Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı 39, S. 26-32, Temmuz 2022

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue 39, pp. 26-32, July 2022

Research Article

Design and Implementation of an Educational FM Transmitter with FPGA Using SDR Techniques

Ahmed Ibrahim M Alghhriuni[1*], Bilgehan Erkal[2]

[1*] Ahmed Alghhriuni University, Faculty of Engineering, Departmant of Electrical and Electronic Engineering, Karabuk, Turkey, (ORCID: 0000-0002-9893-0104), gheriani94@gmail.com

[2] Bilgehan Erkal University, Faculty of Engineering, Departmant of Electrical and Electronic Engineering, Karabuk, Turkey, (ORCID: 0000-0002-1405-6932), berkal@karabuk.edu.tr

(3rd International Conference on Applied Engineering and Natural Sciences ICAENS 2022, Jul 20-23, 2022)

(**DOI:** 10.31590/ejosat.1141421)

ATIF/REFERENCE: Alghhriuni, A. & Erkal, B. (2022). Design and Implementation of an Eductional FM Transmitter with FPGA using SDR Techniques. *European Journal of Science and Technology*, (39), 26-32.

Abstract

The basic purpose of Software Defined Radio (SDR) systems is to use a digital signal processor to numerically handle radio signals. The use of a processor like a Field Programmable Gate Array (FPGA) to perform tasks like modulation, demodulation, signal creation, and line coding on these systems dramatically decreases the demand for analog circuit-based hardware. FPGAs are digital integrated circuits with a wide range of applications which are made up of links between programmable logic blocks. It's aimed at assisting the creator realize the logic functions that the designer needs. In consequence, the user may change the function of each logic block. VHSIC Hardware Description Language (VHDL) is commonly used in FPGA programming. In this study, VHDL code was created, and a FM transmitter was implemented on a FPGA board (CMOD A7) in this research. The sound card interface on the PC used to send and receive the signals while suitable ADC and DAC cards are used on the FPGA side for the same purpose. Audacity program was used to playback the sample wav files while HDSDR SDR program was used to monitor and record the signals in wav file format. Finally, using the MATLAB code, the recorded transmitter signal was demodulated offline, and the output was stored to the hard drive. The demodulated signal obtained is identical to the initial modulation signal, indicating that the modulation was correctly executed. As a result, a perfect foundation for the development and training of SDR systems using FPGA has been established.

Keywords: SDR, FPGA, MATLAB, FM, TX.

Eğitsel SDR Tekniklerine Dayalı FPGA Tabanlı FM Radyo Verici Tasarım ve Uygulaması

Öz

Yazılım Tanımlı Radyo (SDR) sistemlerinin temel amacı, radyo sinyallerini sayısal olarak işlemek için bir dijital sinyal işlemcisi kullanmaktır. Bu sistemlerde modülasyon, demodülasyon, sinyal oluşturma ve hat kodlama gibi görevleri gerçekleştirmek için Alan Programlanabilir Kapı Dizisi (FPGA) gibi bir işlemcinin kullanılması, analog devre tabanlı donanıma olan talebi önemli ölçüde azaltır. FPGA'lar, programlanabilir mantık blokları arasındaki bağlantılardan oluşan çok çeşitli uygulamalara sahip dijital entegre devrelerdir. Yaratıcının, tasarımcının ihtiyaç duyduğu mantık işlevlerini gerçekleştirmesine yardımcı olmayı amaçlar. Sonuç olarak, kullanıcı her bir mantık bloğunun işlevini değiştirebilir. VHSIC Donanım Tanımlama Dili (VHDL), FPGA programlamada yaygın olarak kullanılır. Bu çalışmada VHDL kodu oluşturulmuş ve bu araştırmada bir FPGA kartına (CMOD A7) bir FM vericisi uygulanmıştır. Sinyalleri göndermek ve almak için PC üzerindeki ses kartı arayüzü kullanılırken, FPGA tarafında ise aynı amaç için uygun ADC ve DAC kartları kullanılmaktadır. Örnek wav dosyalarını oynatmak için Audacity programı, sinyalleri izlemek ve wav dosya formatında kaydetmek için HDSDR SDR programı kullanıldı. Son olarak, MATLAB kodu kullanılarak, kaydedilen verici sinyali çevrimdışı olarak demodüle edildi ve çıktı, sabit sürücüye depolandı. Elde edilen demodüle edilmiş sinyal, modülasyonun doğru bir şekilde yürütüldüğünü gösteren ilk modülasyon sinyaliyle aynıdır. Sonuç olarak, FPGA kullanan SDR sistemlerinin geliştirilmesi ve eğitimi için mükemmel bir temel oluşturulmuştur.

