Jordan University of Science
and Technology

College of Computer Sciences & Information Technology

Name: Dina Al-Sadi

Supervised By: Dr. Abdel Rauf Rjoub

Efficient multi-threshold voltage techniques for minimum leakage current in nanoscale technology

ABSTRACT

This research investigates the effect of multi-threshold voltage technique on minimize leakage power in CMOS circuits at the transistor level, using nanoscale SPICE parameters. By integrating Artificial Intelligence optimization, three innovative algorithms are: Slack Time Search Algorithm (STS), Leakage Power Search Algorithm (LPS), and Leakage and Slack Time Search Algorithm (LSS). These algorithms accurately compute the optimal threshold voltage for each transistor, achieving a remarkable 70% reduction in subthreshold leakage current while maintaining circuit performance. Using 22 nm predictive SPICE parameters (BSIM4). Comprehensive simulations confirm the reliability and effectiveness of the proposed algorithms, representing a major breakthrough in low-power design methodologies and opening new avenues for energy-efficient solutions in nanoscale CMOS technologies.

1. INTRODUCTION

In our modern world of digital innovation, reducing power dissipation it is consider one of the most important challenges, especially with the rapid widespread of portable devices. As technology has scaled down from the micrometre era to the nanoscale, the demand and focus on increase the performance and reliability (increase the speed of operation), reduce the energy consumption. In previous technologies, particularly micrometer scales, power dissipation was the primary priority pending to the larger size of transistors and low packing density that means, that the components of the transistor are spread out more loosely, resulting more space between the parts of the transistor. Even so, as transistors sizes have shrunk into nanoscales, new challenges have appeared. In today's nanoscale circuits, where sizes reach as little than 3 nanometer, leakage power has become the major cause of total power loss. This transition from micrometer to nanometer scale is driven by the higher density of transistors, lower threshold voltages and the growing quantum effects like tunneling that is increasing at these tiny dimensions. These development have enabled incredible evolution in performance and minimization but on other hand, have also serious problems and challenges in managing the leakage currents and ensuring energy efficiency. To deal with urgent issues, researchers have developed many techniques like the Multi-Threshold Voltage Technique (Multi-Vth), which has been famous and widely used due to its effectiveness in reducing leakage power. This method uses low threshold voltages for critical paths and high threshold voltages for non-critical paths to decrease power leakage. However, as circuits have developed and begin more complex, the dual Vth methods are not sufficient, so it is necessary to develop more advanced solutions. This paper will introduce three new advanced algorithms that are designed to optimize threshold voltage assignments at the transistor: Slack Time Search Algorithm(STS), Leakage Power Search Algorithm(LPS), and Leakage and Slack Time Search and Algorithm(LSS). In constraint of the previous method that focuses on gate level optimization or improved the predefined Vth values, these algorithms enable more efficient methods to reduce leakage power without effect on the performance. By utilizing this new techniques, this study aims to achieve more power optimization, opening the door for more energy efficient nanoscale design.

2. THRESHOLD SELECTION AND ASSIGNMENT PROBLEMS

The Threshold Assignment Problem (TAP) is a challenge in circuit design where the purpose is to assign a threshold voltage (Vth) suitable for each transistor to reduce the leakage power without effecting on the circuit performance. This is satisfied by adjusting the Vth for each transistor independently, taking advantage of the differences between paths to maximize the power reduction. To solve TAP, the threshold Selection Problem (TSP) must first handle, which includes selecting the optimal Vth values from a predefined range to minimize leakage power.

In this study, the threshold voltage (Vth) of a transistor is adjusted by changing the thickness of its oxide layer (Tox). There are 11 levels of Tox, each one leading to a different

value for Vth. Level 0 represents the thinnest oxide layer with the lowest Vth, while Level 11 represents the thickest oxide layer with the highest Vth.

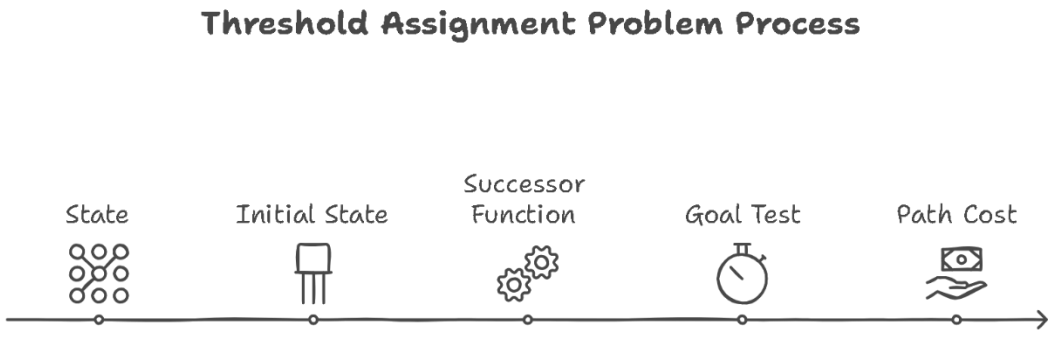
2.1 Formation of the Threshold Assignment Problem

The Problem is addressed though search algorithm in three phases:

1. **Goal Formation:** the goal is reducing leakage power while ensuring the circuit's performance is not impaired.
2. **Problem Formation:** the steps required to reach the goal are identified, like adjusting the Tox levels and deciding which parameters must be changed.
3. **Search Procedure:** a strategy is chosen to explore different states and find the best solution.

The Threshold Assignment Problem process includes five parts:

- **State:** Which shows the current setup of the circuit, including slack time, Tox, and leakage power.
- **Initial State:** All transistors start with the lowest Vth, which matches the minimum Tox. Since Vth (the threshold voltage) is directly affected by Tox (the oxide layer thickness), the smallest Tox leads to the lowest Vth.
- **Successor Function:** Creates new states by increasing the Tox level of transistor, making sure there are no time violations.
- **Goal Test:** It checks if no more valid changes can be made without causing timing violations.
- **Path Cost:** It measures the balance between reducing leakage and maintaining timing.

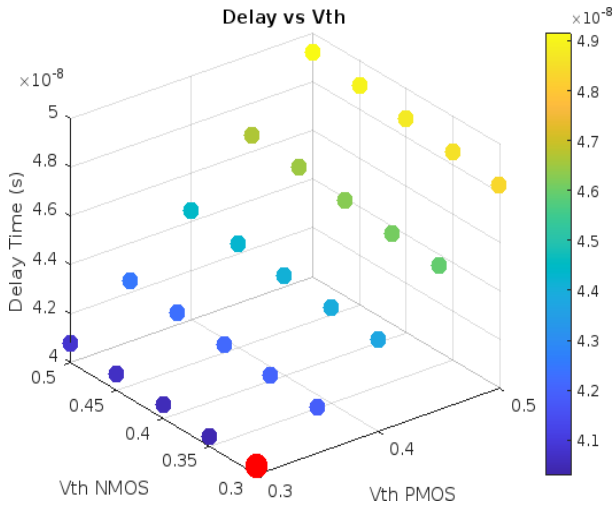


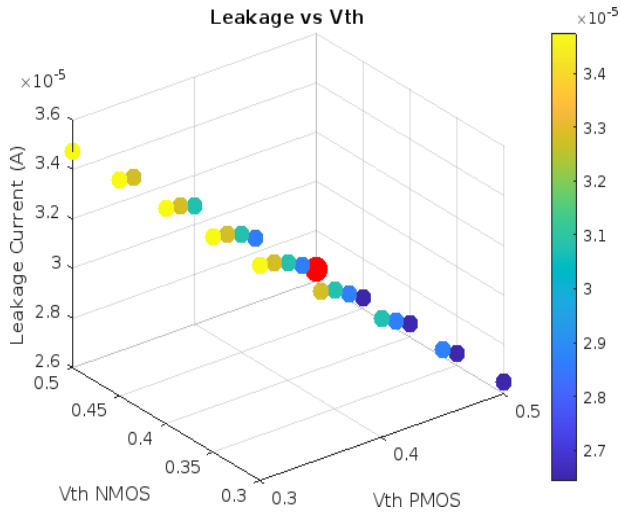
2.2 Solving the Threshold Assignment Problem

Threshold Assignment Problem (TPA) is a challenge where the goal is to reduce leakage power in a circuit by adjusting the Vth. However, increasing Vth optimize lower leakage but its cause timing issues, making the circuit might not perform as expected. The challenge is to find the right balance between minimizing leakage power and ensuring the circuit operates efficiently and reliably. An additional consideration in TAP is energy consumption which is affected by both leakage power and delay. Reducing leakage power helps save energy, but it can also cause delays in the circuit. Since energy depends on both power and time, solving TAP means finding a balance: lowering leakage power while keeping energy use low and ensuring the circuit works well.

To solve this problem, AI-based search algorithms are used. They help explore various configurations fast and efficiently, looking through all possible solutions to find the optimal Vth values that reduce leakage power without impacting performance.

In summary, TAP is solved using AI search methods to find the best balance between leakage power, energy use, and timing. This leads to better circuit designs that save energy, work well, and stay reliable.





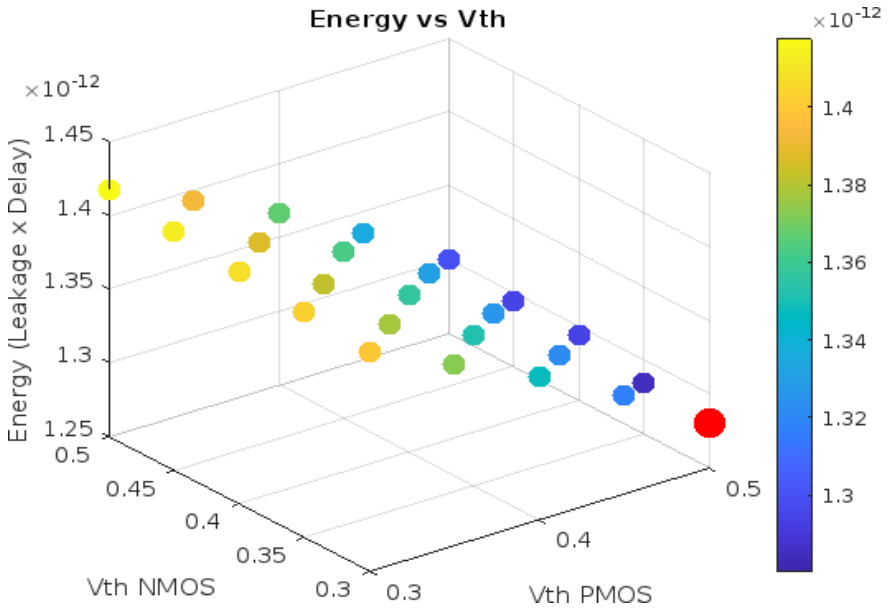


Figure 1: Effect of increasing Vth on delay, leakage and energy.

In the previous three plots we show how increasing the threshold voltage (Vth) affects leakage power, delay, and the energy in the circuit. As Vth increase, leakage power decrease as we show in the previous figure, because higher Vth reduce the current flowing. However, this come with trade-off: the circuit delay increases where the Vth is increasing that will slow down the transistors speed. The red points indicate to the optimal state that have the least leakage with the least delay, which is (0.5 for PMOS) and (0.3 for NMOS) which have delay equal to 4.0314e-08 and leakage equals to 2.6433e-05, so that the point that have the minimum delay is when Vth=0.3, and the point that have the minimum amount of leakage is when Vth=0.5. Energy, which is calculated by that formula:

$$\boldsymbol{Energy = delay \times leakage}.$$

The energy depends on both leakage power, and delay. As, while reducing leakage helps lower energy dissipation, The increase in delay can reduce the overall advantage. Raising the Vth is important to reduce power loss and make circuits more energy-efficient, which is especially needed in portable devices where battery life matters. The real challenge is finding the right Vth values that balance lower energy use while making sure the circuit still works properly, with acceptable performance limits.

