Modified PSO based Adaptive IIR Filter Design for System Identification on FPGA

Lipika Gupta
ECE Department, NITTTR
Chandigarh, India 160019

Rajesh Mehra
Assistant Professor, ECE Department, NITTTR
Chandigarh, India 160019

ABSTRACT
Field programmable gate arrays (FPGAs) are becoming increasingly important implementation platforms for digital circuits. This paper focuses on the implementation of Adaptive Infinite Impulse response (IIR) filter on an FPGA using Modified Particle Swarm Optimization (PSO) Algorithm. The proposed Filter is capable of finding the global optimum solution for system identification problem in less number of iterations. The modified PSO algorithm has been developed and simulated using MATLAB. The result shows the enhanced speed of purposed design in terms of number of iterations it takes to identify the unknown system. The same algorithm has also been realized on various Xilinx FPGA devices and performances have also been analyzed. The area utilization by the proposed design on different FPGA devices has been compared. The results show that proposed filter is consuming very less area in terms of LUTs and Slices to provide enhanced area efficiency.

General Terms
Adaptive IIR Filter, System Identification, Particle Swarm Optimization

Keywords
FPGA, IIR Filter, LUTS, MATLAB, PSO

1. INTRODUCTION
Adaptive filters have become vastly popular in the area of digital signal processing. Adaptive direct modeling or system identification and adaptive inverse modeling or channel equalization find extensive applications in telecommunication, control system, instrumentation, power system engineering and geophysics[1,2].

Due to nonlinearity of the systems, System identification is a challenging and difficult optimization problem. Adaptive Infinite Impulse Response (IIR) systems are used in modeling real world systems because of their reduced number of coefficients and better performance over the Finite Impulse Response (FIR) filters [3,4]. Despite the fact that the digital IIR filter design is a well-researched area, major difficulties still exist in practice. This is because the error surface or the cost function of IIR filters is generally multimodal with respect to the filter coefficients. Thus, gradient-based algorithms can easily be stuck at local minima [5]. This can be overcome by applying an optimization technique such as Particle Swarm Optimization (PSO). Due to its many advantages including its simplicity and easy implementation, the algorithm has been widely applied to optimize the Design of Adaptive Filters [6].

Generally dedicated hardwired implementation of digital filters is required for small area and high-speed digital filtering applications [7]. The recent advancements of Field Programmable Gate Arrays (FPGA)" have enabled them to be used as suitable platforms for realizing adaptive filter algorithms. As the FPGAs can be reconfigured in hardware, they offer complete hardware customization while implementing various DSP applications [8]. In this paper, the adaptive IIR Filter using modified PSO is designed for System Identification. A large variety of FPGAs from different vendors are available in today's market. The purposed filter is then realized on Xilinx-FPGA and a comparative study of hardware utilization of Spartan3E, Virtex-II and Virtex-4 has been carried out. The results show that the purposed filter is more area efficient in terms of LUTs and Slices used by the design.

The paper is organized as follows: Section 2 presents are overview of System Identification problem. Section 3 describes basics of IIR Digital Filter. Here PSO based IIR filter and Conventional PSO algorithm are explained. In Section 4 the IIR Digital filter using Modified PSO algorithm is discussed. Finally, Simulation results and conclusion are presented in Section 5 and Section 6 respectively.

2. SYSTEM IDENTIFICATION PROBLEM
System identification is the mathematical modeling of an unknown system by monitoring its input output data. This is achieved by varying the parameters of the developed model so that for a set of given inputs, its output match that of the system under consideration. By the use of adaptive algorithms, the required parameters can be obtained such that the output of the plant and the model are same for the same set of inputs, which is the goal of system identification [9]. Traditionally, Least Mean Square (LMS) and other algorithms have been studied for the identification of linear and static systems [10]. But, almost all physical systems are nonlinear to certain extent and recursive in nature and hence it is more convincing to model such systems by using nonlinear models. Hence these are better modeled as Infinite Impulse Response (IIR) models as they can provide better performance than a Finite Impulse Response (FIR) filter with the same number of coefficients [11]. Thus the problem of nonlinear system identification can also be viewed as a problem of adaptive IIR filtering.
Also, IIR models are more efficient than the FIR models for implementation as they require less parameter and hence fewer computations for the same level of performance. However, there are few problems associated with the use of IIR models in

identification of a system, such as instability of the algorithms, slow convergence and convergence to the local minimum [12]. In order to overcome these, different techniques have been developed over the years. Fig. 1 shows a block diagram describing the problem of system identification.

