Ultra-Wideband Research and Implementation

Senior Capstone Project 2007

Final Report

by:
Jarrod Cook
Nathan Gove

Project Advisors:

Dr. Brian Huggins
Dr. In Soo Ahn
Dr. Prasad Shastry

Abstract

Ultra-wideband (UWB) is a wireless transmission standard that will soon revolutionize consumer electronics. UWB is interesting because of its inherent low power consumption, high data rates of up to 480 Mbps, and large spatial capacity[1]. Furthermore, the power spectral density is low enough to prevent interference with other wireless services. This new standard has recently been approved by the FCC for unlicensed use and has been gaining interest throughout the consumer electronics industry.

The main goal of the project is to increase interest in UWB technology, and to explore its usefulness. The history and benefits of UWB will be discussed, followed by the project goals, implementation, and results. Texas Instruments DSP platforms were used in conjunction with Simulink and Code Composter Studio to implement the scaled-down baseband modulation scheme. Also, RF microwave components were donated from Hittite Corporation to perform quadrature modulation and up conversion to a frequency of 3.4 GHz.

[1] Spatial capacity is defined as bits/sec/square-meter. UWB is projected at having 6 devices working simultaneously within a 10 m range [16].

Acknowledgements

There are some people that need to be acknowledged for their work with this project. They include, but are not limited to:

- Luke Vercimak and Karl Weyeneth for work on their 2006 Senior Capstone Project, Software Defined Radio.
- Dr. Brian Huggins, Dr. In Soo Ahn and Dr. Prasad Shastry for advising, guidance and general encouragement throughout.
- Bader Al-Kandari for obtaining DSP boards for use in this project.
- Texas Instruments for donating the DSP boards for use in this project.
- Hittite Corp. for donating radio frequency hardware for use in this project.

Table of Contents

1. Introduction

Ultra wideband (UWB) is a recently approved wireless communication standard that has been growing industry wide interest. The main reasons why this standard has been creating so much hype are its proposed data transfer speeds of 480 Mb/s, its low power consumption while transmitting at these high speeds, and its spatial capacity. The spatial capacity (bits/sec/square-meter) is really where UWB outperforms the competition. UWB is projected at having upwards of 6 devices working simultaneously at 480 Mb/s within a range of 10 m, which is unheard of in today's wireless communications [16]. UWB has the potential to revolutionize the consumer electronics industry.

2. Ultra-wideband Research

2.1 UWB Definition

On February 14, 2002 the FCC released a report that officially allocated spectral space for UWB technology. This allocation was strictly to define and restrict the radio frequency (RF) emissions of this technology and bandwidth to allow coexistence. The minimum bandwidth of UWB as defined by the FCC must follow one of the two constraints listed below [1].

- The minimum bandwidth must occupy more than 20% of the center frequency.
- The minimum bandwidth must exceed 500 MHz.

The total bandwidth that could be occupied as defined by the FCC is from 3.1 GHz to 10.6 GHz. This covers a total span of 7.5 GHz. The power regulations for this technology were also strictly defined to allow coexistence. The max power emitted must be under -41.3 dBm/MHz. This is equivalent to 0.5 mW of average continuous power transmission across the full 7.5 GHz bandwidth (3.1-10.6 GHz) [2].

2.2 UWB History

UWB is not a new technology. There are recorded experiments performed by Heinrich Hertz in 1865 that involved transmitting electromagnetic (EM) waves. The transmission of those EM waves can be considered UWB. These experiments led Guglielmo Marconi to invent the Morse-code telegraphy in 1901 [3]. Then between the 1900s and 1950, wireless technology went mainly narrowband or tuned wireless transmission. Such inventions included telephones, amplitude modulation (AM), frequency modulation (FM), and various other narrowband technologies that are taken for granted today [4].

Around the 1950s is when UWB technology started to reappear. During this time period, the main research done on UWB was for communications, radar, and various other applications [4]. Around this same time period, the first patents for UWB type technologies start appearing. In 1952, Louis de Rosa was granted a patent (U.S. Patent No. 2671896) for his random impulse system [1]. Another UWB type

patent was granted to Conrad H. Hoeppner in 1961 for his work with a pulse communication system that reduced jamming capabilities [1].

Next, the UWB impulse radio was invented in the 1970s using digital techniques. This led to the first commercial UWB systems to be invented and sold around the 1980s and 90s [4]. One such company that was selling these products was PulsON or Time Domain Corporation. Subsequent to these important advancements, the FCC released its Report & Order on February 14, 2002 that allowed UWB to be used under Part 15, or commercial unlicensed use [5]. In 2003 there was a push by the Institute of Electrical and Electronics Engineers (IEEE) 802 groups to create a standard for UWB technology.

In January 2003, the IEEE 802.15.3a task group was created to standardize UWB. The first step taken was to call for all the proposals for the different technologies that could be classified as UWB. This led to 21 proposals that were submitted to the task group. Since this number was very large, a few companies tried to merge or consolidate their proposals into one as many of the proposals had only slight differences. Then in May 2003, the task group had a meeting where the proposals were narrowed down from 21 to 2. The two proposals were Multiband Orthogonal Frequency Division Multiplexing (MB-OFDM) and Direct Sequence technique (DS). Through politics and voting loop-holes the IEEE 802.15.3a task group was held up indefinitely. In January 2006, the supporters of both proposals shut down the IEEE 802.15.3a task group, as it had been stalled for 3 years. This task group was ended without conclusion [6].

While the task group was in constant deliberation, outside supporters of the MB-OFDM proposal formed a group of their own to continue pursuing their idea. The Multiband OFDM Alliance (MBOA), now known as the WiMedia Alliance was created. This group continued to work while the IEEE task group was stalled. The WiMedia Alliance decided to release the WiMedia specifications and to get their proposal standardized [6].

WiMedia then searched for an alternative to IEEE to handle the standardization. ECMA International was used because of its strict rules about removing politics and business issues from deliberations. ECMA created its own technical committee TC20 [6]. In December 2005, ECMA International released the ECMA 368 [11] and 369 UWB technical standards using MB-OFDM.

WiMedia is currently working on getting the ECMA standards approved around the world. They believe that a new technology will grow to a successful state only if the following conditions are met [6]:

- The technology is not fragmented among multiple incompatible standards.
- The technology is standardized openly with no individual intellectual property holders.
- The standards are global to assure global trade of products with the new technology.

2.3 Main UWB Techniques

The only positive conclusion that came from the IEEE 802.15.3a task group was the elimination of all proposals for UWB techniques except for two. Those two include direct sequence and MB-OFDM. These two techniques use very different methods to fill up the same allocated bandwidth and power spectral densities laid out by the FCC in 2002.

2.3.1 Direct Sequence Technique (DS)

This technique was the first to fall within the scope of UWB. It was first implemented in the late 1960s after key advancements in measurements technology. Murray Nicolson was the inventor of the tunnel diode constant false alarm rate receiver (CFAR), still in use today [1].

DS uses short impulses to send information. The shorter the pulses are in the time domain, the wider the bandwidth that these signals occupy in the frequency domain. This is due to the relationship defined by the Fourier Transform. In general, the greatest performance is obtained when these pulses occupy the most bandwidth.

One particular DS UWB system was proposed by Fujjita. This system used π/2 BPSK modulation [2]. The system is shown in Figure 1 below.

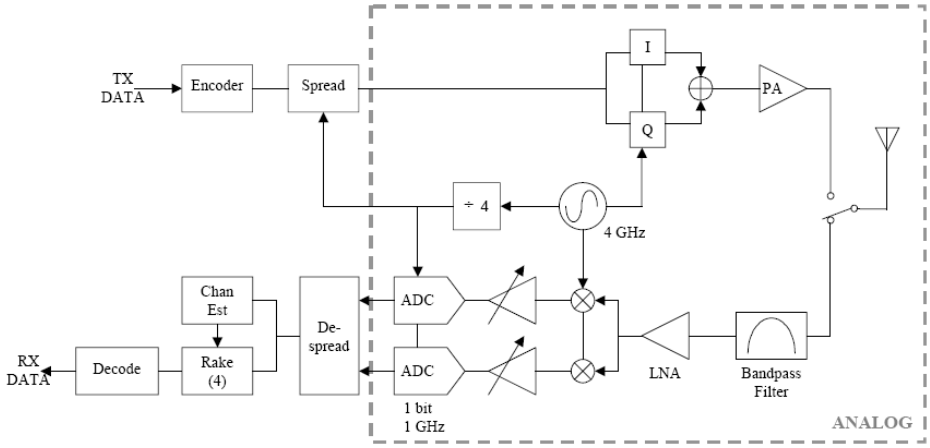


Figure 1 - DS UWB System Proposed by Fujita

This particular model is projected at providing data at the rate of 115 Mb/s at a range of 10 meters while consuming 150 mW of power [2]. There are many other techniques that can be used that still fall within the realm of DS UWB. These techniques include but are not limited to, On-Off Keying (OOK), Bi-Phase Modulation (BPM), and Binary Pulse Position Modulation (BPPM). Each different technique has their advantages and disadvantages, which could be discussed in great detail. However, the goal of this paper is to discuss the later of the two techniques MB-OFDM as the UWB Research and Implementation project focuses on this technique.

2.3.2 Multiband Orthogonal Frequency Division Multiplexing (MB-OFDM)

This technique differs from DS UWB in that instead of using short pulses to fill the bandwidth, a series of subcarriers are used to create a frame of data that is transmitted. Before OFDM, a majority of communications, such as AM or FM, involved taking a stream of data (or a serial data stream) and doing some baseband modulations in the time domain. Then taking that time domain signal and up converting it to higher frequencies for transmission. Instead, OFDM treats incoming streaming data (inherently in the time domain) as if it is already in the frequency domain by grouping the data in a parallel stream. Then this parallel stream of data is orthogonally placed on a frequency spectrum to create an OFDM frame, hence the name orthogonal frequency division. Multiband-OFDM occurs when OFDM frames are interleaved in time to different frequencies.

2.3.2.1 OFDM Transmission

The first step to creating OFDM is to take a serial data stream and convert it to a parallel stream. A serial bit stream is what is commonly thought of as streaming data. Parallel data is different from serial in that it groups data into frames. This is done so that a frame or batch of data can be processed all at one time which is necessary for taking the inverse fast Fourier transform (IFFT), the heart of OFDM modulation [7].

The next step involved in creating an UWB OFDM frame is to quadrature amplitude modulate (QAM) the incoming data. QAM is the process of taking an incoming stream of data and representing that data as symbols on a constellation with in-phase and quadrature phase (I&Q) components. One such constellation for 4 QAM (also known as π/4 quadrature phase shift keying or QPSK) is shown in Figure 2 below. This constellation is used in UWB for data rates between 80 to 200 Mb/s [11].

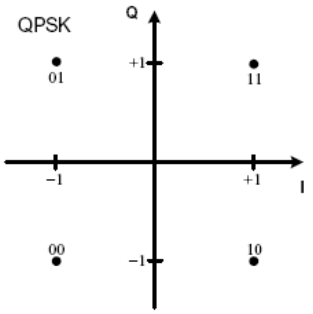


Figure 2 – 4 QAM Constellation (or π/4 QPSK Gray Coding)

As the data rates are increased, a larger constellation can be used that will modulate more information into each signal without increasing the bandwidth. In the case of 4 QAM there are four constellation points; two bits coming in become one symbol on the constellation. That one symbol can be located in one of the four positions on the constellation (Figure 2). As more data is required to be sent, 16 QAM (also known as dual carrier modulation DCM) is used as each symbol can hold four bits. This constellation is shown in Figure 3 below. This constellation is used in UWB to support data rates from 320 to 480 Mb/s as defined in the UWB specification [11].

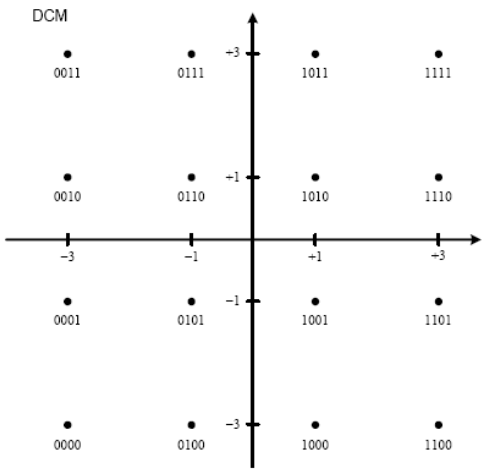


Figure 3 - 16 QAM (or DCM) Constellation

Each symbol then becomes one data subcarrier in an OFDM frame. This frame is constructed by rearranging the incoming batch of data based on a defined OFDM frame. The full UWB OFDM frame is shown in Figure 4 [11]. This spectrum includes pilot subcarriers that are added in between the data subcarriers as well as some guard subcarriers. The pilot subcarriers are used by the receiver to synchronize the symbols after transmission. The guard subcarriers are used to shape the frequency spectrum and to make sure that adjacent frequency spectrums do not overlap. Also note that there is no subcarrier at DC. This is to avoid issues with implementation in hardware (specifically DAC & ADC offsets) [11].

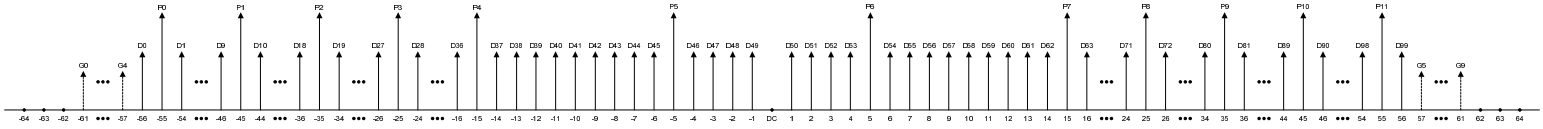


Figure 4 – Full UWB OFDM Frame

Once the OFDM frame is constructed, it is sent through an IFFT. This effectively takes the OFDM frame with each subcarrier representing a different frequency and transforms it into the sum of all of these frequencies in the time domain. The IFFT itself assures that all of the frequencies are orthogonally placed. This process is explained in Figure 5. This figure shows how one frequency, which is a representation of one π/4 QPSK symbol, is transformed into the time domain (middle) with an IFFT. This is the signal that would be transmitted if only one subcarrier were in each frame, but this is not the case. The lower portion of Figure 5 shows how multiple subcarriers represent different symbols with different frequencies. The sum of all of these time domain signals (middle bottom) is what is actually transmitted as OFDM. The mathematics underlying this process is given in Appendix A (also see [7]).

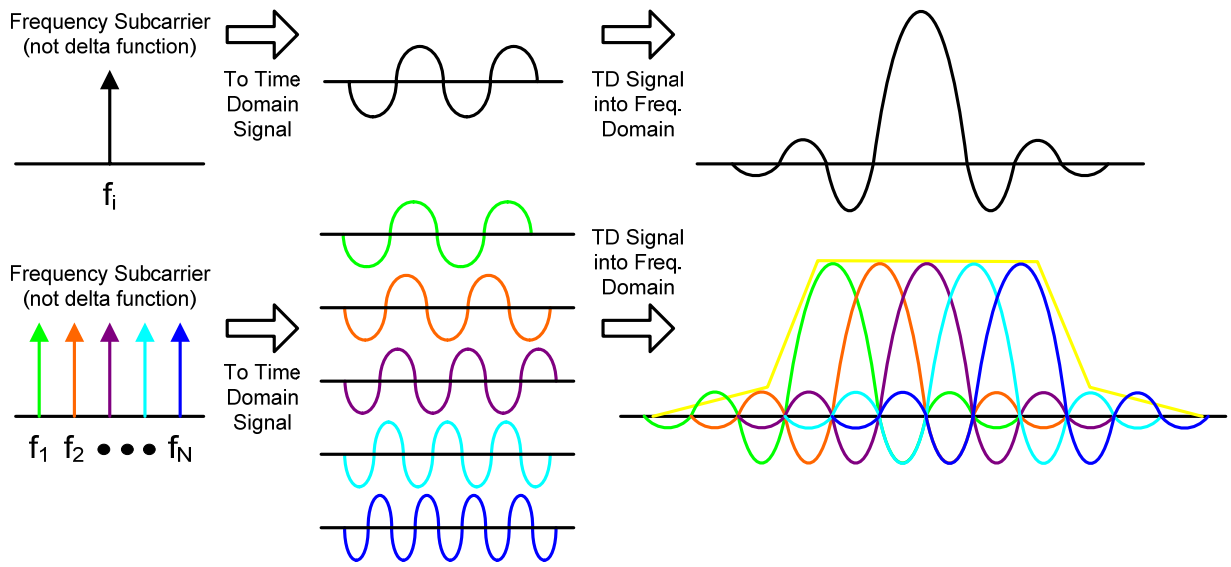


Figure 5 – OFDM Explanation

The signals on the far right (in Figure 5) are the time domain signals then converted back to the frequency domain using a Fourier transform (FFT). Ideally if all of the frequencies were exact sinusoidal waves then signals on the right would look like those on the left, but this is not the case. The signals in the middle are not perfect sine waves but complex sine waves with I&Q components, thus the sinc function (or sin{x}/x) on the right is what results. This second transformation is done to take various measurements such as bandwidth and center frequency.

It is important to remember that one frame of data is processed simultaneously. Thus, the full transmitted spectrum will fill up 528 MHz as defined in the UWB specification [11], which meets the FCC minimum requirements for UWB transmission.

2.3.2.2 Multiband OFDM Transmitter

MB-OFDM is actually done after the OFDM process above is complete. MB-OFDM is the process of time interleaving an OFDM frame across multiple frequencies. Figure 6 shows how this process works in reference to time (x-axis) and frequencies (y-axis) [12]. As time progresses the OFDM frame is transmitted on different center frequencies.

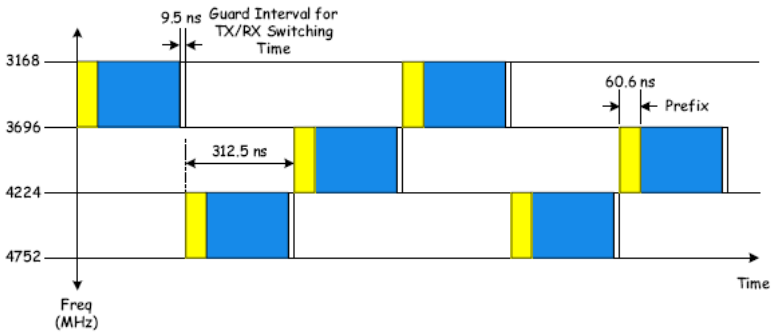


Figure 6 – Multiband OFDM or Time-Frequency Interleaving

The frequencies and the different patterns of time-frequency interleaving are controlled with the time-frequency kernel, defined in the UWB specification [11]. The advantages gained from using MB-OFDM in a UWB system are [1]:

- Reduces the internal precision of the digital logic (IFFT & FFT).
- Reduces the precision required by the analog to digital (ADC) and digital to analog (DAC) converters.
- Relaxes the phase-noise requirements on the carrier synthesis circuitry and improves system robustness.

Figure 7 below is an example of a MB-OFDM transmitter [12]. The blocks that cover techniques for reducing or correcting errors (boxed in red, Figure 7) are not discussed because they are common to many communication systems. However, the constellation mapping, pilots, and IFFT are discussed above.

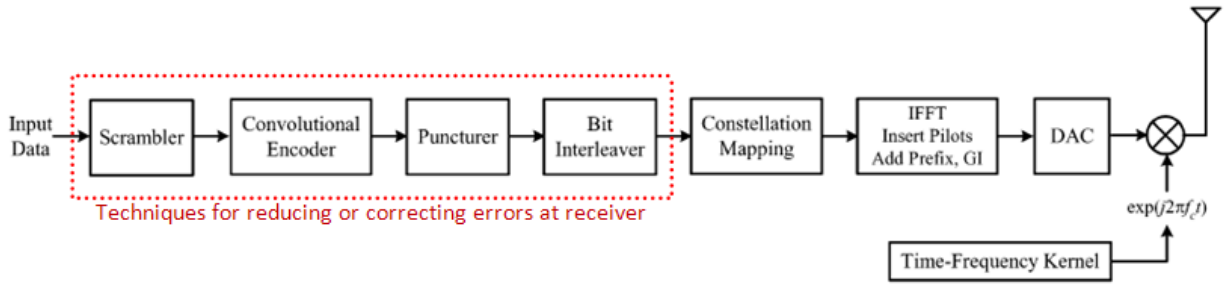


Figure 7 – MB-OFDM Example Transmitter

The implementation of MB-OFDM is done with the Time Frequency Kernel (TFK), shown in Figure 7. This TFK determine the center frequency of the up conversion and changes with time as shown in Figure 6.

The output of the IFFT contains the complex I&Q components from the π/4 QPSK constellations. These I&Q signals cannot be outputted by one DAC because DACs do not accept complex numbers. A technique called quadrature modulation is used to not only add the I&Q components into one real signal, but also to up convert that signal to a higher frequency for transmission. This up converted frequency or center frequency is controlled by the TFK. Figure 8 shows a block diagram of quadrature modulation controlled by a TFK.

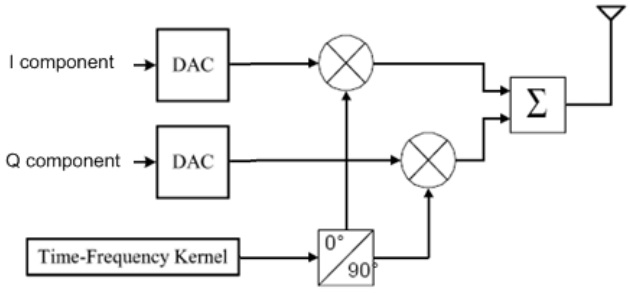


Figure 8 – Quadrature Modulation for MB-OFDM with Time-Frequency Kernel

2.3.2.3 MB-OFDM Preamble or Prefix

In MB-OFDM UWB synchronization is the key to a successful reception of the transmitted information. To accomplish this, preambles are inserted before a pre-determined amount of data. It is important that the preamble be repeated because the preamble provides two important functionalities that are essential for the receiver to work. The preamble provides packet or frame synchronization information. This information allows the receiver to lock onto the information and know when a frame begins and ends. The second is the channel estimation. This is a very important function because it allows the receiver to correct for changing environment conditions as they are changing. There are two types of preambles that are used for MB-OFDM UWB.

- Standard Physical Layer Convergence Protocol Preamble
- Burst Physical Layer Convergence Protocol Preamble

The burst preamble is used when high data rates are desired (for data rates above 200 Mb/s). Even though the burst preamble is used for high data rate transmission, the standard preamble must always be the first preamble sent to the receiver [11].

Both preambles can be divided into two parts, the packet or frame synchronization sequence and the channel estimation sequence. The standard preamble is defined in Equation 1 below [11].

$$s_{sync,n}[k] = s_{cover}[n] \times s_{ext}[k] \quad \text{where } n \in [0, N_{pf}-1],\ k \in [0, N_{SYM}-1]$$

where: N_{SYM} = # of samples per symbol = 165 (N_{FFT}+N_{ZPS})
N_{PF} = # of symbols in packet frame synchronization = 24 (standard preamble)
N_{CE} = # symbols in channel estimation = 6 (both preambles)

Equation 1 – Standard Preamble Definition

The standard preamble block diagram shown in Figure 9 visually represents Equation 1.

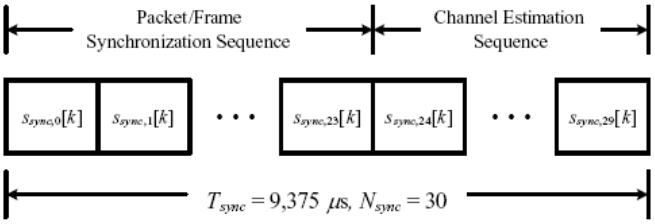


Figure 9 – Standard Preamble Block Diagram

The burst preamble is similar to the standard preamble defined by Equation 1, but has different parameters that control it shown in Equation 2 below [11].

$$s_{sync,n}[k] = s_{cover}[n] \times s_{ext}[k] \quad \text{where } n \in [0, N_{pf}-1],\ k \in [0, N_{SYM}-1]$$

where: N_{SYM} = # of samples per symbol = 165 (N_{FFT}+N_{ZPS})
N_{PF} = # of symbols in packet frame synchronization = 12 (burst preamble)
N_{CE} = # symbols in channel estimation = 6 (both preambles)

Equation 2 – Burst Preamble

Again, Figure 10 below shows a block diagram of the different portions of the preamble constructed [11].

