International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2018): 7.426

# Implementation of LMS and RMS Based Adaptive Noise Cancellation Using Xilinx FPGA

**Athira Sivanandhan**

Assistant Professor, Department of Applied Electronics and Instrumentation Engineering, Mount Zion College of Engineering, Kadammanitta, Kerala, India

**Abstract:** ***Adaptive filtering is an important signal processing area that has wide applications in communications, control, and biomedical engineering fields. The adaptive noise cancelling, adaptive equalization of data transmission channels and adaptive antenna arrays are some of the examples of such applications. Adaptive filtering consists of a digital filter whose weights are controlled by an adaptive algorithm so on minimize the difference between the filter output and a reference signal consistent with some criterion. The character of the reference signal depends on the considered application. There are two main measures for evaluating the performance of an adaptive filter: the convergence rate and therefore the steady state mean square error. In practical applications, it's desired to maximize the convergence rate and minimize the steady state mean square error. There is a conflict between these requirements. Several adaptive algorithms have been developed so as to yield a good compromise between these requirements. The important adaptive algorithms are Sample Matrix Inversion (SMI), Least Squares (LS), and Recursive Least Squares (RLS) algorithm. The main objective of this project is the implementation of LMS and RLS (Recursive Least Square) adaptive filters algorithm using Xilinx system generator. The simulation of the models will be done in Matlab and Simulink for the efficient verification of the algorithm. The core RLS and LMS adaptive filtes and its basic components block will be developed in Xilinx System Generator and implementation in Xilinx FPGA.***

**Keywords:** Least mean square algorithm (LMS), Recursive Least Square algorithm (RLS),Xilinx system generator(XSG), simulink, Spartan -3

## 1. Introduction

Adaptive filter is an important part of DSP applications where the statistics of the incoming signals are unknown or changing. Adaptive filter relies for its operation on a recursive algorithm, which makes it possible for filter to perform satisfactorily in an environment where complete knowledge of the relevant signal characteristics is not available. A wide variety of adaptive algorithms have been developed for the operation of adaptive filters.

Adaptive algorithms are used in many fields of human activities. Many adaptive algorithms have been devised, described and implemented in the last 50 years. They are implemented in hardware devices or software programs to adapt parameters of behavior of the system in unknown or time-varying conditions in the application or its environment. More specifically, in control and digital signal processing (DSP) systems they are used for their ability to change the behavior of the controller or filter according to the incoming signals and the environment of the system. The most frequent applications of adaptive algorithms in these domains are system identification, noise and echo cancellation and signal enhancement.

There are some algorithms used to adjust the weight among them are the LMS (Least Mean Square) and the RLS (Recursive Least Square). The standard or modified LMS algorithm is usually used in DSP applications where up to hundreds of parameters are adapted. The main advantage of the LMS algorithms is their simplicity, and so their implementation is computationally simple with the computational complexity *O(n)* (in other words they are fast). On the other hand, their main disadvantage is a slow convergence rate. Therefore many applications use the RLS algorithm and its modifications.

The convergence rate of the RLS is far superior to that of the well-known least mean square (LMS). The recursive least square (RLS) algorithm is a recursive form solution of the minimum mean square problem. This algorithm is more complicated, and in general its computational complexity is $O(n^2)$. Because the RLS algorithm is recursively computed in finite-word length arithmetic, it can suffer from numerical instability of updated statistics due to round-off errors. To avoid this inconvenience the computed statistics aren't updated directly, but a decomposition of covariance matrix is updated. The most frequently used decomposition is QR which is very suitable for parallel and pipelined processing. Therefore it is often used when implementing RLS in hardware parallel structures like field programmable gate arrays (FPGAs).

