A New Design Process to Reduce Resource Usage in SDR Systems

Sameh Yassin, Islam R Kamel, Hazim Tawfik
Department of Electronics and Communications
Cairo University Faculty of Engineering
Cairo, Egypt
syassin@eece.cu.edu.eg, islam.kamel@eng.cu.edu.eg, Hazim.Tawfik@eece.cu.edu.eg

***Abstract*—Software Defined Radio (SDR) is a recent trend in communication systems. Its primary goal is to flexibly handle different technologies using the same hardware platform by applying different software. To achieve this goal efficiently, the resource utilization should be minimized by implementing algorithms in fixed point arithmetic. This work is focused on the minimization of resource utilization by reducing the data width of the mathematical quantities inside the receiver chain. A design process is proposed to optimize the computational accuracy. The design process is used to link system performance and computational accuracy using simulation. The simulation approach is chosen because of its ability to cope with functional changes in SDR systems, and to satisfy the short time to market requirement. To verify the results, a case study is considered by reducing the resource usage in the SDR platform known as Universal Serial Radio Peripheral (USRP). The functions with highest resource utilization are implemented in fixed point. In addition, the optimized functions within the case study are implemented using Hardware Description Language (HDL) on a Field Programmable Gate Array (FPGA), to verify experimentally the reduction in resource utilization due to the proposed design process.**

Keywords*—*computational accuracy; finite precision; software defined radio; signal processing; USRP.

I. Introduction

Software Defined Radio is a flexible platform that can provide dynamic reconfiguration using software. In other words, the same hardware can be used to implement different transceiver functions, such as modulation, detection, and channel estimation. One motivation for the rapid development of SDR systems, is the need to to add new features to the radio equipment or to upgrade its functionality. This is needed by important sectors such as military and public safety [1]. In these sectors, special purpose radios were the norm rather than the exception. One radio device is needed for few number of functions or wave forms, and it is not straightforward to alter the device functionality.

A SDR system is a generic communication system starting with user data layer and ending with physical data layer [2]. The generality of a SDR system is due to its ability to reconfigure the system modules without changing the hardware. This is obtained by adding control to a communication system, as shown in Figure 1. A SDR system consists of radio front end, base band processing, control bus, and application. These modules are combined to map the required data from the application into physical signal at arbitrary carrier frequency. The application is modified during design time and/or execution time by sending reconfiguration messages through the control bus. Hence, to modify the functionality of the system, only the application is updated using software configuration.

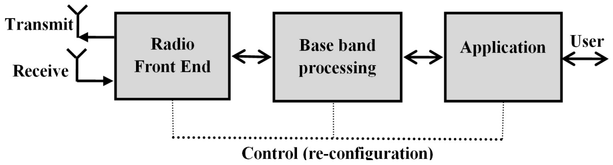

Fig. 1. Software Defined Radio Aspects

In SDR, it is frequently desired to add features to an existing system. However, there are three considerations while adding new features, namely execution time, power consumption, and resource utilization. The main focus of this work is to evaluate the increase in resource usage due to these added features. In addition, the increase in the resource usage should be optimized in order not to affect the system performance. In the next paragraphs, we are going to describe the related work that address computational accuracy and their effects on system performance.

In [3], the main motivation is to reduce the energy consumption, and the complexity requirements to satisfy an embedded system needs. Nguyen et al. use fixed point arithmetic to present an optimized digital receiver. The merit for this work is finding an analytical relation between the receiver performance, and the number of bits needed to satisfy energy consumption constraints. Using this approach, the minimum data width can be estimated with negligible performance degradation compared to floating point arithmetic. However, to deduce the minimum data width one can not reach closed form expression for some equations. For SDR systems, it is impractical to update the performance equations each time the system functions are modified.

In [4], Novo et al. highlighted the need to implement complex signal processing algorithms in fixed point. A metric was developed to compensate for the loss in accuracy due to the conversion from floating point to fixed point. That metric is system dependent, and should be dimensioned carefully for each application.

In [5], the concept of scalable SDR is introduced for battery powered devices. To achieve the two contradicting requirements of fast time to market and minimum energy consumption, a new method is presented to deal with fixed point arithmetic. The new method considers the changes in data format, such as modulation scheme, and number of antennas, and accordingly data width can be adjusted to save energy consumption of the battery. However, the savings in power consumption come at the expense of increased resource utilization.

