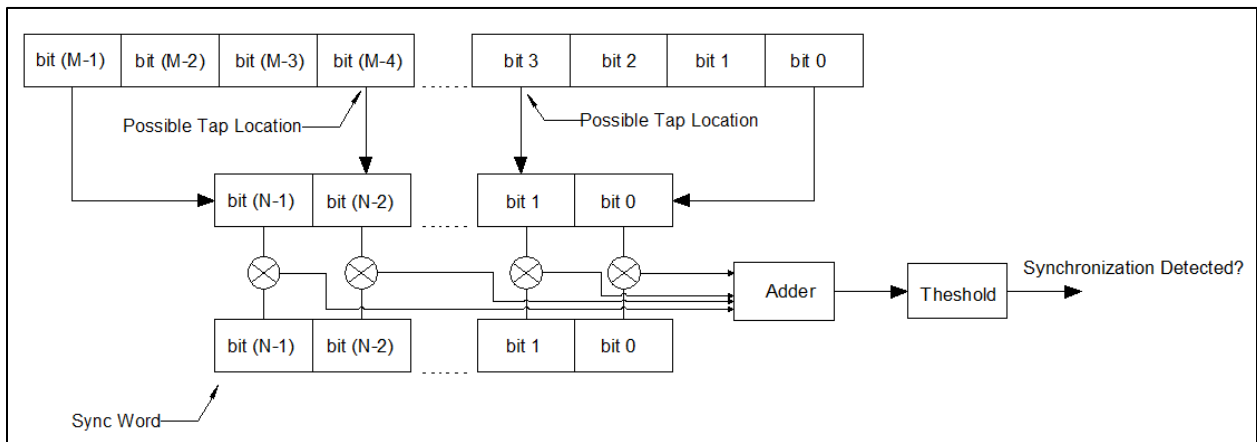
3.2 Oversampling

Without the aid of the preamble, the receiver must oversample the received bit stream in order to determine synchronization. According to [2], the oversampling ratio must be at least double the bandwidth of the incoming data stream.

To achieve packet synchronization at the receiver, the maximum-likelihood parameter estimation is used. The output data stream from the demodulator is oversampled at the rate set by the oversampling ratio. The sampled data is stored in a shift register whose size, M , is

determined by the number of transmitted bits and the oversampling ratio. Another shift register, whose size, N, is set by the number of transmitted bits, has values from specific tap locations of the shift register containing the sampled data. The tap locations are set by the data rate. An exclusive or operation is then performed bit by bit between the tap locations and the synchronization word. The result of the XORs are added together and a decision is made to determine if synchronization has occurred. Figure 6 below shows the design to achieve synchronization.

Figure 6: System overview to achieve synchronization using MLE



The length of first shift register, M, is determined by

$$M = (\text{oversampling ratio}) * (\text{length of sync word} - 1) \quad (2)$$

The length of tap locations, N is determined by

$$N = \text{length of sync word} \quad (3)$$

The tap locations are spaced apart by

$$\text{Spacing} = \text{Oversampling Ratio} \quad (4)$$

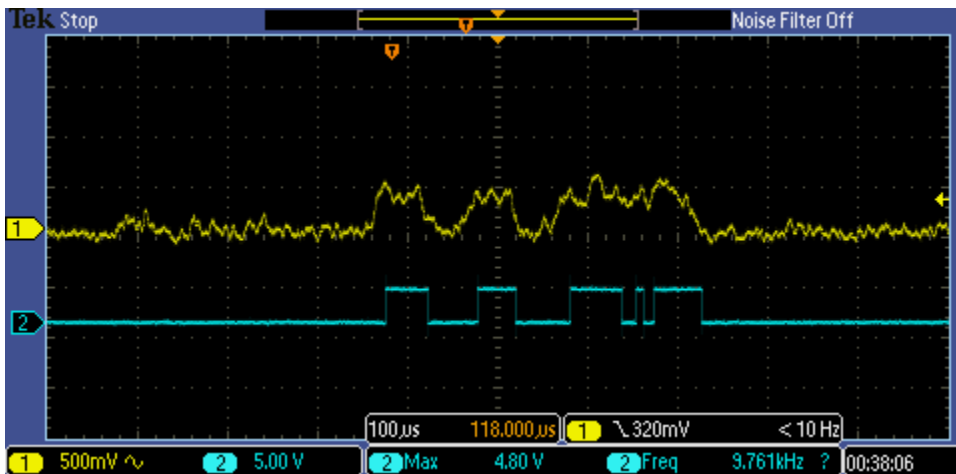
The above example considers that the transmitted data is only at one data rate. If K data rates are possible then there would need to be K tap locations whose spacing would correspond to the Kth data rate.

The parameter used to determine synchronization is the hamming distance, the number of bits that are different from the sync word. To determine if synchronization has occurred, an estimate of the hamming distance is performed through the use of the adder. The likelihood function is a function of the hamming distance, and the MLE (maximum likelihood estimate) is the estimate that minimizes this likelihood function, or in other words, the MLE will determine that synchronization has occurred when the hamming distance is below the threshold.

3.2.1 Noise Model

The simple demodulator is sensitive to low SNRs and the output bit stream from the demodulator is affected by noise. Shown below in Figure 7 is the bit stream output, in blue, from the demodulator on an oscilloscope along with the transmitted bit stream, 01010111, corrupted by noise before it enters the RF input of the BAN board, in yellow. The RF signal generator for Figure 7 is set to -76 dBm which corresponds to an SNR of approximately 6 dB seen by the receiver.

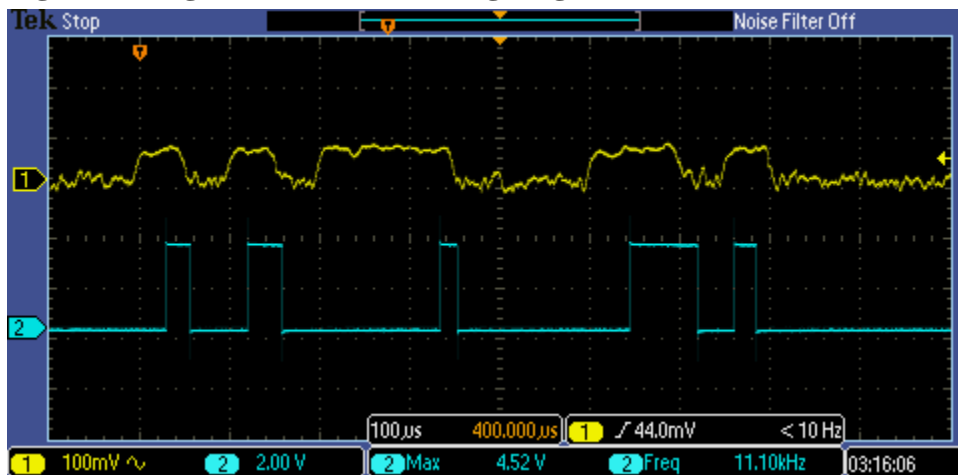
Figure 7: demodulator bit stream on oscilloscope affected by noise



As can be seen above in Figure 7, the bit stream output from the demodulator is not the correct sequence. The signal appears to have been corrupted by noise, and more precisely,

impulse noise. Shown below in Figure 8 is again the bit stream output, in blue, from the demodulator along the transmitted bit stream now being 0101011100011010 with the RF signal generator now set to -77 dBm which corresponds to an SNR of approximately 4 dB seen by the receiver.

Figure 8: longer data stream, RF signal gen at -77 dBm



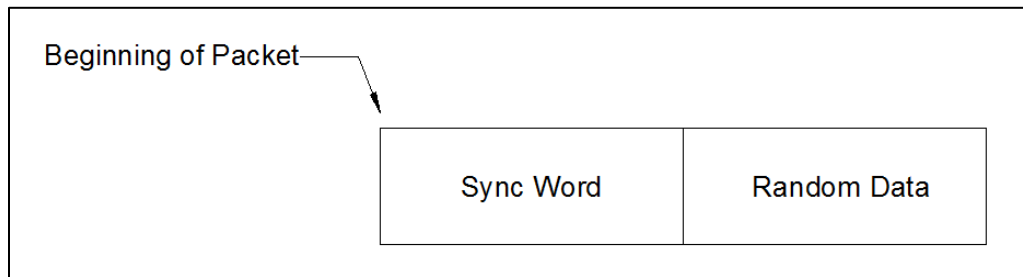
As the SNR gets lower, the impulse noise has a greater effect on the bit stream and the receiver can no longer correctly decode the data. A higher oversampling ratio has the ability to reduce the effect of the impulse noise.

Thus the minimum oversampling ratio of 2 might not be adequate to determine synchronization and decode the bit stream.

3.2.2 Simulation

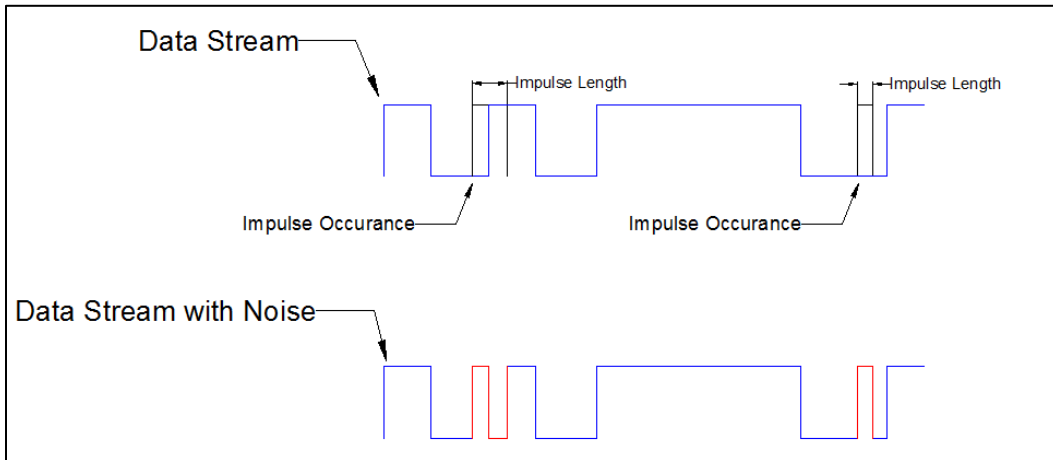
To model the system, first the packet structure must be determined. The packet structure is shown below in Figure 9. The first word of the packet is the synchronization word which is then followed by a random sequence of data that is the same length as the sync word.