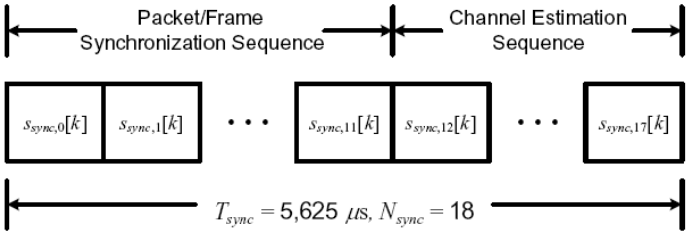


Figure 10 – Burst Preamble Block Diagram

The packet or frame synchronization sequence is chosen from a set of seven different time frequency codes. Each code is used in different spectrums and is selected based on which band group is used and channel characteristics.

With the preambles correctly transmitted, the receiver will be able to understand and decode what the transmitter is sending.

2.3.2.4 MB-OFDM Receiver

The receiver is often the most difficult portion of any communication system; MB-OFDM UWB is no different. Figure 11 is an example of a MB-OFDM receiver.

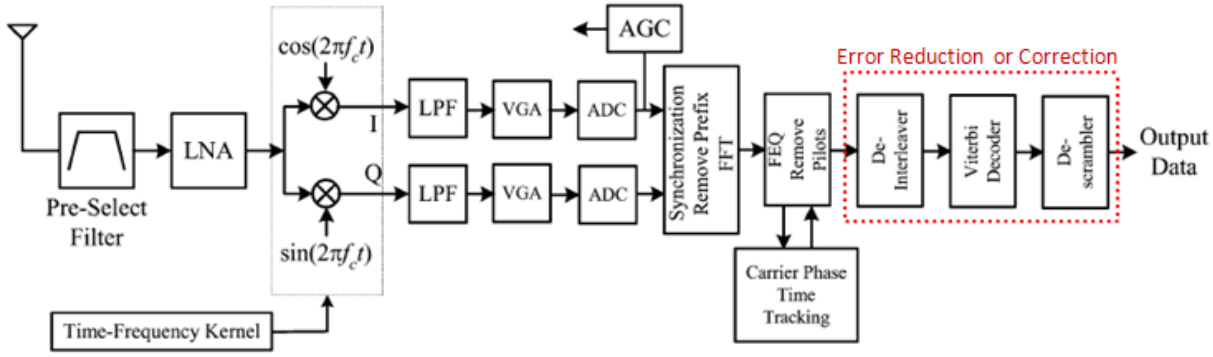


Figure 11 – MB-OFDM Example Receiver [12]

This receiver (Figure 11) is a generic MB-OFDM receiver. The specifics on exactly how to implement each block are not laid out in the MB-OFDM UWB Specification [11]. However, there are specifications that the receiver must be within to be compliant with this standard. Table 1 [11] shows the receiver sensitivity requirements for a packet error rate of less than 8% when sending 1024 octets of information. Also, in Table 1 the minimum receiver sensitivity in additive white Gaussian noise (AWGN) for different data rates are listed.

Table 1 – Min. Receiver Sensitivities for Band Group 1

Data Rate (Mb/s)	Minimum Receiver Sensitivity (dBm)
53,3	–80,8
80	–78,9
106,7	–77,8
160	–75,9
200	–74,5
320	–72,8
400	–71,5
480	–70,4

As long as the specifications for the receiver are met, the exact implementation of the various blocks shown in Figure 11 can be decided on by the design engineers.

2.3.2.5 ECMA MB-OFDM UWB General System Specifications

The ECMA International 368 High Rate Ultra Wideband PHY and MAC Standard has many different specifications for each facet of a MB-OFDM UWB system. There are a few key parameters that are describe in Table 3 that go over a majority of the general UWB system specifications. Also, Table 2 and Figure 12 show the UWB standard spectrum allocation into 14 sub bands [11]. These specifications lay out the basics of a MB-OFDM UWB system (from here forth MB-OFDM UWB will just be referred to as UWB).

Table 2 - UWB Standard Spectrum Allocation

Band Group	BAND_ID (n_b)	Lower Frequency (MHz)	Center Frequency (MHz)	Upper Frequency (MHz)
1	1	3 168	3 432	3 696
	2	3 696	3 960	4 224
	3	4 224	4 488	4 752
2	4	4 752	5 016	5 280
	5	5 280	5 544	5 808
	6	5 808	6 072	6 336
3	7	6 336	6 600	6 864
	8	6 864	7 128	7 392
	9	7 392	7 656	7 920
4	10	7 920	8 184	8 448
	11	8 448	8 712	8 976
	12	8 976	9 240	9 504
5	13	9 504	9 768	10 032
	14	10 032	10 296	10 560

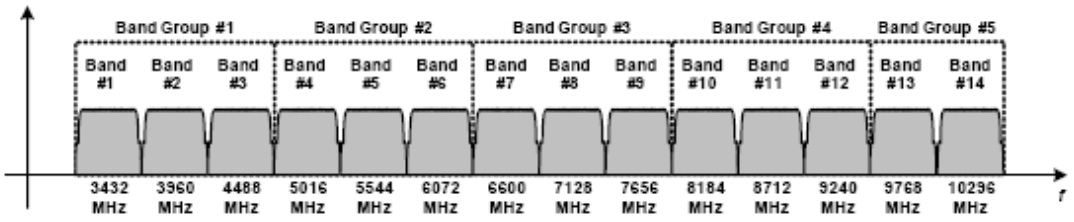


Figure 12 – UWB Standard Spectrum Allocation

Table 3 – MB-OFDM UWB General System Specifications

Parameter	Description	Value
f_s	Sampling frequency	528 MHz
N_{FFT}	Total number of subcarriers (FFT size)	128
N_D	Number of data subcarriers	100
N_P	Number of pilot subcarriers	12
N_G	Number of guard subcarriers	10
N_T	Total number of subcarriers used	122 (= $N_D + N_P + N_G$)
D_f	Subcarrier frequency spacing	4,125 MHz (= f_s / N_{FFT})
T_{FFT}	IFFT and FFT period	242,42 ns (Δ_f^{-1})
N_{ZPS}	Number of samples in zero-padded suffix	37
T_{ZPS}	Zero-padded suffix duration in time	70,08 ns (= N_{ZPS} / f_s)
T_{SYM}	Symbol interval	312,5 ns (= $T_{FFT} + T_{ZPS}$)
F_{SYM}	Symbol rate	3,2 MHz (= T_{SYM}^{-1})
N_{SYM}	Total number of samples per symbol	165 (= $N_{FFT} + N_{ZPS}$)
N_{pf}	Number of symbols in the packet/frame synchronization sequence	Standard Preamble: 24 Burst Preamble: 12
T_{pf}	Duration of the packet/frame synchronization sequence	Standard Preamble: 7,5 µs Burst Preamble: 3,75 µs
N_{ce}	Number of symbols in the channel estimation sequence	6
T_{ce}	Duration of the channel estimation sequence	1,875 µs
N_{sync}	Number of symbols in the PLCP preamble	Standard Preamble: 30 Burst Preamble: 18
T_{sync}	Duration of the PLCP preamble	Standard Preamble: 9,375 µs Burst Preamble: 5,625 µs
N_{hdr}	Number of symbols in the PLCP header	12
T_{hdr}	Duration of the PLCP header	3,75 µs
N_{frame}	Number of symbols in the PSDU	$6 \times \left\lceil \frac{8 \times \mathrm{LENGTH} + 38}{N_{IBP6S}} \right\rceil$
T_{frame}	Duration for the PSDU	$6 \times \left\lceil \frac{8 \times \mathrm{LENGTH} + 38}{N_{IBP6S}} \right\rceil \times T_{SYM}$
N_{packet}	Total number of symbols in the packet	$N_{sync}+N_{hdr}+N_{frame}$
T_{packet}	Duration of the packet	$(N_{sync}+N_{hdr}+N_{frame}) \times T_{SYM}$

2.4 UWB Compared to Current Wireless Technology Standards

Current wireless technology has not yet been able to satisfy the demand for high speed, short range, and low power. UWB has the potential to meet and surpass these demands. When UWB is compared against current wireless technology is it easier to see how beneficial it will be.

2.4.1 Wi-Fi

Wi-Fi technology is currently available on the market today and conforms to IEEE standard 802.11a/b/g. These standards were designed for infrastructure-oriented use so they are often found in an office or home environment providing internet and network access wirelessly throughout a building or office. Whereas UWB is meant for transmitting over shorter distances with higher bit rates directly from one device to another [9]. The difference in throughput plotted verse distance can be seen in Figure 13 below.

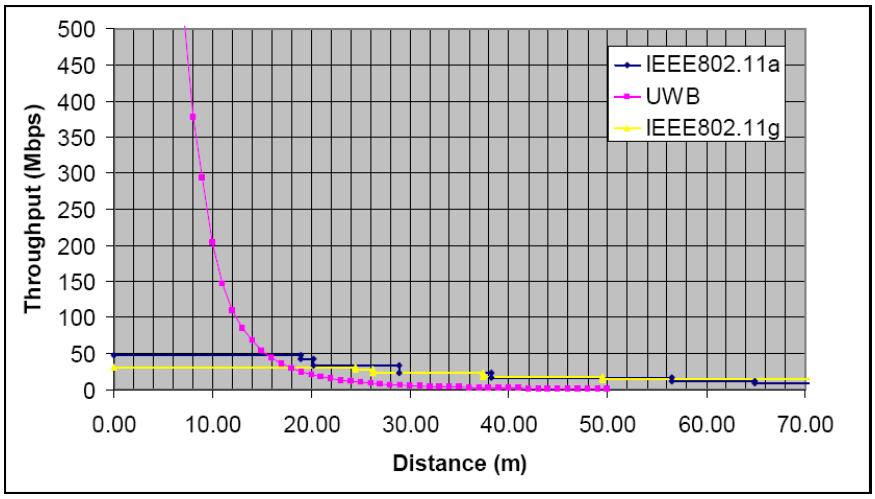


Figure 13 - Data Rate of UWB Compared to Current Wi-Fi Standards [10]

Wi-Fi works well when used for transferring data, but was not designed for high definition streaming audio and video. Wi-Fi 802.11b/g operates in the 2.4 GHz industrial, scientific and medical (ISM) band, while 802.11a operates in a 5 GHz band [9]. These standards are also known to consume power at approximately 2 orders of magnitude greater (around 100 mW) than UMB projections (around 0.5 mW) [10]. This high power consumption makes the Wi-Fi standards less idea for smaller portable devices requiring large data rates.

Wi-Fi Alliance has been working on the new IEEE 802.11n standard. This standard is projected to quadruple the throughput of current 802.11 standards. However, UWB and Wi-Fi standards are designed for different uses. Wi-Fi is designed to provide a large area with multiple user access for Ethernet based applications. Wi-Fi can be configured for point to point use, but it is not optimal. Meanwhile, UWB is ideal for point to point streaming data, such as audio and video. It is projected that IEEE 802.11a/b/g/n will support UWB devices, and they will not directly compete with each other [9].

2.4.2 Bluetooth

Bluetooth wireless technology was created as a way to replace many common cables. Some examples of applications for Bluetooth are wireless mice, keyboards, hands-free headsets, PDAs, and various others. Bluetooth uses Gaussian frequency shift keying (GFSK) with a max frequency deviation of 140-175 kHz. It also uses frequency hopping spread spectrum (FHSS) to hop at 1600 hops/sec between 79 channels (each channels separated by 1 MHz). There are three classes of Bluetooth transmission the lowest class transmits at a max power of 1mW (0 dB) with a typical range of 1m, while the typical commercial class transmits at 2.5mW (10 dB) with a typical range of 10m [14]. The maximum baseband data rate is 1 Mb/s for Bluetooth v1.2. Bluetooth v2.0 supports the same transmission classes but an increased data rate transmission of 3 Mb/s [15].

All of these specifications have been great at providing wireless transmission for a majority of portable devices that do not require high data rate transmissions. When compared to UWB, the fastest possible Bluetooth v2.0 is about 20 times slower than the slowest UWB transmission. On top of that UWB uses less energy. Bluetooth also operates in the noisy unlicensed 2.4 GHz ISM band which was chosen for world-wide capabilities, but this noisy channel often decreases the capabilities of Bluetooth transmission [9].

In the future, there could be competition between Bluetooth and UWB technology devices primarily because Bluetooth will have been around longer so the prices for those items will be significantly lower. It is also projected that most companies will implement a combination of UWB and Bluetooth devices. Either way Bluetooth is here to stay until UWB has proven itself.

2.5 Advantages of UWB

The main advantage that UWB has over other current wireless technology is bandwidth. Shannon's channel capacity equation shown in Equation 3 below describes the absolute maximum data rate a channel can transmit [7]. The equation shows that capacity "C" increases linearly with bandwidth, "B", while it increases only logarithmically with the signal to noise ratio "S/N". Thus, it is easier to send data faster in a communication system (increase its capacity) by increasing the bandwidth rather than the transmitted signal power.

$$C = B \log_2 \left(1 + \frac{S}{N} \right)$$

Where:
C = Maximum Channel Capacity (bits/sec)
B = Channel Bandwidth (Hz)
S = Signal Power (watts)
N = Noise Power (watts)

Equation 3 – Channel Capacity

UWB has a total bandwidth of 7.5 GHz which is unique. This provides the capability for expansion in the number of devices used simultaneously, that other wireless communication standards do not have. UWB is projected at having the capabilities to support approximately 6 (maybe more) devices in the

same 10m radius [16]. When looking at the spatial capacity measurements of different wireless standards, this puts UWB way out front. Spatial capacity in this case is measured in terms of bits/sec/square-meter. Figure 14 below shows the special capacity comparison between Bluetooth v1.2 and 802.11 wireless standards [16].

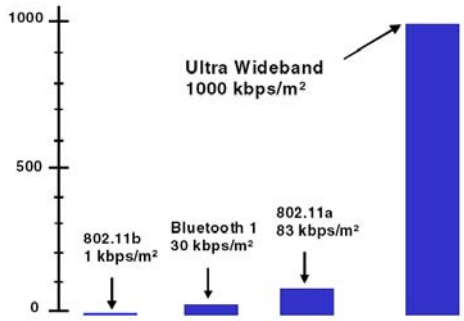


Figure 14 – Spatial Capacity Comparison

However, with every advantage there are disadvantages and challenges. UWB as a technology still has some hurdles to overcome before it is fully realized for production.

2.6 Disadvantages or Challenges for UWB

With any new technology there are bound to be new challenges just around every corner. Because of the high levels of excitement in the industry about UWB, it is difficult to gauge what issues have been solved and which issues still remain. According to Linqing Yang and Georgios Giannakis the following problems exist for UWB systems as of 2004 [17]:

- High-efficiency UWB antennas are required with tight jitter requirements.
- Improved ADC speeds as well as improved high speed automatic gain control (AGC).
- High level of complex integration required for CMOS realization.
- Need to be use high tech components but keep system at low cost.
- Staying within the FCC limits.

Contrary to this list above, various companies have already come out with UWB chipsets boasting full CMOS integration of MB-OFDM specifications. Intel has its Wireless UWB Link 1480 Chipset [18], and Staccato Communications has its SC3502P SiP package [19].

As for the rest of the issues, it is difficult to gauge whether they have been solved or not as there is still an abundance of research and development currently being done in all aspects of UWB.

3. Purpose of UWB Research Report

UWB is currently under development throughout the industry. The benefits of UWB are described in the prior section. These include large spatial capacity, high data throughput, and low power consumption. With all this exciting potential, the overall goal of this project is to inform more people about UWB and its capabilities.

Before the extensive research of UWB technology was completed, the original functional goal of this project was to create a high level system with a UWB development kit. The initial functional goals and research are shown in Appendix B. This project then evolved into creating a scaled down version of an UWB communication system.

Since the initial goals of purchasing a full-scale development kit were not realizable, focus of the project then shifted to developing a scaled-down OFDM based UWB transceiver pair. Even though the system that was developed was scaled-down, this undertaking allowed the project team to gain an understanding of how OFDM theory, UWB, and other aspects of a communications system worked. Moreover, it would have been the first senior project to physically transmit and receive information in a Senior Capstone Project at Bradley University. This was another goal the team hoped to accomplish.

The actual functional goals set forth in this UWB Research and Implementation Project was to implement a scaled-down version of a UWB communications system. This would directly include the following technologies:

- Texas Instrument's TMS320C6414T DSP Starter Kits
- MathWorks Simulink
- RF microwave hardware from Hittite Microwave Corporation

To develop the baseband modulation systems for the transmitter and receiver, the Texas Instruments C6416 DSP starter kits were used. Some of the benefits of these DSP starter kits include:

- a fast DSP processor speed of 1 GHz
- onboard integrated peripherals that included DAC and ADC
- the ability to be programmed from MathWorks Simulink models through Code Composer Studio software

Since the TI DSP boards could be programmed with Code Composer Studio, a majority of the project focused on modeling a UWB system in Simulink. The UWB modeling portion was similar to a project done by Luke Vercimak and Karl Weyeneth in 2006 at Bradley University for their Senior Capstone Project. Their project was a Software Defined Radio using OFDM modulation defined in the IEEE 802.11a specification. Their project successfully created a working simulation of an OFDM based radio in Simulink. However, they were unable to get their simulation working on any DSP platform [20]. Their

project offered the UWB team guidance in some of the more difficult aspects of an OFDM simulation (such as frame and symbol synchronization).

The final aspect of the project was to use RF microwave components to perform the direct quadrature modulation and up conversion for transmission through a channel. Pre-fabricated components were donated from Hittite Microwave Corporation [21], which was done to save time. These pre-fabricated components also came on printed circuit boards (PCBs) for testing. The components received were a direct quadrature modulator, direct quadrature demodulator, and a voltage-controlled local oscillator.

4. Project Implementation

UWB is a complex standard that involves a physical (PHY) layer and a higher level (MAC) layer to control high level operations. The focus of this project was to implement a stripped out versions of the PHY layer and to physically transmit and receive information through a channel. The full UWB standard requires the system to use a MB-OFDM technique (described above), but this was not within the scope of this project. The UWB project incorporated the following from the UWB standard:

- Transmitter
  - π/4 QPSK or 4 QAM modulation of data
  - OFDM modulation of a scaled down OFDM frame of data
    - Full UWB defined packet synchronization (Time Frequency Code #5)
    - Inverse Fast Fourier Transform (IFFT)
    - Zero padding or guard interval to protect against multipath effect
  - Up-sampling data
  - Quadrature modulation to sub band id #1 (within 1st Band Group see Figure 12)

- Receiver
  - Quadrature demodulation from sub band id #1
  - Frame synchronization
  - OFDM demodulation
    - Fast Fourier Transform (FFT)
    - Reorganization of the OFDM frame of data
  - Carrier Synchronization and Channel Phase Compensation
  - 4 QAM or π/4 QPSK demodulation of data

4.1 Overall Functional Description

To implement the UWB system, models were created with MathWorks Simulink. Then using Code-Composer Studio (CCS), the models were automatically converted into C-code. This C-code could then be loaded onto Texas Instruments (TI) TMS320C6414T DSP Starter Kit Boards. The models within the boards would facilitate all of the baseband processing required for UWB transmission and reception. RF hardware would then be used to up-convert or quadrature modulate the UWB signals to their designated frequency as defined by the UWB standard [11].

The project included two TI TMS320C6414T DSP Platforms, one for the transmitter, and one for the receiver. The block diagram for the overall system is shown below in Figure 15. Also, a more detailed block diagram is shown in Figure 16.

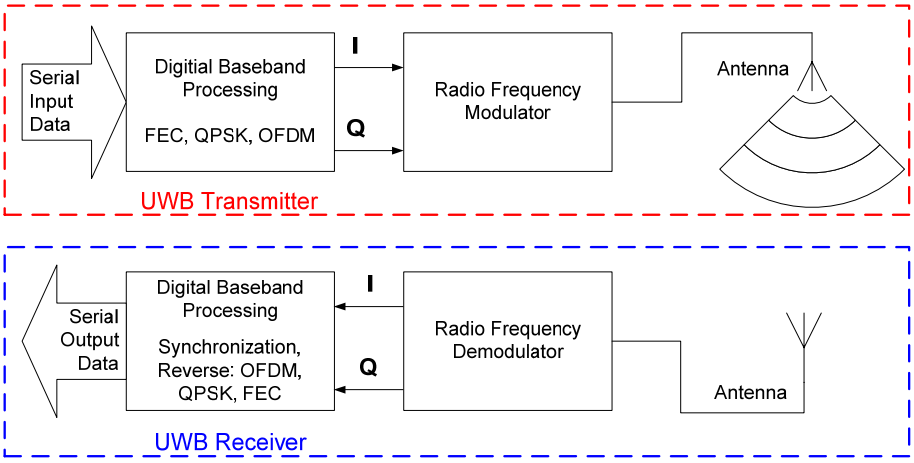


Figure 15 – High-level System Block Diagram

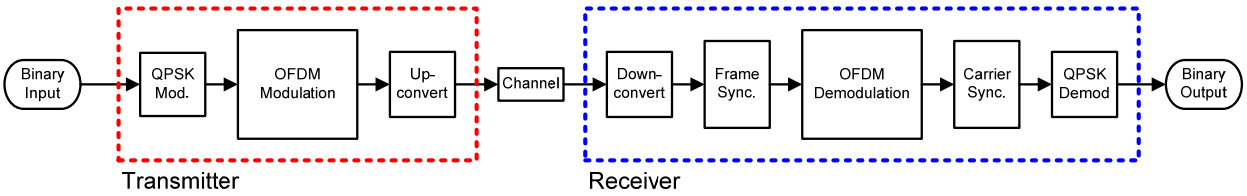


Figure 16 – Detailed System Block Diagram

The TI TMS320C6414T DSP Platforms only have one DAC per board. This meant that the I&Q signals could not both be output from the boards simultaneously. A solution to this problem is to implement the quadrature modulation within the models. The output of quadrature modulation is a real value which can be handled by one DAC. In this way the RF quadrature modulator and demodulator could not be used to their full potential. This issue will be discussed in a later section detailing the project issues and challenges.

4.2 UWB Baseband Transmitter Model Implementation

The baseband transmitter model would was designed to take an information signal sampled from the ADC and modulate that signal into a UWB type format then send it on off the board to be up-converted further for transmission. The overall Simulink model for the Transmitter is shown below in Figure 17.

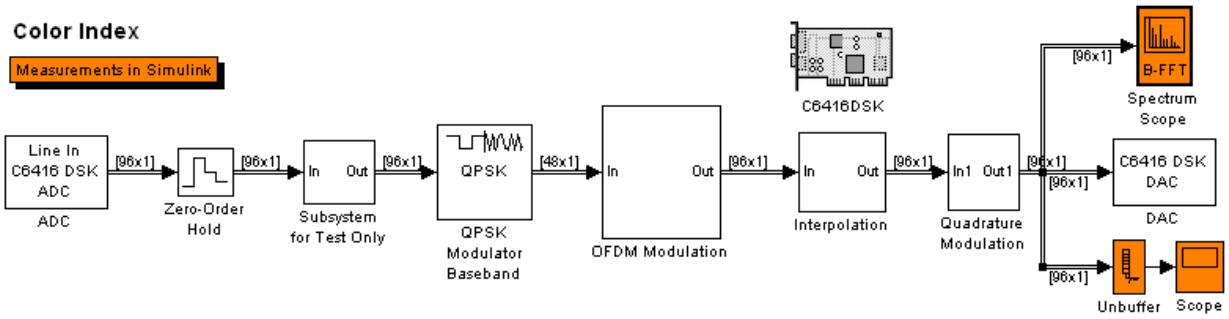


Figure 17 – Transmitter Model using 64 sized FFT

4.2.1 Input to the Transmitter Model

For the model shown if Figure 17 above, the input is an ADC. This ADC samples at a rate of 96 kHz. Since modulating data into an OFDM format requires the information be processed in groups or frames, the rate of the input data must be less than the rate of the output data. However, for both ADC and DAC to operate on the same model they need to run at the same speed. To obtain the largest bandwidth, the DAC must be running as fast as possible, in this case 96 kHz. In turn that means that the ADC must also run at 96 kHz. The zero-order hold or sample-and-hold effectively reduces the incoming data rate from 96 kHz to 125 Hz. This allows both of the converters to run simultaneously while still allowing data to be input and output on the same board.

4.2.1.1 Sample Rate Calculations

To determine how fast the data can be sent into the model without exceeding the limits of the DAC speed and without violating Nyquist's sampling theory, the following calculations were done.

$$\frac{\text{DAC max speed} \quad 96\,\text{kHz}}{2\ \text{Nyquist}\ \ \text{Sampling}\ \ \text{Theory}} = 48\,\text{kHz}$$

Equation 4 – Nyquist Sampling Theory

Thus 48 kHz is the absolute maximum sampling rate that the data can be processed at. Within the Simulink model, this would be the rate that data was coming out of the OFDM modulation block (Figure 17). To allow the output more resolution, 12 kHz was chosen as the rate at which the data would be processed.

