An FPGA based Implementation of Baseband and Passband Modulation for Wireless Transmitters

Saad Zafar, Numair Zulfiqar
College of Electrical and Mechanical Engineering
National University of Science and Technology
Pakistan
saadzafar31@ce.ceme.edu.pk, raonumair2010@yahoo.com

***ABSTRACT*: *There is a plethora of silicon options available for implementing the various functions of a transmitter, but FPGAs are an attractive alternative for many of these tasks due to performance, power consumption and flexibility. This paper presents a complete hardware architecture for transmitter which is then synthesized and mapped on the Spartan 3E FPGA kit. The transmitter carries out data formatting, NRZ line encoding, baseband modulation, pulse shaping and passband modulation. A ROM based raised cosine filter is designed to mitigate the effects of Inter-symbol Interference. Passband modulation is accomplished by designing hardware using Direct Digital Synthesizer. A dedicated module to introduce channel imperfections is also designed to study the behavior of transmitter on practical interfaces. The implemented sub-system is particularly suited for use in a typical WiMAX environment or any other SDR based system that performs downstream digital signal processing.*

Keywords: Digital Design, FPGA, Modulation, SDR, Transmitter

Received: 17 October 2013, Revised 21 November 2013, Accepted 26 November 2013

1. Introduction

The history of software defined radios dates back to 1984 when it was first used by a team at Garland Texas division of Raytheon [1]. A software defined radio is a system in which blocks of a general communication system are present already and they can be configured according to the requirements. These blocks include multipliers, filters, carrier generators, array processors and mixers etc. So instead of first making these blocks through software as in embedded systems they are already present on an SDR so it is just a matter of configuring them. Typical applications of SDR’s involve military and phone applications where a very basic requirement is of flexibility. Now due to advantages that are offered by digital processing schemes most of the communication that takes place is in digital domain. Software Defined Radio (SDR) technology is at the heart of most modern transceivers today, and they help in serviceability across domains for mobile end-users by providing multimode and multi-standard communication [2]. Particularly, the development of 3G and 4G wireless has shown the need for a configurable/programmable platform instead of a rigid and complicated hardware based designs, and this where SDR has been most helpful [3]. As mentioned in [4] the prevalent architecture of most transmitter SDR systems is a combination of analog and digital components. In this architecture, the digital signal processing algorithms are usually implemented on the digital side. This paper tries to separate the digital components of the transmitter and implements it on an FPGA device. The alternatives for hardware implementation part of SDR based systems include digital signal processors, general purpose processors and application- specific integrated circuits

(ASICs) [5]. However, FPGAs afford the greatest flexibility, parallelism (hence performance) and prototyping, so it seems the best choice among these alternatives.

A flowchart depicting the basic stages of processing within a transmitter is given in Figure 1.

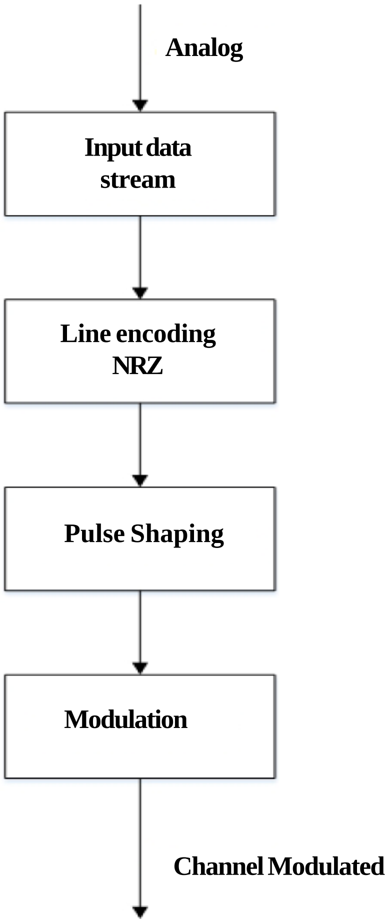


Figure 1. Processing blocks in a transmitter sub-system

Some work on BPSK implementations has already been carried out on GPP/DSP/FPGA or a hybrid of these platforms [6] [7]. Among other modules, this paper will seek to port BPSK to an FPGA-only design, and integrate it within other transmitter functions. In [8] different modulation approaches namely, binary amplitude shift keying(BASK), binary phase shift keying(BPSK), binary frequency shift keying(BFSK) and quadrature phase shift keying(QPSK) are synthesized on FPGA using VHDL and performance compare. However, this paper will attempt a novel DDS based method for carrying out BPSK modulation.

Similarly, in [9] and [10] the hardware design approaches for carrying out baseband modulation are discussed. Some of them are architecture dependent and some are geared to optimize the technology or channel utilization.

This paper is organized in separate sections: II.A is about Matlab algorithm for baseband modulation and II.B transforms this into hardware implementation. Section III.A discusses the architectural details of hardware for passband modulation and III.B develops the hardware that provides channel effects. Then, section IV provides functional results of the designed transmitter and also FPGA synthesis results are also presented. Lastly, section V concludes the paper by mentioning some applications and possible future improvements.

2. Baseband Modulation

2.1 Baseband modulation algorithm

The data originating from analog source that needs transmission might include voice, textual information, visual, or any other format in continuous representation. Initially, this data has to be '*formatted*' so as to obtain a bit stream which then can be processed further in transmitter. Input data sampling, quantization and formatting is shown in Figure 2. After the conversion to digital form, the bit stream looks as in Figure 3. Note the presence of discrete steps in the possible values and removal of continuity.

