Technical University of Denmark

Modulation and Optical frontend

Sai Lou S135234
Ognjen Jovanovic S171736

supervised by

Darko Zibar
Henrik Wessing

June 19, 2018

Abstract

This report aims to describe one of the important functions of a communication system, which is the modulation. This function processing a message signal that can be an analog or digital such that it can be transmitted through a physical medium like fiber.

A simulation was developed in MATLAB in order to obtain performance benchmarks that will be used to evaluate the performance of the hardware implementation. A transmitter and receiver were developed by using hardware description language VHDL in Quartus.

In order to present the performance of different modulations, Bit Error Rate (BER) is used as a parameter for measuring the quality of a signal. The simulation is conducted using Monte Carlo simulations to obtain a result that is accurate as possible. While the test of the hardware design was completed in a scenario where noise is neglected.

Simulations were performed for 2PAM, 4PAM, 8PAM modulations for different SNR levels. The design of the hardware was synthesizable in Quartus tool and it can reach a working frequency of 80.93 MHz, which satisfy the requirement of this system, that is 8 MHz. The hardware implementation was verified using Signal Tap Logic Analyzer. Furthermore, the baud rate that the system can reach is up to 10 Mbaud/s, but due to the limitation of other blocks maximum of 1 Mbaud/s is used.

Contents

1. Introduction

A telecommunication system is a highly complex system described with different protocols, modulations, error correcting codes depending on the quality of service one needs to provide. Different functions are used in the system to transmit data from source to destination. These functions have different functionality and will be used depending on the requirements of the services, which the system is providing. The transmitting and receiving side are connected through a transmission medium (etc. optical fiber, air, copper wires), which is not ideal, since different distortions such as noise, delay, interference, limited bandwidth occur. These problem are coped on the physical layer, where the modulation is used to provide better spectral efficiency and overcome the noise. It should emphasize that there are errors, which are going to be corrected by the error correction part. The random delay, which is introduced in the channel requires synchronization. Thus, the clock recovery is used to ensure that the data is sampled at the right point.

In this project the modulation, demodulation and pulse shaping have been implemented in hardware and software. The goal of the hardware implementation is to be able to transmit and receive a binary sequence at a given rate. While the software implementation is used to evaluate the hardware implementation. The main idea was to develop a simulation model in MATLAB and afterwards implement the system on an FPGA board.

The simulation model represents a transmitter-channel-receiver chain in which the performance of different modulations are reviewed and compared to the theoretical values. Using the developed simulation model, Monte Carlo simulations are conducted in order to obtain numerical analysis and to be able to estimate the performance in different working conditions.

The main goals of this project is to implement the system that supports 2,4 and 8 PAM modulation scheme in hardware. In order to verify the design, the results from hardware are compared with a referent model, which is the simulation model in MATLAB. Based on the obtained results from hardware and simulation in MATLAB, a conclusion will be provided.

The report is organized in several chapters. After this introduction chapter comes the second chapter in which a brief overview of the full system is given, followed by an explanation of the modulation and optics subsystem. This chapter will include the theory behind the used components. Afterwards, in the third chapter, a detailed description of the software and hardware implementation will be presented. Following in chapter four, the results of the both analysis are conducted and discussed. Finally, in the last chapter the main conclusion will be made based on the obtained results.

2. Detailed overview of the modulation subsystem

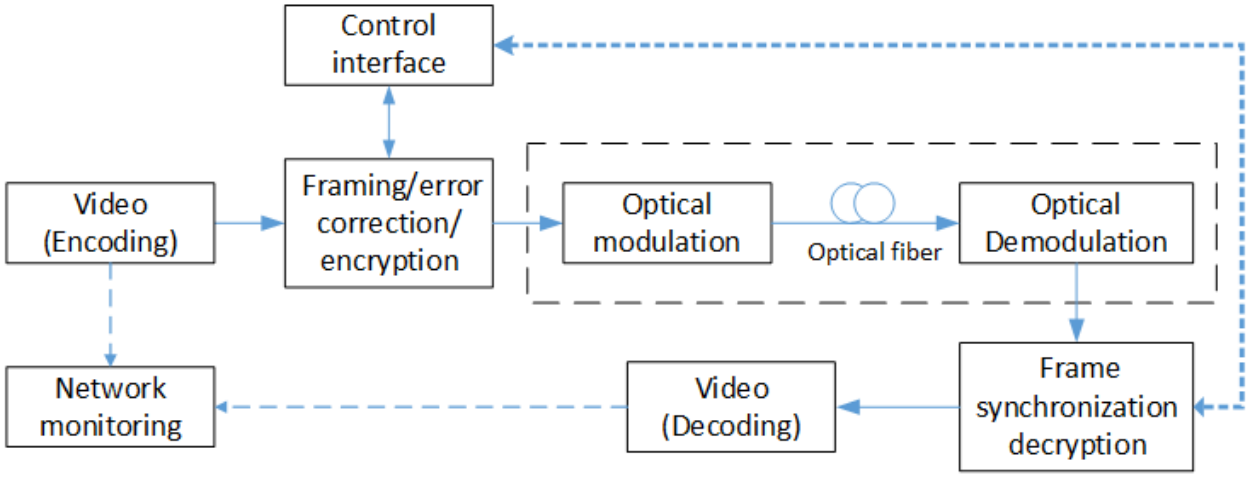


Figure 2.1: Block diagram of video transmission system

A brief overview of full communication system is shown in Figure 2.1. The purpose of entire project is to establish a communication system where it is possible to transmit a video through the system and receive it again. The entire project is divided into small subprojects, where each subproject is carried out by a small group. This subproject is focusing on the optical modulation and demodulation part, which is shown in figure 2.1 (dash box).

2.1 Detailed overview of the modulation part

Figure 2.2, shows the overview of entire subproject, where a stream of information bits is generated with PRBS and transmitted through the system and receive it again. Every block represents a different functionality, where all the blocks are essential to provide a fully working subsystem. These functionalities are presented in different boxes, which are shown in Figure 2.2.

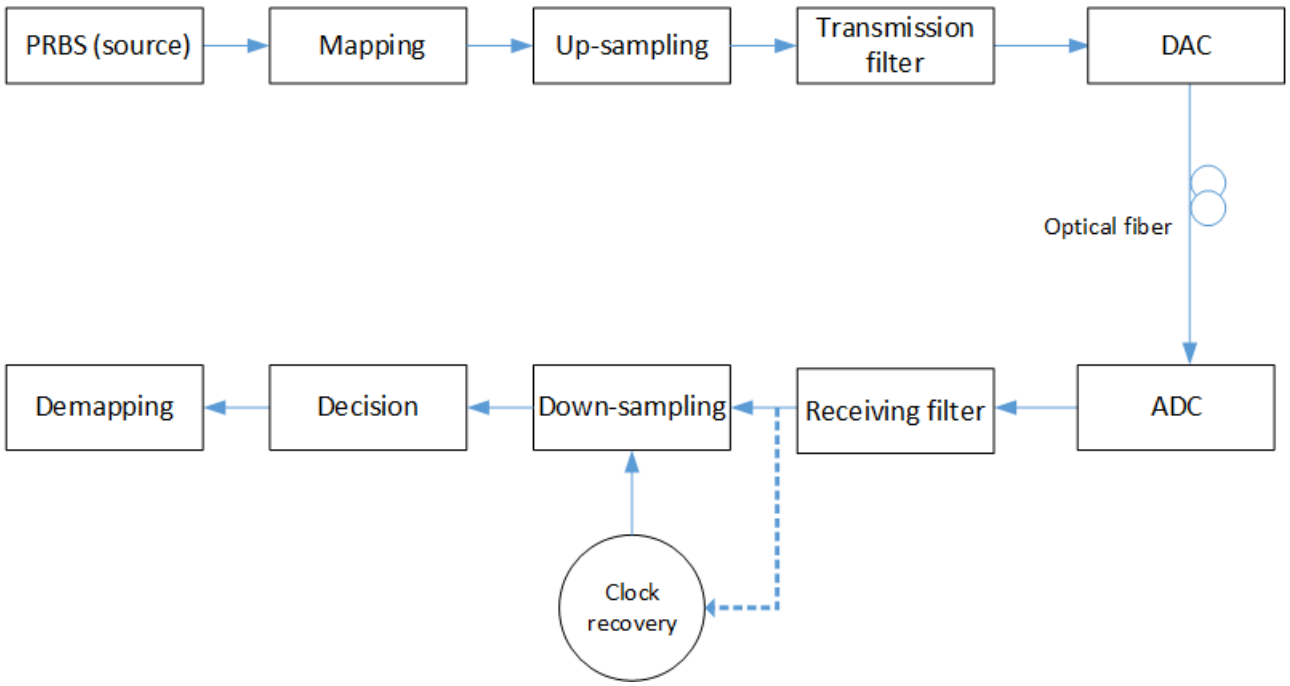


Figure 2.2: Modulation and optical frontend

2.1.1 PRBS

Pseudo random bit sequence (PRBS) is used for the simulation purpose, where PRBS represents the information bits, which will be transmitted through the system and received again. It is used because its power spectrum is almost constant and it has periodic autocorrelation, where the period is defined by the degree of the polynomial m, $2^m - 1$. PRBS is generated by using a polynomial and it repeats itself periodically, the period of the sequence depends on the power of the polynomial. Another reason PRBS is used for testing is due to the simplicity of the implementation in hardware, since PRBS is generated using a polynomial, it can be implemented as a linear feedback shift register (LFSR). Furthermore, PRBS is also used in this case in order to generate random bit sequences, which are close to the real communication system, since the information bits in a communication system have random sequences.