Field programmable gate arrays (FPGAs) are widely used in many areas such as audio and video, signal processing, automotive electronic, digital communication systems, etc. Because of their high performance and flexibility, FPGAs are gradually replacing ASICs and DSPs in many application fields. One of the advantages of FPAGs-based embedded systems is their ability to integrate customized user cores with a soft or hard embedded processor in system-on-a-chip (SoC) solutions. However, programming FPGAs using Hardware Description Languages (HDL) is too time consuming and needs background in a chip design. This situation has been changing over the last decade with the emergence of a new class of high-level programming tools and languages for FPGA design: Handel-C, AccelDSP, Xilinx System Generator for DSP, etc.

Xilinx System Generator (XSG) is high-level software tool that enables the use of MATLAB/Simulink environment to create and verify hardware designs for Xilinx FPGAs quickly and easily. It provides a library of Simulink blocks bit and cycle accurate modeling for arithmetic and logic functions, memories, and DSP functions. It also includes a code generator that automatically generates HDL code from the created model. Generated HDL code can be synthesized and implemented in the Xilinx FPGAs. The XSG blocks are like standard Simulink blocks except that they can operate only in discrete-time and fixed-point format.

Adaptive noise canceller is an interesting application of adaptive filter that has been used in a wide range of applications. The proposed work presents architecture of a LMS-based and RLS based adaptive noise cancellation and these architectures are implemented on FPGA using SPARTAN-3 development board and Xilinx System Generator (XSG).

## 2. Literature Survey

The LMS algorithm was devised by Widrow and Hoff , 1959 in their study of a pattern recognition scheme known as the adaptive linear element, commonly referred to in the literature as the adaline [1]. The LMS algorithm is a stochastic gradient algorithm in that it iterates each tap weight of a transversal filter in the direction of the gradient of the squared magnitude of an error signal with respect to the tap weight. As such the LMS algorithm is closely related to the concept of stochastic approximation developed by Robbins and Monro in statistics for solving certain sequential parameter to control the correlation applied to each tap weight from one iteration to the next, where as in stochastic approximation method [2] the step size parameter is made inversely proportional to time *n* or to a power of *n*.

Least squares technique constitute an integral part of modern signal processing and communication methodology as used in adaptive filtering, beamforming, arrray signal processing, channel equalization etc. Least square algorithms, which solve the least squares minimization problem exactly, offer an alternative to the LMS algorithm. These algorithms provide faster convergence rates and unlike LMS algorithms, the LS algorithms are insensitive to the underlying covariance matrix. The efficient implementation of the LS algorithm particularly the recursive LS (RLS) algorithm, is needed to meet the high throughput and speed requirement of modern signal processing.

Recursive Least Squares (RLS) algorithm is invented by Plackett in 1950. It plays a major role in estimation theory for signal processing and online regression in machine learning[3, 4]. It is derived as the natural extension of the method of least square algorithm. The derivation was based on a lemma in matrix algebra known as the matrix inversion lemma [5]. It is well known that recursive-least-squares(RLS) algorithms produce a faster convergence speed than stochastic gradient descent techniques, such as the basic least-mean-squares(LMS) algorithms[6,7].

The initial work on adaptive echo cancellers started around 1965. It appears that Kelly of Bell laboratories was the first to propose the use of an adaptive filter for echo cancellation, with the speech signal itself utilized in performing the adaptation; Kelly's contribution is recognized in 1967 for echo cancelation application[8].

The adaptive line enhancer was originated by Widrow and his coworkers at Sanford University. An early version of the device was built in 1965 to cancel 60-Hz interference at the output of an electrocardiographic amplifier and recorder which is described in the paper by Widrow [9].

The adaptive echo canceller and the adaptive linear enhancer, although intended for different applications, may be viewed as examples of the adaptive noise canceller. This scheme operates on the outputs of two sensors: a primary sensor that supplies desired signal of interest buried in noise and a reference sensor that supplies noise alone. It is assumed that the signal and noise at the output of the primary sensor that are un correlated and the noise at the output of the reference sensor is correlated with the noise component of the primary sensor output.