Anahtar Kelimeler: SDR, FPGA, MATLAB, FM, TX.

[*] Corresponding Author: gheriani94@gmail.com

1. Introduction

Software Defined Radio (SDR) is a combination of programmable hardware and software technologies developed for wireless communication. John Mittola proposed the first concepts for software defined radio in 1991, with the idea that radios may be set and programmed in software. Other hardware devices perform actions such as signal lowering/amplifying, modulation/demodulation, and filtering in a traditional hardware radio system. In the software radio, on the other hand, there is a programmable system that the user can change the location of these elements at any time. The development of SDR systems has offered benefits such as cheaper costs and modifying the functionality of hardware-based radios [1-4].

FM radios are commonly used to broadcast audio signals. They are also available for limited bandwidth digital communication systems that need lower receiver sensitivity [5].

Instead of classic analog modems, software defined modems are gaining popularity due to several major advantages such as re-programmability, flexibility, and low cost [6].

FM implementation with analog circuits has various drawbacks, such as non-linearity due to voltage-controlled oscillator (VCO) and stability performance [7]. The expansion of low-cost digital signal processing integrated circuits has gained substantial relevance for the design of digital FM because of recent improvements. They additionally feature exceptional noise figure performance and outstanding voice clarity.

FPGAs can realize digital Numerically Controlled Oscillators (NCOs) and high order filters for software defined FM modulation and demodulation.

Two observations may be made based on the FM output signal. The amplitude of the FM signal remains constant regardless of the message signal, resulting in a constant envelope feature. Furthermore, the frequency-modulated output is based nonlinearly on the message signal. As a result, the FM signal qualities are difficult to examine. The FM signal bandwidth, on the other hand, may be approximated using a tone message signal that represents the number of efficient sidebands. A message signal can be extracted from an FM transmission via frequency demodulation. It has a frequency discriminator that works as a differentiator with a specialized envelope detector [7-8].

The first wireless communication was found in the late 1980s, and since then, various breakthroughs in radio communication technology have evolved to ensure radio users' connectivity. The Triumphant radio, which was created in the 1930s and employed voice communication because of bandwidth limitations at the time, is the earliest sort of transmission. Then, in the 1950s, broadcast communication became mainstream, with analog television communication using a large amount of bandwidth and providing exceptional customer service. In the 1960s, computers were more widely used, and they were able to transport data across large distances through cable and wireless connections. Following the introduction of cell phones, wireless voice communications were found, allowing transmission from any location. Nonetheless, because the mobile gadgets were not portable, they were difficult to utilize [9].

Ali HANDER designed an AM receiver using SDR methods and implemented it on an FPGA. The study's goal was to create a simple and inexpensive FPGA-based platform for teaching the fundamentals of SDR. To create the simulation environment, the researchers employed MATLAB programs. A signal was utilized in the simulation to evaluate the FPGA implementation. In addition, the simulation code serves as a framework for the FPGA-based SDR system's VHDL architecture. Another MATLAB script was written by the researcher to examine the simulation and test findings and compare them. Because the greater the SNR ratio, the better, the test results on the two signals revealed that A1 tests signals are better than A2 tests signals. When the actual real-world values were compared to the simulations of each test signal, it was discovered that the real-world SNR findings were somewhat lower than the simulations SNR. SNR value greater than 20dB is regarded as an acceptable level for an AM receiver. The FPGA AM RX system has shown to be a promising contender for AM demodulation and reception based on test and simulation findings. Furthermore, the planned and implemented FPGA AM RX was effective in teaching the fundamentals of basic SDR, which was the study's major focus [10].

In [11] authors utilized MATLAB algorithms to build and implement an AM radio transmitter simulation in an FPGA. Later, using ISE Design Suite 14.7, VHDL code was written and an amplitude modulated transmitter was constructed on the FPGA board (Mimas Spartan 6). The sound card was utilized to send the sample sound that was used in modulation using the Audacity programs to the FPGA card. The ADC (LM4550) card provides an analog signal to the FPGA card, which is then received, demodulated, and recorded using the HDSDR application. The transmitter signal is created in analog form by the FPGA card and delivered to the microphone input of the laptop's sound card through the DAC (LM4550) card. Finally, the recorded transmitter is demodulated offline using the MATLAB code, and the output is stored to the hard disk. The findings of the investigation revealed that there is only a little variation between the simulation and real test results for the same test signal. Because the signal in the genuine test result has been subjected to noise, this is seen to be extremely plausible. Similarly, according to the SNR values obtained, the average value is approximately 20dB, which can be regarded as an acceptable value for an AM receiver. Furthermore, the system is thought to be an excellent platform for implementing and training FPGA SDR systems [11].

