**Omid SHARIFI-TEHRANI**

Young Researchers Club, Majlesi Branch, Islamic Azad University, Majlesi, Iran

# Hardware Design of Image Channel Denoiser for FPGA Embedded Systems

***Abstract***. *In this article an FPGA-based image channel denoiser using a 1D-standard-LMS algorithm is proposed. The designed core is written in VHDL93 language as basis of 1D-FIR adaptive filter. The proposed core is FPGA-brand-independent, hence can be ported on any brand to create a system-on-chip (SoC). Although using a pure-hardware implementation results in better performance, it is more complex than other structures such as digital signal processors and Hardware/Software co-designs. The results show improvements in area-resource utilization and convergence speed in the designed pure-hardware channel denoiser core.*

***Streszczenie.*** *Zaproponowano metodę usuwania szumów w systemie FPGA bazującą na algorytmie LMS i 1D-SOI filtrze adapatacyjnym. Przedstawiono możliwości zastosowania metody a gotowym układzie zintegrowanym. (**Projekt układu odszumiania w systemie FPGA**).*

**Keywords:** Active Filters, Field Programmable Gate Array, Image Communication, Image Denoising, System-On-a -Chip.
**Słowa kluczowe:** FPGA, filtr adaptacyjny, usuwanie szumu.

## Introduction

One of the most important branches of signal processing is adaptive image processing. The main goal of adaptive processing is type of systems which are adaptively changed and self-modified. They can enhance their performance by adaptively learning the characteristics and adjusting their coefficients. Adaptive systems are greatly used in many domains such as communication systems, channel estimation, equalization, sonar, smart antenna, navigation systems, prediction, system identification and active noise control (ANC) systems. Adaptive systems perform an important role in their areas. In some cases particularly where non-stationary and time-varying signals are focused, the significance of adaptive processing is clear [1]. Recently requests for portable, embedded digital signal processing (DSP) systems and system on programmable chips (SoPC) have been increased surprisingly. Applications like audio devices, hearing aids, mobile phones and active noise control systems with constraints like power consumption, speed and area, need an implementation by which these constraints are obtained in a short time. Some possible solutions are ASIC chips, general purpose processor (GPP) and digital signal processor. Although the first option can give a solution to cover hard constraints, it is missing the flexibility that is available in the two others. Utilizing field programmable gate arrays (FPGAs) can reduce the gap between flexibility and high performance [2]. New FPGAs include many primitives that provide DSP applications like multiply and accumulate units (MAC), embedded multipliers, digital clock management (DCM), DSP-Blocks, and soft/hard processor cores (such as PPC). These facilities are embedded in FPGA and improved for high performance applications and low power consumption. The accessibility of soft/hard core processors in new FPGAs allows implementation of DSP algorithms without difficulty [3]. An optional choice is to transfer some parts of the algorithm into hardware (HW) to enhance performance. This is called HW/SW co-design. This solution can result in a better implementation as part of the algorithm is hastened by HW while the flexibility is well-kept. Another more effective and more complicated option is to transfer the whole algorithm into hardware as a pure HW implementation [4, 5]. Despite the fact that, this is an interesting option regarding area, speed, performance and power consumption, the design will be much more complicated [6, 7]. Studies on LMS algorithm principally focus on two outlooks. One is the convergence time from the theoretical viewpoint; some modified LMS algorithms have been proposed in references [8, 9]. The other is hardware implementation, in order to improve data throughput; numerous modified structures for LMS algorithm like pipeline technique were proposed [10-12]. This article can be categorized into the latter. In serial image transmitting, digital image data are sent through communication channels and as a rule of thumb they are degraded with noise and interferences. Channel noise cancellation can be used to identify the channel characteristics and improve image quality [6]. An efficient architecture for singular value decomposition (SVD) based on Brent, Luk, Vanloan (BLV) systolic array, was proposed by A. Ahmedsaid and A. Amira for image de-noising applications. Among all unitary transformations, SVD is optimal for image in terms of energy packed in a given number of transformation coefficients is maximized. One of the applications of SVD is block SVD image filtering by which, SVD is used for each partitioned image block. The results showed good de-noising capability without blurring the image [13]. Z. Min et al. used FPGA technology for designing fast median filter and de-noising processing of images. Also they did analysis of rough sets and the processing of incomplete information in which, the image is divided into target and background areas. Their experimental results showed that parallel processing of image and pipelining based on FPGA can save time and meet the real-time necessities [14]. Standard median filter and multi-level median filter which can keep image characteristics and thin lines, have also been implemented using FPGA technology. It has led to real-time image filtering and a general purpose solution [15]. A new FPGA implementation of bilateral filter was proposed by A. Gabiger et al. in 2009. With the help of this design, the bilateral filter can be implemented as a highly parallelized pipeline structure with efficient resource utilization. Their design innovation is sorting the input data into groups so that kernel-based processing is feasible. It was shown that at least 3db of noise reduction was achievable without loss of image features [16]. J.M. Ramirez et al. proposed an FPGA implementation of linear and morphological image filtering using Xilinx Spartan3E and Nexys II. The system takes advantages of available resources in a Nexys II system based on Spartan3E chips [17]. B. Rajan and S. Ravi proposed a reconfigurable computing for signal and image processing applications with high flexibility, performance and adaptability. Reconfiguration is described by how fast the reconfiguration can be done and how many possible reconfiguration can be utilized. This is called dynamic reconfiguration [18]. They showed that the overall run-time of implementing the image filter on an Spartan-IIE