2.2.1. ***STS algorithm***

The slack time of transistor refers to how much it speeds that can be reduced without affecting the entire circuit performance. The slack time of each transistor depends on its position in the circuit paths. If one transistor delay changes, it can affect the slack time of other transistors on the same path. This is shown in Figure 2. The total slack time for the circuit is initially 18 units. If transistor 1 loses 1 unit of time, transistor 2 will also

lose 1 unit, reducing the total slack delay to 16 units instead of 18 units. Also, if transistor 4 loses 1 unit of time transistors 3,5 and 6 will each lose 1 unit, delivering the slack time down to 14 units of time. In the same way, if the slack time of transistor 6 (which is connected to two paths [2-1] and [3-4-5]), is changed that will affect the slack times of transistors 1,2,3,4, and 5. On the other hand, changing the slack time of transistor 7 will not effect any other transistors as it does not share its path with others in the circuit.

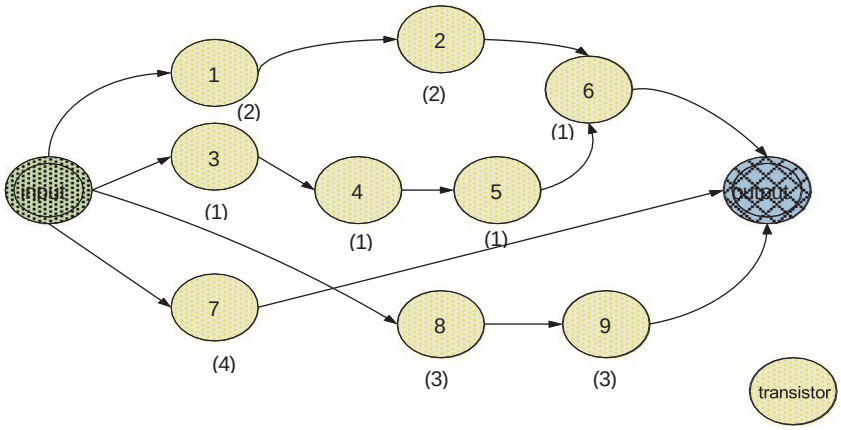
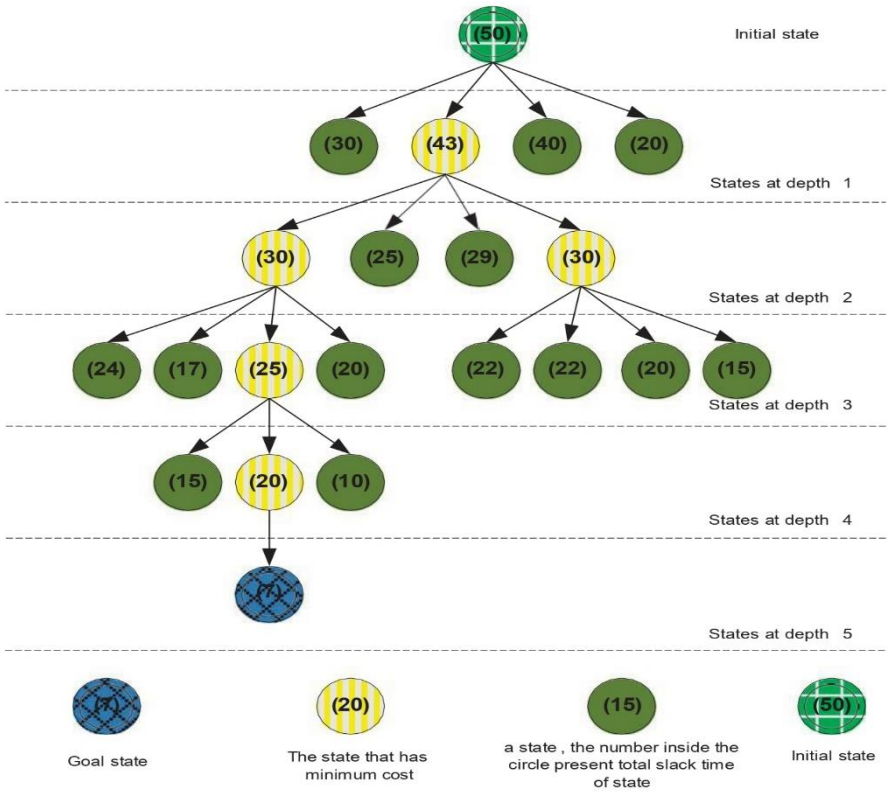


Figure 2: Effects of reducing slake time of the transistor based on total slack time of the circuit using STS Algorithm

Raising the Tox level of a transistor, helps reduce its leakage power, which also decrease the total leakage in the circuit. The more transistors with higher Tox, the more leakage power is reduced, especially at the deepest level of the search, where most transistors are changed. The STS algorithm improves the cost search by focusing on the states with the lowest cost, based on the total slack times. It uses a priority list to give the priority to these states and ignores ones that don't help find a solution. Figure 3 shows how the STS algorithm works.

Figure 3: Slack Time Search (STS) algorithm applied on simple search tree



2.2.2. *LPS algorithm*

The Leakage Power Search (LPS) algorithm is designed to help decreasing the leakage power in circuits by focusing on minimizing unwanted current that flows through transistors even when they are assuming to be off. This unwanted current, that is known as sub-threshold leakage current (I_{sub}), contributes to power loss in circuits. The LPS algorithm use a selfish best-first search strategy, that means that it explores the circuit setup which bring it closer to reducing leakage power. Figure 4 illustrates the logic operation of the Leakage Power Search (LPS) Algorithm.
The total leakage power of a circuit through summing up the leakage current (I_{sub}) of each transistor in the circuit. The leakage current for a transistor is determined by two factors: its maximum leakage current (I_{MAX}) and the probability of the transistor to be on which is ($\propto_{cont}$). The total leakage current of the circuited is calculated using that formula.

$$\boldsymbol{I_{sub}} = \sum^{\forall T \in C} \boldsymbol{I_{T_{MAX}}} \times \propto_{\boldsymbol{T\ cont}}$$

The probability of a transistor being on or off is important in deciding how much it adds to the total sub-threshold leakage current (I_{sub}) in the circuit. For example, consider four transistors. The (I_{sub}) of transistor P1 is significant only if transistor P2 and P3 are turned off and P4 is turned on. The probability of this scenario is calculated as:

$$P = \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{16}$$

Here, the probability of a transistor to being turned off is assumed to be 0.05. If transistors P2 and P3 are turned on, then V_{ds} of P1 will be zero, that result $I_{sub} = 0$. Whereas, if transistor P4 is turned off, V_{ds} off P1 will be equal VDD/2, which leads to a very small I_{sub}, often negligible.
As will as, the I_{sub} of transistor P4 become significant if any of transistor P1, P2 or P3 is turned on and the transistor P4 is turned off. The probability of that to be occur is:

$$P = 1 - \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{7}{16}$$

In this paper, various probability rules are used to determine the probability that $V_{DS} = V_{DD}$ for each transistor. This probability is referred to as the contribution factor (T_{cont}).
The contribution of each transistor to the total I_{sub} for the entire circuit it is calculated by multiplying the contribution factor $\boldsymbol{T_{cont}}$ with its maximum leakage current I_{MAX} as we shown in that formula:

$$\boldsymbol{I_{sub}} = \boldsymbol{T_{cont}} \times \boldsymbol{I_{T_{MAX}}}$$

This LPS algorithm use the priority list to order and manage the different circuit states. States with the lowest leakage current (I_{sub}) are given higher priority and are explored first. This helps the algorithm to focus on the most optimal configuration and move toward decreasing the leakage power more effectively.

The LPS algorithm is evaluated based on some factors:

- **Completeness:** The algorithm ensures that it will find a solution, as it sequentially searches through all possible states.
- **Optimally:** Whereas the algorithm can find a solution, it may not always find the optimal configuration cause of selfish nature of the searches. So, the algorithm does not always guarantee the best possible solution.
- **Time Complexity:** The time complexity depends on the numbers of transistors and number of possible states. As the larger circuit, it will take more time to find a solution.
- **Space Complexity:** The space complexity is defined based on the

number of states the algorithm needs to store during the search process.

In summary, the LPS algorithm efficiently reduces circuit leakage power by giving priority to the low leakage states. Whereas it may not be possible to always give the optimal solution, but it is effective in reducing power quickly.

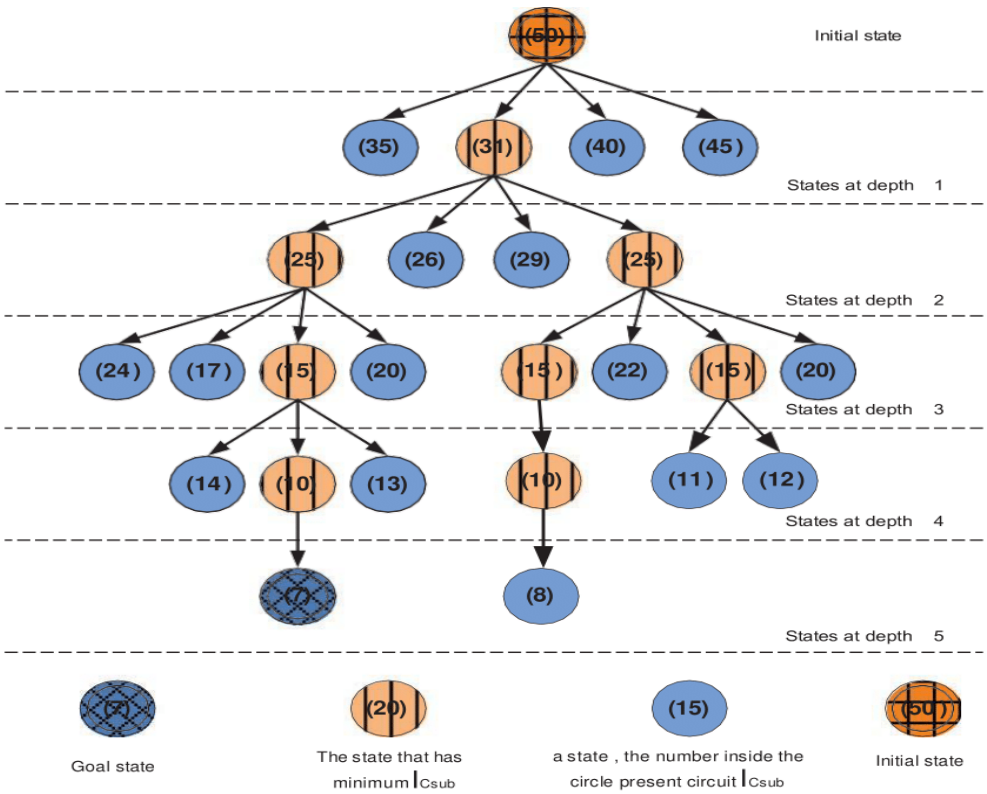


Figure 4: Leakage Power Search algorithm applied on simple search tree.

2.2.3. **LSS algorithm**

While solving a problem, like finding the best setup for a circuit, there are different strategies to explore. The first is the STS algorithm, which is an algorithm that looks for the easiest and cheapest option to move forward, focusing on minimizing the cost. The second strategy is the LPS algorithm, that focuses on reducing power loss, in the circuit, specifically trying to reduce the leakage current (unwanted power loss) by selecting states that reduce it.
The LSS algorithm combines both algorithms STS and LPS algorithm. It looks for a balance of both, the least cost and the lowest power loss. This makes the LSS algorithm smarter and more efficient at finding a good solution. The way this algorithm works is shown in Figure 12.

2.3 Solving Selection Problem

The selection problem is about finding the best threshold voltages (Vth) for a circuit. The threshold voltage decides when a transistor switches from low to high power. If

we increase the difference between Vth values, fewer transistors will switch. To solve that, we divide the Vth range into 11 levels and test different sets of that values. During each test, we will show how quickly the transistors thickness (Tox) changes. Once all the tests are done, we choose the best values that will give the most power save. This process is show in Figure 5.

