Hindawi Publishing Corporation
VLSI Design
Volume 2012, Article ID 505983, 13 pages
doi:10.1155/2012/505983

Research Article

Modeling and Design of a Nano Scale CMOS Inverter for Symmetric Switching Characteristics

Joyjit Mukhopadhyay[1] and Soumya Pandit[2]

[1] *A. K. Choudhury School of Information Technology, University of Calcutta, Kolkata 700009, India*
[2] *Institute of Radio Physics and Electronics, University of Calcutta, Kolkata 700009, India*

Correspondence should be addressed to Soumya Pandit, soumya_pandit@ieee.org

Received 12 October 2011; Revised 11 January 2012; Accepted 26 January 2012

Academic Editor: Chien-In Henry Chen

This paper presents a technique for the modeling and design of a nano scale CMOS inverter circuit using artificial neural network and particle swarm optimization algorithm such that the switching characteristics of the circuit is symmetric, that is, has nearly equal rise and fall time and equal output high-to-low and low-to-high propagation delay. The channel width of the transistors and the load capacitor value are taken as design parameters. The designed circuit has been implemented at the transistor-level and simulated using TSPICE for 45 nm process technology. The PSO-generated results have been compared with SPICE results. A very good accuracy has been achieved. In addition, the advantage of the present approach over an existing approach for the same purpose has been demonstrated through simulation results.

1. Introduction

For digital integrated circuit design, CMOS inverter design is considered to be a fundamental procedure [1]. This is because of the fact that the procedure for designing other complex digital integrated circuits is primarily based on the design procedure of CMOS inverter [1]. In the nano scale regime, the task of manual design of an optimal integrated circuit is very difficult. The fundamental reason for this is that in the sub-90 nm domain, the short channel effects of MOS transistors play critical roles. The performance behaviours of a circuit in this regime depend on the transistor channel length and width through complex high orders of equations. Therefore, for correct design and simulation of nano scale digital integrated circuits, accurate models need to be constructed [2]. The construction of accurate performance model for CMOS inverter valid in the sub-90 nm domain is, therefore, an important problem to be solved. In addition, the design of an inverter circuit for optimized performance is considered as a nonlinear optimization problem.

A number of approaches are available in the literature for modeling the switching characteristics of a CMOS inverter. Sakurai and Newton derived a simple closed-form delay expression for series-connected MOSFET circuits in [3], based on the α-power law MOSFET current model. In [4], an improved analytical propagation delay model has been presented. But it neglects the channel length modulation effect, which is important for modern deep submicron technologies. An analytical charge-based compact delay model for submicrometer CMOS inverters has been derived in [5]. An accurate analytical propagation delay model of nano-CMOS circuits has been derived in [6] based on modified α-power law current model. A look-up table approach for efficient delay characterization of nano scale VLSI logic circuits is presented in [7]. An approach for optimization of digital integrated circuits using geometric programming technique is described in [8]. A particle swarm optimization-based approach for inverter design considering transient performance is described in [9, 10]. Artificial neural network (ANN) has been used in [11] for deciding the MOSFET channel length and width for analog integrated circuits. A technique for technology independent neural network modeling for fundamental blocks of analog circuits has been discussed in [12]. ANN has been used in [13] for modeling and design of CMOS operational amplifier circuit.

This paper presents an approach for the design of a nano scale CMOS inverter circuit with symmetric switching characteristics. The switching characteristics of the inverter circuit has been modeled using artificial neural network (ANN). The input design parameters of the ANN model are the widths of the PMOS and NMOS transistor, the load capacitor and the rise time of the input signal. The output performance parameters are the inverter switching point, the output rise time and fall time, the low-to-high and high-to-low output propagation delay times. The constructed ANN model is embedded within a particle swarm optimization (PSO) algorithm. This determines the channel widths of the transistors and the output load capacitor value such that the difference between the output rise time and fall time is minimized and the difference between the output high-to-low propagation delay and low-to-high propagation delay is also minimized. The PSO synthesized designs have been validated by implementing the designs at transistor-level using 45 nm CMOS technology and comparing the PSO predicted results with actual TSPICE simulated results. The advantage of our approach of designing the inverter circuit over an existing approach is also demonstrated through simulation results. This is for the first time to the best of the author's knowledge that ANN-PSO combined approach has been used for modeling and optimal design of a nano scale CMOS digital circuit.

The rest of the paper is organized as follows. The problem is formulated in Section 2. The methodology is presented in details Section 3. Numerical results and the corresponding discussion are provided in Section 4, and finally conclusion is drawn in Section 5.

2. Problem Formulation

The basic circuit diagram of a CMOS inverter is shown in Figure 1. Let $\overline{X}_d$ be the vector containing the design parameters, that is, the channel width W_n of the NMOS transistor, the channel width W_p of the PMOS transistor, output load capacitor C_L and $\overline{X}_i$ be the vector containing the input signal parameters, that is, the rise/fall time of the input pulse signal, τ_{in}. A circuit design optimization problem consists of determining the values of the design parameters for given input signal parameter values, such that one or more performance parameters are optimized and certain imposed constraints are satisfied. The complete problem can be considered to be divided into two sub-problems. In the first part, the relationships between the design parameters and the performance parameters are constructed. In the second part, the constructed relationships are used to compute cost functions for the subsequent optimization problem.

Let $\overline{\rho}$ be the vector containing the following performance parameters of the design, (i) output rise time (τ_R), (ii) fall time (τ_F), (iii) inverter switching point (V_{SP}), (iv) output low-to-high propagation delay time (τ_{PLH}), and (v) output high-to-low propagation delay time (τ_{PHL}). Thus the inputs and outputs of the performance model are as follows:

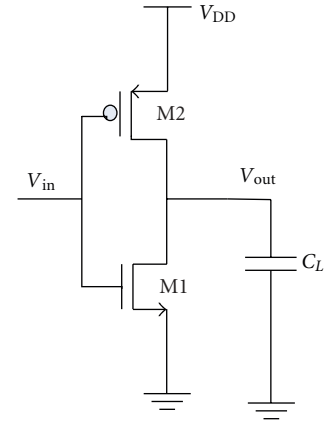


Figure 1: CMOS inverter.

$$\begin{aligned} \overline{X}_d &= \left[W_n, W_p, C_L\right], \\ X_i &= \tau_{\text{in}}, \\ \overline{\rho} &= [\tau_R, \tau_F, V_{\text{SP}}, \tau_{\text{PLH}}, \tau_{\text{PHL}}]. \end{aligned} \tag{1}$$

The performance model of the inverter circuit is defined as

$$\overline{\rho} = f\left(\overline{X}_d, X_i\right). \tag{2}$$

This relationship between the circuit design parameters and the performance parameter is generally strongly nonlinear and multidimensional. In the present work this is approximated through an ANN model as follows:

$$\overline{\hat{\rho}} = f_{\text{ANN}}\left(\overline{X}_d, X_i, w\right), \tag{3}$$

where f_{ANN} is a neural network, $\overline{\hat{\rho}}$ is a q dimensional output vector of neural model responses, $\overline{X}$ is the ANN input vector, and w contains all the weight parameters required to construct the ANN structure. The first part of the work attempts to construct f_{ANN} such that it is a faithful approximation of the original function f.

