electronics

MDPI

Article

A Novel Framework of Genetic Algorithm and Spectre to Optimize Delay and Power Consumption in Designing Dynamic Comparators

Hoang Trong Nguyen and Trang Hoang *

Department of Electronics Engineering, Ho Chi Minh City University of Technology (HCMUT), Vietnam National University Ho Chi Minh City (VNU-HCM), Ho Chi Minh City 700000, Vietnam
* Correspondence: hoangtrang@hcmut.edu.vn

Abstract: In integrated circuit (IC) design, analog circuits contribute significantly as the interface between real and digital world signals. Although they make up a relatively small portion of the overall circuit, their design process is often most time-consuming, mostly from the phase of manual iteration of circuit parameters to meet design specifications. Therefore, the design automation of analog circuits with the help of efficient optimization techniques arises as a promising candidate to address the issue. Among optimization algorithms, while the genetic algorithm (GA) has been shown to be effective in finding near-optimal solutions, it has not been extensively applied to the field of analog circuit design. Hence, this paper proposes a method to utilize GA in the optimization of a widely used circuit topology, namely the comparator. The comparator is considered the fundamental block in the design of most analog-to-digital converters (ADCs). For high-speed ADCs, dynamic comparators are usually chosen for the purpose of high speed and power efficiency. In summary, this paper introduces an innovative GA-Spectre architecture to optimize the dynamic comparator with respect to delay and power consumption. The post-optimized results are optimistic with a 72.61 ps delay and 3.11 μW power dissipation.

Keywords: automation; genetic algorithm; dynamic comparator; delay; power consumption

Citation: Nguyen, H.T.; Hoang, T. A Novel Framework of Genetic Algorithm and Spectre to Optimize Delay and Power Consumption in Designing Dynamic Comparators. *Electronics* **2023**, *12*, 3392. https://doi.org/10.3390/electronics12163392

Academic Editor: Andrea De Marcellis

Received: 4 June 2023
Revised: 1 August 2023
Accepted: 7 August 2023
Published: 9 August 2023

1. Introduction

Comparators are considered the heart of analog-to-digital converters (ADCs). They are used as a means to convert from analog domain signals to digital domain signals in modern signal processing and communications. In the design of high-speed ADCs, low-power and high-speed comparators are of great demand [1]. Thanks to strong positive feedback and dynamic bias provided by a pair of cross-coupled inverters as the latching stage, dynamic comparators have higher speed and less static power consumption compared to static comparators [2]. Therefore, with a view to optimizing comparators' performance with respect to speed and power consumption, the dynamic comparator is chosen as a feasible candidate.

The analog circuit design consists of three main stages: topology selection, component sizing, and layout extraction. In the design of the comparator, this paper focuses on the first two stages. Both stages must ensure that the resulting circuit meets the specifications [3,4]. Since the first phase completes with the topology of the dynamic comparator, the second phase involves choosing the size of components to meet design specifications. Due to the repetitive task of manual iteration of circuit parameters, this sizing procedure is considered time-consuming and monotonous [4,5]. Hence, automation in the process of optimizing the sizes of circuits' components is critical to the ability to design high-performance circuits quickly [6].

To address the issue of laborious circuit sizing in analog circuit design, effective optimization techniques are crucial. Automated component sizing for analog circuit op-

timization can be classified as equation-based methods and simulation-based methods. On the basis of circuit analysis, equation-based methods utilize posynomial or monomial functions built on circuit parameters to represent specific circuit performances of interest. Despite the fast execution time and high certainty of reaching a global optimum, deriving such equations is often challenging and time-consuming. Moreover, to obtain explicit and closed-form expressions for circuit performances, various approximations and simplifications are usually applied, at the expense of MOSFET's higher-order effects, hence the model's accuracy and completeness. By contrast, simulation-based counterparts are independent of analytical functions but instead rely on SPICE simulation data. In the optimization procedure, these methods handle fitness functions (or objective functions) and design constraints in the form of black box functions, which are evaluated by simulated results. This approach might ensure better accuracy, generality, and convenience. Consequently, our optimization system in this research is chosen to be simulation-based.

Among various existing equation-free optimization methods, the genetic algorithm (GA), based on the Darwinian principle of natural selection and concepts of natural genetics, has been found to be an effective solution to large search spaces without being trapped in local minima [5]. In spite of GA's advantages, it has not been extensively applied to the field of circuit design. To the authors' best experience and knowledge, the algorithm is mostly implemented in the design of operational amplifiers as in [5,6] and has not been utilized for the case of the dynamic comparator. Furthermore, the design of [6,7] uses the HSPICE simulator for circuit simulations, which normally requires an additional step of using scripting languages for collecting necessary data. Alternatively, the Spectre simulator allows the use of the SKILL programming language's syntax in Ocean-based scripts. In view of the role of the Spectre simulator in the overall optimization system, the flexibility of SKILL programming establishes the authors' preference of Spectre over its HSPICE counterpart in terms of manipulating output data.

In recognition of GA's strengths and Spectre's convenience of data output, this paper proposed a GA-Spectre model that might break new ground as the prototype for the optimization problem of propagation delay and power dissipation for the dynamic comparator design. With only 100 iterations of GA, the optimized dynamic comparator achieved a power-delay product (PDP) of 0.2258 fJ, including an average delay of 72.61 ps and power consumption of 3.11 μW at a 1 GHz clock frequency and 1.2 V supply voltage. These are desirable and promising values for assessment parameters, especially for the case of PDP since this work's PDP surpasses its counterparts in the works of [8–11]. More importantly, thanks to its flexibility and adaptability, our GA-Spectre framework could also be the optimization tool for different circuits, which is likely to revolutionize the mindset and work approach of analog circuit design engineers.

The remaining part of the paper is organized as follows. Section 2 demonstrates the operation of the single-tail dynamic comparator as well as its delay and power analysis. Subsequently, Section 3 illustrates the optimization process, including GA's flow and the proposed GA-Spectre model to optimize the delay and power of the dynamic comparator. Simulation results and discussion are presented in Section 4, followed by the conclusion of the paper in Section 5.

