EE209AS Project: Investigation on ”Design Transceiver for IEEE 802.15.4 using ZigBee Technology and Matlab/Simulink”

In Hwan Baek
Henri Samueli School of Enginnerig
and Applied Science
University of California Los Angeles
Email: chris.inhwan.baek@gmail.com

Xinhua Wang
Henri Samueli School of Enginnerig
and Applied Science
University of California Los Angeles
Email: wangxinhuadmiral@hotmail.com

***Abstract*— ZigBee technology has been developed for many years, and it is based on IEEE 802.15.4. It is usually used as a wireless personal area networks (PAN) that aimed at control or monitor application due to its low data rate and low power consumption. This project is mainly focus on implement a transceiver by using MatlabSimulink. In this project, Minimum Shift Keying (MSK) demodulation technique is described and analyzed. The bandwidth efficiency of MSK is 2 bit/s/Hz, and the result shows that OQPSK modulation with half sine pulse shaping is perfectly employed in Zigbee technology.**

I. Introduction

A. Subsection Heading Here

ZigBee standard is developed by ZigBee Alliance, which has hundreds of member companies, from the semi-conductor industry and software developers to original equipment manufacturers and installers. The ZigBee alliance was formed in 2002 as a nonprofit organization open to everyone who wanted to join. The ZigBee standard has adopted IEEE 802.15.4 as its Physical Layer (PHY) and Medium Access Control (MAC) protocols. Therefore, a ZigBee compliant device is compliant with the IEEE 802.15.4 standard as well.

ZigBee is a low-cost, low-power, wireless mesh networking standard. First, the low cost allows the technology to be widely deployed in wireless control and monitoring applications. Second, the low power-usage allows longer life with smaller batteries. Third, the mesh networking provides high reliability and more extensive range. ZigBee is a standard that defines a set of communication protocols for low data rate short range wireless networking. ZigBee based wireless devices operate in 868MHz, 915MHz and 2.4GHz frequency bands. It is targeted mainly for battery power applications where low data rate, low cost and long battery life are main requirements. In many ZigBee applications, the total time the wireless device is engaged in any type of activity is very limited. The device spends most of its time in power saving mode, also known as sleep mode.

ZigBee standard is specifically developed to address the need for very low cost implementation of low data rate wireless networks with ultra low power consumption. The ZigBee Standard reduced the implementation cost by simplifying the communication protocols and reducing the data rate. The minimum requirements to meet ZigBee and IEEE 802.15.4 specifications are relatively relaxed compared to other standards such as IEEE 802.11, which reduces the complexity and cost of implementing ZigBee compliant transceivers.

II. Background

According to the specifications of ZigBee technology, the modulation is done with Offset Quadrature Phase Shift Keying (OQPSK) with Half sine pulse shaping. Thus, it is necessary to review the OQPSK sheme and the purpose of half sine pulse shaping

A. Offset Quadrature Phase Shift Keying (OQPSK)

OQPSK is a variant of Quadrature Phase Shift keying (QPSK) modulation sheme. The main difference between QPSK and OQPSK is that in OQPSK, after splitting the bit stream into an odd bit stream and an even bit stream, one of them is shifted by a half bit period. The basis functions of OQPSK are the following:

$$\Phi_1(t) = \sqrt{2\tfrac{2}{T}}\cos 2\pi f_c t \quad 0 \le t \ge T$$
$$\Phi_2(t) = \sqrt{2\tfrac{2}{T}}\sin 2\pi f_c t \quad \tfrac{T}{2} \le t \le \tfrac{3T}{2}$$

B. Half Sine Pulse Shaping

Unlike QPSK, OQPSK can have abrupt phase changes. This may result in spectral components in high frequency and low pass filtering to suppress the side lobes can lead to inter symbol interference. Instead, half sine pulse shaping is applied to overcome such difficulty.

In OQPSK, the signal has a constant amplitude. However, abrupt phase change of 90°can occur. Thus, the waveform is filtered to suppress the sidebands. The problem is that the filter will change the amplitude. To avoid this issue, nonlinear power output stages are employed in the transmitter. The nonlinearity will result in spectral components outside the main lobe. Half sine pulse shaping will help overcome this problem.