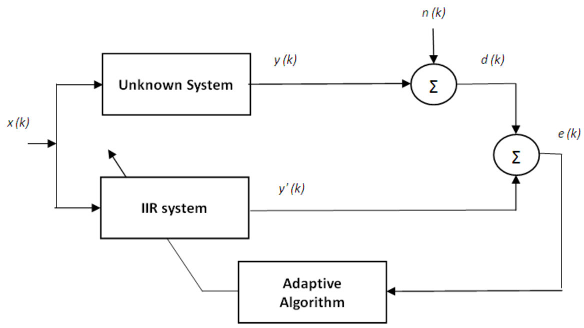


Fig.1 System Identification block diagram

3. IIR DIGITAL FILTER

The structure of a direct-form adaptive IIR filter is shown in Fig-2. In this case, the output of the system is given by

$$y(n) = \sum_{m=1}^{N-1} a_m(n) y(n-m) + \sum_{m=0}^{M-1} b_m(n) x(n-m) \quad (1)$$

The terms $a_m(n)$ and $b_m(n)$ represent the feed forward and feedback coefficients of the filter respectively. In matrix form, $y(n)$ may be written as

$$y(n) = W_n^T S_n \quad (2)$$

where the combined weight vector is

$$W_n = [b_0(n)\ b_1(n) \ldots b_{M-1}(n)\ a_1(n)\ a_2(n) \ldots a_{N-1}(n)]^T \quad (3)$$

And the combined input and output signal vector is

$$S_n = [x(n)\ x(n-1) \ldots x(n-M+1)\ y(n-1)\ y(n-2) \ldots y(n-N+1)]^T \quad (4)$$

The weight update operation of adaptive IIR filter is carried out using either conventional derivative based or derivative free learning algorithms. These systems are useful when the relationship between $d(n)$ and $x(n)$ is not linear such as System Identification.

3.1 PSO based Adaptive IIR Filter:

In System identification, it is necessary to filter one signal y(n) in order to match another signal d(n) as closely as possible as shown in Fig 2. Most nonlinear systems are also recursive in nature. Hence, models for real world systems are better represented as IIR systems. By doing so, the problem of system identification now becomes the problem of adaptive IIR filtering, for which different adaptive algorithms can be applied for adjusting the feed forward and feedback parameters of the recursive system. The adaptive algorithm tries to minimize the error e(n) by adjusting the parameters of the modeled system, which are the pole-zero coefficients in case of an IIR system.

The main complication which IIR filters introduce is that they depend nonlinearly on their coefficients. This can be overcome by applying an optimization technique such as Particle Swarm Optimization (PSO) [13, 14]. Using this technique, several possible collections of IIR coefficients are chosen, and see what error each produces. Based on those results, new points are chosen to test, and continue until all of the points have clustered together and "swarm" in a small area. PSO converges faster than many other optimization algorithms. This makes it an ideal choice for adaptive applications where the optimization will need to be performed often as the signal evolves.

3.2 Conventional PSO Algorithm:

The PSO algorithm is an evolutionary computation technique developed by Kennedy and Eberhart in 1995 [15].