2. Dynamic Comparator Analysis

2.1. Working Principle of the Single-Tail Dynamic Comparator

The operation of the conventional single-tail dynamic comparator depicted in Figure 1 consists of two phases [12]:

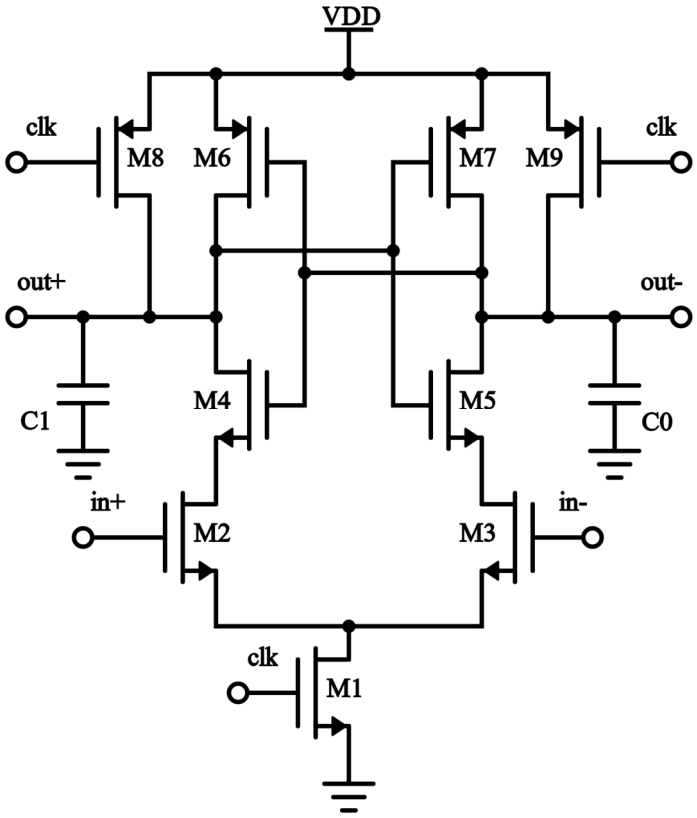

Figure 1. Conventional single-tail dynamic comparator.

Reset phase: The reset phase starts when clk = 0. In this phase, the reset transistors M_8 and M_9 are on while the tail transistor M_1 is off. As a result, output nodes out+ and out− are pulled up to V_{DD}, which ensures the initial condition as well as a valid logic level for the comparator.

Comparison phase (decision-making phase): The comparison phase starts when clk = V_{DD}. In this phase, the reset transistors M_8 and M_9 are off while the tail transistor M_1 is on. The output nodes out+ and out−, previously precharged to V_{DD}, turn M_4 and M_5 on. Also, these two output nodes begin to discharge their voltages, which is still high enough to keep M_4 and M_5 on. The discharging rate of out+ and out− depends on the voltages at two input nodes in+ and in−.

When in+ > in−: Out+ discharges at a faster rate compared to out−. This means that the voltage at out+ drops to $V_{DD} - |V_{THP}|$ before out−, making M_7 turn on before M_6. Since (M_4, M_6) and (M_5, M_7) together form back-to-back inverters, the latch regeneration is activated. Hence, out+ and out− are pulled down to GND and pulled up to V_{DD}, respectively.

When in+ < in−: The circuit works in the opposite manner with the final result of out+ and out− being pulled up to V_{DD} and pulled down to GND, respectively.

In summary, during the comparison phase:

$$\text{in+} > \text{in−} \Rightarrow \begin{cases} \text{out+} = \text{GND} \\ \text{out−} = V_{DD} \end{cases}$$

$$\text{in+} < \text{in−} \Rightarrow \begin{cases} \text{out+} = V_{DD} \\ \text{out−} = \text{GND} \end{cases}.$$

2.2. Delay Analysis

The propagation delay is one of the key features of a comparator. It consists of two parts:

Delay for the capacitors C_0 and C_1 to discharge to the point when M_6 and M_7 turn on:

$$t_0 = \frac{C_i|V_{THP}|}{I_3} \approx \frac{C_i|V_{THP}|}{\frac{I_1}{2}} = 2\frac{C_i|V_{THP}|}{I_1} \quad (1)$$

where C_i is the load capacitor at the output nodes with equal values (i = 0, 1 and $C_0 = C_1$); V_{THP} is the threshold voltage of p-channel MOSFETs M_6, M_7; and I_1, I_3 are the drain currents through M_1, M_3, respectively.

Delay from the two cross-coupled inverters:

Since the threshold voltage of the comparator is considered to be half of the supply voltage, or $\frac{V_{DD}}{2}$, it means that

$$\Delta V_{out} = \frac{V_{DD}}{2} \quad (2)$$

where ΔV_{out} is the output voltage swing and V_{DD} is the supply voltage.

Therefore, the latch delay is calculated as

$$t_{latch} = \frac{C_i}{gm_{eq}} \ln\left(\frac{\Delta V_{out}}{\Delta V_0}\right) = \frac{C_i}{gm_{eq}} \ln\left(\frac{V_{DD}/2}{\Delta V_0}\right) \quad (3)$$

where gm_{eq} is the equivalent transconductance of the latch and ΔV_0 is the output voltage difference.

Also, at time t_0:

$$\Delta V_0 = |V_{out+} - V_{out-}| = \left|(V_{DD} - |V_{THP}|) - \left(V_{DD} - \frac{I_2 t_0}{C_i}\right)\right| = \left||V_{THP}| - \frac{I_2 t_0}{C_i}\right| = |V_{THP}|\left(1 - \frac{I_2}{I_3}\right)\Bigg| = |V_{THP}|\frac{\Delta I_{in}}{I_3} \quad (4)$$

where I_3 is the drain current through M_3 and ΔI_{in} is the current difference at the input ends.

Since $I_3 \approx \frac{I_1}{2}$:

$$\Delta V_0 = |V_{THP}|\frac{\Delta I_{in}}{I_1/2} = \frac{2|V_{THP}|\sqrt{\beta_{2,3} I_1}}{I_1}\Delta V_{in} = 2|V_{THP}|\sqrt{\frac{\beta_{2,3}}{I_1}}\Delta V_{in} \quad (5)$$

where $\beta_{2,3} \equiv \beta_2, \beta_3$ are the current factors of M_2, M_3, respectively.

Substitute (5) into (3):

$$t_{latch} = \frac{C_i}{gm_{eq}} \ln\left(\frac{V_{DD}/2}{2|V_{THP}|\sqrt{\beta_{2,3}/I_1}\Delta V_{in}}\right) = \frac{C_i}{gm_{eq}} \ln\left(\frac{V_{DD}}{4|V_{THP}|\Delta V_{in}}\sqrt{\frac{I_1}{\beta_{2,3}}}\right) \quad (6)$$

The total delay is the sum of its two parts:

$$t_{delay} = t_0 + t_{latch} = 2\frac{C_i|V_{THP}|}{I_1} + \frac{C_i}{gm_{eq}} \ln\left(\frac{V_{DD}}{4|V_{THP}|\Delta V_{in}}\sqrt{\frac{I_1}{\beta_{2,3}}}\right) \quad (7)$$

The simulation results illustrate that t_0 dominates t_{latch} [1] and t_{delay} follows the change in t_0. In other words, when I_1 decreases, t_0 increases and t_{delay} hence increases, and vice versa.

