

Buffer Insertion for Delay Minimization using An Improved PSO Algorithm

Gholamreza Karimi^{1,*} and Alireza Ahmadi²

¹ Electrical and Electronics Engineering Department, Razi University Tagh-E-Bostan, Kermanshah-67149, Iran

² Computer Engineering Department, Razi University Tagh-E-Bostan, Kermanshah-67149, Iran

Received: 25 Aug. 2013, Revised: 22 Nov. 2013, Accepted: 23 Nov. 2013

Published online: 1 Sep. 2014

Abstract: This paper considers the problem of interconnect wire delay in digital integrated circuits. The correct wire sizing and buffer insertion/sizing can reduce the interconnect delay. The interconnect wire is divided into segments and to optionally buffers are inserted between two adjacent segments. But it is important to select appropriate values for the size of buffers as well as the lengths and widths of segments to minimize the delay. Since the delay is a multi-dimensional function, its optimizing process is complex and time-consuming. In this paper we introduce an improved particle swarm optimization for delay minimization. The aim is to reduce the interconnect wire delay. This work is performed in 3 case studies. Sizing a chain of buffers without considering the wire delay is done in case study 1. Wire sizing alone and buffer insertion/sizing with wire sizing are dealt in case study 2 and case study 3 respectively. In these case studies, the our improved PSO results are matched with theoretical results while the proposed technique is very fast and more accurate than standard techniques.

Keywords: buffer insertion/sizing, wire sizing, interconnect wire delay, particle swarm optimization

1 Introduction

As VLSI technology continues to scale down, interconnect delay has become the dominant factor in deep submicron design [1]. With the continued scaling of process technology, the resistance per unit length of the interconnect continues to increase, the capacitance per unit length remains roughly constant and transistor or logic delay continues to decrease [2]. So as the feature size of VLSI devices continues to decrease, interconnect delay becomes increasingly important [3]. Buffer insertion, buffer sizing, and wire sizing have been shown to be effective techniques for interconnect delay optimization [4]. We divide the interconnect wire into several segments and to optionally insert buffers between two adjacent segments. Also we can change the size of buffers as well as the lengths and widths of segments in order to minimize the delay from source to sink. In previous years, several closed form optimal solutions for interconnect optimization problems have been given [1,4,5,6,7,8,9,10]. In these studies, the delay minimization problem of on-chip interconnect wire is considered by simultaneous buffer insertion/sizing, and wire sizing. In

fact, by manipulating the wire width, for example, the trade-off between capacitance and resistance can be balanced, and consequently the delay can be minimized [11]. On the other hand, correct buffer insertion area able to minimize signal delay by repowering the signal using amplifiers or buffers [12]. The minimal interconnect delay can be achieved by suitable wire/buffer sizing.

The mentioned works have introduced classical algorithms to find the best solution. But in recent years, utilization of evolutionary algorithms has increased as an efficient tool for automated design of integrated circuits that need to optimize. Demand for electronic circuit automation has increased due to complexity growth in VLSI circuits [13]. In VLSI circuit design, delay, power dissipation and chip area can be considered as a function of design parameter, such as W/L ratio, interconnect wire width, gate size and etc. But these functions are usually nonlinear and complex. Thus, usage of the classical solutions and algorithms for optimization of these functions is difficult and time consuming. Particle Swarm Optimization (PSO) is an efficient kind of evolutionary algorithms that can solve a variety of complicated optimization problems. The most important difference

* Corresponding author e-mail: ghkarimi@razi.ac.ir

from the classic optimization techniques is that PSO does not require any derivation which simplifies PSO structure. Because of the parameter number to be adjusted is quite small; application of PSO is very easy [13].

Recently, in many works the PSO algorithm has been applied in digital electronic circuit design. For example, the performance of PSO based inverter design considering transient performance has been investigated [13], so that the PSO algorithm found the values of load capacitance and MOS transistor dimensions to minimize the error between rise time and fall time. PSO result was matched with theoretical result. Having successful results, authors minimized the error between propagation delay times and the error between rise and fall times simultaneously using PSO algorithm [14]. PSO result was matched with theoretical result as well. Also Vural and Yildirim [15] utilized PSO for accommodating required functionalities and performance specifications considering optimal sizing of analog integrated circuits with optimization ability in short computational time. In [16] a PSO based framework is purposed for low power testing of VLSI circuits. The entire testvector is set in a frame, so that the frame consists of all those vectors strings which provide high fault coverage and also arrange vectors in frame to produce minimum toggling rate of flip flops. As an optimization tool, PSO is used to minimize number of logic gates needed to realize the 100% feasible circuits [17]. The parameters of sub-35 nm contact-hole fabrication are optimized using particle swarm optimization approach [18]. In case of field programmable gate arrays (FPGA) placement and routing, PSO is proposed to minimize the distance between configurable logic blocks (CLBs) [19]. A discrete PSO (DPSO) version is applied to the FPGA placement problem to find the optimum logic blocks and IO pins locations in order to minimize the total wire-length [20].