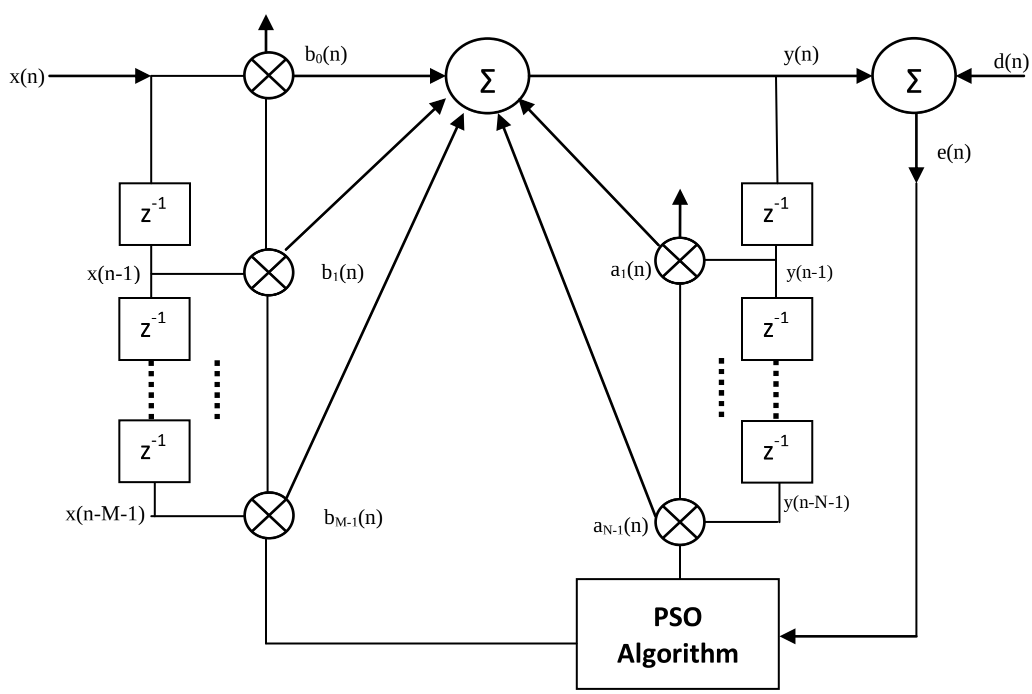


Fig 2: Adaptive IIR Filter Structure

The underlying motivation for the development of the PSO algorithm was the social behavior of animals, such as bird flocking, fish schooling, and the swarm theory.

Particle swarm optimization (PSO) is a high performance optimizer that possesses several highly desirable attributes, including the fact that the basic algorithm is very easy to understand and to implement. It is similar to genetic algorithms and evolutionary algorithms, but requires less computational memory and fewer lines of code. The PSO conducts search using a population of particles which correspond to individuals in GA.

The swarm is typically modeled by particles in multidimensional space that have a position and a velocity. These particles fly through hyperspace and have two essential reasoning capabilities: their memory of their own best position and knowledge of the global or their neighborhood's best. In a minimization optimization problem, "best" simply means the position with the smallest objective value. Members of a swarm communicate good positions to each other and adjust their own position and velocity based on these good positions. So a particle has the following information to make a suitable change in its position and velocity:

i. A global best that is known to all and immediately updated when a new best position is found by any particle in the swarm.
ii. Neighborhood best that the particle obtains by communicating with a subset of the swarm.
iii. The local best, which is the best solution that the particle has seen.

At each time step, each of these particle positions is scored to obtain a fitness value based on how well it solves the problem. Using the local best position (Lbest) and the global best position (Gbest), the particle velocity update equations in the simplest form that govern the PSO are given by

$$v_i(k+1) = w * v_i(k) + c_1 r_1\big(lbest - x_i(k)\big) + c_2 r_2\big(gbest - x_i(k)\big) \quad (5)$$

where *w, c1* and *c2 are* called the coefficient of inertia, cognitive and society, respectively. The r1 and r2 are uniformly distributed random numbers in [0, 1]. The term *vi* is limited to the range ± *vmax*. If the velocity violates this limit, it will be set at its proper limit.

Changing velocity enables every particle to search around its individual best position and global best position. Based on the updated velocities, each particle changes its position according to the following:

$$x_i(k+1) = x_i(k) + v_i(k+1) \quad (6)$$

The design steps of the PSO-based optimization process can be summarized as follows:

Step 1: Generate an initial random population.
Step 2: If the pre-specified iterations is achieved, stop.
Step 3: Calculate the fitness value of each particle.
Step 4: For each particle, find the individual best.
Step 5: Find the global best.
Step 6: Perform the velocity updating for each particle.
Step 7: Perform the position updating for each particle.
Step 8: Go back to Step 2.

When every particle is updated, the fitness value of each particle is calculated again. If the fitness value of the new particle is higher than those of local best, then the local best will be replaced with the new particle. If the fitness value of the new particle is higher than those of global best, then the global best will be also replaced with the new particle. The algorithm repeats the above updating process step by step; the whole population evolves toward the optimum solution. Fitness function is given by following equation

$$fitness = \sum_{n=1}^{N} [d(n) - y(n)]^2 \quad (7)$$

The flow chart of the conventional PSO is shown in Fig-3.