Hikmat N. Abdullah created a design approach and the implementation suggested an SDR system using an Altera Cyclone II family board, as well as Embedded MATLAB blocks and MATLAB/Simulink. The design was originally implemented in the MATLAB/Simulink environment, and then translated to VHDL using the Simulink HDL coder. The design has been synthesized and loaded onto an Altera Cyclone II FPGA board using Quartus II 9.0 Web Edition® software. The findings of the study revealed that using programmable logic tools, the implementation of SDR may be readily produced and understood. In addition, the research revealed an efficient design method for obtaining VHDL netlists that may be downloaded to FPGA boards [12].

2. Material and Method

In this study, Digilent CMODA7 “FPGA Card”, PMOD ADC and DAC cards are used as hardware. In terms of software, XILINX VIVADO design suite is used to perform VHDL encoding, Audacity is used for editing and playback of the audio files, HDSDR is used to control and display the input/output signals,

and MATLAB is used to demodulate the obtained transmitter signal [13-19].

A. Simulation Studies

Simulation of the FMTX system is done using suitable MATLAB scripts. Firstly, simulation of the FM TX is realized. Then FM receiver simulation is done. Lastly, analysis of the results are done.

The transmitter code uses a 10s sample recording sampled at 8KSps which is a 4KHz music recorded in wav format. This sample wav file is up sampled to 48KSps to provide a 24KHz wide frequency modulated intermediate frequency (IF) waveform whose centre frequency is 12KHz. This IF signal is then normalized and recorded as a wav file (FM.wav). The modulating signal is normalized and integrated before modulation. A modulation coefficient determines the maximum frequency deviation which sets the FM bandwidth according to Carson's rule. Maximum frequency deviation Fdmax is derived using the formula below:

$$F_{dmax} = \frac{A_{max}}{2.\pi.T_s}$$

Here,

$Fdmax$: Maximum Frequency Deviation (Hz),

$Amax$: Maximum Amplitude (rad),

Ts: Sampling Period (s),

$Amax$ given in the code sets $Fdmax$ as 4KHz. The integral of the signal is plugged into the carrier signal phase argument. Thus, an indirectly frequency modulated signal is derived through phase modulation which is a frequently used method in deriving FM in sampled systems. The modulated carrier is filtered through a bandpass filter whose centre frequency is 12KHz and bandwidth is + 8KHz. The bandwidth is selected as 16KHz because Carson's rule gives us so as below equation suggests:

$$BW_{FM} = 2.(F_{dmax} + BW_m) = 2.(4KHz + 4KHz) = 16KHz$$

So, the modulation index β, of the frequency modulated carrier is:

$$\beta = \frac{F_{dmax}}{BW_m} = \frac{4KHz}{4KHz} = 1$$

Since β is equal to 1, the FM signal is said to be wideband. The modulated carrier, FM IF, being an infinite bandwidth signal in analogue form is further bandpass filtered through a FIR filter in the code which limits signal to 4-20KHz range whose bandwidth is now limited. Since it has 98% of its energy reside in the Carson's bandwidth it continues to represent the modulating signal. Carson's bandwidth is sufficient for a successful demodulation and reproduction of the original modulating signal. This filtering is mandatory to eliminate residual spectral components by bandpass filtering before transmitting to prevent interference to neighbouring stations.

So, a 16KHz wideband FM IF is obtained which is normalized and recorded as a wav file for further use in simulations and tests. In the simulations and tests two different music recordings are used which will be called as A1 and A2 and their FM results are FM1 and FM2 respectively.

