Optimization of Logic Paths for CMOS-Based Dual Mode Logic Gates

R. Tharun Vishnu Vardhan, PG Student
(M.Tech-Dsce); Rajeev Gandhi Memorial College Of Engineering And Technology, Nandyal, Kurnool (Dist.)

Dr. D. Satya Narayana, Professor
Rajeev Gandhi Memorial College Of Engineering And Technology, Nandyal, Kurnool (Dist.)

Abstract: The project proposes to develop a simple method for minimizing delays and achieving an optimized number of stages in logic paths containing CMOS-based DML gates. This project offers three different approaches (1) Complete Approximated (CA) Method, (2) Complete Un-Approximated (CS) Method and (3) partially/Semi Approximated (SA) Method, which tradeoff between complexity, computation and accuracy. The proposed optimization is shown for the dynamic mode of operation. Theoretical mathematical analysis is presented and efficiency of the proposed methodology is shown in a standard 32 nm CMOS process.

Key Words: - Dual Mode Logic (DML), CMOS, High Performance, Logical Effort, Low Power and Optimization.

I. INTRODUCTION

The basic tasks for digital circuit designers are logic optimization and timing estimations. It was Sutherland, who first presented the logical effort (LE) method, for easy and fast assessment and optimization of delay in CMOS logic paths. The LE method has developed as a very widely held tool for designing and education purposes because of its elegance and is adopted to be the basis for several computer-aided-design tools. Granting LE is mainly used for standard CMOS logic, it is also shown to be useful for other logic families, such as the pass transistor logic.

The novel dual mode logic (DML), which provides the designer with a very high level of flexibility, was suggested. It allows on-the-fly switching amid two modes of operation: 1) static and 2) dynamic modes. In the static mode, DML gates accomplish very low power dissipation, with some deprivation in performance, as compared with standard CMOS. On the other hand, dynamic operation of DML gates attains very high speed at the expense of augmented power dissipation.

A elementary DML gate is composed of any static logic family gate, which can be a conventional CMOS gate, and an extra transistor. DML gates have a very simple and intuitive structure, requiring unconventional sizing methodology to attain the preferred performance. Conventional LE methodology cannot be used with the DML family as it does not contemplate its unconventional sizing rules and topology.

The objective of this project is to develop a humble method for minimizing delays and achieving an optimized number of stages in logical paths containing CMOS-based DML gates. An integrated LE method is introduced for the delay evaluation and optimization of logic paths built with DML logic gates. DML-LE responses complete (un-approximate) design problems, which can be resolved numerically, and streamlines these problems to a straightforward and easy computational problematic [approximate and semi-approximate (SA)] solutions with a unified analytic model. Through this model, it is easy estimate the minimum to maximum error under delay approximation and the error in the impartial optimum number of stages for a given logic function. The efficiency of the developed method is shown by a comparison of the theoretical results, achieved using the proposed method with simulation results of the MICROWIND tool using a standard 32-nm technology.

The rest of this paper is planned as follows: a review of the DML family is described in Section II. DML-LE model for simple inverter chains is established in Section III with three dissimilar levels of approximations. In Section IV compare the methods by simplicity and accuracy. The requirement of the optimum number of stages is also described in Section IV, which delivers an intuitive graphical visualization of the problem. DML-LE is prolonged to complex nets containing branching in Section V. In Section VI, the efficiency of the DML-LE theoretical optimization is examined for a standard 32-nm process.

II. DML OVERVIEW

As previously mentioned, an elementary DML gate architecture is poised with a static gate and a supplementary transistor, whose gate is connected to a global clock signal. In this project, we precisely focus on Dual Mode Logic gates that employ conventional CMOS gates on behalf of the static gate implementation.
DML gates are presented with two possible topologies: 1) Type A and 2) Type B, as shown in Fig. 1(a) and (b), respectively. In the static mode of operation, the transistor M1 is turned off by smearing the high Clk signal for Type A and low Clk for Type B topology. So, the gates of both topologies operate in like way to the static logic gate, which now is a standard CMOS operation.