Figure 9: Packet Structure for Simulations



Several synchronization words were compared against each other using probability of miss detection and probability of false detection. Three parameters are used to characterize the impulse noise. These parameters are impulse length, impulse occurrence, and number of impulses. The first two have units of time while the last parameter is an integer number. First the number of impulses was determined. Three different levels of impulse noise were simulated where the maximum number of impulses could be 5, 10, or 15. The number of impulses was determined by sampling from a uniform distribution that ranged from 0 to the maximum number allowed. Then the occurrence time was determined for each impulse by also randomly sampling from a uniform distribution that ranged from 0 to N , where N is the maximum time value of the bit stream. Finally, the length of each impulse was determined. The length for each impulse was again sampled from a uniform distribution from 0 to T_b , where T_b is the time it takes to transmit a single bit. Figure 10 below shows an example of how the data packet could be affected by impulse noise.

Figure 10: Bit Stream with Impulse Noise



Then the MLE method was used to determine if synchronization had occurred, and if so, did it occur during the correct interval. For example, suppose the synchronization word is 8 bits long, then the synchronization should occur during the transmission of the 8th bit. Otherwise, if the system decides synchronization outside this window, it will be counted as a false detection. For the simulation, 10,000 packets were sent and the number of missed synchronizations and the number of false synchronizations were tallied. Then it is a straight forward process to determine the probability of miss detection and false detection as follows

$$\text{Probability of Miss Detection} = \# \text{Missed Syncs} / \text{Total Sent} \quad (5)$$

$$\text{Probability of False Detection} = \# \text{False Syncs} / \text{Total Sent} \quad (6)$$

Table 1 below is a summary of the simulation results with the probability of failed synchronization circled in red for the oversampling ratio of 16 which will be shown in the results section to be a knee point in the data.

Table 1: Simulation Results

Max number of Impulses = 5			Max number of Impulses = 10			Max number of Impulses = 15		
10,000 Sync Words, 7-bit Barker sequence			10,000 Sync Words, 7-bit Barker sequence			10,000 Sync Words, 7-bit Barker sequence		
Oversampling ratio	Failed Syncs	False Syncs	Oversampling ratio	Failed Syncs	False Syncs	Oversampling ratio	Failed Syncs	False Syncs
64	7%	6%	64	20%	20%	64	32%	40%
32	9%	5%	32	22%	18%	32	33%	37%
16	11%	5%	16	25%	17%	16	37%	31%
8	14%	4%	8	31%	13%	8	44%	24%
4	21%	3%	4	41%	9%	4	54%	16%
10,000 Sync Words, 01010111			10,000 Sync Words, 01010111			10,000 Sync Words, 01010111		
Oversampling ratio	Failed Syncs	False Syncs	Oversampling ratio	Failed Syncs	False Syncs	Oversampling ratio	Failed Syncs	False Syncs
64	7%	10%	64	21%	19%	64	32%	27%
32	8%	9%	32	23%	18%	32	34%	25%
16	10%	9%	16	25%	16%	16	37%	21%
8	14%	7%	8	31%	12%	8	44%	16%
4	22%	6%	4	41%	9%	4	55%	10%
10,000 Sync Words, 10101011, Ethernet			10,000 Sync Words, 10101011, Ethernet			10,000 Sync Words, 10101011, Ethernet		
Oversampling ratio	Failed Syncs	False Syncs	Oversampling ratio	Failed Syncs	False Syncs	Oversampling ratio	Failed Syncs	False Syncs
64	8%	11%	64	19%	20%	64	32%	27%
32	8%	11%	32	22%	18%	32	34%	25%
16	10%	11%	16	26%	17%	16	37%	21%
8	14%	9%	8	31%	14%	8	44%	16%
4	21%	7%	4	41%	9%	4	53%	10%
10,000 Sync Words, 00010011, 802.15.4			10,000 Sync Words, 00010011, 802.15.4			10,000 Sync Words, 00010011, 802.15.4		
Oversampling ratio	Failed Syncs	False Syncs	Oversampling ratio	Failed Syncs	False Syncs	Oversampling ratio	Failed Syncs	False Syncs
64	7%	7%	64	21%	20%	64	32%	36%
32	8%	6%	32	22%	19%	32	33%	32%
16	11%	6%	16	26%	17%	16	37%	28%
8	14%	5%	8	31%	14%	8	44%	22%
4	20%	4%	4	40%	9%	4	53%	15%

As the oversampling ratio increased, the probability of miss detection was lowered and the probability of false detection was increased. This was expected since a higher oversampling ratio should lessen the effect of the impulse noise while at the same time increasing the probability of false detection since a higher oversampling ratio results in more observations of the data. None of the synchronization words tested stood out in terms of probability of false detection or probability of miss detection. Figures 11 through 14 below show a graphical representation of the data.

