MAT JOURNALS

Journal of Radio and Television Broadcast
Volume 3 Issue 2

Low-Latency RLS Architecture for FPGA Implementation with High Throughput Adaptive Applications

[1]Praveen Reddy, [2]Dr.Baswaraj Gadgay
[1]Assistant Professor, EEE, GNDEC, Bidar, Karnataka, India,
[2]Professor, ECE, VTU, Kalaburagi, Karnataka, India
***Email:** [1]reddysirlogin@gmail.com,[2]b_gadgay@rediffmail.com*

Abstract

A novel architecture for QR-decomposition-based (QRD) recursive least squares (RLS) is presented. It offers low idleness for applications where the channel balance and versatile separating is obligatory. This approach lessens the calculations by reworking the conditions in a way that lets extraordinary equipment asset sharing by reusing comparable qualities in various calculations. Additionally, accuracy run change (PRC) takes into consideration joining complex activities, for example, root square and division with least impact on the general quantization blunder. Hence, an efficient Look Up Table based solution has highly enhanced the performance of the design by 2.7 times with respect to the previous works.

Index Terms: *Adaptive filters, Equalizer, FPGA, QR decomposition (QRD), recursive least squares (RLS).*

INTRODUCTION

To prevent the complexity of mathematical operations in RLS technique, QR decomposition (QRD) algorithm based on Givens Rotation method is widely used to perform adaptive weight calculations in variety of applications including multiple input multiple-output (MIMO), beam formings and in the pre-whitening in the independent component analysis (ICA) [2,3,4], which leads to more efficient architectures, stability and accuracy. Another key benefit of using GR is the ease of deploying systolic array architectures using CORDIC blocks that can guaranty a more efficient hardware implementation [5]. In none-recursive applications, this approach improves the clock frequency and the data rate by further pipelining the CORDIC structure [4]. However, in recursive applications, it does not improve the throughput since the process of the next input vector depend on the result of the computation on the previous one [6].

The goal of this paper is to develop a high throughput architecture by minimizing the latency of RLS structure for high speed adaptive applications. In addition, using new implementation techniques, the proposed design offers low hardware complexity and as a result reduction in power consumption which are critical parameters in the realization of wireless communication systems. The remainder of this paper is organized as follows. The next section reviews the matrix inversion, RLS and Squared Givens rotation algorithms. The Architecture design, systolic array and implementation details are described in sections III. Section IV discuses results and comparison followed by conclusion in Section V.

The Recursive Least Squares (RLS) versatile channel is a calculation which recursively finds the channel coefficients that limit a weighted direct minimum squares cost work identifying with the information signals. For the RLS calculation to have the capacity to track the elements of a period variation blurring channel and in the meantime to smother

the impact of the got clamor on the evening out mistake. The important feature of Recursive least mean square algorithm is that its rate of convergence is typically an order of magnitude faster than that of LMS filter, due to fact that the RLS filter whitens the input data by using the inverse correlation matrix of the data, assumed to be zero mean. This improvement in performance is achieved at the expense of an increase in computational complexity of the RLS filter. The block diagram of RLS algorithm is as shown in the figure 1.

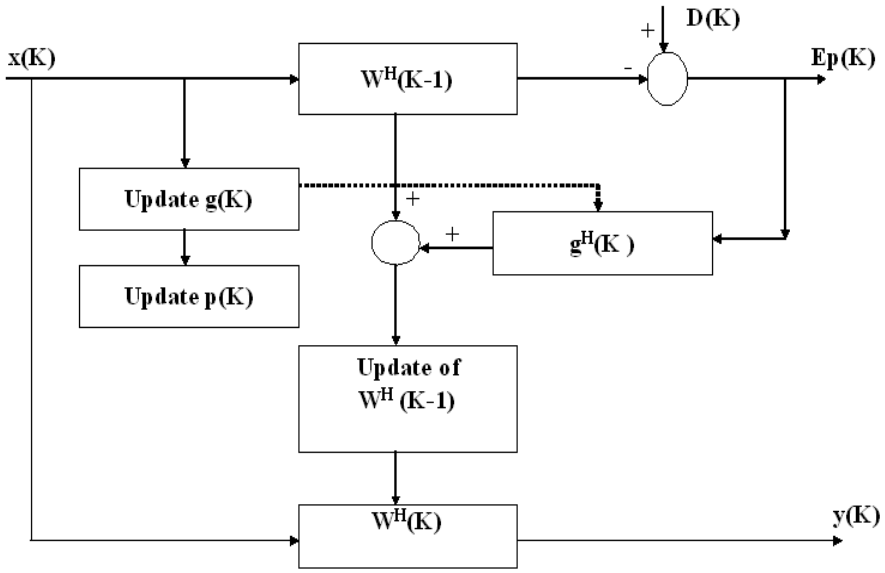

Fig 1. *Block diagram of RLS algorithm*

In recursive minimum square calculation the calculations begin with endorsed beginning conditions and utilize the data contained in new information tests to refresh the old assessments. Then we find the length of observable data