The adaptive noise canceller consists of an adaptive filter that operates on the reference sensor output to produce an estimate of the noise, which is then subtracted from the primary sensor output. The overall output of the canceller is used to control the adjustments applied to the tap weights in the adaptive filter. The adaptive canceller tends to minimize the mean-square value of the overall output, thereby causing the output to be the best estimate of the desired signal in the minimum-mean-square error sense.

It is well known that two of most frequently applied algorithms for noise cancellation [10] are least mean squares (LMS) [11] and recursive least squares (RLS) [12] algorithms. Considering these two algorithms, it is obvious that LMS algorithm has the advantage of low computational complexity. On the contrary, the high computational complexity is the weakest point of RLS algorithm but it provides a fast adaptation rate. Thus, it is clear that the choice of the adaptive algorithm to be applied is always a tradeoff between computational complexity and fast convergence.

The proposed work presents an implementation of adaptive noise canceller in Field programmable Gate arrays (FPGA) using Recursive Least Square (RLS) and Least Mean Square algorithms with the intention to compare their performance in noise cancellation in terms of convergence behavior.

## 3. FPGA Implementation

The important block for the implementation of LMS and RLS algorithm is the dual port RAM The Xilinx Dual Port RAM block implements a random access memory (RAM). Dual ports enable simultaneous access to the memory space at different sample rates using multiple data widths.
The Dual Port RAM block also supports various Form Factors (FF). Form factor is defined as: FF = WB / WA

where WB is data width of Port B and WA is Data Width of Port A. The Depth of port B (DB) is inferred from the specified form factor as follows:

DB = DA / FF. The data input ports on Port A and B can have different arithmetic type and binary point position for a form factor of 1. For form factors greater than 1, the data input ports on Port A and Port B should have an unsigned arithmetic type with binary point at 0. The output ports, labeled A and B, have the same types as the corresponding input data ports. The location in the memory block can be accessed for reading or writing by providing the valid address on each individual address port. A valid address is an unsigned integer from 0 to d-1, where d denotes the RAM depth (number of words in the RAM) for the particular port. An attempt to read past the end of the memory is caught as an error in simulation. The initial RAM contents can be specified through a block parameter. Each write enable port must be a boolean value. When the WE port is 1, the value on the data input is written to the location indicated by the address line. The output during a write operation depends on the write mode. When the WE is 0, the output port has the value at the location specified by the address line. During a write operation (WE asserted), the data presented on the input data port is stored in memory at the location selected by the port's address input. During a write cycle, you can configure the behavior of each data out port A and B to one of the following choices: Read after write, Read before write and No read on write.

The write modes can be described with the help of the figure below. In the figure, the memory has been set to an initial value of 5 and the address bit is specified as 4. When using No read on write mode, the output is unaffected by the address line and the output is the same as the last output when the WE was 0. For the other two modes, the output is obtained from the location specified by the address line, and hence is the value of the location being written to. This means that the output can be the old value which corresponds to Read after write.

