The International Journal Of Engineering And Science (IJES)
||Volume||2 ||Issue|| 6 ||Pages|| 74-80||2013||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

Design and Implementation of Fpgabased Adaptive Filter

[1]Dr. RaaedFaleh Hassan, [2]Ali SubhiAbbood
[1]*Collage of Elec. & Electronic Techniques, Foundation of Technical Education, Baghdad, Iraq*
[2]*Collage of Elec. & Electronic Techniques, Foundation of Technical Education Baghdad, Iraq*

----------**ABSTRACT**----------
The work presented in this paper concerned with the design and implementation of adaptive Finite Impulse Response (FIR) filter using Field Programmable Gate Array (FPGA). The adaptation algorithm is accomplished by using Genetic Algorithm (GA). Genetic Algorithm has been chosen as a reliable and robust adaptation algorithm comparing with the conventional algorithms such as Least Mean Square (LMS) and Recursive Least Square (RLS).The combination of the filter structure and its adaptation algorithm based on GA, has been implemented usingFPGA device.Xilinx XC6VLX760-VERTIX-6 FPGA starter kit device is used as a target device.The Genetic Algorithm based adaptive FIR filter have been coded in VHDL (Very High Speed Integrated Circuits Hardware Description Language). The simulation results shows thatthe steady state Mean Square Error (MSE) of the proposed 32 bit floating point representation GA based adaptive filter implemented on FPGA is 1.1861e-009 compared with a 0.0029 and 0.0015 for the LMS and RLS based adaptive filters respectively.

KEYWORDS: *Digital Signal Processing, Genetic Algorithm, adaptive filter, Active Noise Control, FPGA.*

Date of Submission: 03 June 2013, Date of Publication: 25.June.2013

I. INTRODUCTION

Employing adaptive filters overcomes the difficulties and drawbacks associated with the using of a conventional digital signal processor, or digital filters for processing data with inadequate statistics or when the statistical characteristics of the input data are known to change with time[1].

The "adaptive filter" can be defined as a system which is trying to adjust its parameters so as to respond to some criteria with the aim of meeting some well-defined goal or target which depends upon the state of the system as well as its surrounding [2]. Adaptive filter play an important role according to the following properties: first, it can work effectively in unknown environment; second, it is used to track the input signal of time-varying characteristics [3].

There are various algorithms and approaches that may be used as an adaptation algorithm depending on the requirements of the problem. However, the main approaches to the linear adaptive filter algorithmsare Least Mean Squares (LMS) algorithm and Recursive Least Squares (RLS) algorithm, while the main approaches for nonlinear adaptive filter algorithm are the adaptive Volterra filter, and the Non classical Adaptive Systems such as Artificial Neural Networks (ANNs), Fuzzy Logic (FL) and Genetic Algorithms (GAs) [4].

The first two approaches are very good in locating local minima but unfortunately they are not designed to discard inferior local solutions in favor of better ones. Therefore, they tend to locate minima in the locale of the initialization point. In recent years, a variety of algorithms have been proposed for global optimization including *stochastic* or *heuristic* algorithms [5]. Recombination and mutation operations of the Genetic Algorithm will usually produce solutions that are outside this space of local optima [6].

The paper organization is as follows: Section 2 gives an overview of related work. Section 3 presents theoretical background of adaptive noise cancellation system.Section 4 gives an introduction to theGenetic Algorithm (GA). Section 5 illustrates the FPGA implementation of genetic algorithm based adaptive filter, Section 6 shows the simulation results to the design of adaptive filter using genetic algorithm optimization. Finally, Section 6presents a conclusion from this work.

II. RELATED WORKS

For related worksconcern with the FPGA implementation of adaptive filters, in [7], the authors has proposed a design and implementation of high throughput adaptive digital filter using Fast Block Least Mean Squares (FBLMS) adaptive algorithm on FPGA, in [8], the Widrow-Hoff LMS algorithm is implemented on the Spartan-3-XC3S400 FPGA board for adaptive noise cancellation (ANC) system, in [9], the authors has presented the implementation of an infinite impulse response (IIR) adaptive filter with particle swarm optimization (PSO) on field programmable gate array (FPGA), and finally in [10], the design and

implementation of recursive least square (RLS) Identification Algorithm based on FPGA has been discussed. While the related works concern with the genetic algorithm based adaptive filters, in [1], the authors presents the use of the GA based adaptive (IIR) filter for system identification, in [11], An approach to (IIR) filtering based on genetic algorithms is detailed for system identification, and finally in [12], The authors have presented the results of using a genetic algorithm to adaptively modify the coefficients of an IIR filter for system identification of two-pole model.