In general, the RLS can be used to solve any problem that can be solved by adaptive filters. For example, suppose that a signal d(n) is transmitted over a noisy channel that causes it to be received as

$$x(n) = \sum_{k=0}^{q} b_n(k)d(n-k) + v(n)$$

where $v(n)$ represents adaptive noise. We will attempt to recover the desired signal $d(n)$ by use of

$$\hat{d}(n) = \sum_{k=0}^{p} w_n(k)x(n-k) = \mathbf{w}_n^T \mathbf{x}_n$$

$\mathbf{x}_n = [x(n) \quad x(n-1) \quad \dots \quad x(n-p)]^T$ a $p+1$-tap FIR filter, $\mathbf{w}$ where is the vector containing the p most recent samples of $x(n)$. Our goal is to estimate the parameters of the filter $\mathbf{w}$, and at each time n we refer to the new least squares estimate by $\mathbf{w_n}$. As time evolves, we would like to avoid completely redoing the least squares algorithm to find the new estimate for $\mathbf{w}_{n+1}$, in terms of $\mathbf{w}_n$.

The advantage of the RLS calculation is that there is no compelling reason to upset networks, in this manner sparing computational power. RLS calculation's progression following capacity and its info clamor concealment capacity. The overlooking component is picked by a reverse capacity of the leftover power keeping in mind the end goal to

accomplish a consistent weighted aggregate of the squares of the a posteriori blunders. At the end of the day, the measure of overlooking will at each progression relate to the measure of new data in the most recent estimation, in this manner guaranteeing that the estimation is constantly in light of a similar measure of data. The fundamental idea of the time-shifting ideal overlooking variable can be clarified as takes after. On account of a close deterministic framework, the a posteriori estimation blunder gives the data about the condition of the estimator. When the initial value is set to unity and the error is small, it may be concluded that the estimator is sensitive enough to adjust to the variations of the system parameters and therefore to significantly reduce the estimation error.

Proposed system

The block diagram implementation is shown below. The proposed engineering is intended to perform independent usage and by including proper buffering hinders the design is set aside a few minutes execution. The approaching persistent information is put away in input buffering RAMS to accommodate the computations required until the weight updation. The data is read out alternatively from each of the input RAM blocks and is written into input data memory. Starting tap weights are Stored in tap weight memory square. Both the information and weights are readout of the memory all the while and go to the multiplier which duplicates the information vector and weight vector and passes the information on to the running summer obstruct includes the multiplier yield .And then the running summer yield, proportional to the FIR channel yield, (which in a perfect world requires N multipliers relying upon the quantity of taps which here is diminished to one) is then subtracted from next information test to gauge the blunder. The blunder esteem got is then increased with variable pick up factor and information vector to figure the weight refreshes required decreasing the mistake from every computation. The weight refreshes are then added to the past weight esteems and composed once again into the weight vector memory.