This paper unfolds as follows. In Section 2, the standard PSO algorithm is explained and then our technique for improving it, is presented. In Section 3, both standard and improved PSO algorithms are used to size a chain of buffers and results are compared to mathematical method. Wire sizing and optimization of it using the improved PSO algorithm are investigated in Section 4. Section 5 brings the most complete case study. For reducing the interconnect delay, buffer insertion and wire sizing techniques are utilized and the improved PSO algorithm is used to determine the most optimal sizes for buffers and wire segments. Finally, conclusions and future work are provided in Section 6.

2 Particle Swarm Optimization

2.1 Standard Particle Swarm Optimization

Since optimization of multi-dimensional and nonlinear functions using conventional computing algorithms is a

```

k=1
%initialize random particles
for i=1 to p
    particle(i).X=random value between X_LowerBond and X_UpperBound
    particle(i).V=random value between V_LowerBond and V_UpperBound
    particle(i).cost=cost_function(particle(i).X)
    particle(i).pbest=particle(i).cost
    particle(i).best_position=particle(i).X
end
[global_best_position gbest]=minimum_cost(particle)
%find best position
while(k<=maximum iteration)
    for i=1 to p
        update particle(i).X
        update particle(i).V
        particle(i).cost=cost_function(particle(i).X)
        if (particle(i).cost<particle(i).pbest)
            particle(i).pbest=particle(i).cost
            particle(i).best_position=particle(i).X
        end
    end
    [global_best_position gbest]=minimum_cost(particle)
    k=k+1
end
return [global_best_position gbest]

```

Fig. 1: Standard PSO algorithm

complex, difficult and time consuming process, evolutionary algorithms can be suitable cases in solving such problems.

The particle swarm optimization (PSO) algorithm as evolutionary algorithm for optimization was introduced by Kennedy and Eberhart in 1995 [21,22]. Each optimization problem can be considered the minimization of a cost function $f(X)$ where X is the decision vector consisting of n dimensions. In fact each dimension is an independent variable which must be limited between a determined domain. The aim of optimizing is to find the coordinates of the point which the value of the function at it be minimum. In the PSO algorithm, all particles are initialized with random coordinates in n -dimensional space, so that each particle can be a potential solution. Each particle is also assigned a randomized velocity, and then flown through n -dimensional space.

For each particle, the best coordinates ($pbest$) in n -dimensional space that it has achieved so far with the value of the function are stored. Also the best coordinates among overall particles ($gbest$) with the value of the function are stored. The movement of particles is based on $pbest$ and $gbest$. In other word, each particle adjusts its position in the search space from time to time according to the flying experience of its own and its neighbors [22]. In fact, PSO is an evolutionary computation method based on the social behavior, movement and intelligence of swarms searching for an optimal location in a multidimensional search area [13].

Each particle has a current position vector (X) and a velocity vector (V). X , V , $pbest$ for each iteration and $gbest$ are n -dimensional vectors. At the k^{th} time step (iteration), the position vector and the velocity vector of

the i^{th} particle are updated as follows:

$$V_i^{k+1} = wV_i^k + c_1rand_1^k(pb_{est}^k - X_i^k) + c_2rand_2^k(g_{best}^k - X_i^k). \quad (1)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1}. \quad (2)$$

Here, $k = 1, 2, \dots, \text{maximum iteration}$; and $i = 1, 2, \dots, p$. The number of particles is p . Acceleration (change the velocity) is weighted by a random term, with separate random numbers being generated for acceleration toward pb_{est} and g_{best} [22]. The acceleration factors c_1 and c_2 indicate the relative attraction toward pb_{est} and g_{best} respectively and also $rand_1$ and $rand_2$ are random numbers uniformly distributed between zero and one [13]. Inertia weight parameter w controls the trade-off between the global search ability and local search ability during the optimization process [23]. In order to avoid premature convergence, PSO utilizes a distinctive feature of controlling a balance between global and local exploration of the search space which prevents from being stacked to local minima [13]. The PSO algorithm can be expressed as figure (1).

As we increase the maximum iteration and the number of particles, the accuracy will be raised, but more time is needed for execution. Each dimension of vector X and vector V must be limited between lower bound and upper bound. These bounds are determined based on the parameter of the problem which is supposed to be optimized.

2.2 Improved Particle Swarm Optimization

Determining appropriate limits for search space is important for the success of the PSO algorithm [14]. If the determined domain is symmetric to optimal point, not only PSO result will be more accurate, but also it will be achieved in much less time. Each cost function may have some local optimal point, so that the particles around them which are away from global optimal point get close the PSO result to themselves and took it away from the best result. On the other hand, execution time increases with larger search space. To solve these problems, we modify the standard PSO algorithm as after some iteration and obtaining an approximate result, we limit the domain of each dimension of vector X and make it symmetric to the approximate result, and the standard PSO procedure is called again. Our improved PSO algorithm is given in figure (2). In the algorithm, d is an experimental parameter which is used to limit search space. In this work, d is assigned by 10. It is notable that number of iterations in each standard PSO procedures call is a portion of maximum iteration.

```

k=1 %%k is a global counter
do
{
[global_best_position gbest]=standard_pso(k,X_bond)
X_lower bound=global_best_position - ((X_UpperBound - X_LowerBound)/d)
X_Upper bound=global_best_position + ((X_UpperBound - X_LowerBound)/d)
} while(k<=maximum iteration)
return [global_best_position gbest]
    
```