The rest of this paper is organized as follows. In Section II, the problem formulation due to finite precision is highlighted. Then, a design process is proposed to discover the minimum computation cost required that maintains the system performance in terms of Bit Error Rate (BER). In Section III, a case study is presented to validate the proposed design process. In Section IV, computer simulations are developed to evaluate the system performance. In addition, the case study is implemented in FPGA to verify the simulation result experimentally. The paper is concluded in Section V.

II. Problem description due to finite number representation

A. Number Representation

The use of fixed point numbers to implement algorithms has some limitations. Due to the finite resources of machine number system, there exist many unavoidable issues such as overflow, underflow, scaling. There also exist some errors such as round-off error, and quantization error.

Due to these issues, the implementation of algorithms in fixed point representation can affect system performance considerably. There are two approaches to evaluate the degradation in system performance, namely analytical, and simulation. In this work, the simulation based approach is chosen because it is more adequate to the nature of programmable SDR systems in terms of supporting new features in short design cycle, and coping with fast market changes.

We propose a new process to design any new SDR system, or add features to an existing one. The added value of this process is the link between system performance and computational accuracy. A similar method is proposed Mehard et al. [6] using an analytical approach. The analytical approach was chosen there because it needed less execution time. However, obtaining closed form expressions for each application is not straightforward, and may have to be solved numerically. Therefore, the increase in problem complexity may outweigh the decrease in execution time. Moreover, SDR need to be flexible to changes in system functions. It is impractical to update the performance equations each time the system functions are modified.

B. Proposed Design Process

The aim of the design process is to calculate the optimum data width that can be used without affecting the system performance. A metric will be used to compare between system performance using floating point, and fixed point representation. The data width is optimized by allocating proper number of bits to the integer part i_{wl} and the fraction part f_{wl}. The proposed design process is outlined in the flowchart shown in Figure 2.

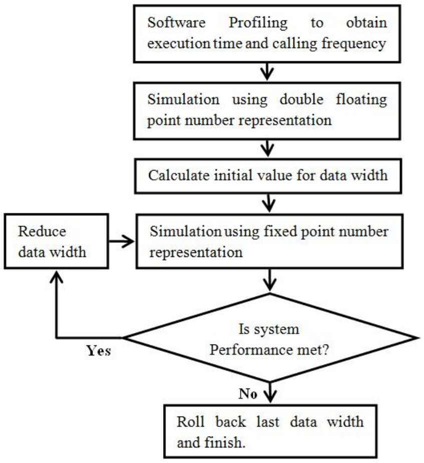
Software Profiling to obtain execution time and calling frequency
Simulation using double floating point number representation
Calculate initial value for data width
Reduce data width
Simulation using fixed point number representation
Is system Performance met?
Yes
No
Roll back last data width and finish.

Fig. 2. Flow chart for the proposed data width minimization process

The first step is to determine the utilization of processing power by each system function. This is achieved using software profiling. For each system function, calculate its execution time and the number of times it is called. Both are needed to avoid optimizing one function with high execution time when it is only called few times. The second step is to simulate the system performance using double floating point numbers, which is efficient when a large dynamic range is required [4]. The dynamic range is defined as the ratio between largest and lowest signal amplitudes. The performance metric is chosen to be either BER [6], or Signal to Noise Ratio (SNR) [3]. In this work, BER is used for comparison because it can be accurately measured. The third step, is to determine an initial data width for fixed point simulation as will be discussed in Subsection II-C. This initial value will be optimized in the following steps. The fourth step, is to simulate the system and obtain performance curves using fixed point data width. The fifth step, is to compare between the BER curves obtained in steps two and four. By comparing both BER curves, the required increase in SNR to obtain the same BER value can be calculated. The required increase in SNR ϵ can be calculated using (1).

$$\epsilon = SNR_{fp} - SNR_0, \qquad (1)$$

where SNR_0, SNR_{fp} are the SNR for floating point, and fixed point, respectively. The value of ϵ is a design parameter, that is chosen arbitrarily. When the system is desired to have the same BER value for both fixed point and floating point, ϵ is set to 0 dB. For further savings in resource usage, ϵ is increased. The fifth step proceeds by increasing SNR_{fp}. If the difference $SNR_{fp} - SNR_0$ is still smaller than ϵ, then reduce the data width (i_{wl}, f_{wl}). Otherwise, roll back previous value of data width and proceed to the end of the process. Note that the data width is reduced gradually first by reducing the integer part i_{wl} by one bit at a time, until its minimum value is obtained. Afterwards, the fraction part f_{wl} is reduced by one bit at a time until its minimum value is obtained.