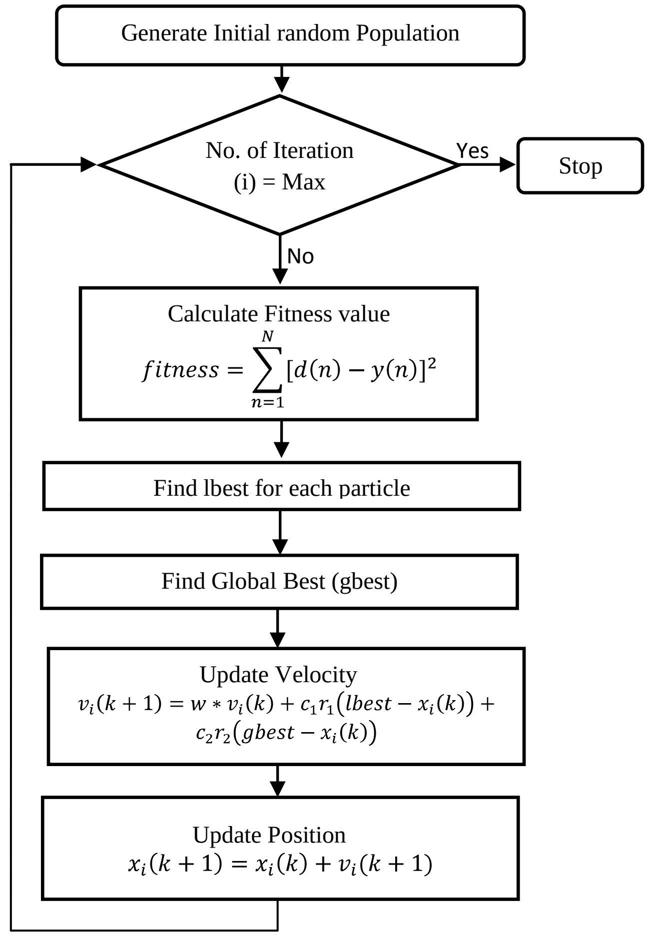


Fig 3: Conventional PSO flowchart

4. MODIFIED PSO BASED IIR FILTER

PSO equation consists of three parameters. The c1 and c2 are called cognitive and social acceleration constants and help to guide the particles towards the gbest. These constant are equal and have the values from 0 to 2 but studies have shown their values set to 2 gives the best results. For the purposed filter design c1 and c2 are equal to 2.

Another parameter of PSO is w called the inertia weight. For unconstrained PSO, w is linearly decreasing from w_{max}=0.9 to w_{min}=0.4 over iterations. For the purposed design, w is calculated using equation 8 for different iterations (i).

$$w = w_{max} - \frac{(w_{max} - w_{min})i}{Max.\ no.\ of\ iterations} \quad (8)$$

Further the inertia weight (w) control the influence of the current velocity on the new velocity. A large inertia weight compels large exploration through the search space; a smaller inertia weight causes reduced exploration. The use of equation 8 to update w leads to sufficient exploration of search space, thus finding out the global optimum solution.
In addition to this maximum velocity is also limited using Signum function. As velocity update leads to acceleration of particles. The smaller the acceleration, the smoother the trajectory of the particle is. However, too small an acceleration may lead to slow convergence, whereas too large an acceleration drives the particles towards infinity.
The flow chart of the Modified PSO is shown in Fig-4.

