ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 60-63 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

GNU Radio Based BIST for SDR Performance Measurement

Remya P S[1], Prof. B I Neelgar[2], Shibu R M[3]
[1]GMR Institute of Technology, Rajam, AP, India, remyaspillai718@gmail.com
[2]HOD of ECE, GMR Institute of Technology, Rajam, AP, India, hod_ece@gmrit.org
[3]Principal Engineer, CDAC, Thiruvananthapuram, Kerala, India, rmshibu@cdac.in

Abstract*:* This paper is an attempt to investigate on methods to develop Built-In Self-Test (BIST) method for performance measurement of a wideband (20MHz) Software Defined Radio with frequency coverage from 50MHz to 2.4GHz. The overall quality of transmission and reception is dictated by various RF system and baseband specifications in wireless systems. Error Vector Magnitude (EVM) is a measure of the digital modulation quality of the wireless system under test which is very sensitive to much impairment in the transceiver.

Keywords*:* Built In Self Test (BIST), Error Vector Magnitude (EVM), GNU Radio, Software Defined Radio (SDR).

INTRODUCTION

Error Vector Magnitude (EVM) is a measure of modulation quality and error performance in complex wireless system. It plies a method for evaluating the performance of Software- Defined Radios (SDRs), both transmitters and receivers. Software Defined Radio technology brings the cost efficiency, flexibility and power to drive communication forward, with wide-reaching benefits realized by service providers and product developers via to end users. EVM encapsulates many non-idealities of the transmitter, including inter symbol interference, non-linearity, mismatches, phase noise and spurs, carrier leakage. It is one of the few parameters stipulated for the transmitter and required to be guaranteed.

Recently, for simplified EVM measurements, various researchers have proposed techniques. The goal in [4] is to bind the number of symbols that need to be transmitted/received within one frame as dictated by the normal operation mode. By using a golden receiver on the load board the whole operations at the receiver side is duplicated and conducted at the tester.

While there has been significant progress in reducing the overhead of EVM measurements, important challenges still remain. Especially when it comes to multi-site testing. Multi-site environments typically share high-frequency channels that complicate board design and limit the number of parallel tested devices. Moreover, characterization of EVM requires sending many symbols spread over multiple frames and conducting the whole receive operation which includes the channel characterization and synchronization. The I and Q signals necessitied to be captured and analyzed at the tester which requires access to these signals. Moreover, the test engineer has to duplicate the complete receiver and to be incorporated into the test program. In order to synchronize the transmitted and received frames and to compute the channel characteristics this operation requires complex procedures. These steps complicate the test development process and can be resulted in long test development times.

There are several sources of distortion in a typical transmission system. These types of signal distortion, though readily identifiable can be somewhat problematic to measure and to minimize subsequently. Following is a list of distortion mechanisms run into:

1. Across the bandwidth of the channel non-uniform amplitude frequency response.
2. Non-uniform group delay as a function of frequency across the bandwidth of the channel.
3. Intermodulation distortion induced by a single channel (in-channel distortion).
4. Intermodulation distortion because of the presence of other channels.
5. Noise (additive and multiplicative).
6. Multipath transmission (not applicable in wired connections).

All of the above items have an effect on transmission quality. As these imperfections increase, the probability of errors in transmission also increases.

Our suggested approach enables true multi-site testing and removes the encumbrance of designing complicated boards for EVM test.

The focus of this paper is EVM measurement to check the performance quality of transmitted and received signals of SDR. Here a built-in method using GNU Radio software is used for measuring EVM that alleviates the need for sophisticated external test equipment and speeds up test application time.

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 60-63 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

SOFTWARE-DEFINED RADIO (SDR)

Signal processing components such as modulators, demodulators, filters, are typically implemented in dedicated hardware chips. Those hardware chips are customized for a particular and static use rather than intended for general purpose use. This resulted in minimal flexibility in supporting wireless communications. However SDR delegates much of its signal processing to software components (i.e., waveform software) running on a generic hardware platform that is equipped with one or more programmable processors, such as general purpose processors (GPPs), DSPs, or field programmable gate arrays (FPGAs). Thus, SDR devices can be controlled through