Fig. 2: Improved PSO algorithm

3 Sizing a chain of buffers using PSO

3.1 Factors affecting the propagation delay

The propagation delay of each logical gate is a function of its equivalent resistance and load capacitance. The load capacitance can be divided into an intrinsic and an extrinsic component. Intrinsic component represents the self-loading or intrinsic output capacitance, and is associated with the diffusion capacitances of the NMOS and PMOS transistors as well as the gate-drain overlap (Miller) capacitances. Extrinsic component is the extrinsic load capacitance, attributable to fanout and wiring capacitance [24]. The equivalent resistance and capacitance are dependent on MOS transistor dimensions. The diffusion and Miller capacitances are proportional to the width of the transistors, but equivalent resistance is reverse proportional to it [24]. On the other hand, while sizing up, input gate capacitance which is a component of the extrinsic load capacitance of previous stage, is increased.

3.2 Closed form of the delay

In [24], the propagation delay of a chain (see figure (3)) is derived as following equations:

$$t_{po} = 0.69R_{ref}C_{iref}, \quad (3)$$

$$t_{p,j} = t_{po} \left(1 + \frac{C_{g,j+1}}{\gamma C_{g,j}}\right) = t_{po} \left(1 + \frac{f_j}{\gamma}\right), \quad (4)$$

$$t_p = \sum_{j=1}^N t_{p,j} = t_{po} \sum_{j=1}^N \left(1 + \frac{C_{g,j+1}}{\gamma C_{g,j}}\right), C_{g,N+1} = C_L. \quad (5)$$

R_{ref} and C_{iref} represent equivalent resistance and intrinsic capacitance of reference gate respectively. Generally, reference gate is an inverter or a buffer with minimum dimensions. γ is a proportionality factor, which is only a function of technology and is close to 1, for most sub-micron processes [24]. The input gate capacitance of j^{th} inverter which is proportional to its size, is shown by $C_{g,j}$. So f_j is the ratio of between the size of $(j+1)^{th}$ and j^{th} inverter. C_L shows load capacitance such that the chain drive it. $t_{p,j}$ and t_p represent the delay of j^{th} inverter and the total delay of the chain respectively.

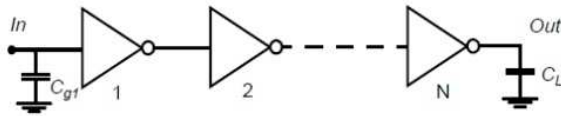


Fig. 3: Chain of N inverters with fixed input and output capacitance [24]

3.3 Closed form of minimum total delay

The aim is to minimize t_p , such that $C_{g,1}$ as input capacitance and C_L are fixed. Generally, $C_{g,1}$ is a minimally-sized device. The total delay is a function that has $N - 1$ independent variable, being $C_{g,2}, C_{g,3}, \dots, C_{g,N}$. The minimum delay can be found using mathematical methods -taking partial derivatives, and equating them to 0-. The result has been given in [24]. The optimum size of each inverter is the geometric mean of its neighbors sizes,

$$C_{g,j} = \sqrt{C_{g,j-1}C_{g,j+1}}. \quad (6)$$

In fact all inverter must be sized up by similar factor. With $C_{g,1}$ and C_L given, sizing factor called is derived as follow:

$$f = \sqrt[N]{\frac{C_L}{C_{g,1}}} = \sqrt[N]{F}, \quad (7)$$

and the minimum delay through the chain is

$$t_p = Nt_{po} \left(1 + \frac{\sqrt[N]{F}}{\gamma}\right). \quad (8)$$

3.4 Experimental results of delay minimization using the PSO algorithm

In this paper we use the parameters of the $0.18\mu\text{m}$ technology that can be seen in table (1). In this section, we consider a chain of buffers with the given parameters in table (1), and apply the PSO algorithm to minimize the total delay as compared to closed form which was expressed. The PSO Algorithm finds the size of buffers, so that the total delay as cost function is minimized. Equation (5) shows the cost function, such that the size of each buffer is proportional to the value of gate capacitance. The inputs of the PSO algorithm are listed in table (2).

In fact, $C_{g,2}, C_{g,3}, \dots, C_{g,N}$ compose the vector X and the total delay is $f(X)$. The PSO algorithm must investigate search space and find the best values of the vector X , for minimizing $f(X)$. As previously mentioned, each element of the vector X , must be limited between a determined domain. The lower bound and upper bound for $C_{g,i}$, with $2 \leq i \leq N$ are fixed to $C_{g,1}$ and C_L respectively.

