A PROJECT REPORT
ON

"MODEL AND ALGORITHM BASED PROJECT"

In Fulfilment of the Requirement for the Project of

Digital VLSI Design

BY

Pooja Srinivas	**20171403**
Swapnil Pakhare	**20161199**
Ayush Anand	**20161085**
Mayank Taluja	**20161102**

UNDER THE GUIDANCE OF
Dr. Zia Abbas

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
IIIT HYDERABAD

Acknowledgements

We are profoundly grateful to **Dr. Zia Abbas** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

We would like to express deepest appreciation towards **Prattek Gupta**, TA of this course, and **Hema Sai Kalluru**, TA whose invaluable guidance supported us in completing this project.

Pooja Srinivas
Swapnil Pakhare
Ayush Anand
Mayank Taluja

ABSTRACT

For most of our digital integrated circuits, the core of the circuits is comprised of MOSFETs. Technology scaling has many perks and the results were good for a very long time resulting in increased logic per area count and also decreased delays.. Due to various increased leakages and power issues in the nanometer technology, we have been trying to shift to Finfets and other technologies. But MOSFETs still dominate the market.

Modelling the working of any circuit to imitate the features of a MOSFET is essental for development of new technologies. Reducing the time of any circuit testing is of high value in VLSI Architecture Design. In this model based project we aim to train models to output the values of any circuit for any technology within seconds, essentially reducing the time consumed for testing.
The essence of the algorithm project is to come up with optimal transistor sizing for minimising circuit functions, with the only functions being power leakage and time delay, circuit parameters being widths and lengths of nMOS and pMOS transistors, using the HSPICE tool for device level circuit simulation.

Keywords: Power leakage, Delays, Error, Modela and Algorithms.

Contents

Chapter 1

Introduction

1.1 Machine Learning Model based Optimization

Reducing the cost and time of any model generation and testing is of essence in research to develop better technologies. Training supervised machine learning models to predict the leakages, power and delays will help us develop models that mimic the circuits in lesser time. These models can be trained once and in turn used multiple times to test the working of a specific technology. The motivation of using this method is to speedup the entire process of testing and analyzing the working of a technology.

1.2 Krill Herd Optimization Algorithm

Our aim is to find the optimal transistor sizing for minimising circuit functions, with the only functions being power leakage and time delay, circuit parameters being widths and lengths of nMOS and pMOS transistors. For this purpose, the algorithm we have been given is the Modified Krill Herd Optimization Algorithm using Focus Group Idea
It is a nature-inspired optimization algorithm which is inspired by herding behavior of krill individuals. In order to improve the performance of this algorithm to deal more effectively with high dimensional numerical functions, Focus Group idea is used. This is used to modify the solutions found by searching agents in group cooperation.

Chapter 2

Model based Project

2.1 Modelling using Neural Networks

The Artificial Neural Network has three layers: input, hidden, and output layers. The hidden layer consists of hidden units where a weighted sum of the inputs is fed to a nonlinear activation function, a sigmoidal function. The output of the hidden units is again linearly combined to obtain the required outputs. Our model has 4 layers: input, output and two hidden layers. We have trained the model as a **Inverter, NAND2, NAND3, NOR2, NOR3, AND3 and XOR3** using the data set provided. The outputs include Propagation delay low-high, Propagation delay high-low and Leakage power. This model is predictive and hence the output can be predicted after training with a sufficiently large data set. The data set given is divided into train and test data with a ratio 80:20. The model developed is highly modular and hence new models can be added for new data sets while still being able to access the previous models.

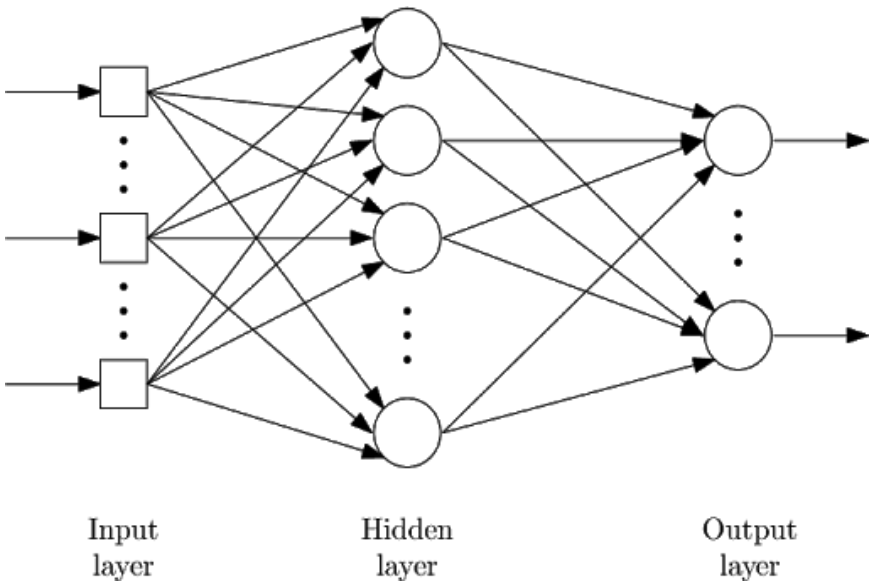


Figure 2.1: Neural Network with 1 Hidden Layer

2.2 Modelling using Linear Regression

Linear regression is a linear approach to modelling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). To model multiple response variables simple linear regression was not sufficient. Multiple Output Regression is used to model a simple linear regression to a multi output linear regression. Multiple Output Regression fits, the regressor model specified, one per output. Hence the entire regressor behaves as one and enables the prediction of multiple outputs.