In Simulink when the parallel data sizes change, the time taken for the frame to be processed, or T_{FRAME}, remains constant. Therefore the T_{FRAME} going into the model should be the same as the T_{FRAME} coming out of the OFDM modulation block, just the number of information symbols within that frame might change. This yields Equation 5.

$$(\text{input data rate}) = \frac{12\,\text{kHz}\,(\text{output data rate})}{96\,(\#\,\text{symbols per frame})} = 125\,\text{Hz}$$

Equation 5 – Data Rate Calculation

For example, if the number of symbols in a frame were 48, which might happen in a 32 symbol FFT simulation, then the input data rate would be 12000/48 = 250 Hz.

4.2.2 Testing Subsystem

This next portion of the transmitter model was used strictly for testing to verify the system was working. One way to definitively tell whether the model was working on the DSP was to send the model a repeating known sequence, such as all ones or all zeros. This repeating sequence would be modulated into a frame of data that would also be identically repeated. This frame of data could be obtained with an oscilloscope and compared against the simulation to confirm the model is working correctly on the board.

Figure 18 below shows how the DSP's onboard dual in-line (DIP) switches can be used for testing. With this setup, when the switches were changed on the board, it would switch between the different inputs in the model. This is handy because it takes about 5 minutes to compile the transmitter model to be downloaded to the board. The use of the DIP switches allows more testing to be done without recompiling. For the configuration below the first two of the four DIP switches are used giving four different setting described in Table 4.

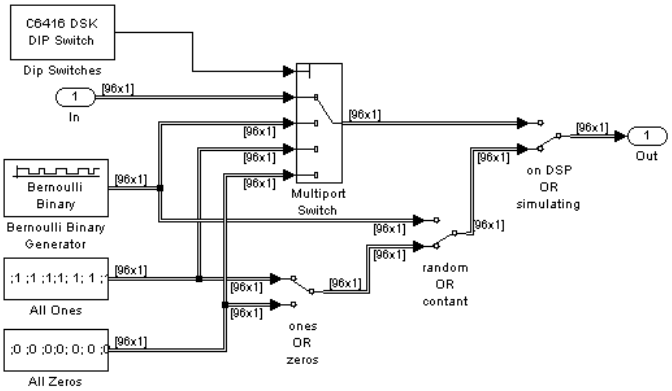


Figure 18 – Subsystem for Test Only

Table 4 – DIP Switch Settings

DIP Switch Settings (up=0 / dwn=1)	Input Selected in Model from:
00	ADC
01	Random Binary Generator
10	All Ones
11	All Zeros

Also when a simulation is being run only within Simulink and these inputs are required, the DIP switch block will output zeros as a default, which selects the ADC input. Since the model is being run within

Simulink, the ADC block will output all zeros. Manual switches are used to route around the multiport switch. To change the throw of the manual switch just double click on them. If the simulation was run with Figure 17, the input data would be all ones. Before loading the model onto the TI DSP boards, make sure last manual switch allows the multiport switching block to be used.

4.2.3 π/4 QPSK Modulation or 4 QAM

The QPSK modulation for the transmitter is done with a pre-fabricated Simulink block. This block is available within the communications block set for Simulink. The QPSK modulation block performs all of the required steps to modulate the incoming data on the correct constellation (shown in Figure 2) as described above in section 2.3.2.1 OFDM Transmission. The size of the frames going into this block are 96 bits wide. The size of the frames coming out of this block are 48 symbols wide because 2 bits are equal to 1 symbol (as described above).

4.2.4 OFDM Modulation

The OFDM modulation is the next block in the transmitter model (shown in Figure 17). This block is a subsystem that contains the model shown in Figure 19. The two main portions of this subsystem are the UWB preamble insertion and the OFDM frame construction. The model structure and subsystem structures shown in Figure 19 were adapted from Luke Vercimak's models for the transmitter [20].

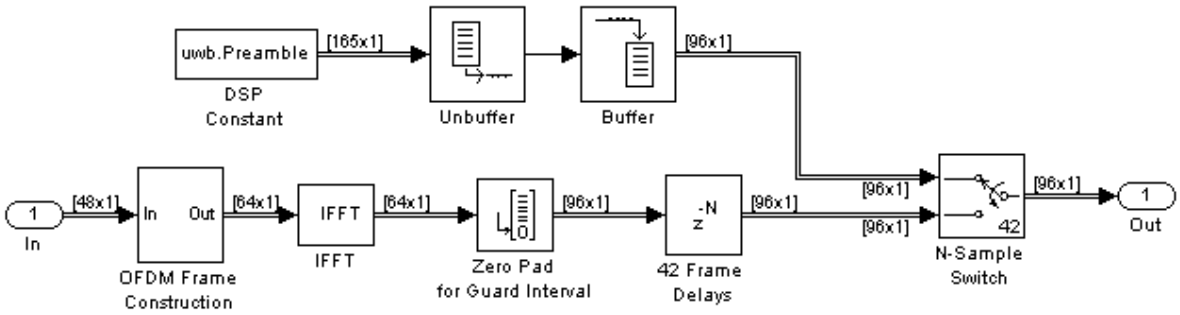


Figure 19 – OFDM Modulation Model

4.2.4.1 OFDM Frame Construction

The OFDM frame construction block handles the arranging of information and the insertion of pilots into an OFDM frame of information. The blocks that accomplish this are shown below in Figure 20.

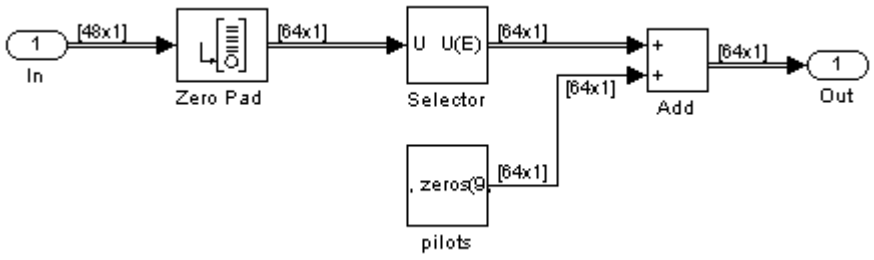


Figure 20 – OFDM Frame Construction Model

The OFDM frame for this model is shown below in Figure 21. This shows how the data is to be arranged and where the pilots are to be inserted. This frame is based off of the actual UWB frame defined in the specification [11].

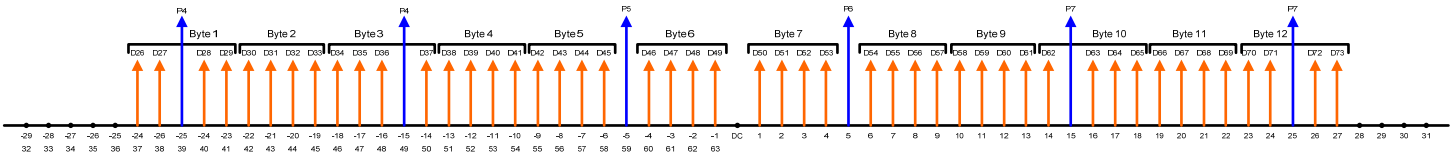


Figure 21 – UWB Model OFDM Frame

The IFFT block will only process frame sizes that are powers of two. For this model the desired frame size is 64 or 2^6. The model above (Figure 20) adds zeros to the incoming QPSK modulated data. This increases the size of each frame to 64. These zeros will be placed where there are zeros (black dots) and pilots (the longer arrows in blue) as described in Figure 21. To do this the selector block is used. This will rearrange an incoming frame of data based on the how the block is setup. For this model the setup is described in Figure 22. This array inserted into the Elements portion of the block setup will output the desired frame. This setup uses the zero based frame index (the positive numbers below the frame in Figure 20). All numbers between 1 and 48 hold data. All the numbers between 49 and 64 are zeros.

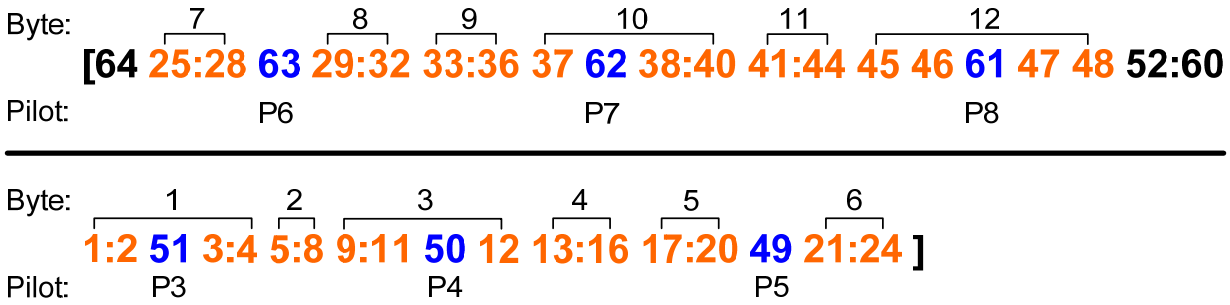


Figure 22 – OFDM Frame Construction Setup

The pilots for this model are all equal to one for simplicity at the receiver end. The pilot block is setup very similar to Figure 22 except that where there are pilots listed a one is placed, and where there is data information zero is placed. Therefore when these two frames are added together, the frame in Figure 21 is what results.

4.2.4.2 Inverse Fast Fourier Transform

The next step in the OFDM modulation is the IFFT (shown in Figure 19). The IFFT is the heart of the OFDM modulation as it inherently spaces all of the data orthogonally (shown in Appendix A) and transforms the data from the frequency domain into the time domain. This operation is done in Simulink with an IFFT block. This block is adjustable for speed or accuracy and will only accept data size in powers of two. For this model a 64 sample IFFT is used adjusted for speed.

4.2.4.3 Guard Interval

Within Luke Vercimak's Final Report he explains the following about guard intervals [20]:

> "…a guard interval is needed to compensate for the effects of a multi-path channel. The length of the guard interval should exceed the maximum excess delay of the multi-path echo channels. This guard interval prevents the effects of the multi-path channel from corrupting the system due to the properties of cyclic convolution. As long as the guard interval is longer than the maximum delay of the multi-path channel, the corruption of the channel should be limited to the guard interval. This guard interval is created by appending a cyclic repetition of the FFT output to the FFT output's beginning."

He goes on to state that the guard interval should be between 1/10 and ¼ the length of the symbol period. For the UWB model that means that a 16 sample guard interval be used. Since this model would not initially be rigorously tested against multi-path channel effects, 32 zeros were just appended to end of the IFFT operation. This may need to be updated if these models are to be used in a multi-path channel.

4.2.4.4 Preamble Insertion

As described in section 2.3.2.3, a preamble is very important for the communication system to synchronize. The preamble allows the receiver to know when information is coming. For this project only the packet synchronization portion of the full UWB preamble was implemented. The channel estimation portion of the preamble sequence was not implemented.

The UWB project under discussion will not be MB-OFDM. Also, the target center frequency for transmission is 3.432 GHz or in the first sub-band of the first band group. Since there is no frequency hopping and the information will stay in the first sub band, time frequency code #5 was used to implement the packet synchronization. This is because it is the only time frequency code designated to remain in the spectrum for this project. This can be seen in Table 5 [11].

Table 5 – Time Frequency Codes and Preamble Patterns for Band Group 1

TFC Number	Base Sequence / Preamble	BAND_ID (n_b) for TFC					
1	1	1	2	3	1	2	3
2	2	1	3	2	1	3	2
3	3	1	1	2	2	3	3
4	4	1	1	3	3	2	2
5	5	1	1	1	1	1	1
6	6	2	2	2	2	2	2
7	7	3	3	3	3	3	3

Time frequency code (TFC) #5 is defined in Table 6 [11]. This TFC was implemented into Simulink with the blocks shown in Figure 19. The length of a packet synchronization frame is N_{SYM}=165 as shown in Table 3 [11]. Note in Figure 19, that the size of the frame must be changed from 165 to 96 in order to

match the other input to the switch. This does not alter the preamble information at all, but merely rearranges it.

Table 6 – Time Frequency Code # 5

l	$s_{base}[l]$	l	$s_{base}[l]$	l	$s_{base}[l]$	l	$s_{base}[l]$
0	0,957 4	32	0,840 0	64	0,585 9	96	-0,852 8
1	0,527 0	33	1,398 0	65	0,305 3	97	-0,697 3
2	1,592 9	34	1,114 7	66	0,894 8	98	-1,247 7
3	-0,250 0	35	-0,473 2	67	-0,674 4	99	0,624 6
4	-0,253 6	36	-1,717 8	68	-0,890 1	100	0,768 7
5	-0,302 3	37	-0,847 7	69	-0,813 3	101	0,796 6
6	1,290 7	38	1,508 3	70	0,920 1	102	-1,280 9
7	-0,425 8	39	-1,436 4	71	-1,084 1	103	1,102 3
8	1,001 2	40	0,385 3	72	-0,803 6	104	0,425 0
9	1,770 4	41	1,567 3	73	-0,310 5	105	-0,161 4
10	0,859 3	42	0,029 5	74	-1,051 4	106	0,754 7
11	-0,371 9	43	-0,420 4	75	0,764 4	107	-0,669 6
12	-1,346 5	44	-1,485 6	76	0,730 1	108	-0,392 0
13	-0,741 9	45	-0,840 4	77	0,978 8	109	-0,758 9
14	1,535 0	46	1,011 1	78	-1,130 5	110	0,670 1
15	-1,280 0	47	-1,426 9	79	1,325 7	111	-0,938 1
16	0,695 5	48	0,303 3	80	0,780 1	112	-0,748 3
17	1,720 4	49	0,775 7	81	0,786 7	113	-0,965 9
18	0,164 3	50	-0,137 0	82	1,099 6	114	-0,919 2
19	-0,334 7	51	-0,525 0	83	-0,562 3	115	0,392 5
20	-1,724 4	52	-1,158 9	84	-1,222 7	116	1,286 4
21	-0,744 7	53	-0,832 4	85	-0,822 3	117	0,678 4
22	1,114 1	54	0,633 6	86	1,207 4	118	-1,090 9
23	-1,354 1	55	-1,269 8	87	-1,233 8	119	1,114 0
24	-0,729 3	56	-0,785 3	88	0,295 7	120	-0,613 4
25	0,268 2	57	-0,703 1	89	1,099 9	121	-1,546 7
26	-1,240 1	58	-1,110 6	90	-0,020 1	122	-0,303 1
27	1,052 7	59	0,607 1	91	-0,586 0	123	0,945 7
28	0,119 9	60	0,716 4	92	-1,228 4	124	1,964 5
29	1,149 6	61	0,830 5	93	-0,921 5	125	1,454 9
30	-1,054 4	62	-1,235 5	94	0,794 1	126	-1,276 0
31	1,317 6	63	1,175 4	95	-1,412 8	127	2,210 2

The .m file that implemented the variable uwb.Preamble that holds all of the information from Table 6 is listed in Appendix D. Within that .m file the values are added with j*1e-30 (j = $\sqrt{-1}$). This is done to trick Simulink into thinking these values are complex when they are not.

4.2.5 Interpolation

The interpolation block is a way to up-sample information using stages of up-sampling to reduce DSP computation time through multiple stages shown in Figure 23 [20].

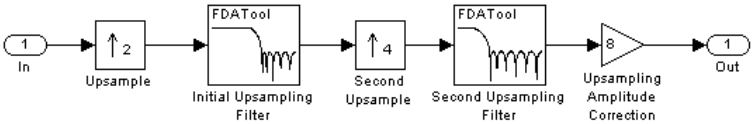


Figure 23 – Interpolation

The interpolation block shown takes a 12 kHz signal and up-samples the data to 96 kHz. This up conversion is necessary so that the data can then be quadrature modulated and then outputted to the

DAC running at 96 kHz. It is important that data in Simulink arrive at blocks with the same sampling rates.

4.2.6 Quadrature Modulator

A model of a quadrature modulator is shown in Figure 24. This model takes the incoming I&Q components and modulates them up to 32 kHz. The theory behind how quadrature modulation works can be seen in Appendix F. This is required because there is only one DAC on the TI DSP boards. With only one DAC the quadrature modulation had to be performed within the model. The output of a quadrature modulator is a real signal. If two DACs were used, one of the DACs could be used for the in-phase component while the other could be used for the quadrature-phase. The original goal was to use two DACs but with the limitations of the hardware this was impossible.

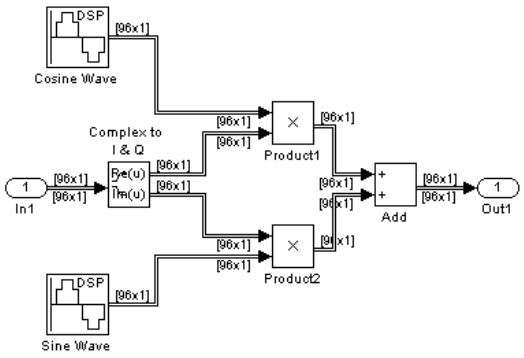


Figure 24 – Quadrature Modulator Model

There is an important setting that allows this model to work on the DSP board that must be noted. The setting is found in the DSP sine / cosine wave generators. By default these waves are calculated with a trigonometric function. This is very labor intensive on the DSP board and can cause undesired results. The preferred setting is for the waves to be generated using lookup tables. This may seem like a simple and obvious selection, but it caused the UWB team many headaches.

4.2.7 Output of Transmitter Model

The output of the transmitter model shown in Figure 17 is the on-board DAC. This DAC runs at a maximum speed of 96 kHz. This is the bottle neck of this project. With a faster DAC, the model could represent a UWB type transmission that would be closer to the UWB specification [see 11]. As the speed of the DAC increased, the transmitted bandwidth widens. This in turn allows the input data stream to also increase in speed (see section 4.2.1.1).

4.3 UWB Baseband Receiver Model Implementation

Once the UWB signals have been transmitted, a receiver is need to interpret those signals. The receiver must also compensate for the constantly changing channel characteristics that are occurring. The Simulink UWB receiver model shown in Figure 25 was created to do just that.

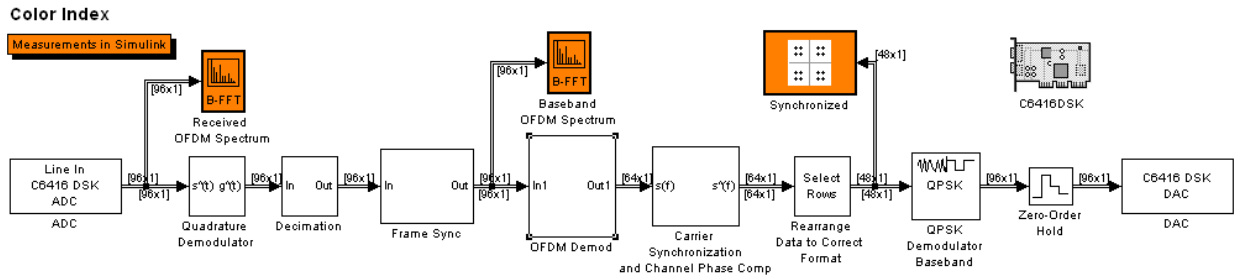


Figure 25 – Receiver model with 64 sample FFT

4.3.1 Input to Receiver Model

The input to the receiver model or baseband processing as shown in Figure 15 and 16, is the stream of data that was just transmitted through the channel. This is done within the model using an ADC as shown in Figure 25. This ADC is running at 96 kHz outputting data bits in groups of 96.

4.3.2 Quadrature Demodulation

The quadrature demodulation takes the incoming data and down converts it from the high center frequency of 32 kHz to its original center frequency of zero. This is an important step that allows the information to be baseband processed. The implementation of this model is shown in Figure 26.

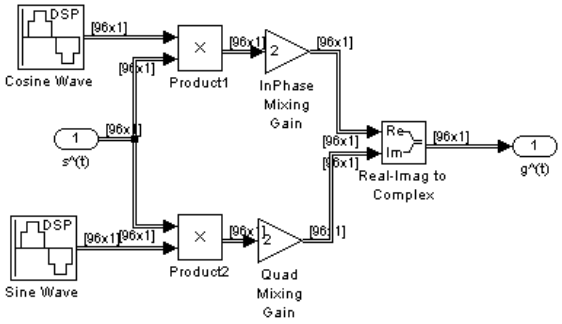


Figure 26 – Quadrature Demodulator

4.3.3 Decimation

Decimation is used to down sample the incoming information to a desired rate. The decimation used for this model is shown in Figure 27 [20].

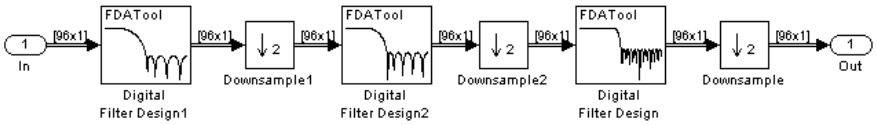


Figure 27 – Decimation

The decimation subsystem shown, down samples the incoming data from 96 kHz to 12 kHz using multiple stages to reduce the computation time required by the DSP processor. It is necessary to down sample because the all of the processing was determined to be carried out at a rate of 12 kHz (see section 4.2.1.1).

4.3.4 Frame Synchronization

Frame synchronization is essential in receiving the transmitted information correctly. Frame synchronization is accomplished with the use of the preamble that was inserted before the OFDM frames were transmitted. The model shown in Figure 28 is based on of model work done by Luke Vercimak in 2006 [20]. This model is different from Luke's because the results that he obtained could not be duplicated.

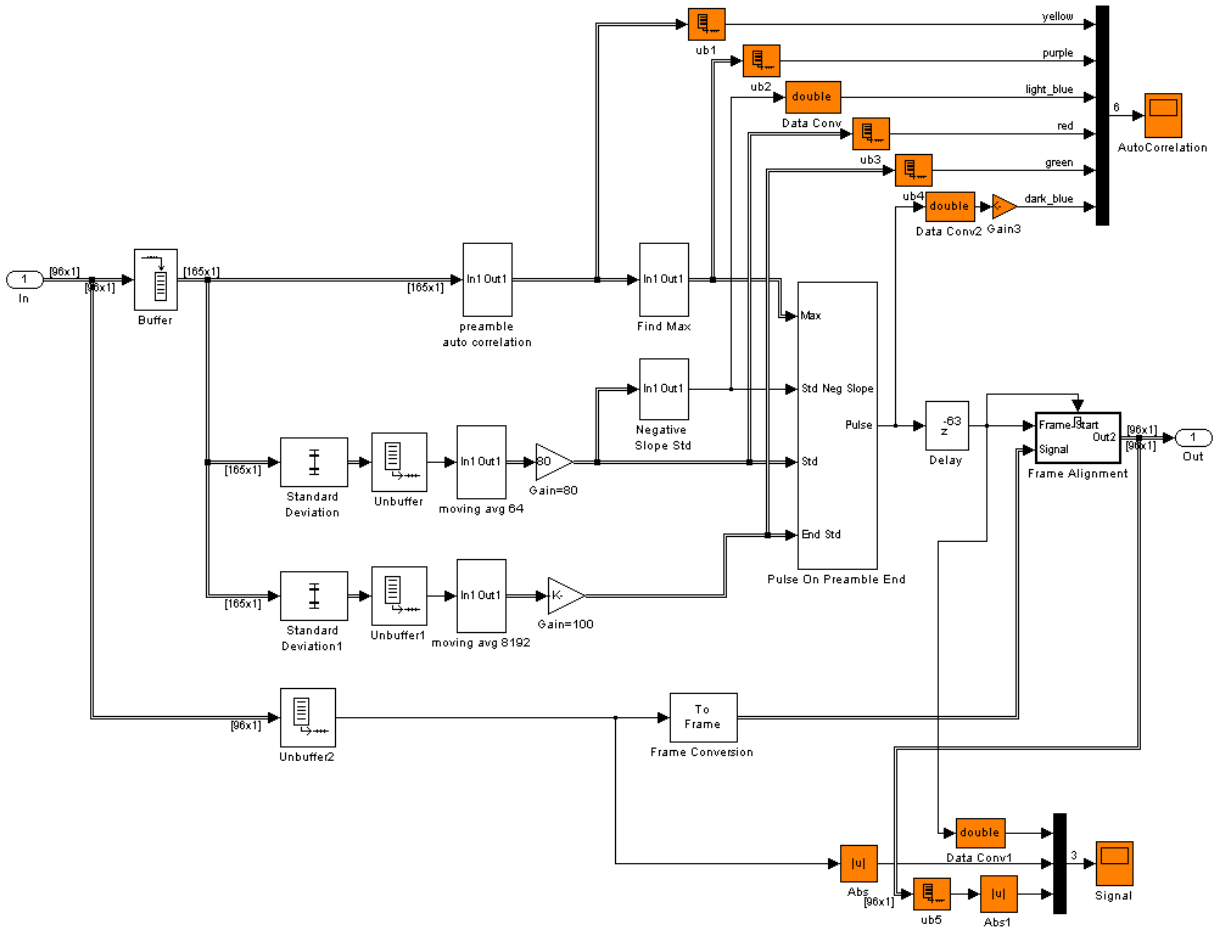


Figure 28 – Frame Synchronization

The frame synchronization shown in Figure 28 can be broken down into three main parts, the autocorrelation, the pulse on preamble end, and the frame alignment.