III. Transceiver Design

A. Transmitter

This section shows the design of ZigBee transmitter system. The implementation was done on Matlab/Simulink using fundamental components in Simulink to demonstrate how reliably complex modulation schemes can be built, cost effectively and efficiently. The design of ZigBee transmitter using OQPSK modulation with half sine pulse shaping is shown in the Figure 1 given below. Here, the input bit stream is having a data rate of 250Kbps.

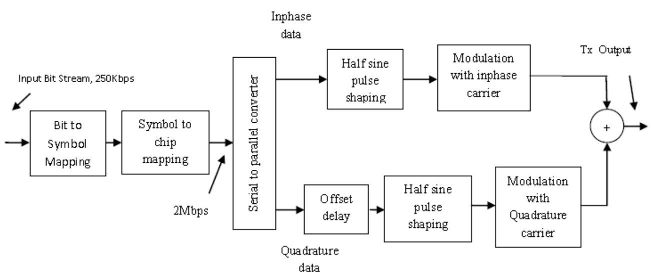

Fig. 1. Block diagram of ZigBee Transmitter

B. Channel

We used 10db AWGN channel in this project.

C. Receiver

This section shows the design of ZigBee receiver system. Here, we are concentrating on the MSK coherent detection technique for recovering original data in receiver. The block diagram of the ZigBee Receiver is shown in Figure 2 below. The step-by-step procedure to implement the ZigBee receiver using Simulink is presented in section IV.

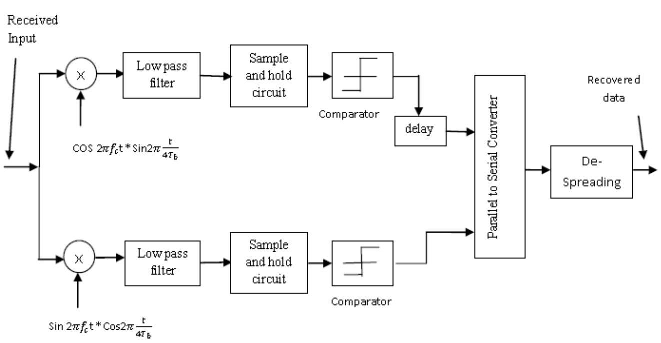

Fig. 2. Design of ZigBee Receiver

1) Coherent Detection vs. Noncoherent Detection: Due to propagation delay and fading in the communication channel, the transmitted signal can undergo phase changes when traveling through it. This means that if we use a base function with a given initial phase at the transmitter, the receiver must be fed by a base-function with a phase that reflects any phase change caused by the channel. Any modulation scheme that makes use of this synchronization is said to be coherently detected.

Coherent detection is performed at a cost, that is, the receiver must be equipped with a carrier recovery circuitry, which increases system complexity, and can increase size and power consumption. Additionally, there is no ideal carrier recovery circuit. Then, strictly speaking, no practical digital communication system works under perfect phase coherence. Non-coherent detected modulations do not need carrier synchronization. So there is no need of carrier recovery circuit in case of non-coherent detection.

In this project, we design a coherent detection that can be used in real life. The result shown in section V will prove that we did a great job.

IV. Implementation on Matlab/Simulink

We implemented the ZigBee transceiver on Matlab/Simulink following the suggestions of the paper. Figure 3 is the Simulink model of the transceiver. The implementation of the subsystems, transmitter and receiver, are explained in subsection IV-A and subsection IV-B.

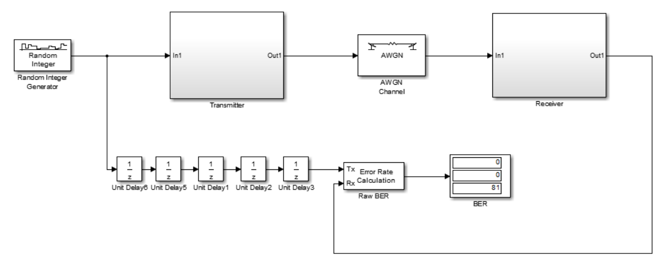

Fig. 3. ZigBee Transceiver

A. Transmitter

Figure 4 shows the blocks used for the ZigBee transmitter implementation.

