First International Conference on Information Technology, Communications and Computing (ICITCC 2017)
24-December-2017, Bhopal, M.P., India | ISBN (Online): 978-81-932623-3-7 | DOI: 10.5281/zenodo.1130795

Evaluation of SDR using open source technology

Mrs. Leena V. Govekar[1], Pravesh C. Patil[2], Dr. Y.S. Rao[3]
[1]Department of Electronics Telecommunication Engineering, Thadomal Shahani Engineering College, Mumbai-50
[2]Department of Computer Engineering, Sardar Patel Institute of Technology, Mumbai-58
[3]Vice Principal, Sardar Patel Institute of Technology, Mumbai-58

***Abstract*—In this paper, an attempt is made to develop adaptable stage for programming characterized Software Defined Radio (SDR) system. Considering the wide demand of wireless communication, the paper aims to propose a flexible platform for software-defined radio, which will be able to meet the wide spectrum from 70 MHz to 6 GHz. This article reviews investigation on the current equipment stage for SDR. It highlights the interfacing of AD-FMCommS4 with ZedBoard. The system can be utilized to extend the future low power gadget devices to integrate with SDR.**

***Index Terms*—AD-FMCommS4 SDR, ZedBoard, GnuRadio, Interfacing**

I. Introduction

Over the last few years, analog radio systems are being replaced by digital radio systems for various radio applications, in the military, civilian and commercial spaces [1]. Commercial wireless communication industry is facing problems due to the constant evolution of link layer protocol standards, (2.5G, 3G, 4G), the existence of incompatible, wireless network technology in different countries inhibiting deployment of global roaming facilities and problems in rolling out new services/features due to the widespread presence of legacy subscriber handsets.

To resolve this issue, SDR is the solution. SDR comprises of reprogrammable, reconfigurable hardware. It uses a reprogrammable ability of Field Programming Gate Array (FPGA) or digital signal processor to build an open architecture with the software implementation of radio frequencies such as modulation, demodulation, encoding, decoding.

As the need grows for the use of wireless networks for more diverse, data-heavy and minimum delay applications, wireless protocols must be adapted to meet the various needs of these applications including cost, lower energy consumption, and higher data rates. As the number of users increases on commonly-accessed mobile bandwidths, congestion becomes another issue, and more versatile methods must be applied to handle the contention inherent in multiple access. Modern-day wireless communications standards are constantly advancing to accommodate the needs of an increasing number of devices. The latest standard for mobile phone technology is known as long-term evolution (LTE), and many research projects are in place to prototype and test the 5th generation (5G) technology. For enhancing the use of the application with the ability to use latest technology standards various SDR has been introduced enhancing the older version with the addition of new advanced features. As shown in TABLE I are some of the latest SDR introduced with some new features which attract developers to use them. We have compared SDR which is full duplex as the SDR we used is 1x1 transmitter and receiver thus observing that FMComms4 SDR satisfies most of the latest specification within low cost and maintaining optimized performance.

AD-FMcomms4 SDR applications can be developed on many software platforms such as Xilinx, Matlab, Simulink and Gnu Radio whereas GnuRadio is an open source tool. We demonstrated a transmitter and receiver application with DPSK modulation and compared the output with ADI-Oscilloscope's output.

II. Literature Survey

The author has implemented modulator, demodulator in one single processing core. High data rate more than 10Mbps. The concept of reconfiguation and ability to upgrate has lot of demand in many application. FPGAs have specific features that enable SDR implementation. These include: high-bandwidth memories, embedded DSP Blocks, phase-locked loops (PLL), and high-speed interfaces. In addition, soft processors plus FPGA co-processors enable reconfiguration of the digital waveforms[2].

SDR is a flexible platform for multiple frequency band. It provides generic physical layer which with the software part is able to tune with multiple frequencies standards. A typical drawback of SDR is coupling between hardware and software. The author has described the implementation using GNU radio and USRP2 platform [3].

OFDM system provides high level of robustness for the intererance. In this paper the design of OFDM system is implemented . Moreover, the design and simulation results for some standards of OFDM using MATLAB will be observed as a practical system examples that uses OFDM as a modulation technique. All the proposed modules are designed using VHDL programming language [4].