Figure 11: 7-bit Barker Sequence as Synchronization Word, Simulation Performance

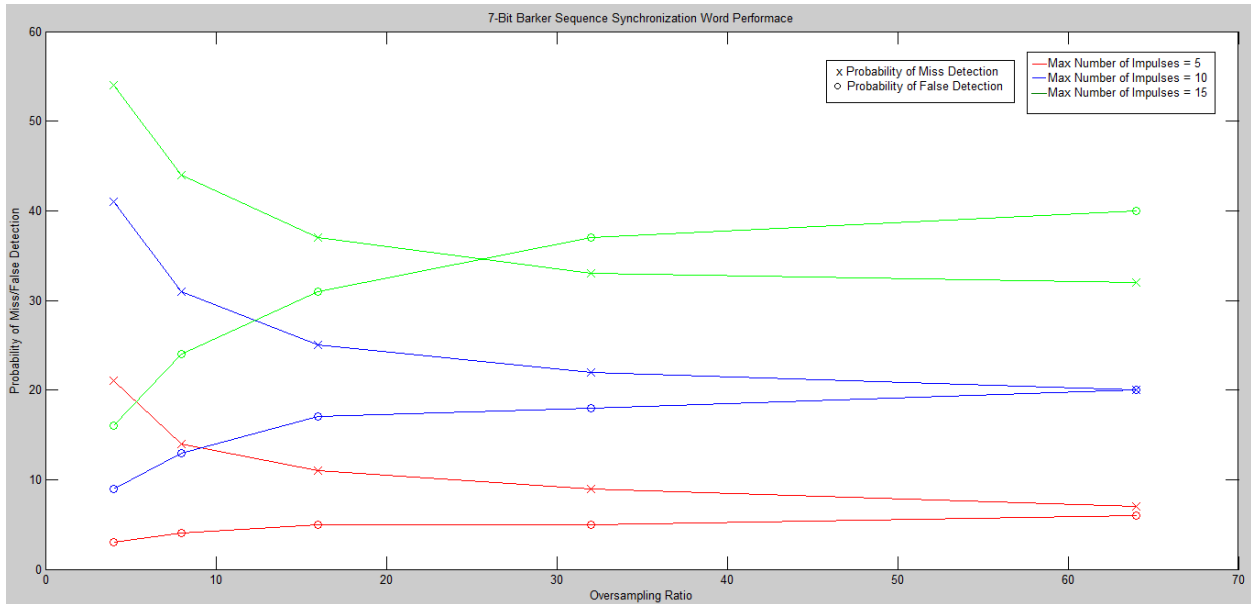


Figure 12: 01010111 Sequence as Synchronization Word, Simulation Performance

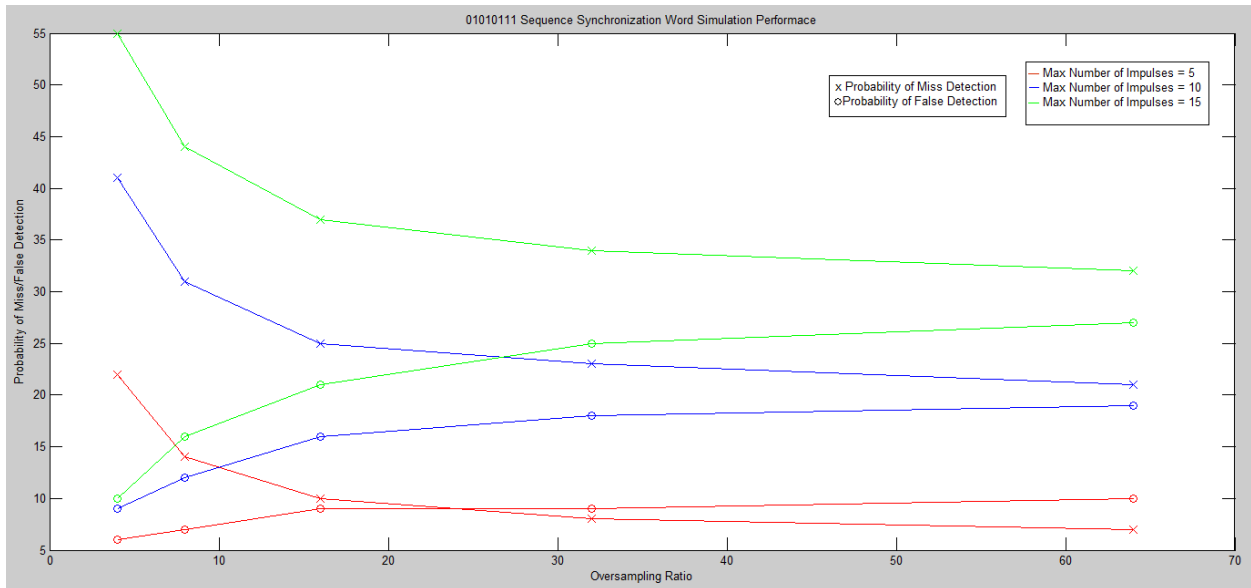


Figure 13: Ethernet SFD as Synchronization Word, Simulation Performance

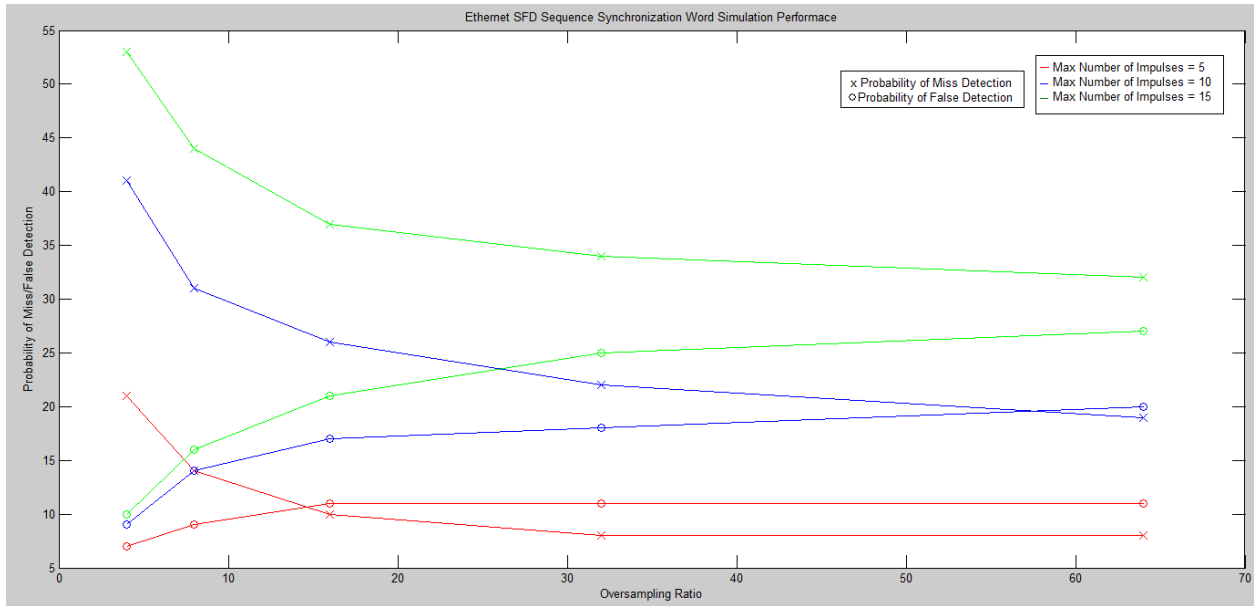
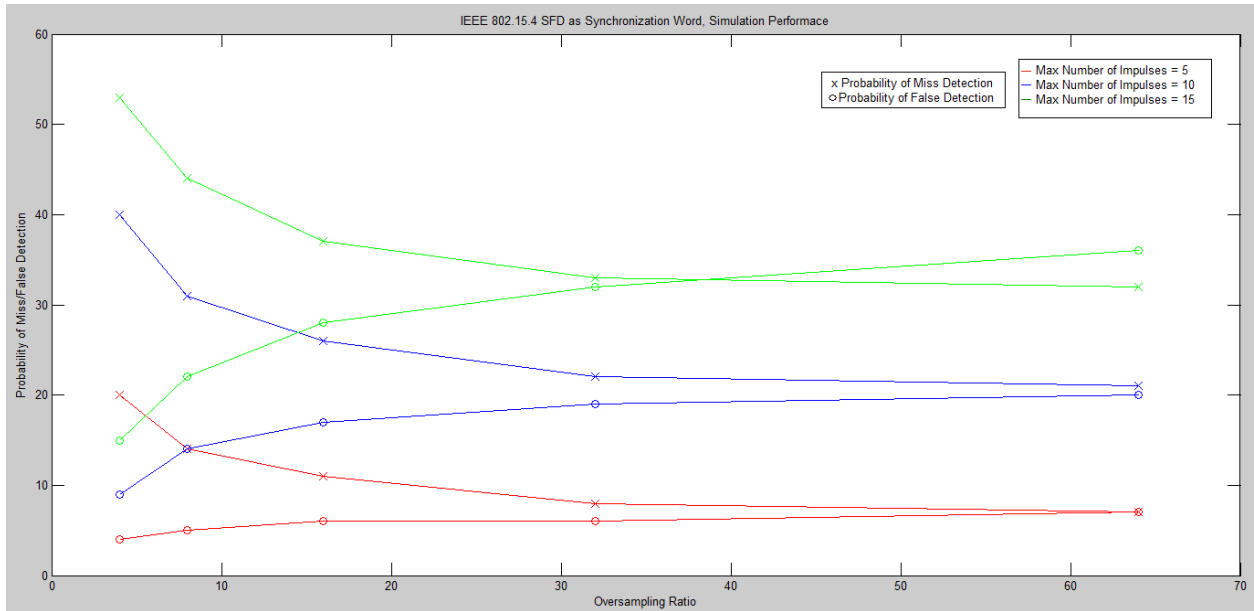


Figure 14: IEEE 802.15.4 SFD as Synchronization Word, Simulation Performance



The graphical representation of the data shows that there is a knee point at the oversampling ratio of 16. Having an oversampling ratio higher than 16 does not seem to lead to better system performance for the three different levels of simulated impulse noise.

Chapter 4 - Test Results

4.1 Test Setup

The communication parameters used for the test setup were as follows

- Burst FSK
- 20 kbps data rate
- Δf of 100 kHz
- f_1 set to 466.9 MHz
- f_2 set to 467 MHz

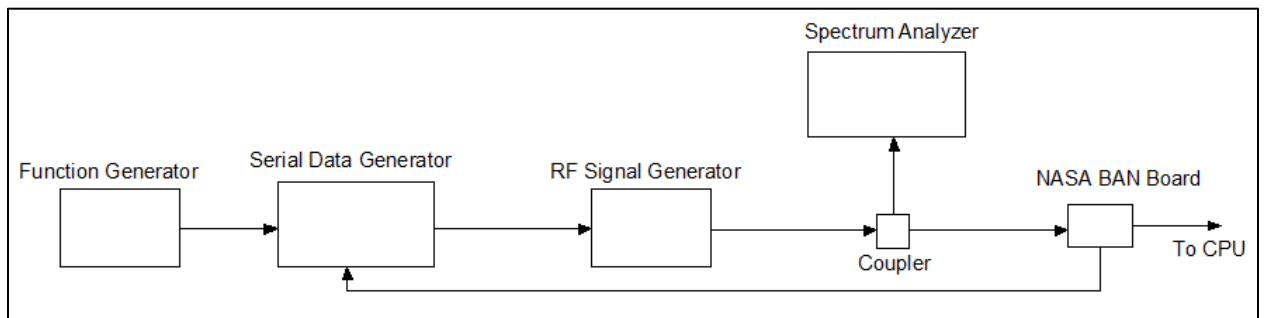
For the frame synchronization word, the 7-bit Barker sequence was used along with the sequence 01010111, the start of frame delimiter for Ethernet, and the start of frame delimiter for the standard IEEE 802.15.4. The goal of the test setup was to answer the following questions.

1. What is the lowest oversampling ratio needed to still detect the synchronization word while reducing impulse noise?
2. Which synchronization word has better performance?

Probability of miss detection and probability of false detection are used to determine the performance of the synchronization word.

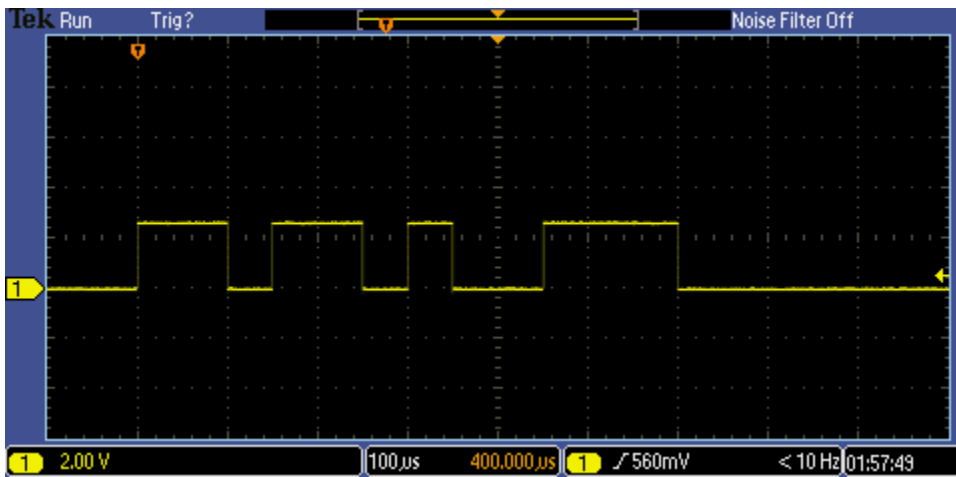
Figure 15 below shows a block diagram of the test setup.

Figure 15: Block diagram of test setup



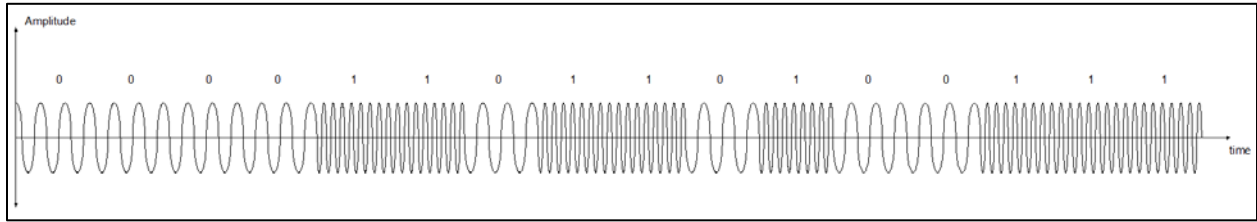
The HP 8081A was used as the serial data generator. In order to set the data rate, an Agilent 33220A 20 MHz Function / Arbitrary Waveform Generator was used as the clock input for the serial data generator and was set to 20 kHz. Once a pulse is received at the cycle input for the serial data generator, the stored word is output at the bit rate set by the function generator. Figure 16 below shows the output of the serial data generator on an oscilloscope with the Barker sequence as the synchronization word.