Table 1: Parameters of the $0.18\mu\text{m}$ technology [4]

r_o	0.0679Ω
r_e	$17.1K\Omega$
c_o	$0.0596fF/\mu\text{m}^2$
c_f	$0.0641fF/\mu\text{m}$
c_g	$0.234fF$
c_d	$3.883fF$
W_{min}	$0.18\mu\text{m}$

r_o : Unit square wire resistance.

r_e : Effective output resistance of a minimum device.

c_o : Unit area wire capacitance.

c_f : Unit length wire fringing capacitance.

c_g : Gate capacitance of a minimum device.

c_d : Drain capacitance of a minimum device.

W_{min} : Minimum wire width

Table 2: Inputs of the PSO algorithm for a chain of buffers

$C_{g,1}$	c_g
C_L	$100c_g$
γ	1
C_{iref}	c_g
R_{ref}	r_e
Number of people	2000
Maximum iteration	120

Table 3: Results of sizing the chain of buffers using 3 methods

Method	Mathematical	Standard PSO	Improved PSO
f_2	3.162219	3.123944	3.162219
f_3	3.162219	3.183042	3.162219
f_4	3.162219	3.157432	3.162219
Total Delay	45.967628ns	45.968699ns	45.967628ns

$f_2: \frac{C_{g,2}}{C_{g,1}}$ $f_3: \frac{C_{g,3}}{C_{g,2}}$ $f_4: \frac{C_{g,4}}{C_{g,3}}$

Total Delay : The minimum delay through the 4 buffers chain.

3.4.1 Minimum delay for various number of buffers

To determine number of buffers, so that the total delay is minimized, we have used the improved PSO algorithm for $N = 1$ to 40. As can be seen in figure (4), the delay is minimized when the number of buffers is equal to 4.

3.4.2 Minimum delay for an optimal number of buffers

Both the improved PSO and the standard PSO have been applied to minimize the delay through the chain with 4 buffers and their results have been compared with the results of mathematical method which are expressed in equations (7, 8). For minimizing delay, the ratio between two adjacent gates must be equal to sizing factor which is given in equation (7). The results are summarized in table (3).

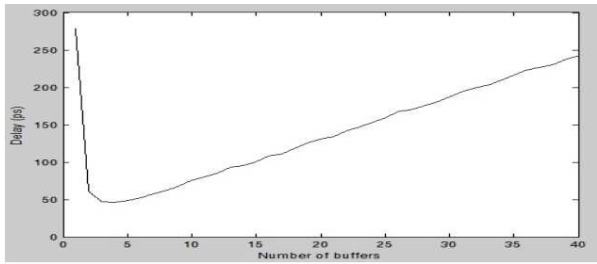


Fig. 4: Total delay through a chain of buffers

Table 4: Characteristics of hardware and software used for this work

Workstation	Laptop Acer 5750G
CPU frequency	2.1 GHz
CPU cores	Core™ i3-2310M
Cache	3MB L3
O.S	Win 7 Ultimate
Software	MATLAB 7.8.0

As shown in table (3), the results of the improved PSO algorithm are exactly matched with the mathematical method whereas the standard PSO is taken error between 0.15 % and 1.21%.

For sizing a chain with 4 buffers to minimize delay, execution time of the improved PSO algorithm was about 1.86 seconds while execution time of the standard PSO algorithm was about 11.07 seconds. It means we have almost 5.93x speedup. Characteristics of hardware and software used for this work are presented in table (4).

4 Wire sizing using PSO

4.1 Closed form of the delay

In past, the interconnect wire delay was not often considered. But nowadays with the advancement in technology and because of reducing device dimensions, increasing parasitic effects of wires and increasing circuits speed, interconnect wire delay has become important factor in design of digital integrated circuits. The interconnect delay is caused by resistive and capacitive behavior of wires. In this section the optimal wire sizing using PSO is investigated to minimize the delay. The interconnect wire is divided into several segments and consequently the delay will be a function of length and width of the wire segments. So, closed form of the delay as a cost function is used for the PSO algorithm, such that search space is composed by the length and width of the wire segments.

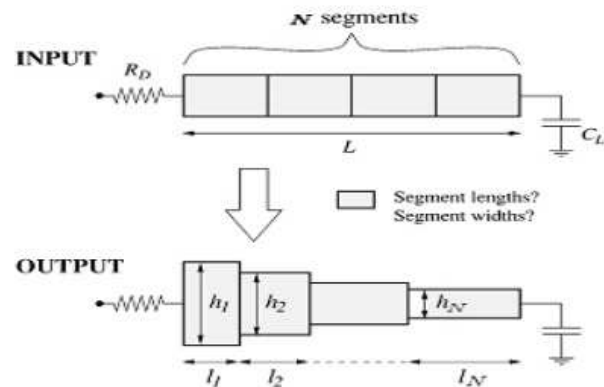


Fig. 5: The segments of an interconnect wire [4]

