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

Advanced Engineering Informatics

Elixir Adv. Engg. Info. 35 (2011) 2868-2870

ISSN: 2229-712X

Analysis of adjoint LMS adaptive filter using backend tools

N.J.R.Muniraj

Karpagam Innovation Centre, Karpagam College of Engineering, Coimbatore.

ARTICLE INFO

Article history:

Received: 7 April 2011;

Received in revised form:

20 May 2011;

Accepted: 27 May 2011;

Keywords

ASIC,
TSMC,
FXLMS,
GDSII,
LMS,
RTL.

ABSTRACT

The proposed Adjoint LMS (ALMS) techniques have been modeled using Verilog HDL. The model has been verified using test benches with a functional coverage of 95%. The results obtained has 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 2.3 square millimeters. The results obtained for ALMS adaptive filtering techniques using pipelining, parallel processing, low power techniques and floating point architectures have proven that the complexities in the industrial applications can be met if the design is implemented on ASIC.

Introduction

The ALMS algorithm is to get fastest convergence algorithm compared to other LMS algorithms (Eric Wan 1996). Here, the error is reduced quickly and SNR is improved compared to other adaptive algorithms. In ALMS, the error (rather than the input) is filtered through an adjoint filter of the error channel as shown in Figure 1.1, performance regarding convergence and maladjustments are equivalent (ASPT User Manual 2006).

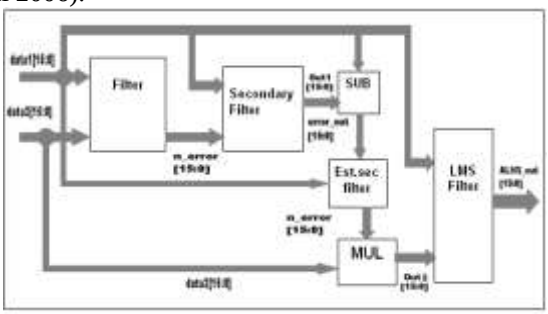

Figure 1.1 Designed architecture of ALMS

Compared to all LMS algorithms ALMS algorithm is the best one. Here, we use two secondary filters in the input side and one in feed back side. By this, the error is reduced compared to LMS algorithm. The iteration of the ALMS algorithm requires the following steps:

1. Filters the input vector $x(n)$ through the adaptive filter coefficients vector $\omega(n-1)$ to produce the filter output victory $y(n)$.
2. Filter $y(n)$ through the secondary path filter s to produce the secondary actuator response at the sensor $ys(n)$.
3. Evaluates the current error sampling $e(n) = d(n) + ys(n)$ .note the error, here, it is formed by adding the signal rather than subtracting them to be compatible with real world sensors such as microphones and accelerometers.
4. Filters the mirrored error vector $e(n)$ through the estimate of the secondary path se to produce the filter-error signal $fe(n)$.
5. Uses $x(n)$ and $fe(n)$ to calculate the normalized gradient vector and uses this to update the adaptive filter coefficients $\omega(n)$.
6. Supports both real and complex signals.

The wiener solution to the above problem is given by $w(\omega) = s(\omega)^{-1} p(\omega)$ w(ω) where $w(\omega)$ is the controller response at frequency ω. $s(\omega)$ is the response of the secondary path and $p(\omega)$ is the response of the primary of the secondary path at the same frequency. The adaptive controller will asymptotically approach this wiener solutions provided that $s(\omega)$ is a minimum phase functions (does not have zeros outside the unit circle) and the controller length is large enough to accommodate the above convolution if $s(\omega)$ is not minimum phase functions the adaptive controller. It will approach the causal part of the solutions if the controller is too short, then, the solutions will be truncated.

Simulation Results

With functional verification, the design is ready for physical implementation, the first step in physical implementation is to use the HDL model developed and converted into an RTL code that can be synthesized. Hence, the RTL model is synthesized

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

using industry standard EDA tool is called Design Compiler from Synopsys. DC is a 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.

Optimization constraints include operating frequency (clock period), input and output delays at the IOs. For the present work TSMC 130 nanometer target technology is adopted for better performances.

The constraints mentioned for the design are maximum operating frequency 81.8 MHz; total number of gates is 318394 cells, and power not to exceed more than 160.7 mW. Figure 2.1 shows the output, which was generated after simulation.

The constraints mentioned select the required gates from TSMC 130 nanometer library. The design requires 4 inputs and it produces 1 error corrected output.

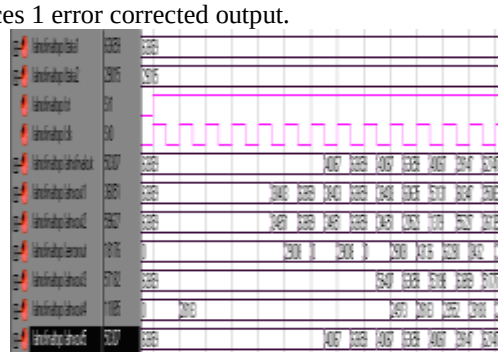

Figure 2.1Simulation report of ALMS architecture

Figure 2.2 shows the schematic, which was generated after synthesizing the RTL code.