2.3. Power Analysis

In order to prevent inaccuracies at boundaries between operating regions, instead of MOSFET's existing models, its time-variant model is applied to analyze the power of the conventional dynamic comparator [13]. The formula for drain current applicable to all operating regions is expressed in the work of [14] as

$$I_D = I_Z\left[\ln^2\left(1 + e^{\frac{V_{GS}-V_T}{2n\phi_t}}\right) - \ln^2\left(1 + e^{\frac{V_{GS}-V_T-nV_{DS}}{2n\phi_t}}\right)\right] \quad (8)$$

where V_{GS} is the gate–source potential difference, V_T is the threshold voltage, V_{DS} is the drain–source potential difference, and ϕ_t is the thermal voltage $\frac{kT}{q}$.

I_Z and n are given by

$$I_Z = 2\mu C_{ox}\frac{W}{L}n\phi_t^2 \tag{9}$$

$$n = \left[1 - \frac{\gamma}{2\sqrt{V_{GB} - V_{T0} + \left(\gamma/2 + \sqrt{2\phi_F}\right)^2}}\right]^{-1} \tag{10}$$

where γ is the body-effect coefficient, V_{GB} is the gate-bulk potential difference, V_{T0} is the threshold voltage with zero source-bulk voltage V_{SB}, $\phi_F = \frac{kT}{q}\ln\left(\frac{N_{sub}}{n_i}\right)$, k is Boltzmann's constant, q is the electron charge, N_{sub} is the doping density of the subtrate, and n_i is the density of electrons in undoped silicon.

For one period of comparison, the average power of the supply voltage is calculated as

$$Power_{avg} = \frac{1}{T}\int_0^T V_{DD}I_{supply}dt = f_{clk}V_{DD}\int_0^T I_{supply}dt \tag{11}$$

where f_{clk} is the frequency of the comparator's clock, V_{DD} is the supply voltage, and I_{supply} is the current drawn from the supply voltage.

When clk = 0 (reset phase), in order to charge the output nodes to V_{DD}, a current is drawn from the supply voltage source.

When clk = V_{DD} (decision-making phase), assuming that in+ > in−, according to the working principle of the conventional dynamic comparator explained above, M_7 will turn on before M_6. Since M_5 has already been on, there is a current drawn from V_{DD} from M_7. Therefore, during this comparison phase, I_{supply} is equivalent to the current through M_7. Meanwhile, as the voltage at out− discharges to the ground, M_5 will be off and there will be no current drawn from V_{DD}. As a result, such a comparator is classified as dynamic [13].

To calculate the average power during the comparison phase, we apply the time-variant model for current through M_7 described in (8) to the formula in (11):

$$Power_{avg} = f_{clk}V_{DD}I_7\int_{t_0}^{t_{latch}}\left[\ln^2\left(1 + e^{\frac{V_{DD} - V_{out-}(t) - |V_{THP}|}{2n\phi_t}}\right) - \ln^2\left(1 + e^{\frac{V_{DD} - V_{out-}(t) - |V_{THP}| - n(V_{DD} - V_{out+}(t))}{2n\phi_t}}\right)\right]dt \tag{12}$$

where the lower and upper bound of the integral in (12), t_0 and t_{latch}, respectively, are clarified in the delay analysis part.

For the integral in (12) to be solvable, it is necessary that the approximation $\ln^2(1 + e^x) \approx \ln^2(e^x) = x^2$ when $e^x \gg 1$ is used. Since the exponential terms of (12) are much larger than 1, the mentioned approximation is valid.

Therefore:

$$Power_{avg} = f_{clk}V_{DD}I_7\int_{t_0}^{t_{latch}}\left[\left(\frac{V_{DD} - V_{out-}(t) - |V_{THP}|}{2n\phi_t}\right)^2 - \left(\frac{V_{DD} - V_{out-}(t) - |V_{THP}| - n(V_{DD} - V_{out+}(t))}{2n\phi_t}\right)^2\right]dt \tag{13}$$

Simplifying the integral in (13), the closed-form expression for power is obtained as

$$Power_{avg} = f_{clk}V_{DD}I_7\left(\frac{1}{8n\phi_t^2}\right)\tau_{latch}|V_{THP}|\left[2k - n|V_{THP}| + (2k + n|V_{THP}|)e^{\frac{-2(t_{latch} - t_0)}{\tau_{latch}}} - 4ke^{-\frac{t_{latch} - t_0}{\tau_{latch}}}\right] \tag{14}$$

where $k = V_{DD} - |V_{THP}|$ and $\tau_{latch} = \frac{gm_{eq}}{C_i}$ (gm_{eq} is the equivalent transconductance of the latch as mentioned in the delay analysis).

For the case in+ < in−, power dissipation can be obtained by substituting I_6 with I_7 in the formula of (14):

$$Power_{avg} = f_{clk}V_{DD}I_6\left(\frac{1}{8n\phi_t^2}\right)\tau_{latch}|V_{THP}|\left[2k - n|V_{THP}| + (2k + n|V_{THP}|)e^{\frac{-2(t_{latch} - t_0)}{\tau_{latch}}} - 4ke^{-\frac{t_{latch} - t_0}{\tau_{latch}}}\right] \tag{15}$$

2.4. Cadence Virtuoso Simulation

The conventional dynamic comparator is designed and simulated in the 65 nm technology of the TSMCN65 process. The frequency at which the circuit functions is f_{clk} = 1 GHz and the supply voltage is V_{DD} = 1.2 V. The voltage at node in− is constant at 1 V as a reference voltage, while in+ is a pulse voltage source with the maximum and minimum value of 1.005 V and 0.995 V, respectively, and a frequency of 100 MHz. With this input configuration, ΔV_{in} = 5 mV.

Figure 2 shows the transient simulation of the conventional comparator in one clock period that consists of both the comparison and the reset phase. t_0 and t_{latch} are the parameters explained above in Section 2. The total propagation delay of the dynamic comparator is the sum of t_0 and t_{latch} in Figure 2, $t_{delay} = t_0 + t_{latch}$.

