

Design of FXLMS adaptive filter architecture using backend VLSI Technology

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 FXLMS techniques have been modeled using Verilog HDL; 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 Floor planning. 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 2.15 Square millimeters. The results obtained for NLMS 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.

© 2011 Elixir All rights reserved.

Introduction

A conventional adaptive algorithm such as the LMS algorithm is likely to be unstable to the phase shift (delay) introduced by the forward path. The well-known Filtered-X LMS-algorithm is, however, an adaptive Filter algorithm which is developed from the LMS algorithm by Rupp and Sayed (1995) where a model of the dynamic system between the Filter output and the estimate, i.e. the forward path is introduced between the input signal and the algorithm for the adaptation of the coefficient vector. The Filtered-XLMS algorithm is suitable for applications where a dynamic system exists between the filter output and the estimate as explained by Miguez-Olivares and Recuero-Lopez (1996). This can be derived from the standard LMS algorithm by commuting the order of the filter and the channel.

Design Architecture

This algorithm employs filtered version of the input signal values that are created by filtering every input signal. A compensated algorithm is obtained by filtering the reference signal to the coefficient adjustment algorithm using a model of the forward path (Douglas 1997a). The equivalent design architecture of FXLMS algorithm is shown in Figure 2.1.

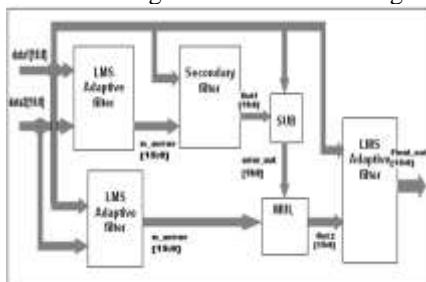


Fig. 2.1 Designed architecture of FXLMS

The basic principle behind the Filtered X-LMS algorithm is that the input vector $x(k)$ is filtered through the adaptive filter co-efficients vector $w(n-1)$ to produce the filter output vector $y(k)$ (Douglas 1997b). This output vector is passed through the secondary path filter to produce the secondary actuator response at the sensor $y(k)$.

The adaptive transversal filter output is evaluated along the error signal as said by Miyagi and Sakai (2001). The adaptive transversal filter coefficients are using the relationship:

$$w(k+1) = w(k) + \mu e(k)x(k) \quad (2.1)$$

The current error sample $e(n)$ is evaluated using the relationship

$$e(k) = d(k) + y(k) \quad (2.2)$$

$S(n)$ is the transfer function of secondary path. It is to be noted that error here is formed by adding the signal rather than subtracting them to be compatible with real world sensors such as microphones and accelerometers (Parhi 1999).

The input signal $x(k)$ is filtered through the estimate of the secondary path to produce the filtered-x signal $\hat{f}_x(k)$. Now $\hat{f}_x(k)$ and $e(k)$ are used to calculate the normalized gradient vector and this is used to update the adaptive filter co-efficients.

Here, the original input becomes filtered by the channel before entering the filter and the error appears directly at the output of the adaptive filter.

Results

The simulation results obtained from the Modelsim is shown in Figures 3.1 and 3.2 of FXLMS and PIPFXLMS 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 to an RTL code that can be synthesized (Basker 2004). Hence, the RTL model is synthesized using industry standard EDA tool is called Design Compiler from Synopsys. DC compiler is a sign of tool for synthesis, in this process, the RTL model is converted into gate level netlist (Veendrick 2000). The gate level netlist should be able to meet area, timing and power requirements (Himanshu 2002).