FPGA, including bus overhead, is up to 400 times faster than a software implementation on a 2.8 GHz Pentium processor.

In this article we first describe the theory of adaptive signal processing and LMS algorithms. Then, description of the designed FPGA-based fixed-point 1D-Standard-LMS algorithm and its use in channel noise cancellation implementation are given respectively. Finally, simulation-implementation results and conclusions from the obtained results are given.

**LMS algorithm**

According to LMS algorithm theory [19] and Fig. 1, the error of filter is calculated by (1).

(1) $$e_n = d_n - y_n$$

(2) $$y_n = W_n^T U_n$$

The error is simply the desired output ($d_n$) minus the filter output ($y_n$). $W_n$ is the filter weight vector and $U_n$ is the filter input vector through taped delay line (TDL). The weight update equation is defined by:

(3) $$W_{n+1} = W_n + \eta e_n U_n$$

This is the Widraw-Haff LMS weight update algorithm [1, 2]. $\eta$ is called learning rate or step-size by which the convergence speed and stability of filter is adjusted. Evaluating the effect of choosing different $\eta$ values and its relation to eigenvalue spread of input signal is beyond the scope of this article.

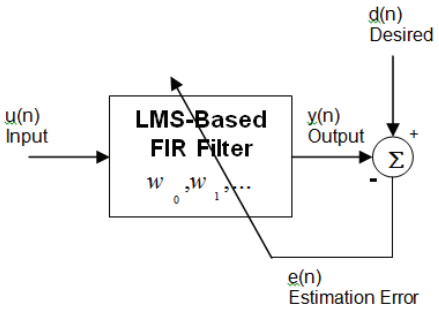


Fig. 1. Block Diagram of Adaptive FIR Filter

**Fixed-point standard LMS model**

In the designed core, it was considered that analog to digital converter (ADC) unit provides 12-bit signed binary output , so the model for input data-bit allocation was designed in two's complement form with 17-bit length as in Table 1. One bit is for sign bit, 12 bits are for representing integer part of number and one bit fraction-length is dummy and not used for input/output data, but is necessary for weight update. Table 2 demonstrates the LMS weight bit-allocation. As is evident from Table 2, the most bits are assigned for fraction part and it is because filter weights are normally between -1 to +1. The output of LMS filter is calculated by equation (2) and is accomplished by FPGA internal MULT18X18 multiplier block, thus the LMS output will be 34-bit long. To fit in 17-bit long format and having minimum discarding error, Y is truncated from 31 down to 15 bit-indexes. As mentioned beforehand, weights are updated based on equation (3), hence, the resulting length for W is 51-bit long. Again to fit in 17-bit long format and having minimum discarding error, W is truncated from 33 down to 17 bit-indexes.

Table 1. Input Data Bit-Allocation

| Sign Bit | Guard Bits | Word Length | Fraction Length |
|---|---|---|---|
| 1 | 3 | 12 | 1 |

Table 2. Weights Bit-Allocation

| Sign Bit | Word Length | Fraction Length |
|---|---|---|
| 1 | 1 | 15 |

**LMS entity**

The designed LMS entity is composed of reset, clock, adc_new_data, desired, input and output ports as illustrated in Fig. 2:

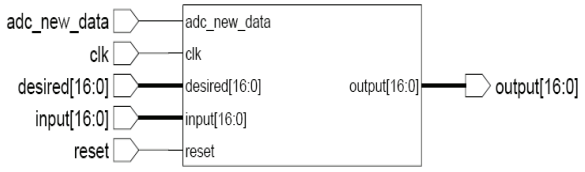


Fig. 2. Entity of Fixed-Point Standard-LMS Channel Noise Canceller

Reset, clock and adc_new_data signals are one-wire ports while the others are 17-bit long ports. The reset signal is used for resetting the algorithm and bringing it to initial values. The clock signal should be connected to processing clock. Because the algorithm should know when ADC has converted a new data, a signal called adc_new_data is provided so that it is asserted for a short time and then de-asserted. When working, first the output of the core is calculated, then is compared to the desired value and the error is obtained. The error is used to update the coefficients of the filter. This process is done repetitively. The DFD (Data Flow Diagram) of designed LMS-based FIR core is illustrated in Fig. 3.