Figure 16: Output of serial data generator on Oscilloscope



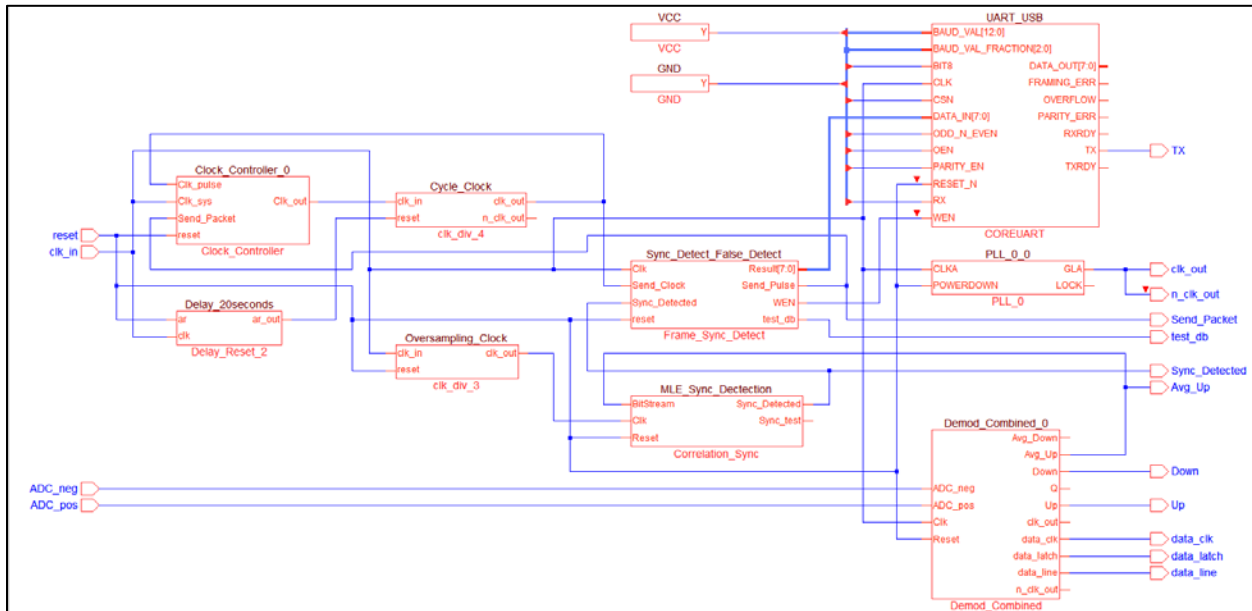
The serial data generator is used as the modulation input of the RF signal generator. The HP 8648C Signal Generator was used at the RF signal generator. The RF signal generator was set to output a frequency of 466.9 MHz. Thus if the bit is a zero, the output will be 466.9 MHz and if the bit is a one, the output will be 467 MHz. The Tektronix MSO 2012 is only rated for signals up to 100 MHz, thus the two RF signals could not be viewed on the MSO. A graphical representation is shown below in Figure 17 of the modulated bit stream.

Figure 17: Visual Representation of Modulated Bit Stream (not to scale)



The output power of the RF signal generator was set to three different power levels for the test setup; see Table 3 in the SNR control section. The RF signal was routed to a Mini-Circuits ZFDC-20-50 Directional Coupler. The out signal from the coupler which adds little attenuation to the signal was connected the spectrum analyzer to view the signal. The coupled output which adds approximately 20 dB of attenuation to the signal was connected to the RF input of the NASA BAN board. The video output of the spectrum analyzer was connected to the MSO input in order to see a decoded version of the transmitted bit stream. The FPGA Verilog design had the following block diagram. Figure 18 below is a screenshot from the Libero IDE design schematic.

Figure 18: Libero IDE Design Schematic, FPGA Verilog Design



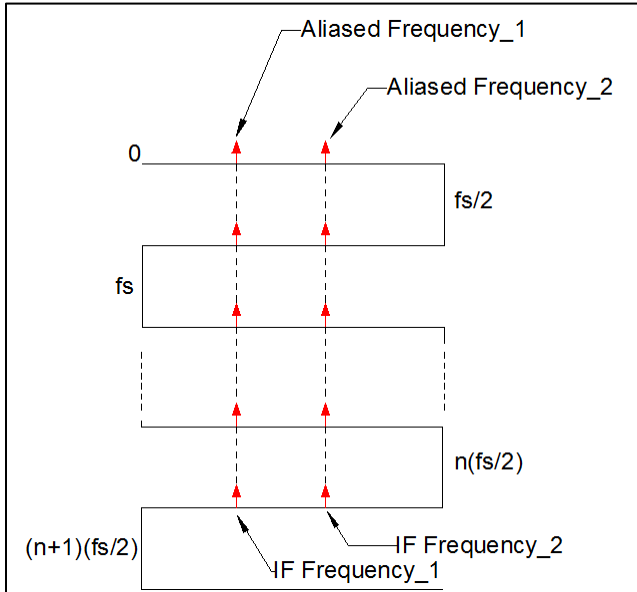
The Clock_Controller, Delay_20seconds, Cycle_Clock, Oversampling_Clock, and PLL modules are simply there for clock control. The UART_USB communicates to the CPU whether a synchronization was correct or false based on the output from the Sync_Detect_False Detect module. The MLE_Sync_Detection module performs the MLE of the received data and is the implementation of Figure 6, system overview to achieve synchronization using MLE. The Demod_Combined module programs the RFIC into receive mode through the RFIC's 61 bit register and also performs the demodulation of the received signal with the PFD and averaging filter. For system testing, ten thousand packets were sent and the number of miss and false detections were tallied. Then the probabilities were calculated and can be seen in Table 4 in the test results section.

4.1.1 Selection of 1-bit ADC Sampling Frequency

The frequency, 466.9 MHz, corresponded to the transmission of a zero and the frequency, 467 MHz, corresponded to the transmission of a one. Therefore, the synthesizer in the RFIC was programmed to 466.95 MHz. This resulted in a baseband signal of 10.65 MHz or 10.75 MHz at

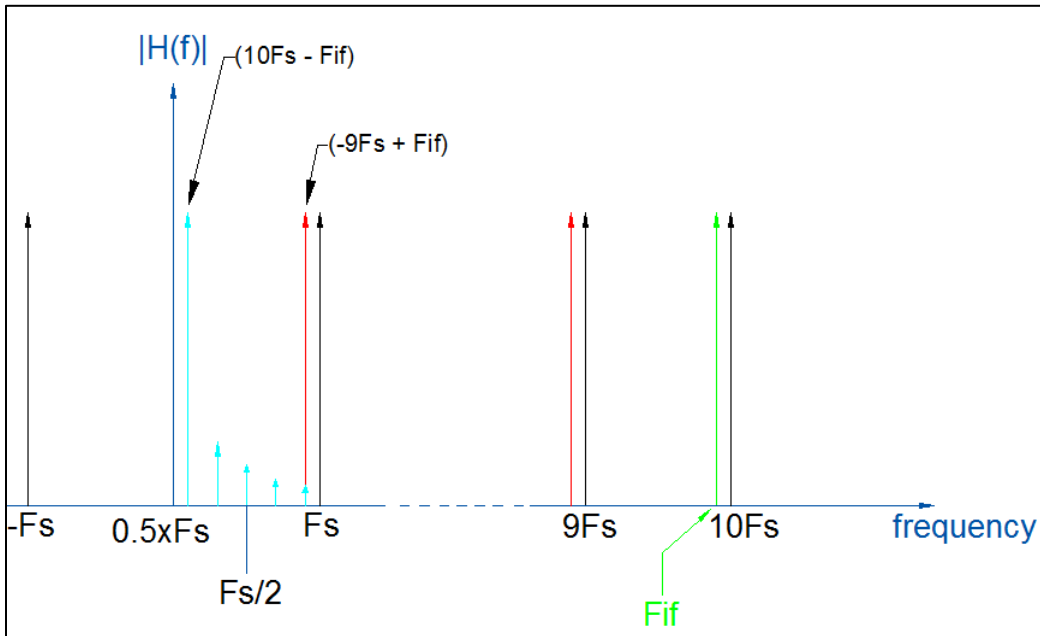
the IF stage. Then an undersampling frequency for the 1-bit ADC needed to be chosen. The Nyquist folding diagram was used to determine the sampling frequency, an example of Nyquist folding is shown below in Figure 16.

Figure 19: Nyquist Folding diagram



In the above figure, only the first harmonics are shown for the sampled frequencies and the aliased signals, however, since the signals are square waves, they have components at odd harmonics. Figure 20 below displays how the sampling/mixing of the harmonics could cause interference.

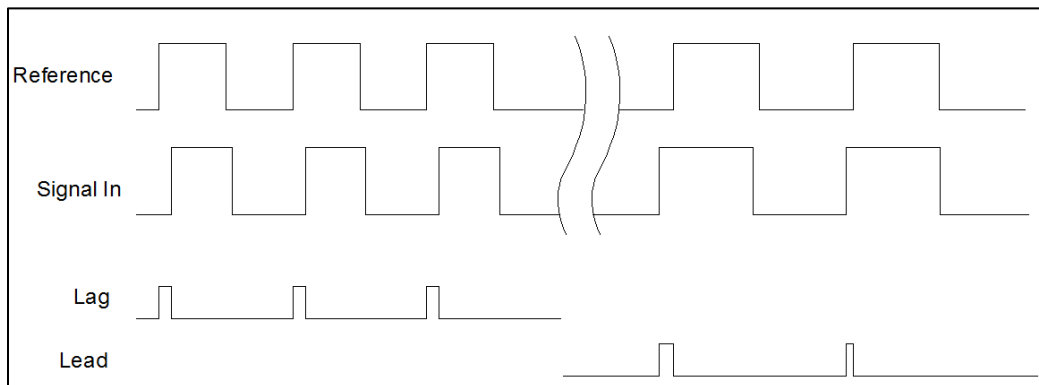
Figure 20: Odd harmonics and possible interference between aliased signals



In Figure 20 above only the harmonics of one of the aliased signals is shown, $10F_s - F_{if}$. Just from examining the harmonics of one of the aliased signals, it can be seen how the aliased signals of the harmonics could cause interference. Therefore, a sampling frequency should be chosen to attempt to lessen the interference. To reduce the interference effects, the aliased signals should be close to $f_s/4$, centered between 0 and $f_s/2$. This will widen the distance between the aliased signals on the folding diagram but will also increase the sampling frequency. Thus there is a tradeoff between higher frequencies, since the goal was to undersample and keep the frequencies low, and the amount of interference between the aliased signals.

