DIGITAL INTEGRATED CIRCUITS DESIGN BASED GENETIC ALGORITHM

M.Sc. Ryadh A. Kadhim
University of Technology
Department of Electromechanical Engineering
Baghdad - Iraq

Abstract

Genetic algorithms (GAs) are search and optimization techniques that are used in engineering applications like systems optimization equation solving and electronic circuits design and synthesis. genetic algorithms are used in the design and synthesis of fixed topology electronic circuits. Two types of digital integrated circuits are considered, CMOS NAND circuit and CMOS NOR circuit.

Several gate circuits are designed and synthesized to obtain the best value of (W/ L) s for the transistors in the circuits. These aspect ratios minimize the required circuit total chip area while still satisfying the logic function of the circuit with minimum deviation from the ideal characteristics. The resultant gate structures have less aspect ratio than those resulting from the traditional design techniques.

الخلاصـــــة

الخوارزمية الجينية هي عبارة عن تقنيات بحث وإيجاد الأفضل، تستخدم في كثير من التطبيقات الهندسية مثل عمليات أيجاد الأفضل، حل المعادلات الرياضية وتصميم وتركيب الدوائر الالكترونية. استخدمت الخوارزمية الجينية في تصميم وتركيب الدوائر الإلكترونية و تم دراسة نوعان من الدوائر الإلكترونية: دائرة CMOS NAND ودائرة CMOC NOR.

تم تصميم و تركيب عدة دوائر بوابات للحصول على أفضل قيم لـ (W/L) للترانزستورات في الدوائر. إن نسب الاستطالة هذه تقلل مساحة الدائرة الكلية المطلوبة على الرقاقة بينما تبقى تحقق الدالة المنطقية المطلوبة من الدائرة بأقل انحراف عن الخصائص المثالية. إن تراكيب البوابات الناتجة تمتلك نسب استطالة اقل من تلك الناتجة من أساليب التصميم التقليدية.

1. Introduction

In this paper, the design and synthesis of a class of digital integrated circuit using Genetic Algorithms are presented. The design includes the design and realization of CMOS NAND and NOR circuits.

The design involves minimizing certain parameters and satisfies the required specified characteristic. The realized circuits are simulated using MATLAB, and compared these results with standard design [1].

The principle the Genetic Algorithm program should obtain the best values of the four aspect ratio $(W/L)_1$ and $(W/L)_4$. since the best values of the aspect ratio mean minimum chip area required for fabrication.

2. Genetic Algorithm in Electronic

Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics. In GAs, the search is divided into generations each generation consists of a number of individuals and each individual describes the problem parameters that should be optimized [2]. The main advantage of GAs over traditional optimization techniques is that they are used in situations in which no analytical formulation can be found for the mathematical description of the problem [3,4].

One of the relatively new and important applications for Genetic Algorithms GAs is the design and synthesis of electronic circuits [5]. This application suits GAs very much because designing electronic circuits is a problem that involves optimizing a very large number of parameters. For example to design a filter or an Op Amp we need to specify the values of tens of elements like resistor, capacitors, the parameters of the transistors and diodes in addition to determining the way these elements are connected together.

To reduce the cost of the design, one has to reduce the number of components (in the case of discrete elements implementation) and their (in the case of implementing the circuit using ICs), and at the same time obtain good performance regarding the frequency and phase response for the filter and the gain, bandwidth, slew rate, and other parameters for the Op Amp [6], response times, fan in, and fan out in the case of digital circuits.

Since solving this problem using traditional maximizing and minimizing techniques is almost impossible, GAs arises as the best solution. The use of genetic algorithms in electronics is very important now as the electronic industry is evolving and almost all circuit design becoming automated.

The main reason why Genetic Algorithms are very successful in electronics is that most electronic system design tasks deal with nonlinear equations that usually not suitable to be dealt with using traditional techniques. Taking the transistor for example, each transistor has a nonlinear relation between voltage and current and usually this relation is different in regions of operation [7].

For example, the equation is different in the active region from the one in the saturation region. The traditional design techniques will not be systematic in these cases and will rely mainly on trial and error principle.

The main issue in GAs in electronic circuit design is that how the chromosome will describe the circuit. The design starts by defining the parameters for encoding in the chromosomes. These parameters determine the circuit behavior.

For example a certain design contains four aspect ratio then the chromosome should include the values of all the four parameters. The number of bits used for each parameter is determined based on the acceptable range for that parameter and the resolution, i.e. the increment between two consecutive values.