Due to its high speed and shorter development time, FPGA will be considered in this paper for implementing of a GA based adaptive FIR filter for noise cancellation system.

1. Adaptive Noise Cancellation

The goal of this system is the cancellation of disturbing noise without affecting original sound signal. The basic system block diagram is shown in figure (1): The primary signal d(n) consists of the superposition of noise signal s(n) and the desired signal n(n). The reference signal x(n) is noise signal measured at the noise source. The output signal y(n) is an estimate of the noise signal with inverted sign. In the headphones this signal and the primary signal are superposed, so that the noise signal is cancelled. The error signal e(n) is the result of this superposition[8].

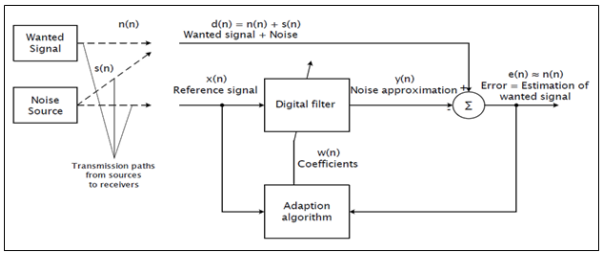

Fig.1: Adaptive Noise Cancellation system [8].

2. Genetic algorithms (GAs): Basic Concepts

Genetic algorithms are search algorithms based on natural selection processes and reinforcement of the species. Because of their power and ease of implementation, genetic algorithms have been widely applied to combinatorial optimization problems, i.e., problems where complexity increases exponentially with the problem dimensions [13].

Genetic Algorithms (GAs) have several advantages over traditional search and optimization algorithms. These advantages stem in part from its ability to maintain simultaneously information about a variety of points in the solution space. This helps prevent the GA from being trapped at inferior local minima. Another feature of GAs is their use of building blocks in creating new solutions. This allows a GA to take advantage of high quality sub-solutions that may already be presented in existing solutions [14]. Figure (2) illustrates the local and global optima [6]. While figure (3) shows the GA cycle [15].

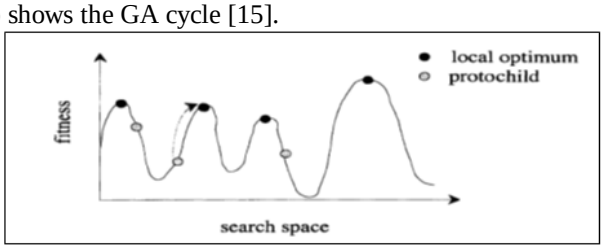

Fig.2: The local and global optima [6].

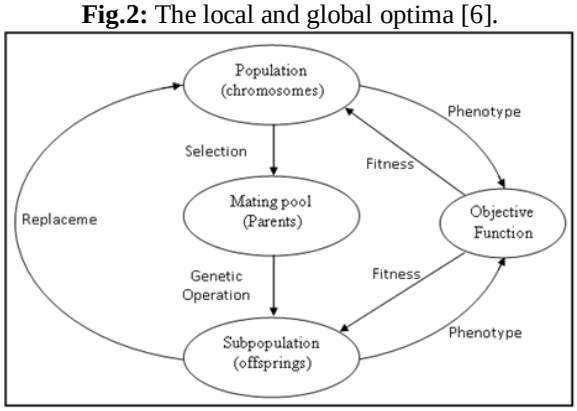

Fig.3:GA cycle [15].

In this paper, equation (1) has been used as a fitness function.

$$fitness = \sum_{n=0}^{N} [d(n) - y(n)]^2 \qquad (1)$$

3. FPGA implementation of genetic algorithm based adaptive filter

The main reason behind the popularity of the FPGA is due to balance that FPGAs provide the designer in terms of flexibility, cost, and time-to-market [7]. In this paper we depend on Randy L. Haupt and Sue Ellen Haupt[16] to build the genetic algorithm processor (GP) of the proposed system. The FPGA-based adaptive filter using genetic algorithm (GA) is organized as a ten modules, Control module, Memory module, Random Number Generator module, Sort fitness module, Selection module, Crossover module, Mutation module, Fitness Evaluation module, Floating point Multiplication module, and Floating point Adder module. The architecture of proposed system is shown in figure (4).