To activate the gate in the dynamic mode, the Clk is allowed, allowing for two discrete phases: 1) pre-charge and 2) evaluation. Throughout the pre-charge phase, the output is charged to $V_{DD}$ in Type A gates and discharged to GND in Type B gates. Through evaluation, the output is assessed allowing to the values at the gate inputs.

DML gates demonstrate a very robust process in both static and dynamic modes in process variation at low supply voltages. The toughness in the dynamic mode is mainly achieved by the in-built active restorer (pull-up in Type A/pull-down in Type B) that also allowed glitch sustaining, charge drip, and charge distribution. It is also exposed that the suitable sizing methodology is the crucial factor to achieve fast operation in the dynamic mode. Fig. 1(c) displays the sizing of Complementary Metal Oxide Semi-Conductor (CMOS)-based DML gates that are optimized to a dynamic mode of operation, whereas Fig. 1(d) displays conventional sizing of a typical CMOS gate. The input and output capacitances of the DML gates are considerably reduced, as related with CMOS gates, due to the application of minimal width transistors in the pull-up of Type 1 or pull-down in Type B networks. The size of the pre-charge transistor is kept equal $S*W_{min}$ to uphold a fast pre-charge period even with the increase in the output load.

Differing to CMOS gates, every DML gate can be executed in two ways, only one of which is effective. The ideal topology is such that the pre-charge transistor is positioned in parallel to the stacked transistors, i.e., NOR in Type A is favored over NAND, and NAND in Type B is desired over NOR. In this event, the evaluation is executed through the parallel transistors and hence it is faster.

The finest design methodology of DML gates is to serially connect Type A and Type B gates, likewise to np-CMOS/NOR techniques. While this design methodology allows maximum performance, area minimization and improved power efficiency, serial connection of the identical type gates is also possible. However, this case shows many disadvantages, for example the need of footer/header and simple glitching. These well-explored problems are normal for dynamic gates design. DML asset is that the static mode CMOS-based DML gates with transistor sizes are optimized for the dynamic mode. Because of reduced static and switching energy consumption, Dynamic mode is actually semi-energy optimal CMOS construction of a gate. The static operation of the DML gates is used to considerably reduce energy consumption at the cost of 2–4 times reduction in performance. A common approach is to optimize the delay for the dynamic mode of operation and drive the system in the static mode only in standby/low-energy mode deprived of severe frequency restrictions, i.e., scale of 2–4 times in performance is approachable.

III. DML MODEL FOR SIMPLE INVERTER CHAIN

To enhance the performance of the DML gates, LE technique is needed to employed, modified, and approximated the well-explored. Though LE method is a renowned and widely used by designers, there are a few altered terminologies and metrics. The terminologies will be used to improve the LE for CMOS-based DML gates are presented. The LE design of DML is quite different from the conventional CMOS LE (and domino logic LE), which is conversed in previous section. This is due to unconventional sizing methodology and unique structure of DML gates. Attaining the ideal, non-approximate solution is relatively an exhausting task. However, by slight simplifications it can be solved similarly to the typical CMOS LE method. First, whole non-approximated LE method for DML CMOS-based gates is shown. Even though this solution is very accurate, it is not designer friendly and very complex. Therefore, two approximated solutions are offered. The difficulty of these solutions is much lesser, while attaining very high precision. Lastly, a detail about these approaches for DML LE for all CMOS-based gates is given.