III. Interfacing of AD-FMcomms4 with Zed board

AD-FMCOMMS SDR are the RF front end for capturing signal as these do not posses processor they can't communicate with computer for software processing of signal so they need a carrier board which has both an embedded ARM processor for software implementations as well as Field Programmable Gate Array (FPGA) fabric. Both are connected via FPGA mezzanine card (FMC) for completing signal processing. Various

First International Conference on Information Technology, Communications and Computing (ICITCC 2017)
24-December-2017, Bhopal, M.P., India | ISBN (Online): 978-81-932623-3-7 | DOI: 10.5281/zenodo.1130795

TABLE I
COMPARISON OF DIFFERENT SDR

	Frequency	Bandwidth	New Feature	Noise Figure	ADC Resolution	Cost in USD
SDR we used[11]	70MHz-6GHz	<200 kHz-56 MHz	1x1 AD9364 (low cost)	<2.5 dB	12 bit	421
FMcomms5[9] Analog Device	70MHz 6GHz	<200 kHz-56 MHz	4x4 AD9361 (MIMO)	2 dB	12 bit	1091
USRP E313 Ettus[10]	70MHz 6GHz	Up to 56 MHz	PoE with surge protection	<8 dB	12 bits	4013
MatchstiqS12 [13]	<1 MHz-6GHz	Up to 50 MHz	Tuning range (S10/S11 has 70 MHz-6 GHz)	<8 dB	12 bits	4600
BladeRF-x115[14]	300 MHz3.8GHz	20 MHz	Has a larger FPGA	--	12 Bits	700
LimeSDR[15]	100 kHz-3.8 GHz	61.44 MHz	App Store on their Ubuntu software	<3.5dB	9 Bits	600

carrier boards which can be used with AD-FMCOMMS4 are as follows. [6]

1) AC701
2) KC705
3) VC707
4) ZC702
5) ZC706
6) Zed Board
7) MITX045

As we are using Digilent's ZedBoard as carrier board our setup includes FMcomms4, ZedBoard, GnuRadio as the main modeules for signal Processing.

A. AD-FMcomms4

AD-FMcomms4 is a FPGA Mezzanine Card (FMC) having highly integrated RF Agile Transceiver IC AD 9364. The purpose of the AD-FMComms4-EBZ is to provide an RF platform to software developers, system architects, etc, who want a single platform which operates over a much wider tuning range (70 MHz 6 GHz). The device integrates an RF front-end portion with a mixed-signal baseband section and frequency synthesizers. It has 2.45 GHz Balun. This balun is rated for a operating frequency of 2400-2500 MHz. If you want to evaluate the part outside of this frequency range, an alternative balun should be installed. The AD-FMComms4 (AD9364) assumes a VDD-interface voltage between 1.71V and 2.625V (1.8 to 2.5 +/- 5 percent), so on FPGA carrier board i.e ZedBoard it must be ensured that VADJ is between these levels. Setting things to 3.3V will damage the part [15].

1) Features for FMComms4 are [9]:

- RF 1 x 1 transceiver with integrated 12-bit DACs and ADCs
- Supports time division duplex (TDD) and frequency division duplex (FDD) operation
- Tunable channel bandwidth (BW) : ¡ 200 kHz to 56 MHz
- 3-band receiver: 3 differential or 6 single-ended inputs
- Superior receiver sensitivity with a noise figure of ¡ 2.5 dB
- Rx gain control
- Real-time monitor and control signals for manual gain
- Independent automatic gain control
- 2-band differential output transmitter
- Highly linear broadband transmitter
- Tx EVM: 40 dB
- Tx noise: 157 dBm/Hz noise floor
- Tx monitor: 66 dB dynamic range with 1 dB accuracy
- Integrated fractional-N synthesizers
- 2.4 Hz maximum local oscillator (LO) step size
- Multichip synchronization
- CMOS/LVDS digital interface

2) Receiver Path: The receiver section contains a low noise amplifier (LNA), matched in-phase (I) and quadrature (Q) amplifiers, mixers, and band shaping filters that down convert received signals to baseband for digitization. Gain control is achieved by pre programmed gain index map that distributes gain among the blocks for optimal performance at each level. This is achieved by enabling the internal AGC in either fast or slow mode or by using manual gain control. Additionally, each channel contains independent RSSI measurement capability, dc offset tracking, and all circuitry necessary for self calibration. The receiver's 12-bit ADCs has adjustable sample rates that produce data streams from the received signals [11].

