FPGA Implementation of FM Demodulators using RTL-SDR

Vidur Shah
Department of Electronics Engineering Sardar Patel Institute of Technology
Mumbai, India
vidur888@gmail.com

Akshay Parekh
Department of Electronics Engineering Sardar Patel Institute of Technology
Mumbai, India
akshayparekh88@gmail.com

Sahil Sawant
Department of Electronics Engineering Sardar Patel Institute of Technology
Mumbai, India
sahil.sawant031996@gmail.com

Prof. Narendra Bhagat
Department of Electronics Engineering Sardar Patel Institute of Technology
Mumbai, India
narendra_bhagat@spit.ac.in

***Abstract*—With the advancements in modern communication technology, newer hardware is developed to replace the current technology. This leads to a significant increase in cost of communication systems. Thus, 'software-controlled' and 'software-defined' hardware, for implementing communication system processes were developed. This research work revolves around the idea of developing a powerful, re-configurable tool capable of implementing multiple communication systems on-chip via software. The overall system developed here is referred to as a 'Reconfigurable Receiver'. The receiver utilizes RTL-SDR(R820T+RTL2832U) by Oscmocom and Spartan 3AN FPGA for implementing FIR filters and demodulating architectures.Such a system offers a greater flexibility over dedicated hardware. The work presented here includes the theoretical background and description of the process flow for developing such systems. Also, various performance parameters are discussed. The system described here is designed to demodulate FM signals from local FM stations.**

***Keywords*—Digital FM demodulation, Digital Filters, Signal Processing on FPGA**

I. INTRODUCTION

A reconfigurable receiver is an entity inside communication systems which is able to receive desired signals from the available wideband spectrum [1]. These signals include FM, AM modulated signals from local radio stations or QAM based TV signals, DVB-T signals, 802.11 MIMO-OFDM signals, GSM signals or DVB-S satellite signals. The primary characteristics of any of these above signals are it's amplitude(magnitude), frequency and phase. Purely amplitude based modulation schemes are less preferred due to the reduced signal-to-noise ratio(SNR) leading to increased bit-error probability. Therefore, frequency and phase modulation is commonly preferred as the magnitude is never varied leading to uniform signal power leading to a high SNR [2]. Recently, digital communication systems are being used compared to their analog counterparts owing to their better noise performance. The reconfigurable receiver should be scalable, in that it should have the ability to receive both analog and digital modulation schemes and also provide optimum noise rejection.

The AM and FM modulation schemes can be mathematically explained as shown in (1) and (2). Today, AM modulation schemes and its variants(DSB-SC,SSB,VSB) are used in relatively simpler applications. FM modulation schemes on the other hand are widely used in commercial radio stations as well as other broadcasting purposes [2].

RTL-SDR is a module commonly used as a TV dongle. It is a wideband receiver with a range of approximately 24MHz-1750MHz. This encompasses a spectrum consisting of FM radios, GSM signals, DVB-T signals, GPS and GNSS signals,etc. The tuning is achieved by R820T which is a silicon tuner consisting of an in-built PLL multiplier with a base clock of 16MHz [3]. This is followed by a I-Q demodulator and an 8-bit ADC with a variable sampling rate. The samples are communicated using the on-board USB 2.0 interface.

The paper flow is as follows: the theoretical background for analyzing and implementing various demodulation schemes is discussed in the following section. Section III describes the system architecture for implementing the receiver. Section IV describes the software level flow. The Simulation results and Hardware outputs are analyzed in Section V. The remaining section discusses the conclusions observed and inspiration for future research.

II. THEORY

The mathematical expressions for a generic AM and FM waves are as follows:

$$A_{AM} = A_c(1 + m(t))\cos(w_c t) \tag{1}$$

$$A_{FM} = A_c \cos(w_c t + 2\pi K_f \int_{-\infty}^{t} m(t)) \tag{2}$$

m(t) is the modulating signal,
A_C the carrier signal amplitude
W_C is the carrier signal angular frequency
K_f is the frequency sensitivity

The bandwidth of the AM signal is $2f_m$ whereas for FM Carson's rule is used.