The second part is an optimization problem which is stated as follows:

$$\begin{aligned} &\text{Minimize} \quad \overline{\Phi}\left(\overline{\hat{\rho}}\right), \\ &\text{subject to} \quad \overline{g}\left(\overline{\hat{\rho}}\right) \le 0, \\ &\text{such that} \quad \left(\overline{X}_d\right)_{\min} \le \overline{X}_d \le \left(\overline{X}_d\right)_{\max}. \end{aligned} \tag{4}$$

In (4), $\overline{\Phi}(\overline{\hat{\rho}})$ is referred to as the cost function, computed on the ANN predicted performance parameters, $\overline{g}(\overline{\hat{\rho}})$ refers to the constraints. It may be noted that circuit optimization problems involve more than one objective.

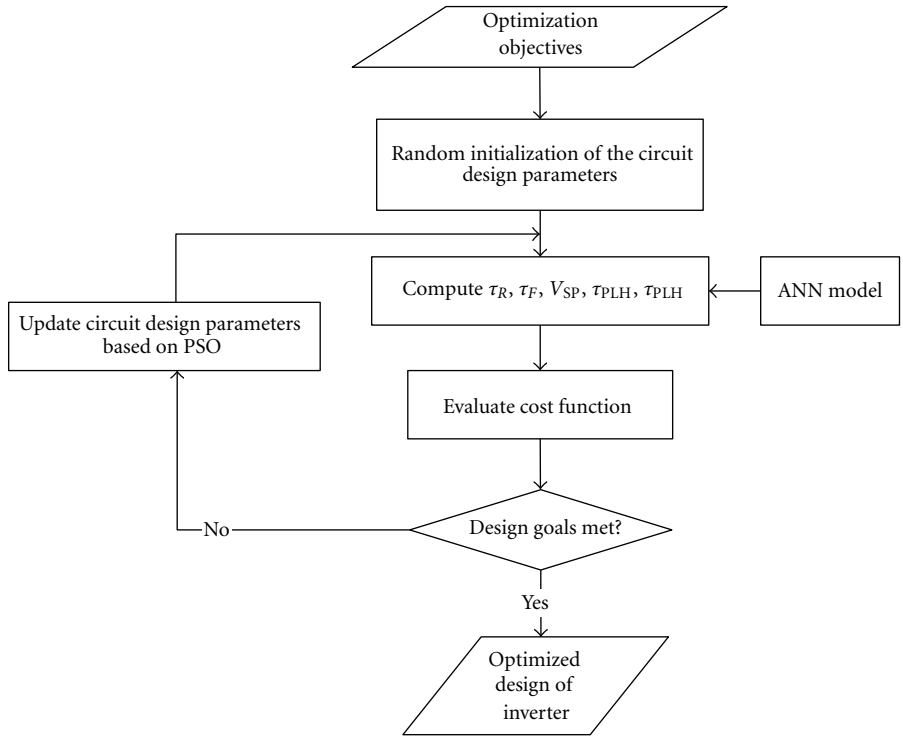


Figure 2: Overview of the design methodology.

3. Description of the Design Methodology

3.1. An Outline. An outline of the design methodology is illustrated in Figure 2. The design procedure starts with a set of optimization objectives. The design parameters are initialized randomly by the PSO algorithm. For these design parameters, the performance parameters $\tau_R, \tau_F, V_{SP}, \tau_{PLH}$, and τ_{PHL} are calculated using the ANN model for a given value of τ_{in}. Cost functions are computed based on these performance parameter values and the input optimization objectives. The design parameter values are then updated through the PSO algorithm according to the minimum cost. This process continues until a desired cost function objective is achieved or the maximum number of iterations is executed. The final output is a set of synthesized values of the design parameters.

3.2. Construction of ANN Model. In this subsection, we will first provide the basic concept of multilayer perceptron ANN followed by a detailed description of the ANN model development procedure.

3.2.1. MLP Structure. Multilayer perceptron (MLP) is a widely used ANN structure [14]. In this network, the neurons are grouped into layers. The first and the last layers are called the input and the output layers, respectively, and the remaining layers are called the hidden layers. The structure of a typical MLP network is shown in Figure 3. The layers in this network are interconnected by network links that are associated with weights that determine the effect on the information passing through them. In an MLP, each neuron processes the input received from other neurons through an activation function in the neuron and the processed information becomes the output of the neuron. The hidden neuron activation function that is used in the present work is tangent-sigmoid function. On the other hand, linear activation is used for the output neurons.

Multilayered feed-forward neural network is trained by supervised learning using iterative back propagation algorithm. The basic flow chart illustrating the ANN-MLP training procedure is shown in Figure 4. The procedure starts with generating a set of sample data. This is divided into two sets: training set and test set. The training set is used for training purpose and the test set for testing purpose. The MLP parameters are initialized randomly. Then the weight parameters are adjusted utilizing the training data set until the training goal is met.

3.2.2. Data Generation. In order to generate training and test data, CMOS inverters are constructed corresponding to the circuit design parameters and input signal parameter listed in Table 1. The channel length of both the transistors is fixed at minimum of the process technology, that is, 45 nm. The other process technology parameters are taken from Berkeley Predictive Technology model file [15]. Based on Halton sequence generator [16], uniformly distributed samples are generated within the specified range. The training and test data corresponding to those sample points are generated through T-SPICE simulation using BSIM4 model. Transient analysis and DC transfer sweep analysis are performed in order to extract the performance parameters.

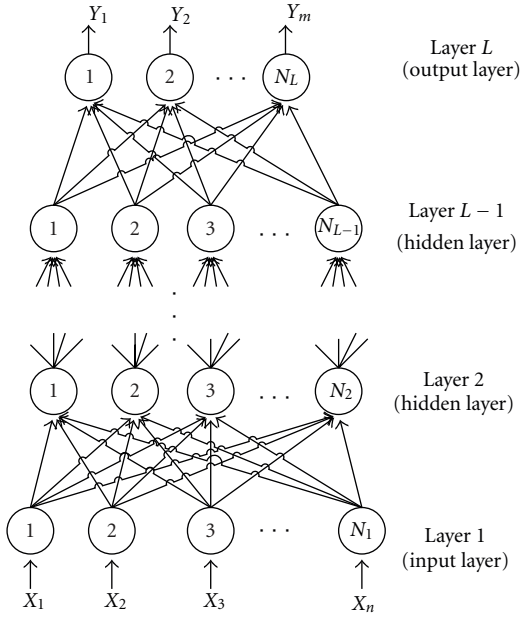


Figure 3: MLP-ANN structure.