The PRBS generator is shown in Figure 2.3, where the generated random bit sequences are x_k, which is calculated with different coefficients g_i where i goes from 1 to m, which is the order of polynomial. Furthermore, the g_i can either be 0 or 1. The calculation of PRBS can be expressed as a linear recursion:

$$x_k = g_1 x_{k-1} + g_2 x_{k-2} + .. + g_m x_{k-m} \ mod \ 2, g_m = 1 \tag{2.1}$$

In the design PRBS7 is used, which has a generator polynomial $1 + z^{-6} + z^{-7}$, which means that the period of the sequence is 127.

Nevertheless, another reason of using PRBS is to generate a bit sequences that provides enough transition for 8PAM, such that all symbols occur. Since 8PAM uses 3 bits, it is enough to use PBRS7.

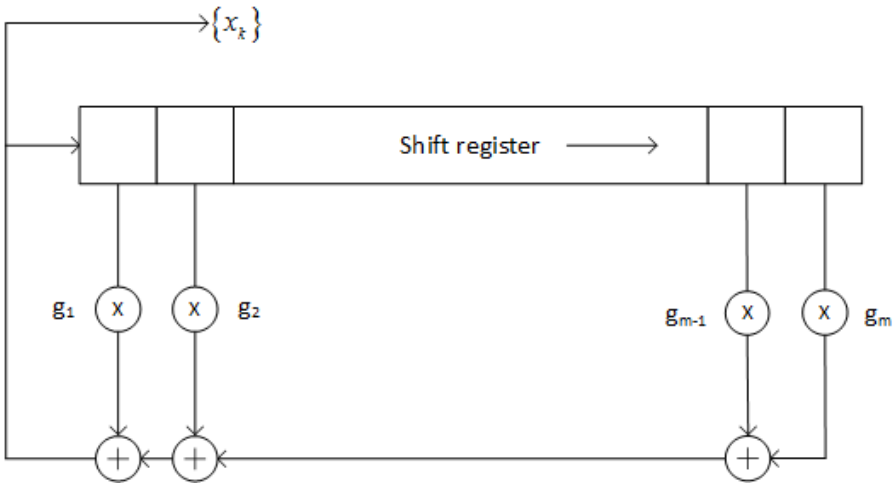


Figure 2.3: Shift register (PRBS generator)

2.1.2 Modulation

The modulation is combining a certain number of consecutive bits into a symbol. The number of modulation symbols are 2^n, where n is the number of bits that is combined. On the receiving part, demodulation is done by converting the symbol into an appropriate combination of n bits. The main problem with going to higher modulation schemes is that it is more sensitive to noise, thus a higher SNR level is required. This is due to the spacing between the symbol is smaller for higher modulation schemes, in order to keep the same power. Thus, the noise that occurs on each symbol will have more impact, when the decision is made in the receiver. The reason, that a higher modulation scheme is desired to be used is because it provides a better spectral efficiency.

Table 2.1: Binary to Gray code 4 PAM

Level	Binary	Gray code
-3	00	00
-1	01	01
1	10	11
3	11	10

Table 2.1 and 2.2 show the table of conversion from binary to Gray code for 4PAM and 8PAM. Notice that the difference between the bits of two symbols before gray code is introduced are greater. Receiving the neighboring symbol due to noise distortion is most probable, therefore it is desired to have only one bit difference between them. This can be solved by using Gray coding, which is shown in table 2.1 and 2.2, where the 4PAM and 8PAM with Gray coding are presented.

Table 2.2: Binary to Gray code 8 PAM

Level	Binary	Gray code
-7	000	000
-5	001	001
-3	010	011
-1	011	010
1	100	110
3	101	111
5	110	101
7	111	100

The following equations show the method to convert binary to Gray coding 2.2 [3] and vice versa for 8PAM 2.3 [3]:

$$y(2) = x(2);\ y(1) = x(1) \bigoplus x(2);\ y(0) = x(1) \bigoplus x(0) \tag{2.2}$$

$$y(2) = x(2);\ y(1) = x(1) \bigoplus x(2);\ y(0) = x(2) \bigoplus x(1) \bigoplus x(0) \tag{2.3}$$

where the x represents the binary and the y represents the Gray coding, the indexes show the position of the bit in the binary symbols.

2.1.3 Pulse shaping

As it was described already, the channel is limited and therefore the shape of the pulse cannot be rectangular. However, this would be ideal, but it has an infinitely large bandwidth which makes it physically impossible, therefore pulse shaping needs to be used. The purpose of pulse shaping is to narrow the bandwidth of the signal as much as possible in order to fit into the limited channel bandwidth. In this way, the appearance of inter-symbol interference (ISI) is avoided. On the receiving side, a matched filter is used, and the output can be thought of as correlation of the received signal and the known pulse. Furthermore, the sampling is done where the correlation has its maximum value. As it was described, it overcomes ISI, but not the Gaussian noise. The pulse shaping filter which will be used, at the transmitter and receiver side, is square root raised cosine (SRRR). The reason SRRR is used as the filter is because a minimum intersymbol interference is desired and SRRR is the filter that can achieve this.

2.1.4 Chanel (Optical fiber)

The channel introduces different distortions such as noise, delay, interference and limited bandwidth. Different type of noises will occur during the transmission of signal through the fiber and when the signal is received at the receiver. Amplifier is used in optical communication to compensate the losses, however amplifier adds noise to the signal. This type of noise is Amplified Spontaneous Emission (ASE) - if EDFA or Raman amplifier are used - which has an impact on

the performance of the signal. ASE will occur in all optical amplifiers, hence it is unavoidable. However, the impact of ASE can be reduced by using different amplification methods like Raman distributed amplification, where rare earth dopant is avoided to be used and the amplification occurs within the fiber. The idea is, less spontaneous emission is introduced if EDFA is not used, since the amplification in EDFA is depending on stimulated emission, where spontaneous emission will also occur. Furthermore, the distributed amplification prevents the signal power drop to the noise limit, hence more noises are avoided [1].

But, there are some noises that are unavoidable and difficult to control in the optical communication system such as receiver noise, which includes shot noise, beat noise and thermal noise [1]:

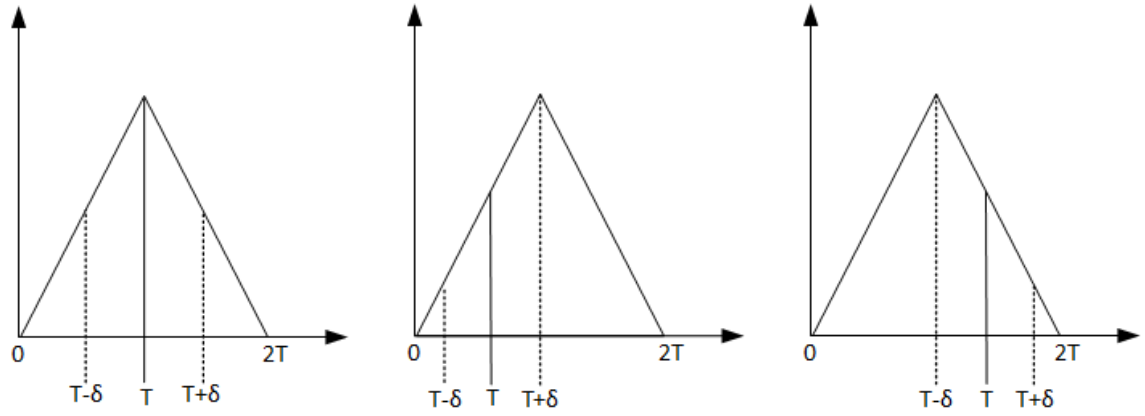
- Shot noise is generated by the receiver itself, where some electrons are generated randomly at random time, when the receiver is active.
- Thermal noise is depending on the temperature, since the electrons will fluctuate more randomly when the temperature is high.
- Beat noise is created by the signal beating with the spontaneous emission, which is generated by the amplifier.