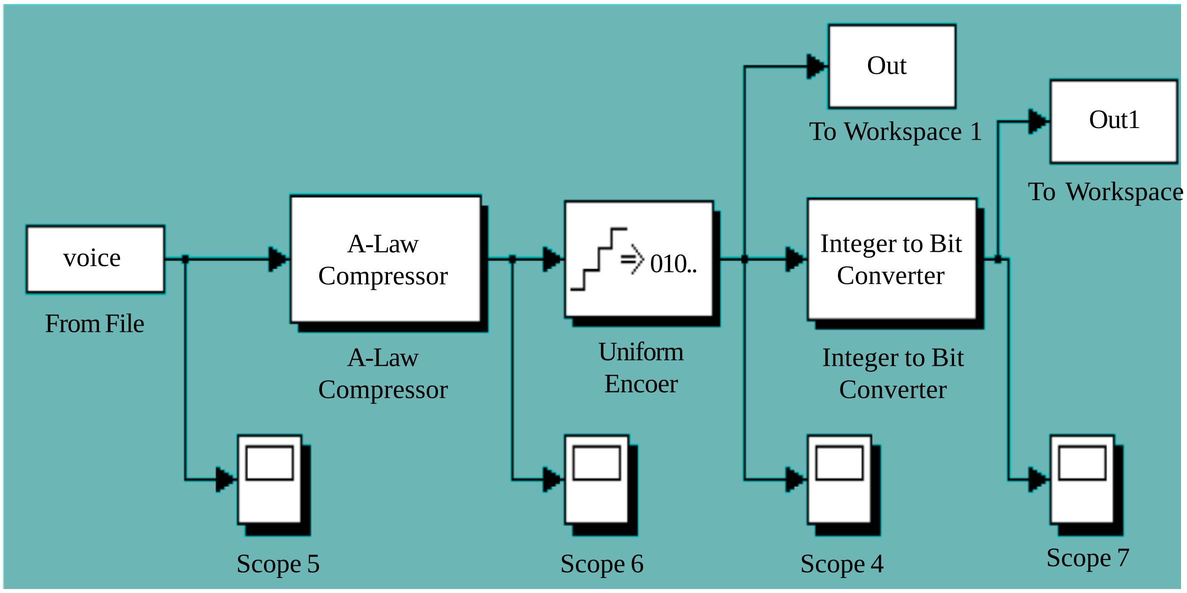


Figure 2. Analog to digital conversion of input stream

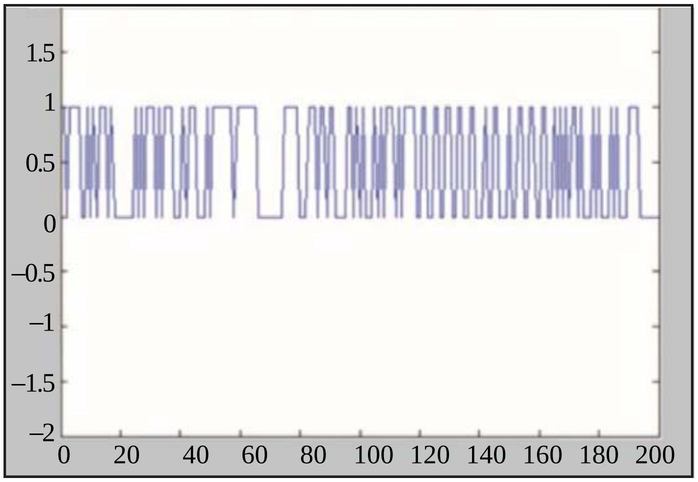


Figure 3. Typical waveform of digital data after conversion

Now at this point in time after data formatting i.e. sampling, quantizing and reshaping, data is in form of bits that are 1's and 0's. Now for baseband modulation the bits have to be encoded using one of the line encoding schemes. Most popular are NRZ, RZ, Manchester code etc. This paper implements the NRZ scheme. In NRZ line encoding a 1 is represented by a positive voltage level which we have chosen to be +1 and zeros are represented by negative level which in our case is -1. So as the name suggests it never goes to zero. But this scheme requires a symbol synchronization technique to be used in receiver. After undergoing NRZ line encoding, the waveform transforms to that shown in Figure 4.

Now note that these are simple rectangular pulses. For rectangular pulse, the equivalent in frequency domain is a sinc fuction. But the problem that we will face while transmitting these rectangular pulses is that practical and physical channels are always band limited and in frequency domain signal is not band limited so when it will pass through a channel that is band limited, significant amount of frequency components will be clipped. This will distort original signal and lead to inter-symbol interference

[11]. Nyquist pulse shaping criteria has to be met in order to achieve zero ISI. Nyquist pulse shaping criteria is practically applied by using pulse shaping filters in base band. It means that just after performing NRZ encoding the data is passed through a pulse shaping filter in baseband that removes high frequency components in signal and smoothens the abrupt transitions in it. Commonly used pulse shaping filters are raised cosine filters and sinc filters, the former of which is implemented in this paper.

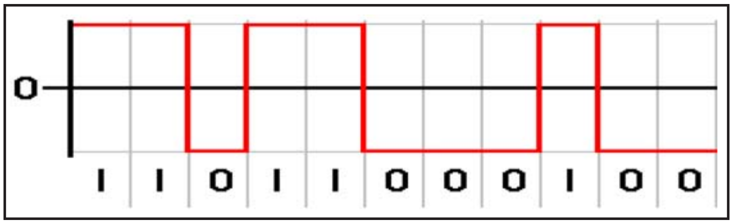


Figure 4. Waveform after NRZ line encoding

2.2 Hardware implementation of Baseband modulation and pulse shaping

A diagram illustrating the hardware architecture of transmitter design is shown in The bits of data that result after formatting of the analog input are stored in FPGA using its block memory generator. This memory is easily accessible through a counter. A digital clock manager (DCM) also had to be used for working with different clock domains - two clocks of 6.25 and 100 MHz are required in the design hierarchy. Input to DCM is 50 MHz system clock of Xilinx Spartan 3E XC3S500E FPGA board which is the target device used for implementing transmitter. One way of implementing raised cosine filter is using conventional approach that is quite unnecessarily sophisticated. So this paper uses the ROM based design approach.