4.3.4.1 Autocorrelation

Autocorrelation is a way to see how similar a signal is to itself. TFC #5 is the signal that is compared against itself. The transmitter sends this known signal a set number of times. Then the receiver using autocorrelation can compare the incoming signal against a signal copy of TFC #5 that is stored on the receiver. The autocorrelation equation is shown in Equation 6 [20].

Equation 6 – Autocorrelation Equation

$$A(n) = \sum_{k=0}^{N} r(k+n)r^{*}(k+n+L)$$

In Equation 6, L is the length of TFC #5 (165), r is the length of the total preamble (TFC #5 repeated 24 times), and N is the length of the OFDM data (64 samples). Using this equation within Matlab, an autocorrelation was done on TFC #5, which is shown in Figure 29. The spike in the middle shows that the TFC #5 compared to itself is very similar. When actually transmitted the preamble the information will be distorted, but the spikes will still be visible.

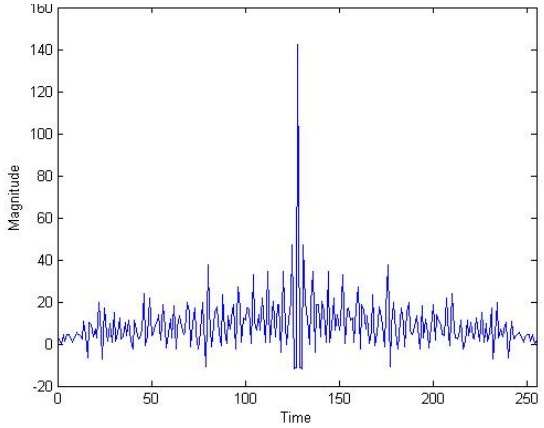


Figure 29 – Autocorrelation of TFC # 5

The implementation of Equation 6 is shown in Figure 30 [20]. This will take the incoming stream of information that was transmitted through the channel and compare it against the know preamble sequence, TFC #5.

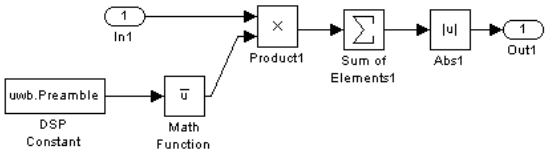


Figure 30 – Autocorrelation Implemented in Simulink

After the autocorrelation is found, the spikes needed to be found. This was done using a block shown in Figure 31 to find the maximum points from the autocorrelation equation [20].

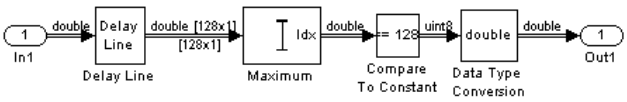


Figure 31 - Find Maximum Signal

The goal in using these autocorrelation spikes is to find the last autocorrelation spike because when the last autocorrelation spike is found, then that signifies the start of the first OFDM frame of data.

4.3.4.2 Pulse on Preamble End: Concept

To signify to the receiver model the start of the first OFDM frame, a pulse was created at the end of the preamble. The problems that were encountered with Luke's models for the frame synchronization came down to not being able to trigger on the exact location of the last preamble spike [20]. To work around this problem, a counting scheme was implemented where the number of spikes would be counted. In this way the last spike could easily be found. This may not be the most efficient solution for frame synchronization, but it works.

Figure 32 shows the scope from Figure 28 (all of the colored lines are labeled in Figure 28). Figure 32 shows how the pulse on preamble end is generated. The standard deviation of the autocorrelation spikes (yellow) was taken and used twice. One of the standard deviations was averaged much slower than the other. This was done to prevent any false positives for the pulse on preamble end. The standard deviation that follows the autocorrelation (yellow) closely is averaged with a moving average equation with a length of 64 (red). A longer moving averaging is performed on the same standard deviation that has a length of 2^{13} (green). The pulse on preamble end will only trigger when the short averaged standard deviation is above the long averaged standard deviation (or it will only trigger when the red signal is above the green signal).

To properly count the autocorrelation spikes the first spike must efficiently be found. No efficient way to definitively find the first autocorrelation spike could be found. However, there is a simple way to find the second autocorrelation spike. Since the standard deviation averaged with a length of 64 (red) follows the autocorrelation spikes, that signal's slope will always go negative between the first and second autocorrelation spike. When this signal's slope goes negative, a signal is sent high. That starts the counting sequence (light blue) for the autocorrelation spikes. Then once the count reaches the end or the last autocorrelation spike, the pulse on preamble end is sent high (dark blue).

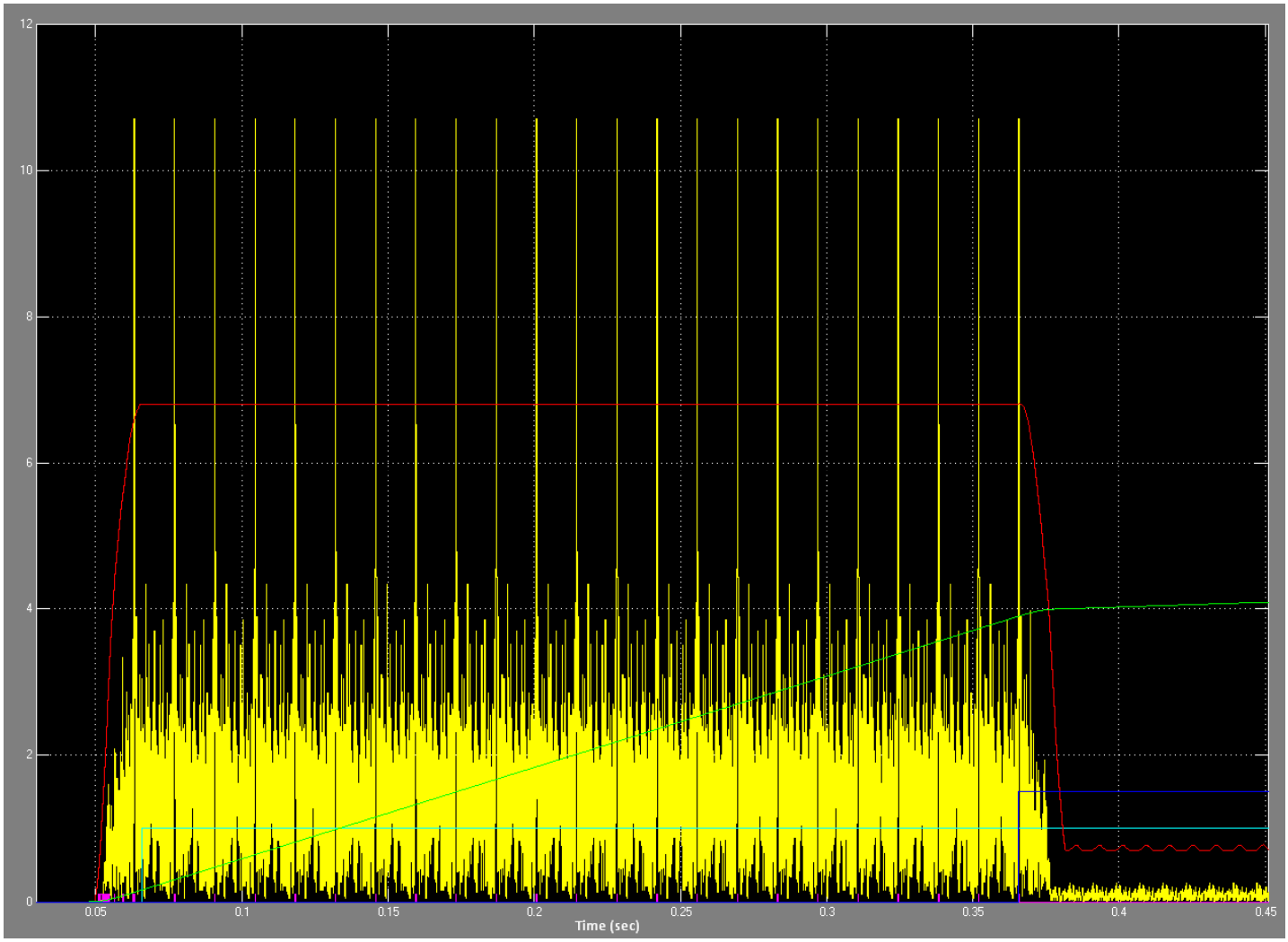


Figure 32 – Autocorrelation / Frame Scope (from Figure 28)

4.2.4.3 Pulse on Preamble End: Implementation

The implementation of the pulse on preamble end is shown in Figure 35. This subsystem takes in four signals from the frame synchronization model. They are the negative slope of the standard deviation (with a moving avg of 64), the autocorrelation maximums or spikes, and the two standard deviations with different moving averages. The negative slope was implemented in Figure 33.

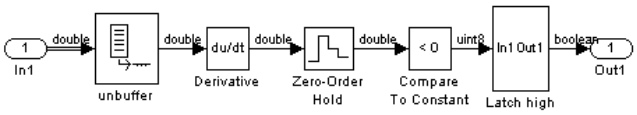


Figure 33 – Negative Slope Subsystem

The latch high block used throughout the pulse on preamble end implementation is shown in Figure 34.

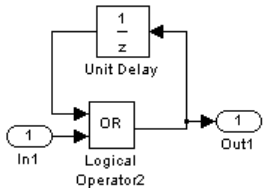


Figure 34 – Latch High

The model of the pulse on preamble end (Figure 35) is implemented the same way as described above (in section 4.2.4.2). The counter will not start counting autocorrelation spikes until the slope of the standard deviation goes negative and the shorter averaged standard deviation is above the longer averaged standard deviation (or the red signal is above the green signal in Figure 32).

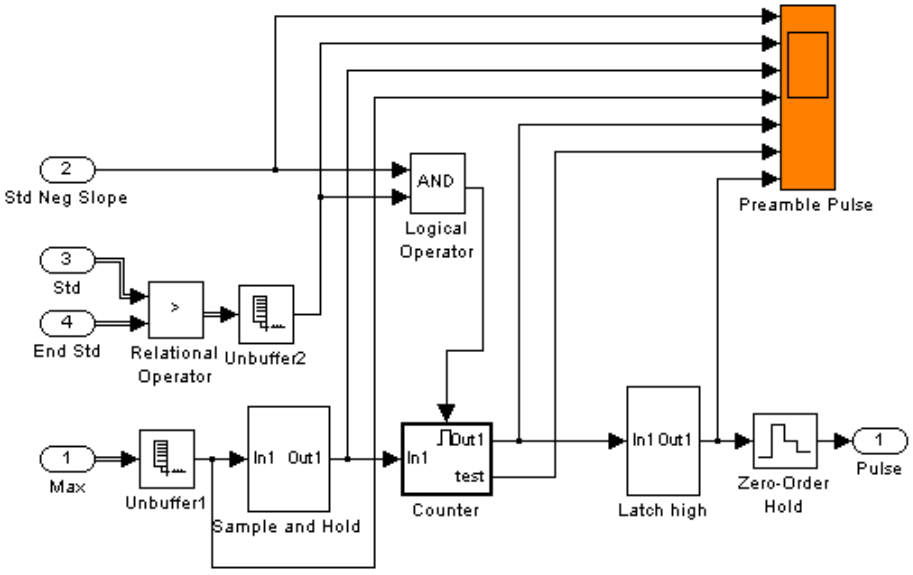


Figure 35 – Pulse on Preamble End

However, the signal that comes from the maximum signal finder block (Figure 31) could find 2 to 3 maximums around one autocorrelation spike. This phenomenon is shown at the bottom of Figure 36. This means that this signal must be sent through a custom sample and hold block to make sure that for each autocorrelation spike there is only one signal as shown in the top of Figure 36.

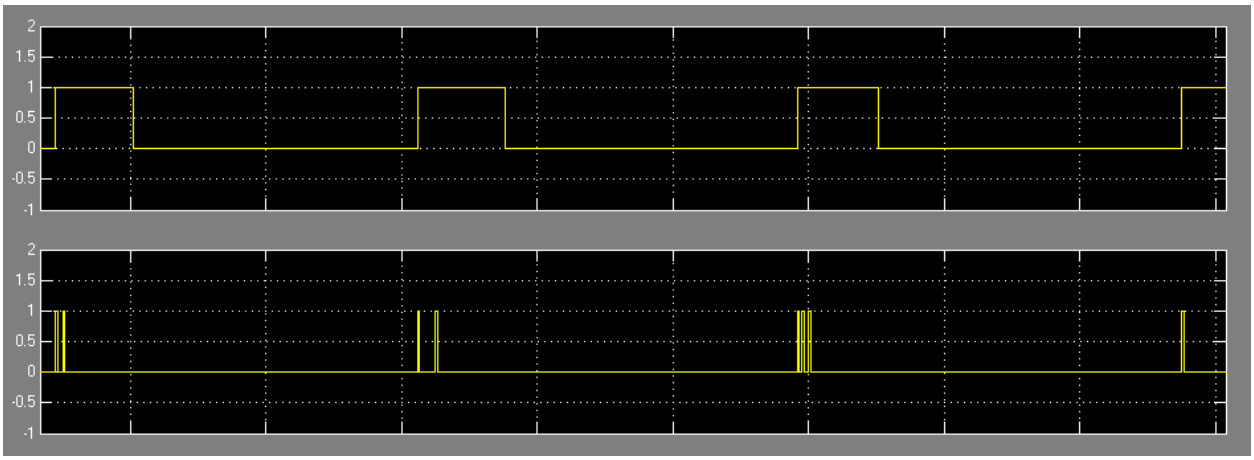


Figure 36 – Pulse on Preamble End Scope: Sample and Hold

The implementation of the sample and hold subsystem is shown in Figure 37. This subsystem is a latch with a reset. The latch is held high (or equal to one) for 30 samples as soon as an input is detected, using the N-sample Switch block. It is then reset after those 30 samples. The resulting signal (Figure 36) provides one pulse for each autocorrelation spike, which allows the counter to keep track of them.

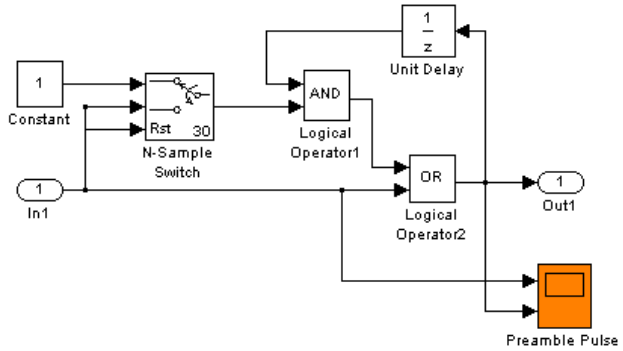


Figure 37 – Sample and Hold

The counter subsystem uses an enabled subsystem, where the blocks will only work when an enable signal is sent to the block (shown in Figure 35). This enable signal, as described above, is active between the first and the second autocorrelation spikes. Once active, the counter keeps a running sum tabulated until it is equal to 22. As soon as the 22nd pulse is found, the pulse on preamble end will be triggered.

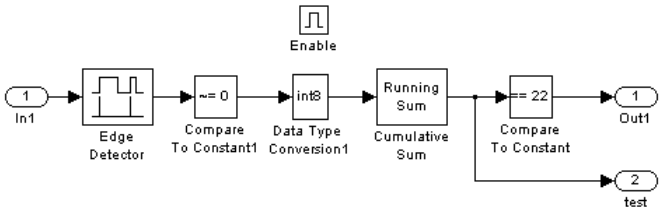


Figure 38 – Counter

The pulse on preamble end scope (Figure 35) display is shown in Figure 39 below. The signals from top to bottom, shown in Figure 39, displays:

1. when a negative slope is detected for the standard deviation signal (with a moving average of 64).
2. when the standard deviation with moving average 64 is greater than the standard deviation with a moving average of 2^{13} (or from Figure with the red signal is greater than the green signal).
3. output of the sample and hold model (Figure 37).
4. input to the sample and hold model (Figure 37) or output of the model that finds the autocorrelation spikes (Figure 31).
5. Boolean output of the counter.
6. running sum within the counter.
7. pulse on preamble end.

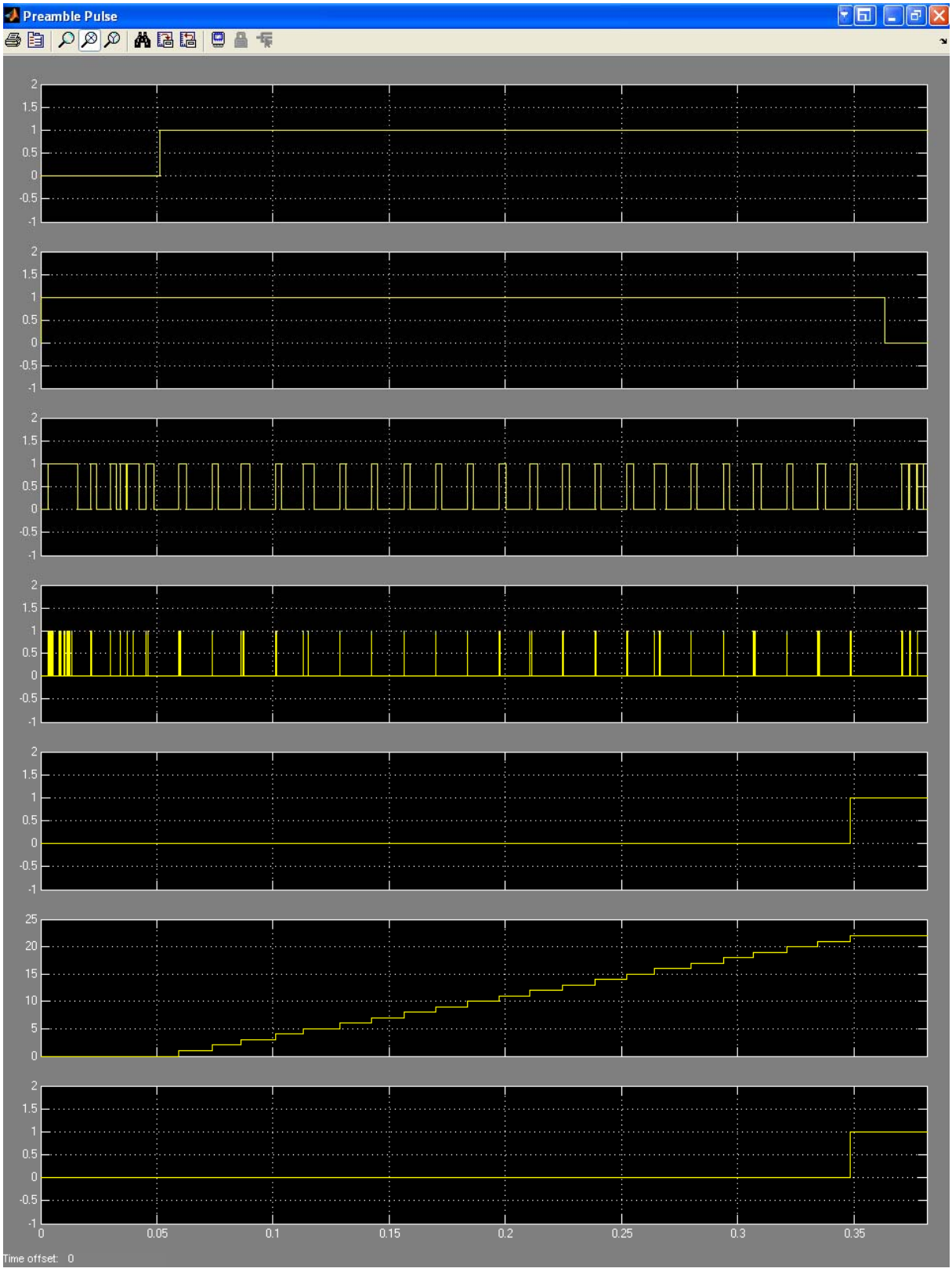


Figure 39 – Pulse on Preamble End Scope Overall

4.3.4.4 Added Delay for Alignment

After the pulse has been generated that signals the start of the first OFDM frame, the signal needs to be delayed as shown in Figure 40.

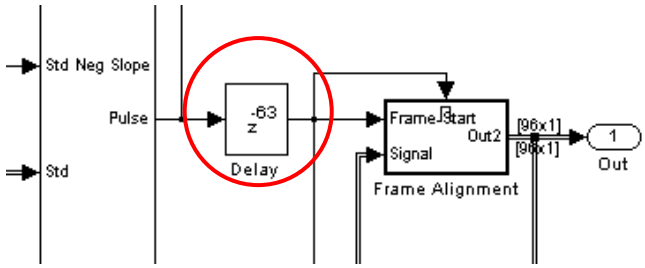


Figure 40 - Added Delay for Alignment

The delay is needed to offset the partial TFC that is sent because 165 symbols repeated 24 times does not factor into a 96 sized frame, shown in Equation 7. As interpreted from the UWB standard [11], the number of times the TFC symbols are to be repeated to make up one packet/frame synchronization portion of the preamble is 24 times.

Equation 7 – Number of frames of length 96 required to send preamble

$$\frac{165\,\text{symbols} \times 24\,\text{times}}{96\,\text{symbol frame}} = \frac{3960}{96} = 41.25\,\text{frames}$$

Equation 8 – Lowest common multiple of whole frames required to send preamble

$$\frac{165\,\text{symbols} \times 32\,\text{times}}{96\,\text{symbol frame}} = \frac{5280}{96} = 55\,\text{frames}$$

To transmit the preamble without transmitting a portion of the last TFC, the preamble would have to include 32 TFCs instead of 24 shown in Equation 8. For all of the models above only 41 frames of length, 96 were used. From Equation 9, this means that 23.85 TFCs were transmitted as a preamble. This is why only 23 preamble spikes are shown in Figure 32.

Equation 9 – Actual number of TFCs transmitted within the preamble

$$\frac{96\,\text{symbol frame} \times 41\,\text{frames}}{165\,\text{symbols}} = \frac{3936}{165} = 23.85\,\text{times TFC repeated in preamble}$$

The delay added compensates for the 0.85 portion of the time frequency code that occurs after the last autocorrelation spike. The exact delay was found through experimentation.

4.3.4.5 Frame Alignment

The frame alignment subsystem shown in Figure 41 is identical to the one used in Luke Vercimak's model [20]. The purpose of this subsystem is to properly align the frames to contain one whole OFDM frame. For more information on the inner workings of this subsystem, consult the Software Defined Radio Final Report [20].

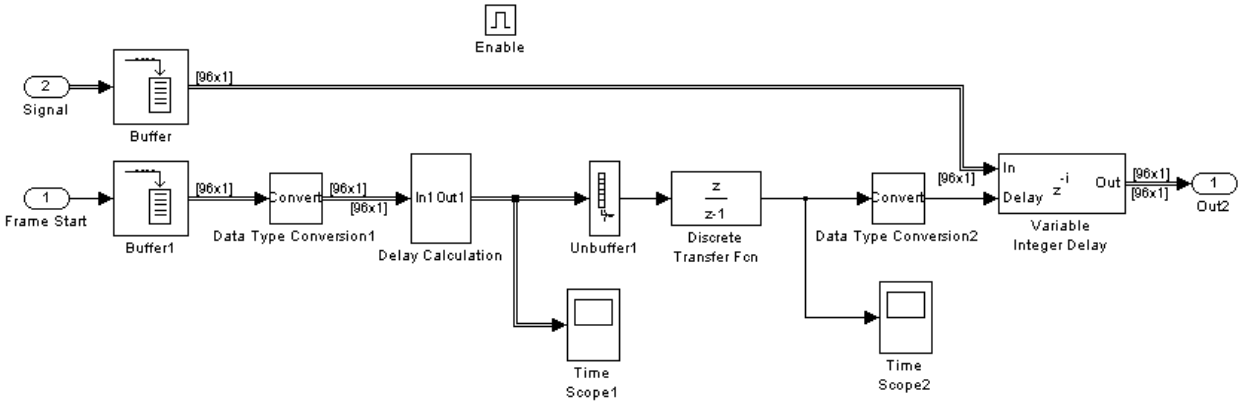


Figure 41 – Frame Alignment

4.3.5 OFDM Demodulation

Now that the signal has been properly frame synchronized so that the data is now in groups of OFDM frames, the information can be OFDM demodulated. This is the reverse of OFDM modulation. Instead of taking the IFFT the FFT is taken. The overall Simulink model for this subsystem is shown in Figure 42.