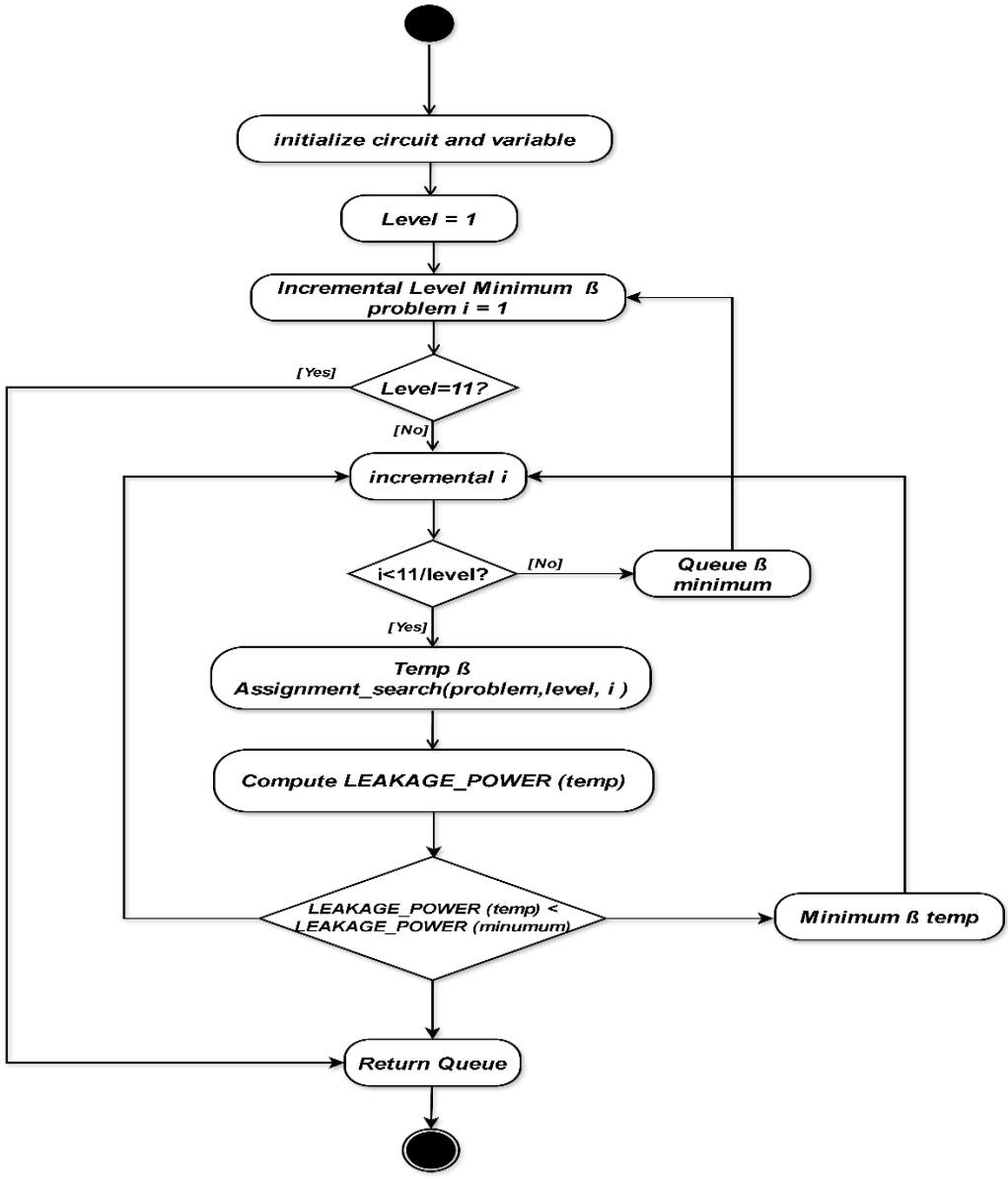


Figure 5: Flow graph of selection search algorithm.

3. SIMULATION RESULTS AND DISCUSSION

To assess how effectively the algorithms reduce sub-threshold leakage current (I_{sub}) at the transistor level, we use SPICE parameter from the 22nm Predictive Technology Model

developed by Barkeley. This allows us to compare the performance of CMOS transistors with and without multi-threshold voltages. The results are discussed in detail, emphasizing the effectiveness of the algorithms that we discussed before, in both cases (with and without multi threshold voltage). It is important to note that these algorithm are not limited only to 22nm technology and can be applied to any SPICE parameters. The main factors that influence the Vth are that the oxide layer thickness (Tox) and the channel doping concentration (NDEP). Adjusting to the NDEP is challenging, but changing the Tox is much easier. Changing the Tox value will affect the Vth, as well as while Tox is increased, the threshold voltage rises and the leakage current decreases. While the Tox is reduced, the Vth will decrease causing increasing at the leakage current. For our simulations, we choose Tox values for testing from 1.8nm and 2.8nm, which are the minimum and maximum values in the specified technology. We analyze how these changes in the threshold voltage will affect the leakage current and delay time of the transistors.

Figure 6: Contributions of transistor's topology in the sub-threshold leakage power.

3.1. *Overview* of simulation environment

In this simulation, we are choosing different values of threshold voltage(Vth). These values are calculated using advanced mathematical calculations called BSIM4, which are widely used to model transistor accurately. The Vth is a key parameter in MOSFEST operation, as it determines when the transistor begins conduct to current (ON state) or OFF. That formula shows how the Vth is computed:

$$Vth = \sqrt{0.8q\varepsilon_{si}N_{sub} + 2\,K_B T N_{sub}\varepsilon_{si}\ln(\frac{N_{DEP}}{n_i})}\;\; T_{ox} + V_{FB} + 0.4 + \frac{K_B T}{q}\,\ln(\frac{N_{DEP}}{n_i})\;\; (1)$$

Where:

- q: is the charge of an electron,
- ε_{si}: is the permittivity of silicon,
- N_{sub}: is the substrate doping concentration,
- K_B: are the Boltzmann constants,
- T:is the temperature,
- T_{ox}:is the oxide thickness,
- V_{FB}:is the flat-band voltage,
- N_{DEP}:is effective doping concentration,
- n_i:is the intrinsic carrier concentration of silicon, which depends on temperature and material properties.

So, as shown by the formula, the Vth of a MOSFEST is dependent on several factors:

1. ***Substrate Doping (N_{sub}) and Effective Doping (N_{DEP}):*** the amount of doping in the semiconductors affects how easily the transistor can turn on. While higher doping amount will lower the threshold voltage, as making the transistor on more easily, and vice versa, as it is demonstrated in Figure 7 and Figure 8.
2. ***Oxide thickness (T_{ox}):*** the thickness of the gate oxide has a significant effect, as well as thinner oxide layer make the transistor easily to turn on, which results in lower Vth, because it is the insulation layer between the gate and the semiconductor channel in MOSFEST, as it is demonstrated in Figure 9.
3. ***Flat-band voltage (V_{FB}):*** that is defined to be the voltage needed to align the energy levels in the semiconductor. As well as changing the material or doping, it can shift this voltage and affect the threshold voltage, as its demonstrated in Figure 10.
4. ***Temperature (T):*** higher temperatures increase the number of free charge carries (electrons or holes), which reduce the Vth, making the MOSFEST more sensitive to the gate voltage, as it is demonstrated in Figure 11.
5. ***Intrinsic Carrier Concentration (n_i):*** that refers to natural concentration of the charge carriers in the semiconductor, which increases with increasing the temperature, and the increasing of temperature reduce the Vth ,thus contributing to the reduction of the Vth, as it is demonstrated in Figure 11.