To test the transmitter, receiver simulation is done through the receiver code. Firstly, FM.wav file at 48KSps is loaded. It is normalized and then sent for demodulation. Demodulation method used is quadrature FM demodulation. It is achieved by delaying the input signal by one sample and then multiplying itself. Selection of 12KHz as IF signal centre frequency is not arbitrary. Phase interval between two samples corresponds exactly 90 degrees for an unmodulated 12KHz carrier sampled at 48KSps. So, one sample delay represents 90 degrees phase delay. If we multiply delayed signal by non-delayed signal, we get zero DC level other than a high frequency component which is filtered after demodulation. This process provides a changing level if instantaneous frequency of the carrier is slightly changed by time. The output signal level and polarity are directly proportional to the direction and amount of frequency change so that we get demodulated signal at the end of the process. The filter at the output is a lowpass FIR filter whose cut-off frequency is set to 4KHz which is compatible with the bandwidth of the modulating signal. The reproduced signal is normalized and recorded as a wav file for analysis.

Analysis of the results provides us a comparison between original test signal with the signal resulted from demodulation. It is very important to time synchronize each signal that will be compared to obtain consistent results. Gain errors are also corrected using a suitable amplitude scaling. It is only possible to see the distortion and noise effects of demodulation over the signal after this pre-processing. Pre-processing of test results is done using the software tool called Audacity which is an easy to use and free audio processing software (Figure 1).

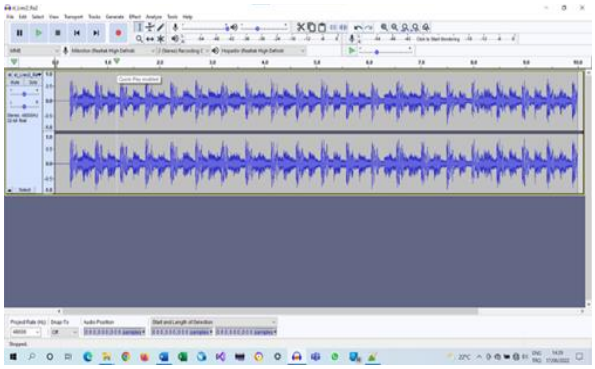

Fig. 1 Audacity audio processing tool

The analysis code uses a dual channel wav file. This input wav file has result signal in the left channel while the original signal is in the right channel. The recording is prepared in the Audacity program. Both signals are added to the project file as stereo. Then both signals are normalized to same level (-1dB). As a last step, result signal is time synchronized to the original by setting a zero crossing as reference which is at the same time point. The synchronization is achieved by discarding enough samples at the beginning of the result signal. The unused parts at the end of the signals are also discarded to set the record length to ten seconds. The test result recordings usually lasts longer than ten seconds to ensure that one full ten second signal is captured in the recording. Recordings are made using the SDR software HDSDR. HDSDR program is also useful in visual and spectral monitoring of the result in real time. HDSDR SDR program in operation is shown in Figure 2.

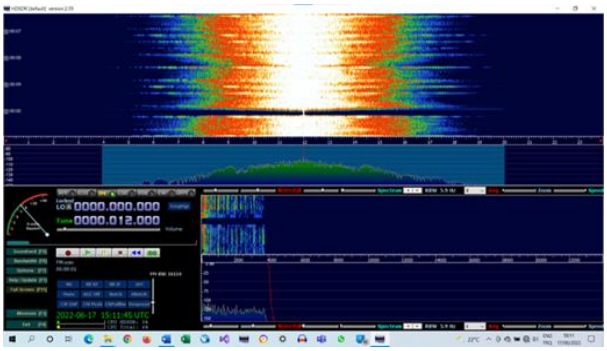

Fig. 2 HDSDR SDR software

After analysis code takes this prepared dual channel recording as input, the channels are separated, and then a suitable scaling factor is applied to correct for gain errors. The scaling factor is determined by trial and error. An optimal value should maximize the Signal-to-Noise ratio (SNR) which is calculated and provided at the end of the code execution. SNR in dB is calculated by dividing rms original signal level to rms error signal level and then this ratio is converted to dB. Error is calculated by taking sample by sample difference of original and result signals. The rms level then calculated by squaring and adding each sample and then taking the square root of the average.

B. Implementation of FM Transmitter in FPGA

The implementation of the FM transmitter in FPGA has two stages: Hardware design and software design in VHDL.

1. Hardware Design:

Hardware design part of the study incorporates PC, FPGA card, ADC and DAC cards. A block diagram of the system is presented in figure 3.