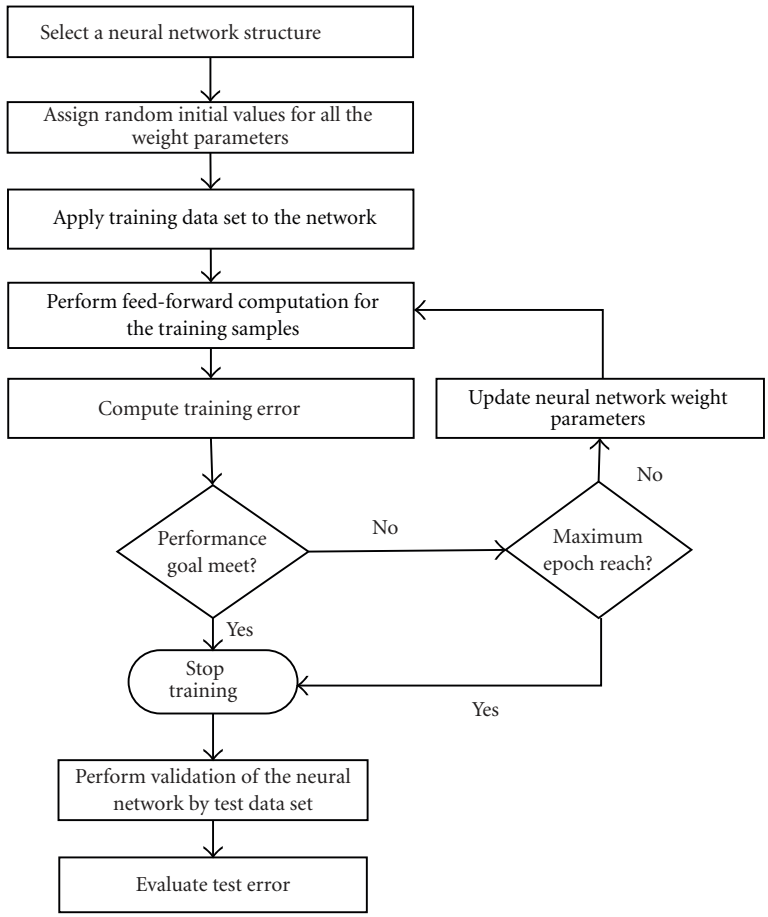


Figure 4: Illustration of ANN Training Procedure.

Table 1: Range of circuit design parameters.

Parameters	Min	Max
W_n (nm)	45	1000
W_p (nm)	45	1000
C_L (pF)	0.2	5
τ_{in}	500 ps	10 ns

3.2.3. Data Scaling. It is observed from Table 1 that the input parameters vary over a wide range. Consequently the output performance parameters will also vary over a wide range. Therefore, a systematic preprocessing of training data, referred to as data scaling is required for efficient construction of the ANN model. In this work, we have used linear scaling of the data between 0 and 1, described by the following formula:

$$\tilde{x} = \tilde{x}_{\min} + \frac{x - x_{\min}}{x_{\max} - x_{\min}}(\tilde{x}_{\max} - \tilde{x}_{\min}), \tag{5}$$

and the corresponding de-scaling formula is given by

$$x = x_{\min} + \frac{\tilde{x} - \tilde{x}_{\min}}{\tilde{x}_{\max} - \tilde{x}_{\min}}(x_{\max} - x_{\min}), \tag{6}$$

where x, $x_{\min}$, $x_{\max}$ represent the original data and $\tilde{x}$, $\tilde{x}_{\min}$, $\tilde{x}_{\max}$ represent the scaled data.

3.2.4. Data Organization. The generated data is divided into two sets, namely, training data set and test data set. The training data is used to guide the training procedure. A portion of the training data set is used for validating the training procedure. The test data is used to independently examine the final quality of the trained neural model in terms of accuracy and generalization capability.

3.2.5. Neural Network Training. A standard 4-layer feedforward MLP architecture has been considered in order to construct the ANN model of the inverter. During the training procedure, the weight parameters and the bias values are adjusted in order to minimize the training error. For this purpose, we have used Levenberg-Marquardt (LM) back propagation method as the training algorithm. The training goal is set to 10^{-7}. The training algorithm of Matlab toolbox has been used.

3.2.6. ANN Model Accuracy. In order to verify the accuracy of the constructed ANN model, statistical measures such as average relative error and correlation coefficient between the neural outputs and actual-SPICE generated values are calculated for each output parameter. These are defined as follows:

$$E = \frac{1}{n\rho}\sum_{1}^{n}(\rho - \hat{\rho}),$$
$$R = \frac{n\sum\rho\hat{\rho} - \sum\rho\sum\hat{\rho}}{\sqrt{\left[n\sum\rho^2 - \left(\sum\rho\right)^2\right] - \left[n\sum\hat{\rho}^2 - \left(\sum\hat{\rho}\right)^2\right]}}. \tag{7}$$

Here, n, $\hat{\rho}$, and ρ are the number of samples in the data set, ANN model output and corresponding SPICE-simulated value, respectively. The correlation coefficient is a measure of how closely the ANN outputs fit with the target values. It is a number between 0 and 1. If there is no linear relationship between the estimated values and the actual targets, then the correlation coefficient is 0. If the number is equal to 1.0, then there is a perfect fit between the targets and the outputs. Thus, higher the correlation coefficient, the better it is.

3.3. Optimization Using PSO Algorithm. In this subsection, we will first provide the basic concept of PSO algorithm, followed by the PSO-based design procedure.

3.3.1. PSO Algorithm. Particle swarm optimization algorithm is a robust stochastic evolutionary computation technique inspired by social behavior, movement, and intelligence of swarms [17]. PSO works on a population of solution candidates referred to as particles. The size of the swarm is the total number of particles. At any particular instance, each particle has a position and a velocity. The movement of the particles are controlled by updating the position and velocity vectors in an effort to find an optimum solution. In a complete swarm, containing N number of particles, each particle is initialized with a random position value. The position vector of the ith particle with feature number D, is denoted as

$$X_i = [x_{i1}, x_{i2}, \ldots, x_{iD}], \tag{8}$$

and the velocity vector is defined as

$$V_i = [v_{i1}, v_{i2}, \ldots, v_{iD}]. \tag{9}$$

For each iteration, the velocity and position vector of the ith particle in the search space are updated as follows

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 \times \text{rand}_1^k\left(pbest_{id}^k - x_{id}^k\right) + c_2 \times \text{rand}_1^k\left(gbest_{id}^k - x_{id}^k\right), \tag{10}$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1}, \tag{11}$$

where $d = 1, 2, \ldots, D$ and $i = 1, 2, \ldots, N$. D is the number of design parameters and k is the iteration count. The factors c_1 and c_2, referred to as the acceleration factors, indicate the relative attraction towards *pbest* and *gbest*, respectively. The particles keep track of two best values. The first one is the best fitness value obtained so far by the particle, the corresponding position being termed as personal best *pbest*. The global best value *gbest* is the location of the best fitness value achieved so far considering all the particles in the swarm. In (10), rand_1 and rand_2 are random numbers uniformly distributed between zero and unity. The inertia weight factor w controls the tradeoff between global and local search capabilities of the swarm. The value of the velocity v of a particle is clamped to the range $[-v_{\max}, v_{\max}]$ to reduce the likelihood that a particle might leave the search space.

Table 2: Architecture of the 1st ANN.

Parameters	Optimized Values
Architecture	Feed forward MLP
Training Algorithm	Back propagation
Hidden layer	2
No. of neurons in the first hidden layer	16
No. of neurons in the second hidden layer	20
Hidden layer transfer function	Tan-sigmoid
Output layer transfer function	Linear
Maximum epoch	1000

Table 3: Accuracy of the 1st ANN model.

