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

Advanced Engineering Informatics

Elixir Adv. Engg. Info. 40 (2011) 5117-5119

ISSN: 2229-712X

Backend design of LMS adaptive filter using cadence tools

N.J.R. Muniraj[1] and K.Sathesh[2]

[1]Tejaa Shakthi Institute of Technology for Women, Coimbatore, India

[2]Karpagam College of Engineering, Coimbatore, India.

ARTICLE INFO

Article history:
Received: 15 August 2011;
Received in revised form:
16 October 2011;
Accepted: 25 October 2011;

Keywords
Adaptive filters,
RTL compiler,
SoC Encounter,
GDSII system.

ABSTRACT

We have dealt with the Backend design of LMS algorithm in Cadence tools. The LMS algorithm is used for the purposes of Polyharmonic Power Calibrator, Active phase cancellation of Hostile radars, and Noise cancellation. We have implemented the algorithm into VLSI technology using Verilog HDL, designed, simulated, and synthesized it using Xilinx Spartan3 3s400pq208 and the backend design is done using Cadence tools. The backend design for the LMS adaptive filter is done using RTL-GDSII So C encounter system and the So C design of the LMS algorithm had a power consumption of 32mW, timing slack of about 14ps and the design frequency is about 4.2Mhz also it uses an area of about 9753microns with a core size of about Ratio (H/W) 0.9204437. It is found that the backend implementation to be more efficient than the other methods of implementation.

Introduction

Recent developments in Field–Programmable Gate arrays (FPGA) technology have changed the traditional method of implementations (Lok-Kee and Roger, 2005). Today, the computational task for the signal processors has been increased dramatically with added functions to perform the more intensive tasks. FPGA has grown over the past decade to the point where there is now an implemented on a single FPGA device. The Backend design of the algorithm is implemented using Cadence tools (SOC encounter).

Adaptive filtering has been a key enabling ingredient in many current as well as newly emerging communication technologies (Bernard and Samuel, 2002). Today Adaptive filtering techniques are used to overcome the channel limitations and also echo and active phase cancellation purposes (Haykins, 1996). Need for LMS algorithm is to reduce the noise in the communication channel. The LMS algorithm is the first algorithm that has been proposed by Widrow.

Algorithm

The LMS algorithm is a linear adaptive filtering algorithm that belongs to the family of the stochastic gradient algorithms (Haykins, 1996). The stochastic gradient algorithms differ from the steepest descent algorithms in that the gradient is not calculated deterministically. The LMS algorithm has two parts. In the first part, the output of a transversal filter is computed according to the tap inputs and the error term is generated according to the difference between the filter output and the desired response. In the second part, the adjustment of the tap weights is done according to the error term. The block diagram of the LMS algorithm is given in figure.

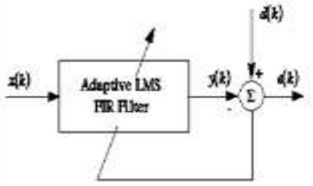


Fig 1 : Block Diagram of LMS algorithm

The algorithm forms a feedback loop by the error term fed back. The filter produces an output and the difference between the output and the desired term is obtained. This difference is the estimation error term. The estimation error is given to the Adaptation Control Block. Adaptation Control Block multiplies the estimation error with the input taps' complex conjugate and a step size. The results of the corresponding taps are added to the corresponding filter taps. So, the new filter is obtained (Proakis, 1995).

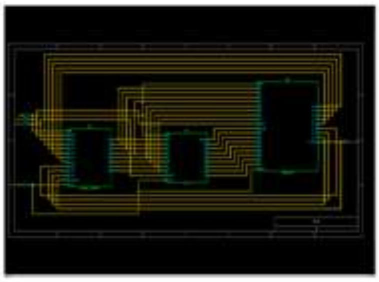
Implementation in cadence tools

The Cadence tools includes four processes: NC Launch – NC Sim & Simvision, RTL Compiler – Netlist, So C Encounter – GDSII, ICFB – Schematic Editor. Initially the Verilog HDL coding is done for the LMS algorithm and simulated in Xilinx as well as modelsim. The synthesis report is viewed for the algorithm implementation in Xilinx. The power analysis report is checked for the algorithm. Now the algorithm can be implemented in cadence tools. The three important steps involved in NC launch is Compile, Elaborate, and Simulate. Here the test bench and the verilog coding is compiled first and the test bench is chosen from the worksheet and elaborated. The final step is simulation and the waveform screen will open. Now it is forced and final output is obtained.

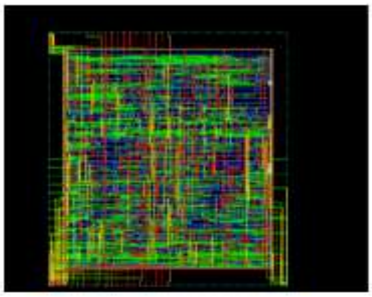
The RTL compiler is carried out after NC launch. The timing constraint file is included from the library and the project file is read and the elaborated. So that we get the following schematic Initially the timing report in RTL compiler will be unconstrained and the timing slack will be obtained after defining the clock.