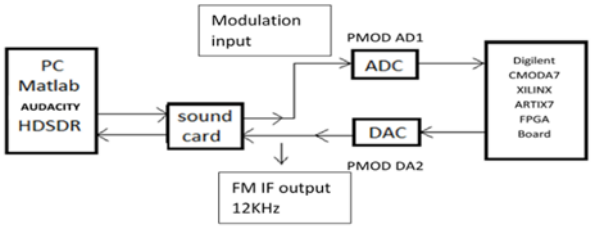

Fig. 3 FMTX System Hardware Block Diagram

FPGA module in the system is Digilent CMODA7-35T which incorporates Xilinx Artix7 FPGA on it. The FPGA chip is XC7A35T in 1CPG236C package whose capacity is 20K-LUT with 225KB Block-RAM. The board has 512KB SRAM with 8-bit bus and 8ns access time, 4MB Quad-SPI Flash to hold FPGA programs and USB-JTAG programming facility which also supplies necessary power from the connected USB bus. The board is in DIP form where total of 48-pins provided at each side. The board provides 44 Digital GPIO pins with 3.3V logic capability. All the connection to the board is provided through a solderless breadboard.

ADC card is Digilent's PMOD-AD1 which uses a dual channel 12-bit, 1MSps/channel sampling rate A/D converter chip AD7476 from Analog Devices. DAC card is Digilent's PMOD-DA2 which incorporates two 12-bit, 1MSps DAC chip DAC121S101 from Texas Instruments. The interface to both cards is through a standardized PMOD connector which encapsulates a standard multi-channel Serial Peripheral Interface (SPI).

There are also an external USB-soundcard and connecting audio cables to carry analog test signals between PC and FMTX system where built-in soundcard of the PC is reserved for listening of the results.

The recorded test signal (modulation signal) is played back through the soundcard speaker output at a 48KSps rate repeatedly using Audacity. This analog signal is digitized through the ADC and then sent to FPGA board for processing. After processing and frequency modulating the signal by the internal structure of the FPGA, the digital FM IF output is sent to D/A card for conversion to analog at 48KSps rate. The internal structure of FMTX system is constructed through programming by the VHDL code whose details are given in the software development part. This analog output signal is then taken through the microphone input of the soundcard and sent to PC for monitoring and recording. Demodulation, monitoring and recording of the modulated signal is through HDSDR SDR software environment on the PC. A photo of the FMTX system is shown on Figure 4.

Fig. 4 Photo of the FMTX system in operation

2. Software Design:

Software for the FMTX system on FPGA is developed under the XILINX VIVADO integrated development environment (IDE). It is the standard development environment for XILINX Artix-7 series FPGAs. The language used is VHDL which is a standard language for implementation of hardware logic circuits in FPGAs. The top module code FMTX provides a main body for the other functional modules. The diagram on Figure 5 shows the functional structure of the FM transmitter.

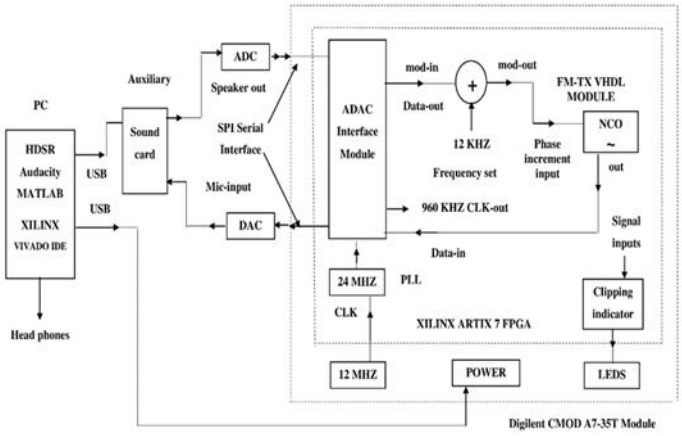

Fig. 5 Block diagram of the FMTX system on the FPGA

PC connection for programming of the module is provided through a spare USB port. This USB port also provides power to the module. Signal flow to the FPGA is analogue through an external USB soundcard interface. This external soundcard provides modulating signal through the speaker output and inputs frequency modulated IF signal through microphone input. Both input and output are single channel (mono) ports with a sampling rate of 48KSps. The input to the FPGA is through the PMOD-AD1 A/D module. The digital output from this module is through a high-speed serial data link in SPI format. The clock provided by the FPGA is 24MHz which is an integer multiple of sampling rate (960KSps). So, 12-bit samples are provided at a rate 960KSps by the ADC. This being an integer multiple of the sampling rate of the soundcard interface at 48KSps, is a very high rate than the required signal bandwidth of 24KHz. However, it does not make any harm because the processed signals remain in the 24KHz bandwidth limit. Such a high sampling rate as 960KSps is necessary because stable operation of ADC and DAC cards used can only be possible at such high rates.