Error	Output	Training	Test
E	τ_R	1.21	1.36
	τ_F	1.82	1.95
	τ_{PHL}	1.03	0.86
	τ_{PLH}	0.65	0.61
R	τ_R	0.9993	0.9995
	τ_F	0.9993	0.9995
	τ_{PHL}	0.9993	0.9995
	τ_{PLH}	0.9993	0.9995

The search space is defined by the bounds $[x_{\min}, x_{\max}]$. In this work, the value of $v_{\max}$ is set to $v_{\max} = k \times x_{\max}$ where $0.1 \le k \le 1$. The flow chart of the PSO algorithm as used in this work is shown in Figure 5.

3.3.2. PSO-Based Design. In the present work our aim is to design an inverter such that the output switching characteristics of the circuit is symmetric. This means that (i) the difference between the output rise time (τ_R) and fall time (τ_F) and (ii) the difference between the output propagation delay times τ_{PHL} and τ_{PLH} should be minimum.

The transistor channel widths W_n, W_p, and the load capacitor C_L are the design parameters. The load capacitor as used here represents lumped capacitance consisting of two broad types of capacitances: (i) intrinsic and (ii) extrinsic. The intrinsic capacitance is composed of diffusion and overlap capacitances. On the other hand, the extrinsic capacitance is contributed by interconnect and fan-out gate. The intrinsic components depend upon the transistor dimensions. However, the extrinsic components are usually independent of transistor dimensions of the inverter under consideration. Using efficient layout techniques, it is possible to make the drain diffusion areas as small as possible. Thus the standard design practice is to assume that the external load capacitor mainly consists of extrinsic components such that the problem of inverter design can be done conveniently using inverter delay models constructed earlier.

The values of these design parameters should be obtained such that symmetry of the output switching characteristics can be achieved. The design problem can thus be summarized as follows. The value of the rise/fall time of the input signal will be taken from the user:

$$\begin{aligned}
\text{Minimize} \quad & \frac{|\tau_F - \tau_R|}{\tau_F} + \frac{|\tau_{\text{PHL}} - \tau_{\text{PLH}}|}{\tau_{\text{PHL}}}, \\
\text{subject to} \quad & (\tau_F)_{\min} \le \tau_F \le (\tau_F)_{\max} \\
& (\tau_R)_{\min} \le \tau_R \le (\tau_R)_{\max} \\
& (\tau_{\text{PHL}})_{\min} \le \tau_{\text{PHL}} \le (\tau_{\text{PHL}})_{\max} \\
& (\tau_{\text{PLH}})_{\min} \le \tau_{\text{PLH}} \le (\tau_{\text{PLH}})_{\max} \\
& 0.45 \times V_{\text{DD}} \le V_{\text{SP}} \le 0.55 \times V_{\text{DD}}, \\
\text{where} \quad & (W_n)_{\min} \le W_n \le (W_n)_{\max} \\
& \left(W_p\right)_{\min} \le W_p \le \left(W_p\right)_{\max} \\
& (C_L)_{\min} \le C_L \le (C_L)_{\max}.
\end{aligned} \tag{12}$$

Thus the PSO algorithm would result in exact values of the design parameters which minimize the cost function value and satisfy the specified constraints. The supply voltage V_{DD} is taken to be 1.0 V. The swarm size is taken to be 30. It has been reported in the literature [10] that the choice of PSO parameters $c_1 = c_2 = 1.49618$ and $w = 0.7298$ ensure good convergence of the PSO algorithm. The maximum number of iterations that has been considered in this work is 1000.

It may be noted that symmetric static and switching characteristics of a CMOS inverter is an important requirement. In traditional manual design procedure this is achieved by adjusting the ratio of the PMOS and NMOS transistor width assuming equal channel length. The same is also done for symmetric switching characteristics so as to make the drain-source ON resistance of the transistors same. Symmetric characteristics make the noise margin wide. However, this often comes at the cost of speed. In addition, the short channel effects of MOS transistors, which play significant role in nano scale domain make the design problem more difficult. Therefore, an optimization strategy is required such that symmetry is achieved without failing to attain the speed constraint. Thus the justification of the optimization problem lies not only in optimizing the single design objective but also in attaining design solution within a desired design specification space.

4. Results and Discussion

4.1. ANN Construction. In the present paper, two ANN models have been constructed. In the first ANN model, the four inputs are W_n, W_p, C_L, and τ_{in} and the four outputs are τ_R, τ_F, τ_{PHL}, and τ_{PLH}. For the second ANN, the two inputs are W_n and W_p and the single output is V_{SP}. For constructing the first ANN, 3000 samples have been considered. Out of these, 80% are considered as the training sample and the rest as the test sample. For constructing the second ANN, a set of 1000 samples have been considered. The structure of the first ANN is described in Table 2. The neural network

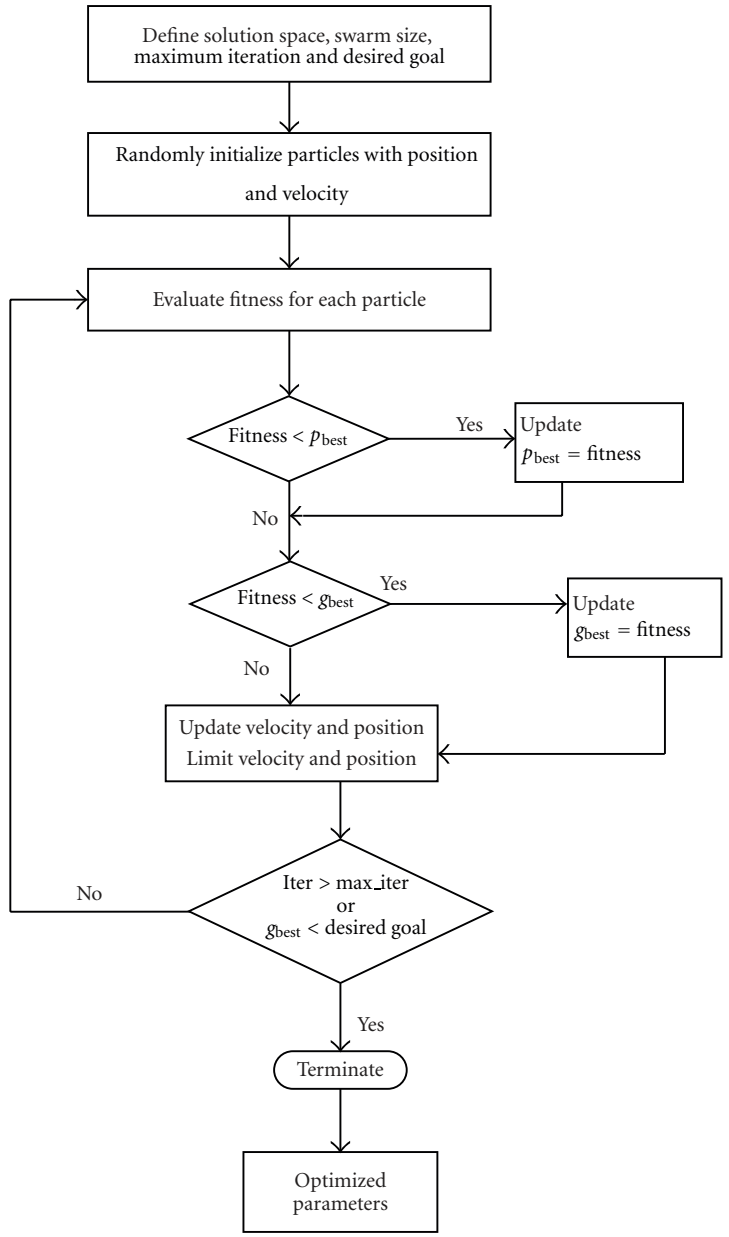


Figure 5: Flow chart of the PSO Algorithm.