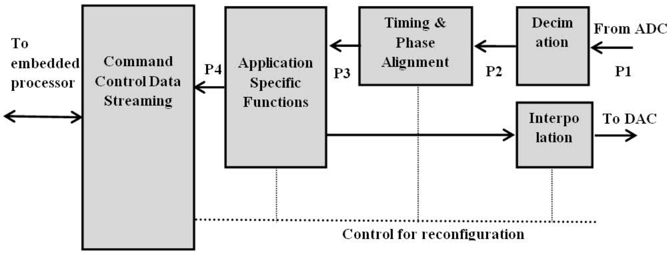

Fig. 3. Signal flow graph for receiver chain in generic SDR system

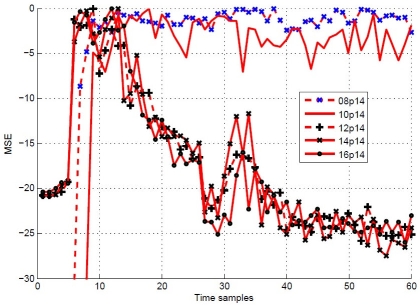

Fig. 4. MSE of ensemble of 50 trials. The integer part is variable

C. Dynamic range estimation

To estimate the initial value of data width a signal flow graph for the SDR system under development should be constructed, as shown in Figure 3. In related work [3][5], the data width and dynamic range are usually estimated for the system at one point. In this work, it is proposed to estimate the data width at each block's output. Afterwards, the dynamic range can be estimated for both receiver and transmitter chains. The initial value for data width will be calculated for receiver chain because it is more complex than the transmitter in terms of the required number of functions. Without loss of generality, the same method can be applied to transmitter chain to calculate its initial data width. For the receiver chain, the points of interest is highlighted and marked as P_i where $1 \leq i \leq N$, where N is the number of blocks with different data width. Next, the initial value of data width can be estimated by observing each point in the receiver chain. At point P_1, the data width is the precision of the front end Analog to Digital Converter (ADC), such that

$$wl_{P_1} = Accuracy_{ADC}. \tag{2}$$

At point P_2, the data is prepared for base band processing by reducing its sampling rate using the decimation process. The input stream sampling frequency is reduced by an integer factor called decimation factor R, even though only the sample rate is changed, and not the bandwidth of the signal. The input signal bandwidth must be filtered to avoid aliasing. Therefore, the decimation process requires an increase in the data width to maintain proper number of bits per sample. The increase in data width can be calculated, depending on the decimation method of choice. One example is the Cascaded Integrated Comb (CIC) filter. The input signal is fed through one or more cascaded integrator sections, then a down sampler, followed by one or more comb sections [7]. The increase in the data width will be dependent on the differential delay M of the comb section, and the number of blocks N as in (3), where $Ceil(A)$ rounds to the nearest integer greater than or equal to A.

$$wl_{P2} = wl_{P1} + Ceil(N * log_2(M * R)). \tag{3}$$

At point P_3, the timing and phase changes relative to the original transmitted signal are estimated. This includes correlation operation which is composed of addition, shift, and multiplication operations. The multiplication operation particularly leads to the most significant increase in the input data width which is proportional to the multiplier $n1$, and the multiplicand $n2$ as shown in (4).

$$wl_{P3} = wl_{P2} + \sum(n1 + n2 - 1). \tag{4}$$

At point P_4, application specific functions can also increase the input data width, and may alter the numerical stability due to the use of fixed point such as channel equalization. The data width for this block must maintain numerical stability of the SDR system. In case of channel equalization, the data width should result in quantization noise power that will not alter the computation of the Mean Square Error (MSE) of the equalization algorithm [8]. To illustrate how to maintain numerical stability of equalization algorithm, one channel equalization algorithm, namely the Recursive Least Squares (RLS) is considered.