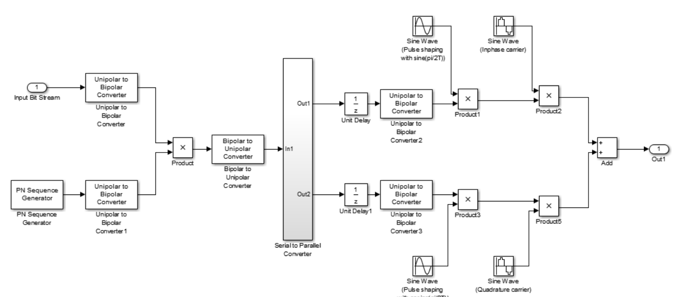

Fig. 4. ZigBee Transmitter

1) Bit to Chip Mapping: Bit to Chip mapping is achieved by generating direct spread spectrum signal (DSSS). We can get DSSS by multiplying a bit stream with a pseudo noise sequence. We used a Random Integer Generator block to generate a bit stream and a PN Sequence Generator to generate a pseudo noise sequence. Before multiplying them, we feed them into Unipolar to Bipolar Converter block to convert them into NRZ forms. Then, the output of the multiplier is converted back using Bipolar to Unipolar Converter block.

2) Serial to Parallel Conversion: Serial to parallel converter is implemented with one J-K flip flop and two D flip flop as shown in Figure 5. A clock with 0.5μs period drives the J-K

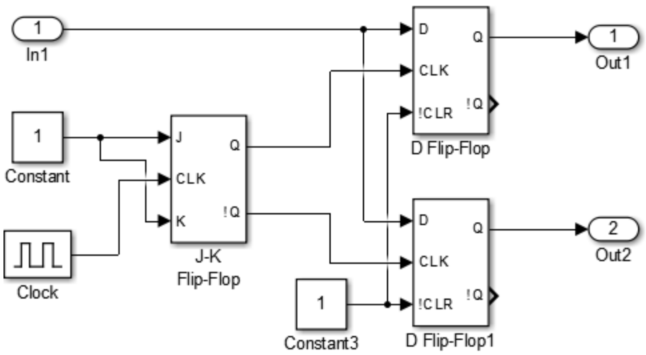

Fig. 5. Serial to Parallel Converter

flip flop, which generates an even clock and an odd clock with 1μs periods. The even clock and the odd clock drives the D flip flops, which generate an Inphase signal and a Quadrature signal respectively. Since the odd clock is 0.5μs behind the even clock, we introduce a delay on the Quadrature signal. These clocks are shown in Figure 6, Figure 7, and Figure 8. As we can see in Figure 7 and Figure 8, there is a problem that the even and odd clocks are not in sync with the signal. Therefor, we added Unit Delay blocks at the output of the serial to parallel converter for adjustment. The Inphase signal is delayed by 0.5μs and the Quadrature signal is delayed by 0.25μs. And such delay on the Quadrature signal is added with the delay from the serial to parallel converter, so the Quadrature signal is 0.25μs in total behind the Inphase signal.

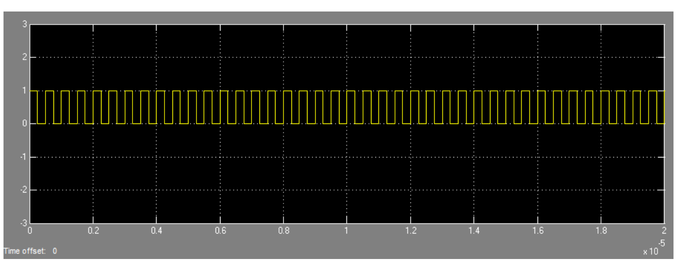

Fig. 6. Clock for J-K Flip Flop

3) Half Sine Pulse Shaping: Half sine pulse shaping is implemented with a Sine Wave block and a multiplier. The parameters for the sine wave block are the followings:
Amplitude: 1
Bias: 0
Frequency: 2*pi*500e3
Phase: 0 (on the Inphase signal arm) and pi/2 (on the Quadrature signal arm)
Sample time: 0
The sine/cosine waves generated by the sine wave blocks are