Overall, it is very important to take noises into account in the system, since it has an impact on the quality of the signal. Furthermore, if more noises are added to the signal, the SNR will decrease, which reduces the performance of the system.

2.1.5 Early-late algorithm

As it was described before, the sampling needs to be done at the point where the absolute value of the correlation is maximum, and this is how the clock synchronization is done.

There are a few algorithms that can be used, but the early late is probably easiest to implement. The illustration of how Early-late algorithm works on the output of a matched filter is shown in figure a, b and c. Figure a shows the desired sampling point (T) of one pulse. Whereas, figure b illustrates, when the sampling is done too early and too late in figure c. Notice that the difference between samples T-δ and T+δ, when the sampling is perfect, is smaller than other two cases. This is the principle on which the early late algorithm is based on.



(a) (b) (c)

Figure 2.4: (a) shows perfect sampling, (b) shows the too early sampling, (c) shows the too late sampling.

The algorithm can be explained in a few steps [2]:

- Calculate the difference between the sample that is before and after the point in which the sampling should be done (figure 2.4b and 2.4c) and average it over S observation.
- The average is compared to a threshold and then n_0 is updated.
- Sampling in n_0

$$S_{n_0} = |V_{n_0+1}| - |V_{n_0+1}| \tag{2.4}$$

$$S_{av} = \frac{1}{S} \sum_{k=0}^{S-1} S_{n_0 - k*m} \tag{2.5}$$

The equation to calculate the average of the samples and the differences between the samples are shown in equation 2.4 [2] and 2.5. These equations are used in the Early-late algorithm to update n_0 based on the comparison between the average, a given threshold and upsampling factor m:

Algorithm 1 Early late

if $s_{av} > \delta$ *(a small value)* **then**
  $n_0 = n_0 + m + 1$ (move 1 to the right)
else if $s_{av} < -\delta$ **then**
  $n_0 = n_0 + m - 1$ (move 1 to the left)
else
  $n_0 = n_0 + m$ (keep the sampling point)

2.2 Interfacing

The important interfaces in this subproject is the interfacing on the transmitter and receiver as shown in Figure 2.5a and 2.5b, respectively. The transmitter interface, Figure 2.5a, receives the information bits in 1 bit from the framing block. Nevertheless, a sequence that includes information about which modulation scheme is desired to be used will also be received at the transmitter and receiver interface.

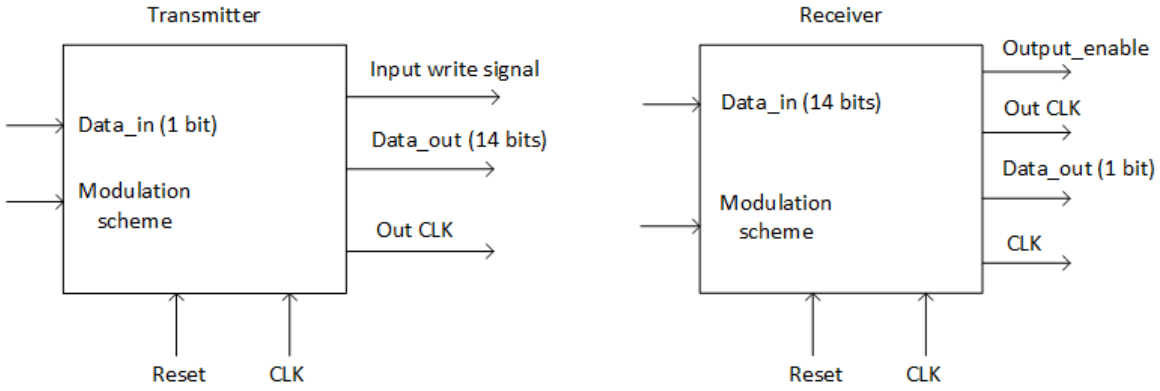


Figure 2.5: (a) shows transmitter interface, (b) shows receiver interface.

Transmitter interface will transmit the received information through the system, Figure 2.5b, and back to framing group through the receiver interface. The receiver interface receives the information bits in 14 bits and transmit to framing group in 1 bit. It should emphasize that the ADC, which is used in this project, provides a 14 bits resolution of conversion from analog to digital.

Table 2.3: Baud rate dependent on modulation

	Modulation block	Upsampler	Baud rate
2 PAM	1 MHz	8 MHz	1 Mbaud/s
4 PAM	0.5 MHz	4 MHz	0.5 Mbaud/s
8 PAM	1/3 MHz	8/3 MHz	1/3 Mbaud/s

The frequency of the entire system is 1 MHz and provides the clock constraint for this subsystem. Table 2.3 shows the output frequency of the mapping and upsampling block shown on Figure 2.2. Upsampling is performed by factor 8. From here, it can been seen that the system design needs to fulfill at least a frequency speed of 8 MHz.

Notice that the baud rate and the output frequency of the mapper are the same. This is because the input to the mapper is 1 bit and it combines the bits as described in Section 2.1.2. In the case of 2 PAM one bit is mapped to one symbol, therefore the baud rate is 1 Mbaud/s. While when combining n bits to a symbol the baud rate is decreased n times.

3. Implementation

3.1 Brief description of the simulation model and procedure

The simulation model is developed in MATLAB and it can be presented with the block diagram shown on figure 2.2. Every block represents a functionality, and they are:

- PRBS is the data source, which is explained in section 2.1.1 and the polynomial used to generate this sequence is $1 + z^{-6} + z^{-7}$.
- Modulation and demodulation are explained in section 2.1.2. Modulation represents a mapping process of $log_2(M)$ bits into a symbol, where M is the order of the modulation, based on the combination of the bits a proper symbol is chosen. Demodulation is the reverse process, a symbol is received and translated into $log_2(M)$ bits. The modulations that are used are 2PAM, 4PAM, 8PAM.
- Pulse shaping, a detailed overview is given in section 2.1.3. In this simulation a square-root raised cosine is used for both the transmission and receiving filter.
- Additive White Gaussian Noise (AWGN) channel that models the channel between the transmitter and the receiver with SNR that spans from 0 to 15 dB and inserts a random delay to the signal.
- Early-late, explained in section 2.1.4. As it was mentioned in the previous point, the channel inserts a delay, therefore this algorithm is used to provide synchronization (clock recovery).

Simulations were conducted to see the performance of each modulation when Gray coding is used and compare them with the theoretical values to verify the design. Afterwards, comparison between the used modulations is given.

3.2 Implementation of FPGA

The FPGA that will be used for the emulation of entire system is DE1-SoC from Altera. Nevertheless, the board where analog to digital and vice versa, will take place on THDB-ADA board. As described before, this ADA board provides a 14 bits resolution of conversion from analog to digital and vice versa. This board also include a build-in clock of 50 MHz, which are used together with a Phase Lock Loop (PLL), where the clock is adjusted to the desired speed.

The entire system is designed with VHDL, since all the components of the system will be placed on the FPGA, hence the logic of the system and the digital signal flow between the components are described with VHDL. Both the transmitter and receiver have multiple clocks in the design, hence a PLL is used in order to provide multiple clocks.

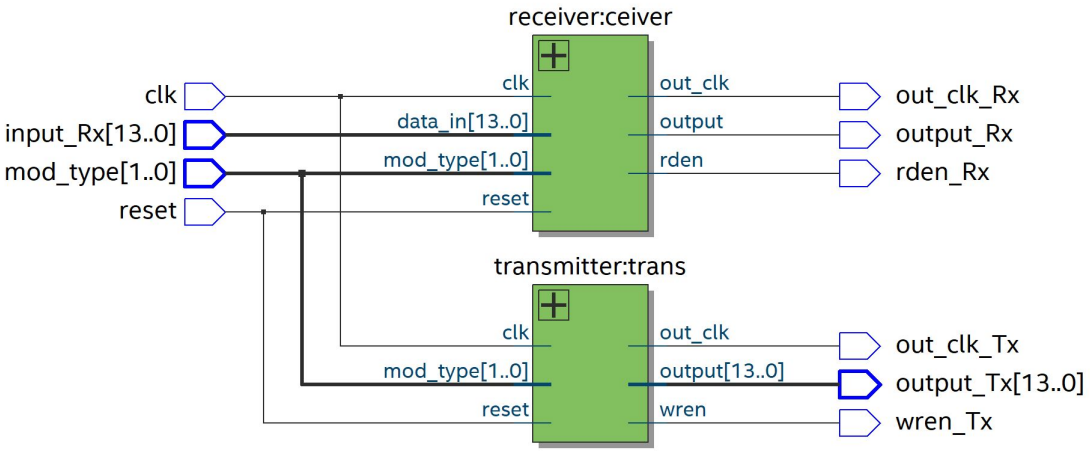


Figure 3.1: VHDL diagram for transceiver

Figure 3.1 shows the top entity of entire subsystem, which includes transmitter and receiver part. A more detailed diagram of transmitter and receiver are presented in the figures below.