On the demodulation front, the first component encountered is the wideband receiver(RTL-SDR in this case). The architecture inside RTL-SDR can be described as shown in Fig.1

RTL-SDR comprises of an internal PLL for clock multiplication(inside R820T) followed by a multiplier and a

978-1-5386-5257-2/18/$31.00 ©2018 IEEE

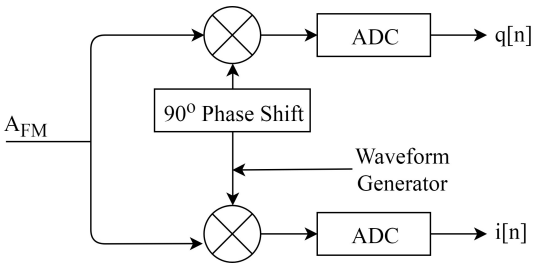

Fig. 1. Internal structure of RTL-SDR

Hilbert Transformer(90 degree phase shifter) [3]. The internal clock tunes to the desired FM station. Thus, the synthesized signal(equal to the FM carrier frequency) is multiplied with the received FM signal. This shifts the spectrum to the baseband level. The following sub-section describes baseband FM demodulation techniques.

The phase information can be extracted from the I-Q samples as follows:

$$\theta = \arctan \frac{q[n]}{i[n]} \tag{3}$$

The frequency is obtained by taking the first derivative of the phase as follows:

$$\omega = \frac{d\theta}{dt} \tag{4}$$

$$\omega = \frac{d \arctan \frac{q[n]}{i[n]}}{dt} \tag{5}$$

$$\omega = \frac{i[n]\frac{dq[n]}{dt} - q[n]\frac{di[n]}{dt}}{i[n]^2 + q[n]^2} \tag{6}$$

$$f = \frac{\omega}{2\pi} \tag{7}$$

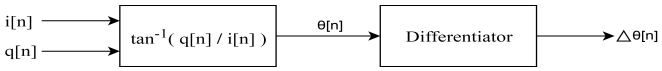

Fig. 2. Arc-tangent based implementation of FM demodulator

Equations (5) and (6) describe a direct method to obtain the angular frequency ω,from which the frequency f can be obtained using (7). The methodology to use (5) through (7) is shown in Fig.2 through 4.

Fig.2 describes a demodulator using arctangent to obtain the angle.Fig.3 indicates a design without the arc-tangent function by simply taking the derivative of the arc-tangent function as in (6) [4], [5]. These structures involve design of a multi-tap differentiator. A simple differentiator can be a single tap differentiator but the response observed is poor. Fig.4 shows a single-tap differentiator based FM demodulator. The inherent advantage is the reduced hardware involved while implementing.

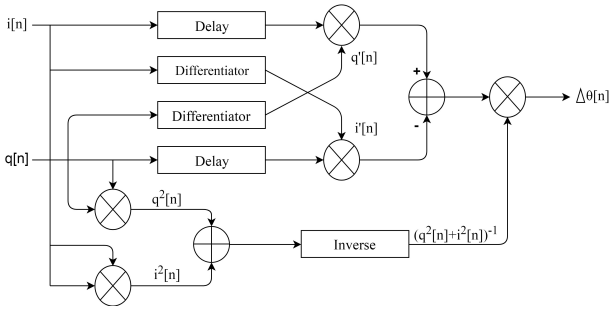

Fig. 3. FM demodulator after taking derivative of Arc-tangent

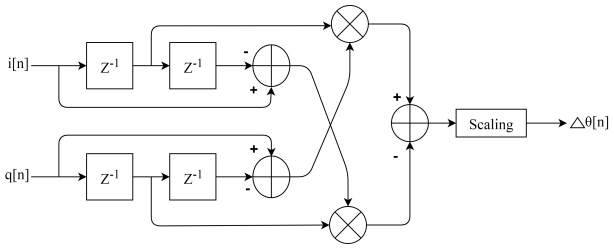

Fig. 4. Simplified FM demodulator

For AM demodulation the information is stored in the carrier. Thus, the modulating signal can be obtained as:

$$A_m[n] = \sqrt{i^2[n] + q^2[n]} \tag{8}$$

Equation (8) can be obtained by taking the square root of complex multiplication of (i[n]+jq[n]) and it's conjugate. Thus, digital AM demodulation is fairly simpler compared to FM.