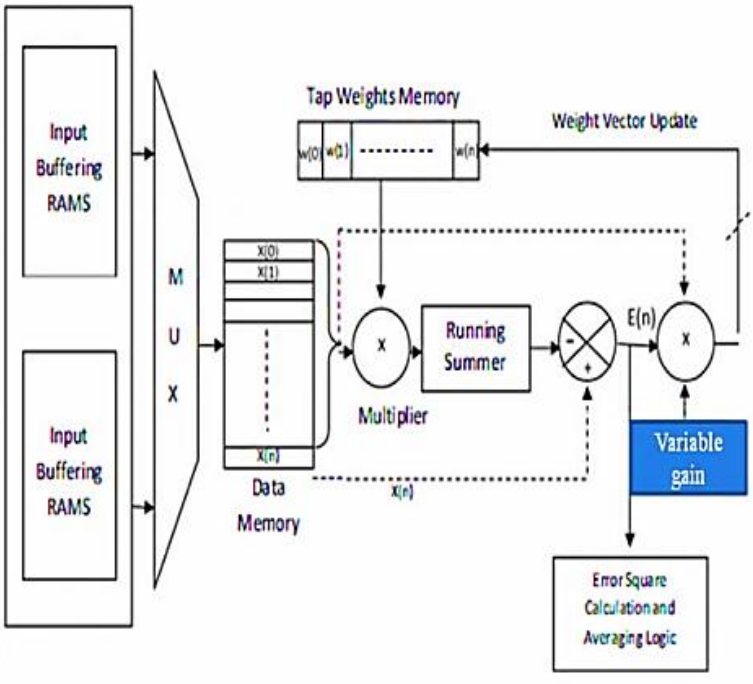

Fig 2. *Block Diagram of the Proposed System*

The updated weights are now ready for next set of data. As mentioned earlier ideally FIR filter structure requires N multiplier depending on the number of weights considered for implementation, but with the architecture proposed. The number of multipliers is reduced to one with the pipelining of the data. The architecture presented consumes minimal hardware making it suitable for FPGA implementation.

The joining rate decides the rate at which the channel unites to its resultant state. Generally a speedier merging rate is a coveted normal for a versatile framework. Union rate isn't, in any case, free of the greater part of the other execution attributes. There will be a tradeoff, in other execution criteria, for an enhanced meeting rate and there will be a diminished joining execution for an expansion in other execution. For instance, if the union rate is expanded, the steadiness qualities will diminish, making the framework more prone to veer rather than focalize to the correct arrangement. Similarly, a decline in meeting rate can make the framework turn out to be more steady. This demonstrates the joining rate must be considered in connection to the next execution measurements, not independent from anyone else without any respects to whatever remains of the framework.

The base mean square mistake (MSE) is a metric showing how well a framework can adjust to a given arrangement. A little least MSE means that the versatile framework has precisely demonstrated, anticipated, adjusted and additionally focalized to an answer for the framework. An expansive MSE typically shows that the versatile channel can't precisely display the given framework or the underlying condition of the versatile channel is a lacking beginning stage to make the versatile channel merge. There are a number of factors which will help to determine the minimum MSE including, but not limited to quantization noise, order of the adaptive system, measurement noise, and error of the gradient due to the finite step size.

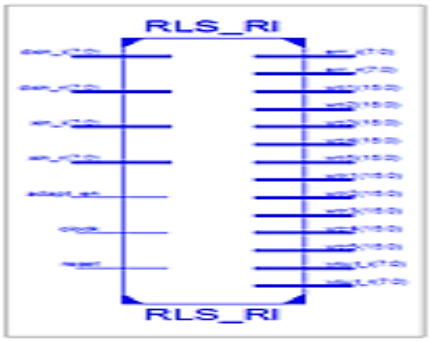

Fig 3. *Top Level RTL schematic of the proposed RSL System*

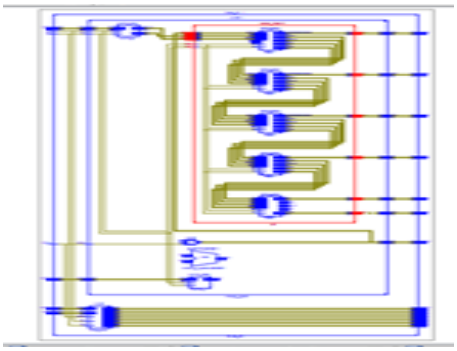

Fig 4. *Internal Hardware architecture of the design*

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	1333	3584	37%
Number of Slice Flip Flops	987	7168	13%
Number of 4 input LUTs	2218	7168	30%
Number of bonded IOBs	227	141	160%
Number of GCLKs	1	8	12%

Fig 5. *Design summary for Proposed system*

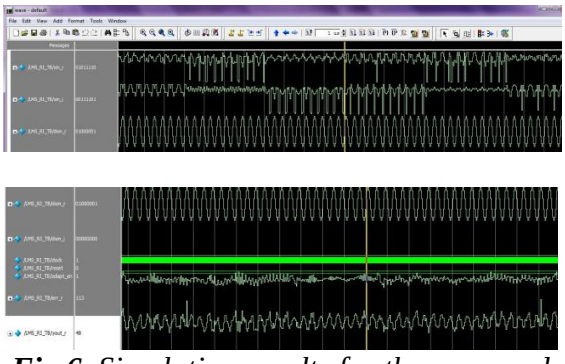

Fig 6. *Simulation results for the proposed system*

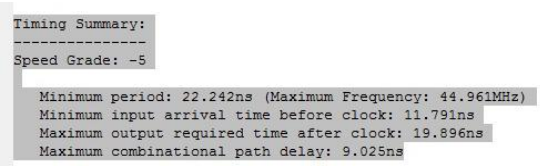

Fig 7. *Timing summary for the proposed System*

CONCLUSION

In this paper, a high speed fixed point RLS system was designed for adaptive filter applications. It is shown that the precision of the input of the critical RINV can be reduced such that the function can easily fit within a ROM with sufficient accuracy while the overall system error remains within an acceptable range. The proposed architecture is simulated with actual data to verify the maximum error level. It was also mapped into various FPGAs to draw a fair comparison with other works. The proposed work outperforms the previously published work by 2.7 times in terms of FOM which includes performance, area, bitwidth and the number of filter taps.

