

Research Article

Simple Exact Algorithm for Transistor Sizing of Low-Power High-Speed Arithmetic Circuits

Tooraj Nikoubin,¹ Poona Bahrebar,² Sara Pouri,² Keivan Navi,² and Vaez Iravani²

¹ Faculty of Electrical and Computer Engineering, Shahid Beheshti University, GC, 1983963113, Tehran, Iran

² Microelectronic Laboratory of Faculty of Electrical and Computer Engineering, Shahid Beheshti University, GC, 1983963113, Tehran, Iran

Correspondence should be addressed to Tooraj Nikoubin, t_nikoubin@sbu.ac.ir

Received 23 May 2009; Revised 8 October 2009; Accepted 21 January 2010

Academic Editor: Wieslaw Kuzmicz

Copyright © 2010 Tooraj Nikoubin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new transistor sizing algorithm, SEA (Simple Exact Algorithm), for optimizing low-power and high-speed arithmetic integrated circuits is proposed. In comparison with other transistor sizing algorithms, simplicity, accuracy, independency of order and initial sizing factors of transistors, and flexibility in choosing the optimization parameters such as power consumption, delay, Power-Delay Product (PDP), chip area or the combination of them are considered as the advantages of this new algorithm. More exhaustive rules of grouping transistors are the main trait of our algorithm. Hence, the SEA algorithm dominates some major transistor sizing metrics such as optimization rate, simulation speed, and reliability. According to approximate comparison of the SEA algorithm with MDE and ADC for a number of conventional full adder circuits, delay and PDP have been improved 55.01% and 57.92% on an average, respectively. By comparing the SEA and Chang's algorithm, 25.64% improvement in PDP and 33.16% improvement in delay have been achieved. All the simulations have been performed with 0.13 μm technology based on the BSIM3v3 model using HSpice simulator software.

1. Introduction

Optimization of VLSI circuits relies heavily on efficient implementation of arithmetic operations considering signal delay, power consumption, and chip area. Such important design considerations and trade-offs lead to a general approach towards transistor sizing that will prove to be extremely useful. In fact, transistor sizing, that is, the operation of enlarging or reducing the channel width of transistors, is a powerful and effective performance optimization tool in the hands of the designer.

Although VLSI circuits can be optimized in a number of ways, such as circuit style selection, structural optimizations, and transistor sizing [1–9], various reasons exist as to why transistor sizing is an important issue. First of all, the power dissipation is a strong function of transistor sizing which affects physical capacitance. Sources of power consumption such as glitches and short-circuit currents can be minimized

by careful circuit design and transistor sizing. Second, transistor sizing affects not only the resistance of devices and time constant but also propagation delay of the gate due to the parasitic capacitors. Third, careful transistor sizing is necessary to maintain sufficient noise margins. Even more important is the observation that transistor sizing becomes critical in ensuring proper functionality of a circuit. For all the above reasons, transistor sizing is an essential means of implementing high-performance circuits [10].

Furthermore, with the continued scaling of technology and reduced transistor sizes, the behavior and performance of a circuit could not be investigated without transistor sizing. From simulation results, it has been noticed that a small change in the transistor size for a given technology leads to a remarkable change in the characteristics of a circuit. Therefore, using an appropriate transistor sizing method is necessary for a circuit prior to measuring its parameters.

One practical formulation that recognizes a designer's objective to achieve the best performance at a given time period can be stated as

$$\begin{aligned}
 & \text{Minimize Power } (w) \\
 & \text{Subject to Delay } (w) \leq T_{\text{spec}}, \\
 & \text{Area} \leq A_{\text{spec}}, \\
 & \text{each Gate Size} \geq \text{Minimize},
 \end{aligned} \tag{1}$$

where both *Power* and *Delay* parameters are functions of the transistor size, $w \in R^n$, and n represents the number of transistors in a circuit. T_{spec} and A_{spec} are, respectively, the constraints on the circuit delay and area, and *Minimize* is the minimum transistor size allowed by the technology [11]. Note that the optimization problem specified in this formula must be solved at one subcircuit at a time, and although the entire circuit may have a million or more transistors, the number of variables in the sizing problem will be reasonably small.

Transistor sizing is somewhat complicated especially within complex circuits. Sizing a transistor to speed up one signal path may slow down another due to the capacitive loading effect of path interactions. This technique should therefore be used with caution. The algorithms which are presented in [12, 13] illustrate a linear method to make a trade-off between power, area, and delay in CMOS circuits. In [14], the relationship between transistor sizes and total circuit delay has been considered as nonlinear. It has been shown by Fishburn and Dunlop [14] that the transistor sizing problem is a convex under the simple lumped RC model.

Hence, a transistor sizing tool to meet our requirements in the performance optimization of VLSI circuits is crucial. An appropriate algorithm is the one that is simple to analyze and implement, decreases the execution time, increases the optimization rate of the goal parameter, and offers flexibility in choosing different optimization parameters.

Selecting the optimization factor is dependent on our needs. At present, for many researches, Power-Delay Product (PDP) which is metric for energy consumption of a circuit, is vitally important and the transistors has been sized to meet the minimum PDP [1, 2, 6, 15, 16].

Several approaches have been applied in transistor sizing, one being a Mathematical optimization method [17–21]. The transistor sizing problem is formulated as a constrained nonlinear mathematical program of optimization factors. MDE (Minimum Delay Estimation) and ADC (Area-Delay Curve) algorithms [22] are within this group. Another approach is a Heuristic approach which was proposed for the first time in TILOS [14]. In this method, the transistor sizes have been changed iteratively until optimization is reached. In another approach, the combination of these two methodologies has been used. In this way a two-stage approach to combine the advantages of the heuristic and the mathematical programming techniques have been proposed. After using a heuristic method to perform an initial sizing, a timing analyzer and a mathematical optimizer are utilized to optimize the design. In this paper, through

the review of the pros and cons of various transistor sizing approaches, a systematic and effective algorithm to size the transistors of various full adder cells for minimal energy consumption is suggested. The objective of our work is to explore the improvement in the performances of different full adders that could be obtained using various transistor sizing algorithms.

The rest of this paper is organized as follows. Section 2 reviews the transistor sizing algorithms based on logical effort. In Section 3, heuristic algorithms are analyzed and surveyed, where we begin with the attributes of Chang's algorithm as it is closest to our proposed algorithm. Then, after exploring the rules of grouping transistors for sizing operation, we propose our new algorithm. In Section 4, the circuits are simulated and the results are analyzed and compared. Finally, the conclusion is presented in Section 5.

2. Transistor Sizing Based on Logical Effort

One of the transistor sizing approaches is to use mathematical techniques. Logical effort is a technique to solve the transistor sizing problem in this way. Two algorithms, MDE and ADC, which are dealt with, are based on logical effort. The work [22] shows how these algorithms can size the transistors without running a heuristic sizing tool by calculating the minimum achievable delay and the cost of achieving a target delay.

This approach estimates the size of transistors without incurring the overhead of running a sizing tool. In this manner, different implementations are evaluated based on two metrics. First, the problem of estimating the minimum delay is considered. This metric allows a designer to determine whether an implementation can meet a given delay specification. The delay of a circuit is the maximum delay of all *Input* to *Output* paths of the circuit. In order to meet design goals, transistor sizing is applied to the circuit to reduce this delay. The smallest delay value that can be obtained in this way is referred to as the minimum achievable delay. Due to the associated high area overheads, most circuits other than their critical paths are rarely sized in order to meet this minimum delay value. In addition, the minimum achievable delay along with the unsized circuit delay helps to determine the range of delay value over which an implementation can be used. In short, these two algorithms present a technique that estimates the minimum achievable delay of a circuit which then goes to trace the area-delay curve.

2.1. MDE Algorithm. Briefly, MDE algorithm estimates the minimum achievable delay of a specific implementation. The most significant point in this algorithm is that the minimum achievable delay is computed for all fan-out paths of a transistor. MDE algorithm obtains the Delay- C_{in} curve (C_{in} is the input capacitance of a circuit) based on some formulas as in [22].

The advantages of MDE algorithm are [22] as follows.

- (1) This algorithm is very adaptable for tree structures as multiple paths can be processed at the same time.

- (2) This approach is utilized in gaining the actual size of a transistor.
- (3) It is faster than the transistor sizing tools which work heuristically. For instance, MDE has less execution time in comparison with TILOS in [14].
- (4) The circuit can be sized at a minimum delay.
- (5) Assume that there are K sizes for each gate in a circuit with N gates and the maximum fan-out on any gate $|FO|$. According to the pseudocode in [21], the innermost *for* loop is executed $O(K \cdot |FO|)$ times and the cost of determining the maximum delay point is $O(|FO|)$. The second *for* loop is executed K times. Finally, since there are N gates in a circuit, the outermost *for* loop is executed N times. Thus, the running time of MDE algorithm is $O(N \cdot K^2 \cdot |FO|^2)$.
- (6) The circuit has been sized for minimum achievable delay, and the total size of circuit increases. Hence, it can just be used for optimizing critical paths.
- (7) According to the first problem, there is no desirable trade-off between chip area, delay, and power consumptions; thus PDP is not optimum.

2.2. ADC Algorithm. Not all circuits need to be sized to operate at the minimum achievable delay. For these circuits, a target delay is known and multiple implementations of the circuit are available. Therefore, ADC algorithm trades area for delay and also achieves a reduction in power. The cost (in terms of area) for achieving the given delay is determined, which results in estimating the entire area-delay curve of each implementation. The characteristics of ADC algorithm are as follows [21].

- (1) It is in the class of mathematical algorithms.
- (2) ADC utilizes the Delay- C_{in} curve for sizing, which is the output of MDE algorithm.
- (3) The sized circuits have further delay versus smaller area.
- (4) It is not an accurate solution but can estimate the suboptimal area value.

According to the pseudocode in [22], the execution time of ADC is $O(N)$, where N is transistor count.

As we see, MDE and ADC algorithms are the complementary of each other. The delay parameters are generated in MDE algorithm and then ADC algorithm uses them for calculating PDP parameter.