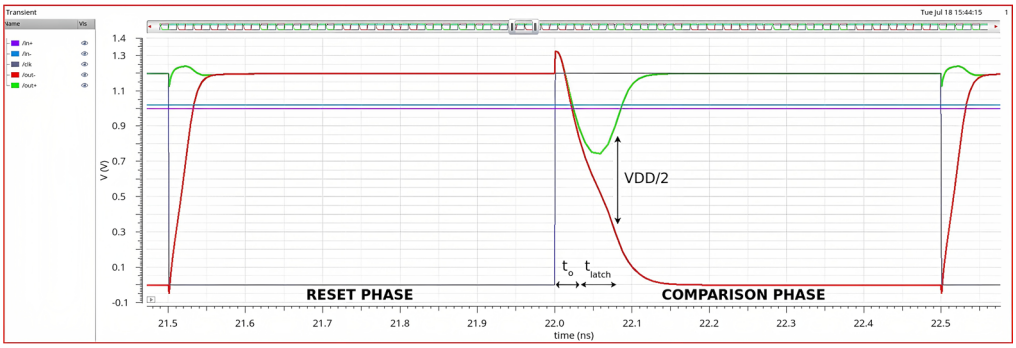

Figure 2. Transient simulation of the conventional dynamic comparator.

Figure 3 demonstrates the transient simulation of the current I_{supply} drawn from the supply voltage V_{DD} in one period from 0 to 1 ns. To calculate the power dissipation of the dynamic comparator, we integrate I_{supply} with respect to time from 0 to 1 ns and multiply with f_{clk} and V_{DD} as in Equation (11) to obtain the result.

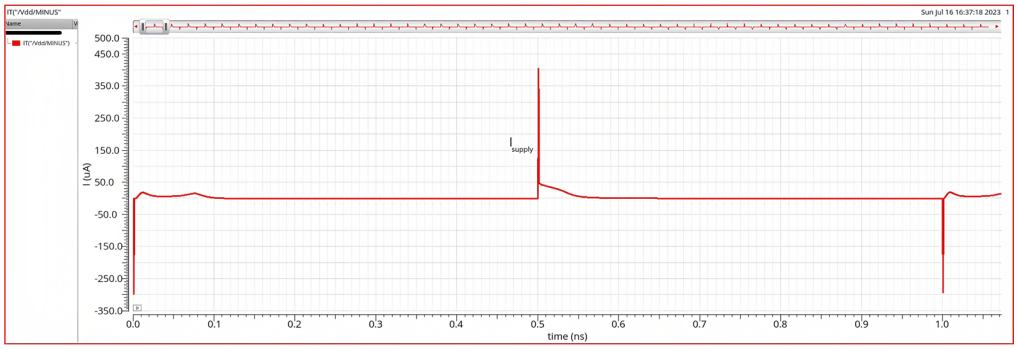

Figure 3. Transient simulation of the current I_{supply}.

3. Optimization by Genetic Algorithm

3.1. Genetic Algorithm (GA)

The genetic algorithm (GA) is one of the evolutionary computation methods used for a wide variety of optimization problems. GA's basis is the principle of natural selection, in which suitable individuals in a specific environment survive and reproduce while others are eliminated [15]. With the idea of "survival to the fittest", better solutions are generated thanks to the successive evolution of generations.

The implementation of GA is described in the flowchart of Figure 4:

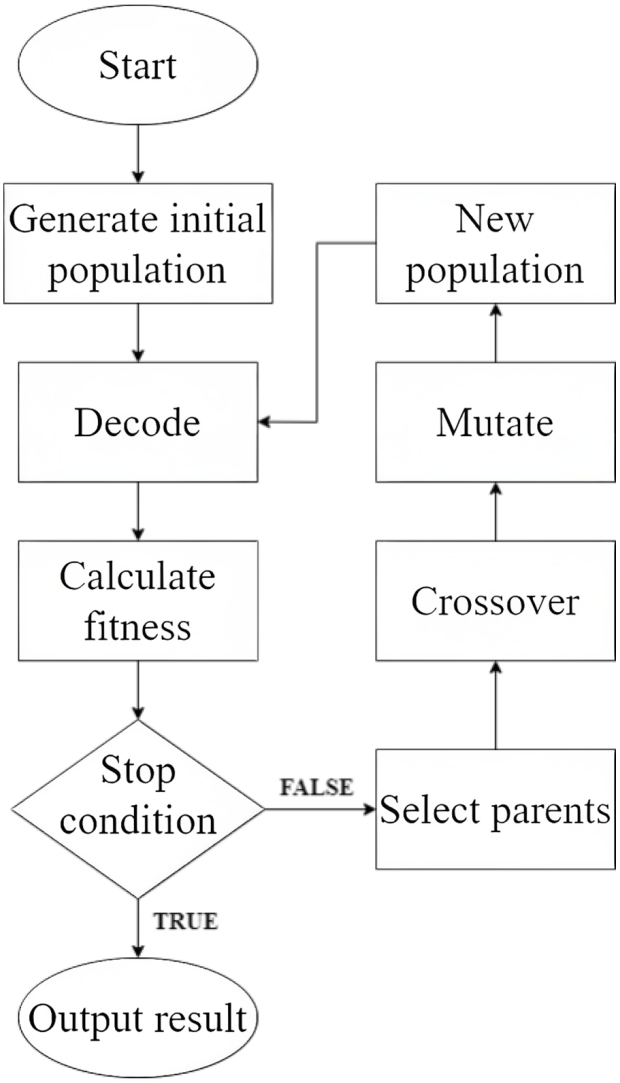

Figure 4. Genetic algorithm's flowchart.

As illustrated in Figure 4, rather than a single solution, GA utilizes a population of individuals for parallel search in the problem space. The algorithm starts by initializing a population of a fixed size with randomly created solutions. These solutions, usually encoded in a bitstring of a fixed length, are modeled as chromosomes. The chromosomes consist of genes, which are represented by one bit or a group of bits. Every generation, the bitstrings of chromosomes are decoded into their actual representation. Then, based on a fitness function, the fitness values of all individuals are evaluated to find the best individual in the current population. These fitness values evaluate how close the current solution is to the predetermined target or the optimal solution of the problem. After this step, the stop condition, including the number of iterations or the comparison with the target, is considered. If this condition is not met, the algorithm continues with the selection process.

