Available online at www.elixirpublishers.com (Elixir International Journal)

Advanced Engineering Informatics

Elixir Adv. Engg. Info. 35 (2011) 2885-2888

ISSN: 2229-712X

Backend analysis and implementation of RLS adaptive filter using VLSI technology

N.J.R.Muniraj

Karpagam Innovation Centre, Karpagam College of Engineering, Coimbatore.

ARTICLE INFO

Article history:

Received: 8 April 2011;

Received in revised form:

20 May 2011;

Accepted: 27 May 2011;

Keywords

HDL,
TSMC,
LMS,
FIR,
SPICE,
RTL,
Astro,
GDS11,
DRC.

ABSTRACT

The role of electronic equipments in the industry has increased tremendously in recent past. With new technologies and techniques being considered in other domains, such as automotive, multimedia communications, mobile applications bring down the cost of the electronic gadgets. As the cost factor controls the reliability and volume issues, there is a need for design and development of low cost, reliable technology for industrial applications. The proposed techniques have been modeled using Verilog HDL and the models have been verified using test benches with a functional coverage of 95%. The results obtained have been compared with MATLAB results, which are considered to be a benchmark. The HDL (Hardware Description Language) code is synthesized using Synopsys Design Compiler targeting 130-nanometer TSMC (Taiwan Semiconductor Manufacturing Company) library and target technology. The synthesized netlist obtained for all the adaptive filtering techniques proposed in this research work is taken through physical design flow consisting of Floorplanning, Placement and Routing steps. The results obtained at each step are simulated for the functionality. The final GDSII (Graphical Design Standard II) file is generated for the proposed techniques.. The floor planning, placement and routing of the netlist ensures that the overall size for the entire chip does not exceed by 7.2 square millimeters. The results obtained for adaptive filtering techniques have proven that the complexities in the industrial applications can be met if the design is implemented on ASIC.

Introduction

The other class of adaptive filtering techniques is known as Recursive Least Squares (RLS) algorithms. These algorithms attempt to minimize the cost function as shown in in Equation (1.1). Where k=1 is the time at which the RLS algorithm commences and λ is a small positive constant very close to, but, smaller than 1. With values of λ<1 more importance is given to the most recent error estimates and thus the more recent input samples, this results in a scheme that places more emphasis on recent samples of observed data and tends to forget the past (Jun Ma et al 2000).

$$\xi(n) = \sum_{k=1}^{n} \lambda^{n-k} e_n^2(k) \qquad (1.1)$$

Unlike the LMS algorithm and its derivatives, the RLS algorithm directly considers the values of previous error estimations. RLS algorithms are known for excellent performance when working in time varying environments. These advantages come with the cost of an increased computational complexity and some stability problems as explained by Nishikawa and Kiya (2000). The RLS cost function of Equation (3.43) shows that at a time n, all previous values of the estimation error since the commencement of the RLS algorithm are required. Clearly as time progresses the amount of data required to process this algorithm increases. The fact that memory and computation capabilities are limited makes the RLS algorithm a practical impossibility in its purest form (Wei 2002). However, the derivation still assumes that all data values are processed. In practice only a finite number of previous values are considered, this number corresponds to the order of the RLS FIR filter and the equivalent design architecture of RLS is shown in Figure 1.1.

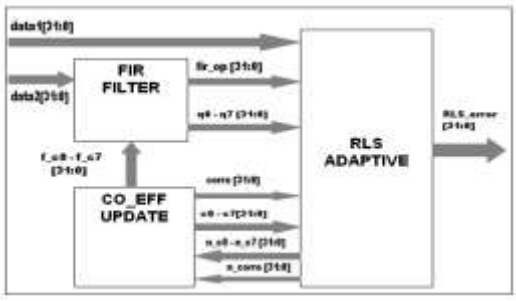

Figure 1.1 Design architecture of RLS

To implement the RLS algorithm, the following steps are executed in the following order.