Table 4: Delay constraints and design parameter bounds.

Case Study	C_L (pF)	W_n (nm)	W_p (nm)	τ_F (ns)	τ_R (ns)	τ_{PHL} (ns)	τ_{PLH} (ns)
1	0.5–2.5	45–135	90–940	0.1–15	0.1–15	0.05–8.0	0.05–8.0
2	0.5–2.5	45–110	90–620	0.1–15	0.1–15	0.05–8.0	0.05–8.0
3	0.5–1.5	45–135	90–940	0.1–15	0.1–15	0.05–8.0	0.05–8.0
4	1.0–3.0	60–160	160–945	0.1–15	0.1–15	0.05–8.0	0.05–8.0
5	1.5–3.5	60–135	135–840	0.1–15	0.1–15	0.05–8.0	0.05–8.0
6	0.3–2.0	45–90	90–540	0.1–8.0	0.1–8.0	0.05–6.0	0.05–6.0
7	0.6–1.9	60–160	135–910	0.1–7.5	0.1–7.5	0.05–5.5	0.05–5.5

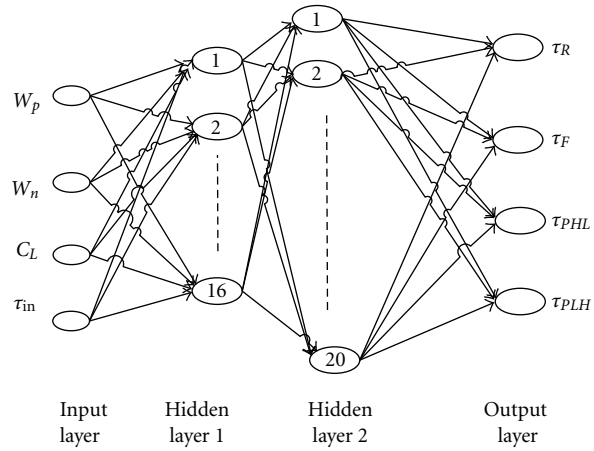
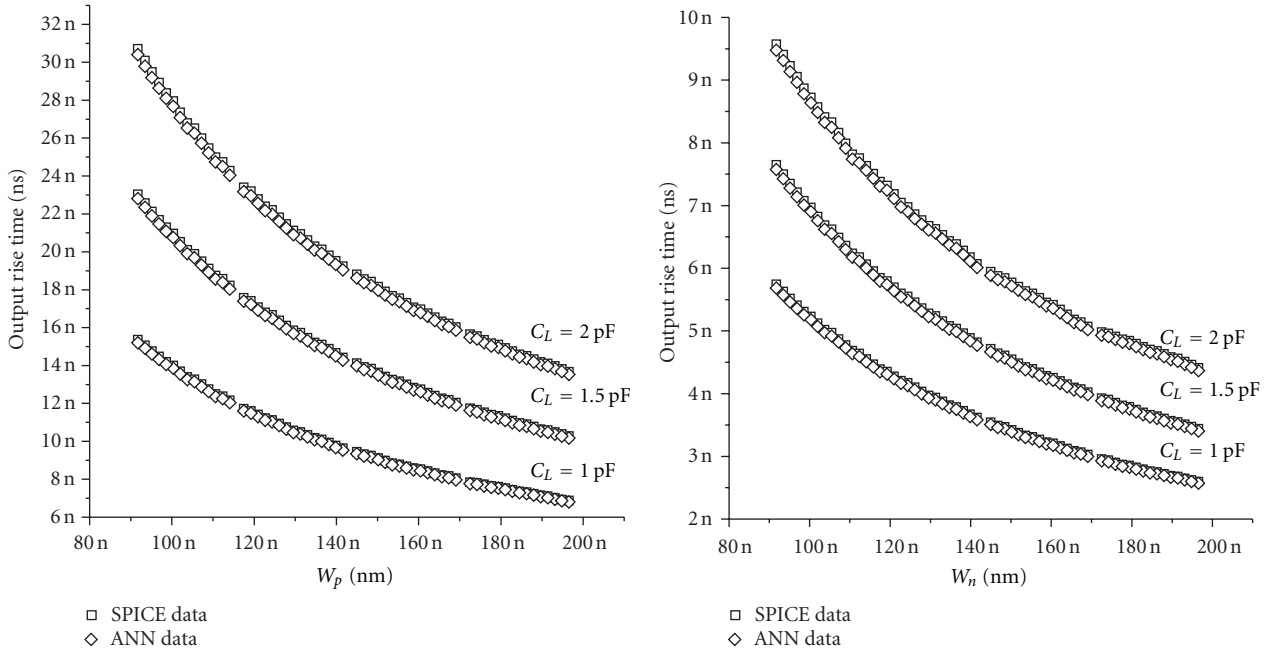


Figure 6: Architecture of the 1st ANN.

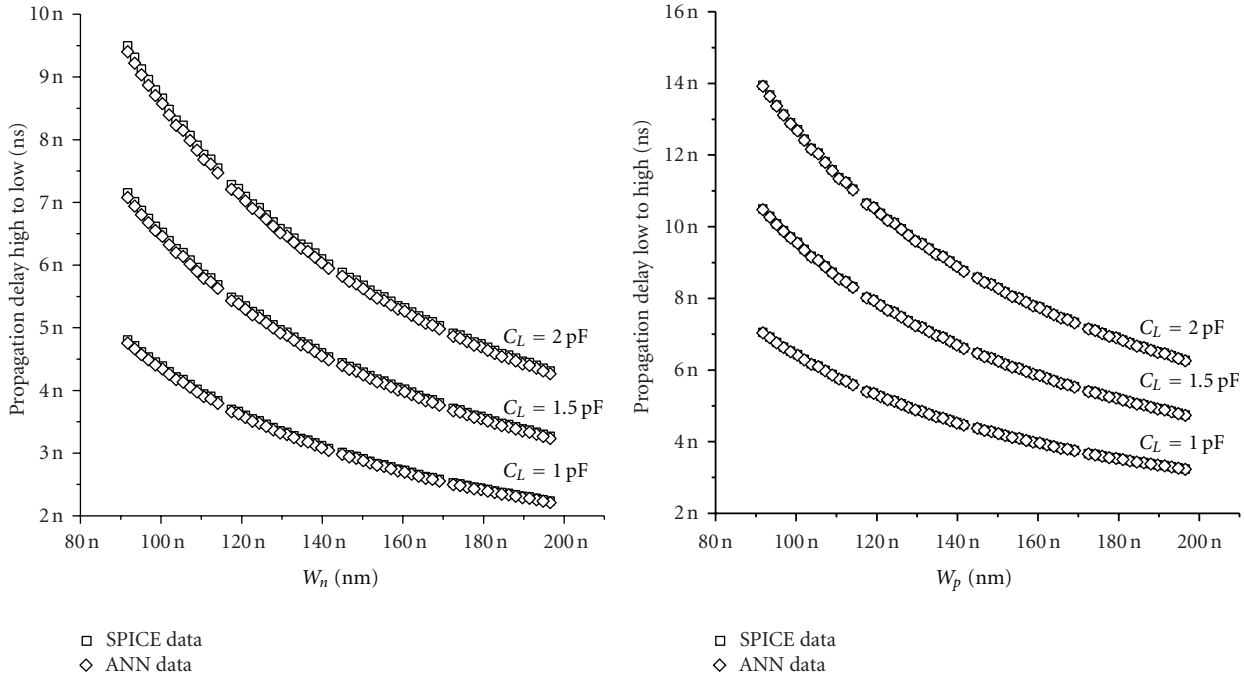


(a) Rise time (b) Fall time

Figure 7: Comparison between ANN results and SPICE results for the rise time and fall time.