To maintain numerical stability of RLS, data width should be increased as shown in Figure 4. A family of MSE curves is obtained for different data widths ranging from ($i_{wl} = 8$, $f_{wl} = 14$) bits, to ($i_{wl} = 16$, $f_{wl} = 14$) bits. This family of curves can be used to conclude that the minimum value is $i_{wl} = 12$, because it keeps MSE decreasing as the time samples advance. To obtain the minimum value for fractional part $f_{wl} = 14$, a similar family of curves is developed but with fixed integer part, and variable fractional part. Note that stabilizing the algorithm will increase the data width. Therefore, fast fixed order filters can be used to reduce the data width [8]. To choose an equalizer algorithm with lowest resource utilization, a fair comparison between different equalizer algorithms was proposed in [9]. Yassin and Tawfik proposed to weight the resource usage of equalizers by mapping the algorithm in terms of mathematical operations. This fair choice will compensate for the increase in data width. The data width for this block can be calculated as in (5), where δ is the required increase in the data width to maintain the numerical stability.

$$wl_{P_4} = wl_{P_3} + \delta. \tag{5}$$

The value wl_{P_4} can be used initially for the design process. To validate the design process, a case study will be considered in Section III.

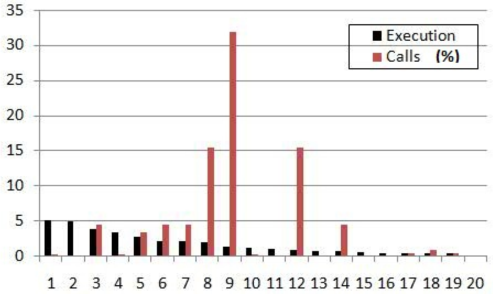

Fig. 5. The percentage of processor execution time, number of calls against the function identities

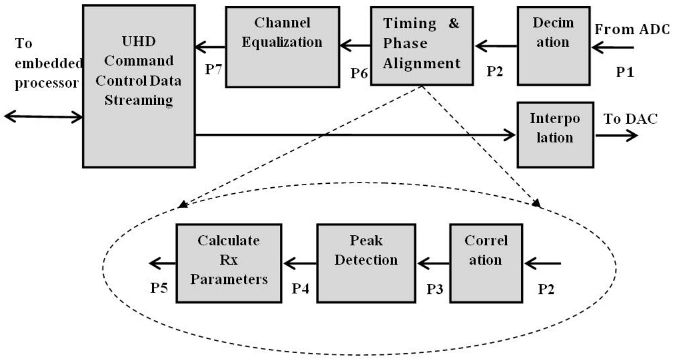

Fig. 6. Signal flow graph for the block under test "Analyze Traffic Burst"

III. Case study: Open BTS receiver chain

The case study is part of the Open Base Transceiver Station (OpenBTS) project [10], which is based on ETTUS research platform named Universal Serial Radio Peripheral (USRP). This platform provides a cheap alternative to standard BTS [10], [11]. Hence, rural communities can enjoy cheap and basic telecommunication services using Global System for Mobile communications (GSM).

The first step, is to perform software profiling for the OpenBTS system as shown in Figure 5. The function identities of the OpenBTS system appear on the x axis, while the y axis shows both the percentage of processor execution time, and the normalized number of calls of each function. As previously mentioned, only those functions with high execution time and high calling frequency should be optimized. It can be observed from Figure 5 that the functions with identities (8, 9, and 12) have the highest utilization. These functions are grouped into one large function named "Analyze Traffic Burst". The purpose of the function is to calculate important parameters of the receiver, namely Time of Arrival (ToA), Valley Power (VP), and channel estimation coefficients. ToA, and VP are used for synchronization between mobile station and base station. The second step, is to simulate system performance using double floating point representation. The results of this step will be detailed in Section IV.

The third step, is to calculate an initial data width for simulation. A block diagram is constructed for the "Analyze Traffic Burst" function as shown in Figure 6. At $P1$, the accuracy of the ADC component in USRP is 12 bits. Then the input data width equals the ADC accuracy $wl_{P1} = 12$. At $P2$, decimation is implemented using CIC filter with $N = 4$ stages and a variable decimation rate $log_2(M \times R) = 7$ resulting in $wl_{P2} = (12 + 28)$. However, the original OpenBTS design chooses to truncate this value to be $wl_{P2} = (12 + 12)$ to save resource usage. This truncation will not affect the design process, the truncation can be ignored and same results will be obtained. At $P3$, the data width will be doubled due to the multiplication in the correlation function resulting in a data width of $wl_{P3} = (24 + 24 - 1)$. At $P4$, the peak detection is divided into two sub blocks, namely Coordinate Rotation Digital Computer (CORDIC) and interpolation. The basic operation is to rotate the x-y axes of the input data to eliminate the y component. Hence, the magnitude of the input vector is stored in the x component only. This will in turn reduce the required resources. On the contrary, the interpolation operation contains multiplications. Therefore, the data width will be proportional to the width of the multiplier and multiplicand. The multiplier is the output of the previous stage, while the multiplicand is a SINC function that is read from memory element. A SINC function has maximum integer value of 1. Hence, it can be represented with minimum accuracy, and it is chosen to be the same as $wl_{P2} = 24$. Now, the data width of multiplier equals $wl_{P3} = 47$, and the data width of multiplicand is 24. Therefore, the initial value for data width in the simulation will be $wl_{P_4} = (47 + 24) = 71$ bits.