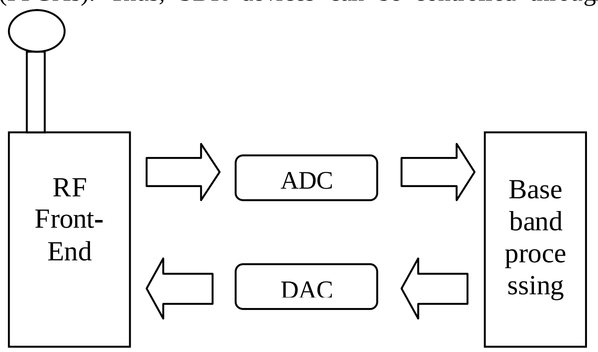


Fig 1: SDR system block diagram

dynamic programming of its waveform software and can be reconfigured and updated to improve its features, such as security, performance, and services.

A number of definitions can be found to describe Software Defined Radio, also known as Software radio or SDR. The SDR Forum working, in collaboration with the Institute of Electrical and Electronic Engineers (IEEE) has worked to establish a definition of SDR that provides consistency and clear overview of the technology and its associated benefits. Simply Software Defined Radio is defined as

"Radio in which some or all of the physical layer functions are software defined"

In SDR, signal will be processed in digital mode as in the conventional radio. The digitization work will be done by a device called Analog to Digital Converter (ADC). Fig 1 shows the concept of Software Defined Radio. The figure shows that the ADC processor comes just after the RF Front-End circuit. The use of RF Front-End is to down convert the signal to the lower frequency which is called as Intermediate Frequency (IF).For further processing such as demodulation, channel coding and source coding the ADC will pass the signal to the baseband processor after digitization. Hence SDR equipment is easier to reconfigure when compared to conventional radio equipment.

The idea of SDR was introduced first for military application. The concept became popular in the civil telecommunication field because of the innovations in enabling wideband radio frequency (RF) front-ends and soft transceivers. In terms of the requirements of communication posed by the civil, military and commercial applications SDR offers inexpensive and an effective solution to the constraints and economic viability present in the current hardware based systems. For example, cellular communication infrastructure systems are increasingly building common platform or multiband-multiprotocol base stations to support multiple cellular infrastructure standards using SDR. SDR is expected to become a prevailing technology in wireless communication.

A. Benefits

The benefits of SDR are:

1. SDR enables a family of radio products to be implemented using common platform architecture, allowing new products to be more quickly introduced into the market for radio equipment manufacturers and system integrators.
2. SDR enables the use of a common radio platform for multiple markets, significantly reducing the operation expenditures and logistical support.
3. For end users that is from business travelers to soldiers on the battlefield SDR technology targets to provide access to omnipresent wireless communication by reducing the cost.

GNU RADIO

For building Software Defined Radios, a free software toolkit called GNU Radio can be used. It is effective for signal processing from the PHY layer in a simulation like environment with readily available external RF hardware which are in low cost or without hardware, It is vastly used for learning about , building and deploying software radios, both in academic and business fields.

GNU Radio applications are mainly written in Python programming language, while the supplied, performance-critical signal processing path is implemented in C++ using processor floating point extensions, where available. Hence the developer can implement real-time, high throughput radio systems to use in a simple way, rapid-application development environment.

The GNU Radio project was founded by Eric Blossom. It supports Linux and packages are pre-compiled for the major Linux distributions. A port to windows has been also developed, but it provides limited functionalities. It performs all the signal processing and is distributed under the terms of the GNU General Public License (GPL). It can be used to write applications to receive or to push data into digital streams and an hardware is used to transmit this data.

A graphical user interface called GNU Radio Companion (GRC) helps to create signal processing applications by drag-and-drop. By writing code GNU Radio can be extended (i.e., add functionality).

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 60-63 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

ERROR VECTOR MAGNITUDE (EVM)

Error Vector Magnitude or EVM is a measure of modulation quality and error performance in complex wireless systems. It provides a method to evaluate the performance of SDRs, both transmitters and receivers. That is to measure how accurately a radio is transmitting symbols within its constellation.