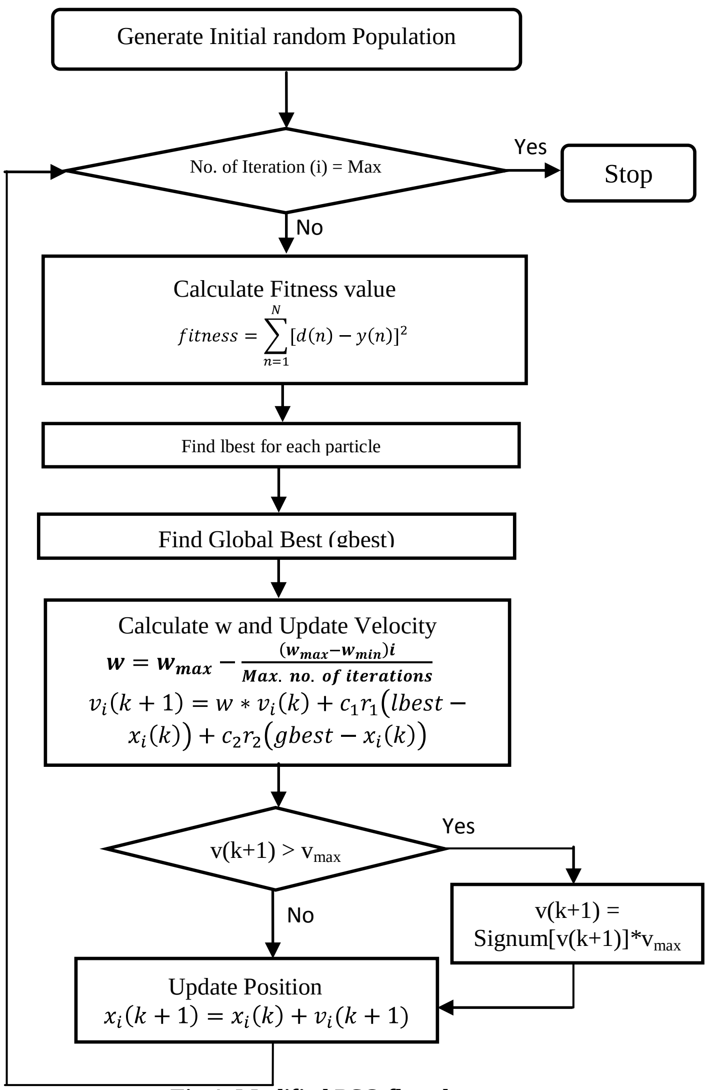


Fig 4: Modified PSO flowchart

5. SIMULATION RESULTS & DISCUSSIONS

To test the performance of Modified PSO algorithm, the coefficients of IIR filter are fixed and used as the unknown system which will be required to be identified. The fixed filter coefficients are considered as global optimal solution.

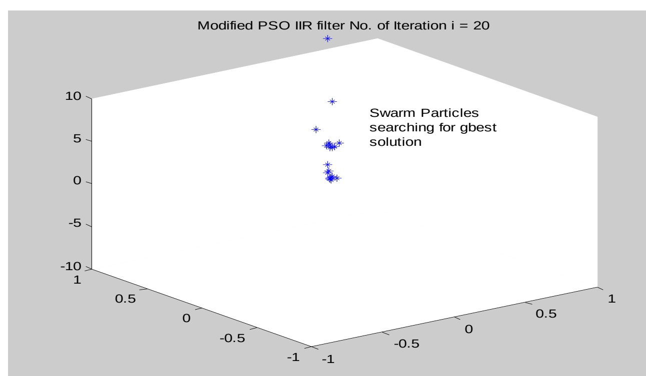
5.1 Modified PSO based System Identification:

The comparison results of the simulation are shown in Table 1, where 3-dimension hyperspace was chosen. The coefficients of fixed filter which is to be identified were set at constant values: a0= 0.4681, b1= 0.0532, b2= 0.1954 and the number of particles=30. Fig-5 show the simulation results for iteration number i=20, i=50 and i=60 respectively.

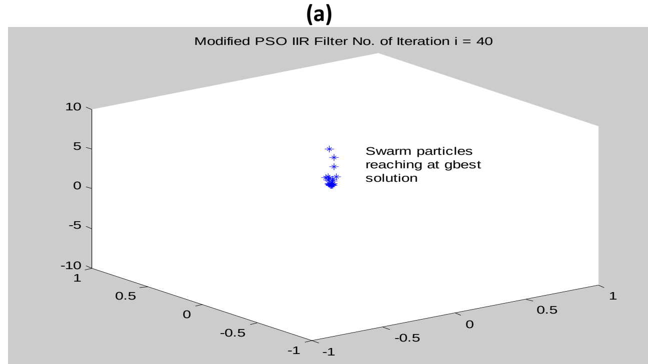
Table 1. Performance Analysis of Modified PSO

S. No.	**Comparing results with [1]**	**Proposed Modified PSO IIR filter**
No. of Coefficients	3	3
No. of Iterations	101	60