3. Heuristic Transistor Sizing Algorithms

3.1. Chang's Algorithm. By optimizing the transistor sizes of the circuits, it is possible to reduce the delay without significantly increasing the power consumption and transistor sizes can be set to achieve minimum PDP [1]. To provide a fair and insightful evaluation of circuits, a systematic and effective way of sizing the transistors for optimal performance is necessary. To provide a good trade-off between the conflicting

sizing requirements for power and delay performances, the goal of optimization is to minimize the power-delay product, that is, the energy consumption. Chang's algorithm is an appropriate algorithm for sizing the transistors of a circuit and it is suitable for optimizing PDP [2]. The characteristics and functionality of this convergent algorithm are similar to our proposed algorithm.

Initially in Chang's algorithm, the sizes of the transistors in the circuit are reasonably set. The scaling operations are carried out in several iterations transistor by transistor. In [2], $w_j(T_i)$ is the width of the i th transistor at step j and Θ_k is the PDP of the circuit of the k th iteration. For every optimization iteration, one transistor at a time is tuned for minimal PDP in $2 \times m$ steps with a step resolution of $\pm\psi$. The optimization stops when the performance difference in two successive iterations is smaller than a given error ϵ . More than one iteration may be necessary because each time a new transistor is sized in the current run, the other transistors sized in the previous run may no longer maintain their optimality [2].

In order to obtain enough coverage so that the optimal or quasioptimal sizing falls in the search region, the step resolution, ψ , is made variable. Large step size is used at the first few iterations. Two optimization strategies are adopted in the previous procedure of transistor sizing to accelerate the process.

- (1) The corresponding pMOS and nMOS in a complementary pair are optimized in successive runs because the output transitions of the node driven by one transistor is often influenced mostly by the driving capability of its complementary counterparts.
- (2) Series transistors or parallel transistors of the same type that source current to or sink current from the same node have equal size and can be optimized simultaneously [2].

3.2. Proposed Transistor Sizing Algorithm

3.2.1. Transistor Sizing in Three-Dimensional Space. Transistor sizing is not a linear issue on the basis that modification of a transistor size within a circuit influences the circuit performances. In this space, we encounter different sizes and different parameters. Therefore, we must focus on an n -dimensional space and in general one of the parameters in this space will be optimized, meaning that; $n - 1$ dimensions are related to K 's. In this paper, K_i is the coefficient which is assigned for i th group of transistors in a circuit. The initial value of each K is in fact the initial coefficient of channel width of transistors in a group.

In other words K 's are known as the sizing factors of transistors and the n th parameter is the actual target which we desire to optimize. This target can be any of the circuit specifications including that of power consumption, delay, power-delay product, chip area, or their combination with a given weight. If transistor sizing is used to optimize only one target parameter, then the other parameters of the circuit may be neglected and may even lead to weak designs. For this reason, we can apply a product of two or

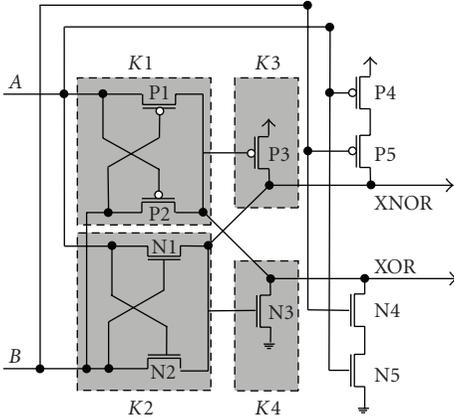


FIGURE 1: The XOR/XNOR part of NEW HPSC Full Adder.

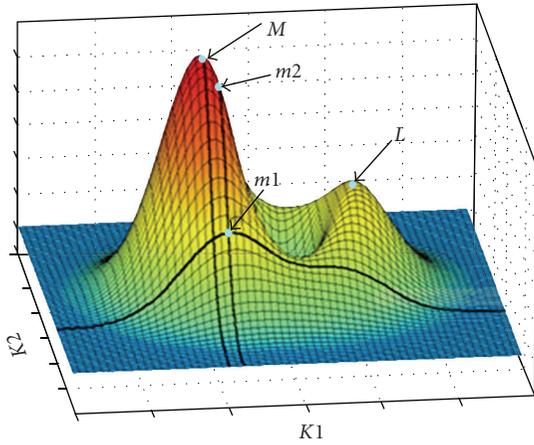


FIGURE 2: Transistor sizing concept in a three-dimensional spaces [6].

more target parameters with determined weights in forming a new parameter. Then transistor sizing is performed for this new parameter [6]. As an example, the $K1$ to $K4$ are sizing optimization coefficients of transistors in illustrated gray groups which is shown in the XOR/XNOR circuit in Figure 1. These coefficients can optimize parameters such as the power-delay product. Therefore, transistor sizing must be performed in a five dimensional space.

Hence, in an n -dimensional space, the Kis are sequentially modified until the target parameter arrives at the best possible position. In fact, changing the size of a transistor in an n -dimensional space can modify the circuit specifications. With this in mind, changing the size of transistor T_i in an n -dimensional space, initiates the process of transistor sizing. There are a lot of directions that transistor sizing can undertake in an n -dimensional space. Unfortunately, we can only change one Ki in an n -dimensional space at a given time. Undertaking various sequential operations goes to generate multiple peaks whereby each has their own advantages.

Consequently, we are seeking an appropriate algorithm for transistor sizing that has the following features. Firstly, the highest peak is to be obtained indicating that target

Simple Exact Algorithm (SEA)

```

Group all the transistors in the circuit using Transistor
Grouping Rules;
Initialize  $W_i$ , width of the transistors in group  $i$ , so that:
 $W_i = k_i \times W^*$ ;
 $k_i = 1; \quad i = 1, 2, \dots, n$ 
 $W^* = 0.13 \mu\text{m}$ ;
Initialize  $S$ , the step size and  $m$ , the number of step sizes;
for ( $t = 1$  to  $m$ ) {
    // This loop sweeps  $W^*$  to obtain  $W_{\text{opt}}$ .
     $W^* = W^* + s$ ;
    Compute the target parameter,  $\Theta$  and save it to an array;}
 $W_{\text{opt}}$  is the point where  $\Theta$  has become minimum for that;
 $W_i = k_i \times W_{\text{opt}}$ ;
do{
    for ( $i = 1$  to  $n$ ) {
        // This loop sweeps  $k_i$ .
        for ( $t = 1$  to  $m$ ) {
             $k_i = k_i + s$ ;
            Compute  $\Theta$  and Save it to another array;}}
     $k_i$  is the point where  $\Theta$  has become minimum for that;}
while (all  $k_i$ 's converge to an specific value);

```

ALGORITHM 1: Proposed transistor sizing algorithm (SEA) for optimizing.

TABLE 1: Results for XOR/XNOR 6 T in 1.2 V.

W_i	W_{N1}	W_{N2}	W_{P1}	W_{P2}	W_{FBN}	W_{FBP}
k_i	k_1		k_2		k_3	k_4
$W^* = 0.65 (\mu\text{m})$	1		1		1	1
PDP (fj)			0.282			
k_i	3		0.5		1	0.6
PDP (fj)	0.257		0.243		0.243	0.208
k_i	3.1		0.5		1	0.6
PDP (fj)	0.207		0.207		0.207	0.207

TABLE 2: Results after using random sizing factors for XOR/XNOR 6 T in 1.2 V.

W_i	W_{N1}	W_{N2}	W_{P1}	W_{P2}	W_{FBN}	W_{FBP}
k_i	k_1		k_2		k_3	k_4
$W^* = 0.39 (\mu\text{m})$	1.9		1.2		1.5	3
PDP (fj)			0.348			
k_i	5		1.2		1.6	0.8
PDP (fj)	0.328		0.328		0.323	0.218
k_i	5.2		1.1		1.6	0.8
PDP (fj)	0.218		0.217		0.217	0.217
k_i	4.7		1		1.6	0.8
PDP (fj)	0.214		0.211		0.211	0.211
k_i	4.5		0.9		1.6	0.8
PDP (fj)	0.211		0.210		0.210	0.210
k_i	4.4		0.9		1.6	0.8
PDP (fj)	0.209		0.209		0.209	0.209

parameter such as PDP has reached its best value and remaining peaks are below the optimum peak. Secondly,

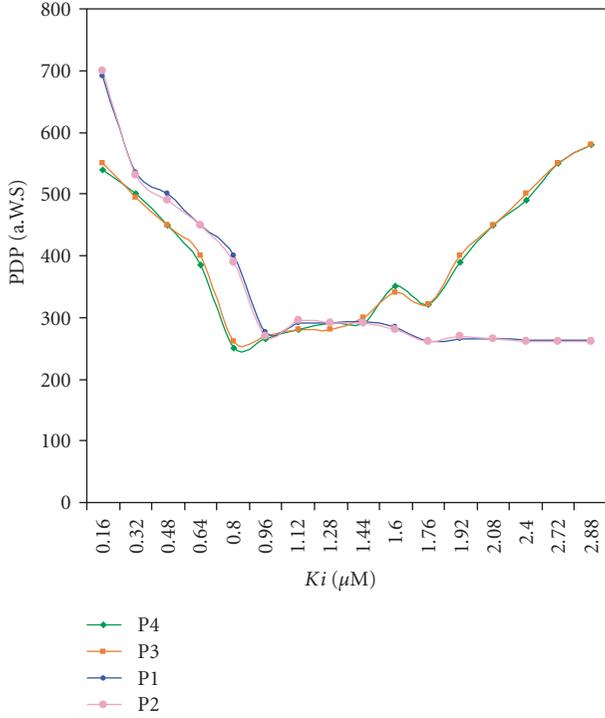


FIGURE 3: The behavior of transistors P1 and P2 which are in the same position, and P4 and P5 which are series in New HPSC circuit.