The initial value will be optimized by repeating steps four and five until the performance condition is violated. This will be discussed in the following section.

IV. Simulation Results

The results obtained in Section III are validated by developing a simulation model using Mathworks tool Simulink Matlab $v7.12$ [12]. The model consists of transmitter, channel, and receiver. The transmitter consists of random data source generator, GSM burst formatter, and digital up converter. The channel is a Rayleigh fading channel, with variable fractional delay, and Additive White Gaussian Noise (AWGN). This channel model [14] can be used to simulate the system for the cases of typical urban, and rural areas. The receiver consists of digital down conversion, timing and phase alignment, channel equalization, burst deformatter, and bit error rate calculator.

The simulation is used to calculate BER curve for different values of SNR as shown in Figure 7. The graph with solid line and 'o' marker is calculated when all variables are represented in floating point. The design process starts with $\epsilon = 0.2dB$ to obtain system performance similar to the case of floating point arithmetic. This results in a minimum data width ($i_{wl} = 62$, $f_{wl} = 4$) bits, which is shown in the curve with the solid line and 'x' marker. It can be observed that for the same BER value, the SNR difference between the two curves is less than $1\ dB$. This was expected because ϵ is small. If the system performance is relaxed to reduce resource utilization, the same process can be run again with $\epsilon = 2\ dB$, resulting in data

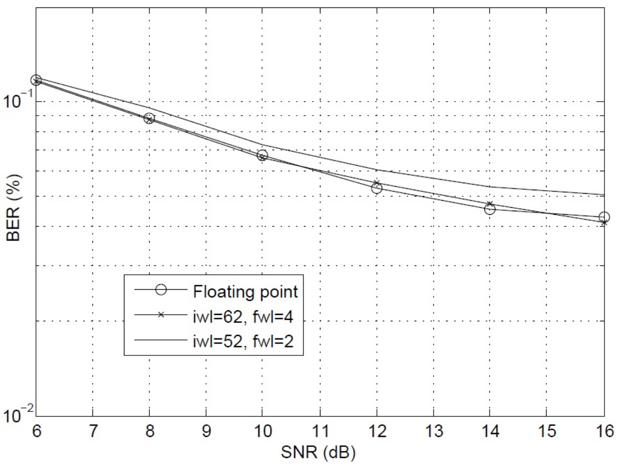

Fig. 7. BER curves for both floating point and fixed point

TABLE I
FPGA RESOURCE UTILIZATION FOR THE CASE STUDY

Unit	$i_{wl} = 52$			$i_{wl} = 62$		
	Used	Available	%	Used	Available	%
Slices	7538	23872	31	9548	23872	40
Flip Flops	9273	47744	19	15278	47744	32
4-LUTs	14074	47744	29	18620	47744	39
BRAMs	7	126	5	7	126	5
GCLKs	2	24	8	2	24	8
DSP48s	57	126	45	57	126	45

width of ($i_{wl} = 52$, $f_{wl} = 2$) bits. It should be mentioned that the advantage of using fixed point is two fold. First, because the data width can be minimized at different points of the receiver chain. Second, all fixed point mathematical operators will consume less resources than floating point operators [13].