3. Chromosome representation

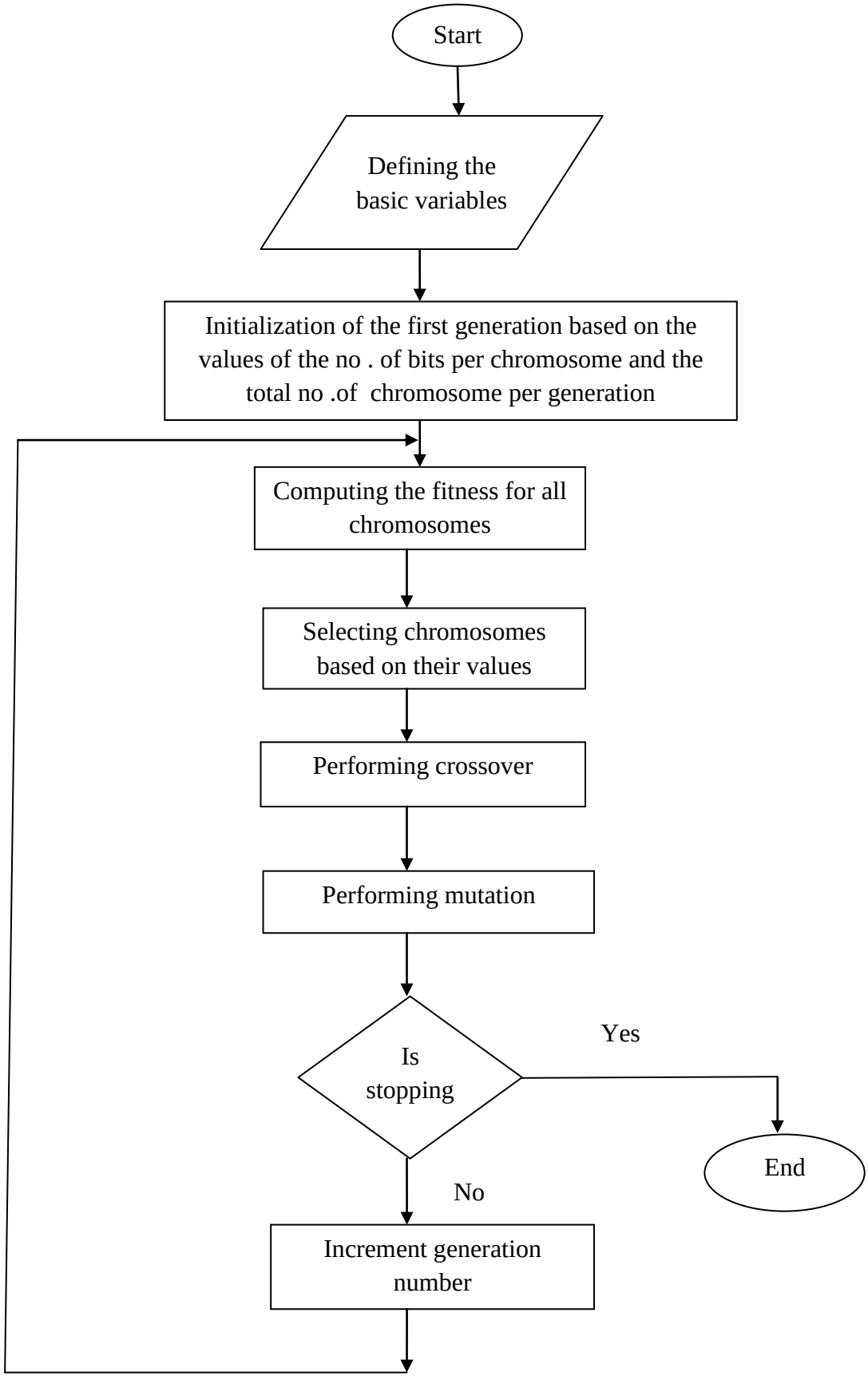
In Genetic algorithms (GAs), the search is divided in to generations each generation consists of a number of chromosomes and each chromosomes describes the problem parameters that should be optimized. Theses individuals are called chromosomes, a tem borrowed from biological genetics [8] .

The chromosomes describes the problem parameters by a specific mapping designed by the programmer. usually the chromosome encodes the information of these parameters using binary encoding (it is most common way of encoding due to simplicity in implementation and ability to encode all types of parameters). An example of how the chromosome encodes parameters is how in figure below.

1	0	1	1	1	0	0	1	0	0	.....	1	0	1
Parameter 1					Parameter 2						Parameter n		

Notice that the parameters should not have the same number of bits each.

The program structure is explained in the following flow chart.