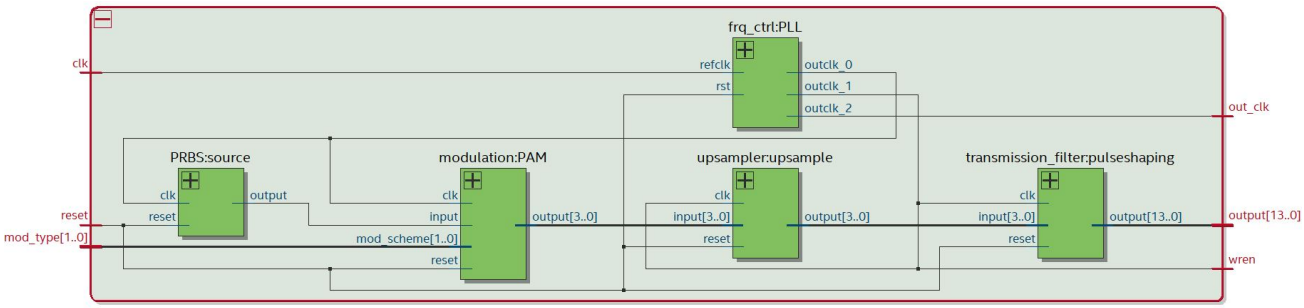


Figure 3.2: VHDL diagram for transmitter

The VHDL diagram for the implemented transmitter is shown in figure 3.2. It is possible to see the transmitter contains the main functionality of the first part of the system, which is shown in the figure 2.2. Figure 3.3 shows the VHDL diagram for the receiver of the system and it contains the main blocks of the receiver, which is the last part of the system.

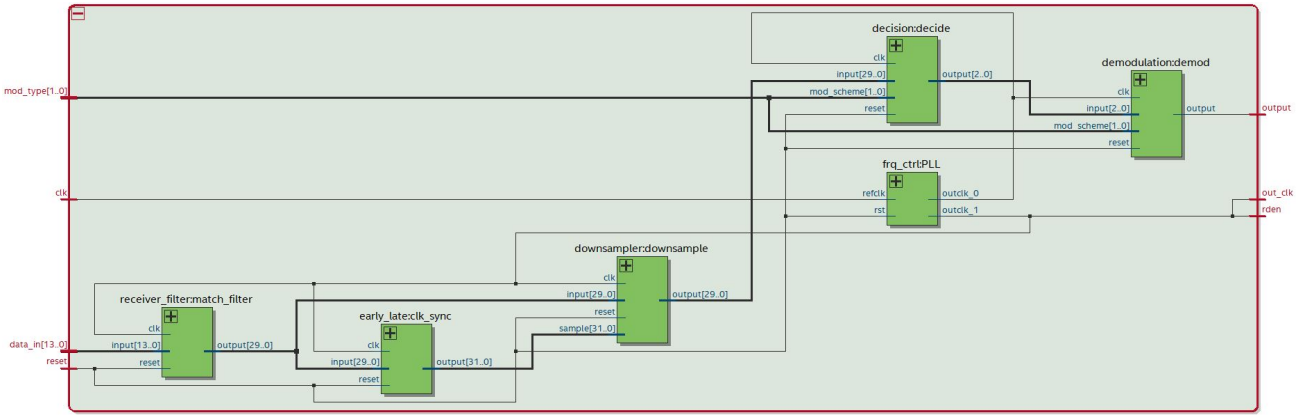


Figure 3.3: VHDL diagram for receiver

3.2.1 Implemenation of PRBS

The structure for implementation of PRBS is illustrated in figure 3.4, where it is possible to see that seven registers are used for generated random bit sequences, since the polynomial that is used to generate the PRBS has an order of 7. This means, PRBS will have a period of 127, which means 127 bits will be generated.

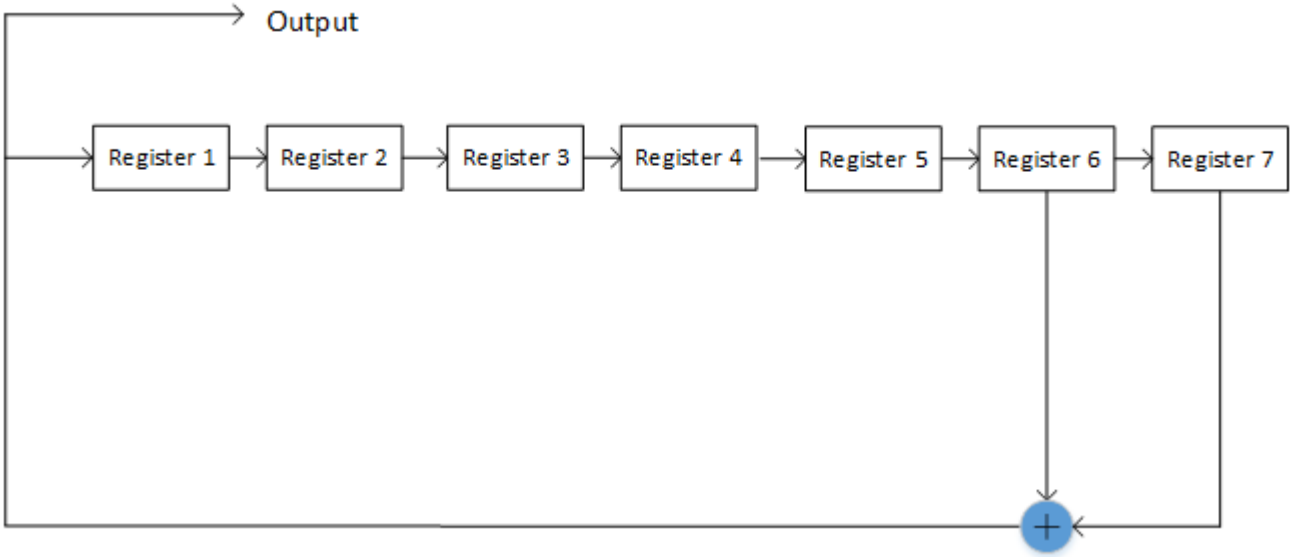


Figure 3.4: Implemenation of PRBS

3.2.2 Implementation of Modulation

The operation of modulations can be described by using state diagrams, which are given on Figures 3.5 and 3.6 for 4-PAM and 8-PAM, respectively. The flow of the state diagrams is controlled by the counter which counts the bits.

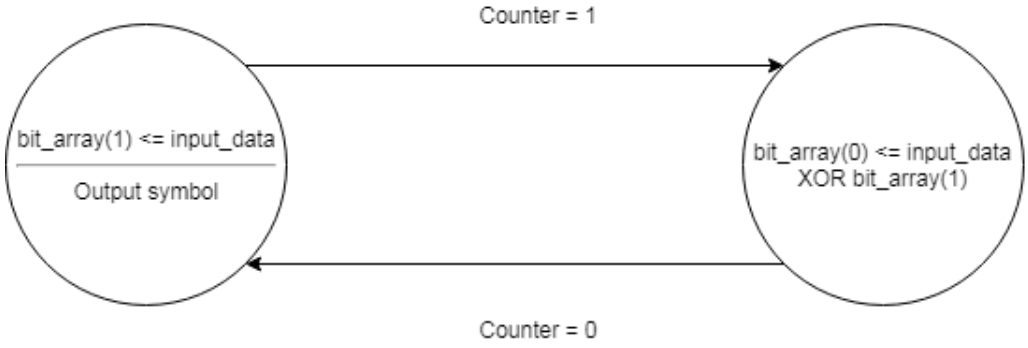


Figure 3.5: State diagram for 4 PAM modulation

In the case of 4-PAM modulation, which is shown in Figure 3.5, the state diagram only has two states. The functionality of each state is to process the information bit and store it into an array, that will be converted into a symbol. As described before, 4PAM has 4 symbols, where each symbol includes two bits. Thus, there are two states in the state diagram to show the process for encoding of bits into symbols for 4PAM. The first state encodes the first bit into the first position in a array. When the counter changes to 1, an event is triggered, where the operation of 4PAM modulation will move to a new state. This new state will encode the next information bit into the last position of the array. In this way, two information bits are encoded and a symbol will be sent to the output, while the new bits are processed. This state is triggered when the counter is 0.