III. SYSTEM STRUCTURE

A basic system-level block diagram is as shown in Fig. 5.

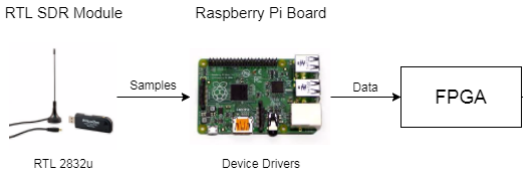

Fig. 5. System Block Diagram

The major system components can be listed as:

- RTL-SDR (RTL2832U)
- Raspberry Pi Model B+
- Spartan 3AN FPGA Starter Kit

The functionalities supported on the RTL-SDR are discussed in detail in the previous sections. The Raspberry Pi B+ is used as a USB to serial converter as readymade drivers for the RTL-SDR are developed by Osmocom. The details of this process are described in the succeeding section.

The heart of the system is the Spartan 3AN FPGA. The entire software flow required for the software defined system

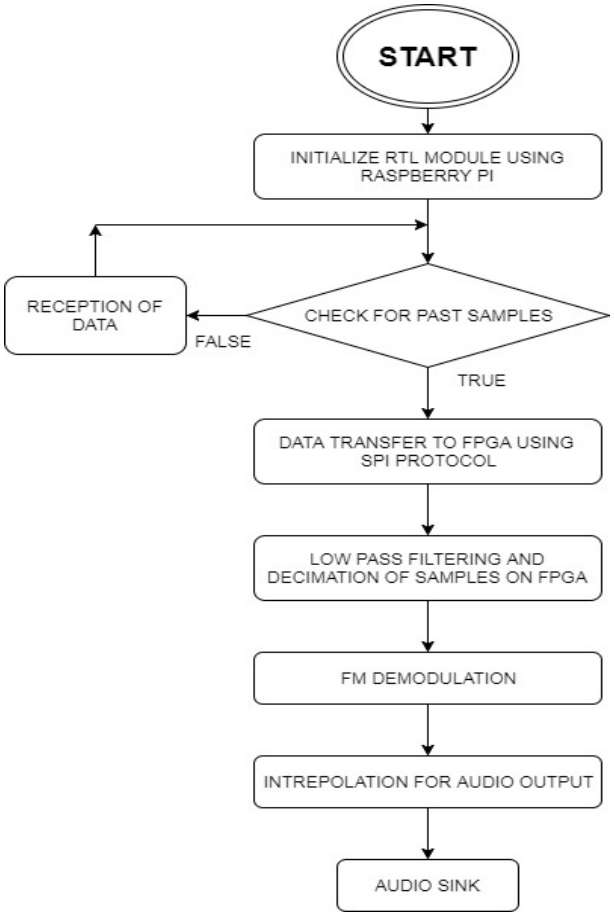

Fig. 6. Flowchart

is implemented on the FPGA. To improve the speed and increase the overall throughput, the system is implemented with 8-bit data width. The FPGA does not have DSP slices, hence trigonometric functions are synthesized using look-up tables (LUTs) only [6]. The basic building blocks include filters, decimators, serial receiver and demodulator. On top of this, few functionalities are added to maintain synchronization between these components during operation. These components are modeled using Behavioral level modeling in Verilog. Hence, the name 'Software-Defined' can be appropriately used to describe such systems.