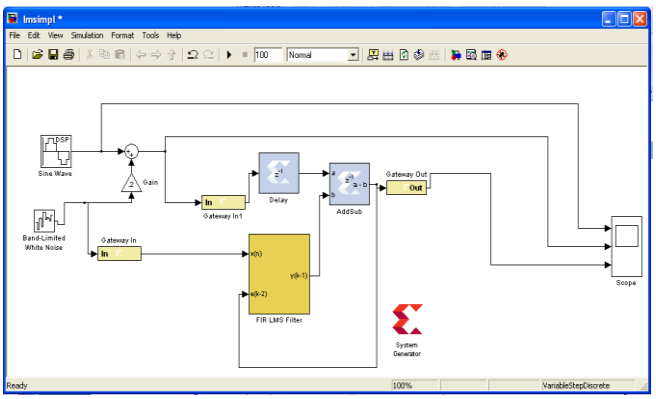

**Figure 2.1:** Top level design of LMS adaptive noise canceller using XSG

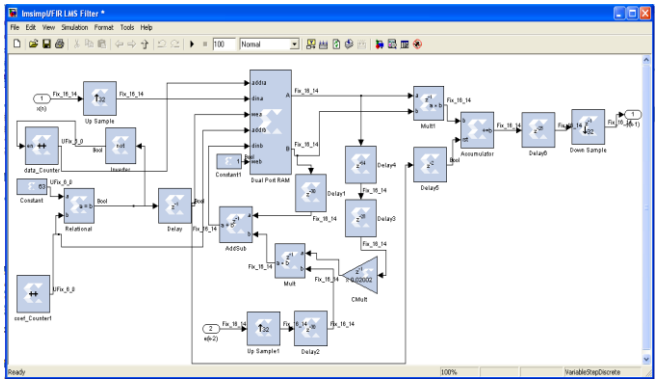

**Figure 2.2:** LMS Adaptive noise canceller subsystem in XSG

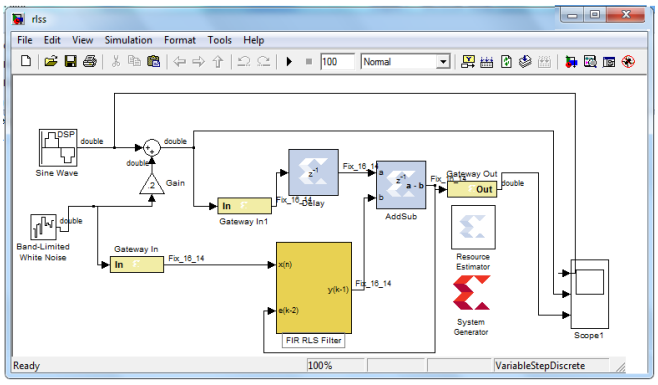

**Figure 2.3:** Top level design of RMS adaptive noise canceller using XSG

Figure 2.1 and 2.2 presents a top level design diagram of an adaptive noise canceller and the detailed sequential architecture of a pipelined LMS based adaptive noise canceller subsystems. The LMS algorithm can easily implemented in this architecture using xiinx basic bock elements such as adder/sub and multiplier bock sets. As labeled on the connecting wires (FIX_16_14), the fixe-point data is a2's complement signed 16-bit number having 14 fractional bits. Xilinx SARTAN-3 (xc3s200-4ft256) board is used to implement this adaptive noise canceller.

Similar architecture is implemented for RLS based adaptive noise cancellation the top level design of RLS filter and its subsystems are shown in figure2.3 and2.4. In the RLS implementation to reduce the complexity of its computation our design using a M-code block . The Xilinx MCode block is a container for executing a user-supplied MATLAB function within Simulink. A parameter on the block specifies the M-function name. The block executes the M-code to calculate block outputs during a Simulink simulation. The same code is translated in a straightforward way into equivalent behavioral VHDL/Verilog when hardware is generated.

The block's Simulink interface is derived from the MATLAB function signature, and from block mask parameters. There is one input port for each parameter to the function, and one output port for each value the function returns. Port names and ordering correspond to the names and ordering of parameters and return values. The MCode block supports a limited subset of the MATLAB language that is useful for implementing arithmetic functions, finite state machines and

control logic. The MCode block has the following three primary coding guidelines that must be followed:

- All blocks input and outputs must be of Xilinx fixed point type
- The bock must have at least one output port.
- The code of the block must exist on the matlab path or in the same directory as the mode that uses the block