Table 5: Synthesis results—τ_{in} = 1 ns.

Case study	C_L (pF)	W_n (nm)	W_p (nm)	τ_F (ns)	τ_R (ns)	τ_{PHL} (ns)	τ_{PLH} (ns)	V_{SP} (V)
1	0.83	128.37	221.93	5.1546	5.1363	2.5277	2.4648	0.4890
2	0.76	69.44	100.08	10.5131	10.5820	4.8127	4.8025	0.4861
3	0.81	125.82	217.53	5.0244	5.1146	2.6854	2.6731	0.4879
4	1.02	157.91	273.01	5.2138	5.2015	2.5812	2.5637	0.4851
5	1.66	134.70	232.88	8.8279	8.8588	4.6977	4.6785	0.4800
6	0.42	65.42	113.10	5.4471	5.4233	2.5805	2.5332	0.4887
7	0.61	95.52	165.14	5.3867	5.3678	2.7219	2.7581	0.4876



(a) High-to-low propagation delay

(b) Low-to-high propagation delay

Figure 8: Comparison between ANN results and SPICE results for the high-to-low and low-to-high delay.

Table 6: Performances of the PSO Algorithm.

Case Study	No. of iterations	No. of function evaluations	CPU time (≈min)
1	2561	76808	46.52
2	2257	67682	41.37
3	3043	91275	55.28
4	2771	83103	50.80
5	3139	94165	53.07
6	2346	70355	42.62
7	2189	65653	39.76

optimization technique is conducted for maximum 1000 iterations. However, after 725 epochs on an average the neural network model has reached the desired training goal. The neuron numbers in the first and second hidden layer is selected through trial and error method. The architecture is shown in Figure 6. For the second ANN architecture, a single hidden layer with 8 neurons has been considered.

4.2. ANN Modeling Accuracy. The accuracy is measured by comparing the results predicted from ANN and the actual SPICE simulation results. Figures 7(a) and 7(b) show the comparison between ANN predicted results and SPICE simulation results for the variations of the rise time and the fall time with W_p and W_n, respectively for various values of the load capacitors. Similarly Figures 8(b) and 8(a) shows the corresponding results for the low-to-high, and high-to-low propagation delay times. We observe that the two results closely match. The average relative error E and the correlation coefficient R are summarized in Table 3. We achieve very good accuracy in terms of both E and R.

The scatter plots between the ANN predicted results and SPICE simulations are shown in Figures 9(a)–9(d). We observe nearly perfect diagram with very close to unity correlation coefficient. These demonstrate the accuracy of the constructed ANN model. Similar levels of accuracy have been obtained for the second ANN also.

4.3. PSO-Based Design. In the present work, we have chosen seven case studies. For each case, the desired rise time, fall time, low-to-high, and high-to-low output propagation delay times are kept within a certain constraint, defined by an upper limit and a lower limit. Similarly the design parameters are also kept within a specified bound. These are tabulated in Table 4. The value of τ_{is} is assumed to be 1 ns. The synthesized values of the design parameters corresponding to which the cost function is minimized and the constraints are satified for all the case studies are shown in Table 5. It also contains the corresponding values of the performance parameters. We observe from Tables 4 and 5 that the synthesized parameters satisfy the design constraints.

The PSO algorithm has been implemented under Matlab 6.5 platform. The reduction in cost function value with number of iterations for two case studies are shown in Figures 10(a), and 10(b). We observe from these two figures, that the cost function value reduces drastically within first few iterations. Then the reduction is not significant. It takes