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

The design frequency will also be obtained from the timing report.



The netlist file with extension of sdc should be generated. In order to implement the design in SOC Encounter system it is necessary to obtain the IOC file which decides upon I/O pin connections in IC. The next step in cadence tools is about working with the SoC encounter. It is necessary to include the max and min timing libraries. The LEF files have to be included and timing constraint file with the extension .sdc is to be included and finally the IOC file ie. the input/output assignment file is added. Then the advanced option is selected and power configurations are done. The power planning is done by specifying the power stripes and the rings. The floor planning is done in order to decide upon the die size and core size. The size should be specified so that minimum area is consumed and also the minimum power is consumed. The final step is nanorouting in order to place each active element in right place.



Then the GDSII file is created and the final design obtained will be while the implementation in FPGA will be having an analysis report of

Number of Slices:	87 out of 3584
Number of Slice Flip Flops:	112 out of 7168
Number of 4 input LUTs:	158 out of 7168
Number used as logic:	150
Number used as Shift registers:	8
Number of IOs:	26

Analysis Report obtained in Cadence tools is

Timing slack:	14ps
Design frequency:	4.42MHz
Power consumed:	32mW
Die size:	800*820microns
Die area:	9753microns
Core area:	0.9204437
Core utilization:	0.500466

The LMS algorithm is implemented in FPGA and the power consumption is identified to be 56mW and the number of slices used is about 87 which is 2% of the number of available slices and when it is implemented in FPGA it takes more time in order to get that final value of the coefficient so that the signal with minimum error can be obtained So in order to reduce the complexity in the design and also to reduce the power consumption and the timing complexities we go for the Cadence tools. The power consumption when implemented in cadence tools is about 32mW and the die size is about 800 * 820 microns. The timing slack calculated is about 14ps and the design frequency is about 4.42Mhz. The total are used is about 9753microns. Thus when we implement the algorithm in cadence tools the power consumed is lesser compared to the power consumption in FPGA and Matlab. Thus the most effective method for the implementation of LMS is possible with cadence tools.

Conclusion

Thus the Backend design of the LMS algorithm is done using in Cadence tools and analysis is done for the algorithm implementation. It could be concluded that the algorithm implementation in the Cadence tools is more efficient compared to the past methods of implementation. The analysis report provides the details of algorithm implementation in terms of power, area, and timing complexities.

References

[1] P.M. Clarkson, "Optimal and Adaptive Signal Processing". CRC Press, London, 1993.

[2] S.Elliot, I.Stothers, P.Nelson, “A Multiple Error LMS Algorithm and its Application of Sound and Vibration” ieee Transaction on Acoustics , Speech & Signal Processing, VOL ASSP-35, N010, October 1987.

[3] Immanuel Ifeachor-“A Practical Approach to Digital Signal Processing and its Application” – Prentice hall.

[4] Rabiner and Gold- “Digital Signal Processing and its Application”

[5] Ahmad R.S.and Bahai. (2002),” Comparing RLS and LMS Adaptive Equalizers for nonstationary wireless channels in mobile ad-hoc networks”, Dept of electrical engineering, Stanford university.

[6] Austin M.E.(1967),"Decision-feedback equalization for digital communication over disper¬sive channels," Tech. Rep. 437, MIT Lincoln Laboratory, Lexington, Mass., August.

[7] Bernard Widrow and Samuel D.Stearns. (2002), “Adaptive signal Processing”, Pearson Education. Second Edition.

[8] Bernard Widrow, Gregory plett, Edson Ferreira and Marcelo lamego. (1998),”Adaptive inverse control based on nonlinear adaptive filtering“, Information Systems Lab, EE Department, Stanford University.

[9] Belfiore C. and Park Jr. J. H.(1979), "Decision-feedback equalization," Proc. IEEE, vol. 67, pp. 1143-1156, August.

[10] Haykin S. (1994), Communication Systems. John Wiley and Sons.

[11] Haykin S. (1996), Adaptive Filter Theory, 3rd Edition, Prentice Hall, New Jersey.

[12] Lucky R. W. (1965), "Automatic equalization for digital communications," Bell Systems Technical Journal, vol. 44, pp. 547—588, April.

[13] Monsen, P.(1974), 'Adaptive Equalization of the Slow Fading Channel', IEEE Trans., Aug, IT-22, pp. 1064-1075.

[14] Proakis J. G. (1995), Digital Communications. New York: McGraw-Hill, 3rd ed.

[15] Qureshi S. U. H.(1985), "Adaptive equalization," Proc. IEEE.

[16] Sklar B.(1997), "Rayleigh fading channels in mobile digital communication systems part I: Characterization," IEEE Commun. Magazine, pp. 90-109, July

[17] Thomas Bruns, Nihar jindal, Raymond wang and Ahmad bahai.(2001), “A comparison between performance of RLS and LMS Adaptive Algorithms in Non-stationary Wireless

Channels”, Department of electrical engineering, Stanford University.

[18] Wan E.A.(1996), Adjoint LMS: An efficient alternative to the Filtered-X LMS and multiple error LMS algorithms. IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-96, vo1.3, pp.1842-1845.