According to the fitness value, a subgroup of chromosomes—the parents—is selected to create the new population of offspring by the process of crossover and mutation. During crossover, two parents are randomly selected from the set of chosen parents; their bitstrings are separated into parts at the same location and swapped together to create two child chromosomes for the next generation. In order to prevent the solutions from becoming stuck at local optima, the next step involves the mutation process, which can be as simple as inverting one bit in the bitstring or more complicated as complete gene modifications [15]. The crossover and mutation steps are implemented with certain probabilities r_{cross} and r_{mut}

inside the range [0, 1], respectively. Since a new population has already been created, the algorithm continues with the decoding and fitness calculation. When the fitness results of the children population are available, the best chromosome can be found. Afterward, the terminating condition is re-evaluated. GA stops only when this condition is true; otherwise, the loop of regeneration carries on. The output result of GA is the decoded solution of the chromosome with the best fitness value.

From another viewpoint, GA can also be summarized by Pseudocode 1:

Pseudocode 1: Genetic algorithm (GA)

```
decoded_solution genetic_algorithm() {
    population = initial_population();
    best = calculate_fitness(decode(population));
    while (!stop_condition) {
        parents = select(population);
        children = crossover(parents);
        new_population = mutation(children);
        best = calculate_fitness(decode (new_population));
    }
    return decode(best);
}
```

3.2. Proposed GA-Spectre Model to Optimize Delay and Power of the Dynamic Comparator

When applying GA to optimization problems, the structure of the algorithm almost remains unchanged. Meanwhile, it is necessary to make suitable modifications to the step of fitness calculation for specific problems. For the optimization of analog circuit design, the "Calculate fitness" block in Figure 4 consists of two substeps. The first one is delay and power simulations, which are executed by the Cadence Virtuoso tool in the TSMCN65 process, with the help of the Spectre simulation platform. The remaining step is the evaluation of fitness values according to delay and power data collected in the previous step.

In this research, GA is implemented by the Python programming language since Python is currently considered the most widely used programming language and has various built-in libraries for AI algorithms' programming. Also, Spectre is chosen as the circuit simulator for the whole design as an alternative to related studies. Circuit simulations in the works of [6,7] are performed by HSPICE, whose output data are mostly presented in the simulator's predetermined format. Thus, scripting languages are usually necessary to create files containing output results of compatible format for Python processing of GA. On the other hand, it is possible for SKILL programming's syntax to be integrated in Ocean-based scripts of the Spectre simulator. This means that simulation results can be arranged in users' desirable format without further use of scripting languages. On account of its function in the optimization system and convenience in output data format, Spectre is preferably chosen as a practical candidate for circuit simulations.

The interrelation between Spectre and Python is described as follows: At first, the values for the population of design variables are created by Python and sent to Spectre via an Ocean-based script. This Ocean script is responsible for automated circuit simulations based on received numbers and the results are sent back and further processed by the Python script to evaluate fitness scores of each individual. This repetitive process carries on until the algorithm reaches its stopping point. The mentioned Spectre-Python correlation is further clarified by the block diagram in Figure 5:

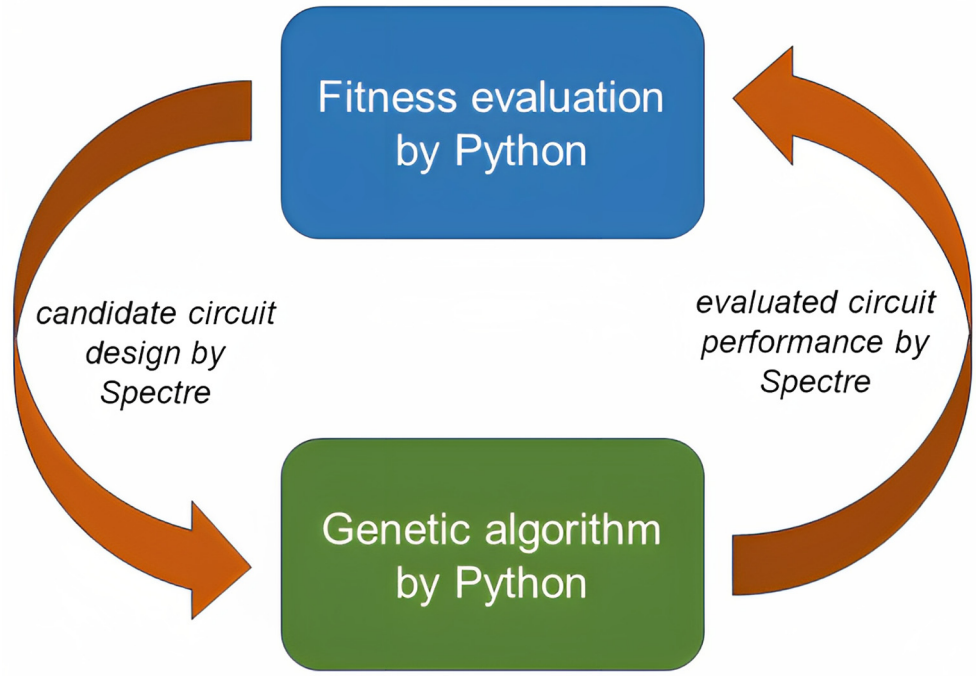

Figure 5. Block diagram of the Spectre-Python-based optimization system.

Regarding the case of optimizing the delay and power of the dynamic comparator, first and foremost, determining the optimization variables—parameters that mainly contribute to delay and power results—is essential. Since we use the TSMCN65 process for the design, the lengths of all the MOSFETs in the circuit are set to 65 nm. According to the delay analysis in Section 2, since ΔV_{in} of the dynamic comparator is fixed at 5 mV, I_1 and C_i are responsible for delay adjustment. Therefore, we declare two delay-related variables: the width of M_1 (W_1) and the capacitors' values (C_i). Similarly, as the values of V_{DD} and the frequency of the clock signal clk are assigned to 1.2 V and 1 GHz, respectively, our power-related variables are determined based on the current I_{supply}, which is the sum of four currents flowing through $M_6, M_7, M_8,$ and M_9. Due to the symmetry of the circuit, the widths of M_6 and M_7, and M_8 and M_9 should be equal. Hence, we declare two more variables: the width of M_6 and M_7 (W_{67}) and the width of M_8 and M_9 (W_{89}). For the remaining MOSFETs, the widths of M_2 (W_2) and M_3 (W_3), and M_4 (W_4) and M_5 (W_5) are also set to equal values to ensure the circuit's symmetry. For optimal performance of delay and power consumption, the simulation results indicate that $W_2 = W_3 = 0.21$ µm, and $W_4 = W_5 = 0.12$ µm. In total, we need four optimization variables, namely $W_1, W_{67}, W_{89},$ and C_i. In order to satisfy the range of the process and assure the functional correctness of the conventional dynamic comparator, the simulation results indicate that the bounds for the four above-mentioned variables are [0.12 µm; 2 µm], [0.12 µm; 2 µm], [0.12 µm; 2 µm], and [0.1 fF; 0.8 fF], respectively.