In ROM based filter design approach a memory is reserved inside FPGA to store possible outputs of filter corresponding to all possible combinations of inputs that are arriving at memory (filter) through shift register of finite length. For example, as it is in this case if shift register is 5 bits wide there are 32 possible combinations of 5 bits; so first simulations are performed in Matlab to obtain all possible outputs for every combination of input bits. Then after performing floating to fixed point conversions these outputs are stored in memory which is in fact a representation of filter table. But actually the filter's address bus is 9 bits wide, 5 MSB's are which as already mentioned coming from input storage. The 4 LSB's are coming from a counter clocked at 100 MHz. Shift register was clocked at 6.25 MHz, so a 16 times faster clock is used for clocking these 4 LSB's and is accounting for oversampling factor of our root raised cosine filter. The 100 MHz clock is also fed simultaneously to access filter memory.

Now arrangement of memory also needs some explanation. As already mentioned there are 32 chunks of data inside memory corresponding to 5 MSB's of address bus. Inside each chunk there are 16 values, all 8 bits wide that correspond to 4 LSB's of address bus and are a requirement of oversampling. So ROM size becomes 512 * 8 bits. First 5 MSB's select 1 of 32 chunks and then 4 bit counter making up LSB's selects among 16 values that reside inside each chunk. Therefore, with arrival of each new bit, address changes and corresponding to that new bit 16 values are sent out by filter (because oversampling factor is 16). In this manner the complete filter structure is realized in hardware. Order of filter was kept to 65 with a group delay of 32. Each value is represented in hardware by 8 bits. Also note that although the input data to raised cosine filter is unipolar because it is acting as address of ROM based filter, the output is shaped bipolar baseband wave.

3. Passband Modulation

The scheme which is used for passband modulation in this paper is binary phase shift keying or BPSK. It is a very simple modulation scheme in which two phases are involved and they are separated by 180 degrees. In other words we assign a sine wave with zero initial phase for a 0 and a sine wave with 180 degrees initial phase angle for 1. This scheme although very simple is effective in the sense that it is very robust to noise. Since all the information is embedded in phase inversions of wave, so amplitude of signal which can be disturbed by noise is unimportant.

3.1 Hardware design of BPSK

As mentioned in previous section, the binary data is in form of shaped pulses which will now be modulated by a carrier. A diagram depicting this scheme is shown in Figure 6.

The carrier that is used to modulate the shaped pulses is generated using DDS core on FPGA.

DDS stands for direct digital synthesis and is also known as numerically controlled oscillator NCO. This is very important component of digital communication systems because it performs the same job as that of a VCO in analog communications. DDS generates a sinusoid using look up table approach. The samples of sine are stored in the look up table. Its phase argument is generated by a simple digital integrator. So simply by giving a phase argument of 90 degrees we can generate cosine wave as well. The cosine generated by DDS is used by as a carrier and its frequency is 10 MHz. Block diagram of DDS is shown in Figure 7 . The use of DDS module, and its position in the entire hardware arrangement of Passband module is given in Figure 8.