IV. SOFTWARE ALGORITHM

This section provides system operation from the algorithm point-of-view. The algorithm is designed considering the various system performance parameters and requirements. The basic algorithm is shown in Fig. 6.

On system start, the Raspberry Pi sends initialization commands to the RTL SDR module via the USB interface. This process is initiated by a script being executed on Pi. The script also checks for past samples if any(which will be stored in a file). If the data is unavailable, it collects new samples from the RTL-SDR module via the USB interface ,else it will initiate transfer of the data to the FPGA via serial protocol. The script is developed as a mixture of C and assembly and is capable of operating GPIOs at higher frequencies compared to Python or Perl [7].The observed data rate is approximately 2Mbps. The maximum sampling rate of the RTL-SDR module is 2MSps which is necessary for real-time sampling and transfer.However, with the clocking limitations of the Pi, the sample rate possible to avoid overwhelming or underwhelming of samples is 2M/8 = 250 kSamples/s

The next stage gets implemented on the FPGA. The FPGA uses its I/Os to implement the receiver of the serial protocol. Once, data is available on the FPGA, further operations can be initiated. The first stage includes low-pass filtering the input samples for decimation. The Decimation filter is a multi-rate filter to reduce the sampling rate of the signal before FM demodulation. Before FM demodulation another low-pass filter is used. After that, the the demodulation is done via the I and Q samples obtained. Here, the I and Q samples are received in an alternating pattern thus, at a time only one sample either I or Q is processed. These samples are combined in the demodulator to obtain a unified output. This output has a higher sampling rate compared to the common audio sinks available. The sampling rate is converted to 48kHz using multi-rate structures such as Decimator and Interpolator networks. The resultant signal thus obtained can be passed to an audio sink. For further audio processing, audio codec can be used.

A. *Communicating Protocol on Raspberry side*

The Raspberry Pi B+ model has the capacity to generate clocking signals on I/O pins in the MHz range particularly around 5MHz. This reduces the overall channel capacity thereby reducing the maximum possible data rates. To maintain the integrity of the sampled data, the sample rate of the RTL-SDR module must be approximately equal to that of the data rate given by the Raspberry Pi's I/O pins. The major problem faced during implementation was the unstable clock reference generated on the Pi due to a multi-process environment. Now, this cannot be controlled without working on the kernel. Instead, here, a 2 data-pin type protocol is used as shown in Fig.8. Further, a synchronizing bit is provided on a separate line to allow synchronization with the FPGA as shown in Fig.7.

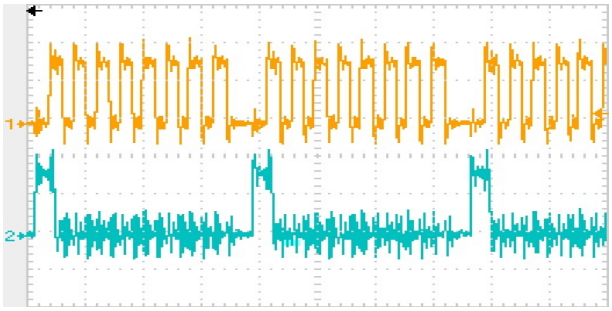

Fig. 7. Clock and Sync pulses

B. Communicating Protocol on FPGA side

As a stable clocking reference is unavailable, edge sensing of this signal using a clocking route inside the FPGA does not give better results. Hence, the FPGA's oscillator is used as a clock reference and at every clock's rising edge, the signal pins are sensed on which required action is taken. Although, an insignificant delay is observed in data transfer and there is lack of synchronization, this algorithm proves to be better from performance point of view.