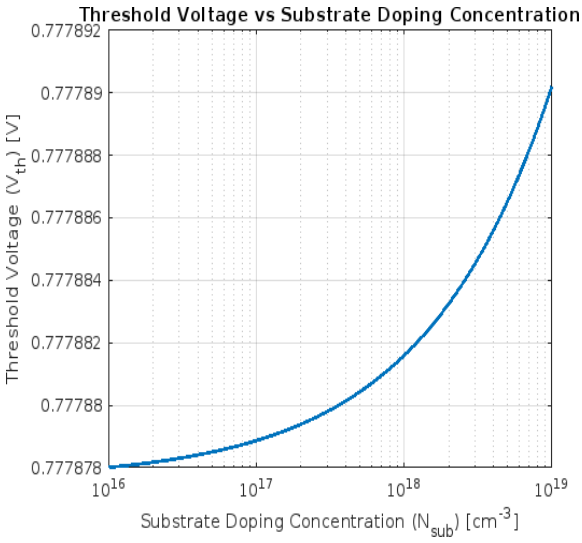


Figure7: effect of N_{sub} on threshold voltage

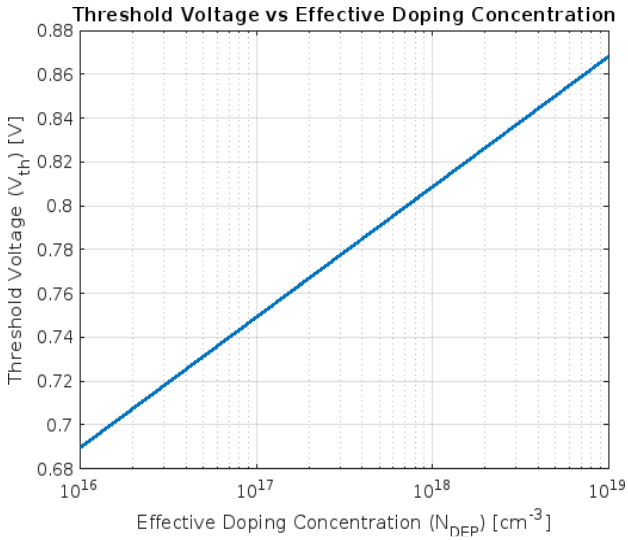


Figure8: effect of N_{DEP} on threshold voltage

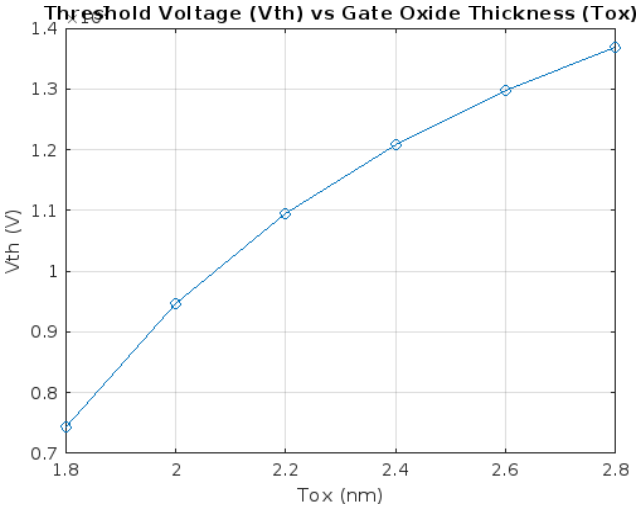


Figure9: effect of T_{ox} on threshold voltage

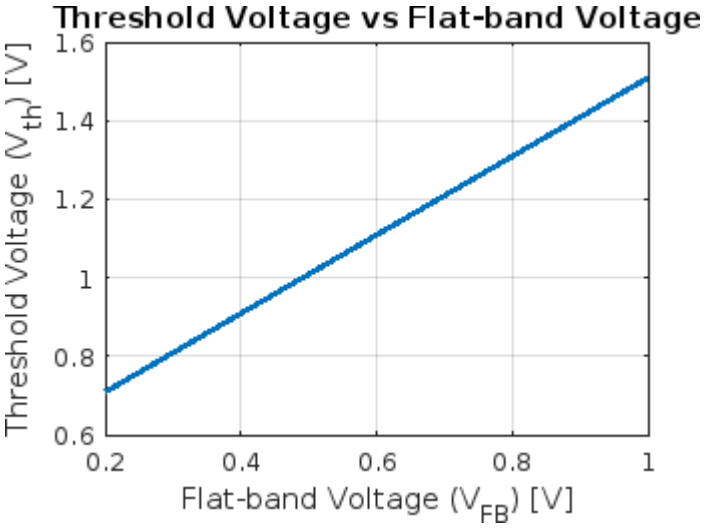


Figure10: effect of V_{FB} on threshold voltage

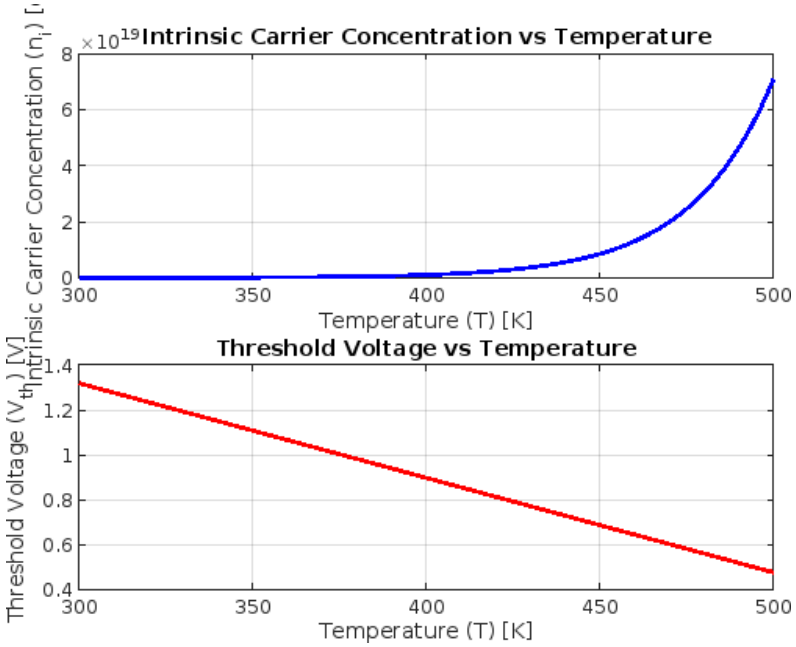


Figure 11: effect of temperature on Vth and the effect of temperature on (n_i).

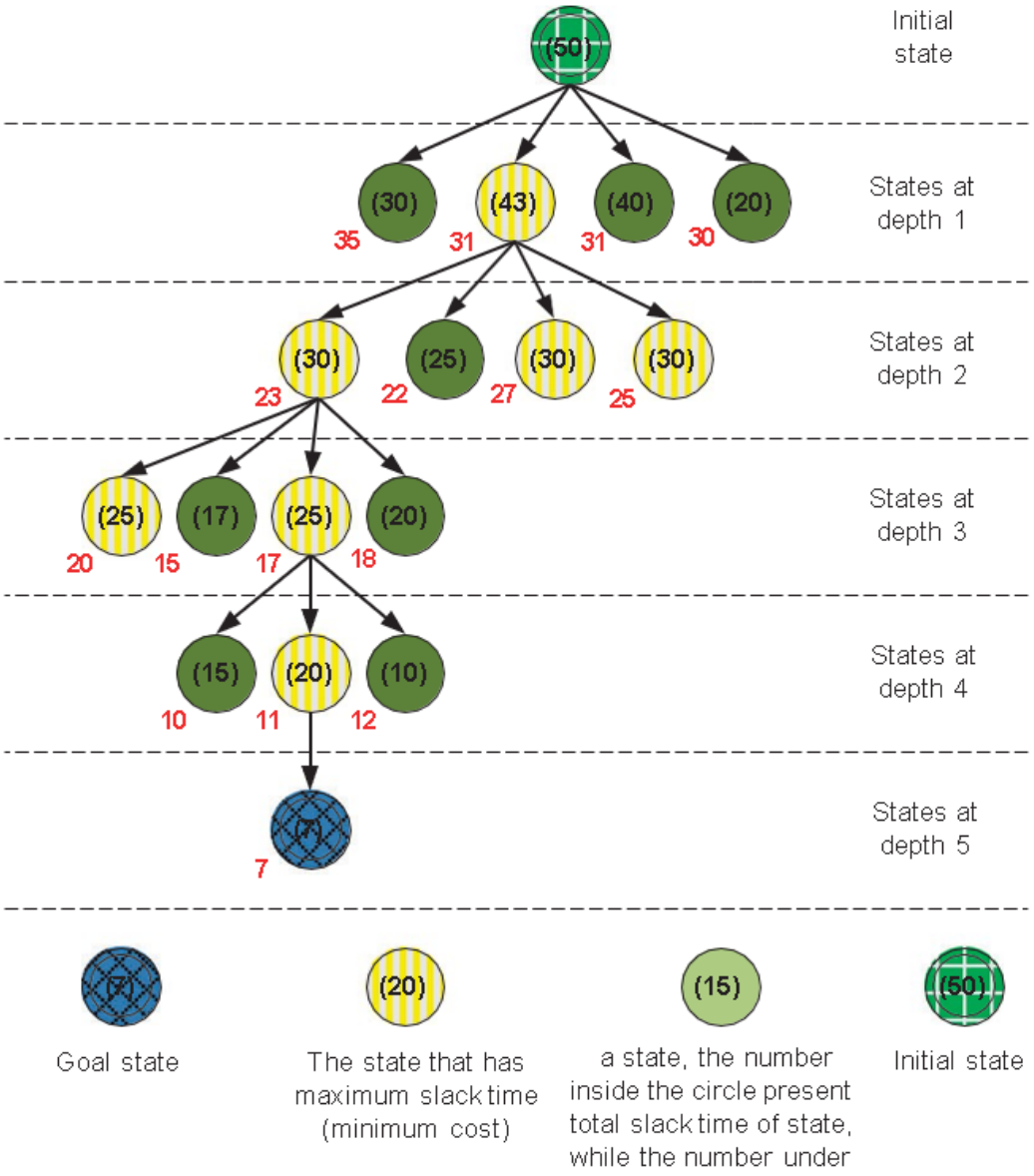


Figure 12: Leakage and Slack Search on simple tree. The algorithm at any depth expands the set states that have the minimum cost and I_{sub}.

At room temperature (T=300K), assuming the substrate-bias voltage (V_{bs}) is approximately equal zero, and the substrate doping (N_{sub}) is half of the effective doping concentrations ($\frac{N_{DEP}}{2}$), the threshold voltage can be computing using this formula:

$$V_{th} = \propto T_{ox} + \beta \qquad (2)$$

Where:

- T_{ox} : represent the oxide thickness,
- $\propto and\ \beta$: are constant defined by the following equations.

1. Constant $\propto$:

$$\propto = \sqrt{0.8\ q\ \varepsilon_{si} N_{sub} + 2K_B T N_{sub} \epsilon_{si} \ln(\frac{N_{DEP}}{n_i})} \qquad (3)$$

Where:

- q: is the charge of an electron,
- ε_{si}: is the permittivity of silicon,
- N_{sub}: is the substrate doping concentration,
- K_B: are the Boltzmann constants,
- T:is the temperature (300K),
- N_{DEP}:is effective doping concentration,
- n_i:is the intrinsic carrier concentration.

2. Constant β :

$$\beta = V_{FB} + 0.4 + K_B T \ln(\frac{N_{DEP}}{n_i}) \qquad (4)$$

Where:

- V_{FB}: is the flat-band voltage,
- The other variables are defined above.

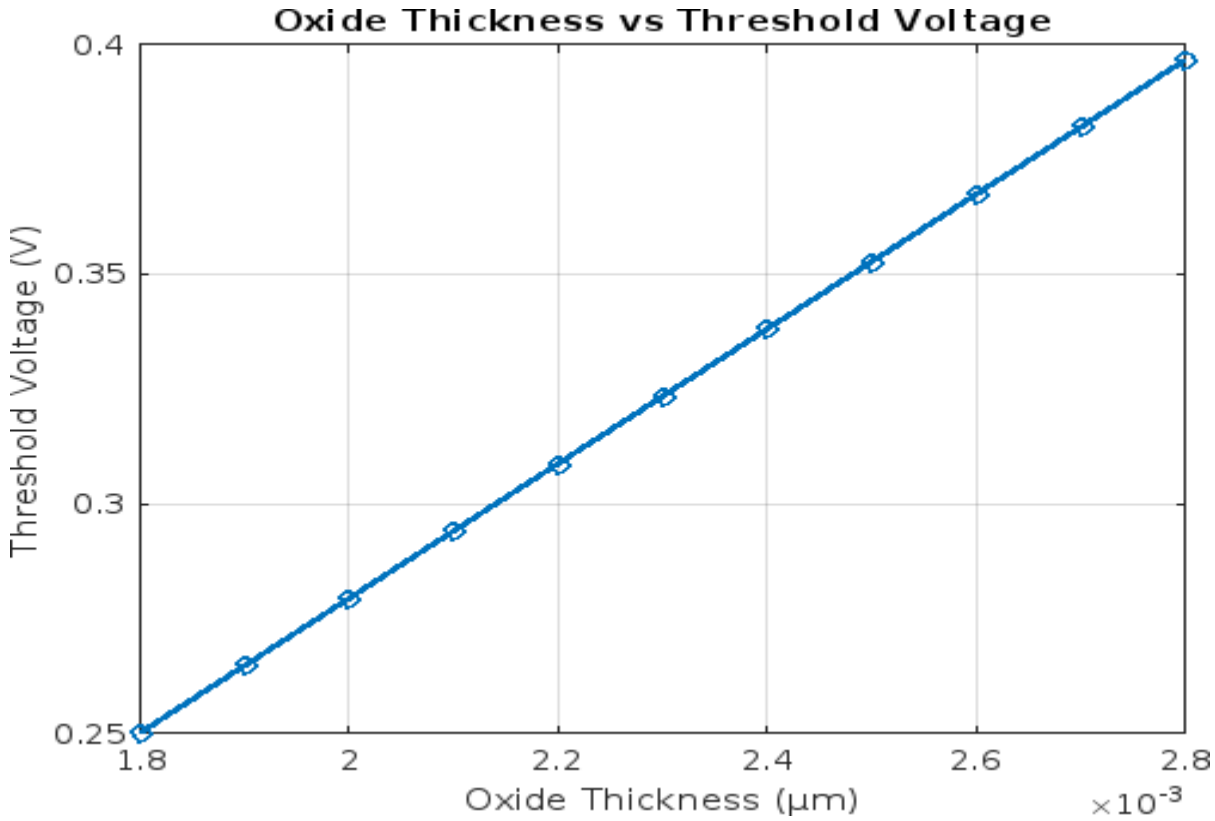


Figure 13: Vth vs Tox where ($N_{DEP} = 2.8 \times 10^{18} cm^{-3}$) and (T=300K)

Using the model parameters, $\propto = 1.467 \times 10^8$ and $\beta = -0.014$, the equation $V_{th=} \propto T_{ox} + \beta$ was applied to calculate the threshold voltage (V_{th}) for varying oxide thickness (T_{ox}). The results, shown in Figure 13, illustrate how V_{th} changes linearly with T_{ox}. Table I provides the corresponding T_{ox}values, their range, and the calculated V_{th} at room temperature (300K) for $N_{DEP} = 2.8 \times 10^{18}\ cm^{-3}$.