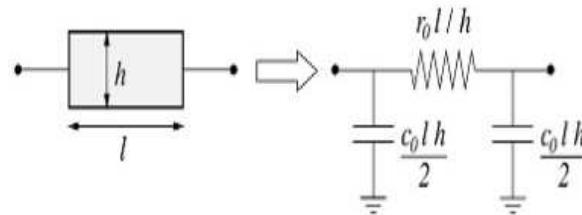


Fig. 6: The π -type RC circuit model for a wire segment with length of l and width of h [4]

The interconnect wire is divided into N segments so that if the wire length is L , sum of N segment lengths is equal to L . Driver resistance and load capacitance are R_D and C_L respectively (see figure (5)). Each wire segment is modeled as a π -type RC circuit as shown in figure (6). Therefore, the interconnect wire accompanied by the driver resistance and load capacitance form an RC network. So, according to Elmore model, the interconnect delay is obtained by equation (9) [4]:

$$\begin{aligned}
 D = & R_D(c_0l_1h_1 + c_0l_2h_2 + \dots + c_0l_Nh_N + C_L) \\
 & + \frac{r_0l_1}{h_1} \left(\frac{c_0l_1h_1}{2} + c_0l_2h_2 + \dots + c_0l_Nh_N + C_L \right) \\
 & + \frac{r_0l_2}{h_2} \left(\frac{c_0l_2h_2}{2} + c_0l_3h_3 + \dots + c_0l_Nh_N + C_L \right) \\
 & \vdots \\
 & + \frac{r_0l_N}{h_N} \left(\frac{c_0l_Nh_N}{2} + C_L \right).
 \end{aligned} \tag{9}$$

Table 5: Inputs of the PSO algorithm for wire sizing

R_D	C_L	Number of people	Muximum teration	L
$r_e/200$	$200C_g$	$1000N$	300	$10000\mu m$

Table 6: Minimum delays for various number of wire segments

Number of Segments	Minimum Delay in [4]	Minimum Delay Using Improved PSO	Percentage of Minimum Delay Reduction
4	233.58 ps	214.89 ps	8.07%
5	231.5 ps	212.90 ps	8.03%
10	227.5ps	210.24ps	7.59%
15	226.75ps	209.74ps	7.50%
20	226.5ps	209.61ps	7.46%
25	226.35ps	209.56ps	7.42%
30	226.25ps	209.52ps	7.39%

4.2 Experimental results of delay minimization using the PSO algorithm

Inputs of the PSO algorithm for this section are listed in table (5). As can be seen both the driver and load are 200 times bigger than the minimum device. Another note that since search space becomes larger due to increasing number of segments, in the same way, we also increase the number of people, consequently the accuracy will not fall.

Chris et al. [4] illustrated for optimal solution of wire sizing, the segment lengths must be all equal to L/N . So we use the PSO algorithm to determine the segment widths for delay minimization. Therefore here, the vector X is composed by h_1, h_2, \dots, h_n . The low bound and upper bound for the width of each wire segment are set to W_{min} and $35 \times W_{min}$ respectively.

We can get a smaller interconnect wire delay by using more segments [4]. Table (6) depicts the minimum delays for different number of wire segments. As shown in table (6), using the improved PSO, the minimum delay reduction is between 7.39% and 8.07% for various number of segments compared to the theoretical methods explained in [4].

The PSO algorithm determines the segment widths to obtain minimum delay. Also, we compute segment widths using the proposed method in [4]. The results of two algorithms are listed in table (7), where the wire is divided into 4 segments. An interesting result revealed by table (7) is that the wire segment widths are approximately halved from source to sink. For each segment, there are two wire widths obtained by two methods in which the last row shows error percentage caused by difference between them. It can be seen that the error percentage is between 3.38% and 7.34% for different segments. These error percentages lead to make the difference between the minimum delays achieved by the methods.

Table 7: The segment widths for the wire with 4 segments

	h_1	h_2	h_3	h_4
The Method Proposed in [4]	$1.96\mu m$	$1.16\mu m$	$0.59\mu m$	$0.28\mu m$
Improved PSO	$2.12\mu m$	$1.10\mu m$	$0.57\mu m$	$0.29\mu m$
Error Percentage	7.34%	5.29%	4.06%	3.38%

5 Buffer insertion/sizing and wire sizing using PSO

Buffers can reduce the interconnect delay in digital integrated circuits due to repowering signals. It has been estimated that the buffers will constitute 35% of circuit cells in the 65nm technology [25]. But it is important to determine buffer sizes, so that the delay is minimized. In this section, we investigate buffer insertion/sizing and wire sizing techniques for delay minimization in a wide range of experiments. To address this issue, firstly we compute the propagation delay as a function of buffer sizes, wire segment widths and wire segment lengths. Next, we minimize the function by means of the improved PSO algorithm and the algorithm proposed in [4]. Finally, the optimum sizes for getting minimum delay are shown in detail.

5.1 Closed form of the delay

The interconnect wire is divided into M segments and a buffer is inserted between two adjacent segments. Also the segment wire between two buffers, in turn, is divided into N segments (as shown in figure (7)). The buffer sizes, the distance between buffers and the wire segment widths are variable and all must be determined by the theoretical methods or the PSO algorithm, for delay minimization. In fact, this section is the most comprehensive case in the interconnect delay optimization problem.