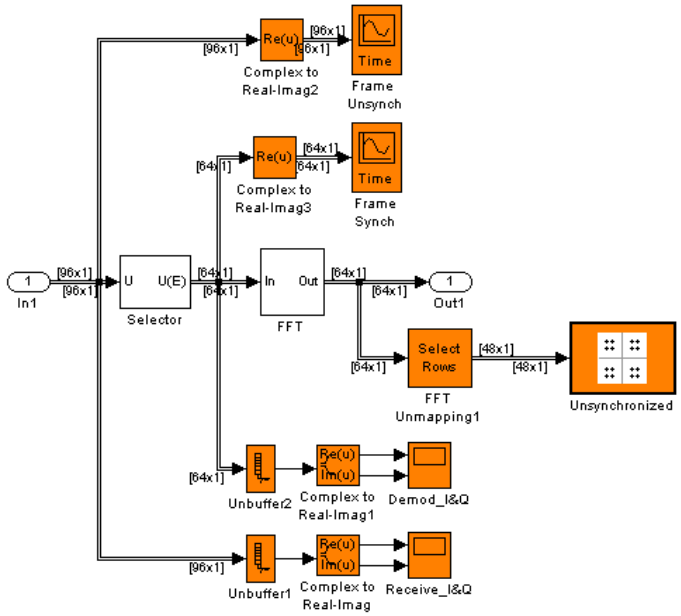


Figure 42 – OFDM Demodulation

The information that is coming into the model in Figure 42 contains one full OFDM frame, but that information does not start with the first symbol in the OFDM frame. This incoming frame is frame synchronized in that is contains all of the correct information. However, it still needs to be rearranged. That is done with the selector block. The transmitted frame that is desired is shown in Figure 43 (minus the zeros appended to the end of the frame). The received frame after frame synchronization and before the selector block is shown in Figure 44. The received frame before it is sent to the FFT is shown in Figure 45, which matches up with the transmitted frame from Figure 43.

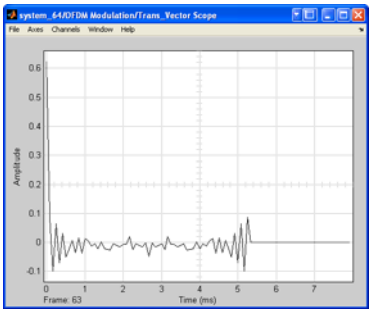


Figure 43 - Transmitted OFDM Frame with Zeros Appended

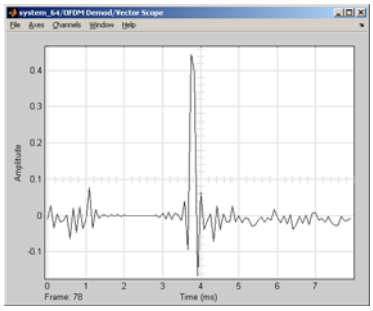


Figure 44 – Received OFDM Frame Before the Selector Block

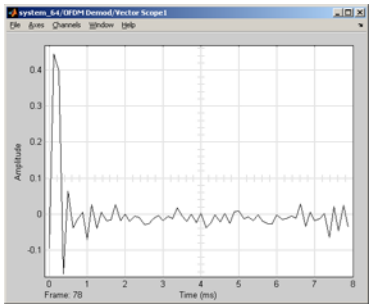


Figure 45 – Received OFDM Frame After the Selector Block

Once the OFDM frame shown in Figure 45 goes through the FFT, the signal is back in the form shown in Figure 21 with subcarriers.

4.3.6 Carrier Synchronization and Channel Phase Correction

The carrier synchronization and channel phase correction subsystem is used to compensate for the (assumed) linear phase delay that is added to the OFDM frame when it is transmitted. Luke Vercimak's model was used for this subsystem [see 20]. The linear phase delay can be shown visually in Figure 46 [20].

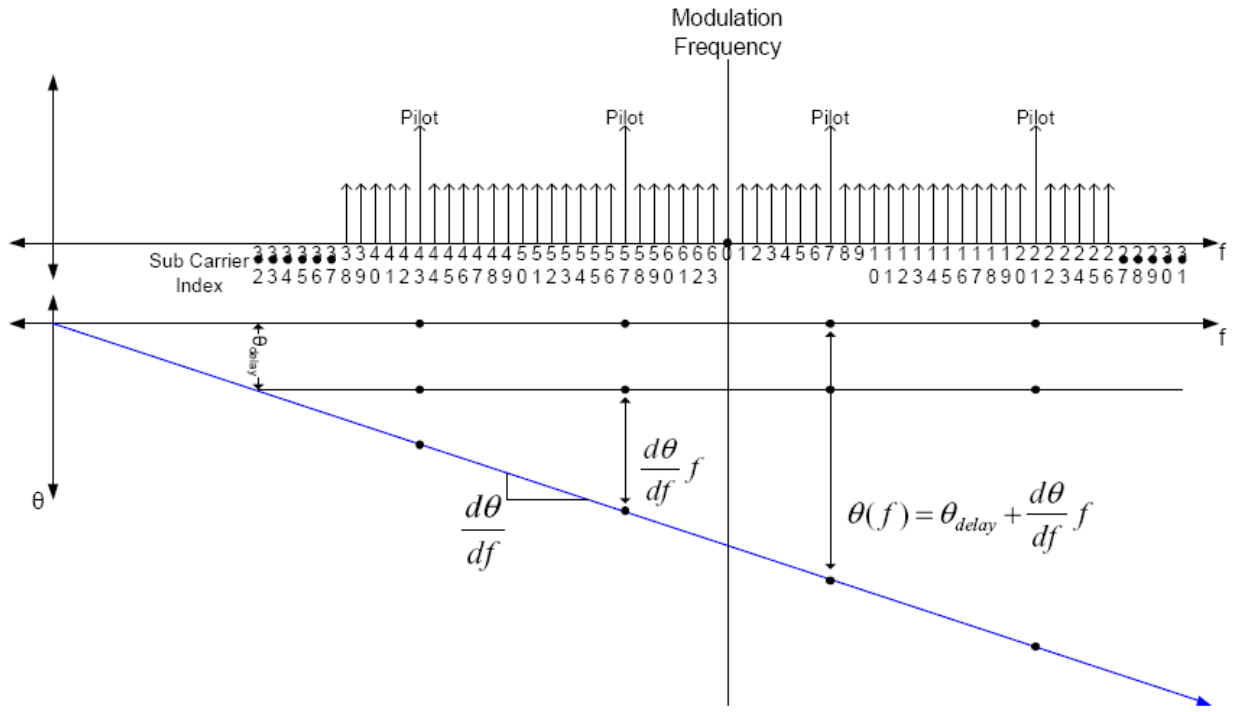


Figure 46 – Linear phase delay across OFDM frame

Figure 46 does not represent the OFDM frame that was used in the UWB models (Figure 21), but it does show how the symbol synchronization and channel phase compensation is implemented. The Simulink model for this subsystem is shown in Figure 47 [20].

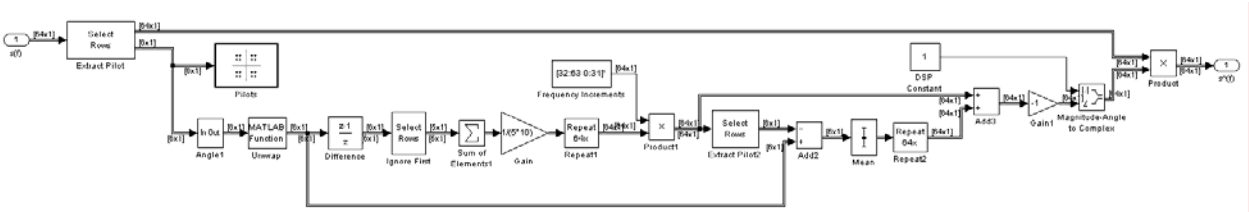


Figure 47 – Symbol Synchronization and Channel Phase Compensation Subsystem Model

The functionality of the blocks shown in Figure 47 is to calculate the linear phase delay, using the known pilot signals, and apply it to the data subcarriers. The linear phase delay is defined in Equation 10 [20].

Equation 10 – Linear Channel Phase Delay

$$\theta(f) = \theta_{delay} + \frac{d\theta}{df} f$$

To modify the model shown in Figure 47 to work for the UWB models some of the setting were modified. They are as follows (from left to right on Figure 47):

- The Extract Pilots Block was modified to {[1:64], [40,50,60,6,16,26]} which corresponds to the six pilots shown in Figure 21.
- The Ignore First Block was modified to {[2:6]}.
- The Gain Block which performs an averaging function was modified to 1/(5*10).
- The Select Rows Block was modified to {[40,50,60,6,16,26]} to reselect the pilots.

Once these modifications were complete the information was mapped correctly on the constellation. Figure 48 shows the transmitted constellation, while Figure 49 shows the received unsynchronized constellation. Then the constellation after symbol synchronization is shown in Figure 50.

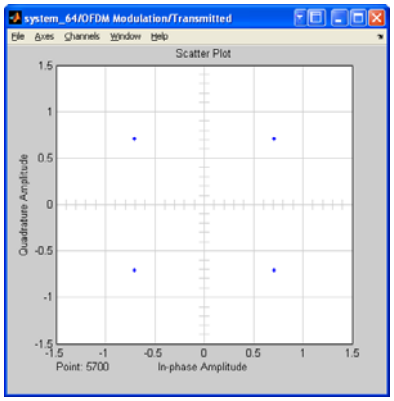


Figure 48 – Transmitted π/4 QPSK Constellation

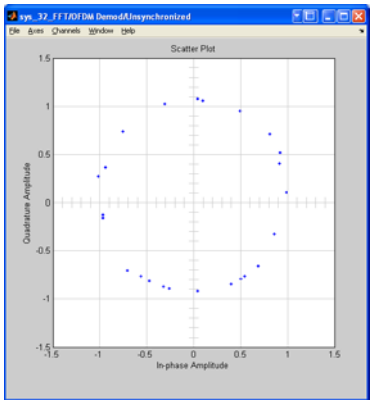


Figure 49 – Received Unsynchronized π/4 QPSK Constellation

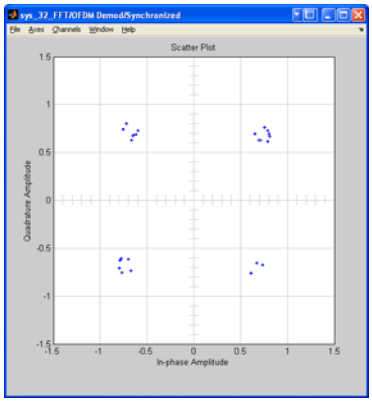


Figure 50 – Received Synchronized π/4 QPSK Constellation

For more information concerning the inner working of the model shown in Figure 47 and the reasons why the modification above were made please consult Software Defined Radio Final Report [20].

4.3.6.1 Extract the Data

After the OFDM frame has been demodulated and symbol synchronized, the data subcarriers need to be extracted. The information post symbol synchronized is still in the form of Figure 21 with pilot subcarriers interlaced. To extract this information a Select Rows block is used with the following setup:

{[38:39, 41:49, 51:59, 61:64, 2:5, 7:15, 17:25, 27:28]}

This will extract the data subcarrier symbols from the frame from byte 1 to 12 (or left to right) as show in Figure 21. Since all of the information being outputted from this block is data it can then go to the π/4 QPSK demodulator.

4.3.7 π/4 QPSK or 4 QAM Demodulation

The QPSK demodulation for the receiver is done with a pre-fabricated Simulink block. This block is available within the communications block set for Simulink. The QPSK demodulation block performs all of the required steps to demodulate the received data from the constellation shown in Figure 50 to its original bits. The size of the frames going into this block are 48 symbols wide. The size of the frames coming out of this block are 96 bits wide because 1 symbol is equal to 2 bits (as described above in section 2.3.2.1).

4.3.8 Output of Receiver Model

The output of the receiver model shown in Figure 15 & 16 is a DAC on the DSP board. This DAC runs at a maximum speed of 96 kHz. If this portion were able to be downloaded to the DSP boards this information would then be compared against the original transmitted signals. This comparison could (theoretically) be made with a microphone for the input to the transmitter board and a headset for the output from the DAC from the receiving board. Also, another check that could have been done would have been to transmit ASCII characters. The fastest character transmit speed would have been about 3 characters per second (125 bits/sec).

4.4 RF Hardware Implementation

The final hardware subsystem was the RF subsystem. The objective was to modulate both the I&Q signals to an RF carrier frequency. Instead of using two mixers for both the I&Q channel, a direct quadrature modulator was researched and acquired to take care of both channels. A schematic diagram of the quadrature modulator is shown below in Figure51. The quadrature modulator accepts the I&Q signals, as well as a local oscillator (LO) signal from the voltage controlled oscillator (VCO). The local oscillator signal is split into two signals 90 degrees out of phase and each of these signals gets mixed with the I&Q signals. These products are then summed to produce the final RF output. It can be shown (in Appendix F) that the output of a quadrature modulator is a Single Sideband (SSB) signal.

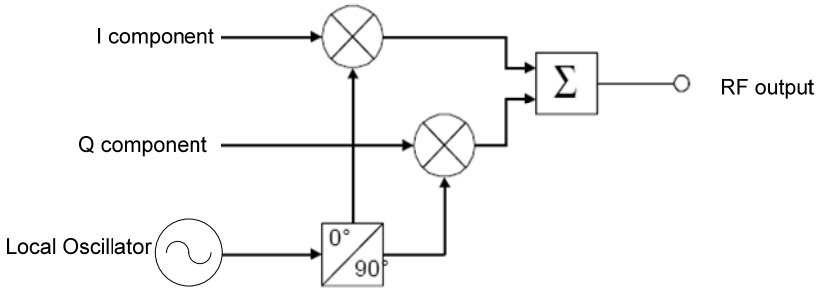


Figure 51 – Direct Quadrature Modulator

The quadrature modulator used for the project was the Hittite Microwave HMC497. The complete specifications are shown below in Table 7 [21]. The main specifications of interest are its wide frequency range of 100 MHz to 4 GHz, output power between 0 dBm and 6 dBm, and wide baseband bandwidth from DC to 700 MHz. This model was chosen because it allows easy modulation of the baseband signals up to 3.432 GHz, and its wide baseband bandwidth capabilities allow it to accommodate UWB to its full potential. The only pitfall was the high output power. However, since this was a scaled-down implementation of UWB, the output power requirements were ignored so that a proof of concept could be achieved. To meet the FCC requirements of a power spectral density of -41.3 dBm/MHz, the output power would need to be -14.1 dBm since the instantaneous bandwidth is 528 MHz in an actual system.

Table 7 – Quadrature modulator specifications

Parameter	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Units
Frequency Range, RF		450 - 960			1700 - 2200			2200 - 2700			3400 - 4000		MHz
Output P1dB		+8			+8			+7			+6		dBm
Output Noise Floor		-161			-159			-157			-150		dBm/Hz
Output IP3		+22			+22			+20			+17		dBm
Output Power	+4	+6		+3	+5		+2	+5		0	+3		dBm
Carrier Feedthrough (uncalibrated)		-32			-30			-26			-24		dBm
Sideband Suppression (uncalibrated)		43			42			33			22		dBc
LO Port Return Loss		25			15			14			13		dB
RF Port Return Loss		11			20			17			11		dB

Parameter	Conditions	Min.	Typ.	Max.	Units
RF Output					
RF Frequency Range		100		4000	MHz
RF Return Loss			15		dB
LO Input					
LO Frequency Range		100		4000	MHz
LO Input Power		-6	0	+6	dBm
LO Port Return Loss			15		dB
Baseband Input Port					
Baseband Port Bandwidth	3 dB Bandwidth with 50Ω source.	DC		700	MHz
Baseband Input DC Voltage (Vbbdc)		+1.4	+1.5	+1.6	V
Baseband Input DC Bias Current (Ibbdc)	Single-ended.		90		µA
Single-ended Baseband Input Capacitance	De-embed to the lead of the device.		4.5		pF
DC Power Requirements	See Test Conditions Below				
Supply Voltage (Vcc1, Vcc2)		+4.5	+5.0	+5.5	V
Supply Current (Icc1, Icc2)			168		mA

The next major RF component is the direct quadrature demodulator. This device performs the exact opposite function of the quadrature modulator. It receives the RF SSB signal, splits it into two paths, and mixes each path with an LO signal, one of which is 90 degrees out of phase, as shown in Figure 52.

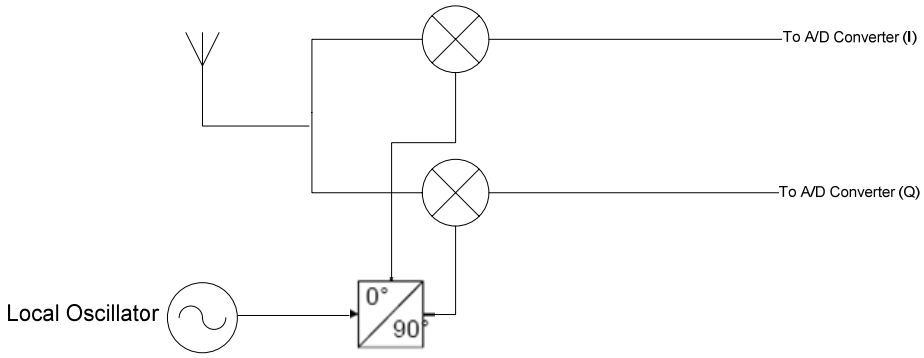


Figure 52 – Direct Quadrature Demodulator

The model that was used was the Hittite Microwave HMC597. Its data sheet highlights are shown below in Table 8.

Table 8 – Direct Quadrature Demodulator Specifications

Parameter	Min.	Typ.	Max.	Units
RF Input Frequency (Direct LO)		0.1 - 4.0		GHz
Input P1dB		12		dBm
SSB Noise Figure		15		dB
Input IP3		+25		dBm
Input IP2		+60		dBm
Conversion Gain		-3.5		dB
LO to RF Leakage @ +3 dBm LO		-66		dBm
IF Port Bandwidth		DC - 250		MHz
IF Output Impedance (Diff.)		400		Ohms
LO Input Power		-6 to +6		dBm
LO/RF Return Loss		12/12		dB
DC Supply		+5V @ 200 mA		mA

Finally, a voltage controlled oscillator was obtained from Hittite Microwave – the HMC389. It featured an output power of around 4.7 [dBm], tunable frequency range of 3.35 3.55 [GHz], low harmonics, and it was easy to use. The data sheet is shown below in Table 9.

Table 9 – Local Oscillator Specifications

Parameter	Min.	Typ.	Max.	Units
Frequency Range		3.35 - 3.55		GHz
Power Output	1.5	4.7		dBm
SSB Phase Noise @ 100 kHz Offset, Vtune= +5V @ RF Output		-112		dBc/Hz
Tune Voltage (Vtune)	0		10	V
Supply Current (Icc) (Vcc = +3.0V)		41		mA
Tune Port Leakage Current			10	µA
Output Return Loss		6		dB
Harmonics 2nd 3rd		 -7 16		 dBc dBc
Pulling (into a 2.0:1 VSWR)		3.3		MHz pp
Pushing @ Vtune= +5V		-3		MHz/V
Frequency Drift Rate		0.4		MHz/°C

5. Project Issues / Challenges

This project, like many, had unforeseen issues. These issues fell into two distinct categories, hardware and software limitations. These limitations meant that the project ultimately could only be a proof a concept. Essentially, the combination of limitations of from the hardware and software prevented the receiver portion of the project from being physically realized.

5.1 Hardware Limitations

The hardware limitations included the slow sampling times on the DACs and the maximum allowable simultaneous periodic sampling times. The DAC on the TI TMS320C6414T DSP Starter Kit ran at a maximum speed of 96 kHz. At this rate the maximum theoretical bandwidth is 96 kHz. This theoretical bandwidth does not include any quadrature modulation. Since this theoretical bandwidth is three orders of magnitude lower than the required bandwidth of a full UWB system, the project then became a proof of concept.

Another hardware related issue that was encountered, also had to do with the TI DSP boards. These boards can only handle 7 simultaneous periodic sample rates. For the transmitter model this was not an issue because it had between 5-6 simultaneous periodic sample rates. However, the receiver model had around 13-15 simultaneous periodic sample rates, so it could not be loaded onto the TI DSP boards.

5.2 Software Related Issues

The software also contributed to this problem. MathWorks Simulink has very little direct user control over the number of sample times created. There are ways to create models that use less simultaneous periodic sample times (by decreasing the complexity of the model).

A periodic sample rate is created with Simulink when any conversion or operation is done on the rate at which data flows through a model. For example if data is coming into a model at 96 kHz and the signal is sent through a sample and hold block running at 48 kHz, then Simulink automatically creates two sampling rates. One of these rates runs at 96 kHz while the other runs at 48 kHz. This operation is just one example; it is possible to have multiple sample times created from one operation or block.

For the receiver in this UWB project, there were many different conversions and calculations required. This left the number of periodic sampling times too great to be ported to the DSP platform.

6. UWB Project Results

For this UWB Research and Implementation Project, there were three main areas of focus. These areas were the baseband transmitter model, the baseband receiver model, and the RF hardware. Table 10 shows the final schedule for the UWB project.

Table 10 – Final Schedule

Description	Percentage Complete
Overall	90%
Hardware	92%
Research/Purchase DSP Development Kits	100%
Research/Purchase RF Hardware	100%
Test Hardware	76%
TMS320C6416 DSP Starter Kit	100%
RF Local Oscilattor	100%
RF Modulator	90%
RF Demodulator	90%
Testing Overall System	0%
Simulink Models	87%
Simple Tranceiver Pair	100%
Sample Time Configuration	100%
Frame Synchronization	100%
Frame Alignment	100%
Carrier Synchroization	100%
Port Transmitter to DSP Platform	100%
Port Receiver to DSP Platform	10%

Overall, the project was about 90% done on both the hardware and software side. There were only a few limitations that prevented us from fully accomplishing our goals as discussed in section 5.

6.1 Baseband Transmitter Model Results

The transmitter model was successfully ported to the DSP boards using CCS. Since there was only one DAC on the TI DSP platform the quadrature modulation was done within the model. There were many different tests done to see how high the information could be quadrature modulated within the model before the spectrum was adversely affected. The signals in Figure 53 and 54 are quadrature modulated up to 12 kHz and 32 kHz respectively and have no adverse effects due to modulation. However, in Figure 55, the effects of aliasing start to appear. This makes sense as the absolute theoretical maximum frequency that could be outputted with a 96 kHz DAC is 48 kHz and the spectrum is centered at 48 kHz in Figure 55.

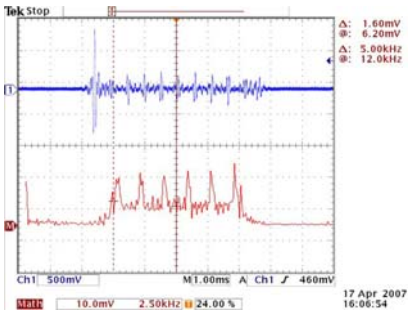


Figure 53 – Quadrature Modulated to 12 kHz

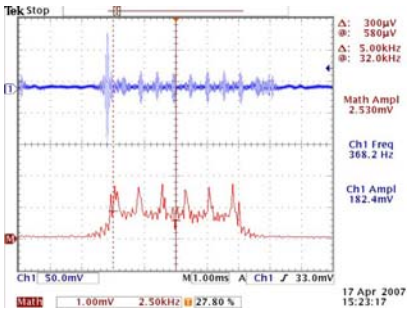


Figure 54 – Quadrature Modulated to 32 kHz

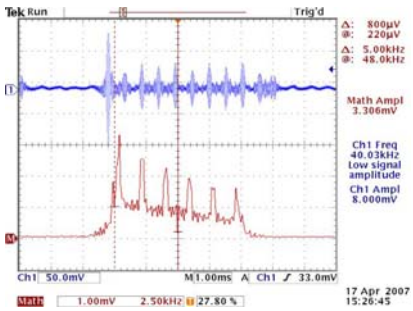


Figure 55 – Quadrature Modulated to 48 kHz

The aliasing frequencies can even be seen in Figure 56. From these results the maximum quadrature modulated frequency obtainable with these boards and models is 32 kHz because the center frequency must be a multiple of 96 kHz and less that 48 kHz (as shown in Figure 55 & 56).