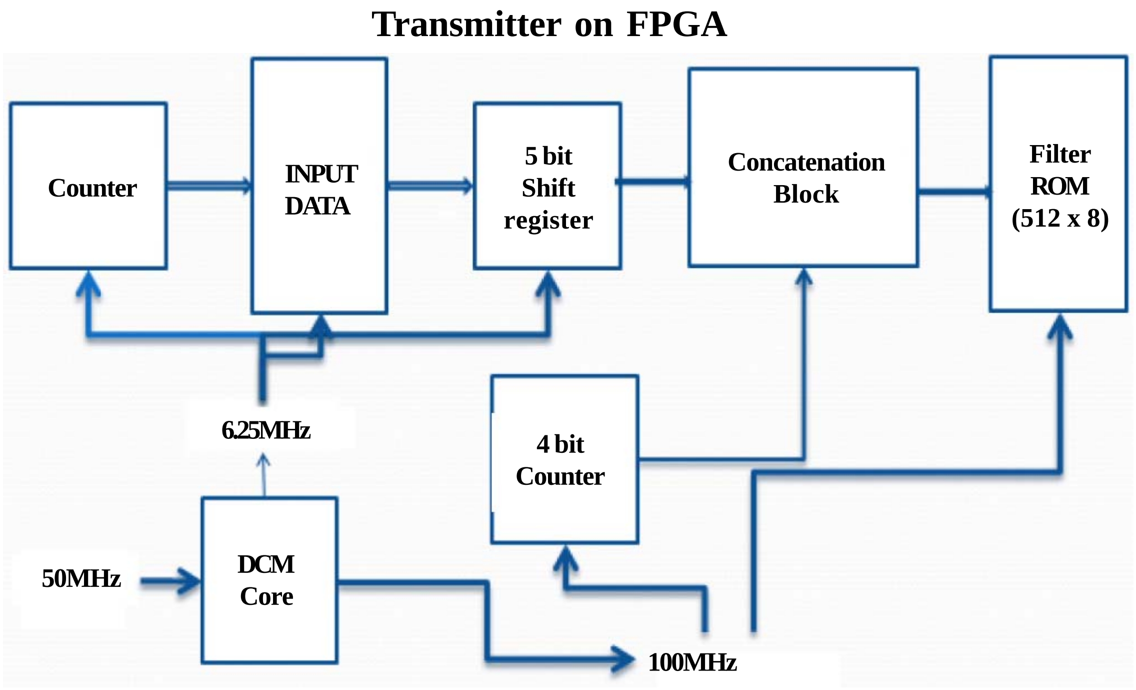


Figure 5. Architecture design of FPGA based transmitter

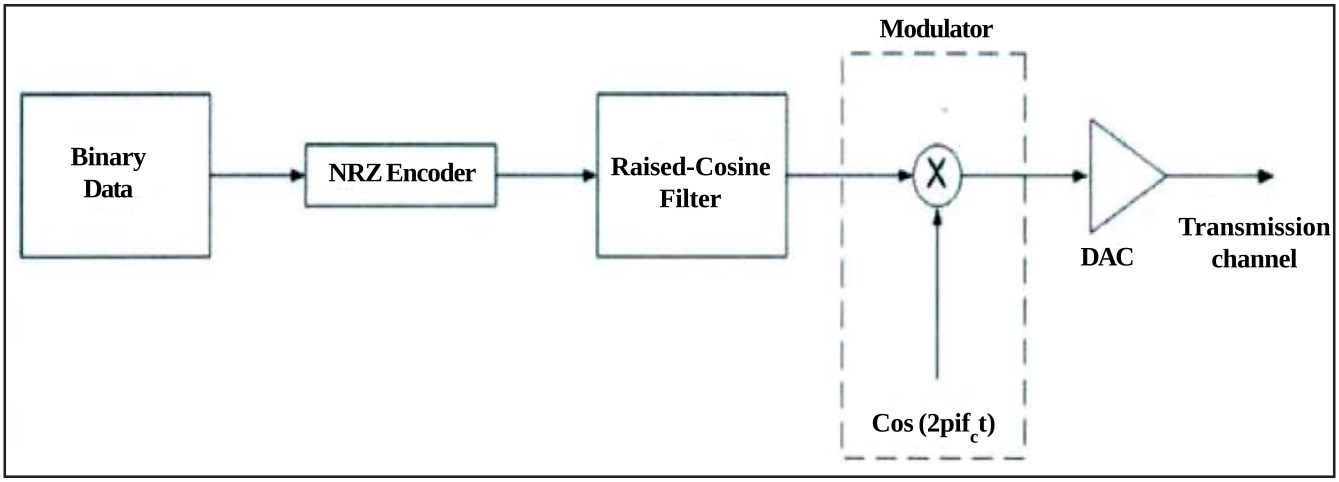


Figure 6. Passband modulation stage emphasized in transmitter sub-system

3.2 Modelling Channel Effects

Just like the real channels, different '*channel effects*' were catered while considering the impact of practical links in a transceiver system. To realize a feasible communication system that can interface over existing practical channels, it is pertinent to take care of the distorting effects. Figure 9 shows how these effects were introduced on the transmitter side.

Propagation time = Distance / propagation speed.