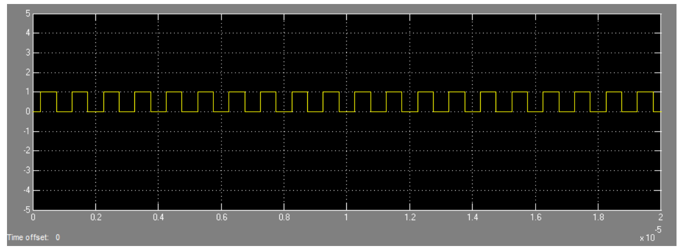

Fig. 7. Even Clock

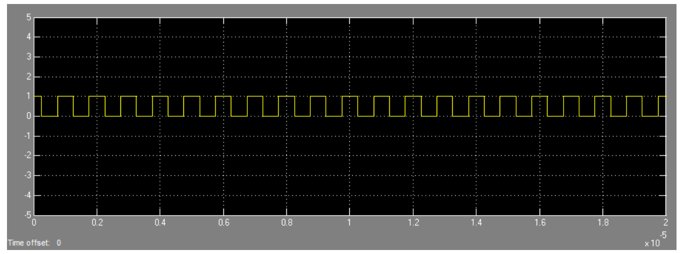

Fig. 8. Odd Clock

then multiplied with the Inphase signal and the Quadrature signal.

4) Modulation: We use 2.4GHz carriers to modulate the signals. These carriers are also generated by Sine Wave blocks. The parameters for the blocks are the followings:
Amplitude: 1
Bias: 0
Frequency: 2*pi*2400e6
Phase: 0 (on the Inphase signal arm) and pi/2 (on the Quadrature signal arm)
Sample time: 5/24e10
The carriers are multiplied with the half pulse shaped signals. Then, these signals are added and transmitted to a channel.

B. Receiver

Figure 9 shows the blocks used for the ZigBee transmitter implementation.

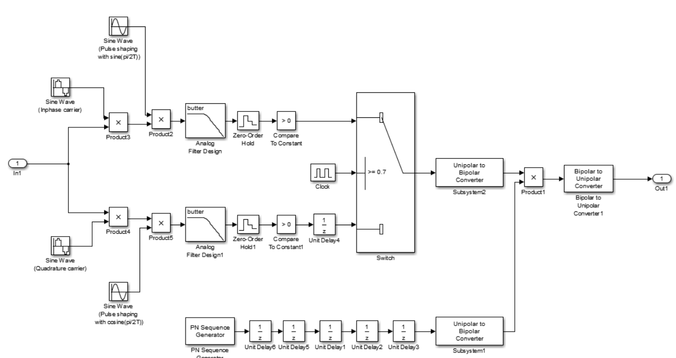

Fig. 9. ZigBee Receiver

1) Carrier Multiplication: The received signal is fed into ”Inphase arm” and ”Quadrature arm”. Each arm has a series of blocks to reconstruct the Inphase signal and the Quadrature signal. The first step is multiplying with the 2.4GHz carrier. On the Inphase arm, the signal is multiplied with the sine carrier while the signal is multiplied with the cosine carrier. The parameters of the carriers are the same as those we used in the transmitter.

2) Half Sine Wave Multiplication: The signals multiplied with the carriers are then multiplied with half sine pulse generated by Sine Wave blocks. The parameters are the same as those we used in the transmitter.

3) Low Pass Filter: We used Analog Filter Design block to implement a low pass filter. A third order Butterworth low pass filter is chosen. 500KHz is used as the passband edge frequency.

4) Sampling: The signal after the low pass filter on each arm is then sampled for every 0.5μs. We used a Zero-Order Hold block to implement this. The continuous signals are now discrete.