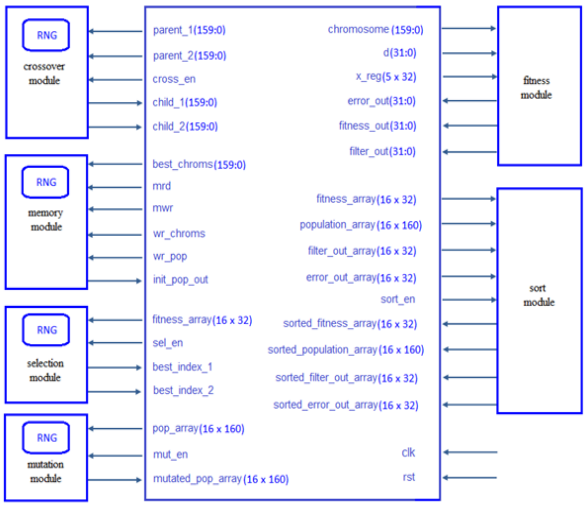

Fig. 4: The architecture of proposed system.

In this paper, the genetic algorithm based adaptive filter has the following parameters as illustrated in table (1) :

Table 1: Genetic Algorithm parameters.

parameter	value
Floating-point format	IEEE single precision format
Adaptive filter length (N_{var})	5 taps
GA population size (N_{pop})	16
No. of bits in a gene (N_{gene})	32 bit
Chromosome length (N_{bits})	$N_{var} * N_{gene}$ = 160 bit
No. of generations	15
Crossover operator	Single point crossover
Selection type	Tournament selection
Selection rate (X_{rate})	0.5
Mutation rate (μ)	0.025

3.1. Memory Module

The GA starts with a group of chromosomes known as the population. The population has N_{pop}chromosomes and is an N_{pop}x N_{bits}matrix filled with random ones and zeros[16]. The memory module is responsible for

- Randomly generateaN_{pop}x N_{bits}matrix of initial population and store the initial population.
- Store the best chromosome (best set of coefficients) of the previous sample to use it as a base to generate the initial population of the current sample.

3.2. Fitness Evaluation Module

After the memory module generate the initial population, the Fitness Evaluation Module is responsible for :

- Calculate the adaptive filter output *y(n)* for each individual in the population such that :
  $y(n) = w^T(n)\, x(n)$(2)
  Where$w^T(n)$ represent thecoefficients(one individual of the population) of the adaptive filter and *x(n)* represent the noise input signal at sample *n*.
- Calculate the adaptive filter output error *e(n)* for each individual in the population such that :

$e(n) = d(n) – y(n)$ (3)

Where*d(n)* represent the desired input signal at sample *n.*

- Calculate the fitness for each individual in the population such that $fitness = \sum_{n=0}^{N-1} e(n)^2$ (4)

3.3. Floating point Multiply and adder Modules

While the floating-point arithmetic has been used, a Floating point Multiplication Module and a floating point adder modulehave been building.

3.4. Sort Module

The Sort Module is responsible for ranking the fitness array and associated with the population array, the filter output array and the error output array from lowest fitness to the highest fitness. The sort algorithm that has been used in this work called the selection sort algorithm.

3.5. Selection Module

The selection module receives the ranked fitness array from the sort module. Then only the best are selected to continue, while the rest are deleted. The selection rate (X_{rate}=0.5) is the fraction of N_{pop} that survives for the next step of mating. The number of chromosomes that are kept each generation is[16] :

$N_{keep} = X_{rate} * N_{pop}$(5) Of the N_{pop}chromosomes in a generation, only the top N_{keep} survive for mating and the bottom $N_{pop} – N_{keep}$ are discarded to make room for the new offspring.Now it is time to select two chromosomes from the mating pool of N_{keep} chromosomes to produce two new offspring. Paring takes place in the mating population until $N_{pop} – N_{keep}$ offspring are born to replace the discarded chromosomes.