3) Transmitter Path: The transmit route provides digital processing, mixed signal, and RF blocks necessary to implement a direct conversion system. The digital data received from the Baseband Processor(BBP) passes through a fully programmable 128-tap FIR filter with interpolation options. The FIR output is sent to series of interpolation filters that gives additional filtering and data rate interpolation prior to reaching the DAC. Each 12-bit DAC has an adjustable sampling rate. Both the I and Q channels are fed to the RF block for upconversion. After conversion to baseband analog signals, the I and Q signals are filtered to remove sampling artifacts and fed to the upconversion mixers. At this point,the I and Q signals are recombined and modulated on the carrier frequency for transmission to the output stage. This combined signal also passes through analog filters that provide additional band shaping, and then the signal is transmitted to the output amplifier. The transmit channel provides a wide attenuation adjustment range with fine granularity to help designers optimize signal-to-noise ratio (SNR). Self calibration circuitry is built into each transmit channel for automatic real-

First International Conference on Information Technology, Communications and Computing (ICITCC 2017)
24-December-2017, Bhopal, M.P., India | ISBN (Online): 978-81-932623-3-7 | DOI: 10.5281/zenodo.1130795

time adjustment. The transmitter block also provides a Tx monitor block. This block monitors the transmitter output and routes it back through the receiver channel to the BBP for signal monitoring. The Tx monitor block is available only in TDD mode operation while the receiver is idle [9].

4) Few Applications:

- Point to point communication systems
- Femtocell/picocell/microcell base stations
- General-purpose radio systems

B. ZedBoard

ZedBoard is a low-cost development board for the Xilinx Zynq-7000 all programmable SoC (System on Chip) combining a dual Cortex-A9 Processing System (PS) with 85,000 Series7 Programmable Logic (PL) cells. It has memory 512 MB DDR3 (128M x32) (Double Data Rate Type 3) and 256Mb QSPI Flash (queued serial peripheral interface). ZedBoard communicates with FMComms4 via FMC-LPC (FPGA Mezzanine Card-Low Pin Count) connector. ZedBoard serves as back end. In back-end operations such as (de)modulation, filtering, and channel (de)coding are performed.

Software portion of the SDR platform is implemented through the usage of ZedBoard. Additionally, the Zedboard includes a Gigabit ethernet interface that allows remote access to the onboard system. ZedBoard board contains everything necessary to create a Linux, Android, Windows, or other OS/RTOS based design [10].

In Interfacing with Zedboard(used here), FMComms4 SD card image has Linux Linaro 14.04 bootable kernel which interacts with Zedboard. Linaro focuses on the use of the ARM instruction set including concrete implementations of these, such as SoCs that contain Cortex-Ax processor(s). Linaro works close to the core services such as multimedia, graphics, security, power management etc. It aims to provide stable, tested tools and code for multiple software distributions to use to reduce low-level fragmentation of embedded Linux software.[14]

C. Libiio

Libiio is a software library to interface the Industrial I/O(IIO) framework. The objective of libiio is to ease the development process of applications using IIO devices, by letting the library be the intermediate between the program and the Linux kernel. As libiio complies with the C99 and POSIX standards so porting the library to Windows has been exceptionally easy and required very little change, due to recent POSIX sockets compatibility layer provided in Windows [15].

The central and most complex piece of the libiio library is the local backend. Local backend will interact with the hardware through the sysfs interface of the Linux kernel. IIO Daemon is for network backend which allows the applications using libiio to stream samples on the network to any connected device.

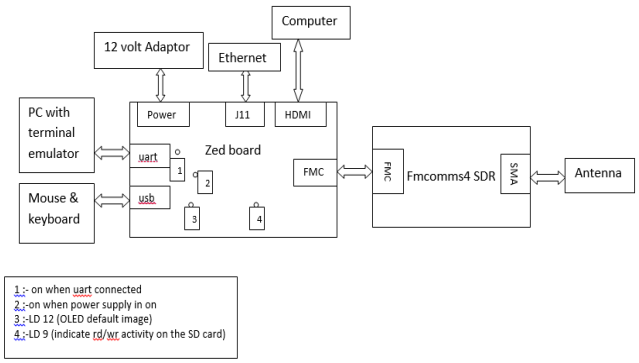


Fig. 1. Interfacing Block Diagram.