4. Design and Simulation of Digital Integrated Circuit Based on Genetic Algorithm

Genetic Algorithm is employed here to minimize the chip area required for fabrication. A CMOS logic NAND circuit in fig.(1) is taken as an illustrative example. It consist of four transistors(2n and 2p). Therefore the Genetic Algorithm program should obtain the best values of the four aspect ratio $(W/L)_1$ and $(W/L)_4$. The minimization of these aspect ratio is very important factor for reducing the chip area. The total chip area which is computed as in equation (1) below [8].

$$Total\ chip\ Area = \sum_{i=1}^{n} W_i L_i \qquad (1)$$

Where: W_i and L_i are the ith width and length of the transistors respectively and n is the total number of transistors in the circuit. Since L is considered unity (the minimum length that can be fabricated on the chip), the value of W/L is usually considered the same as W × L [8] and the area is:

$$Area = \sum_{i=1}^{n} \frac{W_i}{L_i} \qquad (2)$$

This area is measured in units of L^2. For example if L is $1\mu m$, then the area is measured in unit of μm^2. Since the minimization of the total area is given by:

$$Area = (\frac{W}{L})_1 + (\frac{W}{L})_2 + (\frac{W}{L})_3 + (\frac{W}{L})_4 \qquad (3)$$

where:

$(\frac{W}{L})i$ is the aspect ratio of the ith transistor.

And the correct operation as a NAND circuit is the two factors that were used to compute the fitness of each chromosome.

The correct operation is measured using the voltage error factor. This voltage error is the sum of error between ideal output and real output for various input values for V_A and V_B. The following equation was used to compute the fitness [8].

$$fitness = \frac{1}{voltage\ \text{error}} + \frac{1}{area} \qquad (4)$$

The fitness computation function will first decode the chromosomes to get the parameters to be optimized. Because our goal is minimization of total chip area, these parameters are only the set of aspect ratio of the transistors because all other technological parameters in the transistors are fixed like $\mu_n, V_T, \varepsilon, \ldots$etc.

The ranges of possible values for (W/L) are taken to be 0.1to100 with a step size of 0.1. After decoding the values of (W_i / L_i), these values are normalized so that the minimum is 1. The normalization is done by dividing the set of (W_i / L_i) by the minimum aspect ratio.

The range described above needs 10 bits for each (W/L). So the chromosome length will be 10 × n. where n is the number of transistors in the circuit.

Table (1) shows the last generation value of these parameters.

Table (1) the CMOS logic NAND best values

parameter	Values of GAs design	Values of Standard design
$(W/L)_1$	1.0345	2.5
$(W/L)_2$	1	2.5
$(W/L)_3$	1.3009	2.5
$(W/L)_4$	1.348	2.5
Area	4.6834	10
Voltage error	3.8716	8.4351

From the above table it can be seen that the realized CMOS NAND circuit has minimum area and voltage error compared with that obtained by the design found in [1].

The simulated NAND circuit response shown in fig.(2) is obtained using MATLAB package.