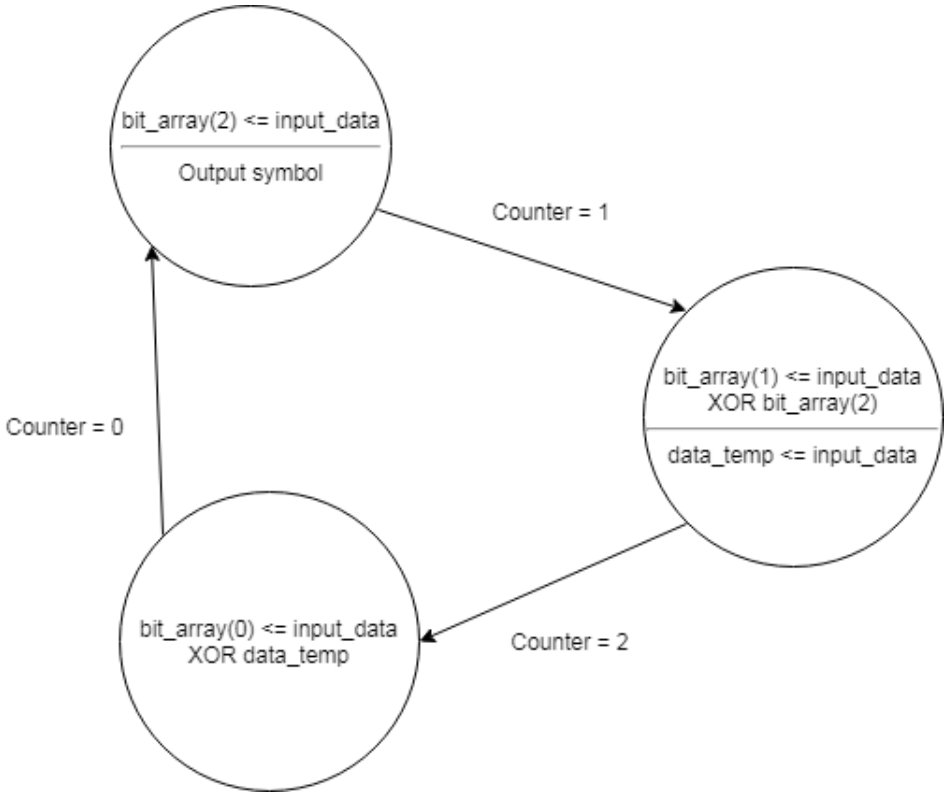


Figure 3.6: State diagram for 8 PAM modulation

While in the case of 8-PAM there are 3 states, each corresponding to one of the bits that will form the symbol which will be the output of the modulation block. The operations done in the states are the same as in the case of 4-PAM. However, three information bits will be encoded into one symbol instead of two, since 8PAM has 8 symbols and each symbol corresponds to 3 information bits. Thus, there are three operation states in 8PAM. Each state encodes the data into the corresponding position in a symbol, which will be the output symbol for 8PAM.

3.2.3 Implementation of upsampler

The upsampler increases the sampling frequency 8 times and between two consecutive symbols inserts 8 zeros, which is called zero-padding.

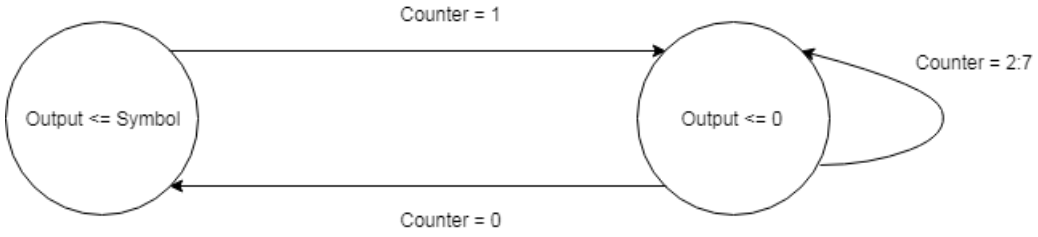


Figure 3.7: State diagram for upsampler

Figure 3.7 shows the behavior of the implemented upsampler, which is represented with a state diagram. A counter is used in the upsampler to control the flow through the states and it is being updated with every new event. This means, when counter changes a transition will take place from the initial state to a new state. In the initial state, the output of the filter is the same as the input symbol. However, the next state will be triggered when the counter turn to 1. The zero-padding process takes place in the next state, where 8 zeros are implemented between the symbols in order to increase the sampling speed. Thus, the upsampler will stay in this state until the counter turn to 7, where zero-padding is completed. The output of the upsampler is presented at the last state, which is triggered when the counter is turn to 0.

3.2.4 Implementation of pulse shaping filter

A finite impulse response filter (FIR) is used in the system, which is shown in Figure 2.2, where transmission and receiver filter are required. The structure of FIR filter is shown in figure 3.8, where the filter is implemented in multiple stages in order to acquire higher working frequency of the design by pipe-lining.

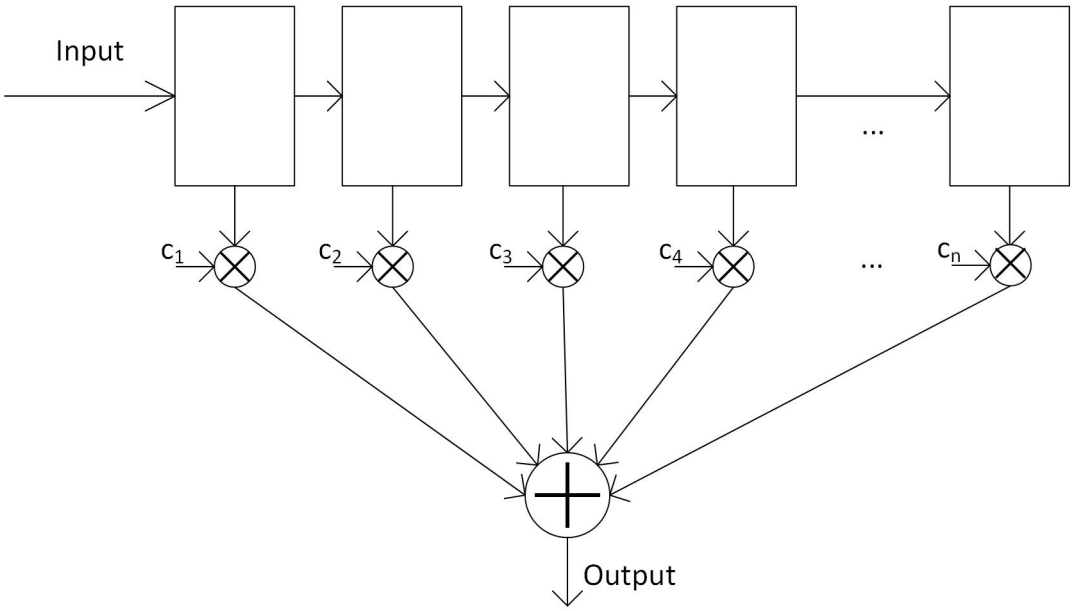


Figure 3.8: Architecture of the implemented FIR filter

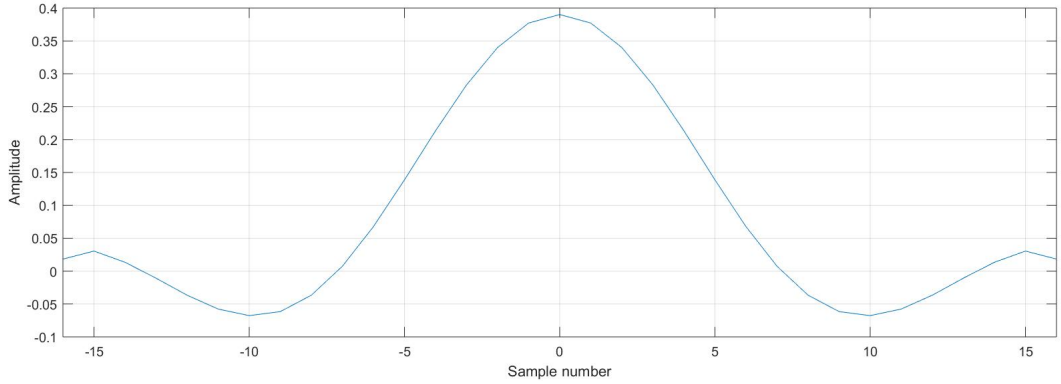


Figure 3.9: Impulse response of the square-root-raised cosine with $\alpha = 0.25$

The impulse response of the filters is shown on Figure 3.9. The coefficients are used to generate the output for FIR filter, which is used to shape the pulse. The values of the coefficients have been represented using 10 bits, where all 10 bits are used for the decimal part. When this filter is being used as a transmission filter the input is of length four and output 14 bits, while in the case of the receiving filter the input is 14 bits and the output 30 bits. In the case of the receiving filter the coefficients are represented using 16 bits, where all of them are used for the decimal part. The 14-bit resolution is dictated by the DAC and ADC for the transmitter and receiver, respectively. The output of the receiving filter is 30 bits in order to have higher resolution and acquire higher precision when it comes to decision making.