For subsequent GA initializing steps, the number of bits for each chromosome representation is defined to be 16 bits and the population size is chosen to be six individuals per population, correspondingly. While crossover has a high probability, the probability of mutation is typically low; the authors in [16] showed that r_{cross} and r_{mut} are inside the ranges [0.8; 0.95] and [0.001; 0.05], respectively. Finally, the choice of a suitable fitness function is also critical. The figure of merit (FoM) to compare the performance of dynamic comparators can be either energy efficiency as in [8] or PDP as in [17]. With a view to optimizing both delay and power of the dynamic comparator, we choose the FoM in the form of PDP as the fitness function for GA. The comparator is considered to perform better with a lower value of PDP, which means that a lower PDP is equivalent to higher fitness values.

In other words, our optimization problem can be summarized as

$$\begin{aligned} &\text{minimize PDP}(W_1, W_{67}, W_{89}, C_i) \\ &\text{subject to } L = 65\text{ nm},\ W_2 = W_3 = 210\text{ nm},\ W_4 = W_5 = 120\text{ nm} \\ &\qquad \Delta V_{in} = 5\text{ mV},\ V_{DD} = 1.2\text{ V},\ f_{clk} = 1\text{GHz} \\ &\qquad 0.12\ \mu\text{m} \leq W_1, W_{67}, W_{89} \leq 2\ \mu\text{m} \\ &\qquad 0.1\text{ fF} \leq C_i \leq 0.8\text{fF} \end{aligned}$$

4. Results and Discussion

The simulation results indicate that the lowest value for PDP of 0.2254 fJ is achieved for the case of r_{cross} = 0.8 and r_{mut} = 0.05. With PDP = 0.2254 fJ, the delay and power of the conventional dynamic comparator are 72.48 ps and 3.11 µW, respectively. PDP fitness values as well as the delay and power over 100 iterations of GA are illustrated in Figures 6 and 7, respectively.

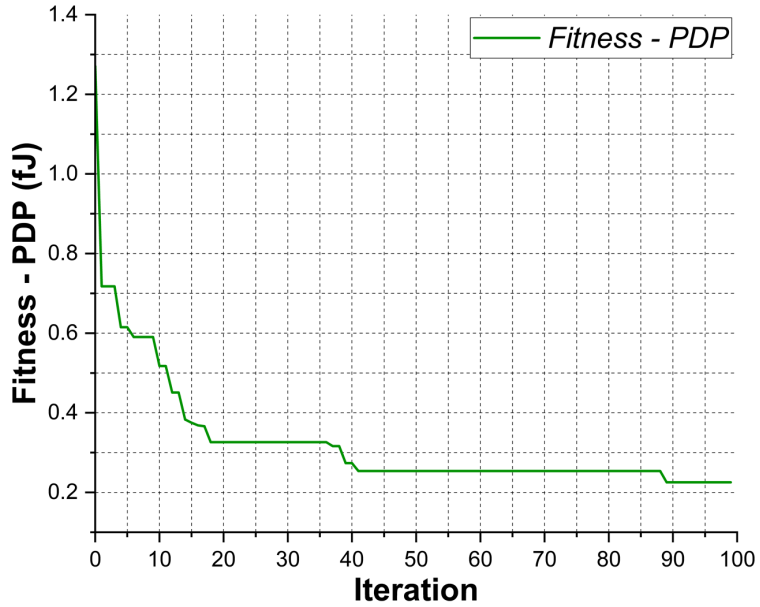

Figure 6. Fitness value of PDP versus 100 iterations.

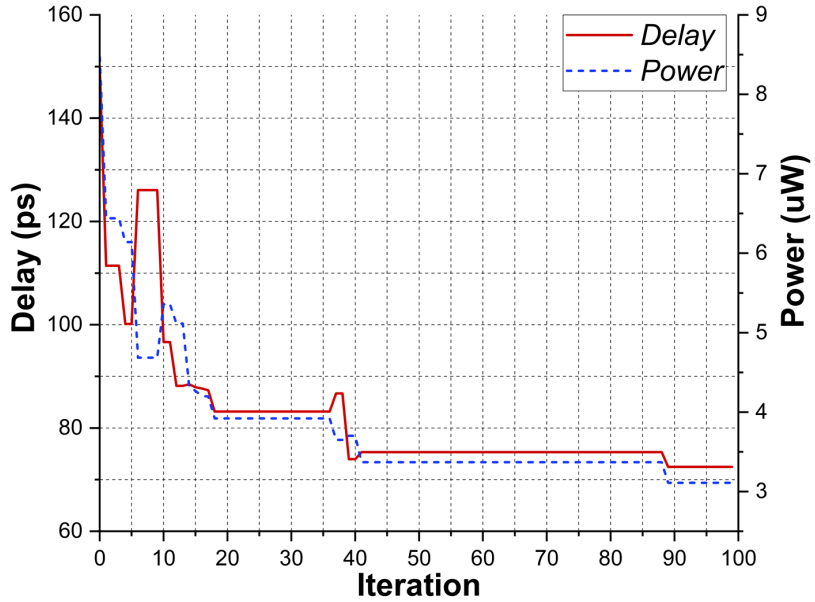

Figure 7. Delay and power versus 100 iterations.

As can be clearly observed in Figure 7, the values for delay and power vary in an unpredictable and non-monotonous manner. However, their corresponding PDP in Figure 6 decreases monotonously throughout the 100 iterations. Since GA produces chromosomes with better fitness values at the end of each iteration, PDP's downward trend conforms to the working principle of the algorithm. In terms of the variables declared for GA, the optimal set $(W_1, W_{67}, W_{89}, C_i)$ = (0.4463 µm, 0.1277 µm, 0.1553 µm, 0.1 fF) is obtained after 100 iterations of the algorithm. Nevertheless, it is worth noticing that the process grid of the TSMCN65 process is 5 nm. This means that the widths of M_1, M_6, M_7, M_8, M_9 need to be rounded to their closest feasible values as (W_1, W_{67}, W_{89}) = (0.445 µm, 0.13 µm, 0.155 µm). Re-simulated results with the set $(W_1, W_{67}, W_{89}, C_i)$ = (0.445 µm, 0.13 µm, 0.155 µm, 0.1 fF) vary slightly with the final values of 72.61 ps for delay, 3.11 µW for power consumption, and 0.2258 fJ for PDP.