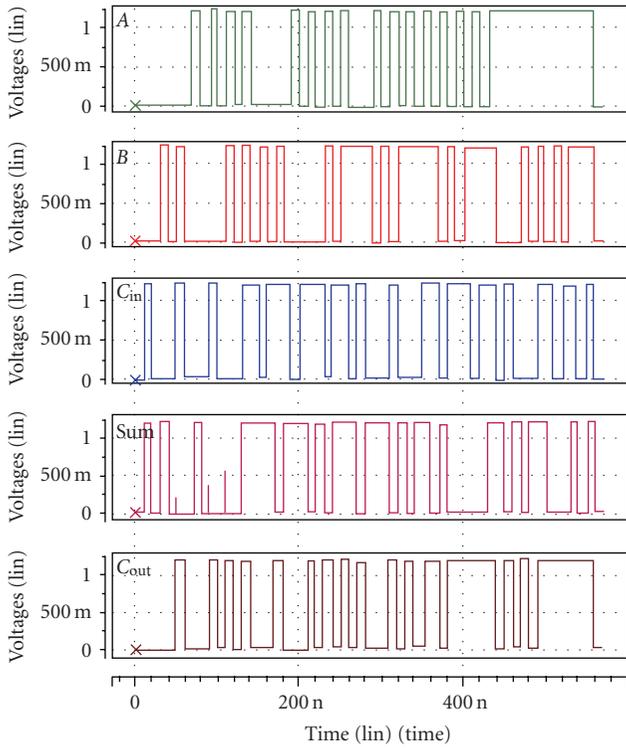


FIGURE 4: Input stimulus for a full adder cell. The first three are inputs, A, B, and C_{in} , and the remaining two are Sum and C_{out} outputs with a supply voltage of 1.2 V.

TABLE 3: Results after using random sizing factors for XOR/XNOR 6T in 1.2 V.

W_i	W_{N1}	W_{N2}	W_{P1}	W_{P2}	W_{FBN}	W_{FBP}
k_i	k_1		k_2		k_3	k_4
$W^* = 0.47 (\mu\text{m})$	1.4		2.5		1.4	2.4
PDP (fj)	0.383					
k_i	4.2		1.1		1.5	0.7
PDP (fj)	0.351		0.307		0.305	0.210
k_i	4.3		1.1		1.5	0.7
PDP (fj)	0.209		0.209		0.209	0.209

TABLE 4: Power, Delay, and Power-Delay Product (PDP) comparison of full adders using MDE and ADC algorithms.

V_{DD} (V)	0.8	1.2	1.6	2.0
	Power (μW)			
C-CMOS	1.52	2.36	4.60	8.24
CPL	1.42	3.65	5.21	9.27
TFA	1.07	3.44	5.29	10.27
TGA	1.1	3.49	5.31	10.4
New 14T	1.13	3.53	15.84	26.92
10T	1.76	5.38	16.20	28.00
New HPSC	0.96	2.44	4.74	8.23
	Delay (ns)			
C-CMOS	0.29	0.26	0.30	0.29
CPL	0.37	0.34	0.28	0.30
TFA	0.34	0.29	0.24	0.21
TGA	0.36	0.31	0.29	0.25
New 14T	0.32	0.3	0.27	0.26
10T	0.40	0.40	0.41	0.40
New HPSC	0.21	0.19	0.22	0.23
	Power-Delay product (fj)			
C-CMOS	0.44	0.61	1.38	2.39
CPL	0.53	1.24	1.45	2.78
TFA	0.36	1.00	1.27	2.16
TGA	0.39	1.08	1.53	2.6
New 14T	0.36	1.05	4.2	6.9
10T	0.70	2.15	6.64	11.20
New HPSC	0.20	0.46	1.04	1.89

this algorithm must be as simple as possibly can be and the number of steps needed to reach the peaks must be at a minimum. This means that it should be neither computationally expensive nor time consuming. Also, this algorithm must be capable of being simulated by related softwares such as HSpice.

Figure 2 describes transistor sizing in the simplest manner. According to this sample figure, we assume that we have a three-dimensional space, which can be considered as a circuit consisting of transistors that can be categorized into two groups. $K1$ and $K2$ are sizing factors of transistors of the mentioned groups. In this space, point M which is the absolute maximum or peak represents the parameter that is intended to be optimized by transistor sizing. To acquire this

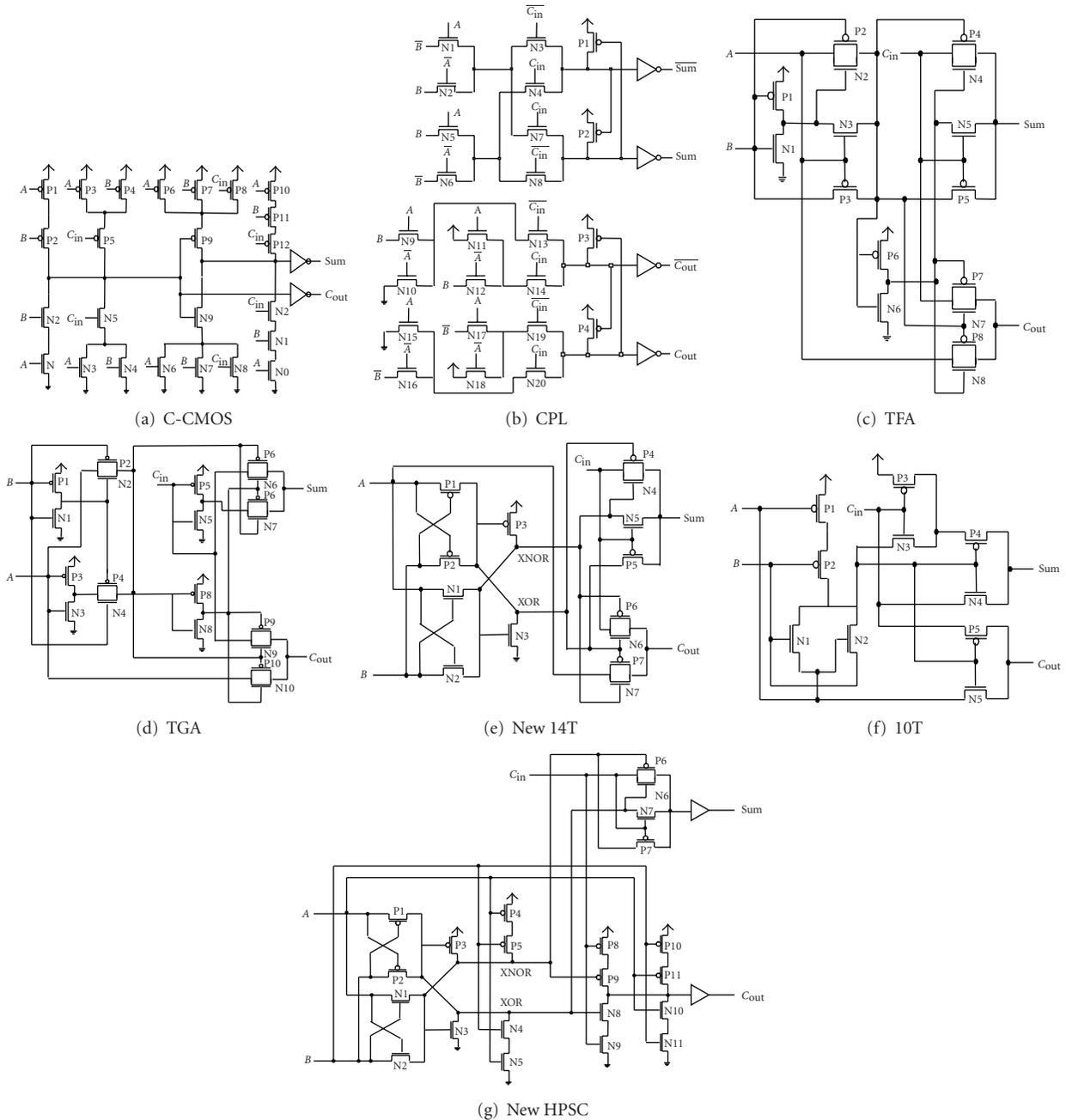


FIGURE 5: Full adder cells used for simulation.

parameter, the sizing factors of transistors must converge or must be within a specified confine. Consequently, our aim in optimizing the circuits or specifying the size of transistors is obtaining appropriate values for $K1$ and $K2$ such that we can reach the absolute optimum. The issue concerning three-dimensional spaces can be easily extended to four dimensions, five dimensions, and up to an n -dimensional space.

This algorithm begins by sweeping $K1$ while keeping $K2$ constant. In this condition, $m1$ on the curve which is the relative optimum in the current state is chosen. Now we consider $K1$ to be constant at point $m1$ and we sweep

$K2$. Then we choose point $m2$ on the curve as the relative optimum parameter. In return, the value of $K2$ at point $m2$ is considered as constant. This part of the algorithm is within a loop, because each time the size of a particular transistor is chosen, it is not guaranteed that the result we obtain is optimum for the other previously sized transistors.

In general, the number of iterations depends on the technique we employ to acquire the peak. The algorithm terminates when values of $K1$ and $K2$ in successive iterations become identical. When this happens, we are assured that point M is the absolute optimized parameter. Otherwise, the algorithm may terminate when we place conditions on

TABLE 5: Power, Delay, and Power-Delay Product (PDP) comparison of full adders using proposed transistor sizing algorithm (SEA).

$V_{DD}(V)$	0.8	1.0	1.2	1.4	1.6	1.8	2.0
Power (μW)							
C-CMOS	0.94	1.54	2.36	3.46	4.87	6.63	8.42
CPL	1.41	2.30	3.57	5.27	7.35	9.89	12.93
TFA	0.95	1.66	2.62	3.63	5.17	7.15	9.23
TGA	1.07	1.73	2.72	3.95	5.48	7.40	9.60
New 14T	1.25	1.84	2.56	3.66	5.15	6.75	8.66
10T	2.57	4.12	5.68	7.82	10.94	16.21	21.97
New HPSC	0.97	1.61	2.60	3.65	5.00	6.70	8.67
Delay (ns)							
C-CMOS	0.38	0.22	0.17	0.13	0.11	0.10	0.10
CPL	0.27	0.17	0.13	0.11	0.09	0.08	0.08
TFA	0.26	0.13	0.09	0.08	0.07	0.06	0.05
TGA	0.24	0.15	0.11	0.09	0.07	0.06	0.06
New 14T	0.44	0.17	0.11	0.08	0.06	0.05	0.05
10T	0.48	0.28	0.18	0.10	0.08	0.06	0.06
New HPSC	0.28	0.17	0.12	0.10	0.08	0.07	0.07
Power-Delay product (fJ)							
C-CMOS	0.35	0.35	0.40	0.46	0.54	0.65	0.80
CPL	0.39	0.40	0.46	0.56	0.68	0.84	1.02
TFA	0.25	0.21	0.24	0.30	0.35	0.42	0.50
TGA	0.26	0.27	0.29	0.34	0.41	0.47	0.57
New 14T	0.55	0.32	0.28	0.29	0.32	0.36	0.43
10T	1.24	1.14	1.01	0.80	0.84	0.98	1.24
New HPSC	0.27	0.27	0.31	0.36	0.42	0.50	0.60

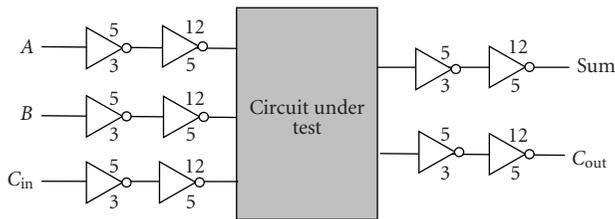


FIGURE 6: Simulation test bench.

the target parameter to be varied over a small range. To ensure that our sizing is optimum within the search window, the step size in sweeping is changeable. In this regard, when we get closer to the optimum peak, we can change the step size to a smaller value to improve accuracy. In other words, if the step size is small, consequently the error rate is small, accuracy is increased, and the optimum peak is gained.