3.2.5 Implementation of early late and down-sampler

The early late is implemented as explained in the Section 2.1.5, but with a few modifications that will decrease the complexity of the design. The main challenge when it comes to the

hardware design is to do division, therefore the change was made there. The division is done after every 2048 received symbols because it corresponds to a right shift by 11 bits. The early-late provides the information where the sampling should be done.

The down-sampler decreases the sampling frequency 8 times. It obtains the information from the early-late algorithm on the location of the symbol where the sampling should be done and then sends that symbol to the decision block.

3.2.6 Implementation of decision and demodulation

The decision block receives the noisy sample from the down-sampling block. It performs a hard decision on which symbol is being received. The thresholds on which the decision is based, depend on the modulation scheme that is used. For the thresholds the middle values between two consecutive symbols is taken. Afterwards, the symbol is converted to the proper combination of bits and sent further through the system.

4. Testing

4.1 Test of implementation in Matlab

The simulation that is completed in Matlab is the performance of the modulation, where the bit error rate is analyzed. The elementary quality measurement for a digital telecommunication system is Bit Error Rate (BER). In order to discuss the influence of noise on different modulation schemes, the BER is calculated after the decision on the received symbol is made. The influence of the noise on different modulation schemes will be shown and the performance of each modulation reviewed.

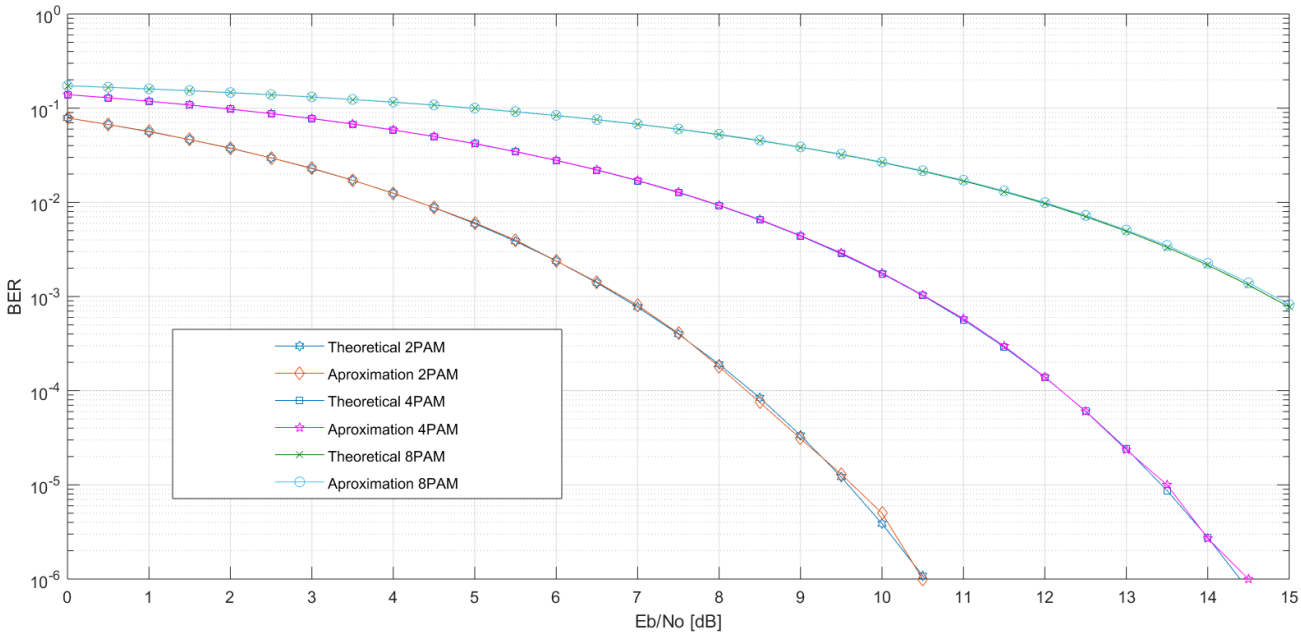


Figure 4.1: Modulation schemes with Gray code

Figure 4.1 shows the performance of 2PAM, 4PAM and 8PAM with Gray coding. Acknowledge that the theoretical values and the numerical approximations are equal, verifying the design. The BER is decreasing, when the E_b/N_0 is increasing. Notice that with the usage of higher modulation scheme, a higher E_b/N_0 level is needed to acquire the same BER. The 2PAM has the best performance in the terms of BER, while to obtain the same BER 4PAM needs up to 4dB more and 8PAM up to 8 dB.

Overall, it is possible to choose an optimal modulation scheme depending on the E_b/N_0 level

to satisfy the BER requirements of the system and improve the spectral efficiency. Further in this project this model will be used as a reference to verify the hardware implementation.

4.2 Testing of implementation in Modelsim

In order to test and verify the design a test bench was made. On Figure 4.3 the waveforms of the transmitter and receiver are presented. On Figure 4.2 the output of the transmitter from Matlab simulation is shown, therefore the waveforms can be compared.

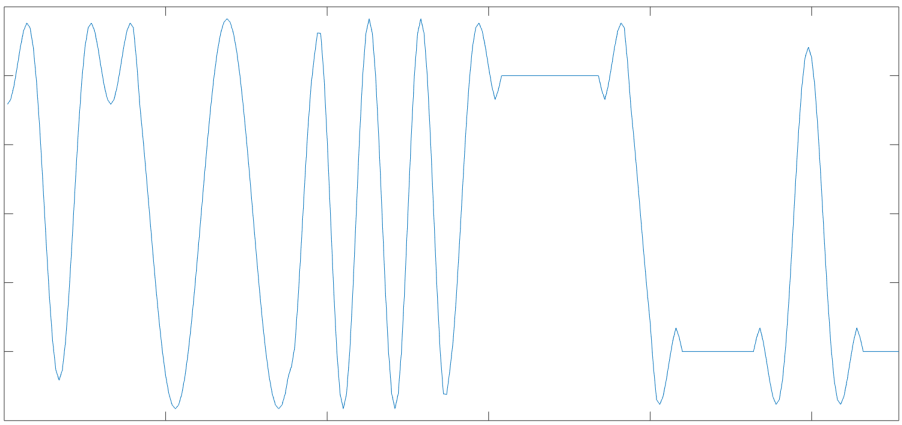


Figure 4.2: Output of the transmitter in Matlab simulation for 2PAM

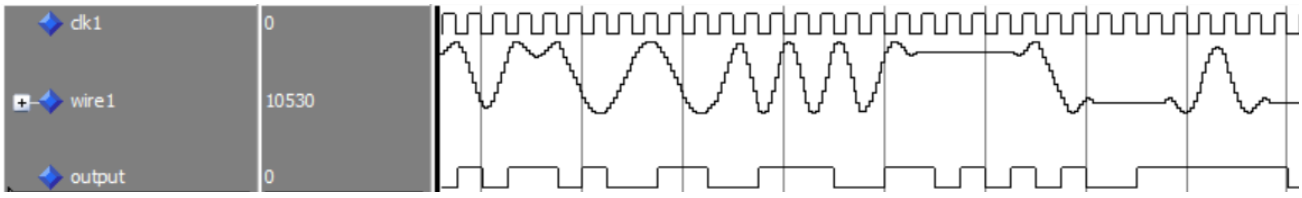


Figure 4.3: Modelsim simulation results for the tranceiver for 2PAM

Signal *wire1* on Figure 4.3 shows the output of the transmitter in Modelsim as an analog signal. Acknowledge that the shape of this signal and the wave presented on Figure 4.2 are very similar. Furthermore, the output of the receiver has been exported into a .txt file and imported into MATLAB and compared to the sent sequence. Figure 4.4 shows the correlation between the two sequences and it can be noticed that the correlation value is 1 when $k = 18$. The correlation is 1 also when k is 145 and 272, because of the properties of the PBRS explained in Section 2.1.1. Since the first peak appears at $k = 18$, it can be concluded that the delay of the system is 18 clock cycles, when the delay of the channel is not taken into consideration. It should be emphasized that there was no noise in these tests.