To validate the results experimentally, a simulation is run on the design after mapping to physical FPGA cells. This is called timing simulation. Xilinx's design tools $v14.1$ are used, namely ISE, and ISIM. The USRP is equipped with Xilinx's FPGA named Spartan 3A-DSP 3400. The resource utilization due to the mapped design is reported in Table I. When the design is implemented using floating point arithmetic, the synthesis operation fails to map the design into the FPGA. This is expected because a floating point multiplier can consume one, or few FPGA units [9][15]. After applying the design process, the design can fit the into the FPGA with resource usage that is found to be less that 50%. The DSP48 blocks have relatively high utilization 45%, because they are required to implement multiplication operation without using the logic slices of FPGA. Note that, the multipliers have 36 bits operands, and hence they can be used for both cases of ($i_{wl} = 62$, $f_{wl} = 4$) and ($i_{wl} = 52$, $f_{wl} = 2$), with the same resource utilization. Finally, the experimental results verify the validity of the proposed design process.

V. CONCLUSION

SDR has a desirable nature of adding new features by reconfiguration. However, this will increase the resource usage and may affect system performance. One solution is to implement algorithms in fixed point number representation. In this work, a new design process is proposed to link between system performance and computational accuracy using simulation. To validate the proposed process, a case study of the OpenBTS project is considered. It was not possible to implement the case study into FPGA without applying the proposed process, while maintaining system performance. Moreover, the system performance was relaxed to obtain more savings in resource usage. Finally, the results were verified experimentally using FPGA implementation. It was shown that the utilization of FPGA did not exceed 50 % of the available resources .

ACKNOWLEDGMENT

This work is supported by the Egyptian National Telecommunication Regulatory Authority (NTRA), as part of the project "A BTS for Rural and Underdeveloped Population Tier" (ABRUPT).

REFERENCES

[1] J. Bard and V. Kovarik "Software Defined Radio The Software Communications Architecture", First Edition, Wiley 2007, ISBN 978-0-470-86518-7

[2] F. Jondral, "Software Defined Radio Basics and Evolution to Cognitive Radio", EURASIP Journal on Wireless Communications and Networking, vol. 2005, no.3, Aug. 2005, pp. 275283.

[3] H. Nguyen, D. Menard, and O. Sentieys, "Design of Optimized Fixed-point WCDMA Receiver", European Signal Processing Conference, Aug. 2009, pp. 993-997.

[4] D. Novo, M. Li, B. Bougard, L. Perre, and F. Catthoor, "Finite Precision Processing in Wireless Applications", DATE09, April 2009, pp. 1230-1233.

[5] D. Novo, B. Bougard, A. Lambrechts, L. Van der Perre, and F. Catthoor, "Scenario Based Fixed-point Data Format Refinement to Enable Energy scalable Software Defined Radios", DATE08, March 2008, pp. 722-727.

[6] D. Menard, R. Serizel, R. Rocher, and O. Sentieys, "Accuracy Constraint Determination in Fixed-Point System Design", EURASIP Journal on Embedded Systems, vol. 2008, no. 1, Jan. 2008, pp. 1-12.

[7] E. B. Hogenauer, "An Economical Class of Digital Filters for Decimation and Interpolation", IEEE Transactions on Acoustics, Speech and Signal Processing(ASSP), vol. 29, no. 2, April 1981, pp. 155162.

[8] H. Sayed, "Fundamentals of Adaptive Filtering", Second Edition, IEEE Press, John Wiley and Sons, New Jersey, ISBN 0-471-46126-1, 2003

[9] S. Yassin, and H. Tawfik,"Reduced Complexity Decision Feedback Channel Equalizer using Series Expansion Division", Advanced International Conference on Telecommunications, June 2013, pp. 219-223.

[10] D. Burgess and S. Harvind, "The OpenBTS Project.", http://openbts.org, Jan. 2014.

[11] M. Ettus, USRP Users and Developers Guide., Ettus Research LLC, 2005.

[12] J. G. Proakis, M. Salehi, and G. Bauch, "Contemporary communication systems using MATLAB", Third edition, CengageBrain. com, 2012

[13] M. J. Flynn and S.F. Oberman, Advanced Computer Arithmetic Design", 2nd ed., New York: John Wiley and Sons, 2001, pp. 113-125

[14] B. Bjerke, J. Proakis, K. Martin Lee, and Z. Zvonar, "A Comparison of GSM Receivers for Fading Multipath Channels with Adjacent- and Co-Channel Interference", IEEE Journal on selected areas in communications, vol. 18, Nov. 2000, pp. 2211-2219.

[15] A. Ramesh, A. Tilak, and A. Prasad, "An FPGA based high speed IEEE-754 double precision floating point multiplier using Verilog", International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics, and Telecommunication System, Jan. 2013, pp. 1-5.