In this paper, the Tournament selection method has been used toselect two chromosomes from the mating pool of N_{keep} chromosomes to produce two new offspring. The tournament selection method randomly picks a small subset of chromosomes (the tournament size) from the mating pool, and the chromosome with the lowest cost (fitness) in this subset becomes a parent. The tournament repeats for every parent needed.

3.6. Crossover Module

The two chromosomes from the selection module now used as inputs to the crossover module. A one point crossover is randomly selected between the first and last bits of the parent's chromosomes.First, $parent_1$ passes its binary code to the left of that crossover point to $offspring_1$. In a like manner, $parent_2$ passes its binary code to the left of the same crossover point to $offspring_2$. Next, the binary code to the right of the crossover point of $parent_1$ goes to $offspring_2$ and $parent_2$ passes its code to $offspring_1$ [16].Figure (5) illustrates the crossover process.

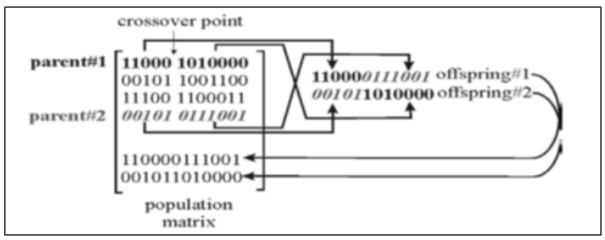

Fig.5: Two parents mate to produce two offspring [16].

The offspring are placed into the population. Consequently the offspring contain portions of the binary codes of both parents. The parents have produced a total of N_{pop}- N_{keep}offspring, so the chromosome population is now back to N_{pop}[16].

3.7. Mutation Module

A single point mutationchanges a 1 to a 0, and vice versa. Mutation points are randomly selectedfrom the $N_{pop}*N_{bits}$population matrix. In this paper,a mutation rate (μ) of 0.025 has been used. In this case the number of mutations is given by [16]:

$\#mutations = \mu * (N_{pop} - 1) * N_{bits}$ (6)

Thus a random number generator creates (*#mutations)* pairs of random integers that correspond to the rows and columns of the mutated bits [16].

3.8. Random Number Generator Module

The random number generator (RNG) module is used withfour of the major modulesof the Genetic Processor (GP).

- Firstly, with the memory module during the initial population creationphase to create a diverse initial population.

- Secondly, during the selection phase to select random individuals for crossover and mutation operations.
- Thirdly, during the crossover phase to select a random crossover point.
- Fourthly, during the mutation phase to select random bits for flipping from 0 to 1 or vice versa.

The random number generator (RNG) module has been implemented using linear shift register (LSHR) based random number generator. This kind of generator is easy to implement and product fairly good pseudo-randomness [17]. The block diagram of the LFSR is shown in Figure (6).

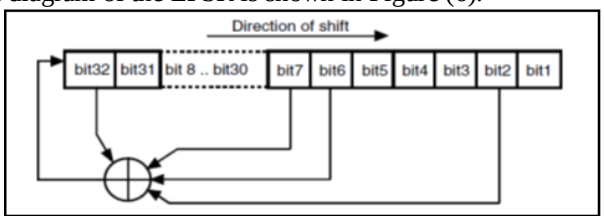

Fig. 6:Linear feedback shift register RNG [13].

3.9. Control Module

The control module is the main synchronization or the controlling unit. It is modeled as a finite state machine (FSM). All other modules interact and exchangeinformation only through the control unit only.Figure (7) illustrates the Finite State Machine (FSM) of the control module.

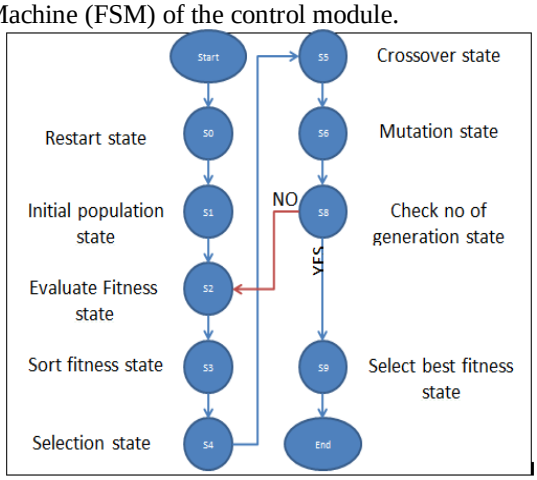

Fig. 7: Control Module.