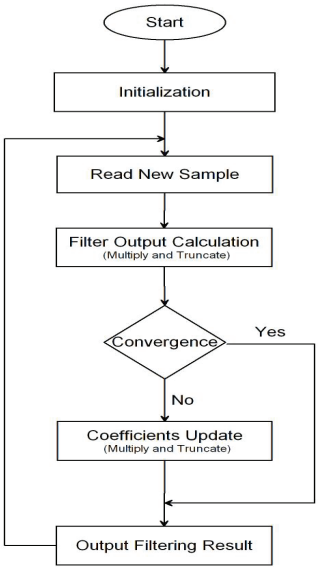


Fig. 3. Data Flow Diagram of LMS-based Channel Noise Canceller Core

**Simulation of proposed model**

The performance of the proposed model is evaluated by using it as an adaptive-FIR channel identifier with 64 coefficients.

1- Test data creation

For simulation and verification of designed core, there should be some image test data. MATLAB was used to create the essential image test data. An M-file is written to write the results in separate text files to use in ModelSim VHDL simulator. The designed core VHDL file and test-bench file is compiled in ModelSim Simulation software as VHDL93 standard. At the end of simulation the output data which has been written in a text file is brought to MATLAB workspace and converted from double to int16.

2- Software simulation results

In our simulation an image containing Persian words, "دوربين تلويزيوني رنگي" is degraded with normally distributed random numbers as channel noise. Then the designed core tries to identify the channel characteristics and remove the channel noise from image. $\eta$ is chosen $1/2^{15}$. Fig. 4 shows the results and Table 3 gives the obtained performance.

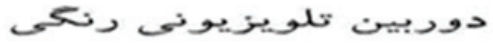

Fig. 4. (a) Clear Image

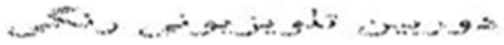

Fig. 4. (b) Noisy Image (Transmitted Through Comm. Channel)

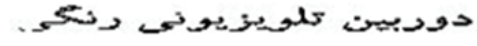

Fig. 4. (c) Reconstructed Image by Channel Identification

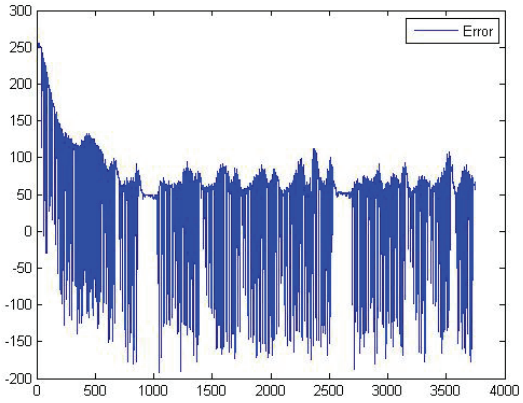


Fig. 4. (d) Residual Error (Identification Curve) – Error which is the difference between filter output and desired output, versus Number of Iterations.

Table 3. Simulation Performance

| Input SNR | Output SNR | SNR Enhancement |
|---|---|---|
| 0.0264 dB | 8.4573 dB | 8.4309 dB |

**Synthesis-implementation results**

The proposed LMS-based image channel noise cancellation core can be synthesized and implemented on any FPGA brands and families. Here for instance, the proposed core with 64 coefficients is implemented on XILINX VIRTEX5 (XC5vlx50t) chip using XILINX ISE software version 10.1. Resource utilization results are given in Table 4.

Table 4. Implementation-Resource Utilization on xc5vlx50t-3ff665

| | Used | Available | Utilization |
|---|---|---|---|
| Slice Registers | 2,618 | 28,800 | 9% |
| Slice LUTs | 2,564 | 28,800 | 8% |
| occupied Slices | 1,264 | 7,200 | 17% |
| Bonded IOBs | 54 | 360 | 15% |
| BUFG/BUFGC | 1 | 32 | 3% |
| DSP48Es | 2 | 48 | 4% |
| Max. Frequency | | 103.581 MHz | |

**Conclusion**

Modified fixed-point image channel noise cancellation core for FPGA-hardware implementation with low resource utilization was proposed. Some data problems were studied and effective methods were discussed. The designed core is based on LMS-FIR structure with 64 taps length and 16-bits signed coefficients. According to the results obtained, the filter have successfully adapted and learned the environment statistics with a fast convergence speed. Comparisons with other implementations showed better convergence speed and lower resource utilization on FPGA. The proposed model is FPGA-brand independent so can be implemented on any FPGA brand (XILINX, ALTERA, and ACTEL). Although using a pure-hardware implementation results in better performance than software or HW/SW co-design implementation, it is more complex and low flexible. Future study would be focused on implementing variable step-size Block-LMS algorithm (VSS-BLMS) and using soft/hard processor cores.