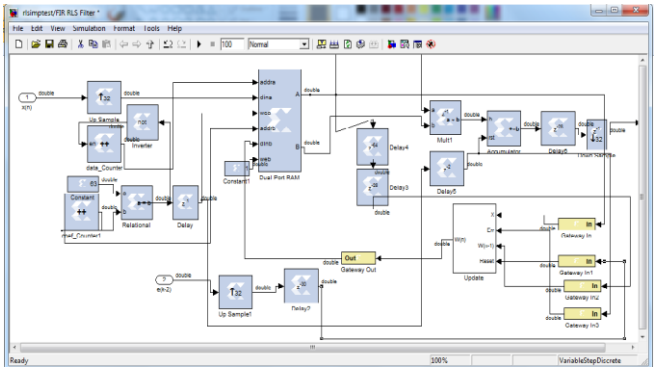

**Figure 2.4:** RLS based Adaptive noise canceller subsystem in XSG

### 3.1 Hardware in the Loop Co-Simulation

*Simulations of adaptive filter algorithms and generate the run-time hardware model*

The system generator token will generate the hardware model for LMS based and RLS based adaptive noise canceller algorithms which are shown in the figure 5.

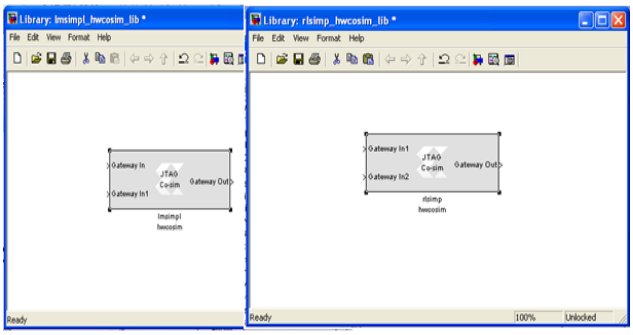

**Figure 3.1:** Run time model of generated bitstream for LMS and RLS Noise cancellers

System Generator will generate the HDL Code and automatically invoke the ISE Foundation software to generate the bit stream. The run-time blocks in figure 6 and 7represents the hardware model of the System Generator design.