REFERENCE

1. S. S. Haykin, *Adaptive Filter Theory*: Pearson Education, Limited, 2013.
2. S. C. Chan and Y. J. Chu, "A New State-Regularized QRRLS Algorithm With a Variable Forgetting Factor," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 59, pp. 183-187, 2012.
3. G. Jian-Feng, S. C. Chan, Z. Wei-Ping, and M. N. S. Swamy, "Joint DOA Estimation and Source Signal Tracking With Kalman Filtering and Regularized QRD RLS Algorithm," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 60, pp. 46-50, 2013.
4. J. A. APOLINARIO, *QRD-RLS Adaptive Filtering*: Springer, 2009.
5. K. R. Santha, B. C. Chava, K. Bragadishwaran, and K. Chandru, "FPGA implementation and resource utilization for QRD-RLS systolic array for signal processing applications," in *TENCON IEEE Region 10 Conference*, 2009, pp. 1-5, 2009.
6. J. M. a. E. F. D. a. K. K. Parhi, "Pipelined Cordic Based QRD-RLS Adaptive Filtering Using Matrix Lookahead," *Conference: Signal Processing Systems*, 1997.
7. Hardware-Software Co-design of QRD-RLS Algorithm with Microblaze Soft Core Processor" by Napur Lodha, Nivesh Rai, Rahul Dubey, Hrishikesh Venkataraman in the Third International Conference on Information Systems, Technology and Management (ICISTM-09), 2009, pp. 197 - 207, 2009.
8. M. Karkooti, J. R. Cavallaro, and C. Dick, "FPGA Implementation of Matrix Inversion Using QRD-RLS Algorithm," in *Signals, Systems and Computers, 2005. Conference Record of the Thirty-Ninth Asilomar Conference on*, 2005, pp. 1625-1629, 2005.
9. H. K. W. M. Gentleman, "Matrix Triangularization by Systolic Arrays," *Real-Time Signal Processing*, vol. 298, pp. 19-26, 1981.
10. C. Dongdong and M. Sima, "Fixed-Point CORDIC-Based QR Decomposition by Givens Rotations on FPGA," in *Reconfigurable Computing and FPGAs, International Conference on*, pp. 327-332, 2011.
11. D. Boppana, K. Dhanoa, and J. Kempa, "FPGA based embedded processing architecture for the QRD-RLS algorithm," in *Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th Annual IEEE Symposium on*, pp. 330-331, 2004.
12. M. Shabany, D. Patel, and P. G. Gulak, "A Low-Latency Low-Power QR-Decomposition ASIC Implementation in 0.13 CMOS," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 60, pp. 327-340, 2013.

13. K. J. Raghunath and K. K. Parhi, "Finite-precision error analysis of QRD-RLS and STAR-RLS adaptive filters," *Signal Processing, IEEE Transactions on*, vol. 45, pp. 1193-1209, 1997.
14. Z. Wang, C. Pan, Y. Song, and C. Sechen, "High-Throughput Digital IIR Filter Design," *Journal of Algorithms and Optimization*, vol. 2, pp. 15-27, 2014.
15. C. Pan, Z. Wang, and C. Sechen, "High speed and power efficient compression of partial products and vectors," *Journal of Algorithms and Optimization*, vol. 1, pp. 39-54, 2013.
16. Y. Y. K. M. A. Hiroyuki, "Resource and performance evaluations of fixed point QRD-RLS systolic array through fpga implementation," *EICE Trans. on Communications*, vol. 4, pp. 1068–1075, 2008.
17. S. M. a. R. K. A. Irturk, "An Efficient FPGA Implementation of Scalable Matrix Inversion Core using QR Decomposition," *UCSD Technical Report*, 2009.
18. A.P.Deshpande, D. G. Rao, N. S. Murthy, and A. Vengadarajan, "Efficient filter implementation using QRD-RLS algorithm for Phased Array Radar applications," in *Technological Advances in Electrical, Electronics and Computer Engineering, International Conference on*, pp. 224-229, 2013.