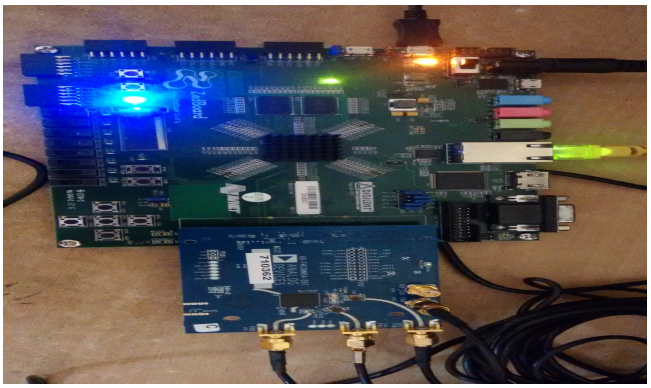


Fig. 2. Hardware setup of FMComms4.

D. GnuRadio

GnuRadio is a open-source software for developing SDR Application. FMComms4 SDR requires its own source and sink blocks to deliver or collect the complex samples generated by the DSP blocks, for which it uses IIO-lib from inside GNU Radio. FMComms4 sink absorbs the samples from the output of flow graph and manages the transmitting process in the hardware. Whereas, the source block manages the receiving process in the RF-frontend and delivers the complex samples to the other blocks in the flow graph. In a lower level, the blocks communicate directly to the DAC and ADC buffers. Using the drivers and libraries (IIO-lib) made available by ADI, it is possible to write and read directly to and from the buffers of the hardware components, taking into consideration the used data types[5].

E. Interfacing Steps

Analog Devices provides Linux images built for the FMCOMMS4 modules that complete and enable the development environment. For interfacing always the latest image must be used available at [17], we used 2016-R2 image. ZedBoard and FMComms4 board are connected as shown in Fig. 1.

Steps for Interfacing

1) Download Linux image zip file from Analog Device site, save it on computer.
2) unzip it in folder on computer.

First International Conference on Information Technology, Communications and Computing (ICITCC 2017)
24-December-2017, Bhopal, M.P., India | ISBN (Online): 978-81-932623-3-7 | DOI: 10.5281/zenodo.1130795

3) Verify md5sum for both zip and unzip file with checksum given on Analog Devices site.
4) Take blank 8 GB SD Card and mount extracted image on it using command
   sudo dd if=your image.img of=/dev/mmcblk0
5) Open SD card go to folder Zed-fmcomms4 copy all files from sub-directory and paste on main directory
6) Safely eject SD Card from computer and insert in ZedBoard's SD Card slot
7) Do Jumper settings on ZedBoard for SD Card Boot mode [18].
8) Connect Computer to ZedBoard via usb-uart cable.
9) Open serial Console and set port as ACM0, Baud rate as 115000 and parity none
10) On seeing zynq-uboot run boot command
11) With successful interfacing you can see root@analog Thus board is now ready to run various applications.

On completion of all steps, interfaced hardware is seen as in Fig. 2 with respective LED's glowing.

IV. Implementation

To verify the interfacing and to evaluate performance of FMComms4 we developed transmitter-receiver flow graph in GnuRadio with DBPSK modulator block as show in Fig. 3 and Fig. 4. The Random Source block generates random bits that are encapsulated in blocks of two by the Packet Encoder block. The information is then modulated and pulse-shaped using the DPSK Mod block. The I and Q samples are delivered to the Fmcomms sink block, which interfaces with the DAC component in the RF front-end. In the receiver portion, the samples are collected from the ADC into the GNU Radio environment using the FMcomms Source block. The transmitted symbols can be visualized in the QT GUI Constellation Sink block.

In result we observe as in Fig.5(a) DBPSK symbols spread over the in-phase axis given the pulse-shaping operation. The received symbols can be noticed in Fig.5(b) that the received symbols present a phase rotation in comparison to the transmitted symbols and presents the output of DBPSK. This rotation is introduced by the RF cable used to connect the RF in and RF out of the FMCOMMS4 board. We verified GnuRadio output with the IIO-oscilloscope application the graphs as shown in Fig. 6 where almost same.