Considering figure (7) and the RC model for a buffer shown in figure (8), the propagation delay between i^{th} and $(i+1)^{th}$ buffer is obtained by equation (10).

$$\begin{aligned}
 d_i &= \frac{r_e}{b_i} (c_d b_i + c_0 l_1 h_1 + c_0 l_2 h_2 + \dots + c_0 l_N h_N + c_g b_{i+1}) \\
 &+ \frac{r_0 l_1}{h_1} \left(\frac{c_0 l_1 h_1}{2} + c_0 l_2 h_2 + \dots + c_0 l_N h_N + c_g b_{i+1} \right) \\
 &+ \frac{r_0 l_2}{h_2} \left(\frac{c_0 l_2 h_2}{2} + c_0 l_3 h_3 + \dots + c_0 l_N h_N + c_g b_{i+1} \right) \\
 &\vdots \\
 &+ \frac{r_0 l_N}{h_N} \left(\frac{c_0 l_N h_N}{2} + c_g b_{i+1} \right) \\
 &= \frac{r_e}{b_i} (c_d b_i + \left(\sum_{k=1}^N c_0 l_k h_k \right) + c_g b_{i+1}) \\
 &+ \sum_{j=1}^N \frac{r_0 l_j}{h_j} \left(\frac{c_0 l_j h_j}{2} + \left(\sum_{k=j+1}^N c_0 l_k h_k \right) + c_g b_{i+1} \right). \quad (10)
 \end{aligned}$$

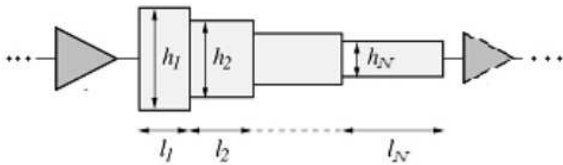


Fig. 7: Wire sizing between two buffers

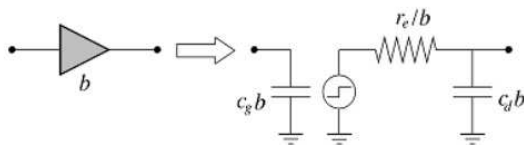


Fig. 8: The model of a buffer of size $b \times$ minimum device by a switch-level RC circuit [4]

The interconnect wire of length L is divided into $M + 1$ segments, when M buffers are inserted. Let for $0 \leq i \leq M$, S_i be length of the wire which is between i^{th} buffer and $(i + 1)^{th}$ buffer. Driver can be buffer 0 and also load can be buffer $M + 1$. For $0 \leq j \leq N$, we consider l_j to be S_i/N . Note that as mentioned, h_j for $0 \leq j \leq N$, S_i for $0 \leq i \leq M$, and b_i for $0 \leq i \leq M$, are all the output results of any minimization mechanism for the buffer insertion/sizing and wire sizing problem. Also for wire fringing capacitance consideration, we add half of the total fringing capacitance to the load [4]. Closed form of the total delay D , explained in following equation, is multi dimensional function, which must be optimized. We employ our improved PSO algorithm to minimize the delay as cost function.

$$\begin{aligned}
 D = & R_D \left(\sum_{k=1}^N c_0 l_k h_k \right) + c_g b_1 \\
 & + \sum_{j=1}^N \frac{r_0 l_j}{h_j} \left(\frac{c_0 l_j h_j}{2} + \left(\sum_{k=j+1}^N c_0 l_k h_k \right) + c_g b_1 \right) \\
 & + \sum_{i=1}^{M-1} d_i + \frac{r_e}{b_M} (c_d b_M + \left(\sum_{k=1}^N c_0 l_k h_k \right) + C_L) \\
 & + \sum_{j=1}^N \frac{r_0 l_j}{h_j} \left(\frac{c_0 l_j h_j}{2} + \left(\sum_{k=j+1}^N c_0 l_k h_k \right) + C_L \right) + \left(\frac{L}{2} \right) c_f \quad (11)
 \end{aligned}$$

5.2 Experimental results of delay minimization using PSO algorithm

Inputs of the PSO algorithm for this section are listed in table (8, 9). To compare between the PSO results and the

Table 8: Inputs of the PSO algorithm for buffer insertion/sizing and wire sizing

R_D	C_L	Maximum iteration	Number of segments between buffers
$r_e/200$	$200 \times c_g$	300	6

Table 9: Inputs of the PSO algorithm for buffer insertion/sizing and wire sizing

Interconnect wire lengths	Number of buffers	Number of people
$1000\mu m \leq L \leq 5000\mu m$	0	5000
$6000\mu m \leq L \leq 13000\mu m$	1	10000
$14000\mu m \leq L \leq 20000\mu m$	2	15000

mathematical results, the parameters in table (9) is given like to paper [4]. Note that when number of buffers is zero, we have wire sizing alone.