It is clear that if at the beginning $K2$ and then $K1$ are swept, point M will be chosen again as the unique absolute optimum. In other words, in this algorithm the order of sizing factors of transistors is not important and it is not effective in obtaining the optimum value of the target parameter.

Another factor that may cause problems in transistor sizing is local peaks on the curve because there is a danger in selecting local peaks instead of the absolute peak. However,

considering Figure 2, if, in the algorithm, point L which is the local peak is assumed to be the optimum point and since the algorithm is iterative, in the next iteration by sweeping $K1$ or $K2$, we certainly exit from point L and point M is chosen as the optimum point.

One important point is that a symmetrical three-dimensional space can increase the probability and speed of selecting an optimum point. For example, if sphere is considered as the most symmetrical object in a three-dimensional space, from any location that $K1$ and $K2$ are swept, the same position is finally reached.

In this regard, the best approach in this field is to place conditions for the n -dimensional space to become symmetrical. This is dependable on how the transistors of a circuit are categorized. In this paper, we will introduce a new rule in grouping transistors; that is, transistors with similar positions are placed into one category. This leads to the improvement of the circuit performances and an increase in the optimization speed by means of a symmetrical n -dimensional space.

3.2.2. The Rules of Grouping Transistors for Sizing Operation.

Due to the fact that the time needed for transistor sizing is severely influenced by the number of sizing factors, it is necessary to reduce these factors as much as possible before determining the transistor sizes. Thus, the complexity and execution time of sizing operation decreases. This abatement

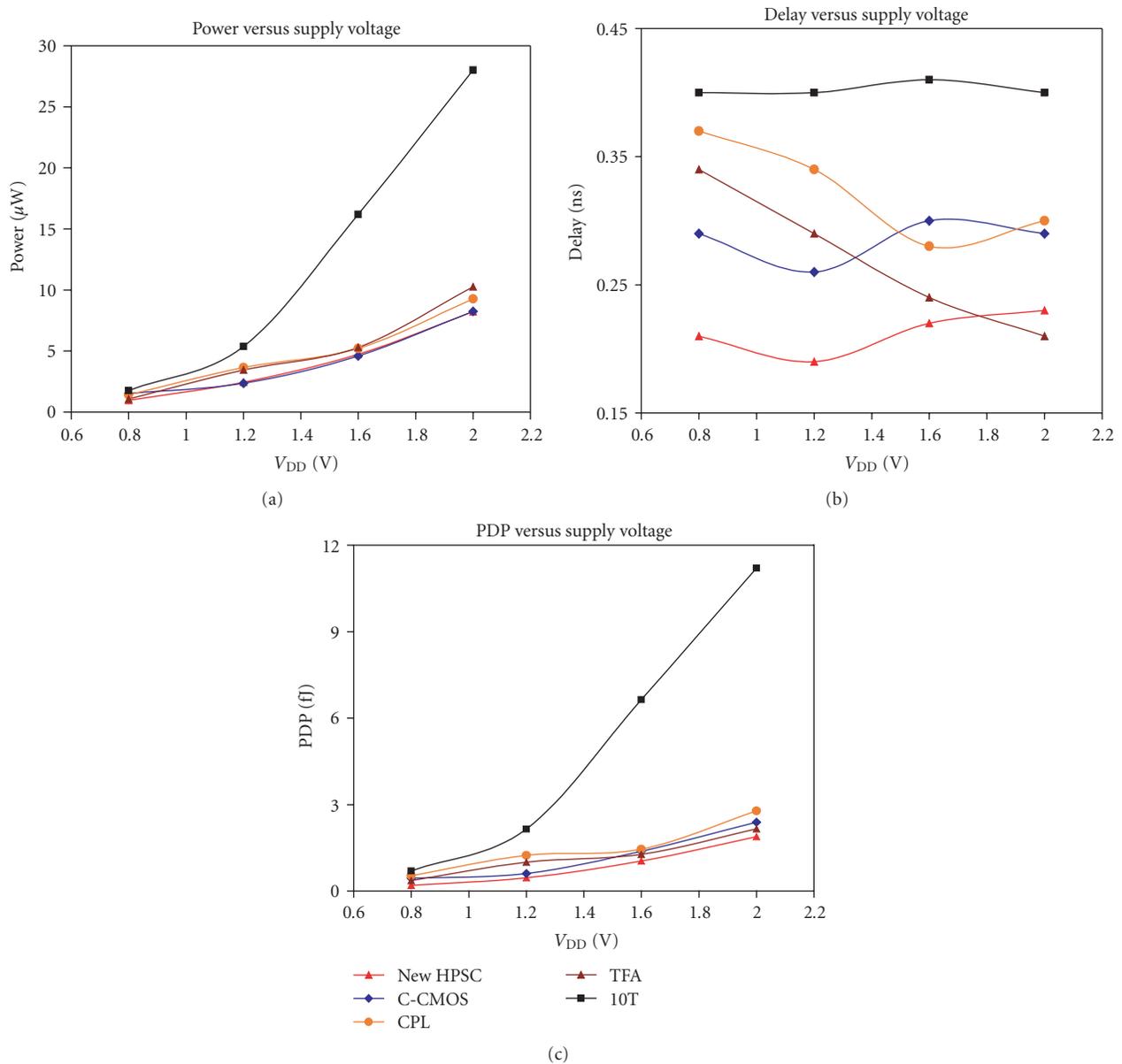


FIGURE 7: Power, Delay, and Power-Delay Product (PDP) comparison of full adders versus power supply using MDE and ADC algorithms.

of parameters must be performed carefully such that disorder is not presented in sizing accuracy. Hence, in this paper, some rules are offered to recognize transistors which are in similar positions and are grouped accordingly. Therefore, from the viewpoint of transistor sizing, the number of sizing operation steps will be reduced. These rules are stated as follows.

- (1) Series transistors of the same type have equal sizes to have the same conductivity. Else, the slower transistor will limit the speed of others; therefore, the delay is increased [2].
- (2) Parallel transistors of the same type have equal sizes to have the same conductivity. Else, the slower transistor will be determinant of the speed of the path, and delay will be increased [2].

- (3) Transistors of the same type which are in similar positions and are not series or parallel must have equal sizes. For instance, consider P1 and P2 transistors or N1 and N2 transistors of XOR/XNOR 6T circuit in Figure 1. Even though these transistors are not series or parallel, they still have to have the same size to equalize the resistance and drivability of the paths which they are located on. Thus, the delays of each path are the same.

3.2.3. Introducing Proposed Transistor Sizing Algorithm. As shown in [23], the transistor sizing for optimal performance is technology dependent. For a certain technology, the channel lengths of all transistors are fixed at the minimal feature size. So, the only variable to be optimized is the

TABLE 6: Power, Delay, and Power-Delay Product (PDP) comparison of full adders using Chang's algorithm.

V_{DD} (V)	0.8	1.0	1.2	1.4	1.6	1.8	2.0
Power (μ W)							
C-CMOS	0.92	1.51	2.33	3.35	4.68	6.32	8.20
CPL	1.32	2.18	3.38	4.95	6.91	9.34	12.23
TFA	0.92	1.54	2.41	3.53	4.93	6.70	8.83
TGA	0.96	1.60	2.47	3.62	5.03	6.78	8.90
New 14T	0.89	1.48	2.26	3.31	4.62	6.30	8.33
10T	1.78	2.98	5.14	8.17	13.10	20.65	27.17
New HPSC	0.95	1.58	2.40	3.44	4.77	6.44	8.27
Delay (ns)							
C-CMOS	0.39	0.25	0.18	0.14	0.12	0.11	0.10
CPL	0.36	0.24	0.19	0.16	0.14	0.13	0.12
TFA	0.40	0.28	0.21	0.18	0.16	0.14	0.13
TGA	0.41	0.25	0.18	0.15	0.13	0.11	0.11
New 14T	0.68	0.20	0.20	0.16	0.14	0.13	0.12
10T	4.41	1.51	0.39	0.17	0.10	0.11	0.08
New HPSC	0.29	0.17	0.12	0.12	0.09	0.08	0.09
Power-Delay product (fJ)							
C-CMOS	0.36	0.38	0.43	0.49	0.59	0.74	0.89
CPL	0.48	0.54	0.66	0.82	1.02	1.27	1.57
TFA	0.40	0.44	0.52	0.64	0.79	0.98	1.21
TGA	0.40	0.41	0.46	0.55	0.66	0.81	0.99
New 14T	0.67	0.34	0.29	0.55	0.67	0.84	1.05
10T	7.80	4.50	2.05	1.39	1.96	2.47	2.30
New HPSC	0.27	0.28	0.30	0.43	0.47	0.54	0.80

TABLE 7: Delay comparison (approximate) between MDE and ADC, Chang's, and proposed transistor sizing algorithm (SEA) of full adders with a supply voltage of 1.2 V.