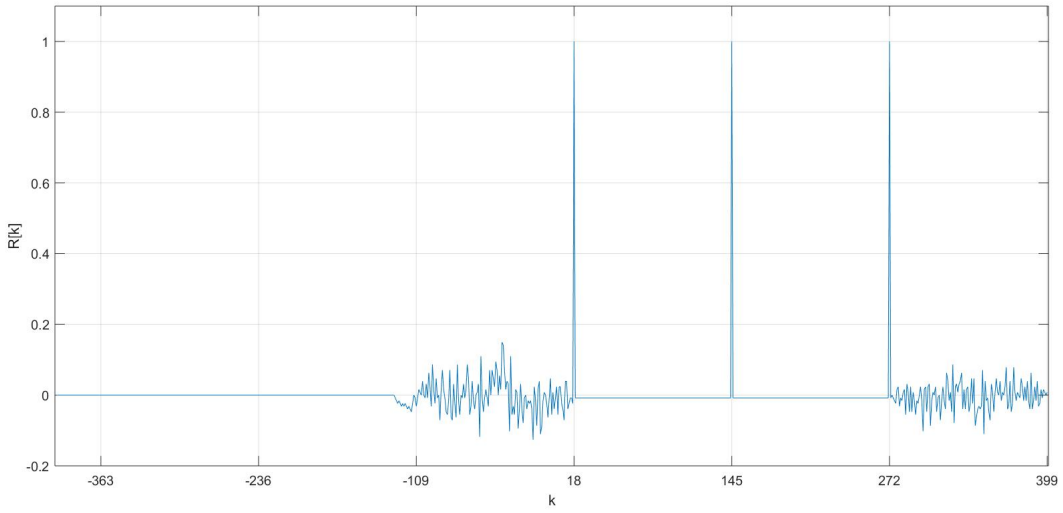


Figure 4.4: Correlation function of the receiver output from Modelsim and sent sequence

4.3 Testing of implementation in hardware

After testing and verifying the design in Modelsim, it should be tested in hardware using an oscilloscope and Signal Tap Logic Analyzer. Before these tests the design needs to be compiled in Quartus to make sure that it can be uploaded to the FPGA board that is used. Compilation report obtained in Quartus is shown on Figure 4.5. From the report it can be seen that the compilation is successful and the design uses only 4% of the logic that is available on the board.

Flow Status	Successful - Mon Jun 18 15:47:17 2018
Quartus Prime Version	17.1.0 Build 590 10/25/2017 SJ Lite Edition
Revision Name	Transceiver
Top-level Entity Name	transceiver
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	1,282 / 32,070 (4 %)
Total registers	3506
Total pins	38 / 457 (8 %)
Total virtual pins	0
Total block memory bits	266,365 / 4,065,280 (7 %)
Total DSP Blocks	32 / 87 (37 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	2 / 6 (33 %)

Figure 4.5: Compilation report for transceiver

	Fmax	Restricted Fmax	Clock Name	Note
1	80.93 MHz	80.93 MHz	altera...ed_tck	

Figure 4.6: Maximum frequency of the transceiver

The results of the TimeQuest Time Analysis have to be checked to verify that the system fulfills the required working frequency stated in Section 2.2. The maximum frequency of the implemented transceiver is 80.93 MHz, which is shown in figure 4.6. It can be concluded that the required frequency is achieved and it should be emphasized that the working frequency of the Framing block can be increased up to 10 MHz.

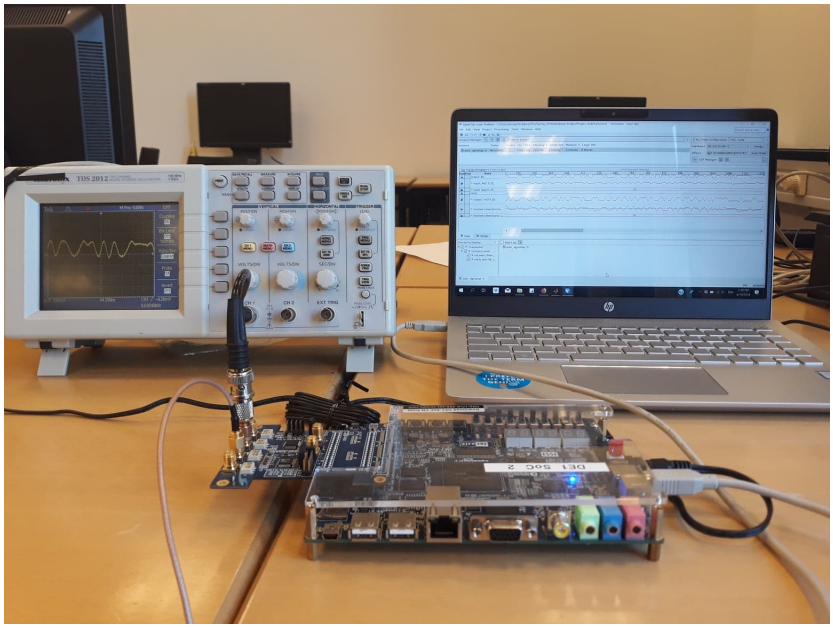


Figure 4.7: Subsystem setup

After a successful synthesis in Quartus the design is implemented in hardware, which is DE1-SoC together with the ADA board THDB-ADA, as described before. The setup for the subsystem is shown in figure 4.7, where an oscilloscope will be used to analyze the output of the transmitter. Furthermore, the received signal is observed by connecting a SMA cable from the transmitter port to the receiver port. In this way, it is possible to see whether the received signal is correct or not.

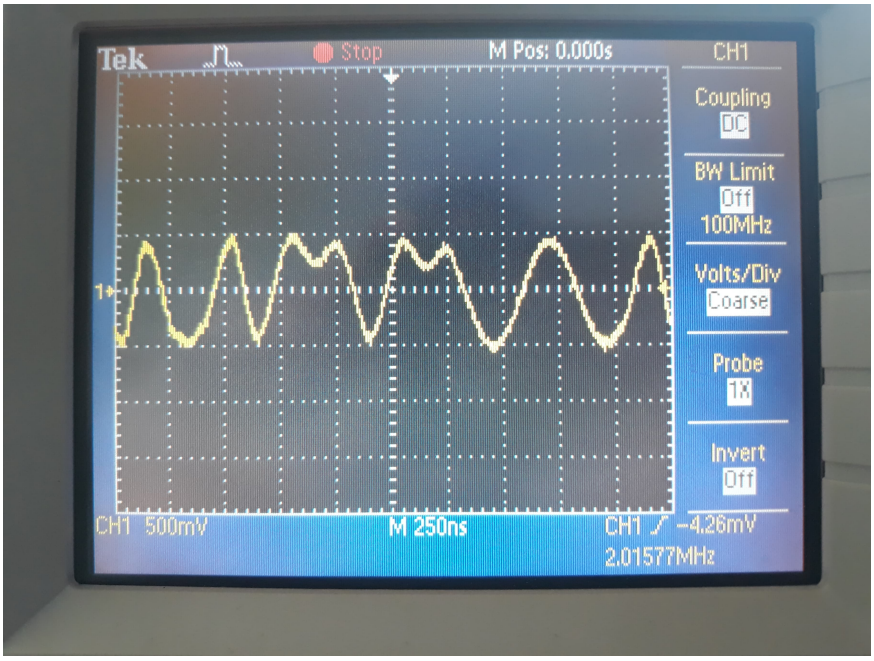


Figure 4.8: Output signal of transmitter

Signal that is shown in Figure 4.8, is the output of implemented transmitter on hardware, where the signal for 2PAM is shown. Furthermore, the observed signal on oscilloscope is similar to the signal that is obtained from MATLAB and Modelsim on Figures 4.2 and 4.3, respectively. Thus, it is possible to confirm that the design of transmitter is implemented correctly in the hardware.

The verification of implemented receiver on hardware is completed by connecting the transmitter to the receiver on THDB-ADA board with a SMA cable. As it is shown in Figure 4.9.