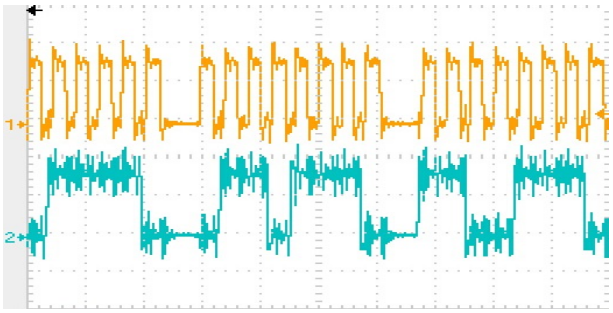

Fig. 8. Clock pulses with bit-wise data

V. Final Assembly

The RTL-SDR is tuned to a particular station via C script on Pi. It invokes commands specified in the device driver of RTL-SDR. The controllable parameters include Number of samples to be received along with mode(synchronous and asynchronous), sample rate, RF gain, tuning frequency,etc. [8]. RTL-SDR has a built in processor that separates this data into I and Q sample and writes it into file, which is then read, wherein I and Q samples are transmitted via separate line to FPGA on SPI. Received data stream which is signed is first converted to unsigned and is then demodulated using multipliers, CORDIC blocks and delay blocks. The demodulated output is again converted into signed and is given to LM386 Audio Amplifier to recover the data at the tuned station

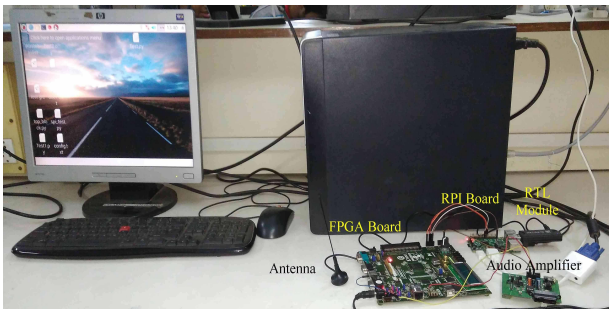

Fig. 9. Test Setup

VI. RESULTS AND DISCUSSION

The first step in constructing a FM broadcast receiver was to implement a sample model. For this GNURadio which is a Linux-based, Open-Source tool is used. It has the ability to directly accept raw I-Q samples from RTL-SDR via USB and utilize the host PC's sound card to give audio samples. Here, a block diagram was built to receive audio data from nearby FM station. The diagram was thoroughly analyzed for system requirements of a real-time demodulator. A FFT plot obtained in the process is shown in Fig.11. The FM station are indicated by the distinctive peaks.

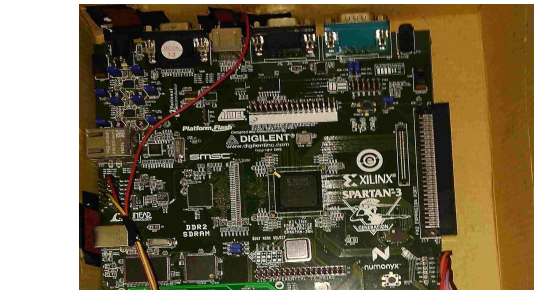

Fig. 10. Final System Setup

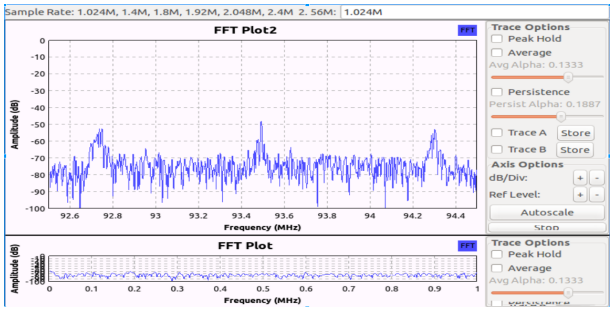

Fig. 11. GNU Radio FFT plot

The next step was to model RTL-SDR in MATLAB's Simulink, in order to simulate the FM demodulator. Simulink provides a block-diagrammatic approach to designing systems. The simulation was carried out at 1MHz-10MHz carrier and frequency deviation of 75kHz (similar to those of the broadcast stations). The modulating signal was chosen to be sinusoidal to analyze the frequency dependence of the system's components. The sampling frequency was 250 KSamples/s. These samples were directly demodulated using the demodulator architectures discussed earlier.Down-conversion of the demodulated signal was carried out to make it compatible to audio amplifiers. The

simulation aimed at faithful reception of the Mono output in the FM baseband spectrum(0-20kHz).