5) Thresholding: We used a Compare to Constant block to threshold the signal so that the output is only either 0 or 1. We set the parameters so that it will compare the signal with 0 and if it is greater than 0, it sets the output as 1 and 0 otherwise. The Inphase signal is 0.25μs behind the Quadrature signal due to the intended delays added to the signals in the transmitter. Therefore, the Quadrature signal is delayed by 0.5μs in order to sync with the Inphase signal before parallel to serial conversion.

6) Parallel to Serial Conversion: We used a Switch block to multiplex the Inphase and Quadrature signals. We used a clock with 1μs to select from the incoming signals. By adjusting parameters, we can select the Inphase signal when the clock is high and the Quadrature signal when the clock is low. The generated signal is direct spread spectrum signal.

7) Despreading: The direct spread spectrum signal is now needed to be despread to recover the bit stream. This can be achieved by multiplying the DSSS with the same pseudo noise sequence used in the transmitter. We copied the PN Sequence block from the transmitter subsystem and pasted into the receiver subsystem. Since there is a delay in the received signal, we also need to delay the PN sequence. A series of Unit Delay blocks are used for this purpose. The output of the multiplier is the recovered bit stream.

V. Simulation Results

In order to clearly show the plots, we ran the simulation for 40μs. However, we used longer time to evaluate the accuracy of the ZigBee transceiver.

A. Transmitter

A random integer generated the input stream. This is presented in Figure 10. It has a data rate of 250Kbps. i.e., it generates a bit for every 4μs as shown in the time axis. Each division for the time axis is taken as 10μs.

The 250Kbps input data is mapped into a symbol, making the symbol rate 62.5Kilo symbols per second. The Pseudo Noise code is generated from a PN sequence generator. This is presented in Figure 9 and it has a data rate of 2Mbps. That

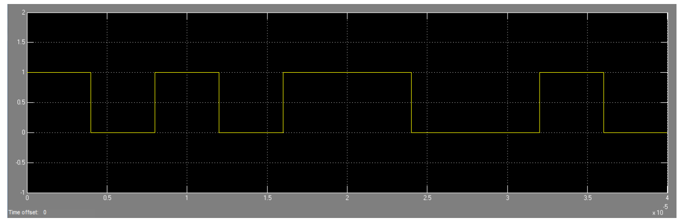

Fig. 10. Input Bit Stream

means each bit in the PN sequence is having a time period of 0.5μs. Clearly the chip rate is equal to eight times the bit rate or it is equal to 32 times of symbol rate.'

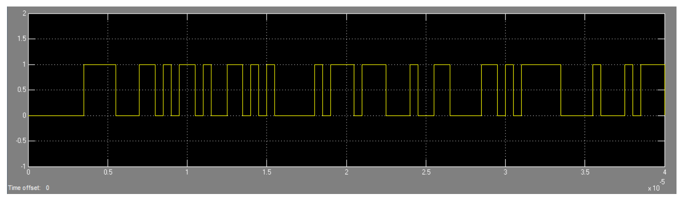

Fig. 11. Pseudo Noise Sequence

Direct Sequence Spread signal is generated by converting the input bit stream and PN bit sequence into NRZ form and multiply the resultant data. This is shown in Figure 12.

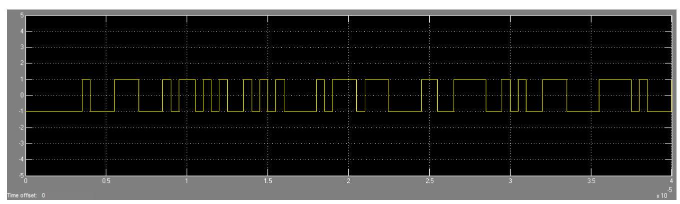

Fig. 12. Direct Spread Spectrum Signal

The DSSS signal has a data rate of 2Mbps. The Inphase data and Quadrature data after serial to parallel conversion are shown in Figure 13 and Figure 14. They are having a data rate of 1Mbps. By doing this, we can transmit two bits at a time and hence bandwidth efficiency improved.