The post-optimization sizes of all MOSFETs in the circuit are presented in Table 1:

Table 1. Post-optimization transistors' sizes.

Device	Size (W/L)
M_1	0.445 µm/65 nm
M_2, M_3	0.21 µm/65 nm
M_4, M_5	0.12 µm/65 nm
M_6, M_7	0.13 µm/65 nm
M_8, M_9	0.155 µm/65 nm
C_0, C_1	0.1 fF

Table 2 summarizes the performance of the conventional dynamic comparator of this research and other research works:

Table 2. Performance summary of different dynamic comparator designs.

Parameters	References					
	[1]	[8]	[9]	[10]	[11]	This Work
CMOS Process (nm)	90	65	65	90	40	65
Supply voltage (V)	1.2	1	1.2	1	1.1	1.2
Clock frequency (GHz)	3.07	20	6	1	6	1
Average delay (ps)	410	14.28	42.7	51.76	54	72.61
Power dissipation (µW)	0.24	67.8	381	32.62	288	3.11
PDP (fJ)	0.0984	0.968	16.3	1.67	15.552	0.2258
Energy per conversion (fJ/conv.)	0.07818	3.39	63.5	32.62	48	3.11
Kickback noise (mV)	N/A	N/A	N/A	N/A	N/A	122

As the conventional dynamic comparator in this work has zero static power consumption, its power consumption at 1 GHz is much lower than that of the circuit of [10]. Since clock frequency is directly proportional to power dissipation as presented in Equation (11), the designs of [8,9,11] with higher clock frequency exhibit a higher power than our design, which is reasonable. Meanwhile, the power consumption in [1] is still lower despite operating at higher frequency. The parameter energy per conversion, which is equal to the ratio of power over sampling frequency (or clock frequency), is therefore needed to evaluate dynamic comparators' performance with respect to power. From Table 2, it is clear that our research work has the second-lowest energy per conversion value at 3.11 fJ per conversion.

In addition, compared to [9–11], our work has approximately a 20% higher average delay. On the contrary, our average delay is less than one-fifth in comparison with the delay of [1]. Because delay and power trade off with each other, PDP is utilized as the FoM in the case of optimizing both parameters. With respect to PDP, our design obtains the second-best value of 0.2258 fJ versus the lowest number of 0.0984 fJ by [1].

For further assessment of our optimization system, the optimization platform of the Analog Design Environment (ADE) GXL, which offers both local and global optimization

of circuit performances of interest, can be utilized as a suitable reference model. For setup steps, PDP is chosen as the optimization function, and the optimization variables and their bounds are similar to the GA-Spectre-based system. Regarding ADE GXL's local optimization, the post-optimization transistors' sizes $(W_1, W_{67}, W_{89}, C_i)$ = (0.445 µm, 0.13 µm, 0.155 µm, 0.1 fF) obtained from the GA-Spectre framework are set as the reference points. With regard to global optimization from ADE GXL, due to our constrained data, the C version Feasible Sequential Quadratic Programming (CFSQP) is selected by the Spectre simulator as the optimization algorithm. It is worth acknowledging that while the CFSQP utilizes a single individual for each iteration rather than a population of individuals, GA in our system is implemented on a population of six individuals per iteration. Hence, for a decent comparison, the PDP result of 100 iterations of our GA-based optimization system should be compared with that of 6 × 100 = 600 iterations of the ADE GXL's global optimization tool.

Table 3 indicates that the result of PDP acquired by our optimization system is lower and has a higher convergence rate compared to that of ADE GXL's local as well as global optimization tool.

Table 3. Comparison between different optimization models.

Parameters	Optimization Model		
	ADE GXL's Local Optimization	**ADE GXL's Global Optimization**	**This Work's GA-Based Optimization**
PDP (fJ)	0.227	0.236	0.2258

Figure 8 depicts the layout of the conventional dynamic comparator, which occupies an area of approximately 198.4 µm^2 (16 µm × 12.4 µm). Additionally, Table 4 demonstrates layout parameters of different designs while Table 5 represents a comparison between pre-layout and post-layout simulation results.

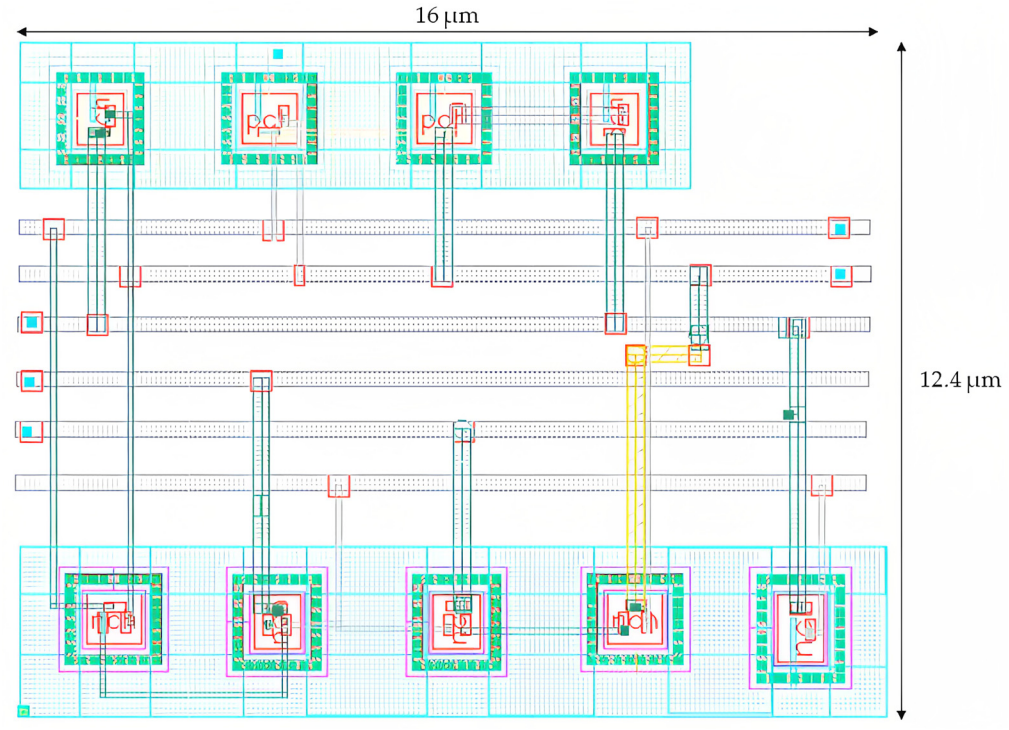

Figure 8. Layout of the conventional dynamic comparator.