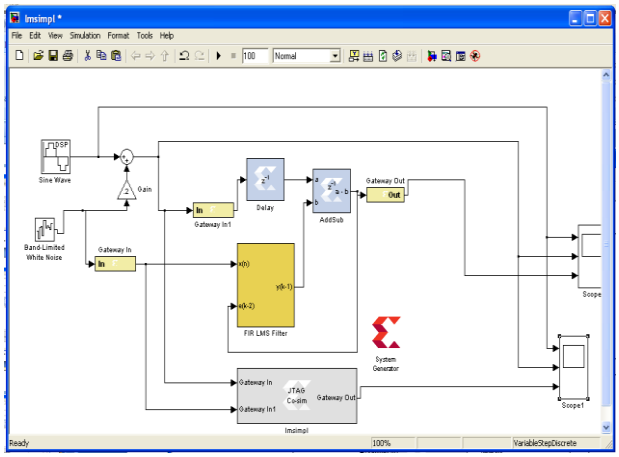

**Figure 3.2:** Hardware-in-the-loop co-simulation of the LMS adaptive noise canceller

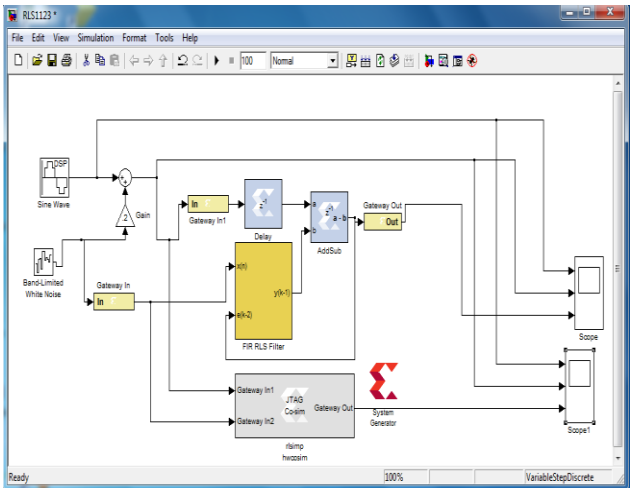

**Figure 3.3:** Hardware-in-the-loop co-simulation of the RLS adaptive noise canceller

## 4. Results and Discussion

The hardware-in-the-loop co-simulation permits to incorporate design running in an FPGA directly into a Simulink simulation. When the design is simulated, the compiled portion (JTAG co-simulation block) is actually running on the hardware and data is transferred between computer and FPGA board. Fig. 4.1, 4.2, shows the result of LMS based adaptive noise cancelation obtained with sinusoidal signal which is contaminated with band limited interference signal. This figures include the sinusoidal signal (top), the primary input contaminated with noise (middle) and the output of the adaptive canceller (bottom), where the interference signal is eliminated once the adaptive process has converged.

In the figure 3.3 scope2 is used for the verification of the adaptive algorithm in software and the scope 1 used for the implementation verification of the system in target device named Xilinx SARTAN-3 (xc3s200-4ft256). The LMS algorithm implementation verification in software and hardware are shown in the figure 4.1 and 4.2 respectively. The RMS algorithm implementation verification in software and hardware are shown in the figure 4.3 and 4.4 respectively. The RTL schematic of LMS and RLS

implementations are also shown in figure 4.5 and 4.6 respectively.