Full adder	Algorithm		
	MDE & ADC	Chang's algorithm	SEA
C-CMOS	0.26	0.18	0.17
Improvement	34.62%	5.56%	
CPL	0.34	0.19	0.13
Improvement	61.76%	31.58%	
TFA	0.29	0.21	0.09
Improvement	68.97%	57.14%	
TGA	0.31	0.18	0.11
Improvement	64.52%	38.9%	
New 14T	0.3	0.2	0.11
Improvement	63.33%	45%	
10T	0.40	0.39	0.18
Improvement	55.00%	53.85%	
New HPSC	0.19	0.12	0.12
Improvement	36.84%	0%	
Average Improvement	55.01%	33.16%	

channel width of each transistor [2]. The proposed algorithm which is called SEA is described by the pseudocodes in Algorithm 1.

TABLE 8: PDP comparison (approximate) between MDE and ADC, and proposed transistor sizing algorithm (SEA) of full adders with a supply voltage of 1.2 V.

Full adder	Algorithm		
	MDE & ADC	SEA	Improvement
C-CMOS	0.61	0.40	34.43%
CPL	1.24	0.46	62.90%
TFA	1.00	0.24	76.00%
TGA	1.08	0.29	73.15%
New 14T	1.05	0.28	73.33%
10T	2.15	1.01	53.02%
New HPSC	0.46	0.31	32.61%
Average Improvement: 57.92%			

At the beginning, all the transistors of a circuit are categorized on the basis of the rules of grouping transistors. Then the channel width of transistors in group i is considered to be equal to W_i , where $i = 1, 2, \dots, n$. According to

$$W_i = k_i \times W^*, \quad (2)$$

and a coefficient k_i is assigned to each group where k_i is the sizing factor of transistors' widths in i th group. k_i is initiated with "1" where $i = 1, 2, \dots, n$.

The point is that, W^* or in other words W_i s are not permitted to be lesser than technology feature size under any

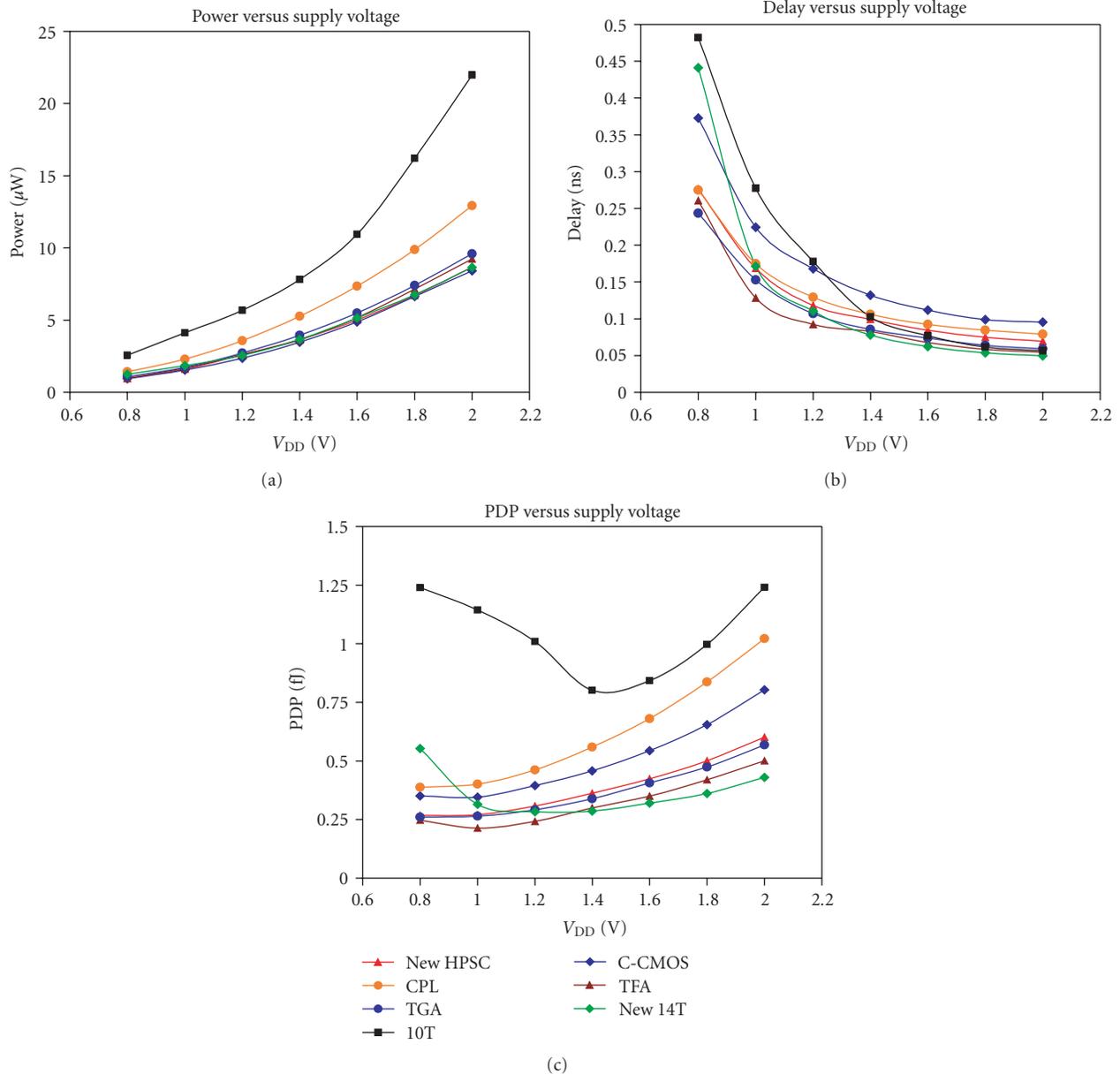


FIGURE 8: Power, Delay, and Power-Delay Product (PDP) comparison of full adders versus power supply using proposed transistor sizing algorithm (SEA).

circumstances. Now assume that all transistors have equal widths. To obtain the optimum width, W^* is swept once from minimum feature size to as much as its value in a two-dimensional space. The value of W^* at the optimum point is the optimum width or W_{opt} .

In each iteration, the sizing factor k_i is swept in an $n + 1$ dimensional space whereas the other k_j values where $j = 1, 2, \dots, n$ and $i \neq j$ are assumed as constants in order to find the optimum point. Then, in return, the value of k_i in optimum point is considered as a constant. This procedure is iterated n times for all k_i s. The iterations are executed since k_i where $i = 1, 2, \dots, n$ converges to a specified value.

In our proposed algorithm, the outermost for loop is executed n times where n is the number of transistor groups. The innermost for loop that implements the sweep operation is executed m times where m is the number of step sizes. Thus, the running time of SEA algorithm is $O(mn)$.

Attributes of SEA Algorithm. The transistor sizing operation in SEA algorithm tends towards a logical and feasible approach. This algorithm is also flexible in determining our target specification. It means that our goal can be one of the optimization parameters or the combination of them. In addition, the algorithm decreases the optimization

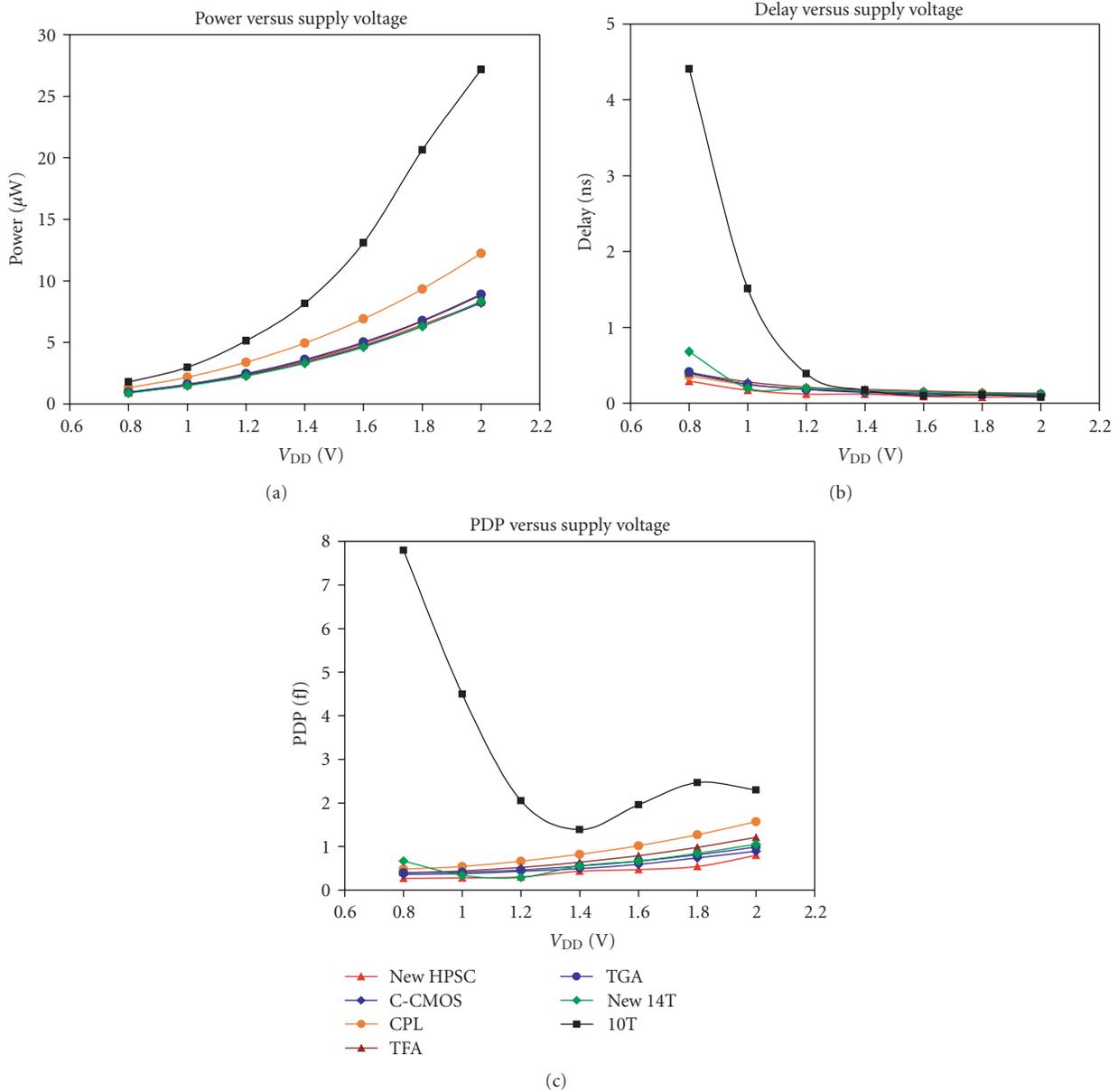


FIGURE 9: Power, Delay, and Power-Delay Product (PDP) comparison of full adders versus power supply using Chang’s algorithm.

parameters by the means of more exhaustive rules of grouping transistors which are in the same positions in symmetrical circuits. The decline rate of optimization parameters not only decreases the execution time of transistor sizing operation but also leads to lower dependency amongst those parameters. Therefore, SEA algorithm seeks the optimum point in a smaller state space which results in a greater probability that the goal parameter assumes its best-case value. It is obvious that the more symmetrical a circuit is, the greater the algorithm’s performance.