1. The filter output is calculated using the filter tap weights from the previous iteration and the current input vector.

$$\bar{y}_{n-1}(n-1) = \bar{w}^{T}(n-1)x(n) \qquad (1.2)$$

2. The intermediate gain vector is

$$u(n) = \bar{\psi}_{\lambda}^{-1}(n-1)x(n) \qquad (1.3)$$

$$k(n) = \frac{1}{\lambda + x^{T}(n)u(n)}u(n) \qquad (1.4)$$

3. The estimation error value is

Tele:
E-mail addresses: njrmuniraj@yahoo.com

$$\bar{e}_{n-1}(n) = d(n) - \overline{y}_{n-1}(n) \quad (1.5)$$

4. The filter tap weight vector is updated and the gain vector is

$$w(n) = \overline{w}^{T}(n-1) + k(n)\bar{e}_{n-1}(n) \quad (1.6)$$

5. The inverse matrix is

$$\Psi_{\lambda}^{-1}(n) = \lambda^{-1}(\Psi_{\lambda}^{-1}(n-1) - k(n)[x^{T}(n)\Psi_{\lambda}^{-1}(n-1)]) \quad (1.7)$$

Each iteration of the RLS algorithm requires $4N^2$ multiplication operations and $3N^2$ additions. This makes its very costly to implement, thus LMS based algorithms, while they do not perform as well, are more favorable for practical situations (Wei 2002). where, d(n)=desired signal, λ(n)= forgetting factor, G(n)=Kalman gain factor, P(n)= inverse of auto correlation matrix, and the superscript $^{-1}$ shows inverse operation as said by Kalavai and Keshab (1995).

Simulation Results

Different architectures considered in this research are modeled using HDL language for hardware implementation. To verify the HDL results and to make sure that the functional verification of the models are meeting the requirements, the results obtained using simulation of HDL codes are cross verified with MATLAB results, which is used for bench marking the design (Alexander Poularikas 2006). With known test inputs applied to MATLAB and HDL, the results obtained are compared for all possible simulation times. A simulation time more than 10 time unit is considered to be safe as this gives the stable results considering all initial conditions. The ModelSim simulation and MATLAB results are compared at the 20th iterations. Figures 2.1 and 2.2 shows the MATLAB and ModelSim simulation results at 20th iteration. Here, the expected output value is 0.2036 and the coefficients values are 1.0439, 0.3869, 0.1557, 0.1214, -0.1888, -0.4204, -0.0379, -0.2333. The simulation output value is 0.21 and coefficient values are 1.05, 0.39, 0.16, 0.13, -0.019, -0.4, -0.07 and 0.24. MATLAB versus Hardware implementations may vary because of floating point precision. The mismatches between the hardware and software implementations are observed and recorded. Increasing the bit width of the operands and using rounding techniques minimize any mismatches observed. This process is performed as long as the mismatches are within tolerable limits. It is observed that by increasing the bit width to 12 bits the mismatches are with tolerable limits of less than 8%, which is far better than the expected and recommended in the literature.

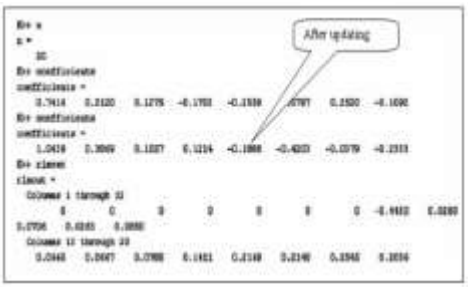

Figure 2.1 MATLAB results at 20th iteration

Figure 2.2 Simulation report of RLS architecture

HDL models are verified by using test benches, appropriate test vectors are generated in the test bench for verification. Using the test vectors 95% of the functional verification is covered. However, 100% coverage is possible only after the design is physically implemented (Geoff Bostock 1996). The results found are meeting the design specifications.

With functional verification, the design is ready for physical implementation, the first step in physical implementation is to use the HDL model developed and convert it into an RTL code that can be synthesized as explained by Lakshmanan et al (2002). In this work, the HDL model developed was itself RTL model; utmost care was taken to develop RTL code directly. Hence, the RTL model is synthesized using industry standard EDA tool, called Design Compiler from Synopsys. DC compiler is tool for synthesis, in this process the RTL model is converted to gate level netlist. The gate level netlist should be able to meet area, timing and power requirements. In order to achieve the required area, timing and power, suitable constrains are to be specified along with target libraries consisting of standard cells. Synthesis is a three-phase process where it starts with translating the RTL code to the gate level netlist

The netlist is optimized using the constraints given. Constraints are two types namely, environmental and optimization constraints. Optimization constraints include operating frequency (clock period), input and output delays at the IOs. Operating temperature, process variations, supply voltage and wire load models come under Environmental constraints. The constraints mentioned for the design are maximum operating frequency 250MHz, total number of gates not to exceed 200 cells, and power not to exceed more than 10 mW. The Figures 2.3 shows the schematic, which was generated after synthesizing the RTL code and gives the top-level gate level netlist obtained after synthesis using Synopsys DC. The design requires 4 inputs and it produces one error corrected output.