i. Basic Assumptions

DML gates are designed to enhance their dynamic modedelay and thus only one transition amid$f_{plh}$and $f_{plh}$, which is a part of the evaluation phase, should be measured. This illustrates that only acorresponding resistance of the Pull-Down Network (PDN) (nMOSs) will perform a role in delay optimization of Type A gates and the Pull-Up Network (PUN) (pMOSs) will be appropriate in optimization of Type B gates. Though designing conventional CMOS gates, the PUN is characteristically upsized with $f_{plh}$, independently of the sizing factor $E_F$, which is the sizing aid of the load driving effort. This $f_{plh}$ is the result of the optimal delay of an unloaded gate. Characteristically, $f_{plh}$, resulting for an optimal gate delay, is dissimilar from $f_{sym}$that attains symmetric gate operation ($T_{plh}=T_{phl}$). Though, in most technologies $f_{plh}$ is approximately equal to $f_{sym}$, and $f_{sym} \approx f_{sym}$.[21]. By DML, every stand-alone gate would not be sized with $f_{plh}$ as the delay in the dynamic mode is defined by a single transition over PDN or PUN and hence there is no necessity in symmetric transitions. One and only sizing factor, $S_i$, for any i stage effect affects the evaluation network and the pre-charge transistor as shown in Fig. 1. In CMOS LE method, the normalization is executed to a typical CMOS inverter. DML gates are normalized to a regular minimal inverter (DML_INV) in Type A, which signifies the least standalone gate delay unit. A minimal inverter of Type B yields an increased delay, as it calculates the data through pMOS. In this project, assume every DML chain would start with Type Agates tailed by Type B gates (in a NORA/np-CMOS style).
As stated in the earlier section, ‘\( \gamma \)’ is the fabrication technology-dependent factor that defines the transistor gate capacitance to transistor drain capacitance ratio. Usually, in most nanometer scale processes, ‘\( \gamma \)’ is close to one. For CMOS inverters, it also defines the gate to drain capacitance of a particular MOS transistor. However, for all minimal transistor width DMLINV Type A or Type B is as follows:

\[
\frac{C_{g_{inv \text{- DML}}}}{C_{g_{inv \text{- DML}}}} = \frac{3C_{d_{MOS}}}{2C_{g_{MOS}}} \quad (1)
\]

Yielding: \( \gamma' = 3 \gamma / 2 \). \( \Box \)

ii. Defining the Problem for a Simple Inverter Chain

For obtaining the optimal sizing factors to a simple DML inverter chain, assume a chain as shown in Fig. 2. The delay of a common gate \( i \) in the chain is known by (3).

\[
i_{pd,i} = \frac{1}{L_{DML}} \left( \frac{R_{gate}}{R_{inv}} \frac{C_{g_{gate}}}{C_{D_{inv}}} \gamma' + \frac{L_{DML}}{C_{Load}} \right) \quad (3)
\]

A minimal DML inverter \( i_{po \text{- DML}} \) as follows:

\[
i_{pd,i,odd} = i_{po \text{- DML}} \left( \frac{(2S_i + 1)}{3S_i} \right) \gamma' + \frac{(S_i + 1)}{2S_i}
\]

\[
i_{pd,i,even} = i_{po \text{- DML}} \left( \frac{(2S_i + 1)}{3S_i} \right) \gamma' - \frac{(S_i + 1)}{2S_i}
\]

Where \( \mu_{np} \) is defined as \( \mu_n / \mu_p \). \( \mu_{np} \) is the \( i \)th stage sizing factor. Before, supposing an even number of inverters \( N \) in the chain, the delay of the chain can be stated by adding up the delays of all the chain constituents as

\[
\sum_{i=1}^{N} i_{pd,i} = i_{po \text{- DML}} \left( \frac{(2S_i + 1)}{3S_i} \right) \gamma' + \frac{(S_i + 1)}{2S_i}
\]

\[
\times \left( \sum_{odd,i}^{Type \ A} \mu_{np} \left( \frac{(2S_i + 1)}{3S_i} \gamma' + \frac{(S_i + 1)}{2S_i} \right) \right) + \left( \sum_{even,i}^{Type \ A} \mu_{np} \left( \frac{(2S_i + 1)}{3S_i} \gamma' - \frac{(S_i + 1)}{2S_i} \right) \right)
\]

In the next sections, three dissimilar solutions to the delay optimization problem are derived as follows: 1) Complete un-approximated; 2) Complete approximated and 3) Partially/SA solutions. These solutions are trading off complexity with accuracy.
variable multiplication. In common, it can be solved numerically, as below:

\[ S_1 = 1 \]

\[ 0 = B_2 S_1 - S_{22} + BS_1 S_3 \]

\[ 0 = B_2 S_2 - B_2 S_{23} + BS_2 S_4 \]

\[ 0 = B_2 S_3 - S_{24} + BS_3 S_5 \]

\[ \vdots \]

\[ 0 = B_2 S_N - B_2 S_{N2} + BS_N S_{N+1} \]

(8)

This is the maximum optimal and accurate resolution for DML inverter chain sizing. But, solving it is a very exhausting task. This un-approximated solution (CS) is much more difficult than a simple CMOS LE optimal solution, which is resultant with no assumptions and approximations. DML CS method complexity is owing to a non-standard sizing of transistors, connected in parallel to the Clock transistor.