The received signal is compared with transmitted signal in signaltap. In this way, it is possible to verify the received signal whether its the same as transmitted signal or not.

Figure 4.10 shows the transmission of signal from the transmitter to receiver. It is possible to see the received signal (receiver:ceiver) is similar to the transmitted signal (output_Tx) with a slightly delay due to the SMA cable and implementation of filter. The implemented receiver on hardware is verified by using signaltap. In order to make sure that the received signal is corresponding to the transmitted signal, the received signal is exported to MATLAB and correlation was performed.

Figure 4.11 shows correlation of the received signal and the signal that is desired to be transmitted. It is possible to see, that the correlation value is equal to one, which means the received signal is equal with the transmitted signal. The distance between the correlation peaks that have value one is 127, which shows the property of PRBS sequence. Since this is a real time system from the correlation function, it is difficult to conclude the size of delay.

It should be emphasized that only the implementation of the 2PAM modulation was completely successful, while the other modulations could not be tested, due to problems in the clock multiplexer.

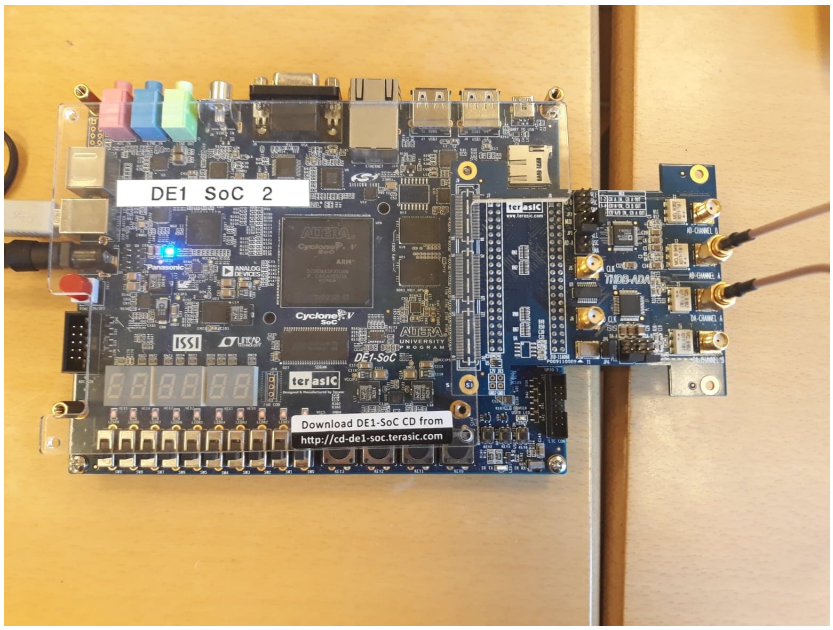


Figure 4.9: DE1-SoC with THDB-ADA

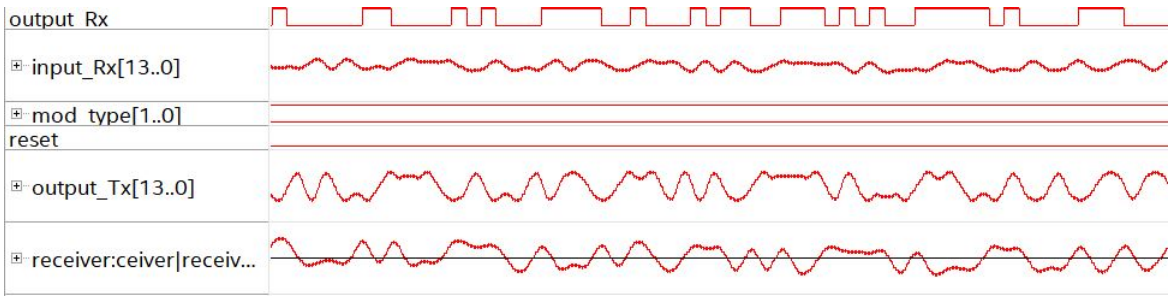


Figure 4.10: Signaltap for transceiver

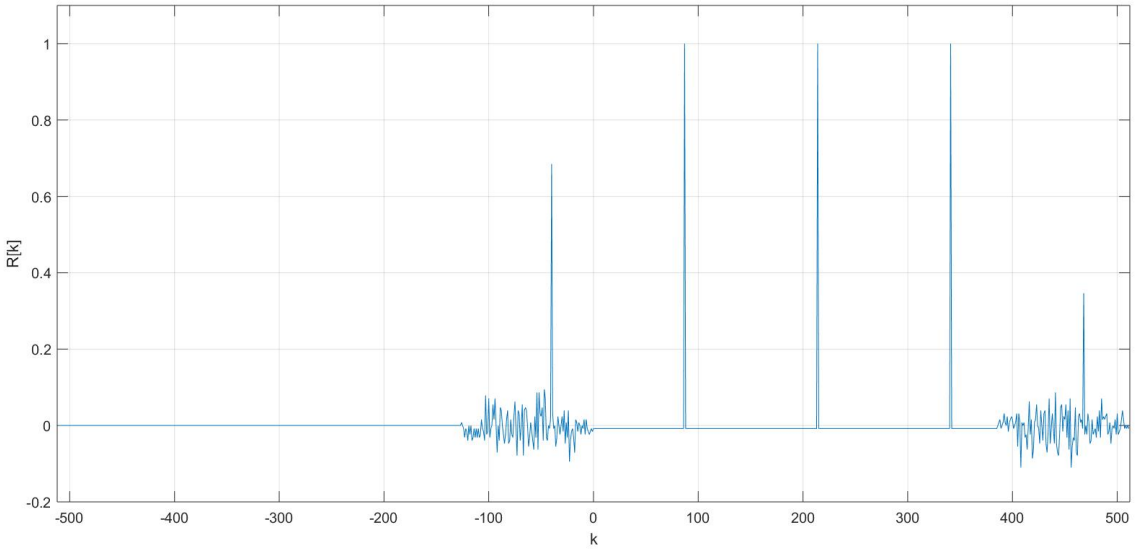


Figure 4.11: Correlation function of the receiver output from signal tap and sent sequence

5. Conclusion

The purpose of this project is hardware implementation of a subsystem, which performs modulation, demodulation for 2,4 and 8PAM modulation schemes and pulse shaping, in order to transmit a binary sequence through a medium. A simulation model was also developed, so that the hardware implementation can be verified.

Analysis of the simulation model performance was based on conducted simulations and calculated BER for different SNR level. Overall, the review of quality is given by the estimated BER, which is a key parameter for the measurement of the quality of the system. While testing of the hardware implementation was done in few steps, using different testing methods and comparing them to the corresponding part of the simulation model.

From the results of the analysis for different modulations, it is concluded that the rising of SNR leads to decreasing of BER. As well as using higher order modulation scheme, the required SNR for the same value of BER is greater.

The first test of the hardware implementation was performed in Modelsim, where the output of the design was exported into MATLAB and compared to the simulation model and verified. Secondly, the design was tested in Quartus and on the behalf of the compilation report, it is synthesizable and can achieve a maximum working frequency of 80.93 MHz, which fulfills the requirements of the overall system. Furthermore, the design was loaded on to the FPGA board and the output of the transmitter connected to the oscilloscope where it was seen that waveforms that are transmitted are similar to the ones that are seen in the signal tap. Finally, the last test was done in signal tap where the received and demodulated signal was exported into MATLAB and verified. The system works completely only for 2PAM modulation scheme, therefore can achieve a maximum baud rate of 1 Mbaud/s .

Taking all the results into account, it is possible to find the most suitable modulation scheme for the system, which is depending on the tradeoff of SNR with the required BER and the spectral efficiency. Furthermore, it was possible to perform an emulation of the entire system when it uses 2PAM and verify that it works as well as satisfy the frequency requirements.

References

[1] Govind P. Agrawal. *Fiber-Optic Communication Systems: Fourth Edition*. 2011.

[2] Knud J. Larsen & Jørn Justesen. Digital communication 34230. 2017.

[3] Donald Ervin Knuth. *The art of computer programming, vlolume 4A. Enumeration and backtracking.*

Table 5.1: Work distribution

Sections	Sai Lou	Ognjen Jovanovic
Abstract		x
1	x	
2	x	
2.1	x	
2.1.1	x	
2.1.2	x	
2.1.3		x
2.1.4	x	
2.1.5		x
2.2		x
3	x	x
3.1		x
3.2	x	
3.2.1	x	
3.2.2	x	
3.2.3	x	
3.2.4		x
3.2.5		x
3.2.6		x
4		x
4.1		x
4.2		x
4.3		x
5	x	x