The behavior of some transistors of New HPSC circuit which is shown in Figure 1 is investigated in Figure 3, concerning their influence on target parameter. This evaluation

is done under the condition that the sizes of other transistors are fixed and just the size of intended transistor is changed and then the PDP values are measured. According to the rules of grouping transistors, P1 and P2 would be in the same group since they have the same position. P4 and P5 are grouped together as they are series. Considering Figure 3, the behavior of P1 and P2 transistors towards the goal parameter is similar. Thus, classification of these transistors in the same group would be reasonable.

Studying XOR/XNOR 6T circuit, P1 and P2 transistors and also N1 and N2 have the same position and could be classified in the same groups. The preliminary results for XOR/XNOR 6T which is shown in Figure 1 are listed in

TABLE 9: PDP comparison between proposed transistor sizing algorithm (SEA) and Chang’s algorithm of full adders with a supply voltage of 1.2 V.

Full Adder	Algorithm		Comparison
	Chang’s algorithm	SEA	
C-CMOS	0.43	0.40	7.50% Improvement
CPL	0.66	0.46	30.30% Improvement
TFA	0.52	0.24	53.85% Improvement
TGA	0.46	0.29	36.96% Improvement
New 14T	0.29	0.28	3.45% Improvement
10T	2.05	1.01	50.73% Improvement
New HPSC	0.30	0.31	3.33% Degradation
Average Improvement: 25.64%			

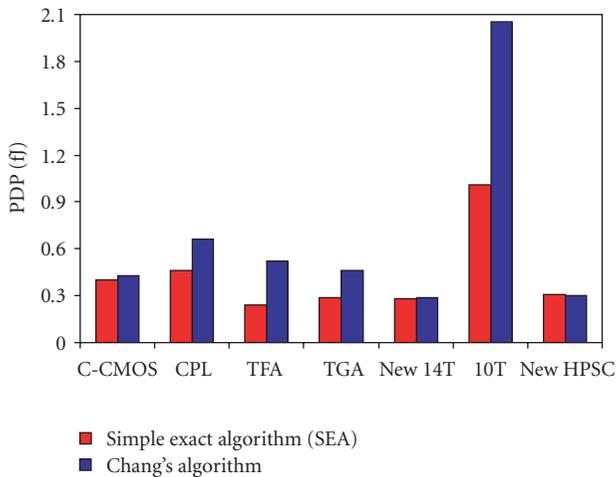


FIGURE 10: PDP comparison between Chang’s algorithm and proposed transistor sizing algorithm.

Table 1. The initial values for all the sizing factors have been considered to be “1”. The initial coefficients in Tables 2 and 3 are randomly selected.

A comparison of PDP results with achieved values in Tables 2 and 3 demonstrate that there is some negligible difference in PDPs in return for different initial values. Thus, the transistor sizing operation in the proposed algorithm is not determined by initial values of sizing factors.

The initial sweep of W^* which is done in the first step is one of the best achievements of SEA algorithm. In the beginning of transistor sizing with sweeping W which is done for all transistors identically, we can seek the optimized performance within a region closer to the absolute optimum point. Hence, the convergence trend will happen faster. This is because each transistor is assigned an allotment in

determining the value of target parameter. The transistors which have such important roles in determining the target parameter are more effective in sizing procedure.

In the other words, by sweeping all the transistors’ widths, the target parameter is impressed by vital transistors. The other transistors do not have such important roles in defining the circuit performance. It is therefore clear that various transistors have different effects on the goal parameter, according to their position in a circuit. This matter is demonstrated by simulations which are presented as convergence trends in Section 4. The big jumps in Figure 11 at the beginning of simulation which results mainly from sweeping W^* at the first run of the proposed algorithm, shows the convergent trend in SEA algorithm in comparison with Chang’s algorithm. This jumping towards the optimum point accelerates the simulation and is witnessed in most of the circuits.

4. Simulation Results and Analysis

4.1. Simulation Setup. In this section, the performance of the proposed algorithm is investigated through simulations under a variety of conditions. Full adder is a versatile and widely used building block in arithmetic circuits and the core element of complex arithmetic operations such as addition, multiplication, division, exponentiation, and so forth. [2, 5, 24–26]. Therefore, the seven full adder circuits C-CMOS, CPL, TFA, TGA, New 14T, 10T, and New HPSC of Figure 5 which are mostly low-power and high-speed circuits are used for simulation. These full adder circuits are all optimized using MDE, ADC, Chang’s algorithm and the proposed transistor sizing algorithms.

In order to experience all the possible transitions as inputs, an input test pattern with 56 transitions must be applied to a full adder circuit. An input transition may or may not result in a change at the output node. Even if there is no switching activity at the output node, some internal nodes may be switching. This switching activity results in some power dissipation. Thus, for an accurate result, all the possible input combinations are considered for all the test circuits [1]. Therefore, a group of input test patterns which offers all the 56 different transitions from one input combination to another are used as the input vectors for the full adder cell. Figure 4 shows the input stimulus, and Sum and C_{out} outputs.

The circuit performances of the seven full adders are evaluated in terms of average power consumption, worst-case delay and power-delay product for a range of supply voltages from 0.8 V to 2.0 V. For all the rise and fall transitions of outputs, the delay is measured from 50% of input voltage level to 50% of the output voltage level. The delay measurement in the simulations is shown by

$$t_{d,Sum} = \text{MAX}(t_{dRise,Sum}, t_{dFall,Sum}), \quad (3)$$

$$t_{d,Cout} = \text{MAX}(t_{dRise,Cout}, t_{dFall,Cout}), \quad (4)$$

$$t_{d,FullAdder} = \text{MAX}(t_{d,Sum}, t_{d,Cout}). \quad (5)$$

TABLE 10: Comparison of timing complexity and target parameter in optimization between transistor sizing algorithms.

Algorithm	Initial time	Time complexity	Target parameter
MDE	—	$O(N \cdot K^2 \cdot FO ^2)$ N : Number of gates, K : Number of different sizes of a gate $ FO $: Maximum fan-out of a gate	Delay
ADC	—	$O(n)$ N : Number of transistors	Area-Delay Product
Chang's Algorithm	—	$O(mn)$ m : Number of steps, n : Number of transistors' groups	Any of circuit specifications including power consumption, delay, power-delay product, chip area, or their combination
SEA	$T_{\text{int}} = (n + 1)t_{\text{swp}} < \varepsilon$ n : Number of transistors, t_{swp} : the time needed for a sweep	$O(mn)$ m : Number of steps, n : Number of transistors' groups	

From (5) it is evident that the full adder delay is the larger delay value of Sum and C_{out} outputs. The power-delay product quantity is defined as

$$\text{PDP}_{\text{FullAdder}} = P_{\text{average}} \times t_{d,\text{FullAdder}}. \quad (6)$$

All the circuits are designed using the TSMC 0.13- μm CMOS technology and were simulated using the BSIM3v3 model with Level 49 technology file. The threshold voltages of the PMOS and NMOS transistors are approximately 0.33 and 0.35 V, respectively. Simulations are carried out using Star HSPICE. Environment temperature has been set to 27°C.

The generic test bench used to simulate the full adders being compared is shown in Figure 6. This simulation environment has been commonly used to compare the performance of the full adders in [1, 7–9, 25]. To simulate a real environment, input buffers for all inputs of the test circuit are used to generate a real waveform and output buffers for both outputs are used to generate output load.

4.2. Simulation Results. In order to have a fair and exact comparison of the proposed algorithm with the three previously reported algorithms, simulation conditions for these algorithms were considered identical as far as possible. Since C_{load} capacitance is needed for MDE and ADC algorithms, the equivalent capacitance of output buffers in single test bench is calculated (about 32 fF) and has been considered as C_{load} .

The simulation results for the MDE and ADC algorithms under supply voltages 0.8 V, 1.2 V, 1.6 V, and 2.0 V for five full adder circuits C-CMOS, CPL, TFA, 10T, and New HPSC are listed in Table 4. The values of power, delay, and power-delay product (PDP) obtained for considered values of V_{DD} (0.8–2.0 V) for these full adders are shown in Figure 7.

The seven full adder circuits of Figure 5 are all simulated to achieve the optimum power-delay product (PDP)

using Chang's algorithm and the proposed transistor sizing algorithms. Optimization of the transistor sizing is carried out at seven different voltages, 0.8 V, 1.0 V, 1.2 V, 1.4 V, 1.6 V, 1.8 V, and 2.0 V. The step size of the subsequent iterations of these algorithms is set to 0.1 μm . Thus, the final transistor sizes have the precision of 77% of the channel length, which is 0.13- μm for our targeted technology. The final transistor widths for the New HPSC full adder cell which is optimized by both of the algorithms at different supply voltages are listed in Table 11.

The power, delay and power-delay product at supply voltage ranges from 0.8 V to 2.0 V of these full adder cells are listed in Tables 5 and 6, respectively for comparison. To make the comparison easier, the simulation results are shown in Figures 8 and 9.