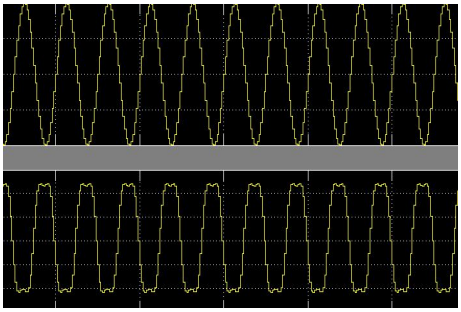

Fig. 12. Given input and observed output in Simulink

Before actual implementation, a serial protocol was implemented between the Raspberry Pi and FPGA. It was developed on the lines of SPI(Serial Peripheral Interface) protocol. To increase the data rates, direct GPIOs of the Pi were interfaced using C programming. Theoretically C on Pi can generate clocks upto 10MHz. For better detection of edges by the FPGA, the clock generated was reduced to 2.5 MHz. The 8-bit samples are transmitted from the Pi along with a Synchronizing bit to ensure correct reception at the FPGA end.

The lowpass filters designed during the simulation phase, were synthesized using Behavioral modeling on the FPGA. HDL Coder was used for the same. The FM demodulator architectures were synthesized one at-a-time. The overall hardware requirements of the system were to be reduced for accommodating more receiver architectures in the system. A base criterion was set to keep the filter order below 15 and 8-bit wide. Due to the reduced bus-width the relative SNR observed is low. But the time required for multiplication reduces significantly compared to a 16-bit implementation.

For arctangent calculation Xilinx's CORDIC(Co-ordinate Rotation Digital Computing) was chosen as the tool. However, due to radix-2 co-ordinate rotation, the accuracy of phase angles beyond 60 degrees was reduced. To overcome this, 2 CORDIC blocks were used in complementary fashion, thus, increasing the overall accuracy, significantly. A sample output observed after the FM demodulator is shown in Fig.13 . The differentiator filters were designed from Simulink, in that a linear magnitude response was observed at low frequencies. This was followed by a bank of lowpass filters. The frequency spectrum observed at the final stage is shown in Fig.14.

The filter outputs simulated in Simulink were verified at the outputs of the synthesized filters. Due to 8-bit quantization of the filter coefficients and entire product and accumulation operations working at 8-bit data width, a significant difference is found in the stopband response but the passband response is found to be similar to the required response.

The synthesis statistics of entire FM demodulator with arctangent and single-tap differentiators is shown in Table.1. The area required is more compared to feedback based demodulators [9] but the overall processing delay is significantly less.

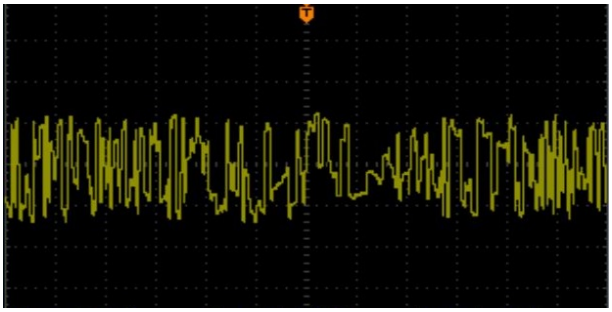

Fig. 13. Real-time Phase calculated by the FM demodulator on the FPGA

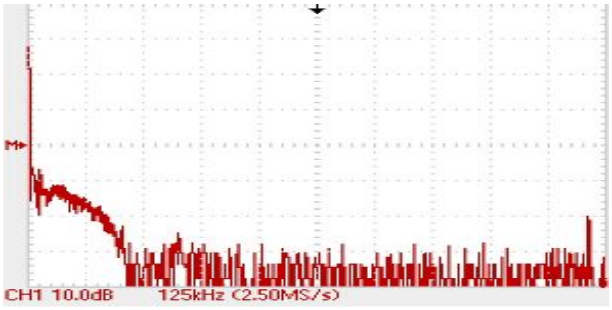

Fig. 14. Final FFT Plot

The single-tap differentiator utilizes less slices and at the same time reduces the latency. The only compromised parameter is the overall accuracy. Thus, a trade-off between speed and accuracy is clearly visible.