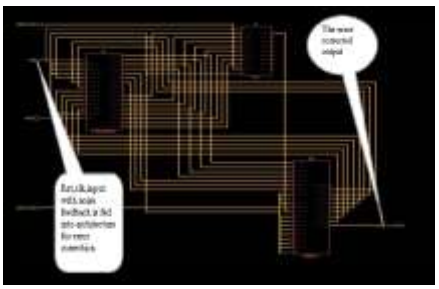

Figure 2.3 Synthesized schematic of RLS architecture

Table 2.1 Chip report of RLS architecture

Parameters	RLS report
Slack	0.31
Maximum critical paths	189
Number of ports	130
Number of nets	646
Combinational area in sq micron	1454847.75
Sequential area in sq micron	80672
Total area in sq micron	1872734.62
Total Dynamic Power	208.65 mw
Cell Leakage Power	3.59 mw

RLS architecture without pipeline requires less cell area, has 646 total numbers of nets to be routed and consumes 220 mw of power at 1pf capacitance load. With pipeline concepts being incorporated on the area by 7%, total power by 42% and congestion of routing inter connect increased by 38% affecting total die size as shown in Table 2.1. However, these effects are

being suitably reduced to a large extent by adopting optimized ASIC design methodology extent with proper floorplanning, placement and routing is being done with the die size of 3*3, the final chip is shown in Figures 3.52 and 3.53.

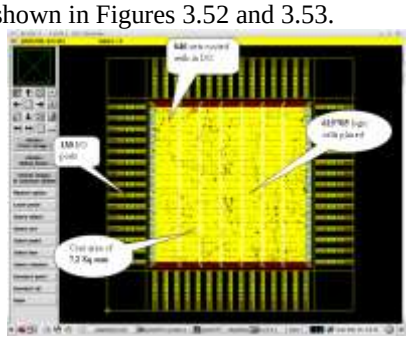

Figure 2.4 Final chip of RLS architecture

Conclusion

Adaptive noise cancellation techniques such as RLS have been extensively used for noise cancellation techniques with good performances in this work These techniques have been extended for use in industrial applications, wherein there is a need for accuracy, speed, reliability and cost. RLS algorithm has been realized on ASIC for comparison. The proposed architectures have been modeled and verified for its functionality successfully.

The models have been taken through the entire ASIC flow. Suitable results obtained at various stages of the ASIC flow using Synopsys clearly indicates that RLS is slow but optimizes area and power. The input signal is sampled at 1K samples per second; has a date rate of 16Kbitsper second when fed through the proposed hardware produces output at 16Kbitsper second with latency of 8 clocks and throughput of 1 clock cycle. The proposed techniques have been modeled using Verilog HDL and compared with MATLAB results, which are then synthesized using Synopsys Design Compiler targeting 130-nanometer TSMC library and target technology.

The synthesized netlist obtained for all the adaptive filtering techniques proposed in this research work is taken through physical design flow consisting of Floorplanning, Placement and Routing steps. The overall size for the entire chip does not exceed by 7.2 square millimeters. Constraints such as area, power and frequency have been used to optimize the design. A tradeoff between all the three have been identified and documented.