Succeeding suppositions will be used in the rest of this project. Leading, as solved in previous section, the first gate of any examined chain will be least sized, i.e., \( S_1 = 1 \). \( S_1 \) can be indiscriminate to some possible sizes in accordance with any input capacitance. Another, even number of stages \( N \) is presumed. This is due to the topology of DML chains that mainly consists of Type B gates succeeding Type A gates. Still, the solution for the chain, which has an odd number of stages, can be easily consequential using the same methodology.

**Load Capacitance Effect on CS Method**

**Load effect on CS:-**

- **Power=64.720 \( \mu \)W
- **Area:-**
  - \( D_x=534 \text{ lambda (8.010 } \mu \text{m}) \)
  - \( D_y=476 \text{ lambda (7.140 } \mu \text{m}) \)
  - \( S_0, (Dx)(Dy)=254184 \text{ lambda (57.194 } \mu \text{m}^2) \)

**Power-Delay Product:-**

For CS,

\[ \text{PDP}_{CS} = (64.720 \text{ } \mu \text{W})(0.073 \text{ ns}) = 4.72456 \text{ fW-s} \]

**iv. Complete Approximated (CA) Method for the Sizing Factors of DML Inverter Chain**

To decrease the difficulty of the LE method, a CA solution, which trades off the accuracy and complexity, is derived.

It is beforehand conferred that (5) defines a common delay expression for the whole chain, supposing an even number of inverters \( N \). The CA method assumes that the involvement of minimal transistors to the drain and gate capacitances is negligible in contrast with \( 2S_i \) and with \( S_i+1 \), for every stage of the chain. As exposed in Section V, ignoring these transistors, for complex gates increases the accuracy w.r.t inverters. Then, (5) can be expressed by

\[ D = \sum_{i=0}^{N} D_i \]

\[ = \text{IP}_{\text{p, CA}} \left( \sum_{\text{Type_A}} \left( \frac{(2S_i+1)}{3S_i} y + \frac{S_{i+1}+1}{2S_i} \right) \right) + \sum_{\text{Type_B}} \left( \mu_{\text{CA}} / \left( \frac{(2S_i+1)}{3S_i} y + \frac{S_{i+1}+1}{2S_i} \right) \right) \]

(9)

These suppositions are acceptable only when the output load capacitance of the chain is high. The sizing factors \( S_i \) is affected by the large load capacitance. As soon as \( S_i \) increases, also increases, along the chain; this calculation will increase in accuracy for high \( i \) values. After generalization, (9) can be revised as follows:

\[ D = \sum_{i=0}^{N} D_i \]

\[ = \text{IP}_{\text{p, CA}} \left\{ \sum_{\text{Type_A}} \left( \frac{2}{3} y \frac{S_i+1}{2S_i} \right) + \sum_{\text{Type_B}} \left( \mu_{\text{CA}} / \left( \frac{2}{3} y \frac{S_i+1}{2S_i} \right) \right) \right\} \cdot \frac{1}{\text{IP}_{\text{p, CA}}} \]

(10)
By differentiating \( dD/ds = 0 \), ensuing the same procedure (Section B) for all odd \( i \) (11) and even \( i \) (12):

\[
\frac{S_i}{S_{i-1}} = \frac{1}{S_i \mu n/p} 
\]

\[
\frac{S_i}{S_{i-1}} = \frac{1}{S_i \mu n/p} 
\]