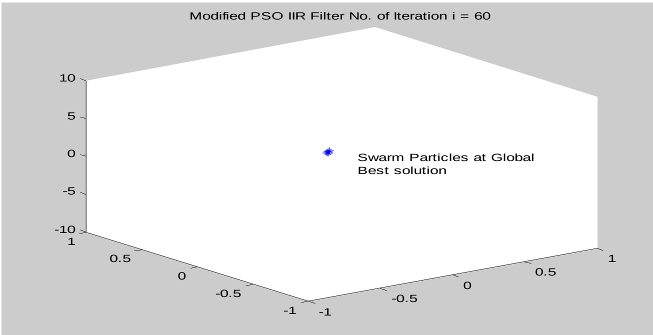
Fig-5(a) shows Swarm position at 20th iteration. Figure shows that particles are dispersed in the 3-D space searching for the global best solution. Position and velocity of each particle is changed after every iteration. The velocity and position of particle reached to gbest solution won't change again.
Fig- 5(b) shows swarm position at 40th iteration. Figure shows that maximum no. of particles has reached at global optimum value.
We can also see that the coefficients of adaptive filter achieved: a0= 0.4681, b1= 0.0532, b2= 0.1954 at i=60 as shown in Fig-5(c). At this point the all swarm particles are at gbest solution. The proposed design is able to converge in 60 numbers of iterations as compared to 101 taken by design [1].



(a)



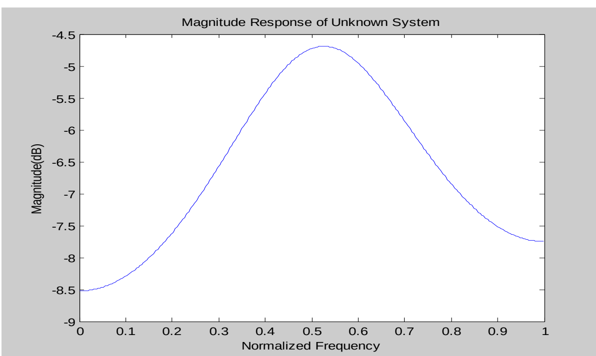
(b)



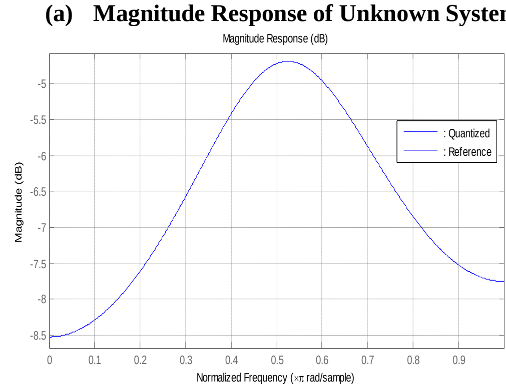
(c)

Fig 5: Modified PSO Simulations

The magnitude responses of the Unknown System (fixed IIR filter) and Adaptive IIR filter which is using modified PSO is shown inFig-6(a) and Fig-6(b) respectively. The response of two filters is identical. Therefore the Modified PSO based Adaptive IIR filter is able to identify the unknown System in lower number of iterations. Hence the algorithm has enhanced the speed.



(a) Magnitude Response of Unknown System



(b) Magnitude Response of Modified PSO

Fig 6: Magnitude responses

B. Performance Analysis on different FPGAs:

The purposed filter design is further converted to VHDL and implemented on different FPGAs using Xilinx ISE-10.1. The Behavioral simulation for the purposed filter is shown in Fig-7. A test bench for the purposed filter is generated and simulation time limit 1000ns is selected. The simulated results show that the filter output in available after a delay of 185ns.

A comparative study on the hardware utilization of different platforms like, Spartan3e, Virtex-II, and Virtex-4 for the algorithm has been carried out. The analyses have been performed based on the utilization of number 4 input LUTs and number of occupied slices. Table II represents the logic utilization (%) for IIR Filter from [3] and the purposed modified PSO based Adaptive IIR Filter. From the analysis it can be shown that the purposed filter is more area efficient i.e. a lower % of 4 input LUTs and Slices are used.