V. Analysis with FMComms4

As BPSK experiment is implemented we analysed FMComms4 performance with respect to it. FMComms4 has default installed balun of 2.4GHz with range 2.3 GHz to 2.5 GHz this is local oscillator's frequency band beyond this baseband processor can not set it's rate and FMComms4 stops giving output as the minimum baseband rate with the FIR filter (decimate by 4) enabled is: 25MSPS /48 = 520.83 kSPS. Without its 25MSPS/12 = 2.083 MSPS. This error can be rectified by installing baluns of higher range specified by Johanson Technology's. Next considering receiver gain, it is set as 64 dB by default if too low is inefficient, as it does

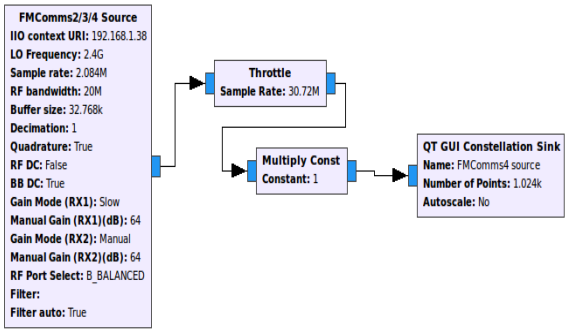


Fig. 3. FMComms4 as Receiver.

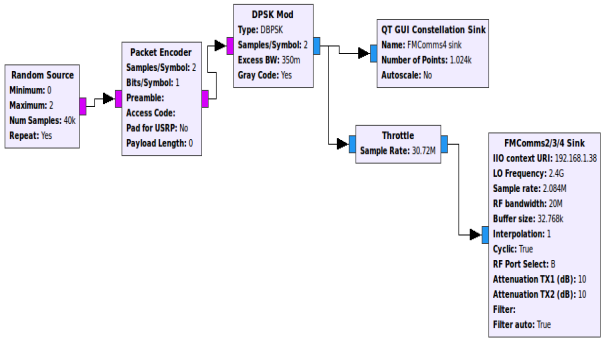
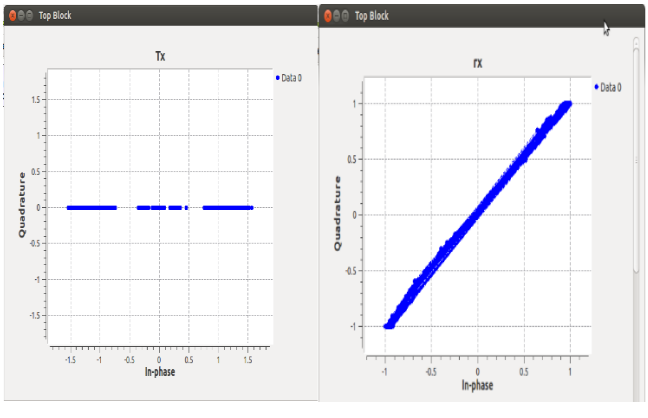


Fig. 4. FMComms4 as Transmitter.

not fully utilize the voltage range of the system's digitizer. If the receiver gain is set too high, the Fourier transformed spectra will be clipped, or cut off. We get signal amplitude as 1 Volt peak to peak at 64 dB gain, amplitude starts decreasing after 45 dB gain as lower limit and is clipped after 70 dB as upper limit on entering higher gain we get error as "FMComms4 ad9361 spi32766.0: Invalid gain 80, supported range [-3 to 71]" showing range of gain. Studying variation of



(a) (b)

Fig. 5. GnuRadio Results (a) Transmitter Output (b) Receiver Output.

First International Conference on Information Technology, Communications and Computing (ICITCC 2017)
24-December-2017, Bhopal, M.P., India | ISBN (Online): 978-81-932623-3-7 | DOI: 10.5281/zenodo.1130795

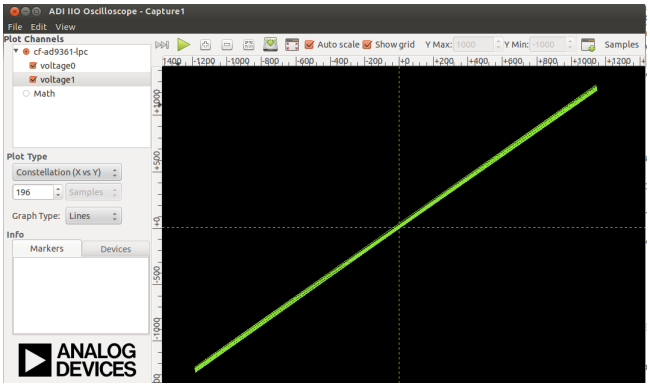


Fig. 6. ADI IIO-Oscilloscope graph for Receiver output.