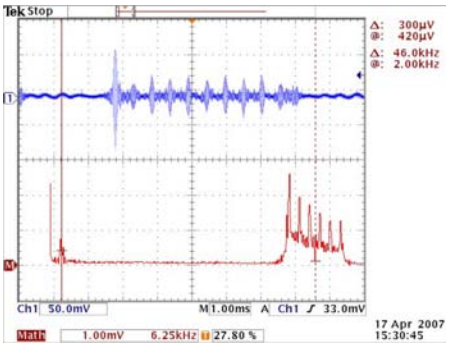


Figure 56 – Aliasing Frequencies for Quad Modulation at 48 kHz

6.1.1 Sampling Frequency Phenomenon

When the output of the DAC was viewed on the oscilloscope the oscilloscope representation of the time domain sequence would change with time. Upon investigating this further, the time domain sequence would cycle through three different sequences repeatedly. The output should have remained like the time domain outputs seen in Figure 54. The three sequences are shown below in Figures 57 through 59.

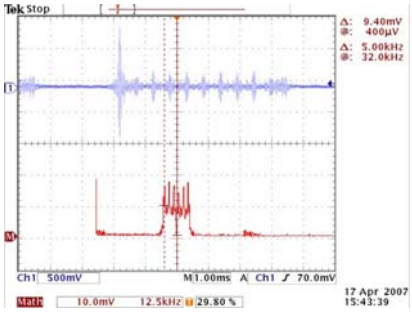


Figure 57 – Sequence #1 (Correct Sequence) Mod to 32 kHz

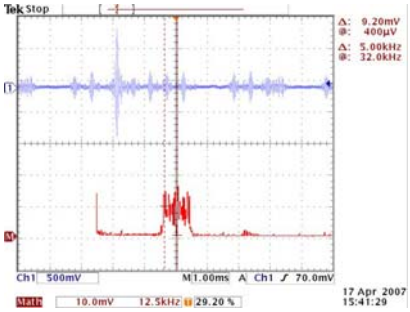


Figure 58 – Sequence #2 Modulated to 32 kHz

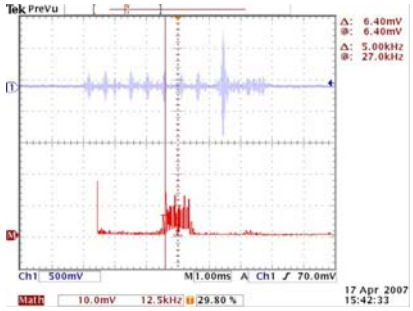


Figure 59 – Sequence #3 Modulated to 32 kHz

Since the three sequences were repeated infinitely and that the first sequence (Figure 57) is what the output should always remain at, it was determined that the reason for this cycle is a miss match in sampling times. The digital oscilloscopes sample at a 1 Msps while the output of the signal is 96 kHz. In view of the fact that 96 kHz is not an exact multiple of 1 Msps, the sampling time of the oscilloscope drifts. This drifting is visually seen as the sequence described above. This drifting in and then out of the sampling time also explains why the correct output is periodically seen. This phenomenon is not a huge deal as the information never gets corrupted, but is noted as to clear confusion for anyone repeating such experiments.

6.2 Baseband Receiver Model Results

The baseband receiver model is working. There are however a few quirks about the model that prevented extensive testing of the overall system model. These quarks include all ones being transmitted for first few frames and periodic loss in symbol synchronization.

6.2.1 Initial Data from Model

The receiver model will synchronize the incoming data correctly as shown above, but the first set of data that is output from the receiver is always all ones no matter what the input to the transmitter model was. The model is transmitting a frame of OFDM data approximately every 8 msec, the information coming in is at the rate of 125 Hz. This means that 1 incoming bit takes 8 msec. The model needs to process a group of information at a time. Inherently the model needs 96 bits to process a new frame. So the transmitter will output the same frame until enough new data is collected. Therefore a set of data takes approximately 96 * 8 msec = 0.8 sec to update. Consequently, 96 of the same OFDM frames are transmitted before the OFDM frame changes to the new data.

This is inherently the problem. The reason that all ones are initially transmitted is because the transmitter does not have enough information to process a frame when it first starts up. Because the transmitter needs 96 bits to form a frame and that takes 0.8 sec as described above. The solution to this problem is to use a faster DAC which would consequently allow the input rate to be increased.

6.2.2 Periodic Loss in Symbol Synchronization

The receiver model once receiving periodically loses synchronization. This happens during a transition from one set of data to the next. This could be because the model is slowed down so much that it temporarily stops working every time a new batch of data is processed. Or, it could be that the calculation blocks were set up incorrectly causing this loss. Either way, the solution to this quirk is unknown. This is the main reason that extensive system testing was not done. This periodic loss in synchronization makes the BER outrageously large. For detailed simulations with various channel characteristics see the Software Define Radio Final Report [20].

6.3 RF Hardware Results

6.3.1 Overall Test Setup

At the end of the project, the RF subsystem was completely up and running. It took plenty of experimental effort before the system was working. In order to test the remaining RF components, a suitable source for the QPSK I&Q signals was needed, since the DSP system could not provide both signals. The HP ESG-D signal generator was used to create these signals for testing. The ESG is capable of providing almost any kind of modulation scheme – including custom QAM and QPSK signals. The built-in π/4 QPSK scheme was used for testing. This allowed the selection of the bit pattern which would map to different locations on the constellation diagram. The back of the ESG had I&Q connections, which were hooked to the respective ports on the quadrature modulator using BNC to SMA adapters and cables. Furthermore, to save time and hardware resources (i.e. power supplies), the same signal generator provided the local oscillator signal for the quadrature modulator as well. This was convenient in that it allowed a quick way to change both the frequency and power output of the LO signal. Also, the same LO signal was used for both the modulator and demodulator, so synchronization would not be an issue. Figure 60 shows the overall hardware connections for the RF subsystem.

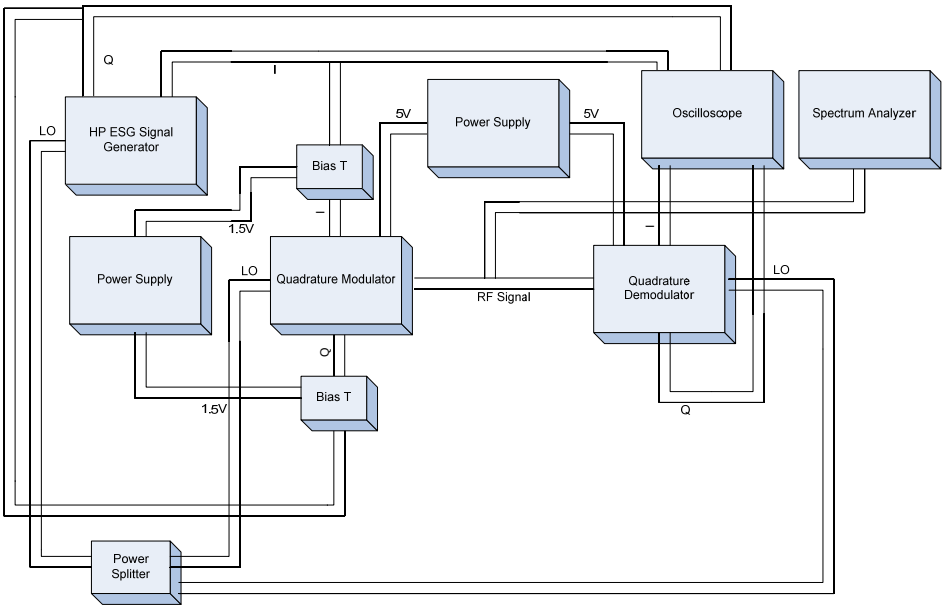


Figure 60 - Overall RF subsystem layout for testing

Another challenge with the RF hardware was interpreting the data sheets provided by Hittite. They were often vague– especially in explaining what external connections to the PCBs were needed. Specifically, it was hard to find out where to hook VCC to the modulator and demodulator. Nick Schmidt made some DC Molex connectors to allow easy access to the small pins on the PCBs. To eliminate confusion for future projects, the images below (Figures 61 & 62) show where the DC power needs to be connected for both the modulator and the demodulator (5 volts).

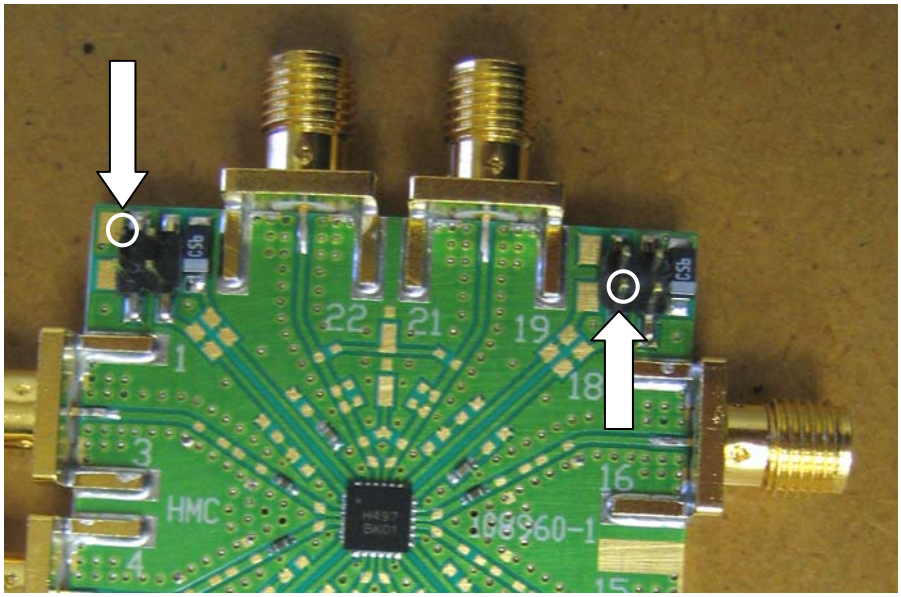


Figure 61 – Quadrature Modulator VCC connections (5V)

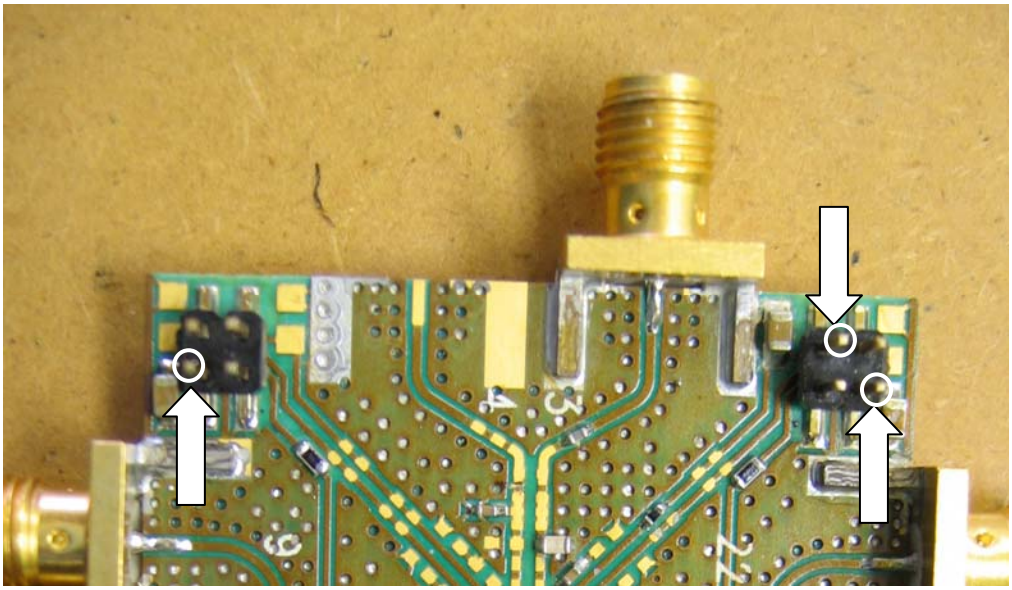


Figure 62 – Quadrature Demodulator VCC connections (5V)

6.3.2 Local Oscillator Results

The first test conducted was on the local oscillator from Hittite. The LO needed two power supplies – one for the bias voltage, and one for the tuning voltage. Several tests were performed with the spectrum analyzer, including frequency vs. tuning voltage (Figure 63), power vs. tuning voltage (Figure 64), and harmonic level measurements (Figure 65). The experimental results are shown below, along with comparisons with the data sheet. The oscillator performed as expected, and matched the data sheet results.

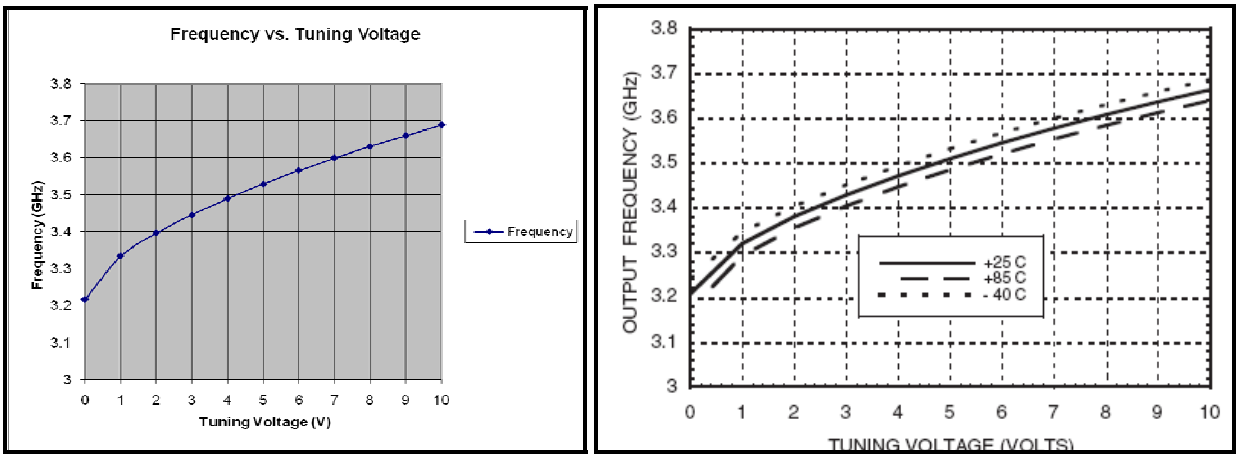


Figure 63 – Frequency vs. Tuning Voltage (left image shows experimental results, right image shows data sheet)

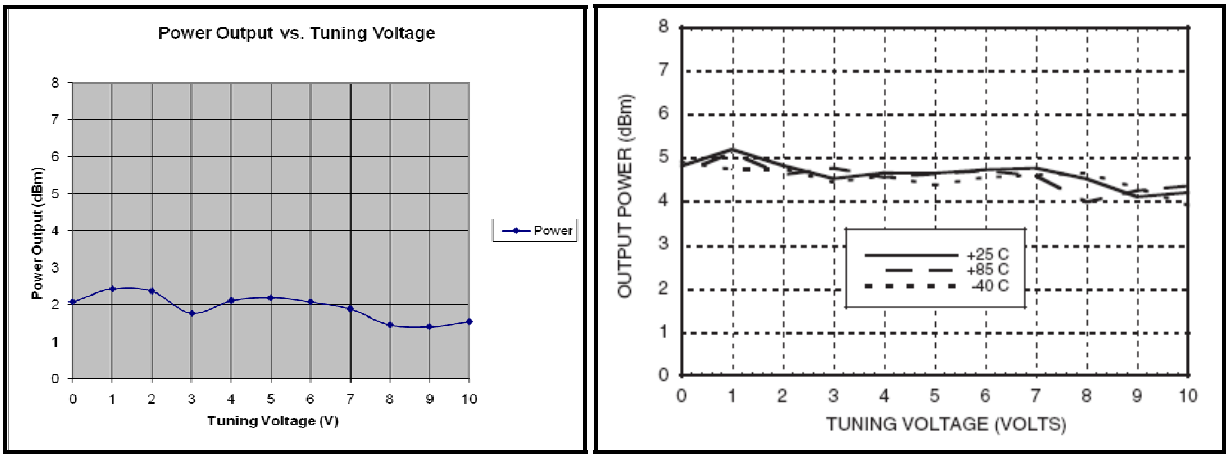


Figure 64 – Output power vs. Tuning Voltage (image shows experimental results, right image shows data sheet)

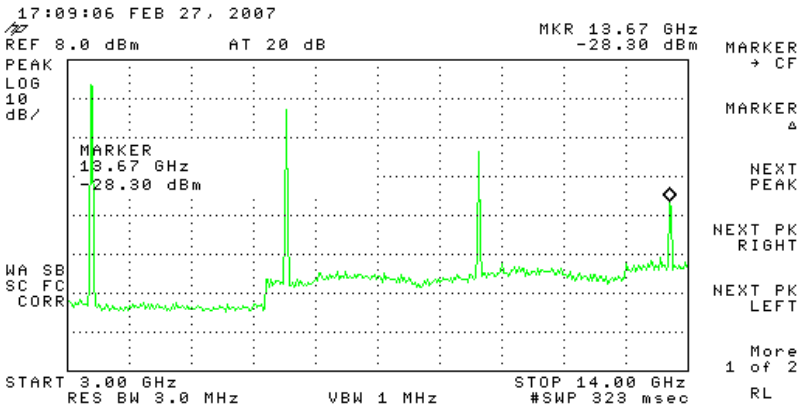


Figure 65 – Harmonic Frequencies and power levels

Figure 65 shows the LO signal with its second, third, and fourth harmonics and their power levels. The measured power levels of the harmonics were within specifications according to the data sheet.

6.3.3 Quadrature Modulator Results

After the VCC is hooked up as shown above, the next step was to put a DC bias of 1.5 [V] on each I&Q signal per the data sheet. To do this, bias-T networks were used, which allow the RF and DC signals to be combined. The inputs to a bias-T are the RF and DC signals respectively, and the output of the bias-T is the RF and DC signals combined. This is a very important step, since power levels will not be correct without the bias. Also note that the quadrature modulator has differential output ports for the I&Q channels, but only one side of the I&Q ports need to be used, as was determined experimentally.

Due to time constraints, the results obtained were not analyzed thoroughly, but the modulator is working correctly. Also note the images shown below were obtained *before* it was realized that DC biases needed to be applied, therefore, the power levels are extremely low. However, you can see how the RF spectrum out of the modulator changes with varying bit streams on the ESG signal generator.

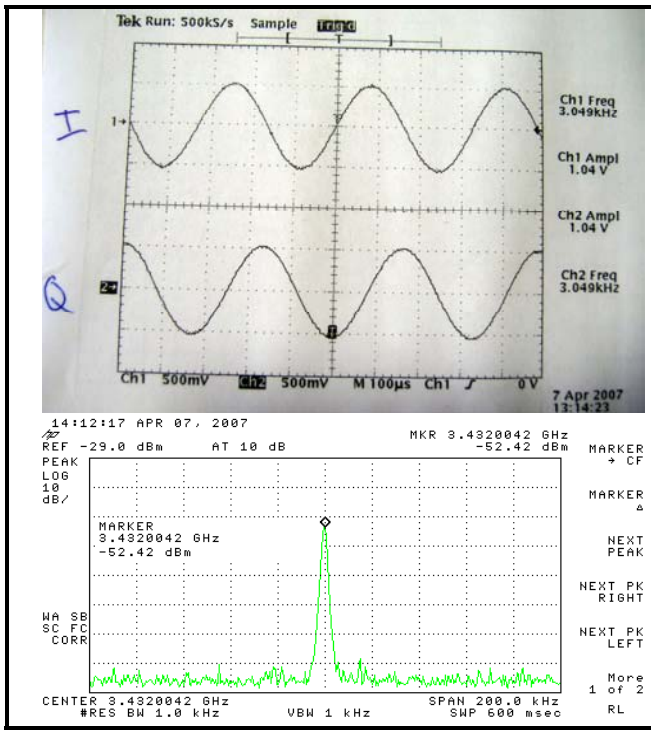


Figure 66 – I&Q signals for "00" and output spectrum

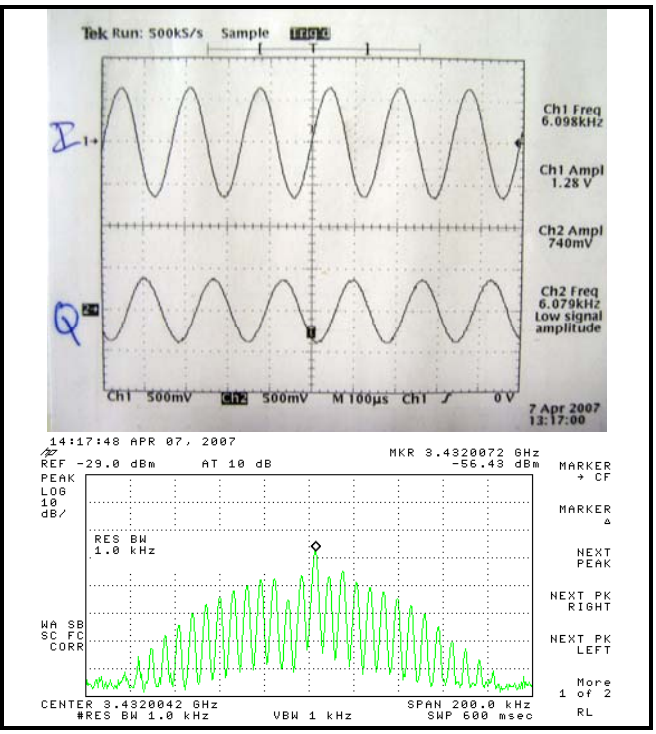


Figure 67 – I&Q signals for "01" and output spectrum

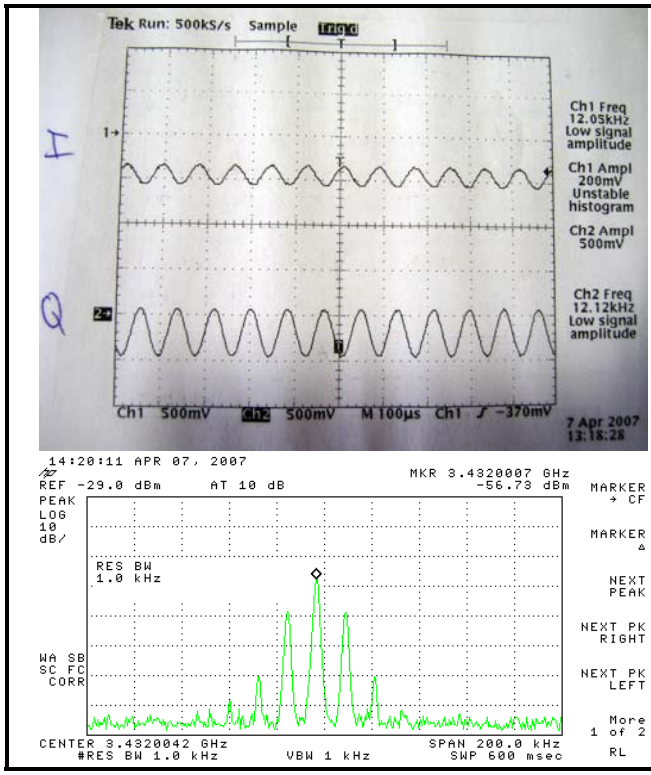


Figure 68 – I&Q signals for "10" and output spectrum

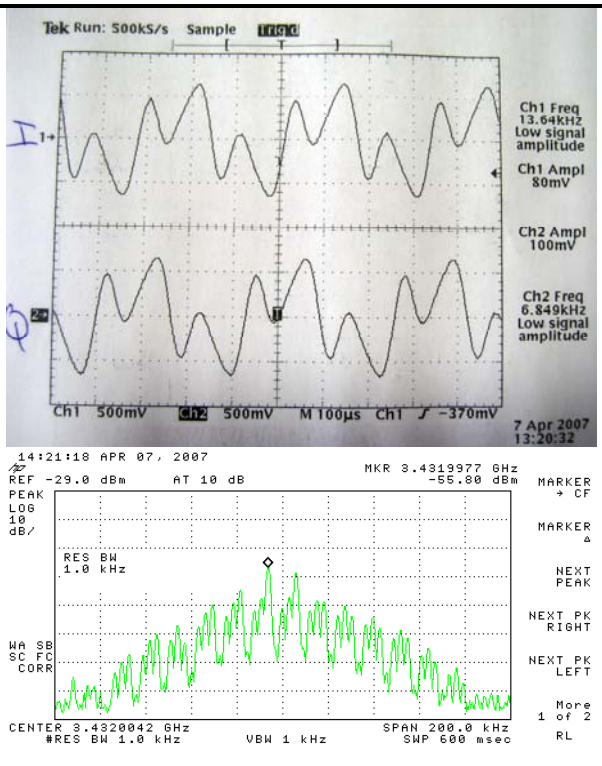


Figure 69 – I&Q signals for "11" and output spectrum

The previous four spectrum plots (Figures 66 to 69) were obtained without biasing the I&Q inputs to the modulator. Once the modulator was biased, the output power went up to approximately 0 dBm. Figure 70 below shows one spectrum plot with the bias now applied.