III. RESULTS

The signals used in the verification of the workability of the genetic algorithm based adaptive filter are a speech signal with amplitude of normalized value in the range between +1 and -1. The noise signal is a Gaussian noise with mean 0 and variance of 0.1. A five tap FIR filter has been used to simulate the unknown noise path in the Acoustic Environment. By using HDL Co- simulation block provided in Simulink library of the MATLAB (2010),a Co-simulation between MATLAB or Simulink and a Hardware Description Language (HDL) simulator has been setting up.Figure (8) represents the design summery of the proposed system.Figure (9) illustrates the Co-simulation between Simulink andModelSim while figure (10) represents the genetic algorithm based adaptive filter simulation result.The *MSE* of the GA based Adaptive Filter compared with the *MSE* of the LMS and RLS adaptive filters are shown in figure (11).

call_comp Project Status (07/17/2012 - 18:50:32)			
Project File:	final_work_32bit.xise	**Parser Errors:**	No Errors
Module Name:	call_comp	**Implementation State:**	Synthesized
Target Device:	xc6vlx760-2ff1760	**• Errors:**	No Errors
Product Version:	ISE 13.3	**• Warnings:**	No Warnings
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	19653	948480	2%
Number of Slice LUTs	104893	474240	22%
Number of fully used LUT-FF pairs	11869	112677	10%
Number of bonded IOBs	130	1200	10%
Number of BUFG/BUFGCTRLs	16	32	50%
Number of DSP48E1s	10	864	1%

Fig 8: Thesynthesized logic utilization Design summery of the genetic algorithm (GA) based adaptive filter.

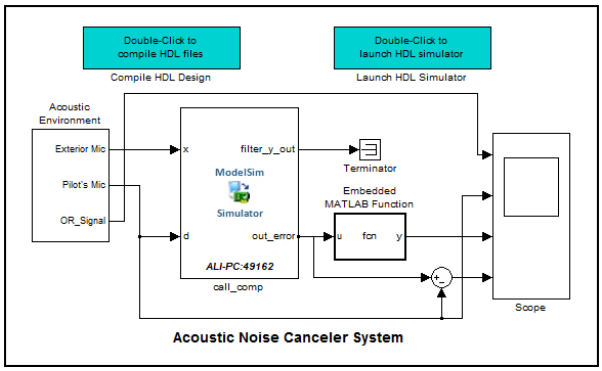

Fig. 9: Cosimulation between MATLAB and ModelSim.

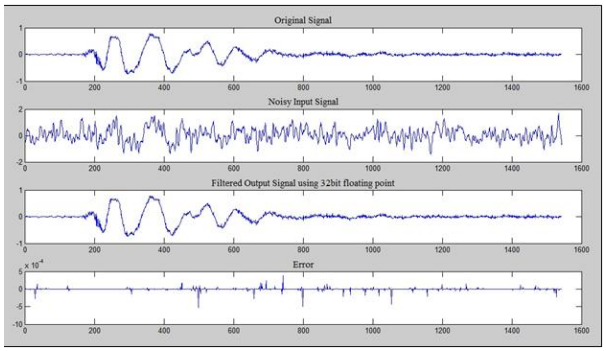

Fig. 10: The GA based adaptive filter simulation result.

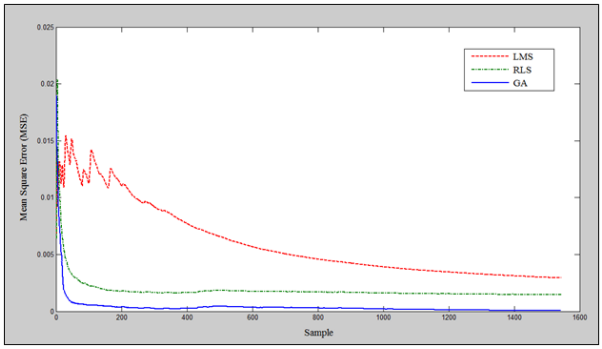

Fig. 11: The MSE of the 32 bit floating point representation GA based Adaptive Filter compared with the MSE of the LMS and RLS adaptive filters.