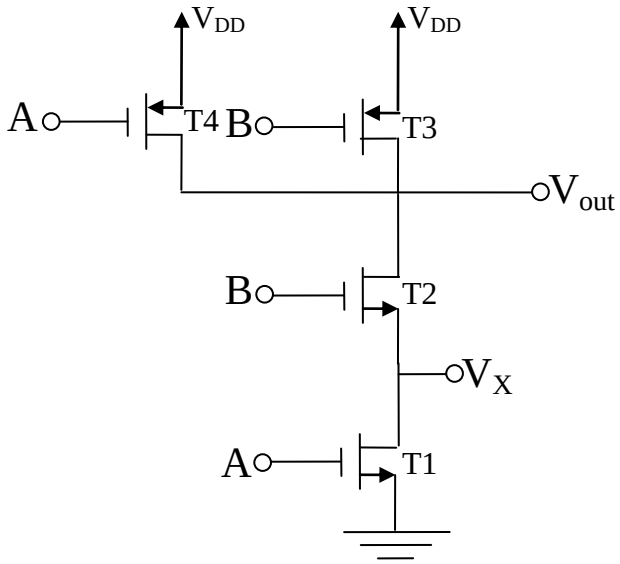


Fig. (1) A CMOS NAND Circuit

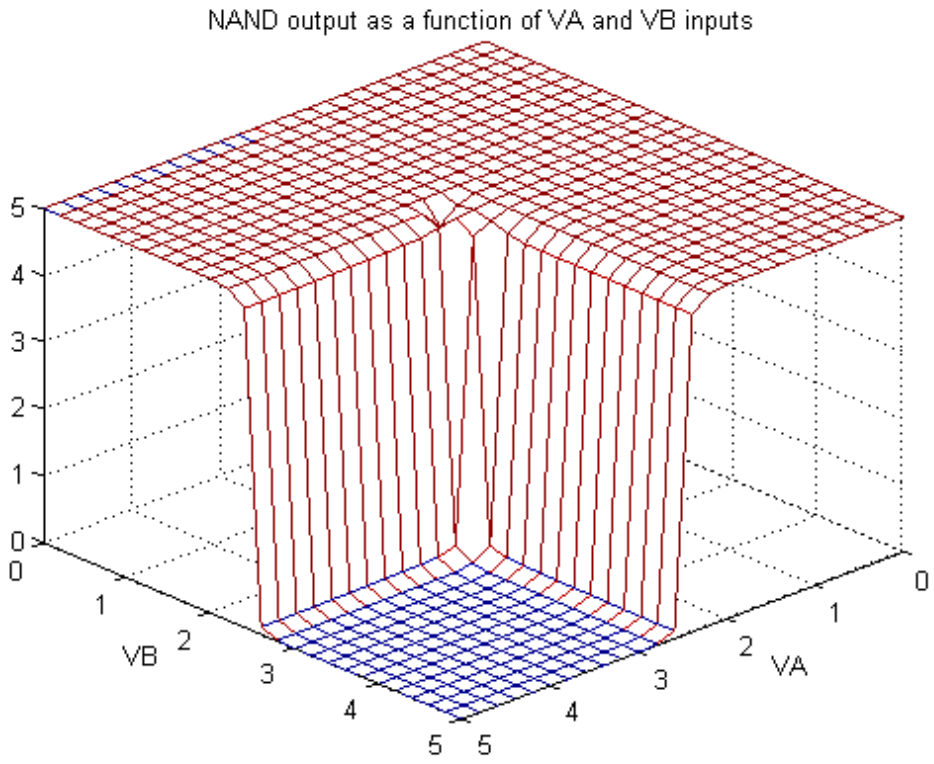


Fig. (2) The NAND Circuit Response to input values of V_A and V_B.

Another example is also investigated a CMOS logic NOR circuit type is designed using the same procedure.

It also consists of four transistors (2n and 2p) as shown in fig.(3).

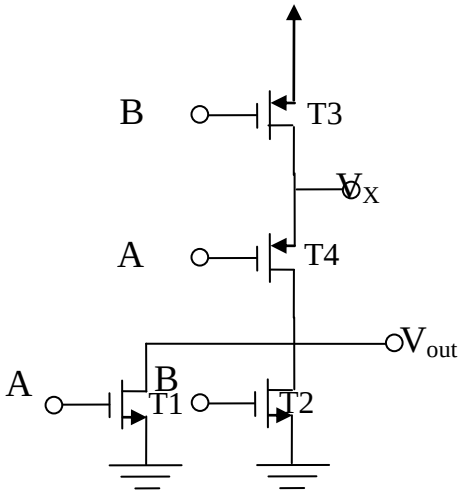


Fig. (3) A CMOS NOR Circuit

Table (2) shows the last generation values of these parameters.

Table (2) the CMOS logic NOR best values

parameter	Values of AGs design	Values of Standard design
$(W/L)_1$	1	2.5
$(W/L)_2$	1.0018	2.5
$(W/L)_3$	1.2921	5
$(W/L)_4$	1.0412	5
Area	4.3351	15
Voltage error	11.5829	13.73

From the above table it can be seen that the realized CMOC NOR circuit has minimum area and voltage error compared with that obtained by the design found in [1]. Beside that the NOR circuit response is preserved as shown in fig.(4).