References

1. Ahmed Elhossini, Shawki Areibi and Robert Dony (2004), 'An FPGA Implementation of the LMS Adaptive Filter for Audio Processing', 16th International Conference on Microelectronics, Tunis, Tunisia, pp. 67-70.
2. Alexander Poularikas (2006), 'Adaptive filtering primer with MATLAB', Taylor & Francis Publications.
3. ASPT User Manual Version 2.1, www.mathworks.com, 12th Oct 2006.
4. Basker J. (2004), 'Verilog HDL Primer', BS Publication, Second Edition.
5. David Smith R. (2000), 'Verilog styles for synthesis of digital systems', Addison-Wesley Longman Publisher. Emmanuel Ifeachor C. and Barrie Jervis W. (2002), 'Digital Signal Processing - A practical approach', Pearson Education Asia.
6. Eweda (1994), 'Comparison of RLS, LMS, and Sign Algorithms for Tracking Randomly Time-Varying Channels', IEEE Transactions on Signal Processing, Vol. 42, No. I I.
7. Geizer R.L., Allen P.E. and Strader N.R. (1990), 'VLSI Design Techniques for Analog and Digital Circuits', McGraw Hill.
8. Geoff Bostock (1996), 'FPGA's and Programmable LSI: A Designer's Handbook', Butterworth-Heinemann, 1st Edition.
9. Greenberg J.E. (1998), 'Modified LMS Algorithms for Speech Processing with an Adaptive Noise Canceller', IEEE Transactions on Speech and Audio Processing, Vol. 6, No. 4.
10. Hachtel G.D. and Somenzi F. (1996), 'Logic synthesis and Verification Algorithms', Springer Publications.
11. Harada A. (1998), 'A pipelined architecture for LMS adaptive FIR filter architecture', IEEE Trans. Signal Process, Vol. 46, pp. 775-779.
12. Haykin S. (1992), 'Adaptive Filter Theory', EnglewoodCliffs, NJ: Prentice-Hall.
13. Himanshu Bhatnagar (2002), 'Advanced **ASIC** Chip Synthesis Using Synopsys® Design Compiler® Physical Compiler® and PrimeTime®', Prentice Hall of India.
14. Jenkins J.H. (1994), 'Designing with FPGAs and CPLDs', Prentice Hall PTR, 1st Edition.
15. Johnson C.R. and Larimore M.G. (1997), 'Comments on and addition to an Adaptive Recursive LMS Filter', Proc. IEEE, Vol. 65, pp. 1399-1401.
16. Jun Ma, Parhi K.K. and Deprettere F. (2000), 'Annihilation - Reordering Look-Ahead Pipelined CORDIC-Based RLS Adaptive Filters and their Applications to Adaptive Beam forming', Proceedings of the IEEE, 1503-587X/00.
17. Khosrow Golshan (2007), 'Physical Design Essentials: An ASIC Design Implementation Perspective', Springer Publishers..
18. Lakshmanan Othman, Ali M. and M.A.M. (2002), 'Implementation of RLS Adaptive Filter Algorithm Using MaxPlus II Software Tools', ICSE2002 Proc, 0-7803-7578-5/02, IEEE, pp.437-440.
19. Lan-Da Van and Chih-Hong Chang (2002), 'Pipelined RLS Adaptive Architecture Using Relaxed Givens Rotations (RGR)', Proc. IEEE, 0-7803-7448-7.
20. Lok-Kee Ting and Roger Woods (2005), 'Virtex FPGA Implementation of a Pipelined Adaptive LMS Predictor for Electronic Support Measures Receivers', IEEE Transactions on Very Large Scale Integration, Vol. 13, No. 1.
21. Mahesh Godavarti (2005), 'Partial Update LMS Algorithms', IEEE Transactions on Signal Processing, Vol. 53, No. 7.
22. Mark Gordon Arnold (1999), 'Verilog Digital Computer Design', Prentice Hall PTR, 1st Edition.
23. Mead C. and Convay L. (1999), 'Introduction to VLSI Systems', Addison Wesley Publication.
24. Meyer-Baese U. (2006), 'Digital Signal Processing with Field Programmable Gate Arrays', Second Edition, Springer International Edition.
25. Miguez-Olivares A. and Recuero-Lopez M. (1996), 'Development of an Active Noise Controller in the DSP Starter Kit', Texas Instruments.
26. Nazzareno Rossetti (2005), 'Managing Power Electronics: VLSI and DSP-Driven Computer Systems', Wiley-IEEE Press.
27. Nishikawa K. and Kiya H. (2000), 'Fast Implementation Technique for Improving throughput of RLS Adaptive Filters', IEEE Trans, Signal Process, 0-7803-6293-4/00, pp. 3319-3322.
28. Parhi K.K. (1999), 'VLSI Digital Signal Processing Systems', John Wiley & Sons, 1st Edition.
29. Pucknell D.A. (1997), 'Kamran Eshraghian: Basic VLSI Design', Third Edition, Prentice Hall of India.