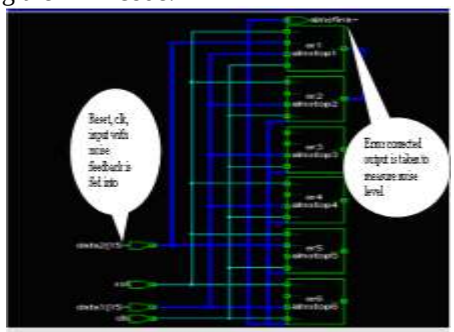

Figure 2.2 Synthesized schematic of ALMS architecture

Table 2.1Chip report c of ALMS architecture

Parameters	ALMS
Slack	1.43
Maximum critical paths	61
Number of ports	50
Number of nets	130
Combinational area in sq micron	200450.75
Sequential area in sq micron	22298
Total area in sq micron	264869
Total Dynamic Power	112.96mw
Cell Leakage Power	495.48uw

ALMS architecture requires less cell area, has 130 total numbers of nets to be routed and it consumes 113.5mw of power at 1pf capacitance load.

The chip report is shown in the Table 2.1. However, the effects in the chip report are being suitably reduced to a large extent by adopting optimized ASIC design methodology extended with proper floor planning, placement and routing being done semi automatically.

The total die size for ALMS architecture is 2.293*2.293, the final chip is shown in Figure 2.3.

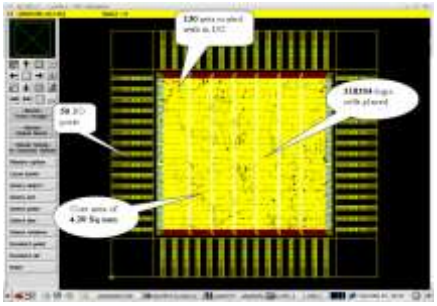

Figure 2.3 Final chip of ALMS architecture

Conclusion

Adaptive noise cancellation techniques such as LMS and RLS have been extensively used for noise cancellation techniques with good performances. These techniques have been extended for use in industrial applications, wherein there is a need for accuracy, speed, reliability and cost. Algorithm such as ALMS has been realized on ASIC. The proposed architectures has been modeled and verified for its functionality successfully. The models have taken through the entire ASIC flow. Suitable results are obtained at various stages of the ASIC flow using Synopsys.

The signal sampled at 1K samples per second has a data rate of 16 Kbits per second when fed through the proposed hardware which produces an output at 16 Kbits per 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 the ALMS adaptive filtering technique proposed in this research work is taken through physical design flow consisting of Floorplanning, Placement and Routing steps. The results obtained for ALMS architecture outperform at the speed of 82 MHz. The overall size of the entire chip is 4.20 sq mm with a gate count of 318394.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

Reference

1. ASPT User Manual Version 2.1, www.mathworks.com, 12th Oct 2006.
2. Basker J. (2004), 'Verilog HDL Primer', BS Publication, Second Edition.
3. Chinnery, David, Keutzer and Kurt (2007), 'Closing the power gap between ASIC and Custom tools and techniques for low power design', Springer Publications.
4. David Smith R. (2000), 'Verilog styles for synthesis of digital systems', Addison-Wesley Longman Publisher.
5. Emmanuel Ifeachor C. and Barrie Jervis W. (2002), 'Digital Signal Processing - A practical approach', Pearson Education Asia.
6. Eric Wan A. (1996), 'ALMS: An Efficient Alternative to the Filtered-XLMS and Multiple Error LMS Algorithms', ICASS P96, Vol. III, pp. 1842-1845.
7. Geizer R.L., Allen P.E. and Strader N.R. (1990), 'VLSI Design Techniques for Analog and Digital Circuits', McGraw Hill.
8. Hachtel G.D. and Somenzi F. (1996), 'Logic synthesis and Verification Algorithms', Springer Publications.
9. Haykin S. (1992), 'Adaptive Filter Theory', EnglewoodCliffs, NJ: Prentice-Hall.

10. Himanshu Bhatnagar (2002), 'Advanced ASIC Chip Synthesis Using Synopsys® Design Compiler® Physical Compiler® and PrimeTime®', Prentice Hall of India.
11. Khosrow Golshan (2007), 'Physical Design Essentials: An ASIC Design Implementation Perspective', Springer Publishers.
12. Mark Gordon Arnold (1999), 'Verilog Digital Computer Design', Prentice Hall PTR, 1st Edition.
13. Marque C., Bisch C., Dantas R., Elayoubi S., Brosse V. and Perot C. (2005), 'Adaptive filtering for ECG rejection from surface EMG recordings', Journal of Electromyography and Kinesiology, Vol. 15, pp. 310-315.
14. Mead C. and Convay L. (1999), 'Introduction to VLSI Systems', Addison Wesley Publication.
15. Meyer-Baese U. (2006), 'Digital Signal Processing with Field Programmable Gate Arrays', Second Edition, Springer International Edition.
16. Miguez-Olivares A. and Recuero-Lopez M. (1996), 'Development of an Active Noise Controller in the DSP Starter Kit', Texas Instruments.
17. Parhi K.K. (1999), 'VLSI Digital Signal Processing Systems', John Wiley & Sons, 1st Edition.
18. Parhi K.K. and Messerschmitt D.G. (1989a), 'Pipeline interleaving and parallelism in recursive digital filters-Part I: Pipelining using scattered look-ahead and decomposition', IEEE Trans. Acoustic, Speech, Signal Processing, Vol. 37, pp. 1118-1134.
19. Samir Panitkar (2003), 'Verilog HDL', Pearson Education, Second Edition.
20. 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.
21. Sebastian Smith M.J. (2001), 'Application-Specific Integrated Circuits', Fifth Edition, Pearson Education Inc.
22. Widrow B. and Stearns S.D. (2002), 'Adaptive signal Processing', Pearson Education, Second Edition.
23. 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.
24. Zaknich A. (2003), 'Principle of adaptive filters and self learning systems', Springer Publishers.