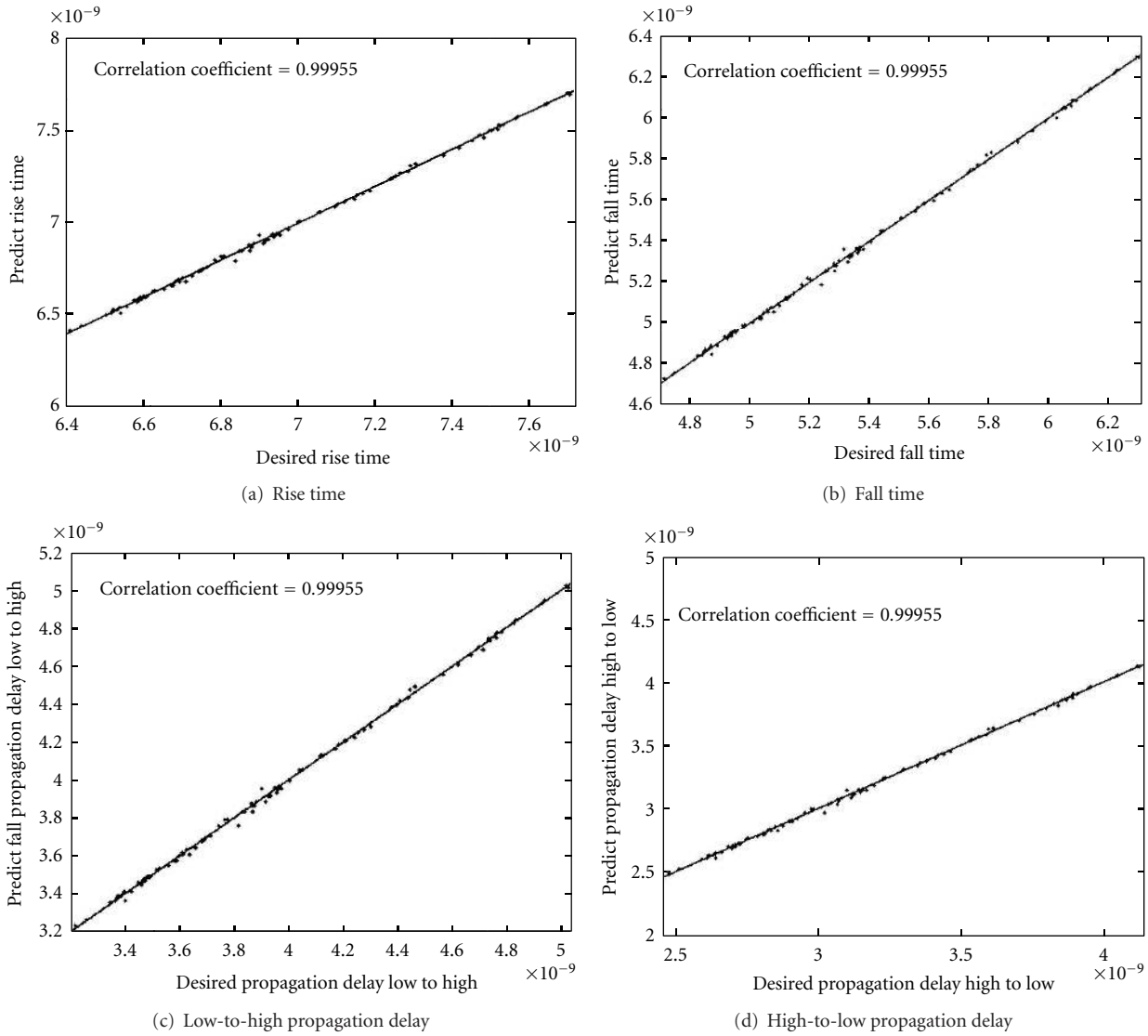


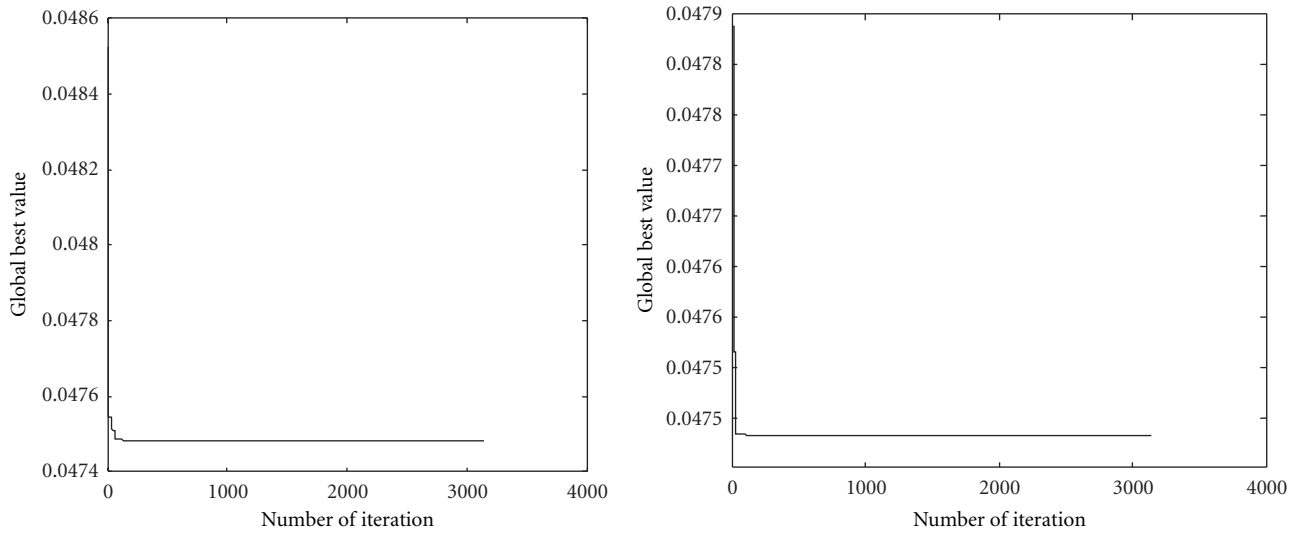
Figure 9: Correlation diagram for the various outputs of the 1st ANN.

several numbers of iterations to reach the ultimate goal. This is a characteristic of any global search algorithm. A possible way to avoid some unnecessary iterations is to judiciously combine some local optimization algorithms with the PSO.

The performance of the algorithm is measured by three factors: number of iterations, number of function evaluation, and time complexity (CPU time consumed). These are summarized in Table 6. The number of iterations is moderate. Consequently, the algorithm produces acceptable results in moderate CPU time. The measured time is w.r.t CPU of PIV processor and 1 GB RAM. The reported time includes the time taken to evaluate the ANN model during each function evaluation.

In order to validate the results obtained through PSO optimization, we select some case studies and implement them at the transistor level. The PSO-synthesized transistor widths and output load capacitor values have been considered. The channel length is taken as 45 nm with 1.0 V supply. We then perform transient simulation using T-SPICE. A comparison between the PSO-generated results and SPICE results is provided in Tables 7 and 8. We observe that the PSO generated designs yield very good results even when simulated at the SPICE level as far the symmetry of the switching characteristics is considered.

We now compare the present approach of designing an inverter for optimizing the switching characteristics with the existing approach of [10] for the same purpose. The same analytical equations as used in [10] have been used in the present work for comparison purpose. The synthesized results are provided in Table 9. With these synthesized values of the design parameters, the circuits are implemented and simulated using TSPICE. Tables 10 and 11 present the comparison results w.r.t the accuracy of the two approaches. It can clearly be observed that our approach outperforms the analytical approach of [10]. This is because the analytical models used in [10] are simplified and approximate. These



(a) Case study 3

(b) Case study 5

Figure 10: Reduction of cost function value with iterations.

Table 7: Comparison between PSO results and SPICE results— τ_R and τ_F.

Case study	PSO results			SPICE results		
	τ_R (ns)	τ_F (ns)	Difference (ns)	τ_R (ns)	τ_F (ns)	Difference (ns)
1	5.1363	5.1546	0.0183	5.0674	5.2532	0.1858
2	10.5820	10.5131	0.0689	10.6200	9.8351	0.7849
3	5.1146	5.0244	0.09012	5.0535	5.3356	0.2821
4	5.2015	5.2138	0.0123	5.0398	5.4025	0.3627
5	8.8588	8.8279	0.0309	8.6485	8.5699	0.0786
6	5.4233	5.4471	0.0238	5.2171	5.6551	0.438
7	5.3678	5.3867	0.0189	5.2649	5.6741	0.4092

Table 8: Comparison between PSO results and SPICE results—τ_{PHL} and τ_{PLH}.

Case study	PSO results			SPICE results		
	τ_{PHL} (ns)	τ_{PLH} (ns)	Difference (ns)	τ_R (PHL)	τ_F (LH)	Difference (ns)
1	2.5227	2.4648	0.0579	2.6455	2.4367	0.2088
2	4.8217	4.8025	0.0192	4.9417	4.9256	0.0161
3	2.6854	2.6731	0.0123	2.8366	2.4292	0.4074
4	2.5812	2.5637	0.0175	2.5157	2.4210	0.0947
5	4.6977	4.6785	0.0192	4.7861	4.5264	0.2597
6	2.5805	2.5332	0.0473	2.5771	2.4957	0.0814
7	2.7219	2.7581	0.0362	2.8629	2.6367	0.2262

Table 9: Synthesis results using approach of [10].