Also, the Up and Down outputs from the PFD demodulator will be dependent on the aliased frequencies. Consider how the PFD generates the Up and Down pulses shown in Figure 21 below.

Figure 21: Frequency Phase Detector Operation

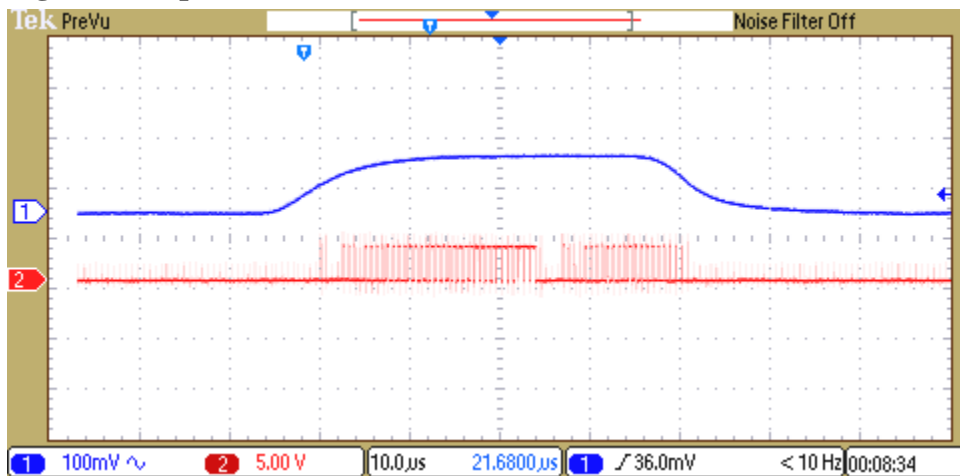


The duration of the up or down pulses is dependent on the phase difference between the aliased signal and the reference signal and the number of up/down pulses during a bit time will be related by the general rule

$$\text{Number of pulses} = (\text{aliased frequency} / \text{data rate}) \quad (7)$$

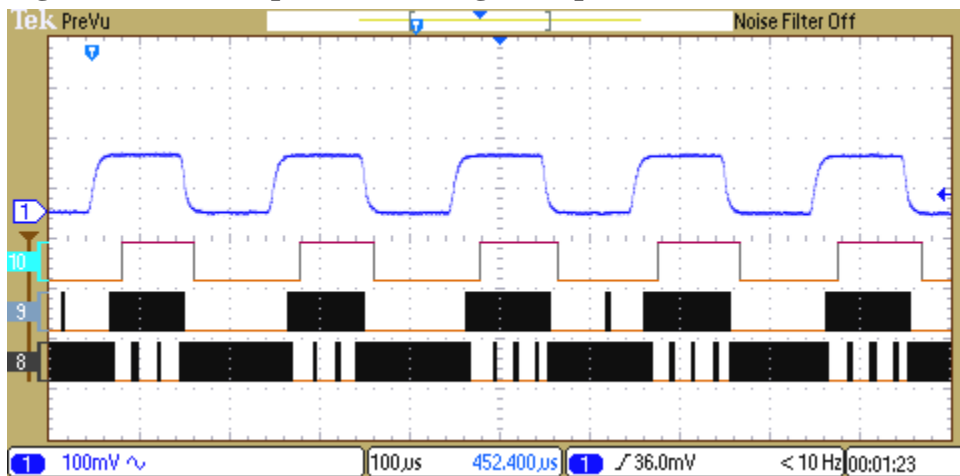
The desired data rate for the system was 50 kbps. With the aliased frequency of 1.6 MHz which will be derived below, the number pulses should be around 30. Therefore, the simple averaging filter was designed to output a one or zero by comparing the number of Up and Down pulses. Figure 22 below shows the Up pulses from the PFD when the transmission of a one at the data rate of 25 kbps.

Figure 22: Up Pulses from PFD with transmission of a one, 25k data rate



From equation 7, the number of up pulses should be around 64 and in the above figure, the number of up pulses is around 57. Shown below in Figure 23 are up (D9), down (D8), and output from the filter (D10) for the data rate of 10 kbps shown in blue.

Figure 23: PFD outputs and averaged output, 10k data rate



Shown below in Figures 24 through 27 are the up (D9), down (D8), and filter outputs (D10) for data rates of 20 kbps, 25 kbps, 30 kbps, and 40 kbps with the transmission of the sequence 1010101010101, shown in blue.

Figure 24: PFD outputs and averaged output, 20k data rate

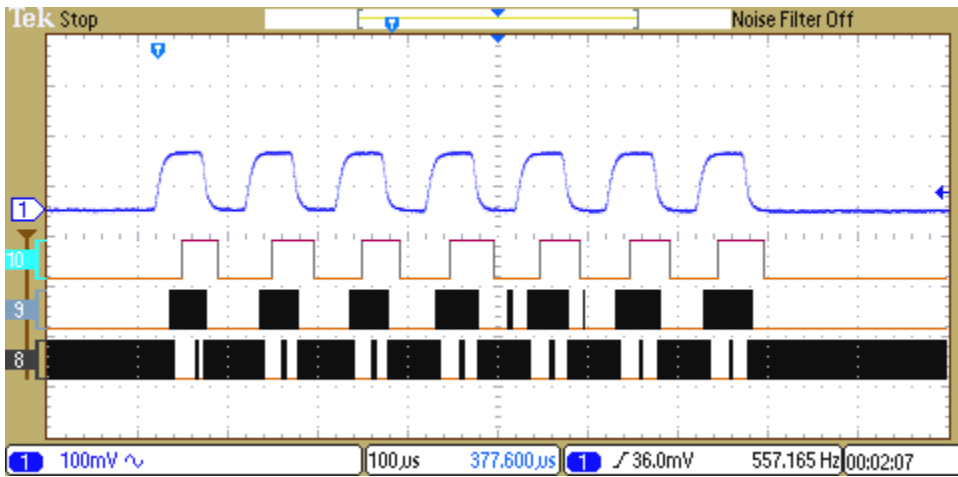


Figure 25: PFD outputs and averaged output, 25k data rate

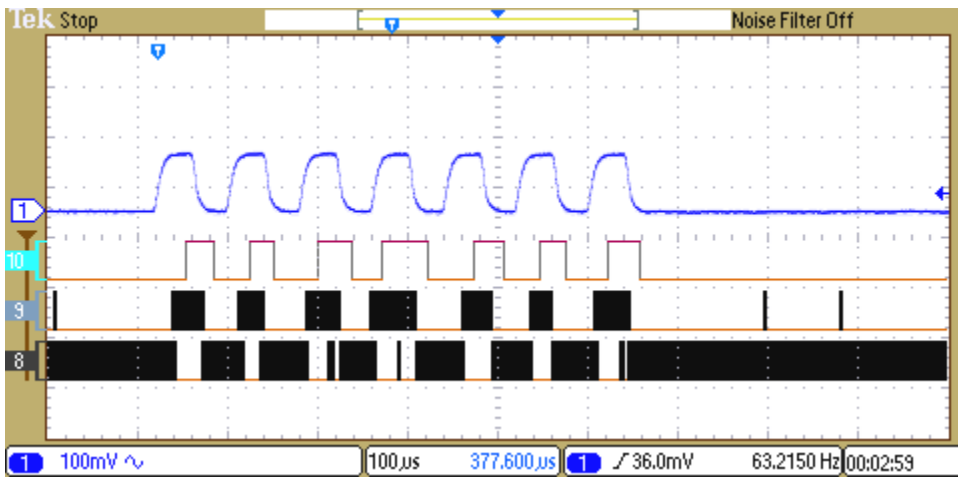


Figure 26: PFD outputs and averaged output, 30k data rate

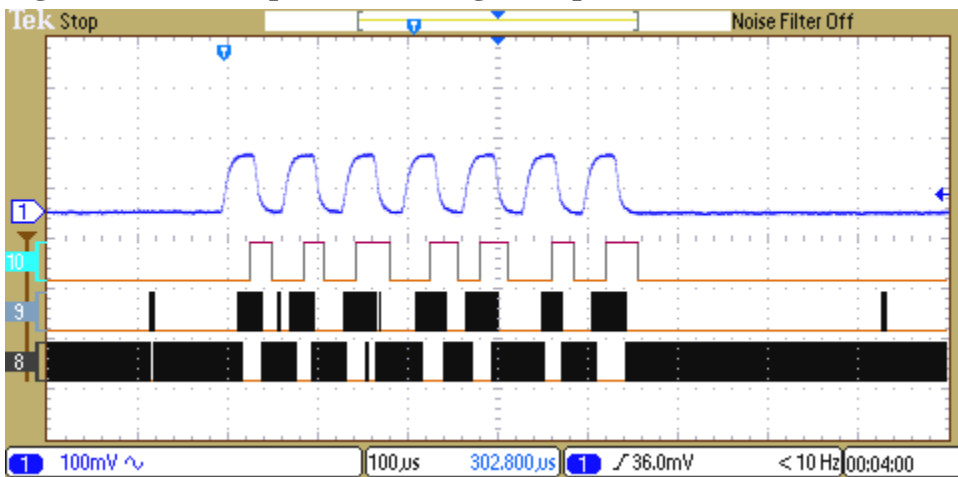
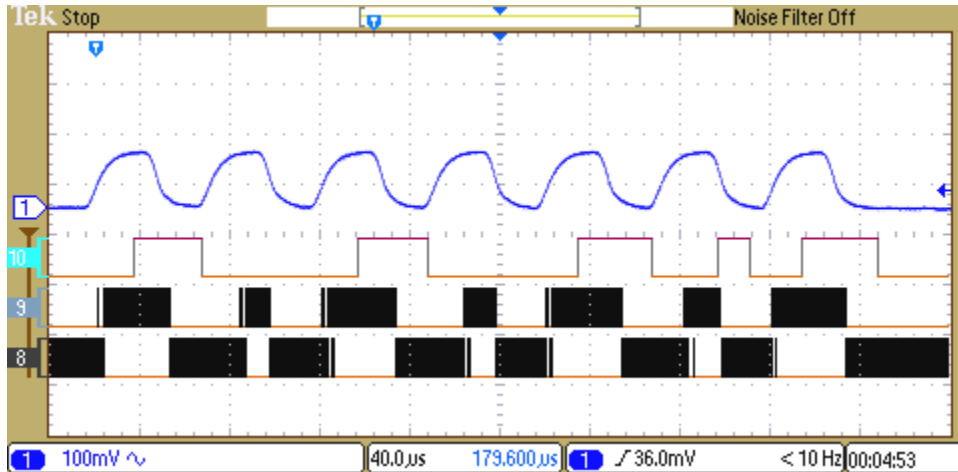