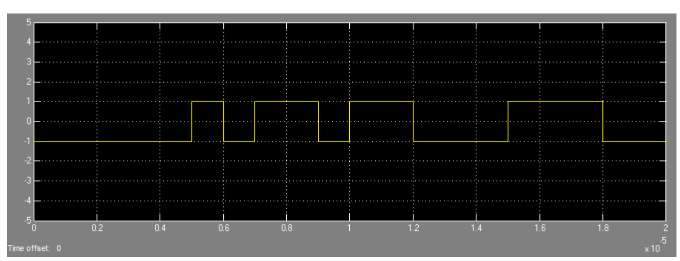

Fig. 13. Inphase Signal

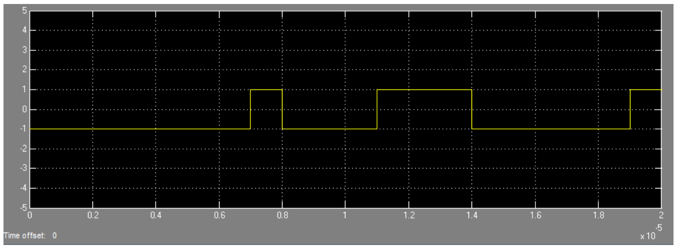

Fig. 14. Quadrature Signal

Following the separation of two bit streams, the two signals are pulse shaped through a half sine wave. Here this sine wave has a time period of 2μs. This Inphase signal after half sine pulse shaping is shown in Figure 15.

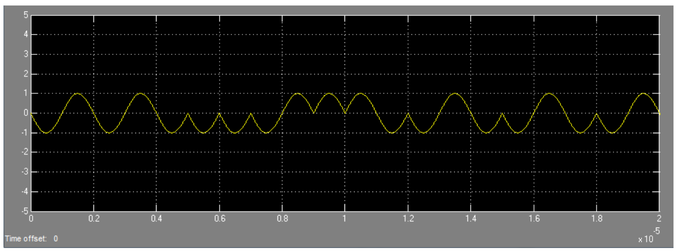

Fig. 15. Inphase Signal After Half Sine Pulse Shaping

Following the pulse shaping, the two signals are multiplied with a 2.4GHz high frequency carrier signal for doing modulation. The inphase signal and Quadrature signal after modulation are shown in Figure 16 and Figure 17.

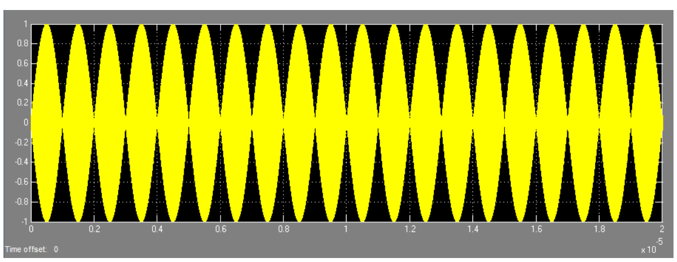

Fig. 16. Inphase Signal After 2.4GHz Carrier Modulation

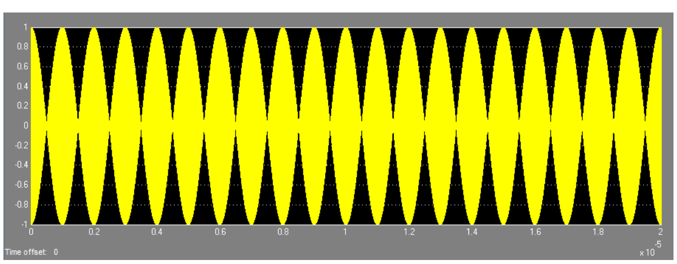

Fig. 17. Quadrature Signal After 2.4GHz Carrier Modulation

B. Receiver

The transmitted signal is passed through a AWGN channel. The noisy version of the transmitted signal at the input of receiver is shown in Figure 18.

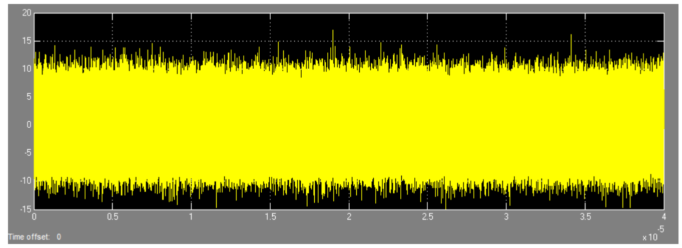

Fig. 18. Received Signal

The outputs of both inphase and Quadrature signals after multiplying with recovered carrier is shown in Figure 19 and Figure 20 respectively.