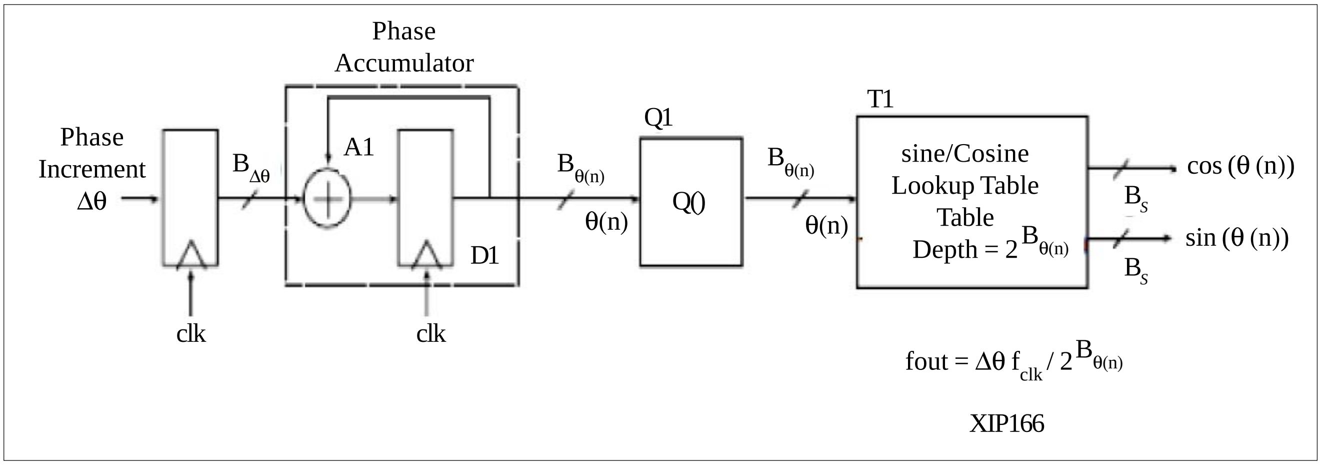


Figure 7. Block diagram of DDS elements

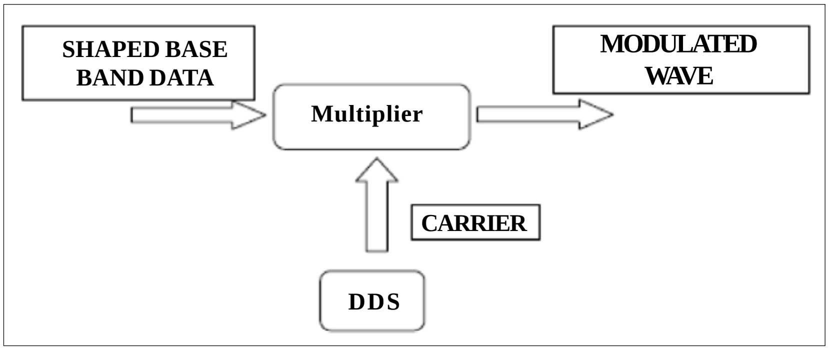


Figure 8. Using DDS to implement hardware Passband modulation

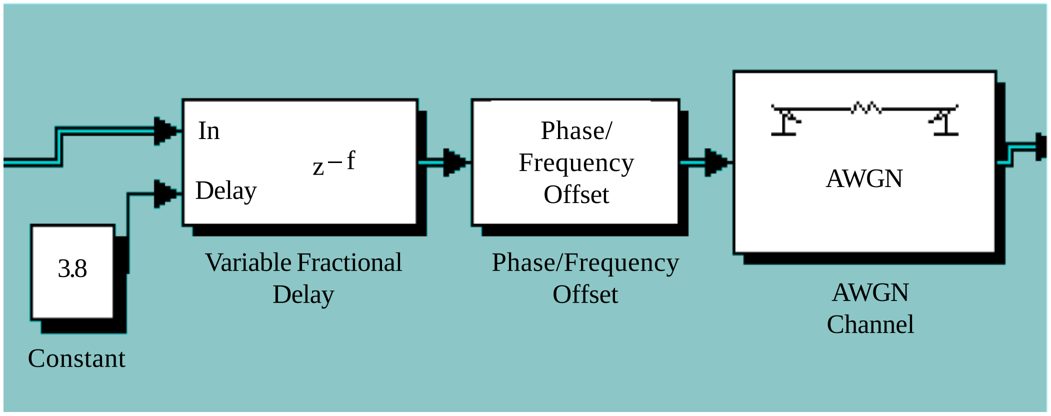


Figure 9. Introducing channel imperfections for modelling real environment

Now as is shown in diagram a delay of 3.8 samples is introduced in order to model propagation delay of medium. In time it will equal:

Delay = (3.8 samples/8 samples per sample) *10m sec per symbol = 4.75msec. In this manner a delay that equals 4.75 msec is added in time scale as is obvious from eye diagrams of Figure 10 and Figure 11.

4. Results

The result that is observed on oscilloscope after the completion of baseband stage i.e. encoding and pulse shaping is shown below in Figure 12.

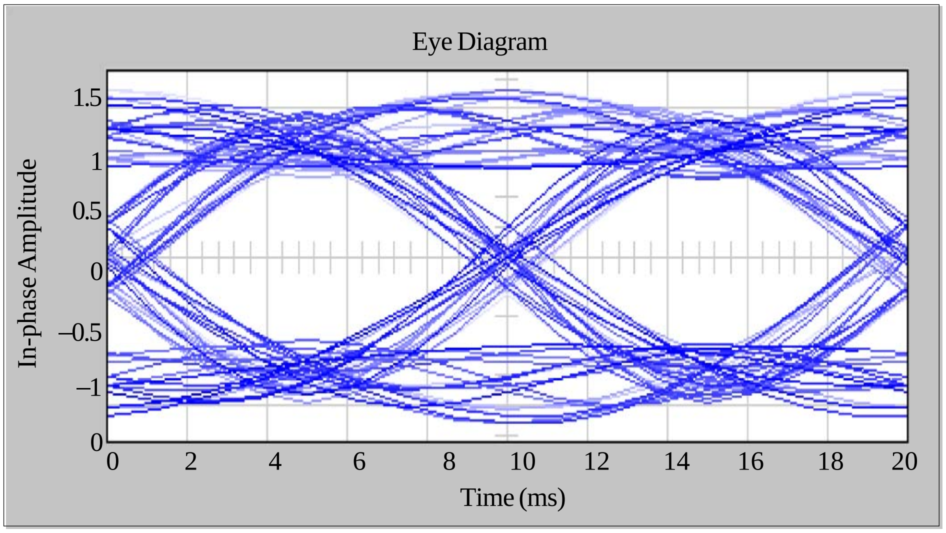


Figure 10. Eyediagram before adding delay

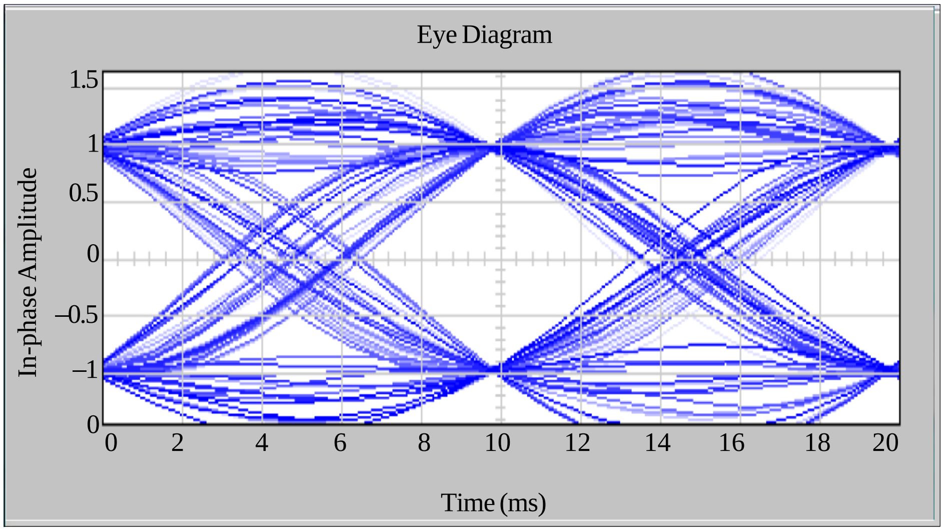


Figure 11. Resultant eyediagram after addition of delay

During baseband stage, the clock by clock simulation results as shown by Xilinx ISim and actual data seen through ChipScope are shown in Figure 13 overleaf. The figure confirms that the results completely agree and verifies the hardware implementation of baseband module.