To computing delay times of a transistor, the BSIM4 delay equations are applied. These equations determine the time taken for transistors in NMOS and PMOS transistors. Specifically, the falling time delay, which is used for NMOS transistors, computed using the formula:

$$t_f = \frac{2C_L T_{ox} L}{\mu_p \varepsilon_{ox} W V_{DD}(1-\frac{V_{thn}}{V_{DD}})}\left[\frac{(\frac{V_{thn}}{V_{DD}}-0.1)}{1-\frac{V_{thn}}{V_{DD}}} + \frac{1}{2}\ln(19 - 20\frac{V_{thn}}{V_{DD}})\right] \quad (5)$$

For rising time delay, which applies to PMOS transistors, the calculation uses a similar formula but instead of using NMOS threshold voltage ($V_{thn)}$ it used NMOS threshold voltage (V_{thp}):

$$t_r = \frac{2C_L T_{ox} L}{\mu_p \varepsilon_{ox} W V_{DD}(1-\frac{V_{thp}}{V_{DD}})}\left[\frac{(\frac{V_{thp}}{V_{DD}}-0.1)}{1-\frac{V_{thp}}{V_{DD}}} + \frac{1}{2}\ln(19 - 20\frac{V_{thp}}{V_{DD}})\right] \quad (6)$$

Where:

- $\boldsymbol{C_L}$**:** is the load capacitance of the transistor.
- $\boldsymbol{T_{ox}}$: is the gate oxide thickness.
- $\boldsymbol{L\ and\ W}$: are the transistor channel length and width, respectively.
- $\boldsymbol{\varepsilon_{ox}}$**:** represents the permittivity of the silicon oxide.
- $\boldsymbol{V_{DD}}$ **:** is the supply voltage.
- $\boldsymbol{V_{thp}\ and\ V_{thn}}$**:** are the threshold voltage for the NMOS & PMOS transistor.
- $\boldsymbol{\mu_p\ and\ \mu_n}$ **:** are the electron mobility (for NMOS & PMOS transistors), which are equals to 0.0128,0.032 for holes & electrons.

To study how increasing the threshold voltage affects delay and sub-threshold leakage current (I_{sub}), a simple circuit was used as an example. The circuit have one PMOS transistor, one NMOS transistor, and a suitable load capacitance.

The transistor in this circuit has a channel length(L) of 22n, which is the smallest value specified by the predictive technology used. This small channel was chosen to improve the circuit performance and reduce the leakage current (I_{sub}). Additionally, the difference in mobility between electrons (in NMOS) and holes (in PMOS) was considered to determining the channel width(W) of the transistor. The setup of this circuit is shown in Figure 14.

level	$T_{ox}\,(m)$	$V_{th}\,(V)$
0	1.8×10^{-9}	0.250
1	1.9×10^{-9}	0.265
2	2.0×10^{-9}	0.279
3	2.1×10^{-9}	0.294
4	2.2×10^{-9}	0.309
5	2.3×10^{-9}	0.323
6	2.4×10^{-9}	0.338
7	2.5×10^{-9}	0.353
8	2.6×10^{-9}	0.367
9	2.7×10^{-9}	0.382
10	2.8×10^{-9}	0.397

Table1: The different levels of Tox and its equivalent Vth.

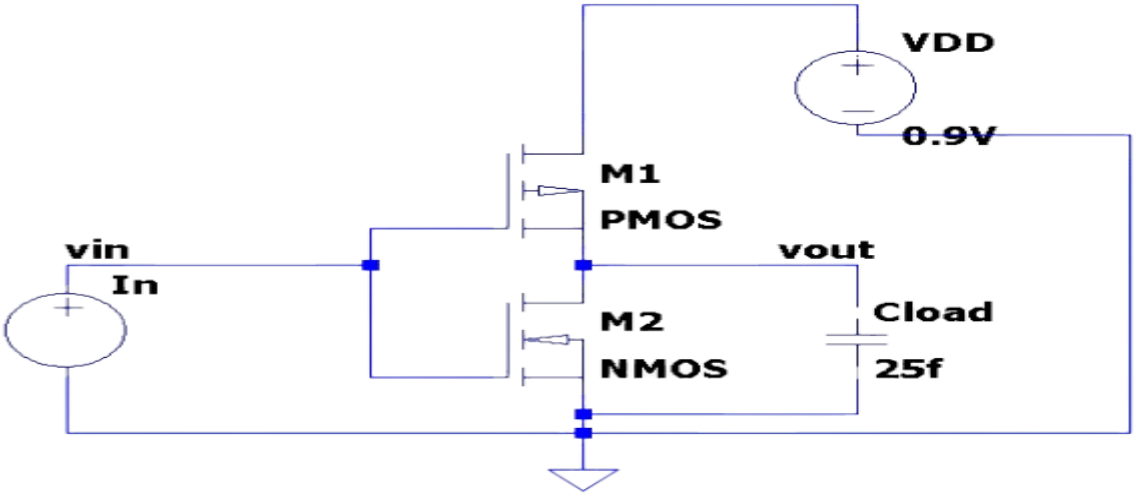


Figure 14: Circuit of Inverter logic gate.

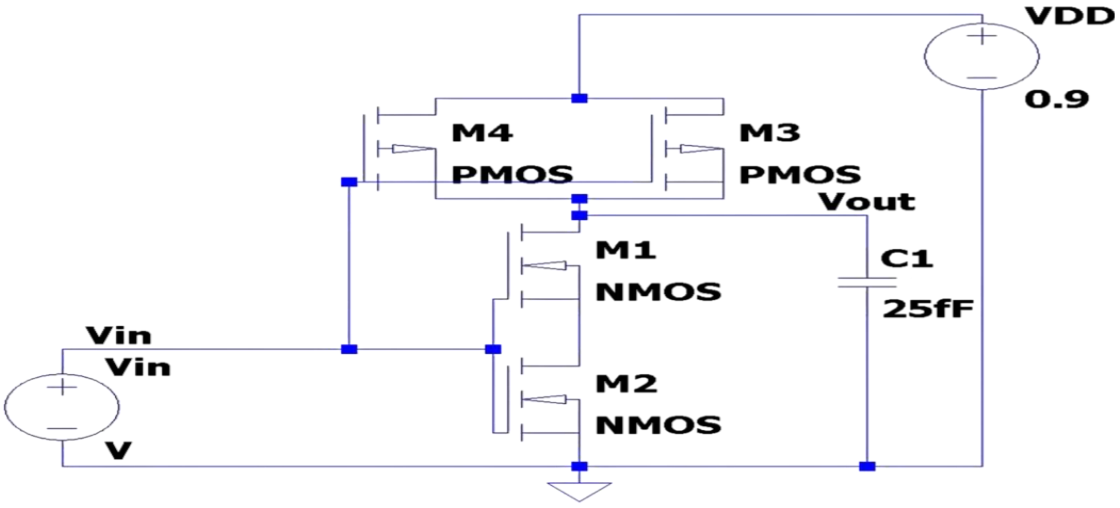


Figure 15: Circuit of NAND logic gate.

In simple terms, as the threshold voltage (Vth) increases in the circuit, the time it takes for the signal to change (delay time) gets longer, the effect of increasing the threshold voltage (Vth) on delay time and sub-threshold leakage current (I_{sub}) was investigated using a simple CMOS circuit. This circuit consists of a single PMOS transistor, a single NMOS transistor, and a suitable load capacitance. The channel length (L) of the transistors was chosen to be 22nm, the smallest value specified by the predictive technology in use. This small channel length was selected to improve circuit performance and reduce leakage current. Additionally, the difference in mobility between electrons (in NMOS) and holes (in PMOS) was considered to determine the optimal channel width (W) of the transistors.

As the threshold voltage increases, the delay time—the time it takes for the signal to change—also increases. For instance, when Vth is increased from 0.25V to 0.4V, the delay time of the circuit rises from 3.55ps to 7.02ps. This is due to the reduced current conduction in the transistors as the threshold voltage increases. At the same time, the sub-threshold leakage current, which flows through the transistors when they are supposed to be off, significantly decreases. This decrease in leakage current is beneficial for reducing power consumption in the circuit. However, the trade-off is that the circuit experiences a longer delay in signal transitions.

To extend the understanding of this trade-off, the study was applied to more complex logic gates, namely inverters and NAND gates. These gates are fundamental building blocks of CMOS circuits and provide essential logic functions.

- **Inverters**: An inverter consists of a single PMOS and NMOS transistor. When the threshold voltage is increased in an inverter circuit, the delay time increases because the transistors' conductivity is reduced. Despite the increase in delay, the sub-threshold leakage current is minimized, leading to lower power consumption. This effect is particularly useful in low-power applications where reducing leakage current is essential, even if it results in a slower signal transition, the setup configuration for the inverter is shown in Figure 14.
- **NAND Gates**: A NAND gate, which consists of multiple PMOS and NMOS transistors, behaves similarly to the inverter when Vth is increased. The delay time increases, but the leakage current decreases. This trade-off becomes even more significant in larger logic circuits, such as the Full Adder, which consists of 28 transistors. Here, careful optimization of the threshold voltage for each transistor is necessary to achieve a balance between power consumption and performance, the setup configuration for the NAND transistor is shown in Figure 15.

In this study, a simulation was set up and executed in LTspice to analyse the impact of varying threshold voltage (Vth) on the delay time, leakage current, and energy consumption of CMOS circuits, specifically focusing on inverters and NAND gates. The simulation was conducted with a supply voltage (VDD) of 0.9V, and a range of

threshold voltages (Vth) was considered for both the PMOS and NMOS transistors and many other parameters.

Setup and Simulation for Inverter and NAND circuits

The circuit for the inverter and NAND gates was carefully designed in LTspice, with the main parameters being the threshold voltage (Vth) of the transistors, the supply voltage (VDD=0.9V), and the operating conditions. For each simulation run, the threshold voltage was varied across a set of values, ranging from 0.25V to 0.4V, to explore how these variations affect the performance of the circuits.

The delay time in the simulation was measured by observing the time it took for the output signal to transition from low to high (or vice versa) after a change in the input signal. This delay time is a critical performance metric, as it directly affects the speed of the circuit. As the threshold voltage increases, it becomes harder for the transistors to conduct when they are supposed to be "on," thus increasing the delay time. On the other hand, a lower Vth improves switching speed but leads to higher leakage current.

Leakage current, specifically sub-threshold leakage current, was also measured in the simulation. This current flows through the transistors when they are intended to be "off," contributing to power consumption even in the idle state. A higher threshold voltage typically reduces the sub-threshold leakage, leading to a reduction in overall power consumption.

Energy consumption was calculated by multiplying the leakage current by the delay time. This product provides insight into the dynamic energy usage of the circuit, considering both the leakage current and the delay time. This relationship is important for assessing how increasing Vth affects the overall energy efficiency of the circuit.

After running the LTspice simulations, the data collected for delay time, leakage current, and energy consumption at various Vth values were imported into MATLAB for further analysis. MATLAB was used to visualize the results through scatter plots that demonstrate the relationships between the threshold voltage and key performance metrics.

1. **Delay vs. Vth**: A scatter plot was generated to show how the delay time increases as the threshold voltage increases. As expected, a higher Vth leads to a longer delay, which indicates slower switching speeds in the circuit.
2. **Leakage vs. Vth** : Another plot was created to illustrate how the leakage current decreases with an increase in Vth. As the threshold voltage increases, the sub-threshold leakage current reduces, thereby improving the power efficiency of the circuit.
3. **Energy vs. Vth**: A third plot was generated to visualize the energy consumption. The energy, calculated as the product of leakage current and delay time, is shown to have a trade-off: increasing Vth reduces leakage but increases delay time, affecting overall energy usage.