In this section, vector X is composed by the buffer sizes (b_1, b_2, \dots, b_M) , the distance between buffers (S_0, S_1, \dots, S_M) , and the wire segment lengths (h_0, h_2, \dots, h_N) . The buffer sizes must be limited between minimum device and 200 times bigger than the minimum device (less than or equal to the load). The lower bound of the distance between buffers is W_{min} . We set the upper bound of the distance between buffers to L . But consider that there is a constraint for them. Sum of the distance between buffers should be equal to L due to satisfy the constraint. The lower bound and upper bound of wire segment lengths are set to W_{min} and $35 \times W_{min}$ respectively similar to previous section.

We run our improved PSO algorithm for different lengths of the interconnect wire similar to paper [4]. The minimum delays using our improved PSO algorithm compared to mathematical method explained in [4], are given in table (10). As can be seen the improved PSO algorithm reduces the minimum interconnect delay between 33.17% and 36.74% for different lengths.

Also more results are listed in table (11) for an interconnect wire of length $20000\mu m$. As can be seen distances between buffers have become equal to each other, in which the errors are lower than 0.28%. The error percentage for segment widths achieved by two algorithms is between 14.21% and 18.28%. To achieve minimum delay in RC model, both algorithms result that the size of buffers should be equal to the biggest size as possible.

6 Conclusions and future work

In this work, we introduced an improved particle swarm optimization and applied it for buffer insertion/sizing and wire sizing optimization problem. The aim was to reduce the interconnect wire delay. Because it is a very effective factor in modern design of the integrated circuits. We

Table 10: Minimum delays for various interconnect wire lengths

Interconnect Wire Length	Minimum Delay in [4]	Minimum Delay Using Improved PSO	Percentage of Minimum Delay Reduction
1000 μm	22.24ps	14.76ps	33.63%
2000 μm	45.87ps	29.50ps	35.69%
3000 μm	75.06ps	47.81ps	36.30%
4000 μm	109.81ps	69.46ps	36.74%
5000 μm	147.34ps	94.31ps	35.99%
6000 μm	180.70ps	116.54ps	35.51%
7000 μm	202.94ps	131.52ps	35.19%
8000 μm	226.57ps	147.45ps	34.92%
9000 μm	252.98ps	164.76ps	34.87%
10000 μm	280.78ps	183.04ps	34.81%
11000 μm	309.97ps	202.97ps	34.52%
12000 μm	339.16ps	222.52ps	34.39%
13000 μm	371.13ps	244.54ps	34.11%
14000 μm	400.32ps	263.69ps	34.13%
15000 μm	425.34ps	281.96ps	33.71%
16000 μm	450.36ps	299.98ps	33.39%
17000 μm	478.16ps	319.17ps	33.25%
18000 μm	505.96ps	338.05ps	33.19%
19000 μm	535.15ps	357.53ps	33.19%
20000 μm	564.34ps	377.14ps	33.17%

Table 11: Results of two algorithms for the wire of length 20000 μm

	Improved PSO	The Method Proposed in [4]	Error Percentage
Distance between driver and buffer 1	6650.35 μm	6666.66 μm	0.24%
Distance between buffer 1 and buffer 2	6664.15 μm	6666.66 μm	0.04%
Distance between buffer 2 and load	6685.50 μm	6666.66 μm	0.28%
Width of segment 1	2.30 μm	1.88 μm	18.28%
Width of segment 2	1.75 μm	1.50 μm	14.21%
Width of segment 3	1.24 μm	1.15 μm	16.17%
Width of segment 4	0.89 μm	0.93 μm	14.56%
Width of segment 5	0.67 μm	0.71 μm	15.10%
Width of segment 6	0.48 μm	0.44 μm	17.88%
size of buffer 1	200x	200x	0%
size of buffer 2	200x	200x	0%

used RC model of the wire and the buffer to calculate its delay and then the delay as the cost function was minimized by the PSO algorithm. The experimental results of mentioned algorithm was matched with theoretical optimization method for case study 1 and better for 2 another case studies. Also we showed our improved PSO was very faster and more accurate than the standard PSO.

Considering inductance effect on the wire delay, we will use RLC model for delay calculation and optimization in future work.