Next step is to simulate the hardware passband module. For this purpose, the waveform before input to the modulator is shown in Figure 14. The results of passing this waveform through passband modulation stage is given in Figure 15. A separate hardware module was also developed for inducting channel imperfection. Figure 16 shows the result on passband waveform after it is sent through this particular module. Note the distortion brought about by channel effects and deviation can be seen from the ideal passband output.

The hardware design was successfully synthesized and mapped onto a Spartan 3E board. The design summary showing FPGA board resource utilization is given in Figure 17.

5. Conclusion and Suggested Improvements

So to review what has been accomplished in this paper: we took input bits (data from source), performed its NRZ line coding,

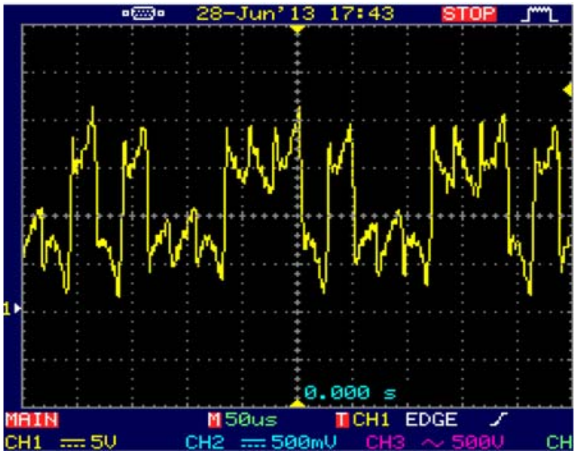


Figure 12. Results after encoding and pulse shaping

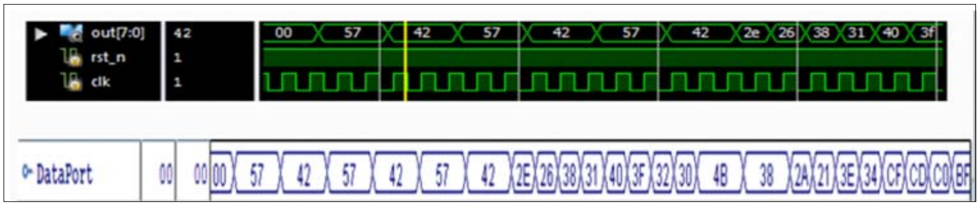


Figure 13. Comparison of simulation results and actual stream

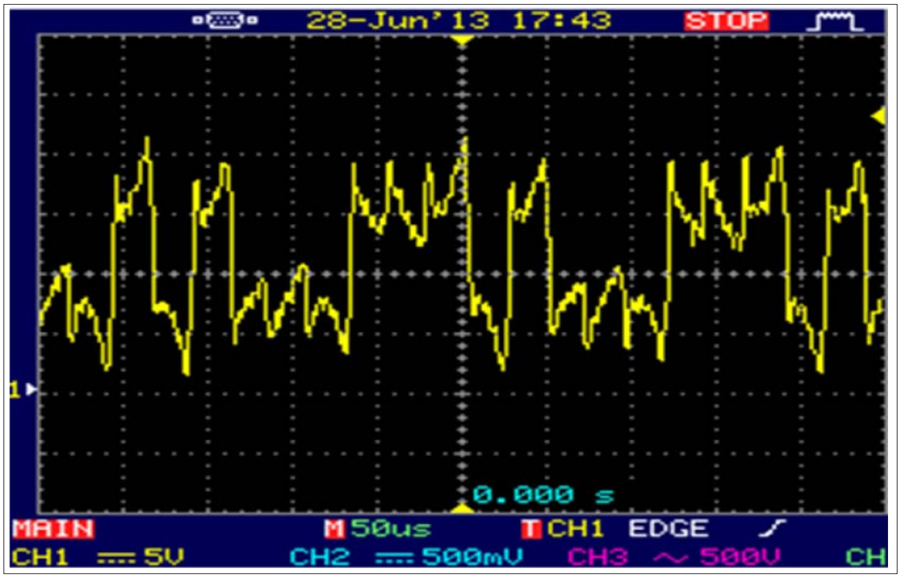


Figure 14. Before passing to passband modulator

then shaped it using raised cosine filter, modulated it using BPSK scheme for passband transmission and finally included channel effects. All this has been successfully mapped onto FPGA and results conform to the expected outputs. In essence, a working transmitter module has been achieved on the FPGA device, and allowance has been left for studying channel behavior by including its effects.

This transmitter can find use in applications requiring low-power and high throughput. These features are inherent in the FPGA bed which underlies this design. Hence, a very suitable candidate for use are communication systems employing WiMAX technology. The transmitter design, if coupled with a synchronizing receiver subsystem can be used in educational setting because the intermediate channel effects have been completely modelled. Therefore a complete communication transceiver system can be realized on single FPGA board for experimentation, research and education.