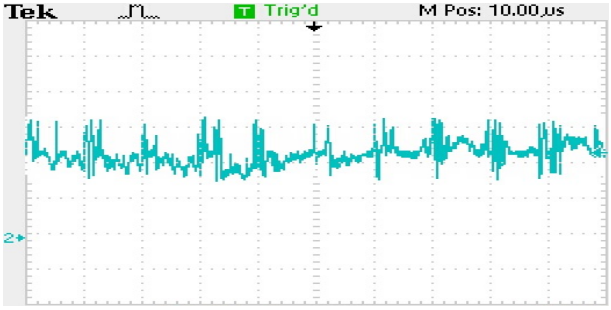

Fig. 15. Audio data after demodulation

VII. CONCLUSIONS

A Reconfigurable Receiver design capable of demodulating FM signal in real-time is presented. The objective of this research work is to demonstrate the use of a FPGA-based reconfigurable hardware for implementing DSP operations required for demodulation and faithful reconstruction. The objective is to develop a receiver comparable to conventional receivers but possessing reconfigurability for receiving other signals as well.

From the results shown in the previous sections, the filters designed above can be used along with the demodulation ar-

TABLE I
SYNTHESIS REPORTS OF VARIOUS DEMODULATOR ARCHITECTURES

Demodulator Architecture	Number of Slice Flip Flops	Number of 4 input LUTs	Number of occupied slices
Arc-tangent based	530/11776	1577/11776	1155/5888
Single-tap differentiator	575/11776	919/11776	754/5888

chitectures for accomplishing complete demodulation. Further simulations show that such filters can be extended over to higher frequencies thereby increasing the adaptability of the presented system as the normalized passband and stopband specifications remain the same. However, due to integer type filter coefficients, the filter performance tends to reduce. Reduced bit widths of the filter coefficients highly affects the performance. Thus, a clear trade-off is observed between the area of the synthesized circuit(on FPGA) and the accuracy.

Different demodulator architectures presented in Section II were implemented. Out of these, the single-tap differentiator required the least on-chip area and less latency compared to the arctangent based demodulators. The design flow used here, can be used for further developing different receiver architectures. The test hardware presented here is limited to the clocking rates of the components used. For better accuracy, the signaling rate between the components must be increased which points to developing a USB interface directly from RTL-SDR module to the FPGA.

REFERENCES

[1] E. Grayver,"Implementing Software Defined Radio", New York, NY: Springer New York, 2013.

[2] W. Tomasi, "Electronic communications systems", 5th ed. Upper Saddle River, N.J.: Pearson/Prentice Hall, 2004.

[3] "R820T Archives - rtl-sdr.com", rtl-sdr.com, 2018. [Online].

[4] Richard G. Lyons "Understanding Digital Signal Processing, 3rd edition", Pearson ,2011

[5] F. Yu,"FPGA implementation of a fully digital FM demodulator",The Ninth International Conference on Communications Systems, 2004. ICCS 2004.

[6] "Spartan 3AN FPGA family datasheet", xilinx.com, 2013. [Online].

[7] "GPIO", raspberrypi.org, 2018. [Online].

[8] "rtl_sdr manual", manpages.ubuntu.com", 2018. [Online].

[9] M. Rice, M. Padilla and B. Nelson,"On FM Demodulators in Software Defined Radios Using FPGAs",MILCOM 2009 - 2009 IEEE Military Communications Conference.