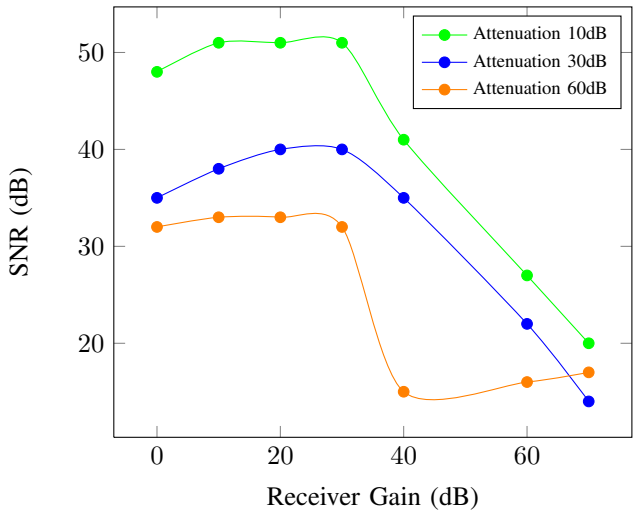


Fig. 7. Variation of SNR over Receiver Gain

power with respect to receiver gain by the graph as shown in Fig.7 concludes that as gain increases received signal power decreases also found that percentage of overshoot increases as gain increases. On increasing attenuation power also increases. Upto 60 dB attenuation a very small signal with phase change as required can be seen but beyond 65 dB attenuation only noisy signal was observed.

Observing buffers at transmitter(sink) and receiver(source) side of FMComms4, input buffers at transmitter were 0.515 percent full while at receiver output buffer were 0.068 percent full. Comparing input buffer has higher value because sink is a final block it holds all data to make sufficient strength of signal and forward it for successful transmission. While output buffer of FMComms4 immediately passes the received value to next block here it is throttle so FMComms4 source doesn't hold data for long as seen in Fig.8 buffer for throttle buffer is higher than FMComms4 source block as throttle holds data for rate-limiting purpose.

VI. Future Work

In the future, we plan to perform tests with radio transmissions and develop few applications based on wireless standards

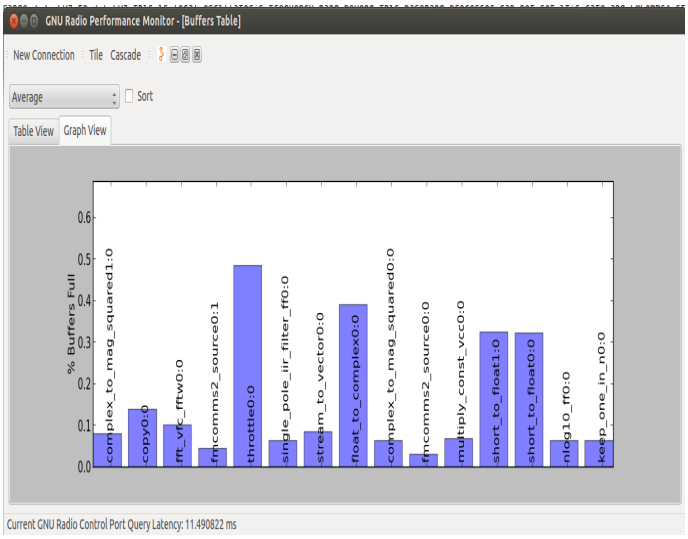


Fig. 8. Buffer table of BPSK FMComms4 Receiver.

so as to explore the performance of FMComms4. Presently only 2.4 GHz to 2.5GHz frequency band is been used due presence of 2.4GHz balun in FMcomms4 in future we will install other frequency baluns also and try to explore its complete wide range from 70Mhz to 6GHz . Further more stress testing of FMComms4 parameters will be performed for analysing upper and lower extreme limits.