References

- [1] C.-P. Chen, Y.-P. Chen and D.F. Wong, Optimal Wire-Sizing Formula Under the Elmore Delay Model, Proc. ACM/IEEE Conf. Design Automation Conference, 487-490 (1996).
- [2] C.J. Alpert, A. Devgan and S.T. Quay, Buffer Insertion for Noise and Delay Optimization, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, **18**, 1633-1645 (1999).
- [3] C. Chu and D.F. Wong, A New Approach to Simultaneous Buffer Insertion and Wire Sizing, Proc. IEEE/ACM Conf. Computer-Aided Design, 614-62 (1997).
- [4] C. Chu and D.F. Wong, Closed Form Solution to Simultaneous Buffer Insertion/Sizing and Wire Sizing, ACM Transactions on Design Automation of Electronic Systems, **6**, 343-371 (2001).
- [5] C.-P. Chen and D.F. Wong, A Fast Algorithm for Optimal Wire-Sizing Under Elmore Delay Model, Proc. IEEE Symp. Circuits and Systems (ISCAS 96), **4**, 412-415 (1996).
- [6] C.-P. Chen, H. Zhou and D.F. Wong, Optimal Non-Uniform Wire-Sizing Under the Elmore Delay Model, Proc. IEEE/ACM Conf. Computer-Aided Design (ICCAD 96), 38-43 (1996).
- [7] C. Alpert and A. Devgan, Wire Segmenting for Improved Buffer Insertion, Proc. ACM/IEEE Conf. Design Automation Conference, 588-593 (1997).
- [8] C.-P. Chen and D.F. Wong, Optimal Wire-Sizing Function With Fringing Capacitance Consideration, Proc. ACM/IEEE Conf. Design Automation Conference, 604-607 (1997).
- [9] C. Chu and D.F. Wong, A Quadratic Programming Approach to Simultaneous Buffer Insertion/Sizing and Wire Sizing, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, **18**, 787-798 (1999).
- [10] C. Chu and D.F. Wong, A Polynomial Time Optimal Algorithm for Simultaneous Buffer and Wire Sizing, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, **18**, 1297-1304 (1998).
- [11] J. Lillis and C.K. Cheng, Simultaneous Routing and Buffer Insertion for High Performance Interconnect, Proc. Sixth Great Lakes Symp. VLSI, 148-153 (1996).
- [12] L.P.P. van Ginneken, Buffer Placement In Distributed RG Tree Networks for Minimal Elmore Delay, Proc. IEEE Symp. Circuits and Systems, **2**, 865-868 (1990).
- [13] R.A. Vural, O. Der and T. Yildirim, Particle Swarm Optimization Based Inverter Design Considering Transient Performance, Digital Signal Processing, **20**, 1215-1220 (2010).
- [14] R.A. Vural, O. Der and T. Yildirim, Investigation of Particle Swarm Optimization for Switching Characterization of Inverter Design, Expert Systems with Applications, **38**, 5696-5703 (2011).
- [15] R.A. Vural and T. Yildirim, Analog Circuit Sizing Via Swarm Intelligence, International Journal of Electronics and Communications, **66**, 732-740 (2012).
- [16] B. Singh, S.B. Narang and A. Khosla, Particle Swarm Optimization Framework for Low Power Testing of VLSI Circuits, International Journal of Artificial and Applications, **2**, 13-20 (2011).
- [17] P. Moore and G.K. Venayagamoorthy, Evolving Digital Circuits Using Hybrid Particle Swarm Optimization and Differential Evolution, International Journal of Neural Systems, **16**, 163-177 (2006).
- [18] T.-S. Li and C.-M. Hsu, Parameter Optimization of Sub-35 nm Contact-hole Fabrication Using Particle Swarm Optimization Approach, Expert Systems with Applications, **37**, 878-885 (2010).
- [19] V.G. Gudise and G.K. Venayagamoorthy, FPGA placement and routing using particle swarm optimization, Proc. IEEE Symp. IEEE Computer society Annual, 307-308 (2004).

- [20] M. El-Abd, H. Hassan, M. Anis, M.S. Kamel and M. El-masry, Discrete Cooperative Particle Swarm Optimization for FPGA Placement, *Applied Soft Computing*, **10**, 284-295 (2010).
- [21] Kennedy and R.C. Eberhart, Particle swarm optimization, *Proc. IEEE International Conf. Neural Networks* (Perth, Australia), 1942-1948 (1995).
- [22] R.C. Eberhart and J. Kennedy, A New Optimizer Using Particle Swarm Theory, *Proc. Sixth International Sym. Micro Machine and Human Science*, 39-43 (1995).
- [23] D. Zhou, X. Gao, G. Liu, C. Mei, D. Jiang and Y. Lio, Randomization in Particle Swarm Optimization for Global Search Ability, *Expert Systems with Applications*, **38**, 15356-15364 (2011).
- [24] J.M. Rabaey, *Digital Integrated Circuits : A Design Perspective*, Calif.: PHI Learning, 201-204 (2003).
- [25] N. Menezes, P. Cocchini and D.A. Kirkpatrick, Repeater scaling and its impact on CAD, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **23**, 451-463 (2004).
-



Gholamreza Karimi

was born in Kermanshah, Iran in 1977. He received the B.S. and M.S. and PhD degrees in electrical engineering from Iran University of Science and Technology (IUST) in 1999, 2001 and 2006 respectively. He is currently an Assistant Professor in

Electrical Department at Razi University, Kermanshah, since 2007. His research interests include low power analog and digital IC design, RF IC design, modeling and simulation of RF mixed signal IC, microwave devices and artificial intelligence systems.



Alireza Ahmadi received the B.S. degree in Computer Engineering (Hardware) from Azad University of Kashan, Iran, in 2011 and he is now an M.Sc collegian in Computer Architectural Engineering in Razi University of Kermanshah, Iran. His research interests include

optimization solutions, evolutionary algorithms, high performance systems design, parallel programming, and hardware implementation of image processing algorithms.