EVM basically defines the amount of received symbol from its ideal location in the complex IQ plane. For example Fig 2a explains the constellation diagram of QAM16. In that constellation points are shown as circles and they are the ideal positions of the symbol constellations. Points that are shown by plus (+) signs represent symbols that are affected due to noise mechanism or unwanted distortion. As a result of impairments or induced noise the constellation points changes. Thus, by comparing the received symbol constellations with their ideal locations, the quality of received signals can be estimated. Fig 2b focuses on one symbol from the constellation diagram. The vector between the ideal symbol location and the received symbol location is called the error vector and the magnitude of that is known as error vector magnitude or EVM.

BUILT-IN SELF-TEST (BIST)

BIST is a type of mechanism that allows a machine to test itself. If we install the code in hardware it can test the quality of channels by itself and hence can improve the performance. The main function of BIST is to reduce depending upon external test equipment like VSA in this case. It also decreases the complexity and thereby cutting down the cost.

PROPOSED METHOD

The goal of the design strategy is to minimize computational overhead of computing EVM. In other words the focus is to reduce or eliminate the complex operations such as divisions, squaring etc. Several types of operations such as arithmetic operations, square/square root and memory for storing intermediate result are needed for EVM computation.

The instantaneous EVM value is affected immensely by measurement errors and thermal noise. This stochasticity can be rather high hence instantaneous EVM value is typically of little use. Therefore a more useful metric is the root-mean-square (RMS) of the EVM for a number of symbols. As a matter of fact, based on its RMS value many standards define the EVM specification.

$$EVM = \sum_{i=1}^{N} \sqrt{[(I - I_0)^2 + (Q - Q_0)^2]} \qquad (1)$$

Where, N is the number of samples, I and Q are the position of received symbols, I_0 and Q_0 are the ideal position of the symbols I and Q.

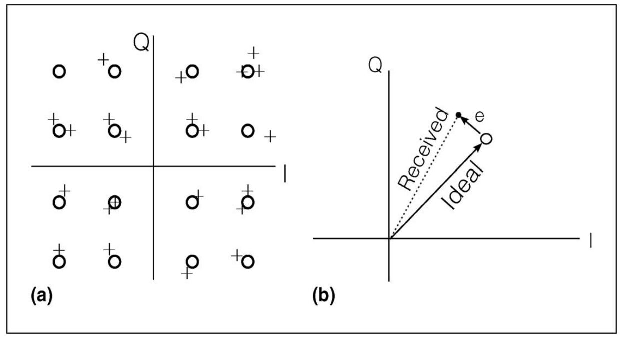


Fig 2: a) Constellation diagram for QAM16 b) Definition of EVM

The start is with the original equation (1) and reduces it to generate a more affordable approximation of the equation. The equation calculates the square of original EVM. But we use the squared version of the EVM specification limit instead of computing the square root of the estimate because using squared value of a known constant is less expensive when compared to the calculation of square root of a variable. Here this manipulation simply converts a square root-and-sum operation to a sum-and-square root operation and finally a sum-of-squares operation. Since the frame to frame correlation is zero changing the order has negligible effect.

$$EVM = \sum_{i=1}^{N} [(I - I_0)^2 + (Q - Q_0)^2] \qquad (2)$$

Based on the above equation i.e., (2) an out of tree module is designed.

GNU Radio includes a library of signal processing blocks which are used to construct a radio. It requires the USRP (Universal Software Radio Peripheral) to receive real radio waves or to transmit. It provides the possibility to use a pre-recorded file as input also. In order to test the out of tree module which was designed, named as, 'EVM measurement', a flow graph is set up as shown in Fig 3.