The sizing factors solution for this CA method is quite related to standard CMOS solution. Likewise to CMOS, the upsizing factor is constant. But every even stage is factored by an additional \( \sqrt{\mu n/p} \). On behalf of the \( N \)-size chain, the sizing factors can be shown in series as in Table I where \( A \) is expressed in (14). In CMOS, the sizing factors are resulted from the load to input capacitance ratio, while in DML, they are illustrated by the ratio of the first to last sizing factors.

\[
\text{Delay of the total chain is denoted by the sum of delays of all } n \text{ logic stages and of all the added } n \text{ inverters.}
\]

Distinguishing the chain delay by \( N \) then equating to zero as follows:

\[
\gamma \left( \mu n/p + \mu n/p \right) + \sqrt{F_{\text{DML}}} \cdot \ln \left( F_{\text{DML}} \right) = 0
\]
Simulation Result of Load Capacitance Effect in SA Method

Delay for Load Capacitance effect in SA Method

Load effect on SA:-
  Power=39.180 μW
Area :-
  Dx=550 lambda (8.25 μm)
  Dy=493 lambda (7.395 μm)
So, (Dx)(Dy)=271150 lambda (61.01 μm²)

Power-Delay Product:-
For SA,
PDP_SA=(39.180 μW)(0.146 ns)= 6.61249 fW-s

IV. COMPARISON OF THE DML METHODS

Now, a comparison between the SA, CS, and CA techniques is shown. The techniques are compared with simplicity, accuracy and depend on delay in the optimum number of stages.

Comparison of DML, CS, CA and SA Methods

Comparison between CA, CS and SA Methods:-
  Power=26.995 μW
Area :-
  Dx=494 lambda (7.410 μm)
  Dy=297 lambda (4.455 μm)
So, (Dx)(Dy)=146718 lambda (33.01 μm²)

Power-Delay Product:-
For CA,
PDP_CA= (26.995 μW)(0.146 ns)= 3.94127 fW-s
For CS,
PDP_CS=(26.995 μW)(0.073 ns) = 1.970635 fW-s
For SA,
PDP_SA= (26.995 μW)(0.146 ns)= 3.94127 fW-s

V. DML EVALUATION FOR COMPLEX GATES AND BRANCHES IN 32 nm PROCESS

The proposed methodology is observed by results derived by MICROWIND tool. The evaluation is executed on two different complex logic networks, realized in a low power typical 32nm technology.

The DML Methods are compared for universal gates NAND & NOR gates, also complex gates like AOI₂¹ and OAI₂¹ gates.

For 3-input NAND Gate:
First, 3-input NAND gate is implemented using DML methods and compared for Type A and Type B for every methods are tabulated in Table A.

For 3-input NOR Gates:
Now,3-input NOR gate is implemented using DML methods and compared for Type A and Type B in every method are tabulated in Table B.

For AOI₂¹ Gate:
Just like above, the same procedure is followed for AND-OR-INVERTER gate which is derived as (A(B+C))'. This AOI₂¹ gate is implemented using DML methods and compared for Type A and Type B in every method are tabulated in Table C.
For OAI₂₁ Gate:

For OAI₂₁ (OR-AND-INVERTER) gate i.e., ((A+B)C)', also executed as in AOI₂₁ Gate and is tabulated in Table D.

VI. CONCLUSION

The proposed approach permitted an efficient optimization of DML logic networks for full performance in the dynamic mode of operation, which was the focus of this project. DML logic, optimized conferring to the proposed DML methods, allowed long flexibility in optimizing several structures of DML networks. This optimization used the DML inherent properties which significantly condensed parasitic capacitance and ultra-low power dissipation in the static operation mode.

This project offered three different approaches, which traded off between complexity, computation and accuracy. The complex CS method was only spoken for error analysis of the further methods. The CA method was indistinguishable to CMOS computation with very minor error and the SA method was also identical to the CMOS computation assisting one more lookup table (which easily derived for all cases and loads). Analysis showed that with these tools only a design can attain very high performance results.

The desired DML gates topology is such that the pre-charge transistor is located in parallel to the stacked transistors, i.e., NOR in Type A is preferred over a NAND, and NAND in Type B is preferred over NOR.

Advantages and drawbacks of each one of the methods were conferred. Simulation results were carried out in a standard 32-nm process, verified the efficiency of the proposed approach and again compared it with existing CMOS.