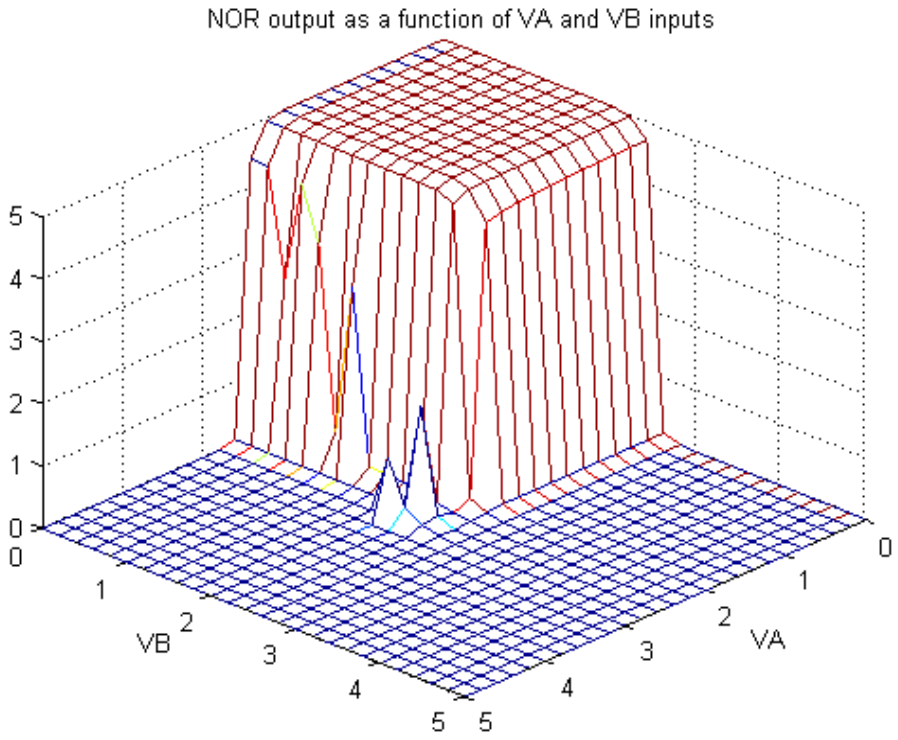


Fig. (4) The NOR Circuit Response to input values of V_A and V_B.

It can be seen from the result obtained from GAs, that minimum chip area can be achieved. To prove this, a compression between standard design method and GAs is made. Table (3) presents the results obtained using the Genetic Algorithm program and a standard design like traditional CMOS gates given in [1]. The comparison states clearly that GAs results are much better in achieving compromise between the total area and the voltage error for the NAND and NOR circuits than the design found in [1].

Table (3) Comparison between GAs Design and the Standard Design

	NAND		**NOR**	
	GA design	**Standard design**	**GA design**	**Standard design**
$(W/L)_1$	1.0345	2.5	1	2.5
$(W/L)_2$	1	2.5	1.0018	2.5
$(W/L)_3$	1.3009	2.5	1.2921	5
$(W/L)_4$	1.348	2.5	1.0412	5
Area	4.6834	10	4.3351	15
Voltage error	3.8716	8.4351	11.5829	13.73

4. Conclusions

This paper is concerned with design and realization digital integrated circuits based on using genetic algorithm concept. The main objective of the design method for this types is the minimization of the chip area required for fabrication. In digital IC the chip area is determined by transistors aspect ratio, this is then the function of this minimization procedure.

Genetic algorithms are realized to minimize the chip area required while sustaining the other performances such as frequency characteristic, transfer curve, etc'. To conclude the objectives of this work the following points are presented.

- GAs is in general more suitable to the problem of electronic circuits design and synthesis than a traditional technique like Newton-Raphson method that works on approximate and trial and error solutions.
- The advantage of GAs over traditional techniques is that it is more suitable for complicated structure. i.e. large set of equations with larger number of parameters, nonlinearities and nonsystematic equations.
- The GAs programs should be designed in a very flexible manner especially when the program deals with a complicated problem and the execution time is relatively long (in ours). Flexibility includes:

1. Storing the results after each generation.
2. Stopping the program and resuming its execution.
3. Monitoring the program status by observing as much amount of details as possible while the program is running.

REFERENCES

[1]. Ken. Martin, **"Digital Integrated Circuit Design"**, Oxford University Press 2000.

[2]. Melanie M., **"An Introduction to Genetic Algorithms"**, Massachusetts Institute of technology, 1996.

[3]. Goldberg D. E., **"Genetic Algorithms in Search, Optimization and Machine Learning"**, Addison Wesley Publishing Company, Inc., 1989.

[4]. James B. Grimbleby, **"hybrid Genetic Algorithms For Analogue Network Synthesis"**, IEE Conference Publication, Vol.2491, No. 446, PP 103-107, 1997.

[5]. Jason D. Lohn, **"A Parallel genetic Algorithm For Automated Electronic circuit Design"**, IEE Trans. On Evolutionary Computation, Vol.3, No.3, PP 205-219, 1999.

[6]. Kenneth V. Noren & John E. Ross, **"Analog Circuit Design Using genetic Algorithms"**, Scientific American, PP66-72, 1992.

[7]. Navid Azizi, **"Automated Analog circuit Design Using genetic Algorithms"**, Communications of the ACM, Vol.42, No.4, PP 71-9, 1999.

[8]. Jacob Millman & Arvin Grabel, **" Microelectronics"** second Edition, Mc Graw- Hill, 1988.