Case study	C_L (pF)	W_n (nm)	W_p (nm)	τ_F (ns)	τ_R (ns)	τ_{PHL} (ns)	τ_{PLH} (ns)	V_{SP} (V)
1	1.190	121.45	589.80	6.9369	6.9168	3.2797	3.2796	0.4861
2	0.866	76.33	370.70	7.9424	7.9193	3.7549	3.8483	0.4861
3	0.983	102.52	497.90	6.7833	6.7637	3.2070	3.2866	0.4861
4	1.87	151.78	737.10	8.5364	8.5116	4.0358	4.1361	0.4861
5	1.528	125.26	608.33	8.9247	8.8988	4.2193	4.3242	0.4861
6	0.379	77.80	377.85	3.3873	3.3775	1.6014	1.6412	0.4861
7	0.873	151.12	733.91	3.9767	3.9653	1.8802	1.9269	0.4861

Table 10: Comparison between our approach and approach of [10]—τ_R and τ_F.

Case study	Our approach			Approach of [10]		
	τ_R (ns)	τ_F (ns)	Difference (ns)	τ_R (ns)	τ_F (ns)	Difference (ns)
1	5.0674	5.2532	0.1858	2.7083	8.9890	6.2807
2	10.620	9.8351	0.7849	3.1583	10.749	7.5907
3	5.0535	5.3356	0.2821	2.6573	8.8727	6.2154
4	5.0398	5.4025	0.3627	3.3978	11.1474	7.7496
5	8.6485	8.5699	0.0786	3.4920	11.863	8.371
6	5.2171	5.6551	0.438	1.4841	4.7308	3.2467
7	5.2649	5.6741	0.4092	1.7170	5.0863	3.3693

Table 11: Comparison between our approach and approach of [10]—τ_{PHL} and τ_{PLH}.

Case study	Our approach			Approach of [10]		
	τ_{PHL} (ns)	τ_{PLH} (ns)	Difference (ns)	τ_{PHL} (ns)	τ_{PLH} (ns)	Difference (ns)
1	2.6455	2.4367	0.2088	4.6379	1.3758	3.2621
2	4.9417	4.9256	0.0161	5.5482	1.5727	3.9755
3	2.8366	2.4292	0.4074	4.5841	1.3535	3.2306
4	2.5157	2.4210	0.0947	5.7678	1.6712	4.0966
5	4.7861	4.5264	0.2597	5.9667	1.7114	4.2553
6	2.5771	2.4957	0.0814	2.2657	0.7583	1.5074
7	2.8629	2.6367	0.2262	2.6839	0.8558	1.8281

are not valid for nano scale CMOS circuit designs. For nano scale circuits more advanced compact models need to be used [2]. However, these models are often very difficult to handle in an optimization-based design approach. Thus, the present approach of using ANN for modeling purpose and embedding the constructed ANN within a global search algorithm is the most useful technique for designing any nano scale logic circuits.

5. Conclusion

This paper presents an approach for the design of a nano scale CMOS inverter using artificial neural network and particle swarm optimization techniques. The compact models used to describe MOS transistors in the nano scale domain are often very complicated and difficult to handle within an optimization-based design approach. In addition, the accuracies of such models are questionable. In this approach, ANN has been used for modeling purpose. The input design parameters of the ANN model are the widths of the PMOS and NMOS transistor, the load capacitor and the rise/fall time of the input signal. The output performance parameters are the inverter switching point, the output rise time and fall time, the output low-to-high, and high-to-low propagation delay times. It has been found that the ANN results are quite good compared to actual SPICE-level simulation results. The ANN results are used for constructing the cost functions of the PSO-based optimization procedure. The PSO-synthesized designs are actually implemented using 45 nm CMOS technology and are simulated using TSPICE. It has been observed that the ANN predicted results and actual SPICE results match very closely. The present approach of designing an inverter has been compared to an existing approach [10] and the advantage of our approach has been demonstrated through simulation results.

In addition, to achieving symmetry in switching response, optimizing delay and power consumption are important in digital CMOS circuit design. In addition, the effects of gate-level parameter variations on these performances and optimization of the same are some other challenging issues. In the next phase of our research these will be taken up. However, the general framework for modeling the performances through ANN and optimization of the design parameters through PSO will remain same.

Acknowledgment

Soumya Pandit likes to thank the Department of Science and Technology, Government of India for financial support of the present paper through Fast Track Young Scientist Scheme, Ref. no. SR/FTP/ETA-063/2009.

References

[1] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*, Pearson Education, 2004.

[2] R. A. Rutenbar, G. G. E. Gielen, and J. Roychowdhury, "Hierarchical modeling, optimization, and synthesis for system-level analog and RF designs," *Proceedings of the IEEE*, vol. 95, no. 3, Article ID 4167778, pp. 640–669, 2007.

[3] T. Sakurai and A. R. Newton, "Delay analysis of series-connected MOSFET circuits," *IEEE Journal of Solid-State Circuits*, vol. 26, no. 2, pp. 122–131, 1991.

[4] L. Bisdounis, S. Nikolaidis, and O. Koufopavlou, "Analytical transient response and propagation delay evaluation of the CMOS inverter for short-channel devices," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 2, pp. 302–306, 1998.

[5] J. L. Rosselló and J. Segura, "An analytical charge-based compact delay model for submicrometer CMOS inverters," *IEEE Transactions on Circuits and Systems I*, vol. 51, no. 7, pp. 1301–1311, 2004.

[6] Y. Wang and M. Zwolinski, "Analytical transient response and propagation delay model for nanoscale CMOS inverter," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'09)*, pp. 2998–3001, 2009.

[7] S. Miryala, B. Kaur, B. Anand, and S. Manhas, "Efficient nanoscale VLSI standard cell library characterization using a novel delay model," in *Proceedings of the 12th International Symposium on Quality Electronic Design, (ISQED '11)*, pp. 458–463, 2011.

[8] S. P. Boyd, S. J. Kim, D. D. Patil, and M. A. Horowitz, "Digital circuit optimization via geometric programming," *Operations Research*, vol. 53, no. 6, pp. 899–932, 2005.

[9] R. A. Vural, O. Der, and T. Yildirim, "Particle swarm optimization based inverter design considering transient performance," *Digital Signal Processing*, vol. 20, no. 4, pp. 1215–1220, 2010.

[10] R. A. Vural, O. Der, and T. Yildirim, "Investigation of particle swarm optimization for switching characterization of inverter design," *Expert Systems with Applications*, vol. 38, pp. 5696–5703, 2010.

[11] M. Avci, M. Y. Babac, and T. Yildirim, "Neural network based MOSFET channel length and width decision method for analogue integrated circuits," *International Journal of Electronics*, vol. 92, no. 5, pp. 281–293, 2005.

[12] N. Kahraman and T. Yildirim, "Technology independent circuit sizing for fundamental analog circuits using artificial neural networks," in *Proceedings of the 4th Ph.D. Research in Microelectronics and Electronics Conference (PRIME'08)*, pp. 1–4, June 2008.

[13] G. Wolfe and R. Vemuri, "Extraction and use of neural network models in automated synthesis of operational amplifiers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 2, pp. 198–212, 2003.

[14] S. Haykin, *Neural Networks and Learning Machines*, Pearson Education, 2009.

[15] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45 nm early design exploration," *IEEE Transactions on Electron Devices*, vol. 53, no. 11, pp. 2816–2823, 2006.

[16] L. Kuipers and H. Niederreiter, *Uniform Distribution of Sequences*, Dover Publications, 2006.

[17] S. K. Mandal, S. Sural, and A. Patra, "ANN- and PSO-based synthesis of on-chip spiral inductors for RF ICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 1, pp. 188–192, 2008.