Figure 27: PFD outputs and averaged output, 40k data rate



The demodulator appears to work for data rates up to 30 kbps. For FSK (Frequency Shift Keying) the equation for determining the bandwidth needed to receive the signals is

$$BW = 2*\Delta f + 2*B \quad (8)$$

Where $2*\Delta f$ is the frequency difference between the two transmitted frequencies and B is the bandwidth of the baseband signal with $2*B$ being the data rate. For the above testing, Δf was 100 kHz and the bandwidth of the baseband signal changed with the data rate. The receiver is limited in bandwidth by the IF filter, which has a bandwidth of 300 kHz. Thus the theoretical maximum data rate is

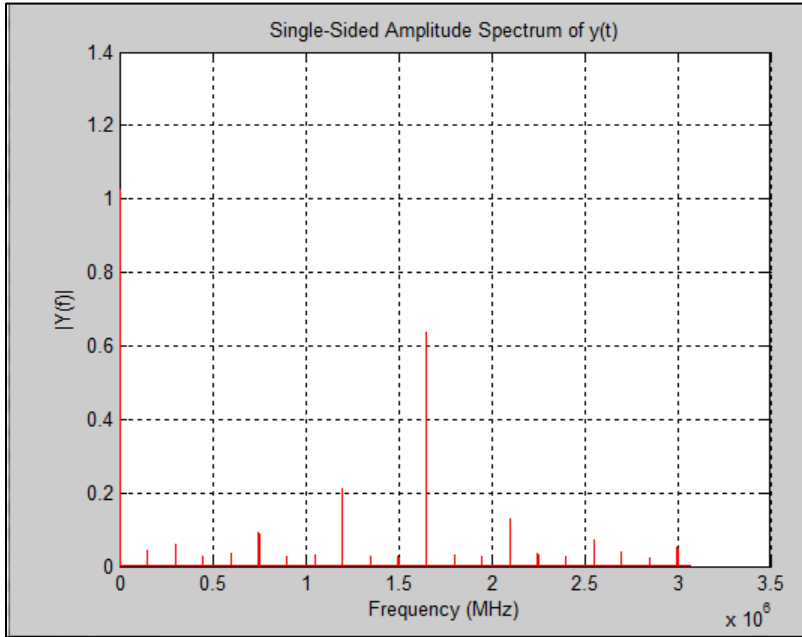
$$2B = BW (IF) - 2\Delta f = 300 \text{ kHz} - 100 \text{ kHz} = 200 \text{ kbps.}$$

So the simplified demodulator can only handle throughput at 30 kbps or 15% of the max capacity.

A higher aliased frequency will improve the PFD demodulator performance by increasing the number of either up or down pulses however this would counter the objective of undersampling. Thus there is a tradeoff between having lower frequencies or better PFD demodulator performance. Several sampling frequencies were simulated in MATLAB, and the sampling frequency of 6.15 MHz was chosen since it created aliased frequencies that limited

harmonic interference by generating aliased frequencies of 1.55 and 1.65 MHz where $f_s/4$ is 1.54 MHz. Shown below in Figure 28 is the simulated aliased signal, $y(t)$, down from 10.65 MHz for the sampling frequency of 6.15 MHz.

Figure 28: MATLAB plot with 6.15 MHz sampling frequency of 1-bit ADC



4.1.2 Bit Pattern Generation

The packet to be transmitted was chosen to be similar to the format used for simulations. A serial data generator was used to create the data stream. The serial data generator lacks the ability to create the random sequence that follows the synchronization word. Therefore, a constant sequence was used for testing purposes. The sequence was chosen so that no bias would be introduced into the results. If the data was randomly generated, the hamming distance would vary from 0 to 8, compared to the synchronization word. With the statistic used for determining synchronization as the MLE, where the hamming distance is used to determine synchronization, the sequence following the synchronization word was chosen to have a hamming distance of four

from the sync word. Table 2 below shows the sequence and corresponding synchronization word.

Table 2: Synchronization words and the next sequence

	Sync Word	Next Sequence
	01010111	00011010
Barker Sequence	00001101	10100111
Ethernet	10101011	01101101
802.15.4	00010011	10001001

4.1.3 SNR Control

The FPGA on a Kansas NASA BAN board first sends a pulse to the serial data generator. The pulse triggers the serial data generator to output the stored sequence. The sequence is used as the modulation input of a RF signal generator. The RF signal generator is setup to transmit 466.9 MHz if the bit is a zero and 467 MHz if the bit is a one. The RF signal generator is connected to a coupler. The coupler has two outputs, out and coupled. The out signal was routed to a spectrum analyzer and the coupled output, which approximately attenuates the signal by 20 dB, is connected to the RF input of the NASA BAN board. To vary the SNR (signal to noise ratio) seen by the receiver, the RF generator's output was adjusted and then the SNR was measured at the output of the IF filter on the BAN board. Three different RF signal outputs were used which corresponded to three different SNRs seen by the receiver. Table 3 below shows the various RF signal outputs and corresponding SNRs seen by the receiver. In order to calculate the SNR seen at the receiver, the IF output on the BAN was measured on a spectrum analyzer using a RF probe. Then the relationship between the measured SNR on the SA and the SNR seen by the receiver is

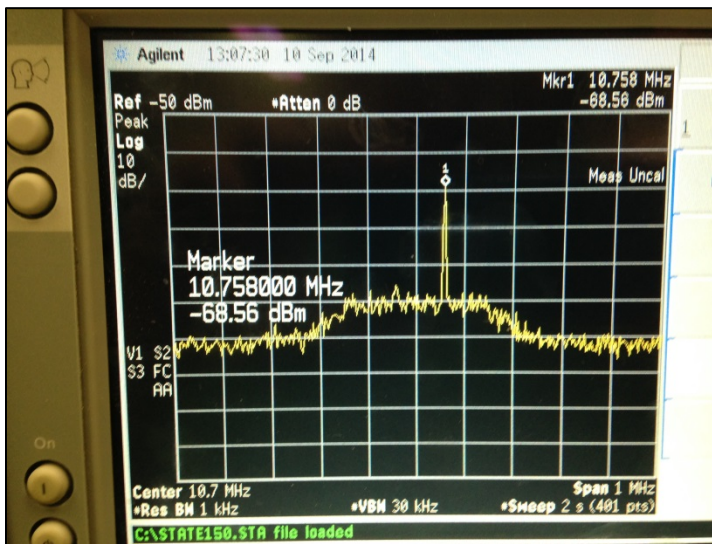
$$\text{SNR}_{\text{rx}} = (\text{SNR}_{\text{on SA}}) + 10 \cdot \log(\text{resolution}_{\text{bw}} / \text{measured IF}_{\text{bw}}) \quad (9)$$

Figure 29 below shows a screenshot of the IF output on a spectrum analyzer with the RF signal generator set to -75 dBm.

Table 3: RF signal power into BAN board, measured SNR, SNR seen by receiver

RF signal Power into BAN Board	Measured SNR (dB)	SNR (dB) seen by receiver
-80 dBm	43 dB	18 dB
-95 dBm	32 dB	7 dB
-97 dBm	29 dB	4 dB

Figure 29: IF output on SA with RF generator set to -75 dBm



In the above Figure, the measured SNR on the SA is approximately 32 dBm, the IF bandwidth is approximately 300 kHz, and the resolution bandwidth is 1 kHz.

4.2 Tables and Plots of Results

Table 4 on the next page shows the test setup results with the 16 times oversampling ratio for the failed synchronizations circled in red.