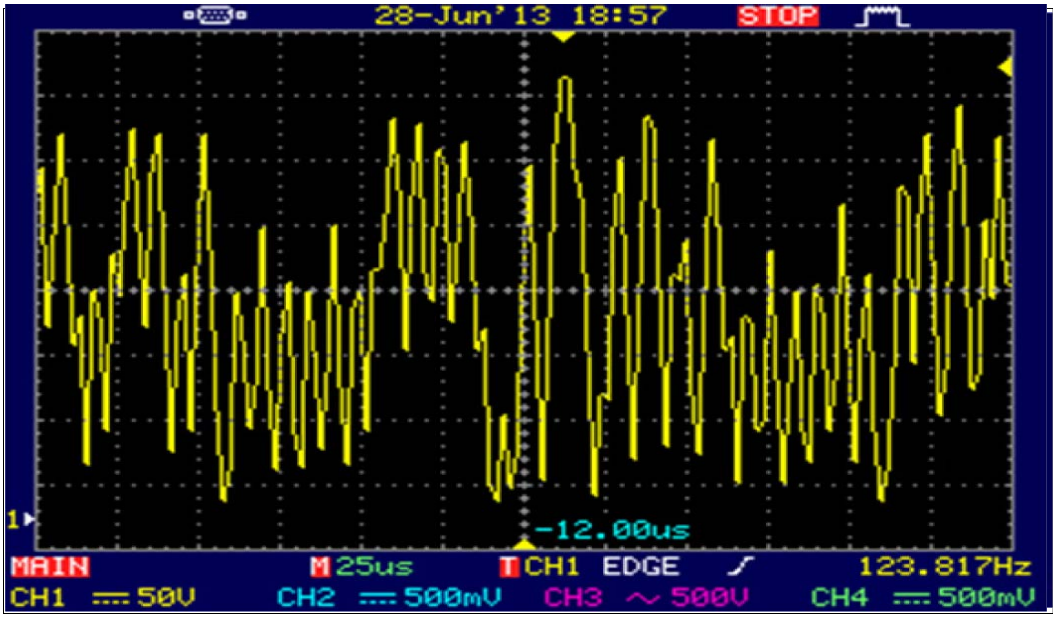


Figure 15. Result of passband modulation

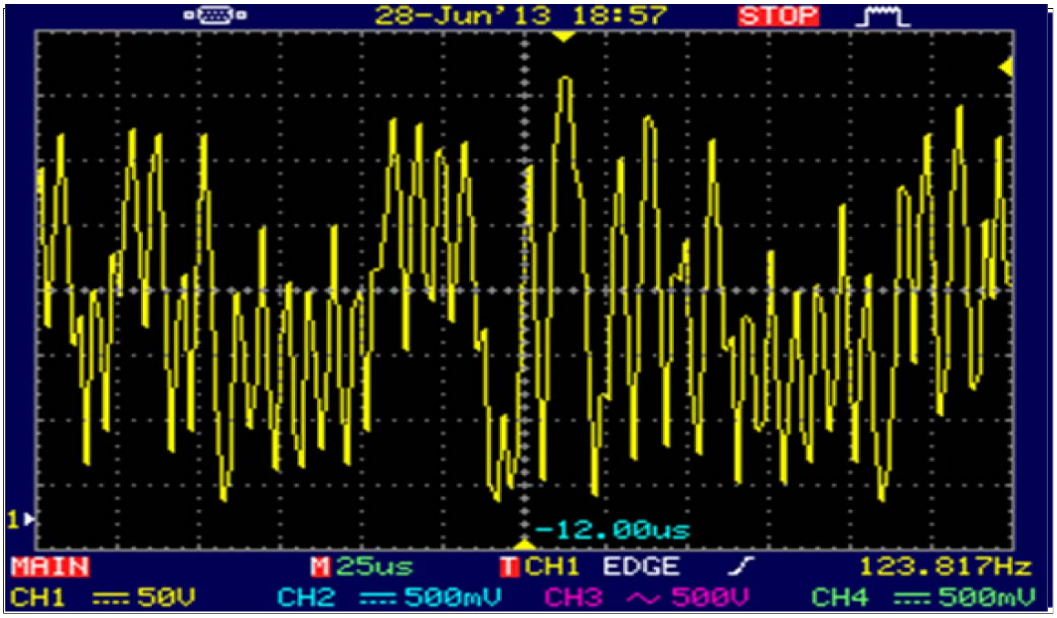
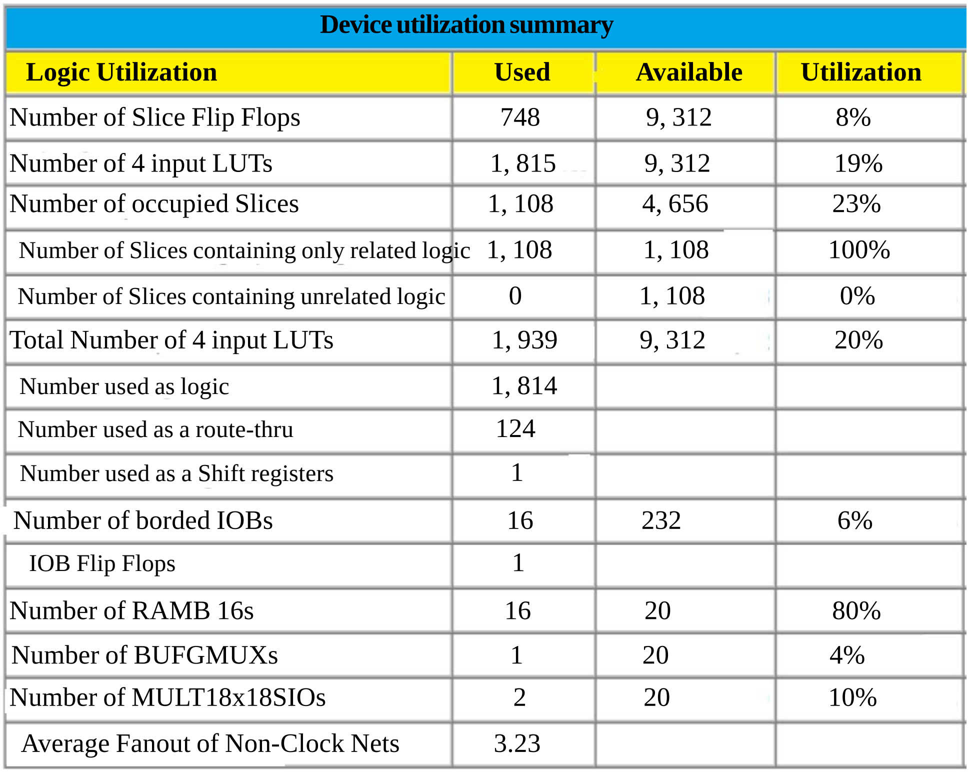