4.3. Comparison of Algorithms. As discussed earlier, MDE and ADC algorithms are based on logical effort and are classified as arithmetic transistor sizing algorithms. Therefore, the transistor sizing approach in these algorithms is different from Chang's algorithm and the proposed algorithm which are both heuristic. Once the delay is optimized through the use of MDE algorithm, ADC algorithm uses the delay measured and optimizes area-delay product during some calculations. Hence, the delay performance is the target parameter in both of these algorithms but the power is not optimized and we are allowed to measure power for a given delay.

Thus, MDE and ADC algorithms are not particularly optimizing the power-delay product (PDP) performance, but in this paper, PDP is the target parameter for Chang's and the proposed algorithms. In regard to the nature of MDE and ADC algorithms, comparison of these algorithms with Chang's and the proposed algorithms which are particularly optimizing PDP, is not naturally exact. Generally, discussion of these four algorithms has two aspects: 1-Delay 2-PDP.

TABLE 11: Transistor sizes (μm) of New HPSC full adder optimized for PDP with different power supplies.

Proposed transistor sizing algorithm (SEA)													
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P_{inverter}	
0.8 V	0.22	0.22	0.13	0.27	0.27	0.13	0.13	0.14	0.14	0.13	0.13	0.14	
1.0 V	0.39	0.39	0.13	0.30	0.30	0.13	0.13	0.14	0.14	0.13	0.13	0.14	
1.2 V	0.56	0.56	0.15	0.39	0.39	0.17	0.17	0.19	0.19	0.15	0.15	0.15	
1.4 V	0.63	0.63	0.13	0.36	0.36	0.15	0.15	0.17	0.17	0.14	0.14	0.14	
1.6 V	0.59	0.59	0.13	0.36	0.36	0.13	0.13	0.18	0.18	0.14	0.14	0.17	
1.8 V	0.61	0.61	0.14	0.31	0.31	0.15	0.15	0.20	0.20	0.17	0.17	0.19	
2.0 V	0.63	0.63	0.14	0.36	0.36	0.15	0.15	0.20	0.20	0.17	0.17	0.19	
	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N_{inverter}	
0.8 V	0.13	0.13	0.14	0.20	0.20	0.13	0.13	0.13	0.13	0.13	0.13	0.13	
1.0 V	0.14	0.14	0.14	0.20	0.20	0.13	0.13	0.13	0.13	0.13	0.13	0.13	
1.2 V	0.27	0.27	0.19	0.20	0.20	0.17	0.17	0.17	0.17	0.15	0.15	0.17	
1.4 V	0.24	0.24	0.17	0.15	0.15	0.15	0.15	0.15	0.15	0.14	0.14	0.15	
1.6 V	0.29	0.29	0.15	0.17	0.17	0.13	0.13	0.17	0.17	0.13	0.13	0.15	
1.8 V	0.22	0.22	0.15	0.19	0.19	0.15	0.15	0.20	0.20	0.17	0.17	0.17	
2.0 V	0.20	0.20	0.14	0.17	0.17	0.14	0.14	0.20	0.20	0.13	0.13	0.13	
Chang's transistor sizing algorithm													
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	$P_{\text{inverter1}}$	$P_{\text{inverter2}}$
0.8 V	0.16	0.16	0.13	0.27	0.27	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
1.0 V	0.20	0.13	0.13	0.23	0.23	0.16	0.13	0.20	0.2	0.13	0.13	0.14	0.16
1.2 V	0.32	0.13	0.13	0.25	0.25	0.16	0.13	0.16	0.16	0.13	0.13	0.13	0.16
1.4 V	0.20	0.13	0.13	0.23	0.23	0.16	0.13	0.20	0.2	0.13	0.13	0.14	0.16
1.6 V	0.52	0.17	0.13	0.13	0.13	0.13	0.13	0.17	0.17	0.13	0.13	0.13	0.14
1.8 V	0.52	0.19	0.13	0.13	0.13	0.14	0.13	0.19	0.19	0.13	0.13	0.16	0.17
2.0 V	0.52	0.17	0.13	0.13	0.13	0.13	0.13	0.17	0.17	0.13	0.13	0.13	0.14
	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	$N_{\text{inverter1}}$	$N_{\text{inverter2}}$
0.8 V	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
1.0 V	0.13	0.13	0.14	0.13	0.13	0.18	0.13	0.18	0.18	0.13	0.13	0.16	0.14
1.2 V	0.23	0.14	0.13	0.16	0.16	0.13	0.13	0.14	0.14	0.13	0.13	0.14	0.13
1.4 V	0.13	0.13	0.14	0.13	0.13	0.18	0.13	0.18	0.18	0.13	0.13	0.16	0.14
1.6 V	0.25	0.13	0.13	0.13	0.13	0.13	0.13	0.14	0.14	0.13	0.13	0.25	0.13
1.8 V	0.25	0.25	0.13	0.26	0.26	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.14
2.0 V	0.25	0.13	0.13	0.13	0.13	0.13	0.13	0.14	0.14	0.13	0.13	0.25	0.13

Table 7 shows the comparison of the proposed algorithm with MDE, ADC, and Chang's algorithms in terms of the delay values obtained at 1.2 V V_{DD} for five full adder circuits C-CMOS, CPL, TFA, 10T, and New HPSC. This table shows that although MDE algorithm optimizes the delay performance only and then ADC algorithm is used to optimize the area-delay product, the delay measured for full adder circuits using the proposed algorithm is on average 55.01% lesser compared to MDE and ADC algorithms. This result is noticeable due to the fact that the proposed algorithm optimizes PDP and the circuits are not optimized for the delay performance only. It can also be observed from Table 7 that the average improvement in delay is 33.16% using the proposed algorithm as compared to Chang's algorithm.

Table 8 shows the simulation results from the MDE, ADC and the proposed algorithm comparison regarding power-delay product. As it can be seen in the table, there is 57.92% improvement in PDP on an average using the proposed

algorithm. It should be noticed that although comparison of these algorithms is not fair but it is useful to determine the approach of these algorithms to optimize the target parameter which is PDP.

An exact comparison is being made between the proposed algorithm and Chang's algorithm due to the fact that both of these algorithms are used to optimize the power-delay product of the full adder circuits. Table 9 tabulates the values of the power-delay product at 1.2 V V_{DD} in order to compare circuit optimization using these algorithms. Compared to Chang's algorithm, the PDP characteristics are improved by 25.64% on average using the proposed algorithm. The optimization results for the seven full adder circuits are also shown in Figure 10.

Figure 11 shows the convergence process of obtaining PDP for some of the full adder circuits at 1.2 V supply voltage using Chang's algorithm and the proposed algorithm.

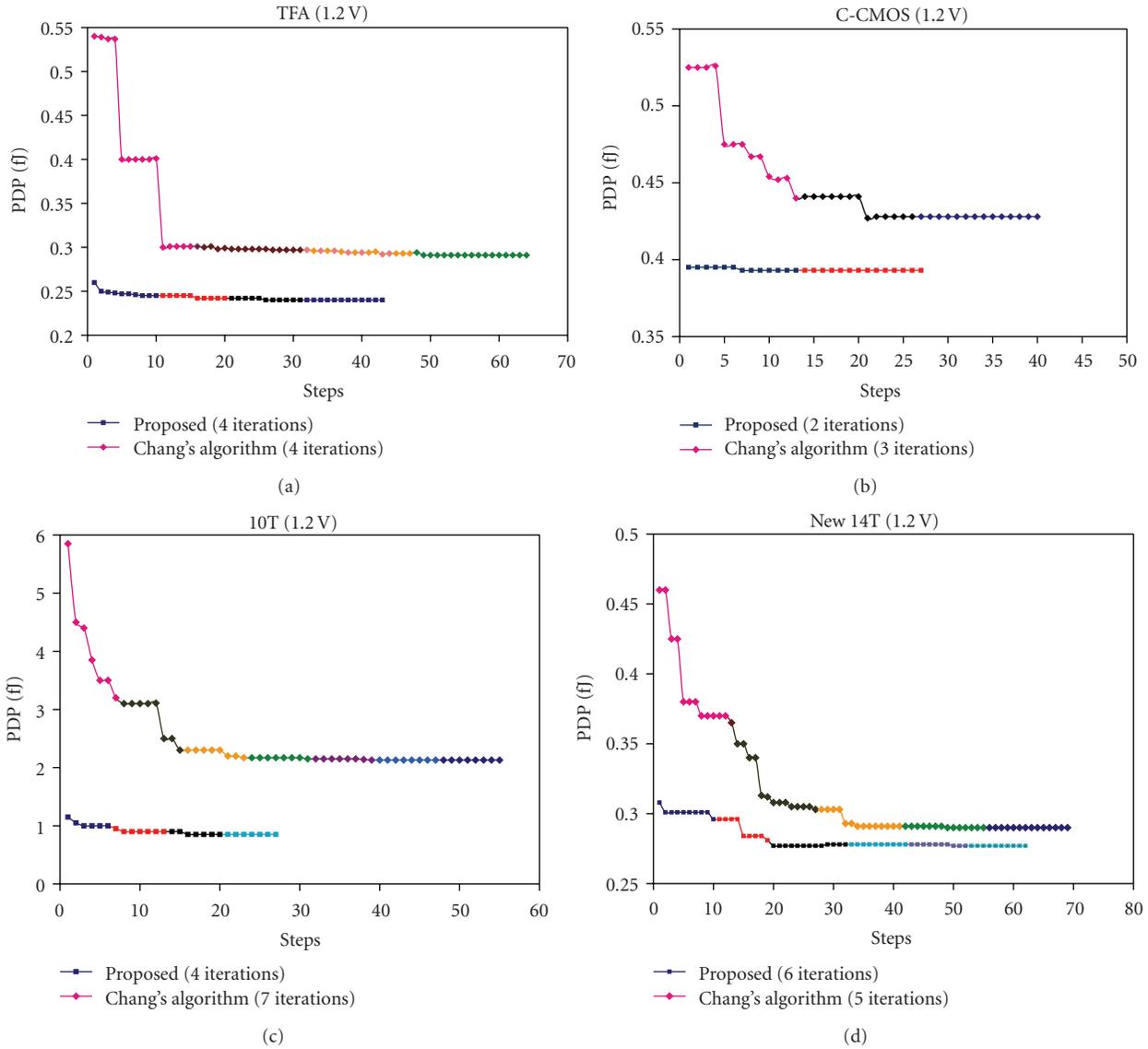


FIGURE 11: Convergence trend comparisons between Chang's algorithm and proposed transistor sizing algorithm of C-CMOS, TFA, New 14T, and 10T full adders with a supply voltage of 1.2 V.