Table 2: Logic Utilization Comparison

Device Logic	**Logic utilization factor for IIR filter on FPGAs [3]**			**Logic utilization factor for Proposed Modified PSO based IIR filter on FPGAs**		
	Spartan 3e	Virtex-II	Virtex-4	Spartan 3e	Virtex-II	Virtex-4
% of 4-input LUTS	1.00%	2.00%	1.00%	1.00%	2.00%	1.00%
% of Occupied Slices	5.00%	8.00%	3.00%	2.00%	3.00%	1.00%

6. CONCLUSION

In this paper a modified PSO based Adaptive IIR filter has been presented for System identification. For adaptive IIR filtering applications in system identification, the primary performance considerations for an optimization algorithm are the rate of convergence and the minimum mean squared error. These parameters have been optimized by implementing the proposed designing the IIR filter using Modified PSO algorithm. The simulated results have shown that the modified particle swarm algorithm has enhanced the convergence speed by 49%. Further performance analysis of the purposed filter on different FPGAs show that it consumes considerably less LUTs and Slices of target FPGA. Therefore the proposed design can provide high speed and area efficient solution for adaptive signal processing applications.

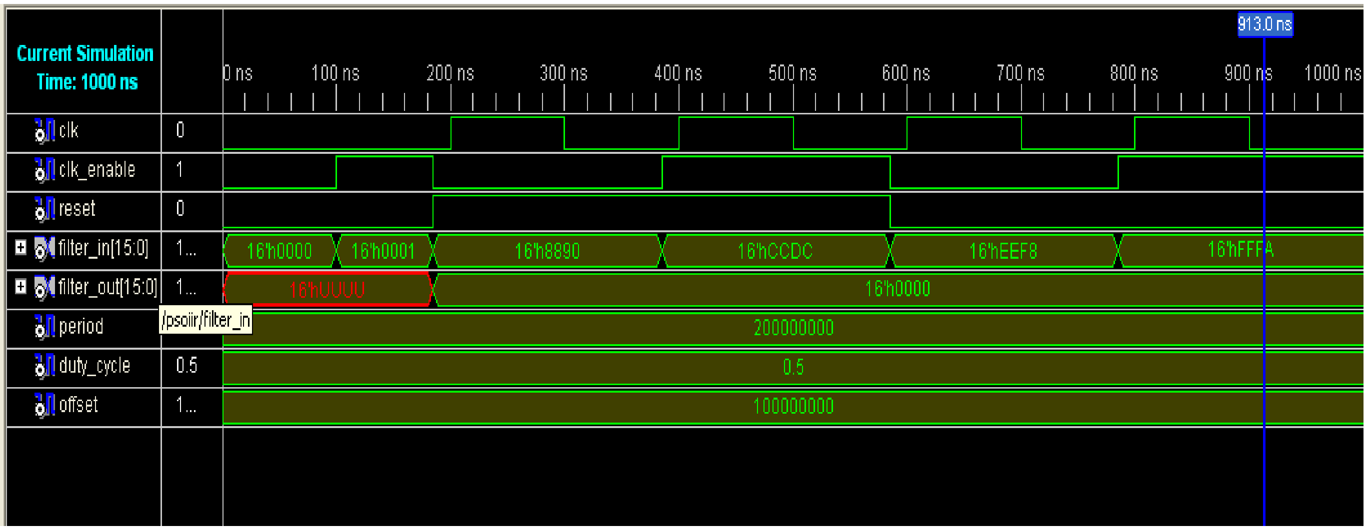


Fig 7: Simulation Results

7. REFERENCES

[1] Zhenbin Gao; Xiangye Zeng; Jingyi Wang; Jianfei Liu, "FPGA implementation of adaptive IIR filters with particle swarm optimization algorithm" ICCS-2008; 11th IEEE International Conference on Communications Systems held on 19 to 21 Nov 2008 in China; Page(s): 1364-1367, 2008.

[2] Yamille del Valle, Ganesh Kumar Venayagamoorthy, Salman Mohagheghi, Jean-Carlos Hernandez, and Ronald G. Harley, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems" IEEE Transactions on Evolutionary Computation, Vol. 12, No. 2, April 2008, Page(s): 171-195, 2008

[3] Sharbari Banerjee, Amitabha Sinha, "Performance Analysis of Different DSP Algorithms on Advanced Microcontroller and FPGA" ACTEA 2009; IEEE International conference held on July 15-17, 2009 Zouk Mosbeh, Lebanon; Page(s): 609-613, 2009

[4] Faten BEN ARFIA, Mohamed Ben Messaoud, Mohamed Abid "Nonlinear adaptive filters based on PSO" Leonardo Journal of Sciences, Issue 14, Jan-Jun 2009.