The output from the FPGA is also provided in serial digital data and converted to analog form by the PMOD-DA2 D/A module. The interface is again SPI where the clock rate is 24MHz which must be in conformance to the A/D converter and sampling rate of the FPGA. This clock frequency is derived from an onboard crystal clock module running at 12MHz by a Digital Clock Management (DCM) IP module. The other necessary clocks are derived from 24MHz master clock using suitable divider module which ensure synchronicity through all the FPGA fabric.

Handling of the data acquisition to and from DAC and ADC modules is carried over by the ADAC module. It provides the data to FM modulator and sends the modulator output to the DAC. It also derives the necessary sampling clock of 960KHz which is used by the modulator.

Frequency modulator is a simple direct frequency modulator. A Numerically Controlled Oscillator (NCO) is used as a VCO in this case. Phase increment value is controlled by the digital input samples. A higher phase step means an increase in the instantaneous frequency of the NCO. So, changing phase increment input of the NCO causes a frequency modulation at the output. The output of the ADC is first normalized by adjusting the bit length and then factorized and put as a modulator input. Second input to the modulator, which is a simple 32-bit adder, is an offset value which is factorized to give out a frequency offset adjusted to 12-KHz. So, without a modulation input NCO provides a 12-KHz smooth sinusoid. The phase increment input of the NCO is 32-bit while the output is 16-bits. 32-bit is standard to obtain a suitable frequency resolution in NCOs. The NCO is operated at 960KHz, so the input and output sample rates are also 960KSps. The frequency output of the NCO, fo is calculated using the equation below:

$$f_o = \frac{f_{clk} . \Delta\theta}{2^{B\theta(n)}}$$

And for calculating the phase increment value ($\Delta\theta$) necessary to generate an output frequency is:

$$\Delta\theta = \frac{2^{B_{\theta(n)}} . f_o}{f_{clk}}$$

Here,

fo: Output frequency in Hz,

$B\theta(n)$: Phase increment bit length in number of bits,

$fclk$: Clock speed in Hz,

$\Delta\theta$: Phase increment value.

So, for 32-bit phase increment register, 960 KHz clock rate and 12-KHz output the phase increment value must be 0x03333333. And for the same parameters the frequency resolution will be 0.0002235174Hz which is a very small value and hence the error in the actual frequency output will be very low.

Another useful feature of the FMTX system implemented in the FPGA is the clipping indicator module. This module uses two onboard LEDs for the purpose of monitoring the output level. If output or input of any module is overloaded (the level crosses a determined threshold) the corresponding led is lit for approximately 1 seconds. One second time delay is necessary to see even a one-time event since human eye cannot follow an event that lasts only 1/960000 of a second. Use of this feature ensures that the digital output of the ADC and digital input of DAC is not saturated which leads to clipping distortion. Saturating input of any module in the signal chain can lead to unexpected results and hard to determine faults.

3. Results and Discussion

The experiments in the study are obtained in two steps: simulation results and test results. Simulation results come from ideal simulation efforts using MATLAB codes. Test results come from as recordings from actual operational tests of FMTX system implemented on FPGA.

Two modulation test signals used in each stage are A1 and A2. Each test signal lasts 10 seconds and sampled at a rate 48KSps. Results from simulation and tests, which are demodulated audio, are recorded as a separate wave file. These results are also 48KSps wave files. A post-processing is applied to these results under audio processing program Audacity. These post-processes are mainly normalization and synchronization processes which after demodulation results and original modulating signals are combined into a single stereo (2-channel) recording which lasts exactly 10 seconds. These recordings hold demodulation product on left channel (upper signal in the stereo track), while original is held on right channel (bottom signal in the stereo track). Figures 6 and 7 show the simulation results for the A1, and A2 test signals respectively. Results for the test signals A1 and A2 are provided on Figures 8 and 9 respectively.