Table 4. Layout summary of different dynamic comparator designs.

Parameters	References					
	[1]	[8]	[9]	[10]	[11]	This Work
CMOS Process (nm)	90	65	65	90	40	65
Estimated dimension (μm × μm)	N/A	12.22 × 15	10.9 × 13	7.2 × 8.1	13.5 × 4.5	16 × 12.4
Estimated area (μm^2)	N/A	183.3	141.7	58.32	60.75	198.4

Table 5. Comparison between pre-layout and post-layout simulation results of this work's comparator.

Parameters	Pre-Layout	Post-Layout
Average delay (ps)	72.61	82.32
Power dissipation (μW)	3.11	3.58
PDP (fJ)	0.2258	0.2947
Energy per conversion (fJ/conv.)	3.11	3.58
Kickback noise (mV)	122	163

5. Conclusions

GA can be considered a novel approach of using a software algorithm to optimize analog circuits. Instead of a trial-and-error process of circuit sizing, the GA-Spectre model solves the problem in a much more effective and time-saving manner. This paper proposes a new GA-Spectre model to design the dynamic comparator and successfully applies this model to optimize the dynamic comparator with a result of 72.61 ps of delay and 3.11 μW of power dissipation. Hence, the algorithm proves to be a promising solution to the optimal circuit's parameters. Since the mentioned GA-Spectre architecture could also be applied to optimization problems of different circuits, it might be the driving force to transform the way analog circuit engineers work.

Author Contributions: Methodology, T.H.; software, H.T.N.; investigation, H.T.N.; data curation, H.T.N.; supervision, T.H.; funding acquisition, T.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by Vietnam National University—Ho Chi Minh City (VNU-HCM) under grant number: DS2023-20-03.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jakhara, S.; Mandloi, V.S.; Goswami, R.; Kandpal, K. A Low Power, High Speed 1.2 V Dynamic Comparator for Analog-to-Digital Converters. In Proceedings of the Third International Conference on Computing and Network Communications, Kerala, India, 18–21 December 2019.
2. Wang, Y.; Yao, M.; Guo, B.; Wu, Z.; Fan, W.; Liou, J.J. A Low-Power High-Speed Dynamic Comparator with a Transconductance-Enhanced Latching Stage. *IEEE Access* **2019**, *7*, 93396–93403. [CrossRef]
3. Cohen, M.W.; Aga, M.; Weinberg, T. Genetic Algorithm Software System for Analog Circuit Design. In Proceedings of the 25th CIRP Design Conference, Haifa, Israel, 2–4 March 2015.
4. Nguyen, H.M.; Pham, L.D.; Hoang, T. A novel Li-ion battery charger using multi-mode LDO configuration based on 350 nm HV-CMOS. *Analog Integr. Circuits Signal Process.* **2016**, *88*, 505–516. [CrossRef]
5. Noren, K.V.; Ross, J.E. Analog Circuit Design Using Genetic Algorithms. In Proceedings of the Online Symposium for Electronics Engineers; 2002. Available online: http://www.johnross.com/2001_OSEE_Paper.pdf (accessed on 4 June 2023).
6. Mallick, S.; Kar, R.; Mandal, D.; Ghoshal, S.P. Optimal sizing of CMOS analog circuits using gravitational search algorithm with particle swarm optimization. *Int. J. Mach. Learn. Cyber.* **2019**, *8*, 309–331. [CrossRef]
7. Barari, M.; Karimi, H.R.; Razaghian, F. Analog Circuit Design Optimization Based on Evolutionary Algorithms. *Math. Probl. Eng.* **2014**, *2014*, 593684. [CrossRef]
8. Folla, J.K.; Crespo, M.L.; Wembe, E.T.; Bhuiyan, M.A.S.; Cicuttin, A.; Essimbi, B.Z.; Reaz, M.B.I. A low-offset low-power and high-speed dynamic latch comparator with a preamplifier-enhanced stage. *IET Circuits Devices Syst.* **2021**, *15*, 65–77. [CrossRef]

9. Ghasemian, H.; Ghasemi, R.; Abiri, E.; Salehi, M.R. A novel high-speed low-power dynamic comparator with complementary differential input in 65 nm CMOS technology. *Microelectron. J.* **2019**, *92*, 104603. [CrossRef]
10. Savani, V.; Devashrayee, N.M. Analysis and design of low-voltage low-power high-speed double tail current dynamic latch comparator. *Analog Integr. Circuits Signal Process.* **2017**, *93*, 287–298. [CrossRef]
11. Huang, S.; He, L.; Chou, Y.K.; Lin, F. A 288-µW 6-GHz hybrid dynamic comparator with 54-ps delay in 40-nm CMOS. In Proceedings of the 2016 IEEE MTT-S International Wireless Symposium (IWS), Shanghai, China, 14–16 March 2016.
12. Rameshkumar, R.; Bharathiraja, S. A Comparative Analysis of High-Speed Dynamic Comparator in 180nm and 90nm Using H-Spice. *Int. J. Innov. Sci. Eng. Res. IJISER* **2014**, *1*, 452–456.
13. Babayan-Mashhadi, S.; Daliri, M.; Lotfi, R. Analysis of power in dynamic comparators. In Proceedings of the 2013 21st Iranian Conference on Electrical Engineering (ICEE), Mashhad, Iran, 14–16 May 2013.
14. Tsividis, Y.; Suyama, K.; Vavelidis, K. Simple reconciliation MOSFET model valid in all regions. *Electron. Lett.* **1995**, *31*, 506–508. [CrossRef]
15. Jiang, Y.; Ju, J.; Zhang, X.; Yang, B. Automated analog circuit design using Genetic Algorithms. In Proceedings of the 2009 3rd International Conference on Anti-Counterfeiting, Security, and Identification in Communication, Hong Kong, China, 20–22 August 2009.
16. Yang, X.S.; Chien, S.F.; Ting, T.O. Bio-Inspired Computation and Optimization: An Overview. In *Bio-Inspired Computation in Telecommunications*; Yang, X.S., Chien, S.F., Ting, T.O., Eds.; Elsevier Inc.: Waltham, MA, USA, 2015; pp. 1–21. ISBN 9780128015384.
17. Samid, L.; Volz, P.; Manoli, Y. A dynamic analysis of a latched CMOS comparator. In Proceedings of the 2004 IEEE International Symposium on Circuits and Systems (ISCAS), Vancouver, BC, Canada, 23–26 May 2004.