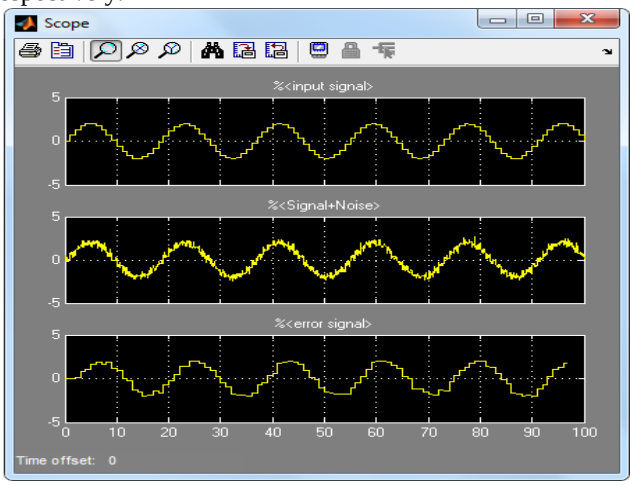


**Figure 4.1:** LMS adaptive noise cancellation in software

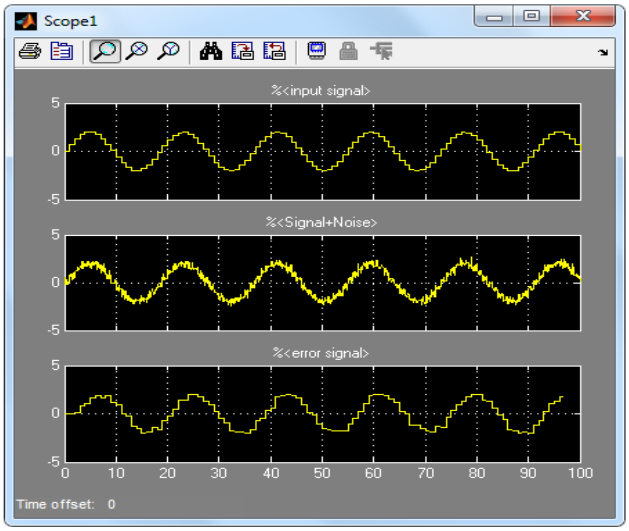


**Figure 4.2:** LMS adaptive noise cancellation in hardware

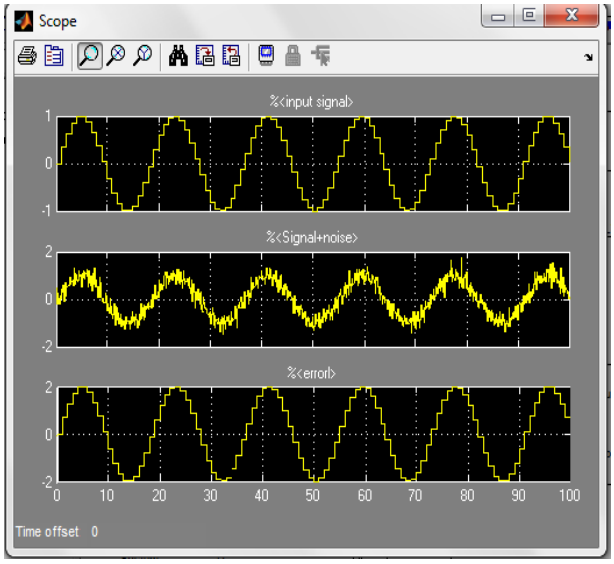


**Figure 4.3:** RLS adaptive noise cancellation in software

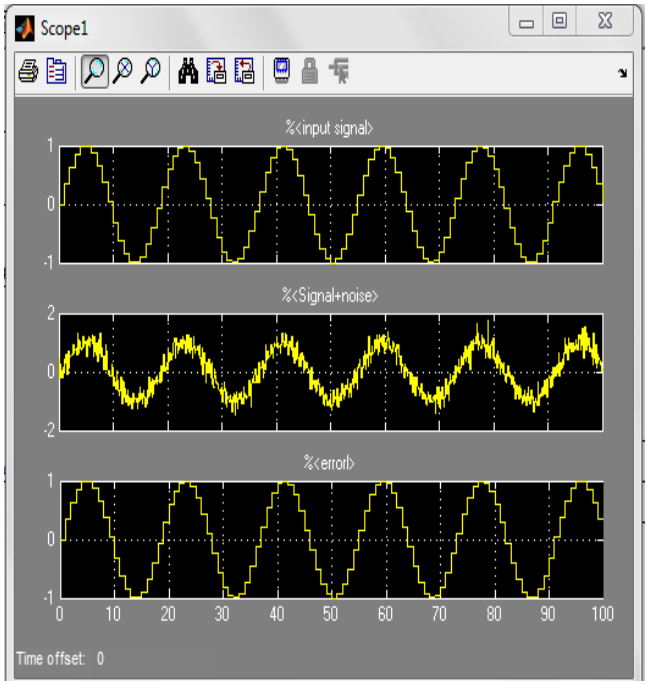


**Figure 4.4:** RLS adaptive noise cancellation in hardware

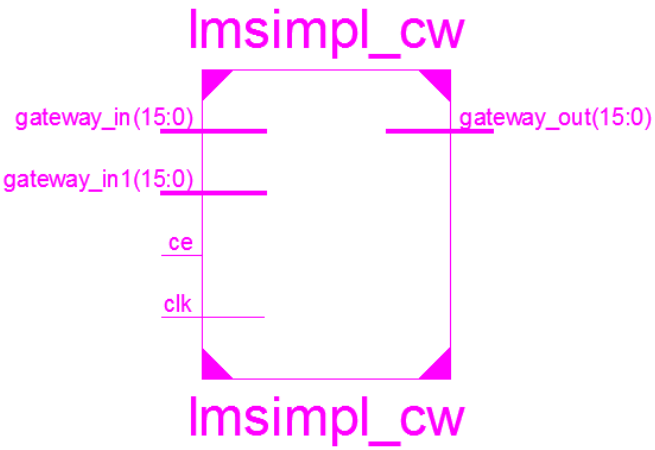


**Figure 5.7:** RTL schematic of LMS implementation

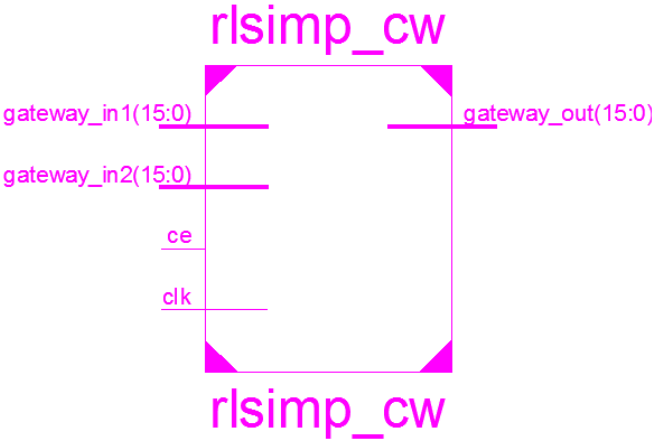


**Figure 5.8:** RTL schematic of RLS implementation

## 5. Conclusion and Future Scope

The result shows that RLS based adaptive noise canceller has better convergence than LMS based adaptive noise canceller. The major advantage of the proposed system is its ease of implementation on FPGA using Xilinx System

Generator. This algorithm has wide applications in wireless communications and signal processing such as beam forming, channel equalization and HDTV. Although the calculation load is large, it achieves fast convergence and is thus effective under fading environments; it shows great promise in mobile communication application.

## 6. Acknowledgment

I would like to express profound gratitude to our Vice principal, Prof. Thomas George, and head of the department Arun M.S. for their encouragement and for providing all facilities for my work.

## References

[1] Widrow B and Hoff CSJ, " *Adaptive switching circuits*" IRE WESCON Conv. Rec., pp. 96–104, 1960.
[2] Herbert Robbins and Sutton Monro, "*A Stochastic Approximation Method*", Ann. Math. Statist. Volume 22, Number 3, 400-407, 1951.
[3] R.L. Plackett, "*Some theorems in least-squares*," Biometrika, vol. 37, pp. 149, 1950.
[4] R.L. Plackett, "The *discover of the method of least-squares*," Biometrika, vol. 59, pp. 239–251, 1972.
[5] Haykin S ,Adaptive Filter Theory: Prentice Hall, Englewood Cliffs, NJ, 2001.