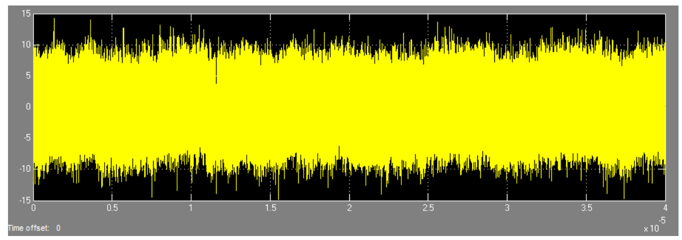

Fig. 19. Inphase Signal Multiplied with Carrier

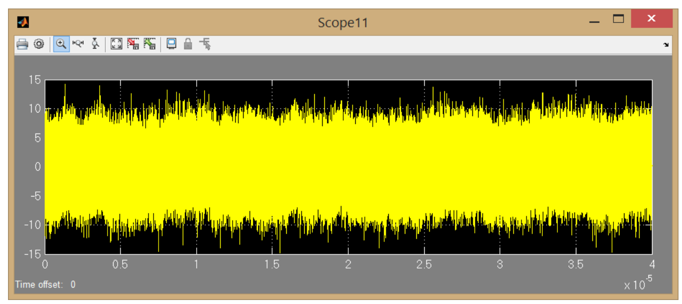

Fig. 20. Quadrature Signal Multiplied with Carrier

The outputs of inphase signals after multiplying with half sine signal are shown in Figure 21.

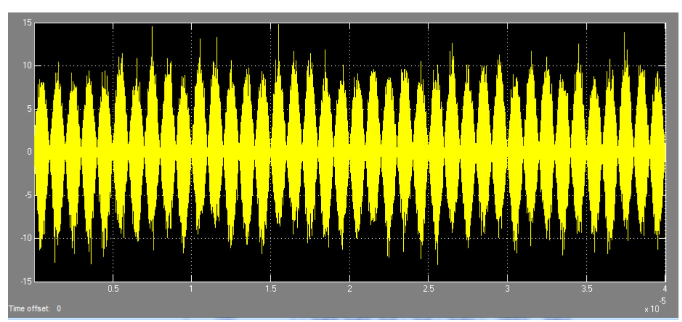

Fig. 21. Inphase Signal Multiplied with Half Sine Pulse

The resultant signals containing both the high frequency harmonics and baseband signal components. Separate the baseband signal components form high frequency harmonics by passing through a low pass filter. These signals are passed through a 3rd order Butterworth low pass filter having cutoff frequency of 500 KHz for extracting only baseband data. Outputs after low pass filtering are shown in Figure 22.

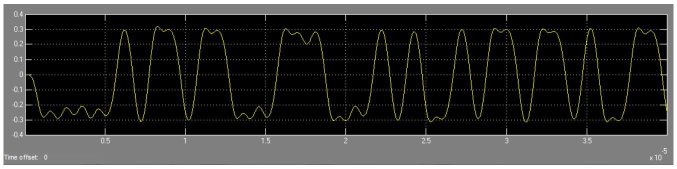

Fig. 22. Inphase Signal After Low Pass Filtering

Next, this baseband output is passed through a sample and hold circuit for sampling of base band signal. The sample period used for sampling is 2μs. This sampled output at both inphase demodulators are shown in Figure 23.

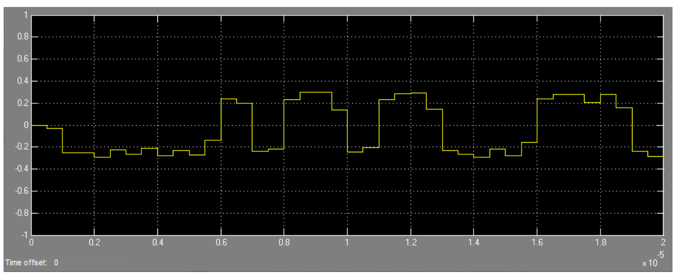

Fig. 23. Inphase Signal After Zero-order Hold

This sampled data is passed through a comparator, for deciding whether the transmitted bit is ”1” or ”0”. Set the threshold for comparator as ”0”. The comparator output at inphase demodulators are shown in Figure 24.