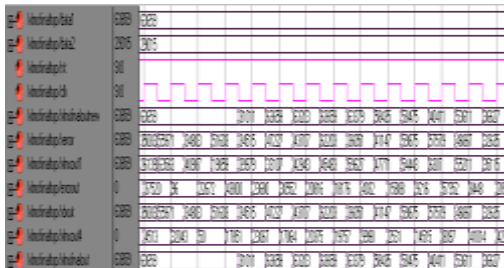


Figure 3.1 Simulation report of FXLMS architecture

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 as mentioned in literature (Douglas 2000). The constraints mentioned for the design are maximum operating frequency 79.2 MHz, total number of gates 264,618 cells, and power not to exceed more than 170mW. Figure 3.2 shows the schematic, which was generated after synthesizing the RTL code (Sebastian 2001).

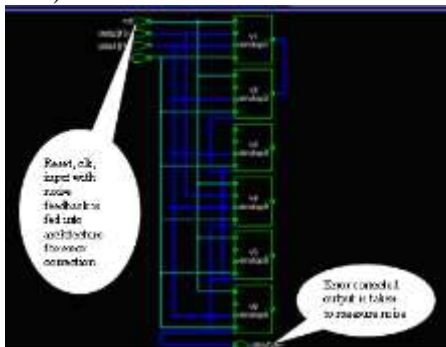


Fig. 3.2 Synthesized schematic of FXLMS architecture

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

FXLMS	Parameters
0.00	Slack
61	Maximum critical paths
50	Number of ports
178	Number of nets
187606.25	Combinational area in sq micron
21985.25	Sequential area in sq micron
249607	Total area in sq micron
170.56mw	Total Dynamic Power
544.48uw	Cell Leakage Power

Table 3.1 Chip report comparison of FXLMS architecture

FXLMS architecture requires less cell area, has 178 total numbers of nets to be routed and consumes 171 mw of power at 1pf capacitance load. However, the effects like chip size and power are being suitably reduced to a large extent by adopting optimized ASIC design methodology.

It is extended with proper floorplanning, placement and routing is being done semi automatically, the total die size for FXLMS architecture is 2.271*2.268, and the final chip is shown in Figure 3.3.

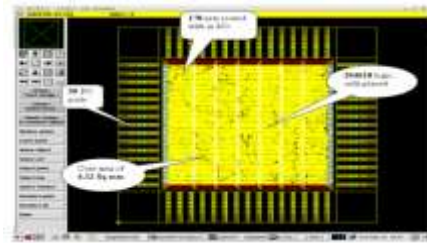


Fig. 3.3 Final chip of FXLMS architecture

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 FXLMS 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 FXLMS 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 FXLMS architecture outperform at the speed of 79MHz. The overall size of the entire chip is 4.12 sq mm with a gate count of 264618. 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. Basker J. (2004), 'Verilog HDL Primer', BS Publication, Second Edition.
2. Douglas S.C. (1997a), 'Fast Exact Filtered-XLMS and LMS Algorithms for multichannel Active Noise Control', Department of Electrical Engineering, University of Utah.
3. Douglas S.C. (2000), 'Reducing the computational and memory requirements of the multichannel Filtered-XLMS Adaptive Controller', Department of Electrical Engineering, University of Utah.
4. Himanshu Bhatnagar (2002), 'Advanced ASIC Chip Synthesis Using Synopsys® Design Compiler® Physical Compiler® and PrimeTime®', Prentice Hall of India.
5. Miyagi S. and Sakai H. (2001), 'Performance Comparison between the Filtered-Error LMS and the Filtered-X LMS Algorithms', Dept. of Systems Science, Kyoto University, Japan. Parhi K.K. (1999), 'VLSI Digital Signal Processing Systems', John Wiley & Sons, 1st Edition. Rupp M. and Sayed A.H. (1995), 'Modified FXLMS algorithms with improved convergence performance', Conf. Rec. 29th Asilomar Conf. Signals, Syst., Comput., Vol. 2 (LosAlamitocs, CA., IEEE Comput. Soc. Press), pp. 1225-1259. Sebastian Smith M.J. (2001), 'Application-Specific Integrated Circuits', Fifth Edition, Pearson Education Inc. Veendrick H. (2000), 'Deep-submicron CMOS ICs - from basics to ASICs', Springer Publications.