Table 4: Test Setup Results

SNR = 18 dB (RF_gen at -60dBm)			SNR = 7 dB (RF_gen at -75dBm)			SNR = 4 dB (RF_gen at -77dBm)		
10,000 Sync Words, 7-bit Barker sequence			10,000 Sync Words, 7-bit Barker sequence			10,000 Sync Words, 7-bit Barker sequence		
Oversampling ratio	Failed Syncs	False Syncs	Oversampling ratio	Failed Syncs	False Syncs	Oversampling ratio	Failed Syncs	False Syncs
60	1%	6%	60	35%	19%	60	61%	16%
32	1%	6%	32	38%	18%	32	66%	15%
16	1%	4%	16	33%	17%	16	58%	16%
8	2%	2%	8	40%	14%	8	65%	14%
4	6%	1%	4	51%	10%	4	74%	10%
10,000 Sync Words, 01010111			10,000 Sync Words, 01010111			10,000 Sync Words, 01010111		
Oversampling ratio	Failed Syncs	False Syncs	Oversampling ratio	Failed Syncs	False Syncs	Oversampling ratio	Failed Syncs	False Syncs
60	3%	3%	60	59%	1%	60	86%	0%
32	3%	3%	32	61%	1%	32	87%	0%
16	3%	3%	16	59%	1%	16	84%	0%
8	4%	2%	8	60%	1%	8	86%	0%
4	5%	1%	4	64%	1%	4	87%	0%
10,000 Sync Words, 10101011, Ethernet			10,000 Sync Words, 10101011, Ethernet			10,000 Sync Words, 10101011, Ethernet		
Oversampling ratio	Failed Syncs	False Syncs	Oversampling ratio	Failed Syncs	False Syncs	Oversampling ratio	Failed Syncs	False Syncs
60	7%	5%	60	65%	4%	60	87%	1%
32	7%	3%	32	69%	3%	32	89%	1%
16	7%	3%	16	64%	3%	16	87%	2%
8	13%	5%	8	65%	4%	8	86%	3%
4	10%	1%	4	71%	2%	4	90%	1%
10,000 Sync Words, 00010011, 802.15.4			10,000 Sync Words, 00010011, 802.15.4			10,000 Sync Words, 00010011, 802.15.4		
Oversampling ratio	Failed Syncs	False Syncs	Oversampling ratio	Failed Syncs	False Syncs	Oversampling ratio	Failed Syncs	False Syncs
60	1%	13%	60	34%	8%	60	60%	4%
32	2%	14%	32	35%	8%	32	68%	4%
16	2%	18%	16	34%	9%	16	60%	4%
8	2%	21%	8	38%	9%	8	65%	4%
4	2%	34%	4	44%	12%	4	70%	5%

The graphical representation of the data can be seen in Figures 30 to 33, page 33 and 34.

Figure 30: 7-bit Barker Sequence as Synchronization Word, Test Performance

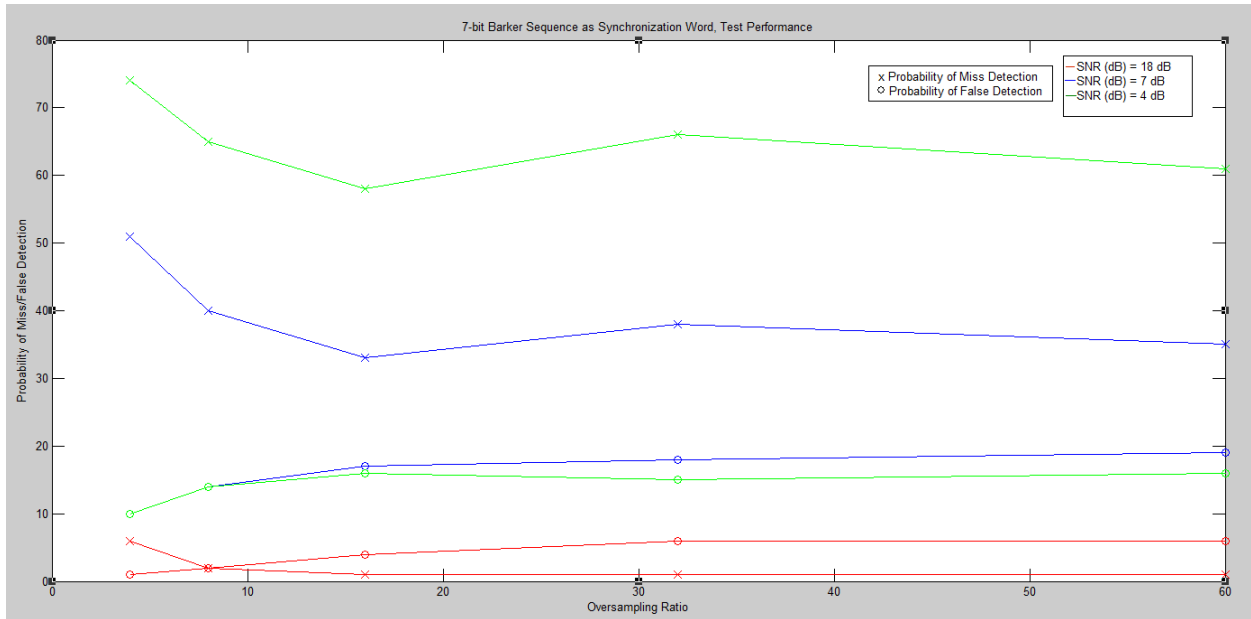


Figure 31: 01010111 Sequence as Synchronization Word, Test Performance

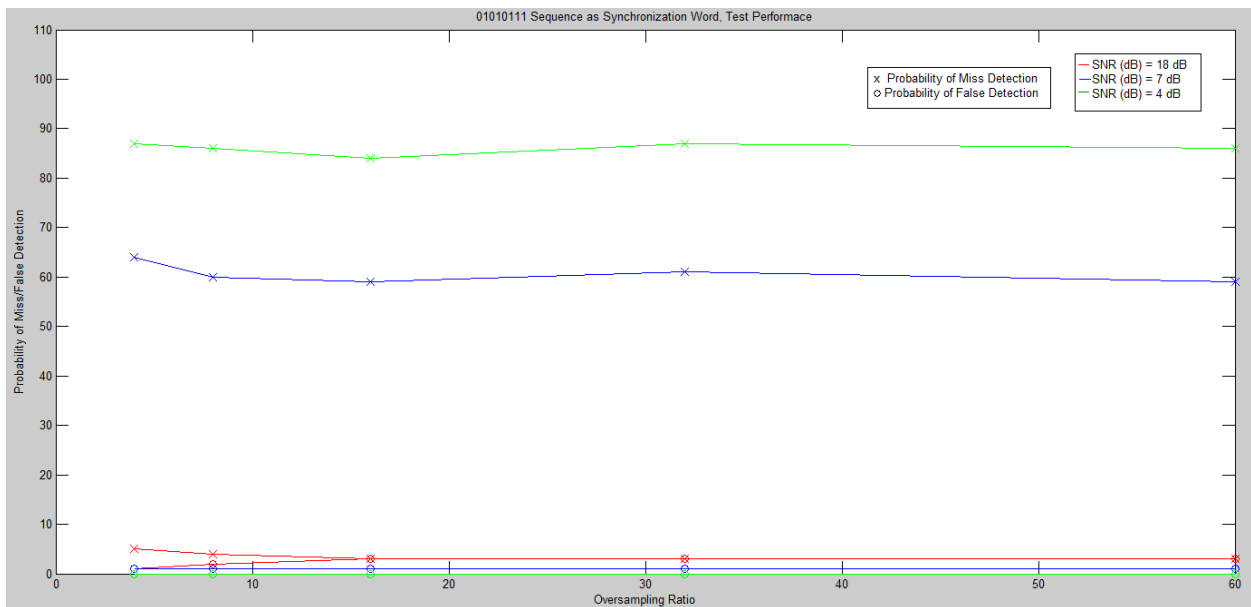


Figure 32: Ethernet Start of Frame Delimiter as Synchronization Word, Test Performance

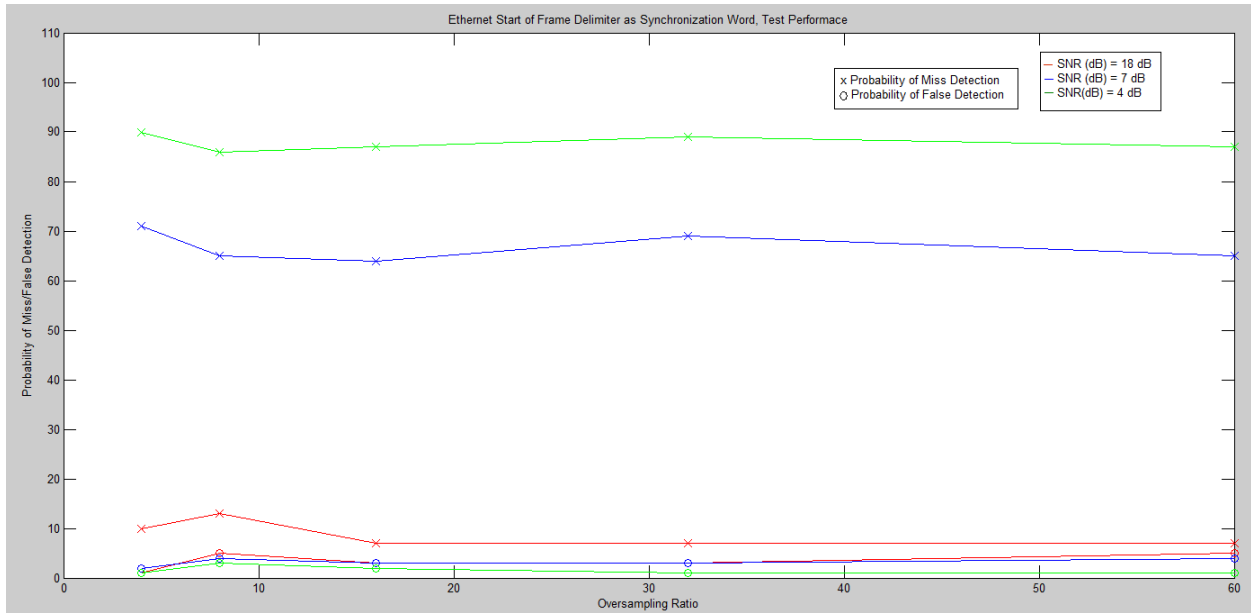
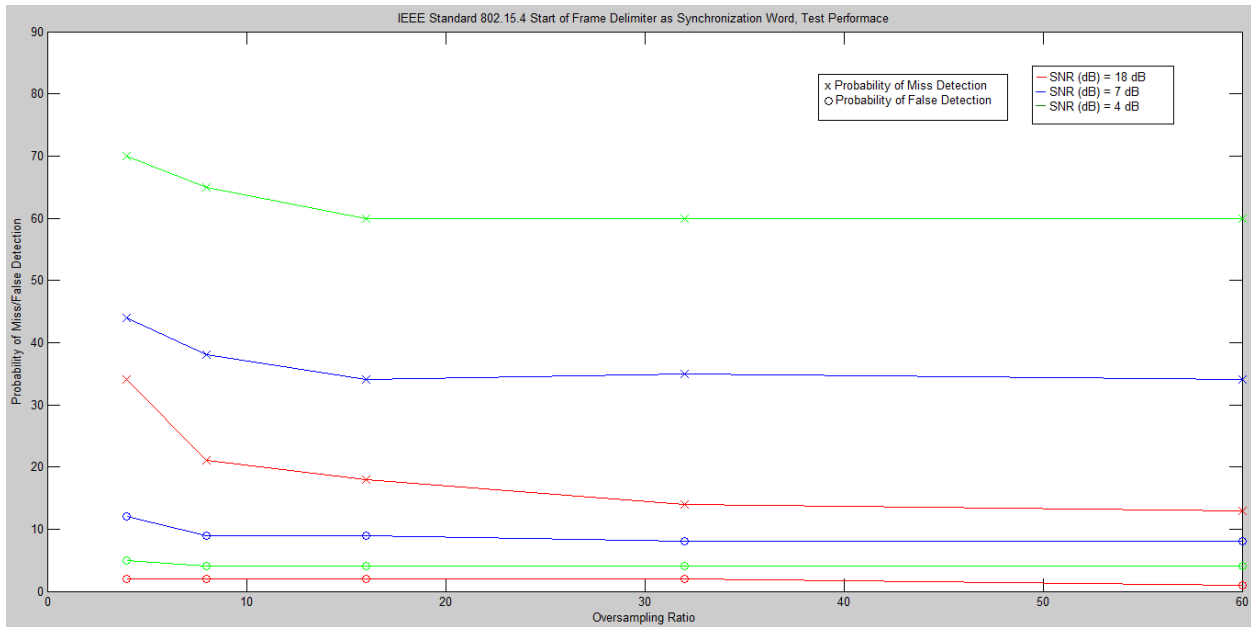