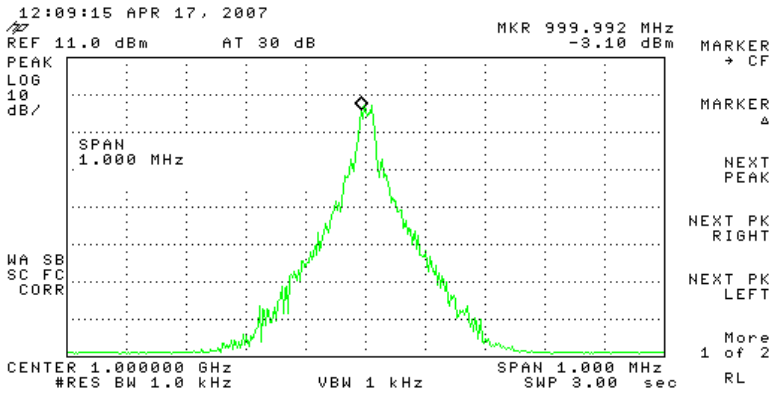


Figure 70 – Output spectrum with a DC bias on the I and Q channels

6.3.4 Quadrature Demodulator Results

After much time was spent figuring out how to bias the demodulator, it was successfully tested in conjunction with the modulator. The time domain oscilloscope was used to compare the I and Q channels from both the modulator and demodulator. Also note that the demodulator has connections for differential I and Q inputs, but these weren't used, and are not needed. Once the chip is biased, the I and Q signals were seen to be exactly the same as what was shown coming out of the modulator, except smaller in amplitude. The comparison is shown below in Figures 71and 72.

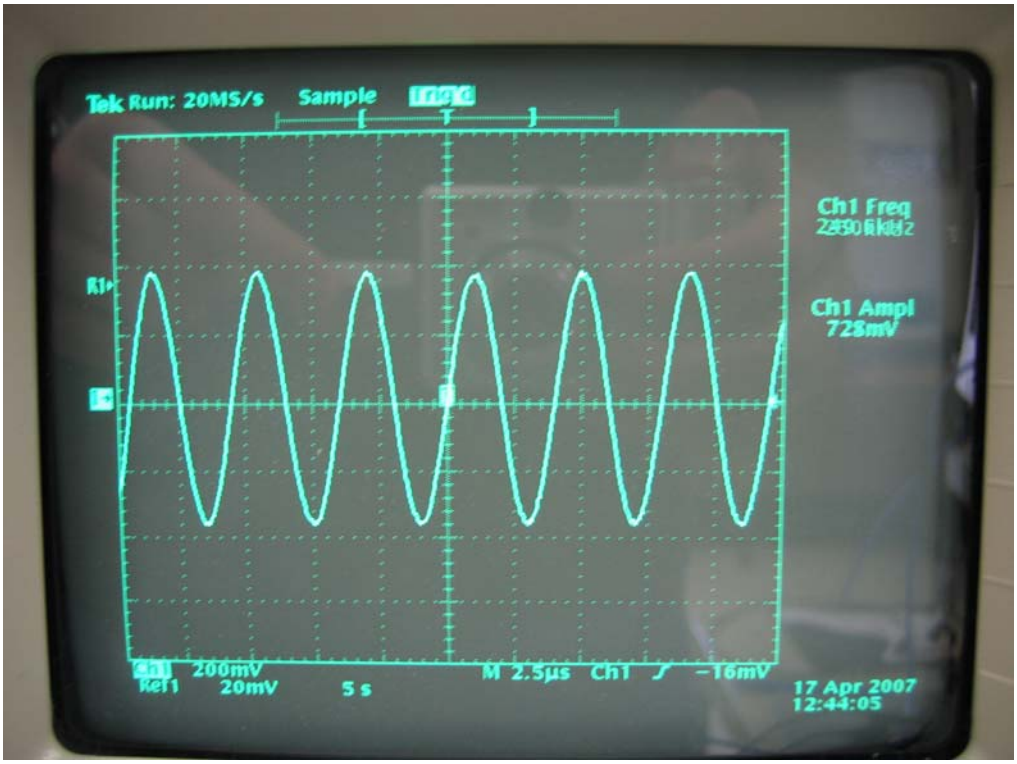


Figure 71 – I channel from modulator

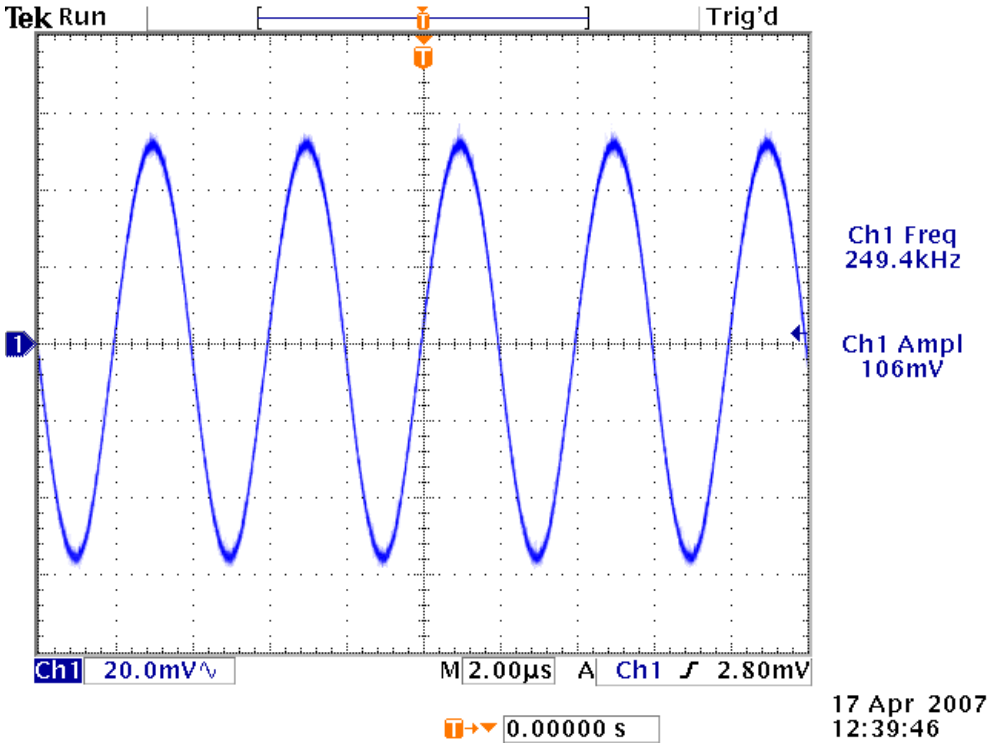


Figure 72 – I channel from demodulator

7. Recommendations for Future Work

There is plenty of work left to continue this project for many future senior capstone projects. There are several things future groups could do to continue work on UWB. Some things that future groups could do include but are not limited to: purchasing a full development kit (if one is available at the time), integrating TI DSP daughter boards with the DSP kits to increase DAC converter functionality, or a group could purchase an entirely different DSP development kit.

7.1 UWB MB-OFDM Development Kit

As was noted earlier, development kits that were of interested for a MB-UWB project were not quite done with development in time for this project. However, one such kit may be available by the Fall of 2007, and by 2008 several models should be available to choose from. By purchasing a kit, time would be saved by trying to develop a system from scratch with slow hardware not specifically designed for the tasks required. In addition, testing would prove the usefulness of UWB with such a kit. Finally, with a development kit in hand, the project team could integrate the system for use for a high-level objective, such as transmitting high quality video wireless at high speeds.

7.2 Integrate TI DSP Daughter Boards with Simulink

If a full-scale development kit is not available to purchase at that time, the best thing a future team could do is purchase the TI Daughter Boards for use on the current DSP platforms. These kits allow speeds near 400 Msps on the D/A converters – which would increase overall speed and bandwidth tremendously. This would allow the system to more closely approximate the actual UWB specifications mentioned earlier. However, this undertaking is no walk in the park. The primary obstacles to overcome are interfacing the daughter boards with Simulink using C-code. This would involve having a much higher understanding of the TI DSP platforms as interrupts and routines would have to be written as well.

7.3 Use another DSP Platform

Finally, one last option would be to research and purchase another DSP development kit, other than models from TI. Several such kits exist that sound promising. One kit in particular was found late in the second semester. This kit is produced by Sundance Multiprocessor Technology Ltd [13]. These kits utilize the same processor as the 6416 development kits that run at 1 GHz, but uses faster converters on the kit itself. This is a special university offer called the Software Defined Radio Kit. It is based on the SMT8036 hardware platform and is fully integrated with Mathworks Simulink. Some of the specifications for this kit include the following:

- Two 14-bit ADCs each running at 105 MSPS

- Dual channel TxDAC running at 400 MSPS
- Ultra-high performance TMS320C64x DSP processing

The reason these specifications look so promising are because they make up for the down falls of the current hardware. The limitations of the current TI TMS320C6416T DSP platforms are the converter speeds that run at a max of 96 kHz. With these boards and the models above, the UWB team believes a physically realizable UWB communication system that is also close to meeting the full UWB specification [11] can be created within one senior capstone project.

Also with a successful completion of a communication project similar to the goals for this project, future projects beyond that could also be explored. These projects would include UWB antenna projects, a very hot area right now as UWB is gaining interest. Another project could be a higher level project that could use the system as a sophisticated range finder or to transmit high quality streaming video. The potentials are seemingly endless.

There are a few questions that should be asked before buying any board which include, but are not limited to the following:

- Does the board have 2 DAC channels to output both an in-phase component and a quadrature-phase component simultaneously?
    - This is necessary to get both the I&Q components to the quadrature modulator.
- Is there a Simulink block set that goes along with the DSP platforms that integrate the on board peripherals or converters automatically?
    - This should be a yes. If it is a no, then strongly reconsider purchasing (see section 7.2).
- How many simultaneous periodic sample rates are supported in multi-tasking mode with these DSP platforms?
    - The more the merrier. The TI platforms only supported 7 which was insufficient for our receiver model. Note the model could be re-worked to only have 7 periodic sample rates, but it would not be easy. This could also be a limitation of Code Composer Studio.

8. Conclusion

The Ultra-Wideband Research and Implementation Project's goals were two-fold: to research the technology while raising awareness within the Bradley community, and to develop a scaled-down version of an OFDM communications system which can be adapted for UWB use. UWB allows extremely fast data rates of up to 480 Mbps, allowing USB2.0 devices to go wireless. To do this, it uses a wide bandwidth for transmission, while consuming very little power. The power spectral density is so low that it allows co-existence with current wireless technologies. UWB will likely revolutionize the consumer electronics industry, especially with the explosion of High Definition devices. These devices would require the large bandwidths to accommodate the high speeds that UWB can offer in order to go wireless.

9. References

[1] Aiello, Robert, and Anuj Batra. Ultra Wideband Systems: Technologies and Applications. Burlington, MA: Elsevier Inc, 2006.

[2] Green, Evan R., and Sumit Roy. System Architectures for High-Rate Ultra-Wideband Communication Systems: a Review of Recent Developments. Intel Corporation. Hillsbro, OR: Intel Labs, 2004. 06 Feb. 2007 <http://www.intel.com/technology/comms/uwb/download/W241_Paper.pdf>.

[3] Trubin, Julian. Guglielmo Marconi: the Invention of Radio. 2003. 6 Feb. 2007 <http://www.juliantrubin.com/bigten/marconiradioexperiments.html>.

[4] Wolff, Richard. "A Spark of a Good Idea! Ultra Wideband Radio Past, Present and Future." Deparment of Electrical and Computer Engineering. Advanced Topics in Communications Systems. Montana State University. 19 May 2003. 6 Feb. 2007 <http://www.coe.montana.edu/ee/rwolff/EE548/EE548-S06/UWB/Bilbao.pdf>.

[5] First Report and Order: Revision of Part 15 of the Commission's Rules Regarding Ultra-Wideband Transmission Systems. Federal Communications Commissions. FCC, 2002. 6 Feb. 2007 <http://www.llnl.gov/ipac/technology/profile/sensor/MicropowerImpulseRadar/images/FCC_rules.pdf>.

[6] Wood, Stephen. UWB Standards. WiMedia Alliance. WidMedia Alliance White Paper, 2006. 6 Feb. 2007 <http://www.wimedia.org/imwp/idms/popups/pop_download.asp?contentID=9042>.

[7] Couch, II, Leon W. Digital and Analog Communication Systems (Seventh Edition). Upper Saddle River: Pearson Prentice Hall, 2007. 367-372.

[8] Wilson, James M. "Ultra Wideband Technology Update At Spring 2003 IDF." Intel Developer Update Magazine (2003): 1-9. <http://www.intel.com/technology/magazine/communications/wi01031.pdf>.

[9] Ultrawideband: High-Speed, Short-Range Technology with Far-Reaching Effects. MultiBand OFDM Alliance (MBOA). 2004. 6 Feb. 2007 <http://www.wimedia.org/imwp/idms/popups/pop_download.asp?ContentID=6527>.

[10] Wilson, James M. Ultra-Wideband / a Disruptive RF Technology? Intel Research & Development. Intel Corporation, 2002. 6 Feb. 2007 <http://www.intel.com/technology/comms/uwb/download/Ultra-Wideband_Technology.pdf>.

[11] Standard ECMA-368: High Rate Ultra Wideband PHY and MAC Standard. ECMA International. 2005. 6 Feb. 2007 <http://www.ecma-international.org/publications/standards/Ecma-368.htm>.

[12] Batra, Anuj, Jaiganesh Balakrishnan, Roberto Aiello, Jeffrey R. Foerster, and Anand Dabak. "Design of a Multiband OFDM System for Realistic UWB Channel Environments." IEEE Transactions on Microwave Theory and Techniques 52.9 (2004): 2123-2138.

[13] "University Promotions: Offer 1; Software Defined Radio Solution." Sundance. 2007. Sundance Multiprocessor Technology Ltd. 20 Apr. 2007 <http://www.sundance.com/web/files/static.asp?pagename=usa-smt8036>.

[14] "Bluetooth Specification: Bluetooth Technology Overview." The Wireless Directory. 3 Dec. 2003. 20 Apr. 2007 <http://www.thewirelessdirectory.com/Bluetooth-Overview/Bluetooth-Specification.htm>.

[15] "Bluetooth Basics." Bluetooth.Com. 2007. 20 Apr. 2007 <http://www.bluetooth.com/Bluetooth/Learn/Basics/>.

[16] Foerster, Jeff, Evan Green, Srinivasa Somayazulu, and David Leeper. Ultra-Wideband Technology for Short- or Medium-Range Wireless Communications. Intel Architecture Labs. Intel Corporation, 2001. 2 Feb. 2007 <http://www.intel.com/technology/itj/q22001/pdf/art_4.pdf>.

[17] Yang, Liuqing, and Georgios B. Giannakis. "Ultra-Wideband Communications: an Idea Whose Time Has Come." IEEE Signal Processing Magazine 04 (2004): 26-54. 6 Feb. 2007.

[18] "Intel Corporation's Wireless UWB Link 1480 Chipset." Advertisement. 6 Feb. 2007 <http://www.intel.com/network/connectivity/products/uwb/prodbrief.pdf>.

[19] "Staccato Communications' Ripcord SC3502P Chipset." Advertisement. 6 Feb. 2007 <http://www.staccatocommunications.com/pdfs/briefs/SC3502.pdf>.

[20] Vercimak, Luke, and Karl Weyeneth. Software Defined Radio. Dept of Electrical and Computer Engineering, Bradley University. 2006. 21 Nov. 2006 <http://cegt201.bradley.edu/projects/proj2006/sradio/archives/Final_Project_Report.pdf>.

[21] "Direct Quadrature Modulators." Hittite Microwave Corporation. 6 Feb. 2007 <http://www.hittite.com/>.

Appendix A – Derivation of OFDM

The complex envelope signal or the signal before up-conversion for an OFDM signal is represented by equation 1. In this equation, (N) represents total number of sub-carriers, while (n) represents a single sub-carrier within the spectrum.

$$g(t) = A_c \sum_{n=0}^{N-1} w_n \, \phi_n(t)$$

Equation 1

Shown in Figure 28 each sub-carrier location is defined by frequency (f_n). Note also that (f_n) is equal to equation 2.

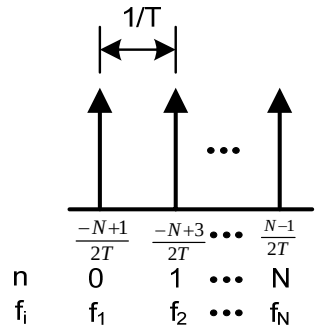


Figure 1 - OFDM Spectral Sub-carriers

$$f_n = \frac{1}{T}\left(n - \frac{N-1}{2}\right)$$

Equation 2

The locations of the sub-carriers which are defined in the complex envelope function as ϕ_n(t). This ϕ_n(t) is also equal to equation 3.

$$\phi_n(t) = e^{j2\pi f_n t}$$

Equation 3

Two functions are said to be orthogonal with respect to each other over the interval a<t<b if the following condition is met:

$$\int_a^b \phi_n(t)\phi_m^{\,*}(t)dt = 0$$

Equation 4

Using any two consecutive signals in the OFDM spectrum (shown in Figure 28), equation 4 becomes equation 5 by substituting ϕ_n(t) for each frequency, in this case n=i & j, where $|i-j|=1$.

$$\int_0^T e^{j2\pi f_i t} e^{-j2\pi f_j t} dt$$

Equation 5

Doing algebraic manipulations on equation 5 and evaluating the integral, equation 6 is obtained.

$$\int_0^T e^{j2\pi(f_i-f_j)t} dt = \frac{e^{j2\pi(f_i-f_j)T}-1}{j2\pi(f_i-f_j)} = \begin{cases} 1, & when \ \ f_i = f_j \\ 0, & when \ \ f_i \neq f_j \end{cases}$$

Equation 6

The result in equation 6 is obtained from Euler's Identity. When equation 6 is equal to zero then the two frequencies are orthogonally spaced. Therefore all frequencies defined within the complex envelope function shown in equation 1, will be orthogonally placed.

From Figure 28 f_i is approximately equal to $\frac{n}{T}$. Substituting into equation 1, equation 2 is obtained.

$$g(t) = A_c \sum_{n=0}^{N-1} w_n e^{j2\pi \frac{n}{T} t}$$

Equation 7

Substituting $t = \frac{kT}{N}$ into equation 7, equation 8 is obtained.

$$g(\frac{kT}{N}) = A_c \sum_{n=0}^{N-1} w_n e^{j2\pi \frac{n}{T}\frac{kT}{N}}$$

Equation 8

Finally by reducing equation 8, equation 9 is obtained.

$$g(k) = A_c \sum_{n=0}^{N-1} w_n e^{\frac{j2\pi nk}{N}}$$

Equation 9

Equation 9 is the definition of an inverse direct Fourier transform (IDFT). This IDFT can be implemented with an IFFT, which is done in our model to create the OFDM spectrum.

Appendix B – Initial Project Goals

When the project first took off, the initial goal was to research and purchase a full-scaled development kit from one of several vendors. This was before any in-depth research on UWB theory was conducted.

This kit was going to be used to evaluate the performance and versatility of the UWB MB-OFDM standard. To do this, several measurements were first going to be taken. Some of the measurements would include: Bit Error Rate (BER), spectrum measurements, data rate measurements, transmission range, power consumption, and antenna and RF characteristics.

The second goal to accomplish with a development kit would be to actually use the kit for a high-level project or goal. The UWB project team wanted to use the kit to transmit high speed video for a mobile platform – such as a radio controlled car, or perhaps an autonomous robot for another senior project. This would be beneficial because of the high frame rates that could be achieved – would aid in controlling a vehicle from a remote location.

There were five companies that were examined for available UWB platforms. The comparison of these companies and their products are shown in the Table 5. Of these companies, Staccato Communications had one of the more impressive development kits currently on the market. It included many interfaces that could be implemented in a variety of different ways. Wisair also had a board with many features that would have been useful for this project; however, the kits were very expensive and would lead to a "plug and play" evaluation board that would not be suitable for a senior project.

The next two companies were quite deceptive with their documentation and advertising. Focus Enhancements Semiconductor Group in particular had great documentation and features that could have been practical, but their product was still in development. Nevertheless, they are scheduled to release their product early to mid 2007, which is just barely out of our time frame. The same issue was also seen with Alereon. Their projected launch was also forecasted for release in mid 2007.

Finally, the last company left was PulsON Technologies. PulsON had been in production of their UWB evaluation kit for some time. They also had an education program setup specifically for universities level projects. This was promising, but on further investigation it was discovered that their modulation scheme was not what this project was focusing on. PulsON's P210 modulation scheme was based on direct sequence UWB (DS-UWB).

With all of the current evaluation boards out of reach, this project needed a new direction. It was decided that the direction of the project change from purchasing an evaluation kit to making a scaled down version of UWB system.

Table 1 – Development Kit Comparison

Company	Staccato Communications	Focus Enhancement	Wisair	Alereon	PuslON Time Domain
Product Name	Ripcord	Talaria	UWB Development Kit	WiMedia PHY Eval Board	e^2 Evaluation Kit
Product Part Number	SC3111D	TT-2101	DV9110	AL4401-EVK	P210
Contact Info	Martin Humphries		Sadir	Jim Meyer	Jon Hedges
Contact Info	(858) 812-1000	(503) 615-7700	(408) 399-7747	(512) 345- 4200	(256) 922-9229
Email	mhumphries@staccatocommunication.com		wisusa@wisair.com		jon.hedges@timedomain.com
Purchasing Issues	Too expensive	Still in development. Launch early ’07.	Too expensive	Still in development. Launch mid ’07.	Not desired modulation scheme
Total Cost	$25,000	N/A	$12,000	N/A	$7,500
Cost Per Board	$12,500	N/A	$6,000	N/A	$3,750
Lead Time	unknown	N/A	1 month	N/A	3 months
Frequency	3.1-4.8 GHz	3.2-7.2 GHz	3.1-4.8 GHz	3.1-5.0 GHz	3.1-6.3 GHz
Modulation Schemes	Multiband OFDM	Multiband OFDM	Multiband OFDM	Unknown	Direct Sequence
		Direct Sequence			
Interfaces:					
USB 2.0 Host	Yes	Yes	Yes	Unknown	Unknown
USB 2.0 Device	Yes	Yes	Yes	Unknown	Unknown
Direct CPU Interface	No	No	Yes	Unknown	Unknown
SDIO 1.1 Device	Yes	No	No	Unknown	Unknown
GPIO	Yes	No	No	Unknown	Unknown
Ethernet	No	Yes	Yes	Unknown	Unknown
Firewire	No	Yes	No	Unknown	Unknown
SPI to MPEG	No	No	Yes	Unknown	Unknown
Memory	Serial flash	SRAM/DRAM	Unknown	Unknown	Unknown
Internal	Unknown	1 MB SRAM	Unknown	Unknown	Unknown
External	N/A	DRAM Interface	Unknown	Unknown	Unknown
Security	128-bit AES Hardware Encryption	128-bit AES Hardware Encryption	No	Unknown	Unknown
Antenna	Unknown	Unknown	Yes / Omni	Unknown	Unknown
Documents	Yes	Yes	Yes	Unknown	Unknown

Appendix C – Simulink Models

Transmitter Models

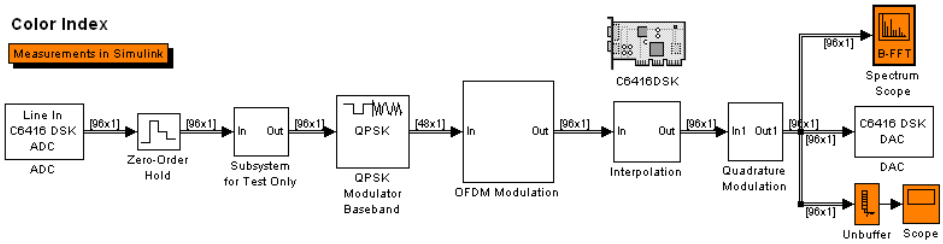


Figure C1 – Transmitter Model with 64 symbol IFFT

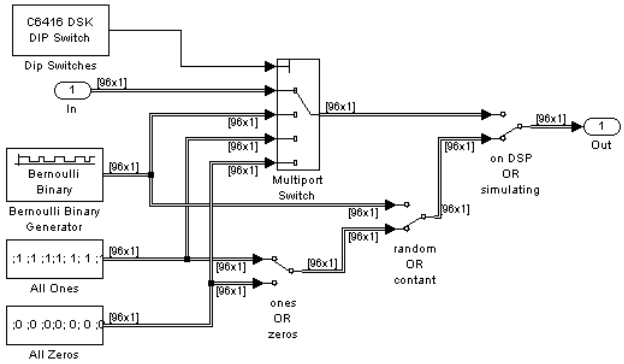


Figure C2 – Subsystem for Test Only

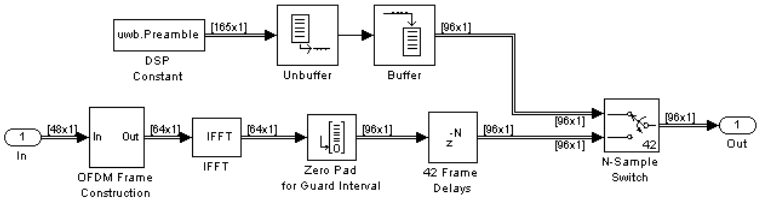


Figure C3 – OFDM Modulation

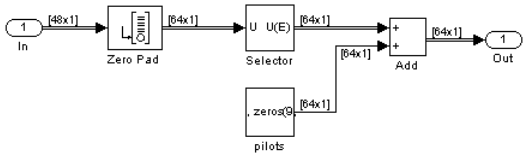


Figure C4 – OFDM Frame Construction

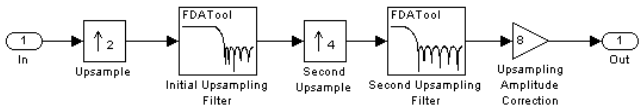


Figure C5 – Interpolation [20]