VII. Conclusion

We have demonstrate a method for interfacing FMComms4 SDR with Digilent's ZedBoard and details of interconnecting processes. We discussed comparison of FMComms4 SDR features with various other SDR of nearly same features and found that within the low cost most of the features can be successfully covered. FMCOMMS4 utilizes the AD9634, a high performance, highly integrated RF agile transceiver on a single chip that functions on a 70 MHz to 6 GHz range and provides a maximum bandwidth of 56 MHz. With such specifications, the FMCOMMs boards constitute the most powerful hardware RF analog front-ends currently available. We also provided experiments using GNU Radio Companion and the FMCOMMS4 hardware platform that attest to the correct functionality of the proposed interface. Further evaluation with the proposed experiment running on FMComms4 SDR was analysed.

References

[1] Dr. Hikmat N. Abdullah, B.Sc. Hussein A. Hadi, "Design And Implementation Of FPGA Based Software Defined Radio Using Simulink HDL Code",in *Engineering and Technology Journal, Vol.28.No.23*,on 2010, pp. 6750-6767.

[2] John Huie, et al. , "Synthesizing FPGA Cores For Software-Defined Radio", CF-SDR031405-1.0.

[3] Roberto de Matos, Antonio Augusto Frohlich, and Leandro Buss Becker, "Using Multiple Channels to Improve SDR Flexibility and Performance",in *International Conference on Computing, Networking and Communications (ICNC)* 2012.

[4] Mohamed Salahl, Abdel-Halim Zekry, Mohammed Kamel,"An Efficient FPGA Implementation of OFDM Physical Layer for SDR-Based Applications ",in *IEEE 10th International Conference on Computer Engineering Conference (ICENCO)* 2014.
[5] Raquel G. Machado, Alexander M. Wyglinski, "Software-Defined Radio: Bridging the Analog-Digital Divide", in *Proceedings of the IEEE*, Vol. 103, No. 3, March 2015
[6] Analog devices, "Prerequisites for AD9361/AD9364 based boards" [Online]. Available: https://wiki.analog.com/resources/eval/user-guides/ad-fmcomms2-ebz/prerequisites
[7] Analog devices, "Introduction to boards based on the AD9361/AD9364" [Online]. Available: https://wiki.analog.com/resources/eval/user-guides/ad-fmcomms2-ebz/introduction
[8] USRP E313 Datasheet [Online]. Available: https://www.ettus.com/content/files/USRP-E313-Datasheet.pdf
[9] Analog devices, "AD9364 Datasheet" [Online] .Available: http://www.analog.com/media/en/technical-documentation/data-sheets/AD9364.pdf
[10] Digilent Zedboard [Online]. Available: https://reference.digilentinc.com/reference/programmable-logic/zedboard/start
[11] Matchstiq datasheet [Online]. Available: https://epiqsolutions.com/matchstiq/legacy/matchstiq-flyer.pdf
[12] bladeRF x115 [Online]. Avilable: https://www.nuand.com/blog/product/bladerf-x115/
[13] LimeSDR [Online]. Available: https://www.crowdsupply.com/lime-micro/limesdr
[14] About Linaro, "Linaro seeks to simplify ARM Linux landscape" [Online]. Article Available: https://lwn.net/Articles/391189/2010-06-09.Retrieved 2012-07-22.
[15] Analog devices, "About libiio" [Online]. Available: https://wiki.analog.com/resources/tools-software/linux-software/libiio-internals
[16] Analog devices, "AD-FMCOMMS4-EBZ Functional Overview" [Online]. Available: https://wiki.analog.com/resources/eval/user-guides/ad-fmcomms4-ebz/hardware/functional-overview
[17] Analog Devices, "AD-FMCOMMS4-EBZ User Guide" [Online]. Available: https://wiki.analog.com/resources/eval/user-guides/ad-fmcomms4-ebz
[18] ZedBoard Getting Started Guide [Online]. Available: http://zedboard.org/sites/default/files/documentations/GS-AES-Z7EV-7Z020-G-V7-1.pdf
[19] AD9361 datasheet [Online]. Available: http://www.analog.com/media/en/technical-documentation/data-sheets/AD9361.pdf