[6] Saymon Haykins, and Thomas Kailath, Adaptive Filter Theory, Fourth Edition, Pearson Education.
[7] B.widrow and S.D.Stearns, adaptive Signal Processing. Englewood Cliffs, NJ:Prentice-Hall. 1985,p.474.
[8] Sondhi M.M "*An adaptive echo canceller*", Bell Syst.Tech .J vol.46,pp 497-511,1967.
[9] Widrow B. Etal."*Adaptive noise cancelling: Principles and applications*, "Proc.IEEE, vol.63,pp.1692-1716,1975.
[10] W. Harrison, J. Lim, E. Singer, "A new application of adaptive noise cancellation," *IEEE Trans. Acoustic Speech Signal Processing*, vol.34, pp. 21-27, Jan 1986. G. Goodwin, k. Sin, *Adaptive Filtering Prediction and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
[11] G. Goodwin, k. Sin, *Adaptive Filtering Prediction and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

### Author Profile

**Athira Sivanandhan** obtained her Bachelor Degree in Electronics and Instrumentation from Anna University in 2007. Then she obtained her Master's of Technology Degree in Embedded Systems from Kerala University in 2012. She is currently working as Assistant **Professor in the department of Applied Electronics & Instrumentation at Mount Zion College of Engineering, Kadammanitta,** A P J Abdul Kalam Technological University, Kerala. Her specialization includes Embedded System, , Control Engineering, Instrumentation.

**Thomas George** obtained his Bachelor Degree in Electronics from Cochin University of Science & Technology, Kerala in 1999. Then he obtained his Master's of Engineering Degree in Applied Electronics from Bharathiar University Coimbatore in 2001. He is currently working as **Professor in the department of Applied Electronics & Instrumentation at Mount Zion College of Engineering, Kadammanitta,** A P J Abdul Kalam Technological University, Kerala. His specialization includes Advanced Communication, Space Technology, Control Engineering, Embedded System and Instrumentation. His current research interests are Control Engineering, Instrumentation and Space Technology.