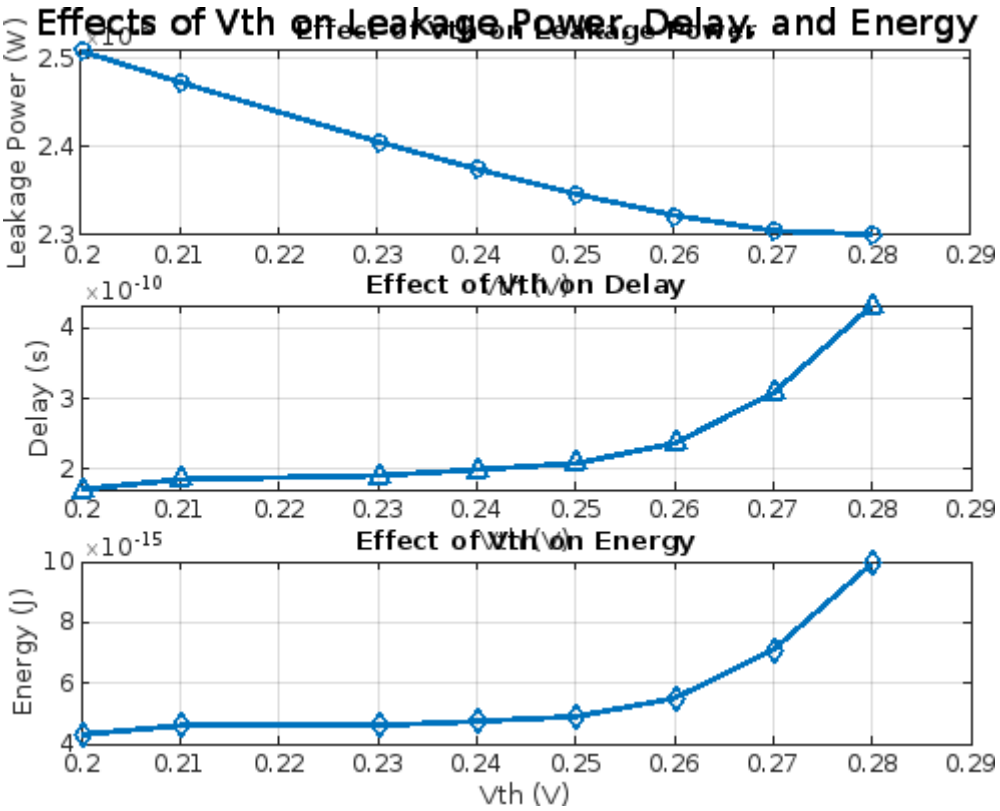


Figure 16: That figure demonstrates the delay, leakage and energy vs Vth for NAND circuit.

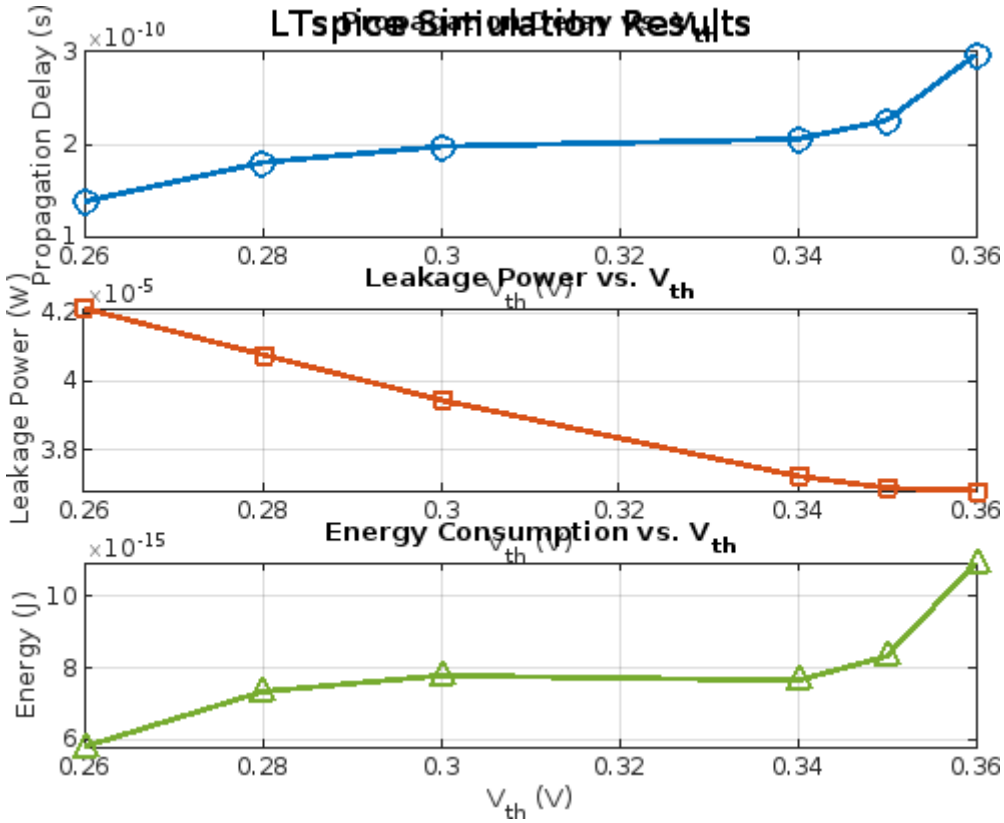


Figure 17: That figure demonstrates the delay, leakage and energy vs Vth for NAND circuit.

The results of the simulations show a clear trade-off between delay time, leakage current, and energy consumption. As the threshold voltage increases, the delay time also increases due to reduced transistor conductivity, which results in slower switching times. However, this comes at the benefit of reduced leakage current, which leads to lower power consumption in the idle state.

The energy consumption curve demonstrates that while increasing Vth reduces leakage, the corresponding increase in delay time can offset the benefits in terms of energy efficiency. This highlights the need for careful optimization of the threshold voltage to balance both speed and power consumption, particularly in low-power or high-performance applications.

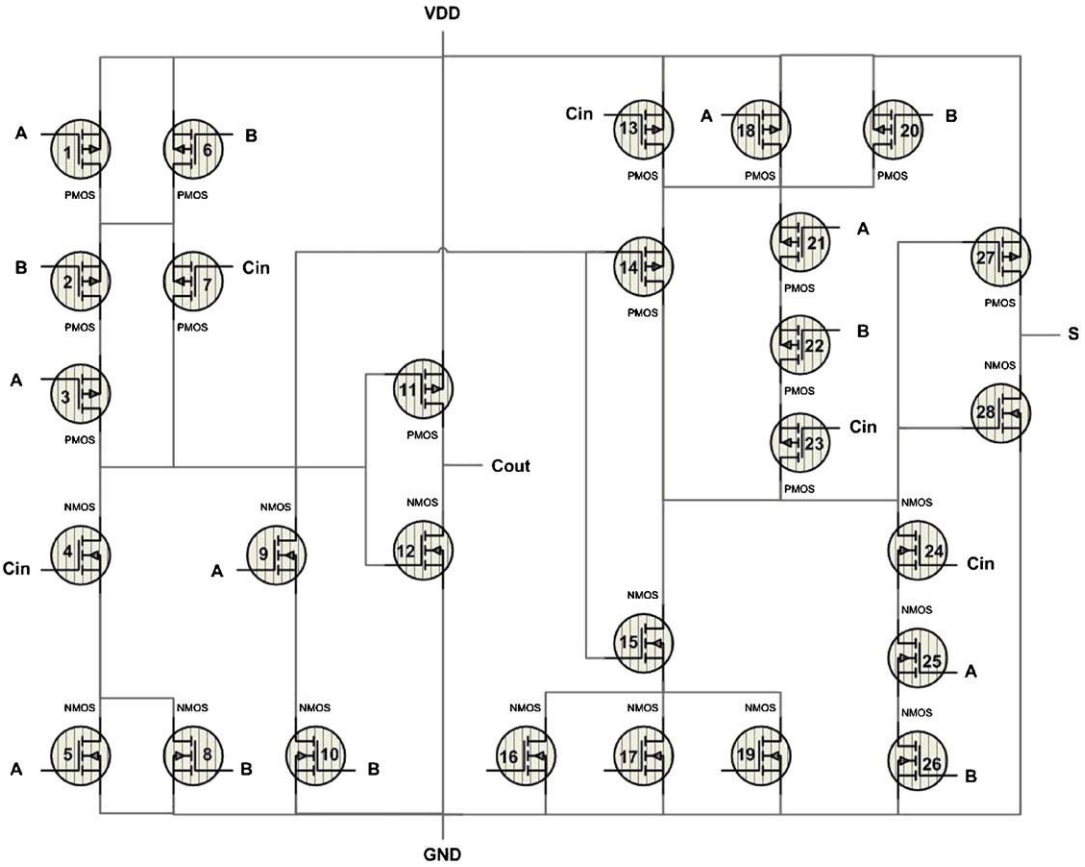


Figure 18: Conventional Full Adder in transistor level.

Figure 19. the variation mobility between NMOS and PMOS transistors at different Vth.

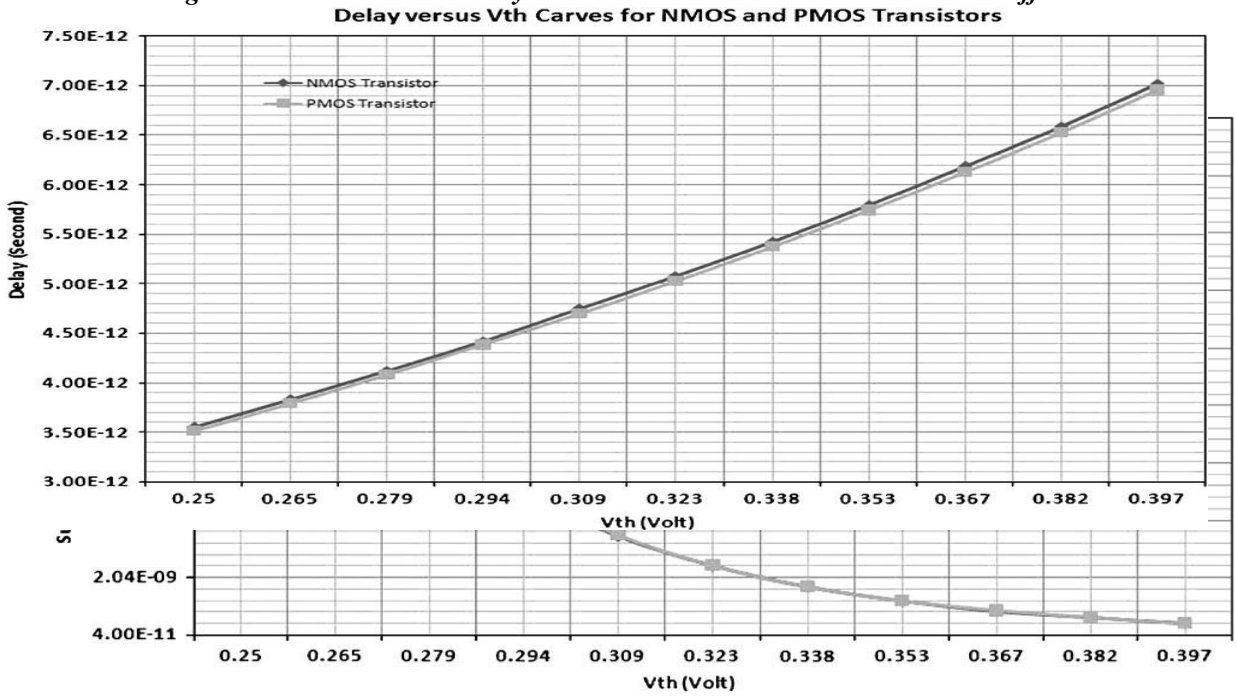


Figure 20. NMOS and PMOS transistors I_{sub} at different Vth.

In the case of a Full Adder circuit, the circuit was analysed across various Vth levels using a C++ program to estimate the leakage current (I_{sub}) and delay for each transistor. The results, shown in Figures 19 and 20, reveal that as Vth increases, the delay time increases exponentially, while the subthreshold leakage current decreases significantly.

Figure 19 demonstrates that the delay time for both NMOS and PMOS transistors rises with Vth. For example, at Vth= 0.25V, the delay is 3.55 ps, and at Vth=0.4V, it increases to 7.02 ps. The delay time is exponentially proportional to Vth, with a 7.6% increase in delay when Vth goes from 0.25 V to 0.265 V, and a 9.2% increase when it goes from 0.382 V to 0.397 V.