[5] D.J. Krusienski, W.K. Jenkins "PSO for Adaptive IIR Filter" IEEE Transactions on Signal processing 2004.

[6] Yunhong Gao, Yibo Li, Hong Qian, "The Design of IIR Digital Filter based on Chaos Particle Swarm Optimization Algorithm" Second International conference on Genetic and Evolutionary Computing Proceedings IEEE-2008; Page(s): 303-306, 2008

[7] R. Yousefi, A. Ahmadi, S. M. Fakhraie, "Design Flow for Hardware Implementation of Digital Filters", 2008 Internatioal Symposium on Telecommunications, 2008 IEEE; Page(s)-586-591, 2008

[8] R. Mustafa, M. A. Mohd Ali, C. Umat, D.A. Al-Asady "Design and Implementation of Least Mean Square Adaptive Filter on Altera Cyclone II Field Programmable Gate Array for Active Noise Control" 2009 IEEE Symposium on Industrial Electronics and Applications (ISIEA 2009), October 4-6, 2009, Kuala Lumpur, Malaysia, Page(s): 479-484, 2009

[9] Sergio L. Netto, Paulo S.R. Diniz, Panajotis Agathoklis, "Adaptive IIR filtering Algorithms for System Identification: A General Framework" IEEE Transactions on Education, Vol. 38 No.1, Feb-1995; Page(s): 54-66

[10] Xiaochun Guan, Xiaojing Chen, Guichu Wu, "QX-LMS Adaptive FIR Filters For System Identification" IEEE-2009 International Conference Proceedings

[11] Sheng Chen, Bing L.Luk, "Digital IIR Filter design using Particle Swarm Optimization", International Journal of Modeling, Identification and Control, Vol-9, No.4, 2010; Page(s): 327-335, 2010

[12] Xia Yu, Jianchang Liu, Hongru Li, "An Adaptive Inertia Weight Particle Swarm Optimization Algorithm for IIR Digital Filter" 2009 International Conference on Artificial Intelligence and Computational Intelligence proceedings IEEE-2009; Page(s): 114-118, 2009

[13] Bipul Luitel, Ganesh Kumar Venayagamoorthy, "Particle Swarm Optimization with Quantum Infusion for the Design of Digital Filters" 2008 IEEE Swarm Intelligence Symposium St. Louis MO USA, September 21-23, Page(s): 1031-1035, 2008

[14] Bipul Luitel; Ganesh K.Venayagamoorthy, "Differential Evolution Particle Swarm Optimization for Digital Filter Design", 2008 IEEE Congress on Evolutionary Computation (CEC 2008), Page(s): 3954-396, 2008

[15] J. Kennedy, R.C. Eberhart "Particle Swarm Optimization", Proceedings of the IEEE International Conference on Neural Networks, 1995, pp. 1942-1948.

8. AUTHORS PROFILE

Lipika Gupta: Ms. Lipika Gupta received B.Tech Degree in Electronics & Communication engineering from Punjab Technical University, Jalandhar, India in 2002. She is currently pursuing towards the M.E. degree from NITTTR, Chandigarh, India. She has 8 years of academic experience. Ms. Gupta's interest areas are Signal Processing, Wireless Communication and VLSI Design.

Rajesh Mehra: Mr. Rajesh Mehra is currently Assistant Professor at National Institute of Technical Teachers' Training & Research, Chandigarh, India. He is pursuing his PhD from Panjab University, Chandigarh, India. He has completed his M.E. from NITTTR, Chandigarh, India and B.Tech. from NIT, Jalandhar, India. Mr. Mehra has 15 years of academic experience. He has authored more than 15 research papers in reputed International Journals and more than 30 research papers in National and International conferences. Mr. Mehra's interest areas are VLSI Design, Embedded System Design, Advanced Digital Signal Processing, Wireless & Mobile Communication and Digital System Design. Mr. Mehra is life member of ISTE.