30. Rabaey J. (1996), 'Digital Integrated circuits', Prentice Hall of India.
31. Raghunath K.J. and Parhi K.K. (1995), 'A 100Mhz pipelined RLS Adaptive Filter', Proceedings of IEEE Intl. Symp. on Circuits and Systems, pp. 3187-3190.
32. Raghunath K.J. and Parhi K.K. (1996), 'Pipelined RLS Adaptive Filtering Using Scaled Tangent Rotations (STAR)', Proceedings of the IEEE, 1053-587X/96.
33. Rajeev Murgai, Robert K., Brayton and Alberto Sangiovanni-Vincentelli (1995), 'Logic Synthesis for Field Programmable Gate Arrays', Kluwer Academic Publisher.
34. Richard Munden (2004), 'ASIC and FPGA Verification, A Guide to Component Modeling', Elsevier Publications.
35. Samir Panitkar (2003), 'Verilog HDL', Pearson Education, Second Edition.
36. Sankaran S.G. and Louis Beex A.A. (1999a), 'On ways to improve adaptive filter performance', Ph.D. Thesis in Electrical Engineering, Virginia Polytechnic and State University.
37. Sankaran S.G. and Louis Beex A.A. (1999c), 'Balanced - Realization Based Adaptive IIR Filtering', Proceeding of International Conference on Acoustics, Speech and Signal Processing, pp. 1837-1840.
38. Sebastian Smith M.J. (2001), 'Application-Specific Integrated Circuits', Fifth Edition, Pearson Education Inc.
39. Shou-Sheu Lin and Wen-Rong Wu (1996), 'The ASIC design of an LMS-Based decision feedback equalizer for TDMA digital cellular radio', IEEE Transactions, 0-7803-3692-5, pp. 218-222.
40. Sinead Mullins and Conor Heneghan (2002), 'Alternative Least Mean Square Adaptive Filter Architectures for Implementation on Field Programmable Gate Arrays', Digital Signal Processing Group, Department of Electronic and Electrical Engineering, University College, Dublin.
41. Sumit Roy and John J. Shynk (1990), 'Analysis of the Momentum LMS Algorithm', IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 38, No. 12, pp. 2088-2098.
42. Thomas D.E. and Moorby P.R. (1991), 'The Verilog Hardware Description Language', Kluwer Academic Publishers, 1st Edition.
43. Ting L.K. (2006), 'Algorithms and FPGA implementation of adaptive LMS based predictors for radar pulse identification', Ph.D. dissertation, Queen's Univ. Belfast. N. Ireland, www.belfastuniv.com/thesis/thing.pdf.
44. Treichler J.R., Richard Johnson C. and Larimor M.G. (2001), 'Theory and design of Adaptive Filters', Prentice Hall of India Pvt. Ltd.
45. Veendrick H. (2000), 'Deep-submicron CMOS ICs - from basics to ASICs', Springer Publications.
46. Wai-Kai Chen (2006), 'The VLSI Hand Book', Taylor & Francis Publications.
47. Wang T. and Wang C.L. (1995), 'Delayed Least Mean Square Algorithm', IEEE Electronic Letters, Vol. 35, No. 7.
48. Wayne Wolf (1998), 'Modern VLSI Design', Prentice Hall of India, Second Edition.
49. Wei P.C. (2002), 'Comparative Tracking Performance of the LMS and RLS Algorithms for Chirped Narrowband Signal Recovery', IEEE Transactions on Signal Processing, Vol. 50, No. 7.
50. Weste N. and Harris D. (2004), 'CMOS VLSI Design: A Circuits and Systems Perspective', Addison Wesley Publications.
51. Widrow B. and Stearns S.D. (2002), 'Adaptive signal Processing', Pearson Education, Second Edition.
52. Widrow B., McCool J.M., Larimore M.G. and Johnson C.R. (1976), 'Stationary and Non stationary Learning Characteristics of the LMS', Proceedings of IEEE, Vol. 64, No. 8, pp. 1151-1162.
53. Zaknich A. (2003), 'Principle of adaptive filters and self learning systems', Springer Publishers.