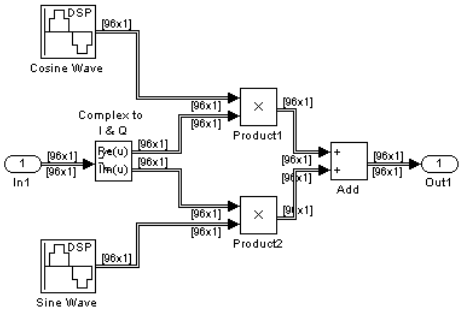


Figure C6 – Quadrature Modulator

Receiver Models

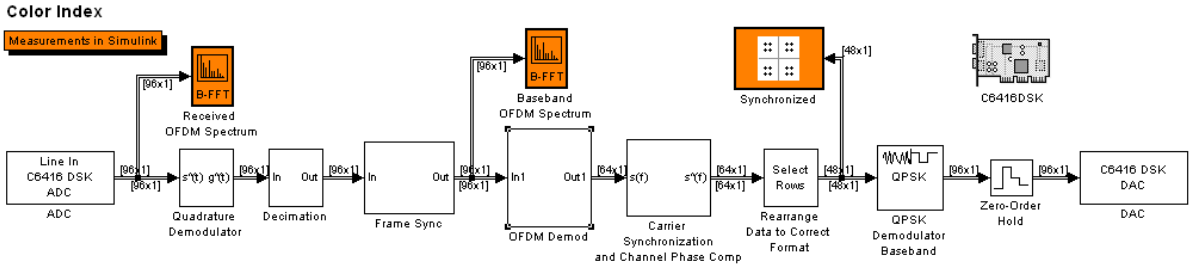


Figure C7 – Receiver Model with 64 symbol FFT

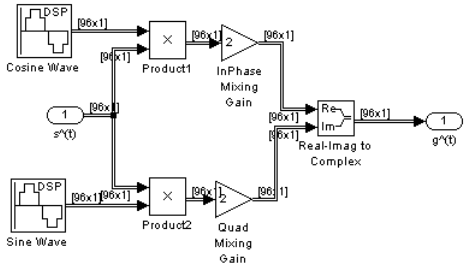


Figure C8 – Quadrature Demodulation [20]

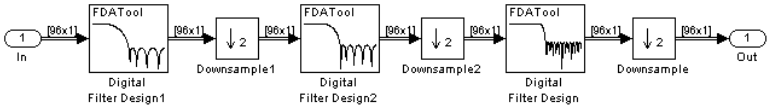


Figure C9 – Decimation [20]

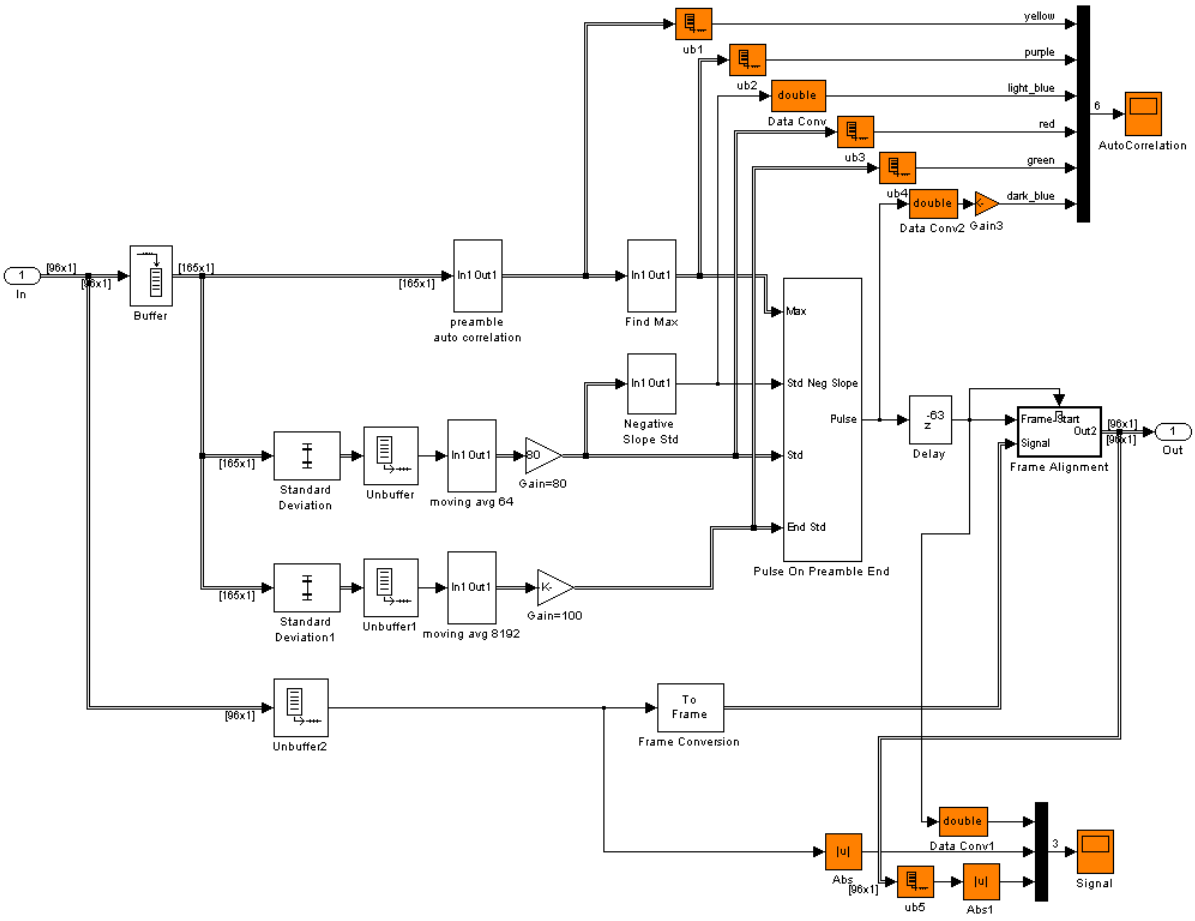


Figure C10 – Frame Synchronization

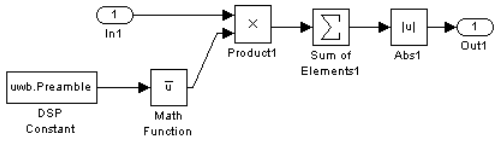


Figure C11 – Autocorrelation of Preamble

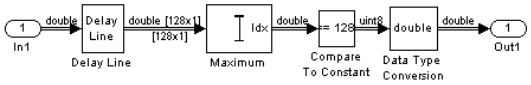


Figure C12 – Find Max

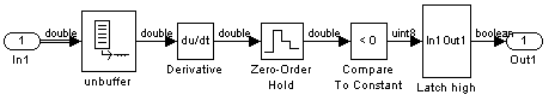


Figure C13 – Negative Slope

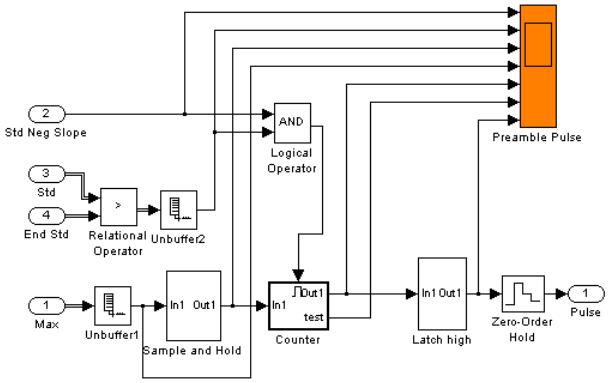


Figure C14 – Pulse on Preamble End

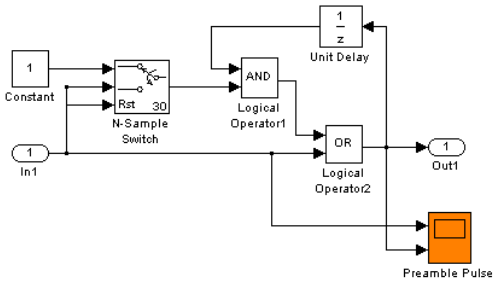


Figure C15 – Sample and Hold

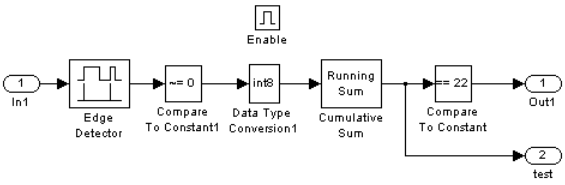


Figure 16 – Counter

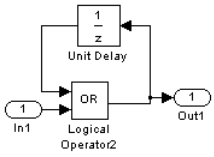


Figure C17 - Latch High

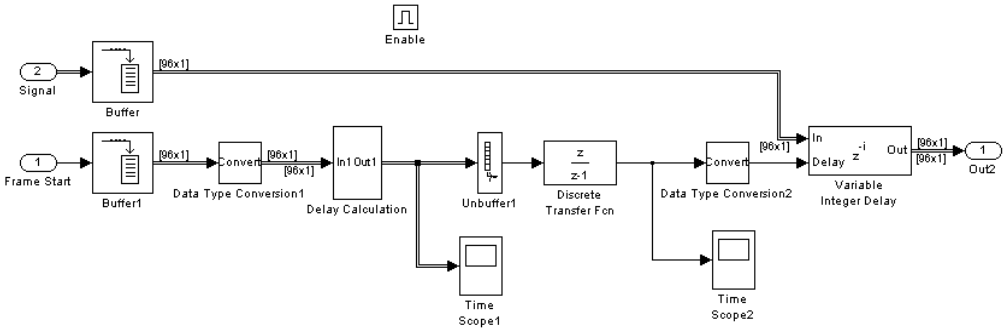


Figure C18 - Frame Alignment

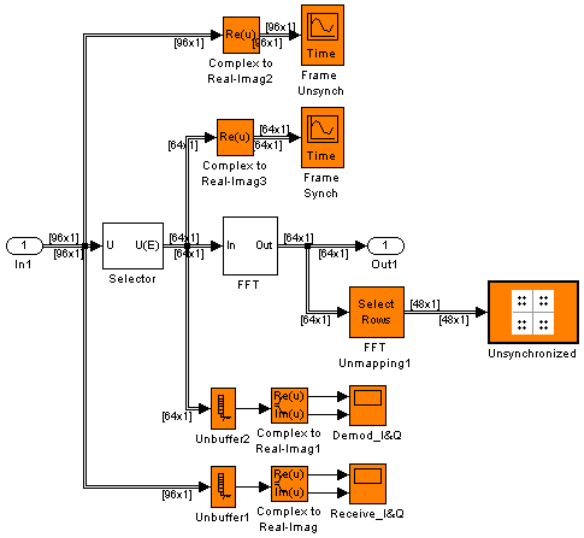


Figure C19 – OFDM Demodulation

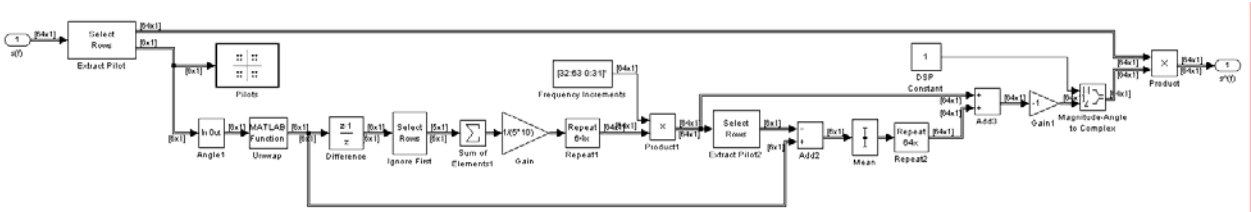


Figure C20 – Carrier Synchronization and Channel Phase Compensation [20]

Appendix D – Matlab .M Files

File Name = uwb_start.m

```
function uwb_start;

assignin('base','uwb',uwb_param_v4);
```

File Name = uwb_param_v4.m

function parameters = uwb_param

```
o.Fs = 96e3;
o.Ts = 1/96e3;

o.F_pre_mod = o.Fs/8; %The max this can be is o.Fs/2 from Nyquist
o.T_pre_mod = 1/o.F_pre_mod;

o.Data_Rate = o.F_pre_mod/96;  %96 is from the OFDM modulation
o.F_mod = o.Fs/4;   %This is the frequency at which the quadrature modulator centers the signal
at.

%Do not change
o.carrier_data = 32;
o.carrier_pilot = 4;
o.FFT_size = 64;

o.carrier_zero = 8;
o.zeropadOFDM = 16;

o.P5 = (-1+j)/sqrt(2);
o.P6 = (-1-j)/sqrt(2);

N_zeropad = 37; %= N_sym-N_fft = 165-128=37
o.Preamble = padarray(FramePacketSync,N_zeropad,0,'post');
o.Preamble_Len = 165;

parameters = o;

function m = FramePacketSync
n = [...
    0.9574
    0.527
    1.5929
    0.25
    0.2536
    0.3023
    1.2907
    0.4258
    1.12
    1.7704
    0.8593
    0.3719
    -1.3465
    0.7419
    1.535
    -1.28
    0.6955
    1.7204
    0.1643
    0.3347
    -1.7244
    0.7447
```

1.1141
-1.3541
0.7293
0.2682
-1.2401
1.527
0.1199
1.1496
-1.544
1.3176
0.84
1.398
1.1147
0.4732
-1.7178
0.8477
1.5083
-1.4364
0.3853
1.5673
0.295
0.4204
-1.4856
0.8404
1.111
-1.4269
0.3033
0.7757
0.137
0.525
-1.1589
0.8324
0.6336
-1.2698
0.7853
0.7031
-1.1106
0.6071
0.7164
0.8305
-1.2355
1.1754
0.5859
0.3053
0.8948
0.6744
0.8901
0.8133
0.9201
-1.841
0.8036
0.3105
-1.514
0.7644
0.7301
0.9788
-1.1305
1.3257
0.7801
0.7867
1.996
0.5623
-1.2227
0.8223
1.2074
-1.2338
0.2957
1.999
0.201
0.586

```
    -1.2284
    0.9215
    0.7941
    -1.4128
    0.8528
    0.6973
    -1.2477
    0.6246
    0.7687
    0.7966
    -1.2809
    1.1023
    0.425
    0.1614
    0.7547
    0.6696
    0.392
    0.7589
    0.6701
    0.9381
    0.7483
    0.9659
    0.9192
    0.3925
    1.2864
    0.6784
    -1.909
    1.114
    0.6134
    -1.5467
    0.3031
    0.9457
    1.9645
    1.4549
    -1.276
    2.2102];
m=complex(n,1e-30);
```

Appendix E – Simulink Tutorial

The purpose of this tutorial is to provide the knowledge about Simulink upfront that is often learned the hard way. Hopefully this knowledge will allow future students to progress further and faster than this project did with this extra knowledge.

General Tips

1. Terminate any unused port.

   Often times in Simulink, there are blocks being thrown down so quickly that it's hard to keep track of them. An issue that was encountered involved a bizarre unrelated error that could easily have been fixed by terminating a port. The error was "Too many periodic sample rates"; "Embedded target does not support more than 7 periodic sample rates in multitasking mode." This error can also occur if there are too many periodic sample times. However, if there are not more than 7 periodic sample times then make sure all of the ports that are unused are terminated with the termination block (Simulink > Sinks > Terminator).

2. Set a majority of sampling times. Don't rely on inherent sampling times (-1) to always work.

   Specifically be sure and set any block that is inputting information into the Simulink model. Simulink does not automatically know what you want and will often do undesired actions if left to figure its own sampling times out.

3. Use a multiplexing block to view more than one signal on the same scope simultaneously in Simulink (Simulink > Signal Routing > Mux).

   This is very handy technique for comparing signals. This technique works great when trying to work out synchronization timing for the receiver model.

4. Work in a file path without any spaces.

   Code composer studio (CCS) will complain if there are any spaces anywhere in the file path of the model or of CCS file paths. Thus, CCS cannot be installed in nor can the models be in a file path of C:\Program Files\etc for example.

5. When using a DSP sine wave source block, never use the default computational setting of Trigonometric Function.

   This setting will significantly slow down the board. When this option was used with the default setting the complier would run perfectly but nothing would actually come out of the board. This was because the trigonometric computations were overloading the processor. In most cases a simple lookup table will work fine.

Don't Understand what Exactly a Block is Doing?

The best way to solve this problem is to follow these steps:

1. Read the Help files extensively.

   Mathworks Simulink Help files are generally extremely useful. However, they are often cryptic, but can provide guidance into how various blocks work.

2. Experiment with the block in a smaller separate model.

   Experimenting with blocks is the best way to understand exactly how the more complex blocks work. Often times it is difficult to isolate a block, in that case try copying a portion of the model (only what I necessary for the block to work) to another simulation and experimenting with its operation.

3. Talk to Dr. In Soo Ahn if all else fails.

   Dr. Ahn is a great resource for information. One important piece of advice I will give, is to make sure you know exactly what it is you are asking. No offense intended for Dr. Ahn, but if you go in with a general question you will get a general answer that is often not what is needed.

Finding a List of the Current Sampling Times in a Model

There could be a more direct route to finding the sampling times within a model, but this is the route that was stumbled upon.

1. Open the Simulation Configuration Parameters (Ctrl+E).
2. Make sure you are in the Solver menu from the list of menus on the left.
3. The Type should be Fixed step; The Solver should be discrete (no continuous states).
   a. Set the Periodic Sample Time Constraint to Specified.
   b. Enter one known sample time in the Sample Time Properties in the format:
   c. [$1/F_{SAMPLE\_TIME}$, offset, priority] Note: offset is often set to zero, the higher the priority value the higher the task priority.
4. Run the simulation.
5. The error that comes up show list all of the sample periods which can be used to figure out the sample frequencies.
6. Just switch back the Period Sample Time Constraint to Unspecified once done.

Variables with Structures and Call Back Functions in Matlab

CallBack Functions and Variables are very useful in the Simulink environment. It allows you to define one variable that can be used in any block definitions. For example, if you wanted to vary sample times each block would have to be changed individual, whereas a variable would only have to be changed once.

Callback functions are essentially functions that are called depending on which Callback function is used. For example, the StartFcn will execute whatever .m file is setup for the StartFcn Callback before the simulation starts. Search "Using Callback Functions" in the Help Files for a complete list and more detailed explanation.

Setting a Callback Function

1. In the highest level of the model go to the File drop down menu and select Model Properties.
2. Click on the Callbacks tab. This tab will show each call back and which functions are set for each callback. Each callback listed executes a .m file at a different time.
3. Click on PreLoadFcn and type the name of the .m file without the suffix. For example all of the models in the project use a file named uwb_start.m that resides in the working directory of the model. In this setting, uwb_start shows up under PreLoadFcn so every time the model is opened uwb_start.m is run.
4. Or to perform the same action type the following in the Matlab command window:

   set_param('model_name', 'PreLoadFcn', 'uwb_start')
5. Within uwb_start.m a list of variables can be put, but often it is difficult see all of the variables when there is a large number of variables. In this case a structure can be used.

Structure of Variables (Data Object Class)

The structure of a data object class is Package.Class, where the package is the structure to which the class or variable belongs to.

1. Within the uwb_start.m file the assignin function must be used. This function allocates variables to the workspace. The function works like the following:

   assignin('*type of workspace*', '*name of structure*', *.m file with parameters listed*)
2. An example of this from the models for this project is:

   assignin('base', 'uwb', uwb_param);
3. Then within the uwb_param.m file the parameters are listed in the following way.

```
function param_1234 = uwb_param

s.variable_x = 1302.2;
s.beta = 3.2;
```

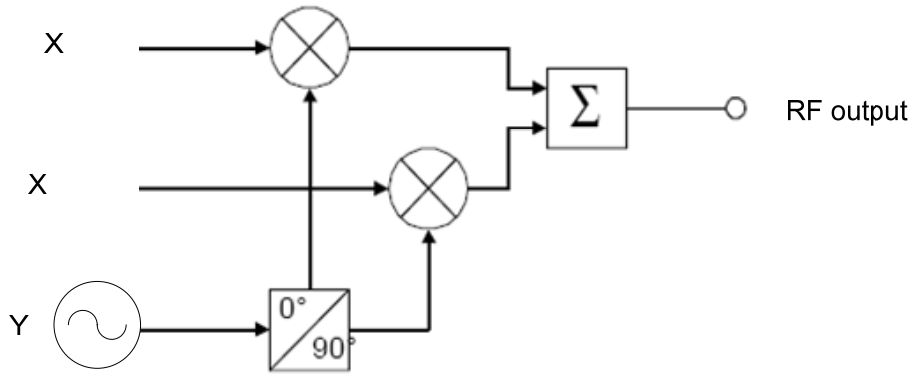
```
s.time_freq = 102;

param_1234 = s;
```

4. When a parameter is updated in the uwb_param.m file, save the file and then type “uwb_start” in the Matlab command window to reassign the variables. If desired the uwb_start can be updated every time the simulation is run, just update the callback functions.

Appendix F – Single Sideband Theory

Proof of Quadrature Modulation



Assume the I&Q signals are the same signal (represented by X), except they are 90 degrees out of phase. The LO signal is designated Y.

Then the RF output signal is:

$$[\cos(x)\cos(y)] + [\sin(x)\sin(y)] = \left[\frac{1}{2}\cos(\mathrm{x}+\mathrm{y}) + \frac{1}{2}\cos(\mathrm{x}-\mathrm{y})\right] + [\frac{1}{2}\cos(\mathrm{x}-\mathrm{y}) - \frac{1}{2}\cos(\mathrm{x}+\mathrm{y})]$$

(Trig. Identity A)

The right hand side can be written as:

$$\frac{1}{2}[\cos(x)\cos(y) - \sin(x)\sin(y)] + \frac{1}{2}[\cos(x)\cos(y) + \sin(x)\sin(y)]$$
$$+ \frac{1}{2}[\cos(x)\cos(y) + \sin(x)\sin(y)] - \frac{1}{2}[\cos(x)\cos(y) - \sin(x)\sin(y)]$$

(Trig Identities B and C)

This can be simplified as:

$$\cos(x)\cos(y) + \sin(x)\sin(y)$$

Using Trig Identity A again, this becomes:

$$\frac{1}{2}[\cos(x-y) + \cos(x+y) + \cos(x+y) - cos(x+y)]$$

which is finally:

$$\cos\,(x-y)$$ (Lower Sideband)

Trig Identity A:

$$\boldsymbol{cos}(\boldsymbol{x})\boldsymbol{cos}(\boldsymbol{y}) = \frac{\mathbf{1}}{\mathbf{2}}[\boldsymbol{cos}(\boldsymbol{x}-\boldsymbol{y}) + \boldsymbol{cos}(\boldsymbol{x}+\boldsymbol{y})]$$

Trig Identity B:

$$\boldsymbol{sin}(\boldsymbol{x})\boldsymbol{sin}(\boldsymbol{y}) = \frac{\mathbf{1}}{\mathbf{2}}[\boldsymbol{cos}(\boldsymbol{x}-\boldsymbol{y}) - \boldsymbol{cos}(\boldsymbol{x}+\boldsymbol{y})]$$

Trig Identity C:

$$\boldsymbol{cos}(\boldsymbol{x} \pm \boldsymbol{y}) = \boldsymbol{cos}(\boldsymbol{x})\boldsymbol{cos}(\boldsymbol{y}) \mp \boldsymbol{sin}(\boldsymbol{x})\boldsymbol{sin}(\boldsymbol{y})$$