<table>
<thead>
<tr>
<th>Gate</th>
<th>Power (µW)</th>
<th>Delay (ns)</th>
<th>Power-Delay Product (f W-s)</th>
<th>Dx (or µm)</th>
<th>Dy (or µm)</th>
<th>Area (or µm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAND_normal</td>
<td>0.331</td>
<td>0.030</td>
<td>0.009</td>
<td>162(3.24)</td>
<td>120(2.40)</td>
<td>19440(7.78)</td>
</tr>
<tr>
<td>NAND_TA_CA</td>
<td>0.305</td>
<td>0.053</td>
<td>0.016</td>
<td>184(3.68)</td>
<td>130(2.60)</td>
<td>23920(9.57)</td>
</tr>
<tr>
<td>NAND_TA_CS</td>
<td>0.346</td>
<td>0.029</td>
<td>0.010</td>
<td>187(3.74)</td>
<td>129(2.58)</td>
<td>24684(9.87)</td>
</tr>
<tr>
<td>NAND_TA_SA</td>
<td>0.352</td>
<td>0.029</td>
<td>0.010</td>
<td>185(3.70)</td>
<td>129(2.58)</td>
<td>23865(9.55)</td>
</tr>
<tr>
<td>NAND_TB_CA</td>
<td>11.472</td>
<td>0.056</td>
<td>0.642</td>
<td>185(3.70)</td>
<td>129(2.58)</td>
<td>23865(9.55)</td>
</tr>
<tr>
<td>NAND_TB_CS</td>
<td>17.511</td>
<td>0.031</td>
<td>0.543</td>
<td>186(3.72)</td>
<td>129(2.58)</td>
<td>23994(9.60)</td>
</tr>
<tr>
<td>NAND_TB_SA</td>
<td>17.500</td>
<td>0.036</td>
<td>0.630</td>
<td>183(3.66)</td>
<td>128(2.56)</td>
<td>23424(9.37)</td>
</tr>
</tbody>
</table>

Table A: Simulation Results comparison of NAND gate

<table>
<thead>
<tr>
<th>Gate</th>
<th>Power (µW)</th>
<th>Delay (ns)</th>
<th>Power-Delay Product (f W-s)</th>
<th>Dx (or µm)</th>
<th>Dy (or µm)</th>
<th>Area (or µm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOR_normal</td>
<td>0.357</td>
<td>0.090</td>
<td>0.032</td>
<td>160(3.20)</td>
<td>107(2.14)</td>
<td>17120(6.85)</td>
</tr>
<tr>
<td>NOR_TA_CA</td>
<td>12.595</td>
<td>0.085</td>
<td>1.070</td>
<td>184(3.68)</td>
<td>112(2.24)</td>
<td>20608(8.24)</td>
</tr>
<tr>
<td>NOR_TA_CS</td>
<td>29.716</td>
<td>0.046</td>
<td>1.367</td>
<td>182(3.64)</td>
<td>114(2.28)</td>
<td>20748(8.30)</td>
</tr>
<tr>
<td>NOR_TA_SA</td>
<td>13.163</td>
<td>0.085</td>
<td>1.199</td>
<td>184(3.68)</td>
<td>111(2.22)</td>
<td>20424(8.17)</td>
</tr>
<tr>
<td>NOR_TB_CA</td>
<td>0.324</td>
<td>0.169</td>
<td>0.055</td>
<td>183(3.66)</td>
<td>111(2.22)</td>
<td>20313(8.13)</td>
</tr>
<tr>
<td>NOR_TB_CS</td>
<td>0.374</td>
<td>0.092</td>
<td>0.034</td>
<td>183(3.66)</td>
<td>113(2.26)</td>
<td>20679(8.27)</td>
</tr>
<tr>
<td>NOR_TB_SA</td>
<td>0.354</td>
<td>0.169</td>
<td>0.059</td>
<td>183(3.66)</td>
<td>112(2.24)</td>
<td>20496(8.20)</td>
</tr>
</tbody>
</table>