Figure 33: IEEE 802.15.4 Start of Frame Delimiter as Synchronization Word, Test Performance



Another aspect to consider is the space required on the FPGA for the different oversampling ratios. Considering the 7-bit Barker sequence, Table 5 below shows the drop in probability of

miss detection compared to the increase in logic gates for the various oversampling ratios with SNR (dB) seen by the receiver being 7 dB.

Table 5: Drop in Miss Detection vs Increase in Logic Gates, 7-bit Barker Sequence

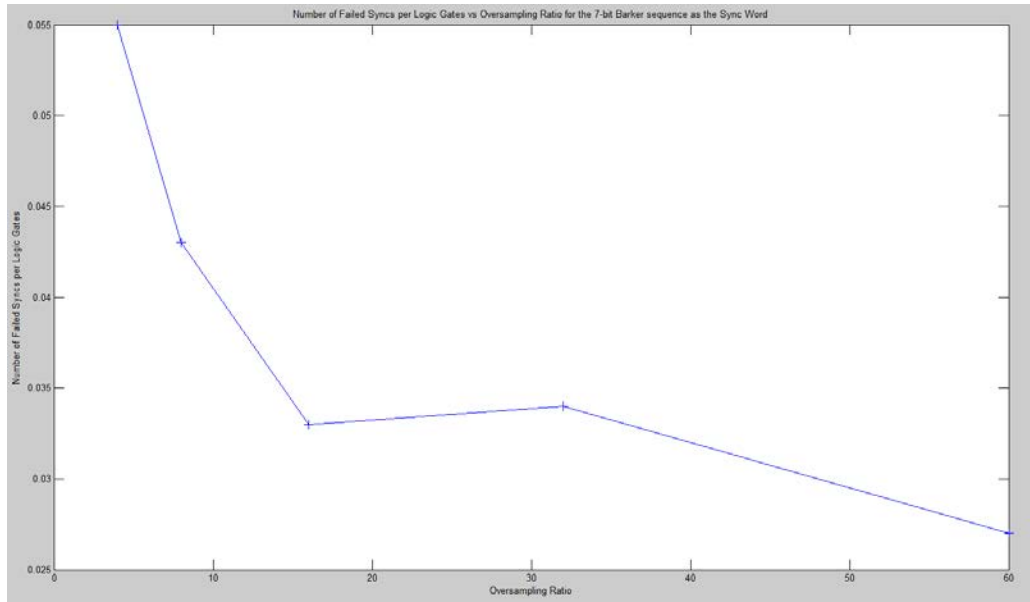
Oversampling Ratio	Improved Failed Sync Performance (4x as reference)	Increased Number of Logic Gates (4x as reference)	Number of Failed Syncs per Logic Gates
60	-3%	192	0.027
32	5%	109	0.034
16	7%	54	0.033
8	9%	25	0.043
4	x	x	0.055

4.3 Test Results Conclusion

In terms of probability of miss detection, the 7-bit Barker sequence and the IEEE 802.15.4 start of frame delimiter outperformed the sequence 01010111 and the start of frame delimiter for Ethernet. However, in terms of false detection, all synchronization words had their probability of false detection drop with increasing oversampling ratio except for the IEEE 802.15.4 start of frame delimiter. The 01010111 sequence along with the Ethernet start of frame delimiter and the IEEE 802.15.4 start of frame delimiter had similar probabilities for false detection and outperformed the 7-bit Barker sequence. The false detection results could have been influenced by the choice of sequence that followed the synchronization word, and contrary to the simulation results, the number of false detections did not increase as the SNR (dB) was lowered. This was especially true for the sequence 01010111.

Shown on the next page in Figure 34 is a plot of oversampling ratio versus number of failed syncs per logic gates.

Figure 34: Oversampling ratio versus number of failed syncs per logic gates



As can be seen in Table 5, doubling the oversampling ratio also doubles the number of logic gates for the system. As was seen in the simulation results, increasing the oversampling ratio above 16 does not improve the results significantly to justify the increase in logic elements.

Chapter 5 - Summary and Conclusions

Both the simulation and test results showed that the optimal oversampling ratio is 16. Having a higher oversampling ratio does not improve system performance enough to justify the extra logic elements required as was shown in figure 34. For the synchronization words compared, the simulation results showed that all synchronization words have comparable results for both probability of miss detection as well as probability of false detection. The probability of false detection increased with the oversampling ratio for all synchronization words for simulation. This seems reasonable since a higher oversampling ratio would result in more estimations of synchronization for a given time window. However, this was not the case for all synchronization words for the test setup. This was most likely due to the simulation considering impulse noise as the only noise source while in the system testing; there are multiple sources of noise. For instance, as the SNR was lowered, the probability of bit error increased for a channel operating in AWGN (Additive White Gaussian Noise) along with the increases in impulse noise due to the simple demodulator being sensitive to low SNRs. When the synchronization words were compared in the test setup, the 7-bit Barker sequence along with the start of frame delimiter for IEEE 802.15.4 showed improvements in the probability of miss detection compared to the sequence 01010111 and the start of frame delimiter for Ethernet. This was expected since Barker sequences are known to have autocorrelation functions with small sidelobes.

Chapter 6 - Future Work

For simulation and testing, only one data rate was used. The PFD demodulator has the ability to work at several different data rates, so for the system to operate at more data rates, the only module that would need adjusting is the maximum-likelihood estimation. A physical layer protocol would need to be first determined as is done in IEEE standard 802.15.4. For example, the communication system could operate at three different data rates, 10 kbps, 20kbps, or 30 kbps. However, since the data rate is only determined at the start of the packet, the packet would need to be uniform in terms of bit times. For instance, it wouldn't be possible to transmit the first half of the packet at one data rate and the other half at another data rate. It would only be possible to accomplish this if another frame synchronization word were placed in the middle of the packet in order to reestablish synchronization and determine the new data rate using the method described in this thesis. The number of possible data rates would determine the number of maximum likelihood estimators that would need to be created as was pointed out in section 3.2.

Another channel parameter that was fixed for simulation and testing was the carrier frequency. The RFIC has an RSSI (Received Strength Indicator). Therefore, with a physical layer protocol specifying the different possible carrier frequencies, the FPGA could program the RFIC to scan through the possible carrier frequencies while measuring the RSSI to determine which carrier frequency was used. However, this would make the design more complicated which would make the design consume more power.

References

- [1] IEEE Standard 802.15.4, pp. 43, 2003.
- [2] U. Mengali and Aldo D'Andrea, "Synchronization Techniques for Digital Receivers". *Plenum Pub Corp*, 1997.
- [3] W. Kuhn, "Mars Proximity Microtransceiver RFIC Interface Design Document," rev. 09, pp. 1-21, Jan. 2008.
- [4] Microsemi, "IGLOO Low Power Flash FPGAs with Flash*Freeze Technology," FPGA datasheet, rev. 24, 2014.
- [5] Ma Chixiang and Hao Cao and Ping Lin, "A Low-Power Low-Cost Design of Primary Synchronization Signal Detection," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no.7, pp. 1161-1166, July 2012.
- [6] MASSEY JL., "Optimum Frame Synchronization," *IEEE Transactions on Communications*, vol. 20 (2), 1972, pp. 115-119.
- [7] R. Mehlman and H. Meyr, "Optimum frame synchronization for asynchronous packet transmission," in *Proc. ICC*, pp. 826-830, IEEE, 1993.
- [8] R. Talluri, "Error-resilient video coding in the ISO MPEG-4 standard," *IEEE Commun. Mag.*, vol. 36, no. 6, pp. 112-119, Jun. 1998.
- [9] L. E. Franks, "Carrier and bit synchronization in data communication – a tutorial review," *IEEE Trans. On Comm.*, Vol. com-28, No. 8, August 1980.
- [10] T. Schaub and A. Hansson, "Frame synchronization for spontaneous transmissions," in *Proc. IEEE Global Telecomm. Conf. (Globecom 1990)*, vol. 1, Dec. 1990, pp. 617-622.
- [11] E. Bastaki, H. Tan, Y. Shi, and K. B. Letaief, "Frame synchronization based on multiple frame observations," *IEEE Trans. Wireless Commun.*, vol. 9, no. 3, pp.1097-1107, Mar. 2010.
- [12] Linder, J., "Binary sequences up to length 40 with best possible autocorrelation," *Electron. Lett.*, 1975, vol. 11(21), p.507.