Figure 16. Passband waveform after undergoing channel degradations

Also, the design has been kept flexible making it readily possible to attach this core in a larger system (not necessarily communication) employing wireless data transfers.

There is some room for improvement in this work because the main aim was to design a barebones architecture on FPGA without caring for much details. Firstly, source coding modules can be installed in this transmitter, similar to one discussed in [12]. It will allow data compression features which come essential in saving bandwidth and power.

Secondly, channel coding is another possible improvement in this module. In channel coding redundant bits are added to symbols which are already subjected to source coding. The addition of these extra bits gives an opportunity to detect and correct errors at the receiver. Channel coding using Forward Error Correction (FEC) is a major part of many existing error detection and correction algorithms. A good discussion on this particular implementation is provided in [13].



Device utilization summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	748	9, 312	8%
Number of 4 input LUTs	1, 815	9, 312	19%
Number of occupied Slices	1, 108	4, 656	23%
Number of Slices containing only related logic	1, 108	1, 108	100%
Number of Slices containing unrelated logic	0	1, 108	0%
Total Number of 4 input LUTs	1, 939	9, 312	20%
Number used as logic	1, 814		
Number used as a route-thru	124		
Number used as a Shift registers	1		
Number of borded IOBs	16	232	6%
IOB Flip Flops	1		
Number of RAMB 16s	16	20	80%
Number of BUFGMUXs	1	20	4%
Number of MULT18x18SIOs	2	20	10%
Average Fanout of Non-Clock Nets	3.23		

Figure 17. Resource utilization summary on FPGA board

Thirdly, a module to carry out encryption can be developed to provide information security. In order to transmit the message secretly, original message at transmitter is encoded using an encryption algorithm so that no intruder or eavesdropper is able to decipher it. Design details of AES encryption on FPGA for wireless communication are discussed in [14].

References

[1] Johnson, P. (1985). New Research Lab Leads to Unique Radio Receiver, *E-Systems Team,* 5 (4) 6-7, May.

[2] Cummings, M., Haruyama, S. (1999). FPGA in the software radio, *Communications Magazine, IEEE*, 37 (2) 108-112.

[3] Mitola, J. (1995). The software radio architecture, *Communications Magazine, IEEE,* 33 (5) 26-38.

[4] DoCosta, K. (2001). Virtex-II DSP Engines Enable Software, *Xcell Journal*, p. 82-83, Fall/Winter.

[5] Safadi, M. S., Ndzi, D. L. (2006). Digital Hardware Choices For Software Radio (SDR) Baseband Implementation, in *Information* and Communication Technologies. ICTTA '06. 2*nd*, Damascus.

[6] Tachwali, Y., Refai, H. (2008). Implementation of a BPSK Transceiver on Hybrid Software Defined Radio Platforms, *In*: *Proc. Information and Communication Technologies: From Theory to Applications, IEEE*, Damascus.

[7] Mohamed, K. E., Ali, B. M. (2005). Digital design of DS-CDMA transmitter using VHDL and FPGA, in Networks. Jointly held with the 2005 IEEE 7th Malaysia International Conference on Communication.

[8] Demiri, F. M., Kafadar, U., Dikmese, U., Dincer, H. (2007). FPGA Based Implementation of Communication Modulation, *In*: IEEE 15th Signal Processing and Communications Applications, Eskisehir.

[9] Hatai, I. (2012). FPGA Implementation of a Reconfigurable Baseband Modem for SDR System: Reconfigurable Multi-Standard Baseband Modulator and Demodulator Design, LAP LAMBERT Academic Publishing.

[10] Wei, W., Shu-Shin, C., Sangjin, H. (2002). A coarse-grained FPGA architecture for reconfigurable baseband modulator/ demodulator, in *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA.

[11] Sklar, B., Ray, P. K. (2001). Intersymbol Interference, in *Digital Communications: Fundamentals and Applications*, Second Edition ed., Dorling Kindersley (India) Pvt. Ltd., p. 136-149.

[12] Boukadourn, M., Tabari., M., Bensaoula, A., Starikov, D., Aboulhamid, E. M. (2005). FPGA implementation of a CDMA source coding and modulation subsystem for a multiband fluorometer with pattern recognition capabilities, *In*: IEEE International Symposium on Circuits and Systems.

[13] Ahlquist, G. C., Rice, M., Nelson, B. (1999). Error control coding in software radios: an FPGA approach, *Personal Communications, IEEE*, 6 (4) 35-39.

[14] Gupta, A., Ahmad, A., Sharif, M. S., Amira, A. (2011). Rapid prototyping of AES encryption for wireless communication system n FPGA, in 2011 IEEE 15th International Symposium on Consumer Electronics (ISCE), Singapore.