MDE and ADC algorithms which are based on logical effort cannot participate in this comparison.

Considering Figure 11, it is apparent that the convergence process of the proposed algorithm is faster than Chang's algorithm since the proposed algorithm performs fewer iterations (each color presents an iteration) to obtain the optimal PDP. As mentioned earlier, this is owing to more exhaustive rules of grouping transistors, considering the transistors with identical positions in a circuit and also sweeping W^* at the first run of the algorithm.

Therefore, the number of transistor groups using the proposed algorithm is lesser than Chang's algorithm. Thus, the entire process of obtaining a better performance is faster after application of our proposed algorithm. Second, as it can be seen in Figure 11, the target parameter obtained in the

circuits using the proposed algorithm is almost lesser than that of Chang's algorithm. Hence, the proposed algorithm is more successful in providing much lower PDP.

4.4. SEA Algorithm in Comparison with Chang's Algorithm at a Glance. In Chang's algorithm, there is no logical trend for initial values of transistor sizes. The designer experimentally assigns proximate values to transistors according to their kind and position in a circuit. In addition, there is no specific rule for the order of sizing the transistors. Therefore, the mentioned parameters, that is, initial values of transistor sizes and the order of selecting them for sizing operation could be variable which result in different optimization rate of the target parameter. In some cases, this difference is negligible but in others it is considerable.

SEA algorithm performs better than Chang's algorithm in both cases. These characteristics show briefly the advantages of SEA algorithm over Chang's algorithm.

- (1) In SEA algorithm, sweeping W^* at the beginning of sizing operation leads to the determination of more sensitive transistors which speeds up the convergence procedure (First sizing run in Figure 10). This strategy guides the optimization procedure to the advantage of more sensitive transistors.
- (2) In SEA algorithm, the initial values of transistor sizes are not accidentally chosen. This logical trend has been obtained with initial sweeping of W^* either.
- (3) The new rule of grouping transistors in the same position results in lesser optimization parameters and execution time, more reliability and an increase in the optimization rate in many symmetrical circuits.

5. Conclusion

In this paper, we proposed a novel transistor sizing algorithm to optimize low-power high-speed arithmetic circuits, known as the Simple Exact transistor sizing Algorithm (SEA). The proposed algorithm has been compared with MDE, ADC and Chang's algorithm, whereby in comparison it has been shown that its overall performance is significantly superior. The simplicity of the transistor sizing approach using the proposed algorithm leads to a reduction within its sizing execution time. Moreover, using the SEA algorithm, reliability will increase since the order and initial values of sizing factors of transistors do not affect the final value obtained as the target parameter. This advantage lies in the properties of the symmetrical circuits.

Faster convergence trend and achieving a better target parameter by the proposed algorithm have been shown to outperform Chang's algorithm. Comparison being made between the proposed algorithm and the other three in terms of delay characteristics exhibits on an average 55.01% improvement over MDE and ADC algorithms and 33.16% over Chang's algorithm. Comparison of the proposed algorithm with MDE and ADC in terms of PDP is not naturally exact since these algorithms are not designed to optimize Power-Delay Product (PDP) and power is measured for the optimum delay. Despite the approximate comparison being made, there is almost 57.92% saving in PDP which proves that the proposed algorithm has preference over Chang's algorithm.

Using Chang's and the proposed algorithm to optimize PDP for seven full adder circuits, simulation results show 25.64% improvement on an average which depicts the efficiency and ability of this algorithm in optimizing the target parameter.

Comparison of the algorithms in terms of time complexity and flexibility in choosing the target parameter can be observed from Table 10. There are some reasons for superiority of SEA algorithm with the others in running time: (1) grouping transistors will reduce the running time of each iteration. (2) within decreasing the number of transistor

groups and reducing the dependency of parameters to each other, the speed of convergence trend will improve; (3) the first sweep of W of all transistors at the beginning of algorithm will cause a great jump in finding PDP in optimization operation (Figure 11). Timing complexity of SEA and Chang's is nearly the same but the running time is remarkably improved.

Although the proposed transistor sizing algorithm (SEA) has been used in this paper to minimize the PDP performance for some full adders, it is highly flexible in determining the target parameter which the transistor sizing approach is performed to optimize. This parameter could be one of the circuit performances such as power dissipation, delay, area or a combination of them.

Execution time of the proposed algorithm is lower than MDE but higher than ADC. The rule of finding transistors with identical positions can be defined and used in symmetric circuits such as balanced XOR/XNOR circuits and results in fewer number of transistor groups and presents a smaller value for n . Thus, the execution time for the proposed algorithm could be lower than Chang's algorithm. The proposed algorithm could be used for other VLSI circuits and could be evaluated in terms of compatibility for certain parameters such as reliability, transistor grouping rules.

References

- [1] S. Goel, A. Kumar, and M. A. Bayoumi, "Design of robust, energy-efficient full adders for deep-submicrometer design using hybrid-CMOS logic style," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 12, pp. 1309–1321, 2006.
- [2] C.-H. Chang, J. Gu, and M. Zhang, "A review of 0.18- μm full adder performances for tree structured arithmetic circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 6, pp. 686–694, 2005.
- [3] O. Kavehie, K. Navi, T. Nikoubin, and M. Rouholamini, "A novel DCVS tree reduction algorithm," in *Proceedings of the IEEE International Conference on Integrated Circuit Design and Technology (ICICDT '06)*, Padua, Italy, May 2006.
- [4] D. Radhakrishnan, "Low-voltage low-power CMOS full adder," *IEE Proceedings: Circuits, Devices and Systems*, vol. 148, no. 1, pp. 19–24, 2001.
- [5] U. Ko, P. Balsara, and W. Lee, "Low-power design techniques for high-performance CMOS adders," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 3, no. 2, pp. 327–333, 1995.
- [6] T. Nikoubin, S. Pouri, P. Bahrebar, and K. Navi, "A new transistor sizing algorithm for balanced XOR/XNOR circuits," in *Proceedings of the 12th Annual International CSI Computer Conference*, Tehran, Iran, February 2006.
- [7] A. M. Shams and M. A. Bayoumi, "A structured approach for designing low power adders," in *Proceedings of the Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 757–761, Pacific Grove, Calif, USA, November 1998.
- [8] M. Aguirre and M. Linares, "An alternative logic approach to implement high-speed low-power full adder cells," in *Proceedings of the 18th Annual Symposium on Integrated Circuits and Systems Design*, pp. 166–171, Florianopolis, Brazil, September 2005.

- [9] K. Navi, O. Kavehie, M. Rouholamini, A. Sahafi, and S. Mehrabi, "A novel CMOS full adder," in *Proceedings of the 20th IEEE International Conference on VLSI Design*, pp. 303–307, January 2007.
- [10] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits (A Design Perspective)*, Prentice-Hall, Upper Saddle River, NJ, USA, 2nd edition, 2002.
- [11] S. S. Sapatnekar and W. Chuang, "Power-delay optimizations in gate sizing," *ACM Transactions on Design Automation of Electronic Systems*, vol. 5, no. 1, pp. 98–114, 2000.
- [12] M. R. Berkelaar and J. A. Jess, "Gate sizing in MOS digital circuits with linear programming," in *Proceedings of the European Design Automation Conference*, pp. 217–221, 1990.
- [13] M. R. C. M. Berkelaar, P. H. W. Burman, and J. A. G. Jess, "Computing the entire active area/power consumption versus delay trade-off curve for gate sizing with a piecewise linear simulator," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 474–480, 1994.
- [14] J. P. Fishburn and A. E. Dunlop, "TILOS: a posynomial programming approach to transistor sizing," in *Proceedings of the International Conference on Computer Aided Design (ICCAD '85)*, 1985.
- [15] M. Haghparast and K. Navi, "Design of a novel fault tolerant reversible 12 full adder for nanotechnology based systems," *World Applied Sciences Journal*, pp. 114–118, 2008.
- [16] K. Navi, M. Maeen, V. Foroutan, S. Timarchi, and O. Kavehei, "A novel low-power full-adder cell for low voltage," *Integration, the VLSI Journal*, vol. 42, no. 4, pp. 457–467, 2009.
- [17] A. E. Ruehli, P. K. Wolff Sr., and G. Goertzel, "Analytical power timing optimization technique for digital system," in *Proceedings of the 14th Design Automation Conference*, pp. 142–146, June 1977.
- [18] K. S. Hedlund, "Electrical optimization of PALs," in *Proceedings of the 22nd Design Automation Conference*, pp. 681–687, June 1985.
- [19] W. T. Nye, A. L. Sangiovanni-Vincenteli, J. P. Spoto, and A. L. Tits, "DELIGHT-SPIICE: an optimization-based system for the design of IC," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 233–238, May 1983.
- [20] M. D. Matson, *Macromodeling and optimization of digital MOS VLSI circuits*, Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, Mass, USA, January 1985.
- [21] E. Polak, *Computational Method in Optimization*, Academic, New York, NY, USA, 1971.
- [22] S. K. Karandikar and S. S. Sapatnekar, "Fast comparisons of circuit implementations," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 12, pp. 1329–1339, 2005.
- [23] W. Donath, P. Kudva, L. Stok, et al., "Transformational placement and synthesis," in *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 194–201, March 2000.
- [24] M. Sayed and W. Badawy, "Performance analysis of single-bit full adder cells using 0.18, 0.25, and 0.35 μm CMOS technologies," in *Proceedings of the 35th IEEE International Symposium on Circuits and Systems*, vol. 3, pp. 559–562, 2002.
- [25] O. Kwon, K. Nowka, and E. E. Swartzlander, "16-bit \times 16-bit MAC design using fast 5:2 compressors," in *Proceedings of the International Conference on Application-Specific Systems, Architectures and Processors*, pp. 235–243, July 2000.
- [26] A. M. Shams, T. K. Darwish, and M. A. Bayoumi, "Performance analysis of low-power 1-bit CMOS full adder cells," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 1, pp. 20–29, 2002.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