Table B: Simulation Results comparison of NOR gate

<table>
<thead>
<tr>
<th>Gate</th>
<th>Power (µW)</th>
<th>Delay (ns)</th>
<th>Power-Delay Product (f W-s)</th>
<th>Dx (or µm)</th>
<th>Dy (or µm)</th>
<th>Area (or µm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOI_normal</td>
<td>0.160</td>
<td>0.044</td>
<td>0.007</td>
<td>167(3.34)</td>
<td>124(2.48)</td>
<td>20708(8.283)</td>
</tr>
<tr>
<td>AOI_TA_CA</td>
<td>5.866</td>
<td>0.055</td>
<td>0.323</td>
<td>185(3.70)</td>
<td>129(2.58)</td>
<td>23865(9.550)</td>
</tr>
<tr>
<td>AOI_TA_CS</td>
<td>13.118</td>
<td>0.030</td>
<td>0.340</td>
<td>186(3.72)</td>
<td>130(2.60)</td>
<td>24180(9.670)</td>
</tr>
<tr>
<td>AOI_TA_SA</td>
<td>5.988</td>
<td>0.032</td>
<td>0.311</td>
<td>187(3.74)</td>
<td>130(2.60)</td>
<td>24310(9.720)</td>
</tr>
<tr>
<td>AOI_TB_CA</td>
<td>6.530</td>
<td>0.083</td>
<td>0.542</td>
<td>185(3.70)</td>
<td>128(2.56)</td>
<td>23680(9.470)</td>
</tr>
<tr>
<td>AOI_TB_CS</td>
<td>15.023</td>
<td>0.045</td>
<td>0.676</td>
<td>184(3.68)</td>
<td>128(2.56)</td>
<td>23552(9.42)</td>
</tr>
<tr>
<td>AOI_TB_SA</td>
<td>6.546</td>
<td>0.058</td>
<td>0.380</td>
<td>186(3.72)</td>
<td>131(2.62)</td>
<td>24366(9.75)</td>
</tr>
</tbody>
</table>

Table C: Simulation Results comparison of AOI₂₁ gate

<table>
<thead>
<tr>
<th>Gate</th>
<th>Power (µW)</th>
<th>Delay (ns)</th>
<th>Power-Delay Product (f W-s)</th>
<th>Dx (or µm)</th>
<th>Dy (or µm)</th>
<th>Area (or µm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAI_normal</td>
<td>0.356</td>
<td>0.045</td>
<td>0.016</td>
<td>160(3.2)</td>
<td>119(2.38)</td>
<td>19040(7.62)</td>
</tr>
<tr>
<td>OAI_TA_CA</td>
<td>0.325</td>
<td>0.056</td>
<td>0.018</td>
<td>183(3.66)</td>
<td>129(2.58)</td>
<td>23607(9.44)</td>
</tr>
<tr>
<td>OAI_TA_CS</td>
<td>0.369</td>
<td>0.031</td>
<td>0.011</td>
<td>183(3.66)</td>
<td>128(2.56)</td>
<td>23424(9.37)</td>
</tr>
<tr>
<td>OAI_TA_SA</td>
<td>0.346</td>
<td>0.044</td>
<td>0.015</td>
<td>183(3.66)</td>
<td>131(2.62)</td>
<td>23973(9.59)</td>
</tr>
<tr>
<td>OAI_TB_CA</td>
<td>2.012</td>
<td>0.084</td>
<td>0.169</td>
<td>182(3.64)</td>
<td>128(2.56)</td>
<td>23929(9.32)</td>
</tr>
<tr>
<td>OAI_TB_CS</td>
<td>4.410</td>
<td>0.046</td>
<td>0.202</td>
<td>185(3.70)</td>
<td>128(2.56)</td>
<td>23680(9.47)</td>
</tr>
<tr>
<td>OAI_TB_SA</td>
<td>4.144</td>
<td>0.059</td>
<td>0.244</td>
<td>185(3.70)</td>
<td>129(2.58)</td>
<td>23865(9.55)</td>
</tr>
</tbody>
</table>

Table D: Simulation Results comparison of OAI₂₁ gate
REFERENCES