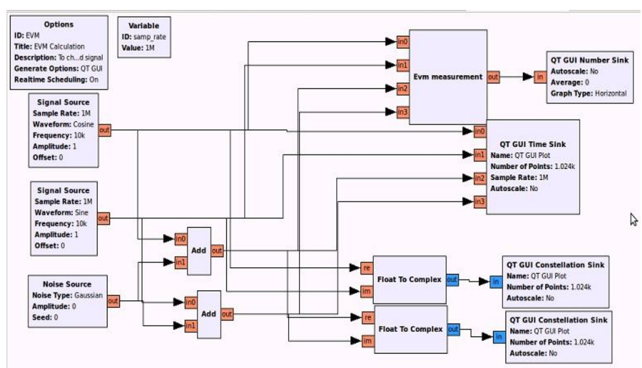
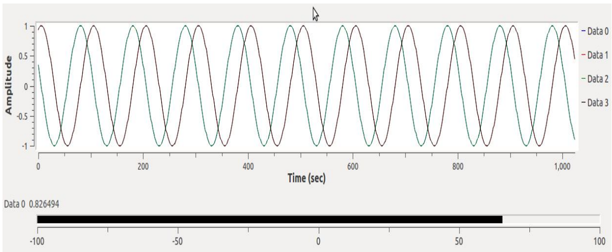
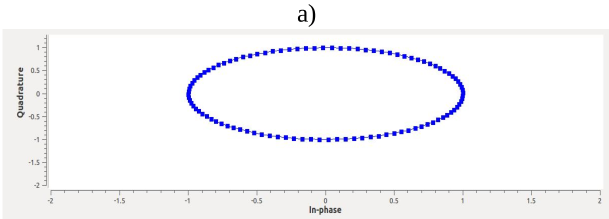


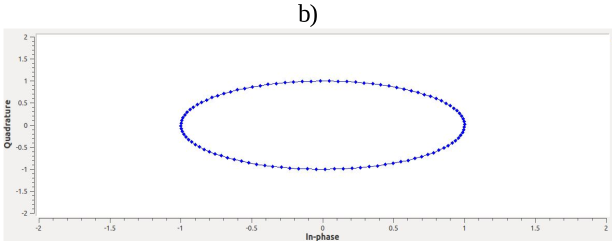
Fig 3: Test set-up for EVM measurement block.



a)



b)



c)

Fig 4: a) The input and EVM value b) Constellation plot of noised I c) Constellation plot of noised Q.

EXPERIMENTAL RESULT

The resultant plots are as shown in Fig 4. In this experiment the value of N is taken as N=100. The EVM value obtained was 0.82% which means the error is low and quality is good. Instead of using the pre-recorded file as input an SDR can be connected to PC to estimate its EVM value by using out of tree module.

CONCLUSION

In this paper, an EVM measurement method has been introduced by using GNU Radio software, open-source software. EVM needs complex operations that are to be implemented at the tester, which is a great burden. A groundbreaking method is proposed for checking the quality of received signals with very little overhead. This finds an application in satellite integration & testing, and results in major reduction in test time and provides diagnostic information. This also has application in cognitive radio communication. Another important application is in Adaptive Modulation and Coding (AMC) of next generation communication system. This has expansion possibilities in the ASIC Design of RF SoC.

ACKNOWLEDGEMENT

I express my sincere gratitude to the SEG Department of CDAC, Thiruvananthapuram, Kerala for providing the required assistance and extending all the facilities of the department for carrying out my work successfully. I am also evincing my earnest gratitude to my guide and all the faculties of EC dept. for plying necessary help during the course of my work.

REFERENCES

[1] Ender Yilmaz, Afsanesh Nassery and Sule Ozev, "Built-in EVM measurement with negligible hardware overhead", Design & Test, IEEE, vol. 31, Issue:1, 2014, pp. 75-82.

[2] Overdorf. R, "Testing methods and error budget analysis of a softwaredefined radio", Militiary Communications Conference, IEEE, 2008, pp.1-6.

[3] El Kassir, B.Kelma.C, Jarry.B, campovecchio.M, "Built in self test for error vector magnitude measurement of RF transceivers", International Test Conference, IEE, 2009.

[4] E.Acar, S. Ozev, G. Srinivasan, and F.Taenzler, "Optimized EVM testing for IEEE 802.11a/n RF ICs", in Proc. 2008, IEEE, Int. Test Conf., 2008, pp. 1-10.

[5] Ender Yilmaz, Afsanesh Nassery, Sule Ozev and E.Acar, "Built-in EVM measurement for OFDM tranceiversusing all-digital DFT", IEEE, Int. Test Conf. 2009, pp. 1-10.

[6] Afsaneh Nassery, Sule Ozev, Marain Verheltz and Mustapha Slamani, "Extraction of EVM from transmitter system parameters", IEEE Computer Society, Sixteenth IEEE European Test Symposium, 2011, pp. 75-80.

[7] http://ieeexplore.ieee.org/.

[8] http://gnuradio.org/redmine/projects/gnuradio/wiki.

[9] en.wikipedia.org/wiki/Software-defined radio.