Figure 20 shows that I_{sub} decreases exponentially as Vth increases. For example, the leakage current for the NMOS transistor decreases by 32.5% when Vth increases from 0.25 V to 0.265 V, and by 27% when it increases from 0.382 V to 0.397 V. To evaluate the proposed algorithms, a conventional Full Adder circuit consisting of 28 transistors was tested. The circuit uses a fixed channel length of 22 nm to reduce transistor size, increase speed, and minimize dynamic power dissipation. The transistor widths were optimized for balance between rising and falling times in the circuit, as shown in Figure 18.

3.2. *Comparison results*

The multi-threshold voltage technique effectively reduces leakage power in CMOS circuits, especially in transistor level. Optimal results depend on carefully choosing the right threshold voltage levels to reduce the leakage current while maintaining delay timing constraint. Simulations show that the optimal (ideal) values of threshold voltage levels vary by circuit, balancing power saving and cost (delay). This variation arises from differences in transistor contributions to leakage and circuit topology, which create unique critical paths. The proposed algorithms identify the most suitable voltage levels for each design. Using Full Adder, NAND, and Inverter as examples, the algorithms were tested with different threshold levels. The Leakage Power Search (LPS) and Leakage and Slack Time Search (LSS) algorithms achieved better leakage power optimization than the Slack Time Search (STS) algorithm, which explores deeper search trees and replaces more transistors. Test also showed significant energy savings with two threshold oxide levels. Overall, the algorithms adaptively optimize leakage power, with LPS and LSS providing a better balance of power efficiency and cost compared to STS.

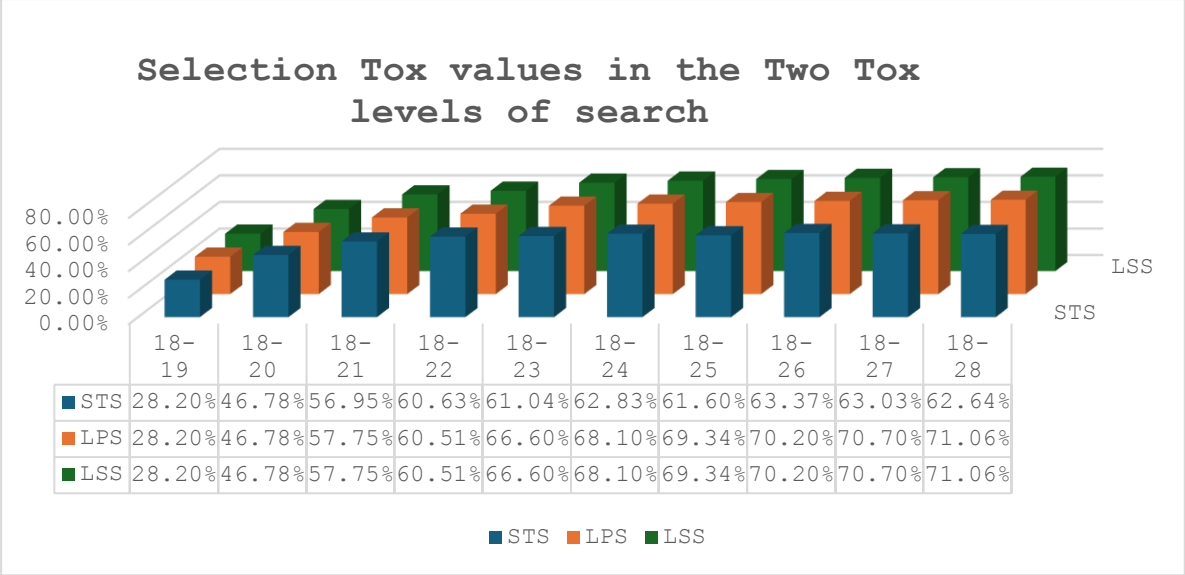


Figure 21: Percentage of power saved with different T_{ox} values in dual-T_{ox} search from STS, LPS, and LSS algorithms

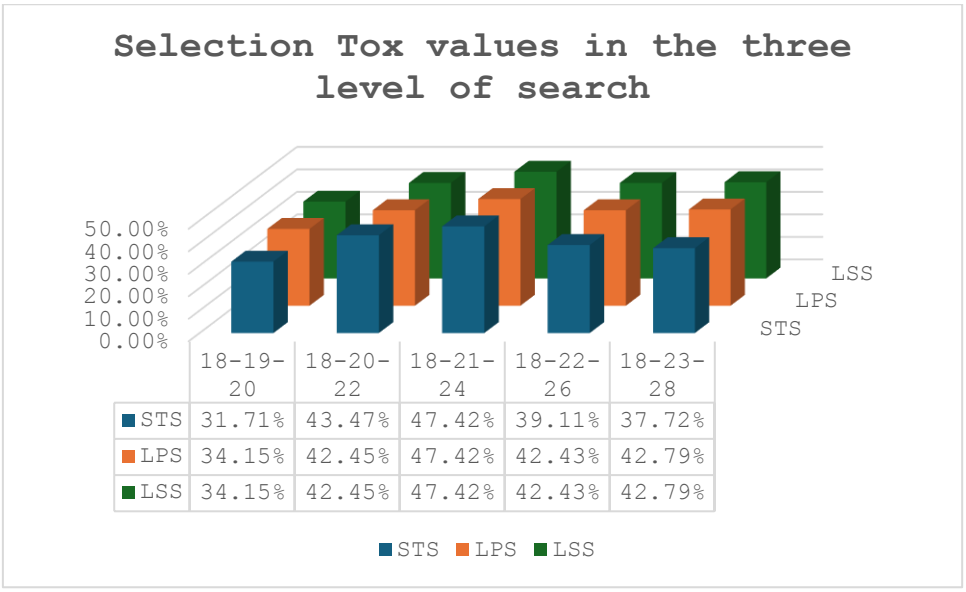
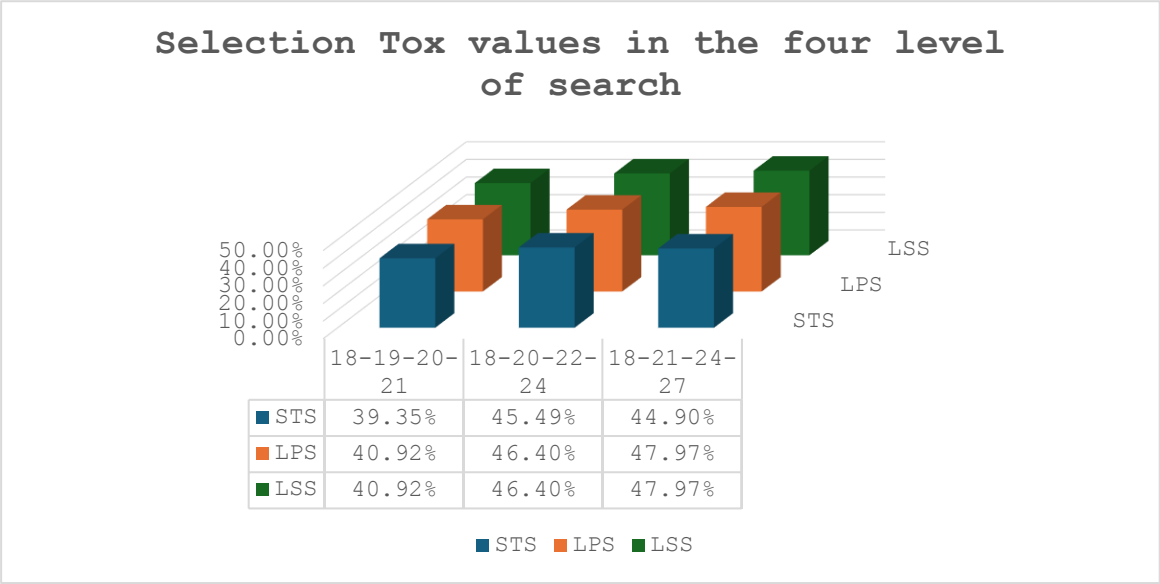
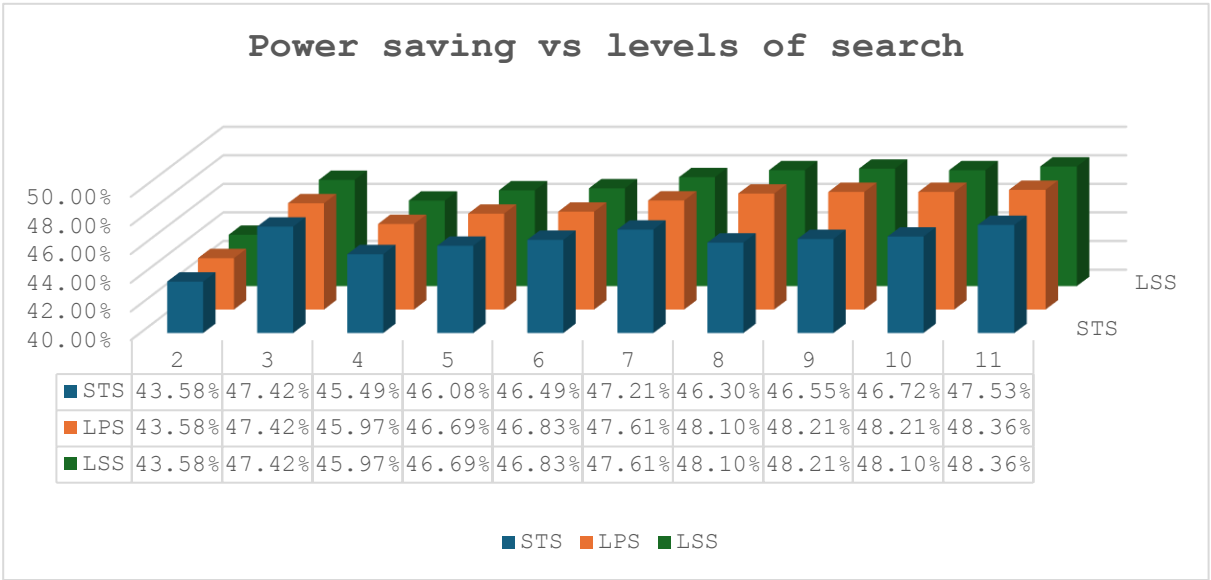


Figure 22: Percentage of power saved with different T_{ox} values in three T_{ox} levels search from STS, LPS, and LSS algorithms.



	18-19-20-21	18-20-22-24	18-21-24-27
STS	39.35%	45.49%	44.90%
LPS	40.92%	46.40%	47.97%
LSS	40.92%	46.40%	47.97%

Figure 23: Percentage of power saved with different T_{ox} values in four-T_{ox} levels search from STS, LPS, and LSS algorithms.



	2	3	4	5	6	7	8	9	10	11
STS	43.58%	47.42%	45.49%	46.08%	46.49%	47.21%	46.30%	46.55%	46.72%	47.53%
LPS	43.58%	47.42%	45.97%	46.69%	46.83%	47.61%	48.10%	48.21%	48.21%	48.36%
LSS	43.58%	47.42%	45.97%	46.69%	46.83%	47.61%	48.10%	48.21%	48.10%	48.36%

Figure 24: The maximum leakage power reduction results from STS, LPS, and LSS algorithms for each number of T_{ox} levels.

.