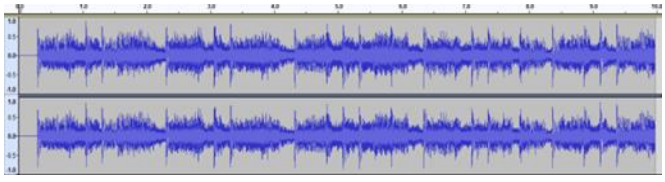

Fig. 6 Simulation Result for A1

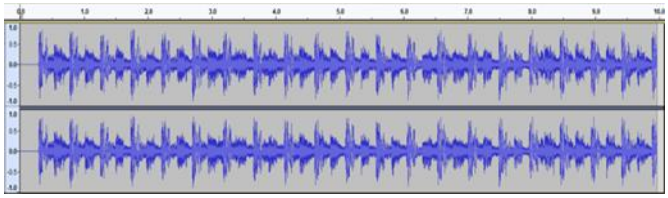

Fig. 7 Simulation Result for A2

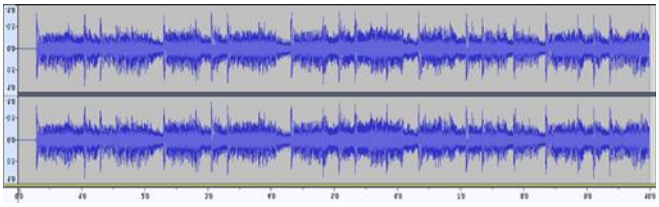

Fig. 8 Test Result for A1

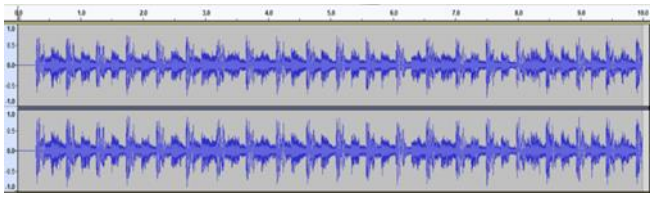

Fig. 9 Test Result for A2

The analysis code compares two signals by subtracting from each other. From this difference (error) signal rms error is calculated. Then S/N ratio is calculated in dB using rms error and rms original signal level whose definition is given below:

$$S/N_{dB} = 20\log(S_{rms}/E_{rms})$$

SNR and rms error results for each simulation and test is given on Table 1.

Table 1. SNR and RMS Error for each Simulation & Test

Test or Simulation	RMS Error ($\times 10^{-3}$)	SNR (dB)
A_1 Simulation	1.048	44.18
A_1 Test	15.597	20.73
A_2 Simulation	1.181	43.09
A_2 Test	17.144	19.86

For the comparison of results, higher SNR and lower rms error level is better. So, when we compare the performances, the best results are obtained from the simulations with A1 test signal. The simulation with the test signal A2 provided slightly worse results. The worst result belongs to experiments with test signal A2. It is slightly below the 20dB acceptable performance threshold, while the experiments with test signal A1 provided slightly higher performance above the 20dB threshold. The results prove the FMTX system as a good candidate for generating quality FM signals for broadcast purposes.

4. Conclusions and Recommendations

In this study, an FM transmitter (FMTX) system is designed and implemented on an FPGA platform using SDR techniques. The main purpose is to provide a platform for education of practical SDR systems. The PC is used as a data administration and control central. The obtained results are evaluated using MATLAB scripts. Suitable ADC and DAC modules are used to process analog signals on the FPGA side and an external soundcard is used for the same purposes on the PC side. The collected signals are processed by the FPGA fabric. A signal chain is developed using VHDL hardware description language under XILINX VIVADO IDE for this purpose. The FMTX system generates a 12KHz IF signal as a frequency modulated signal. The frequency modulator depends on a wideband direct modulation method which is a very simple technique to encode frequency modulated signals. An NCO IP is used to provide this facility. Minimal use of filters is preferred in the design of the system to keep it as simple as possible. This ensures that the cost, complexity, and power consumption of the system is minimal.

The study consists of two stages: first one is the simulation and design of the FMTX system and the second stage is the implementation on FPGA and verification of the actual results. The first stage is accomplished through suitable MATLAB codes which simulates a FM transmitter using a 10 second music recording. Two different samples, A1 and A2 are derived using different audio recordings in wav format. A demodulator code provides an ideal demodulation result used for comparison purposes in the design verification stage. So, a suitable MATLAB code is also used to analysis the experiment and simulation results.