REFERENCES


[1] O. Sharifi-Tehrani, Novel Hardware-Efficient Design of LMS-based Adaptive FIR Filter Utilizing Finite State Machine and Block-RAM, *Przeglad Elektrotechniczny*, 87 (2011), No. 7, 240-244.

[2] O. Sharifi-Tehrani and M. Ashourian, An FPGA-Based Implementation of ADALINE Neural Network with Low Resource Utilization and Fast Convergence, *Przeglad Elektrotechniczny*, 86 (2010), No. 12, 288-292.

[3] S.M. Kuo, D.R. Morgan, "Active noise control a tutorial review," IEEE Proc., Vol. 78, pp. 943-973, 1999.

[4] O. Sharifi-Tehrani, M. Ashourian, P. Moallem, "An FPGA-Based Implementation of Fixed-Point Standard-LMS Algorithm with Low Resource Utilization and Fast Convergence," J. Inter. Rev. on Comp. and Soft. (IReCOS), Vol. 5(4), pp. 436-444, 2010.

[5] D. Alberto, E. Falletti, L. Ferrero, R. Garello, M. Greco, M. Maggiora, "FPGA implementation of digital filters for nuclear detectors," J. Nuclear Inst. and Meth. in Phys. Res., Vol. 611, pp. 99-104, 2009.

[6] Y.H. Seo, H.J. Choi, J.S. Yoo, D.W. Kim, "An architecture of a high-speed digital hologram generator based on FPGA," J. Sys. Arch., Vol. 56, pp. 27-37, 2010.

[7] K.S. Chaitanya, P. Muralidhar, C.C. Rama Rao, "Implementation of reconfigurable adaptive filtering algorithms," International Conference on Signal Processing Ssytems (ICSPS), Singapore, 2009, pp. 287-291.

[8] T. Aboulnasr, "A robust variable step-size LMS-type algorithm: analysis and simulations," IEEE Trans. on Signal Proc., Vol. 45(3), pp. 631-639, 1997.

[9] W. Peng, B. Farhang-Boroujeny, "A new class of gradient adaptive step-size LMS algorithm," IEEE Trans. on Signal Proc., Vol. 49(4), pp. 805-810, 2001.

[10] K. Matsubara, K. Nishikawa, "A New Pipelined Architecture of the LMS Algorithm without Degradation of Convergence Characteristics," IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP), Munich, Germany, 1997, Vol. 5(4), pp.4125-4128.

[11] H. Zheng-wei, X. Zhi-yuan, "Application of pipeline technique in realization of LMS algorithm," J. North China Electric Power Univ., Vol. 31(3), pp. 93-96, 2004.

[12] Y. Yi, R. Woods, L.K. Ting, C.F.N. Cowan, "High speed FPGA-based implementations of delayed-LMS filters," J. VLSI Sig. Proc., Vol. 39, pp. 113-131, 2005.

[13] A. Ahmedsaid, A. Amira, "Accelerating SVD on reconfigurable hardware for image denoising," International Conference on Image Processing, 2004, pp. 259-262.

[14] Zh. Min, L. Feng, L. Jian-yu, "The research of real-time image clarity processing method based on FPGA," IEEE Int. Sym.on IT in Medicine & Education (ITIME '09), Jinan, 2009, pp. 1302-1306.

[15] Y. Hu, H. Ji, "Research on image median filtering algorithm and its fpga implementation," WRI Global Congress on Intelligent Ssytems (GCIS '09), Xiamen, 2009, pp. 226-230.

[16] A. Gabiger, M. Kube, R. Weigel, "A synchronous FPGA design of a bilateral filter for image processing," Annual Conference on Industrial electronics (IECON '09), Porto, 2009, pp. 1990-1995.

[17] J.M. Ramirez, et al., "An fpga-based architecture for linear and morphological image filtering," 20th International Conference on Electronics, Communications and Computer (CONIELECOMP), Cholula, 2010, pp. 90-95.

[18] B. Rajan, S. Ravi, "Fpga based hardware implementation of image filter with dynamic reconfiguration architecture," Inter. Journ. of Comp. Scie. and Net. Secu., V. 6(12), 121-127, 2006.

[19] S. Haykin, Adaptive Filter Theory. Prentice Hall, Upper Saddle River, NJ, 2002.



***Author**: Omid Sharifi-Tehrani, Ph.D. candidate in electrical engineering, E-mails: omidsht@sel.iaun.ac.ir, omid-sharifi-tehrani@ieee.org, omidsht@yahoo.com*