IV. CONCLUSION

With problems that are not convex, have several local minima, the GA offers the advantages of global optimization in addition the GA does that in a fraction of the number of evaluations required to obtain the solution. Adaptive filters has already implemented on FPGA devices [7], [8], [9], and [10] but, not to GA based adaptive filters, this is in contrast to [1], [11], and [12], where the GA has applied to adaptive filters, but, didn't implemented on FPGA device.In this paper, the GA based adaptive noise cancelation (ANC) system has been implemented on XC6VLX760-VERTIX-6 FPGA device. As seen from the simulation results, the proposed method is able to design adaptive filters andimproved design has been achieved.From figure (10), it has been shown that the error between the original signal and the filtered signal using 32 bit floating point representation varied between 3.8552e-004 and -5.3549e-004 while from figure (11), it has been shown that thesteady state *MSE* of the proposed 32 bit floating point representation GA based adaptive filter implemented on FPGA is 1.1861e-009 compared with a 0.0029 and 0.0015 for the LMS and RLS based adaptive filters respectively.

REFERENCES

[1] Michael S White and Stuart J Flockton. (2000),'Adaptive Recursive Filtering Using Evolutionary Algorithms', Department of Physics, Royal Holloway, University of London,Egham, Surrey TW20 0EX, United Kingdom, 1-17.

[2] B. Farhang. (1998),*Adaptive Filters: Theory and Applications*,England: JOHN WILEY & SONS Ltd.

[3] Ying He*et al*. (2008),'The Applications and Simulation of Adaptive Filter in Noise Canceling',*International Conference on Computer Science and Software Engineering*, IEEE, 1-4.

[4] A. Zaknich. (2005),*Principles of Adaptive Filters and Self-Learning Systems*, London: Springer.

[5] Sabbir U. Ahmad. (2008),'Design of Digital Filters Using Genetic Algorithms', Ph.D. University of Victoria, Dept of Electrical and Computer Engineering.

[6] MITSUO GEN and RUNWEI CHENG. (2000), *Genetic Algorithms and Engineering Optimization'*,Canada: John Wiley & Sons, Inc.

[7] SudhanshuBaghel and RafiahamedShaik. (2011), 'FPGA Implementation of Fast Block LMS Adaptive Filter Using Distributed Arithmetic for High Throughput', IEEE, 443-447.

[8] Wolfgang Fohl and JörnMatthies. (2009), 'A FPGA-BASED ADAPTIVE NOISE CANCELLING SYSTEM', Proc. of the 12th Int. Conference on Digital Audio Effects (DAFx-09), Como, Italy, 1-7.

[9] ZhenbinGao, XiangyeZeng, Jingyi Wang and Jianfei Liu. (2008), 'FPGA implementation of adaptive IIR filters with particle swarm optimization algorithm', ICCS, IEEE, 1364-1367.

[10] Jia Li, Xuezhe Wei and Haifeng Dai. (2009), 'Design and Implementation of RLS Identification Algorithm Based on FPGA', *The Ninth International Conference on Electronic Measurement &Instruments(ICEMI'2009*: IEEE*)*, 599-601.

[11] S. J. Flocktonand M. S. White. (2000), 'THE APPLICATION OF GENETIC ALGORITHMS TO INFINITE IMPULSE RESPONSE ADAPTIVE FILTERS', IEEE, 1-4.

[12] D. M. Ettert, N. U. Hickstt, and K. H. Chot. (1982), 'RECURSIVE ADAPTIVE FILTER DESIGN USING AN ADAPTIVE GENETIC ALGORITIIM', IEEE, 635-638.

[13] Paolo Dadone and Hugh VanLandingham. (2003),'Adaptive Online Parameter Tuning Using Genetic Algorithms'.

[14] T. Arslan and D. H.Horrocks. (1995),'The Design of Analogue and Digital Filters Using Genetic Algorithms',*The Institution of Electrical Engineers*: IEE, 1-5.

[15] K. S. TANG *et al*. (1996),'Genetic Algorithms and Their Applications',*IEEE SIGNAL PROCESSING MAGAZINE*: IEEE, 22-37.

[16] Randy L. Haupt and Sue Ellen Haupt. (2004),*Practical Genetic Algorithms*. 2nd edition. Hoboken, New Jersey: A JOHN WILEY & SONS, INC.

[17] MattiTommiska andJarkkoVuori. (1996),'Implementation of Genetic Algorithms with Programmable Logic Devices', Hardware Implementation of GA,*2NWGA*, Vaasa, Finland, 71-78.