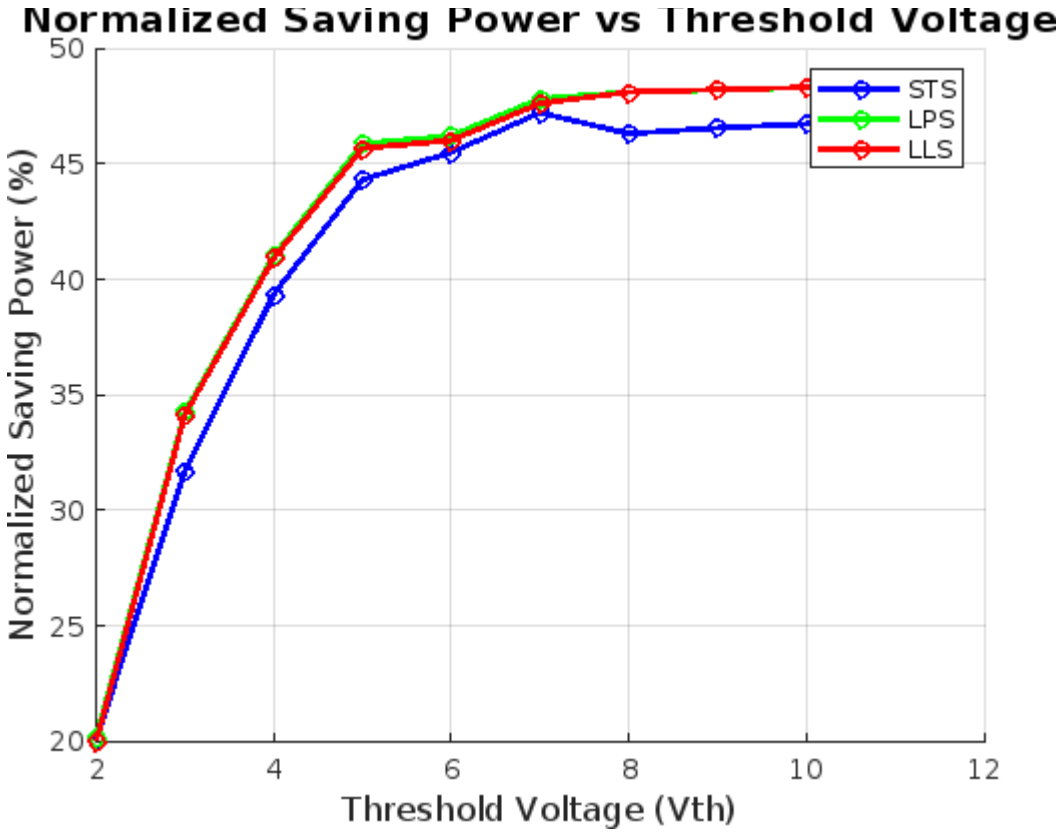


Figure 25: The maximum leakage power reduction results from STS, LPS, and LSS algorithms for each number of T_{ox} levels.

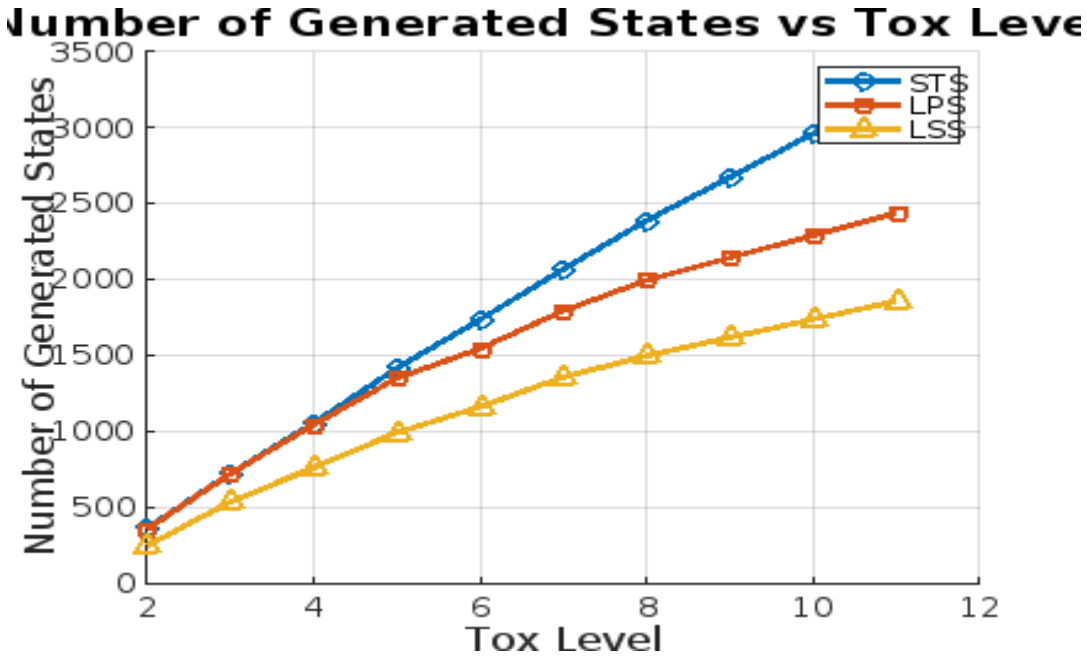


Figure 26: Number of generated state versus level of search in STS, LPS, and LSS algorithms.

	2	3	4	5	6	7	8	9	10	11
STS	362	721	1052	1414	1734	2067	2386	2670	2958	3238

LPS	*355*	*720*	*1042*	*1348*	*1543*	*1792*	*1993*	*2141*	*2289*	*2437*
LSS	*246*	*534*	*760*	*990*	*1160*	*1356*	*1496*	*1616*	*1736*	*1856*

Table 2: Number of generated state versus level of search in STS, LPS, and LSS algorithms.

Level	STS	LPS	LLS
2	1.8-2.6	1.8-2.8	1.8-2.8
3	1.8-2.1-2.4	1.8-2.1-2.4	1.8-2.1-2.4
4	1.8-2.0-2.2-2.4	1.8-2.1-2.4-2.7	1.8-2.1-2.4-2.7

Table 3: The optimal T_{ox} (nm) values in three levels and four levels using three proposed algorithms.

Level	STS (%)	LPS (%)	LLS (%)
2	20.9	20.20	20.02
3	31.71	43.25	34.15
4	39.35	41.02	40.91
5	44.31	45.87	45.67
6	45.49	46.21	46.01
7	47.21	47.81	47.61
8	46.30	48.10	48.10
9	46.55	48.20	48.21
10	46.72	48.30	48.30
11	47.53	48.31	48.36

Table 4: Normalized Saving power of the three algorithms vs different levels of threshold voltage

Figures 22 and 23 illustrate the optimization results for the Full Adder circuit using three and four voltage threshold levels. Both the LPS and LSS algorithms outperform STS in terms of leakage current optimization. Additionally, Figure 21 shows the percentage of saved power with two Tox levels, highlighting the power savings achieved by LPS and LSS. Figure 26 compares the number of states generated by STS, LPS, and LSS, showing that STS generates more states than LPS and LSS, as the search space in STS is broader.

Figure 24 and 25 column or as a MATLAB plot displays the maximum leakage power reduction achieved by each algorithm for different Tox levels, with LPS and LSS providing consistent results. These algorithms demonstrate superior performance in leakage power saving compared to STS, which yields less favourable results. The optimal Tox values for three, four, and two levels of threshold voltage, which minimize leakage current, are summarized in Table II.

The normalized leakage power savings relative to the conventional Full Adder design are presented in Table III, showing that LPS and LSS provide better leakage power savings than STS. Simulation results reveal that the minimal leakage current for the Full Adder circuit is achieved with three Tox levels using the STS algorithm, whereas LPS and LSS achieve minimal leakage current with four Tox levels. However, the difference in power savings between three and four levels of voltage threshold is not

significant enough to justify the additional cost of extra Tox levels. Therefore, the best leakage current reduction is achieved using three levels of Tox, specifically 1.8, 2.1, and 2.4 mV.

In conclusion, the position of the transistors within the leakage current path plays a crucial role in leakage power reduction, meaning that it is unnecessary to change the threshold voltage of all transistors to achieve significant savings. The LPS and LSS algorithms provide more accurate leakage power reduction compared to STS, as STS tends to explore deeper into the search tree, which doesn't always lead to minimal leakage power. These differences are attributed to the structure of the algorithms and the number of search loops implemented.

The percentage of saved power depends on the circuit design, where the timing differences between paths determine the power savings achievable through the multi-threshold voltage technique. For example, the LLS algorithm applied to the C17 circuit from the ISCAS benchmark reduces sub-threshold leakage current by approximately 72% without degrading performance.

It is important to note that the simulations were conducted under predictive SPICE parameters, and discrepancies between simulation results and real-world performance may occur due to factors such as inaccuracies in device models, parasitic resistances, and capacitances. These challenges are particularly pronounced in nanotechnology, which may benefit from new tools such as Monte Carlo simulators to predict and mitigate these errors in the future.

In summary, LPS and LSS consistently provide superior results for power saving, while STS offers deeper search states without necessarily minimizing leakage. Each algorithm behaves differently due to its structure and search methods, emphasizing the importance of tailored approaches for specific circuit designs.

4.CONCLUSION

In this paper, three new algorithms were introduced to address the Threshold Assignment Problem (TAP): the Slack Time Search Algorithm (STS), the Leakage Power Search Algorithm (LPS), and the Leakage and Slack Time Search Algorithm (LSS). These algorithms are designed to reduce leakage current at the transistor level. To solve the Threshold Selection Problem (TSP), each algorithm is run multiple times with different numbers of Tox levels, aiming to find the optimal Tox that minimizes leakage current while keeping the number of levels as low as possible. The simulation results highlight that the position of transistors in the leakage current path is the key

factor in reducing leakage power. This means it's not always necessary to adjust the threshold voltage of most transistors to achieve significant power savings. Among the algorithms, LSS performed the best, delivering the lowest leakage power and fastest calculation of Tox values. LPS produced similar results to LSS but took more time to calculate. The paper suggests that this approach is useful for future applications, especially in the context of nanotechnology-scale parameters.

REFERENCES

1. Rabaey J. Low Power Design Essentials. Springer: Berlin, 2009.

2. Liu W, Nannarelli A. Techniques for Leakage Power Reduction in Nanoscale Circuits: A Survey: Informatics and Mathematical Modeling. Technical University of Denmark, DTU, 2007.

3. Trivedi S. Cahay BM. Switching voltage, dynamic power dissipation and on-to-off conductance ratio of a spin f ield effect transistor. IEEE Transaction on Circuits, Devices and Systems, IET 2007; 1(6):395–400.

4. Orshansky WM. Estimation of leakage power consumption and parametric yield based on realistic probabilistic descriptions of parameters. Journal of Low Power Electronics 2007; 4:1–12.

5. Rao R, Devgan A, Blaauw D, Sylvester D. Analytical yield prediction considering leakage/performance correlation. IEEE Transactions on CAD 2006; 25(9):1685–1695.

6. Ramalingam A. Devgan DP. Wakeup Scheduling in MTCMOS circuits using successive relaxation to minimize ground bounce. Journal of Low Power Electronics 2007; 3(1):28–35.

7. Tawfik SA, Kursun V. Multi-threshold voltage FinFET sequential circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 2009; 99:1–5.

8. Srivastava, Simultaneous Vt selection and assignment for leakage optimization. Proceedings of ISLPED'03, Seoul, Korea, August 2003; 25–27.

9. Sill F, Grassert F, Timmermann D. Total leakage power optimization with improved mixed gates. Proceedings of the 18th Annual Symposium on Integrated Circuits and System Design,Florian´ opolis, Brazil, 2005; 154–159.

10. Itoh K, Hitachi F. Leakage- and variability-conscious circuit designs for the 0.5-V nanoscale CMOS era. Proceedings of the ISLPED, TX, U.S.A., 2009; 273–274

11. Yang W, Dunga M, BSIM4.6.1 MOSFET Model, User Manual, Available from: http://www-device.eecs.berkeley. edu/~bsim3/bsim4.html [Accessed on 19/12/2008].

12. Joobbani R. An Artificial Intelligence Approach to VLSI Routing. Kluwer International Series in Engineering and Computer Science, Kluwer Academic Publishers: Boston, MA, 2007.

13. Bang-Jensen J, Gutin G. Digraphs: Theory, Algorithms and Applications (2nd edn). Springer: London, 2008.

14. Kuroda R et al. Circuit level prediction of device performance degradation. Journal of Microelectronics Reliability, Science Direct 2007; 47:930–936.