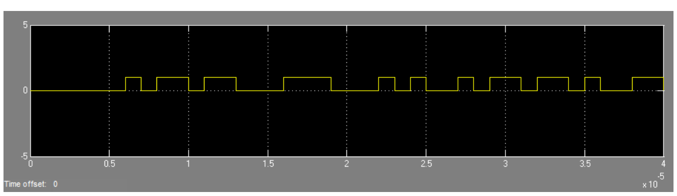

Fig. 24. Inphase Signal After Compare

This resulting inphase data is multiplexed using a multiplexer or a parallel to serial to converter and the output after passing through a parallel to serial converter is given in Figure 25.

The output bit stream shown in Figure 26 and the input bit stream shown in Figure 1 are same except a small amount of delay (1.5μs).

The BER is shown in Figure 27, and we can see that a perfect detection has been accomplished. The BER is 0 because the coherent direct detection.

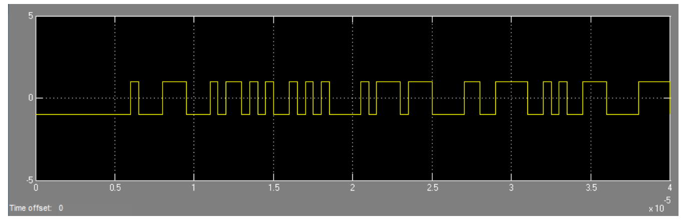

Fig. 25. Reconstructed Direct Spread Spectrum Signal

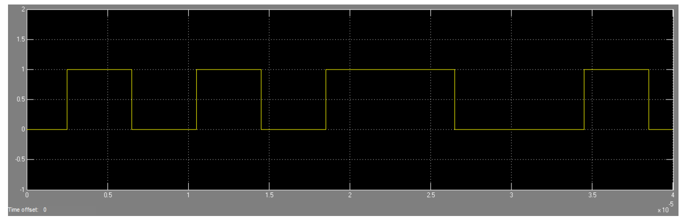

Fig. 26. Recovered Bit Stream After Despreading

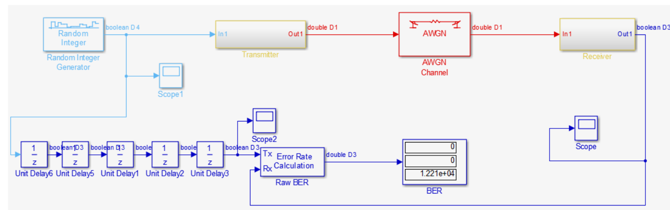

Fig. 27. BER

VI. Conclusion

In this project, the work that we present here helps to implement a transceiver for ZigBee wireless communication system using Matlab/Simulink. Without using mathematically complex blocks, we designed and tested a ZigBee wireless transceiver in Matlab/Simulink. A model for MSK has been presented, an analysis of which shows that the theoretical maximum bandwidth efficiency of MSK is 2 bits/s/Hz, Here, we are indirectly implementing Minimum Shift Keying modulation and demodulation (OQPSK with half sine pulse shaping). As discussed previously, Half sine pulse shaping avoids the abrupt phase shifts in the transmitted signal so that it reduced lot of burden and the modulated signal is amplifier friendly in real time scenario. The use of direct spread spectrum technique reduces the interference effects. The bandwidth of Zigbee is 2MHz.

Acknowledgment

The authors would like to thank Professor Cabric and Mihir Laghate.

References

[1] Byunghun Song, Haksoo Choi, Hyun Su Lee *Surveillance Tracking System Using Passive Infrared Motion Sensors in Wireless Sensor Network*, International Conference on Information Networking, pp.1-5, Jan 2008.