Design of the FMTX system is made on XILINX VIVADO IDE. The codes are written using VHDL. The design is based on the FM transmitter code written under MATLAB. The modulating signals are continuously played back using Audacity through the PC soundcard speaker output and modulation signal is recorded through microphone input of the same soundcard interface using HDSDR SDR software. HDSDR provides both a visual means for monitoring the results in frequency domain using its waterfall and spectrogram displays and recording them in the hard disk while listening the demodulation products through another spare soundcard output. The recorded results from the experiments are then analyzed and compared to the results from the simulation stages. The analysis results include rms error level which shows the level of noise from the original and calculation of SNR in dB from the rms error and original rms signal level.

Analysis of the results show that FMTX system implemented on FPGA is successful in the generation of FM signals. So, it can be accepted as a good candidate in the training and studying of the practical SDR principles of FM signals. The designed FMTX system can be utilized as a practical FM backend for a SDR equipment such as Softrock Ensemble TX which can be set to accept a 12KHz IF. Thus, FM communication can be possible on Civilization Band (CB) which is located at 27MHz. The FMTX system can be integrated with a suitable FM receiver built on the same FPGA and can be used as a transceiver on HF or VHF band if combined with a suitable up-down converter which can operate with a 12-KHz IF.

References

[1] Mady Z.G.A. (2016). Transmit and Receive of Quadrature Phase-Shift Keying (QPSK) Signal Using Softrock SDR and Matlab *Natural and Appliance Science of Karabuk University, Turkey*,

[2] Gareane A.G.A (2016). Transmit and Receive of FM Signals Using Softrock SDR and Matlab, *Natural and Appliance Science of Karabuk University, Turkey*

[3] Eame M.A.M (2016). Transmit and Receive of FSK Signals Using Softrock SDR and Matlab, *Natural and Appliance Science of Karabuk University, Turkey*

[4] Feng Z. (2013). A Software Defined Radio Implementation Using Matlab, *Vaasan Ammattikorkeakoulu University of Applied Sciences, Finland*

[5] N.H. Sephus, A.D. Lanterman and D.V. Anderson, "Exploring frequency modulation features and resolution in the modulation spectrum," in *Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE), 2013 IEEE* , vol., no., pp.169-174, 11-14

Aug. 2013
[6] D.V I. Vitas, D. Šimuniand P. Kneževi, "Evaluation of software defined radio systems for smart home Environments," in *MIPRO*, Opatija, Croatia, 2015, p 562.
[7] I. Hatai and I. Chakrabarti "A new high performance digital FM modulator and demodulator for software defined radio and its FPGA implementation," *International Journal of Reconfigurable Computing*, New York, United States, 2011
[8] I. Hatai and I. Chakrabarti "FPGA Implemantation of a Digital FM Modem," *International Conference on Information and Multimedia Technology*, 2009, pp 1-4
[9] D. V. W. L. and f. h., "*How to pack a room of analog FM-modulators into a Xilinx FPGA," Xilinx DSP Magazine, April 2007*
[10] Ali Ibrahim Khalifa HANDER, 2021 "*Design and Implementation of an Educational AM Receiver With FPGA Using SDR Techniques*", Karabük University, Department of Electrical and Electronic Engineering
[11] C. Kiremitci and B. Erkal, "*Eğitim Amaçlı SDR Tekniklerine Dayalı FPGA Tabanlı Genlik Modüleli Radyo Vericisi Tasarımı ve Uygulaması*", *Avrupa Bilim ve Teknoloji Dergisi*, pp. 184-189, Oct. 2020, doi:10.31590/ejosat.803492
[12] Hikmat N. Abdullah., "*SOFTWARE DEFINED RADIO USING SIMULINK HDL CODER",* University of Al-Mustansiryah, College of Engineering, Electrical-Engineering Department, Baghdad-Iraq, (2020)
[13] Internet: Digilent, "CMOD-A7 FPGA board", https://digilent.com/reference/_media/cmod_a7/cmod_a7_rm.pdf
[14] Internet: Digilent, "PMOD-AD1", https://digilent.com/reference/_media/pmod:pmod:pmodAD1_rm.pdf
[15] Internet: Digilent, "PMOD-DA2", https://reference.digilentinc.com/reference/pmod/pmodda2/reference-manual
[16] Internet: VIVADO Design Suite: https://www.xilinx.com/products/design-tools/vivado/vivado-ml.html
[17] Internet: Audacity Audio Software Systems: https://www.audacityteam.org/
[18] Internet: High Definition Software Defined Radio: https://hdsdr.software.informer.com/
[19] Internet: Mathworks Systems (MATLAB): https://www.mathworks.com/products/matlab.html/