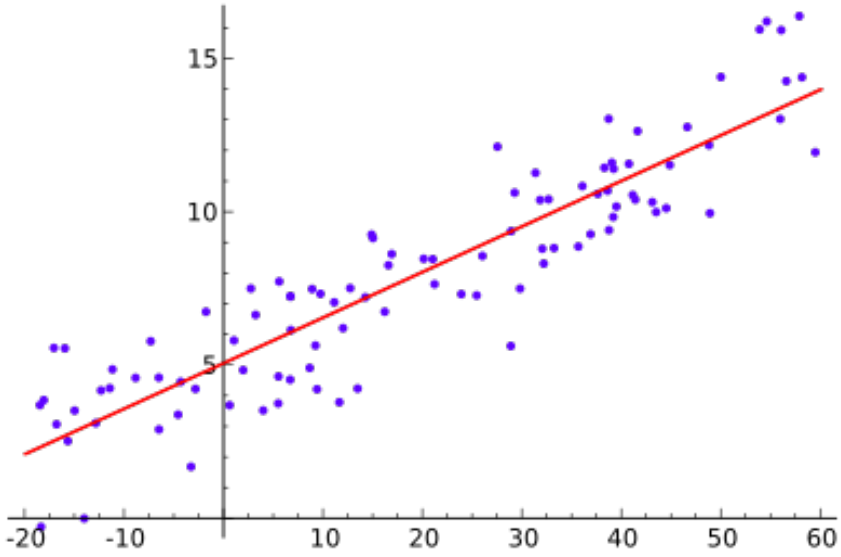


Figure 2.2: Linear Regression

2.3 Modelling using Support Vector Regression

The Support Vector Regression (SVR) uses the same principles as the SVM for classification The main idea is to minimize error, individualizing the hyperplane which maximizes the margin, keeping a check so that, part of the error is tolerated. Non Linear SVR was also implemented but not retained because there was no significant decrease in error and hence all the kernel functions were redundant adding to the computation complexity. SVR was performed over Linear, polynomial and rbf. Of all the three kernels linear gave the best results but for this case we retained the values of rbf to demonstrate the variance in output with different kernels. This shows that understanding the data and choosing kernels based on that is highly essential for model training to receive best results.

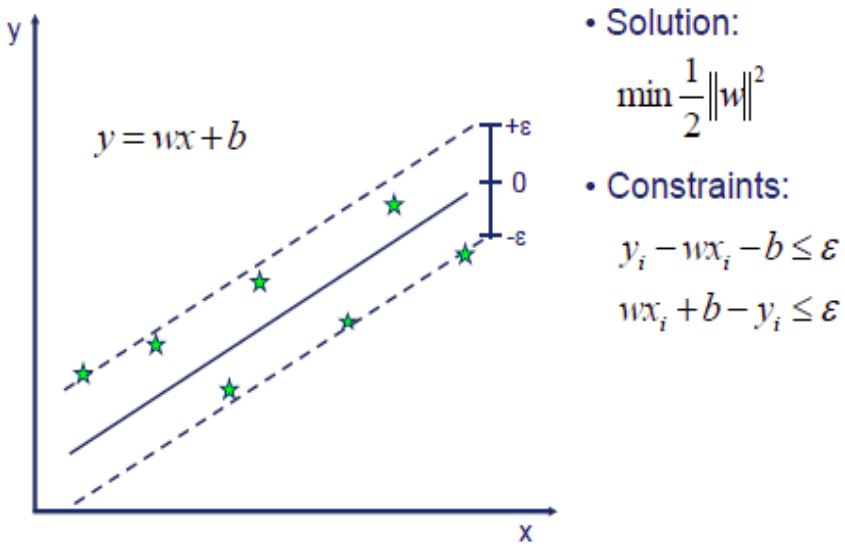


Figure 2.3: Support Vector Regression

2.4 Modelling using Decision Tree Regression

Decision tree builds regression or classification models in the form of a tree structure. This model learns local linear regression with a tree structure based on the depth to fit the training data. The model was tested varying the depth but no significant change was observed for the given data set. Increasing the depth to very large number caused over-fitting which is not desirable. Further, over-fitting led to an overall decrease in the accuracy of the model. Decision trees are desirable over SVR or LR because they are internally modelled to be multi output regression models unlike the other two. This will help us model outputs with inter dependencies which is limited in the case of SVR and LR.

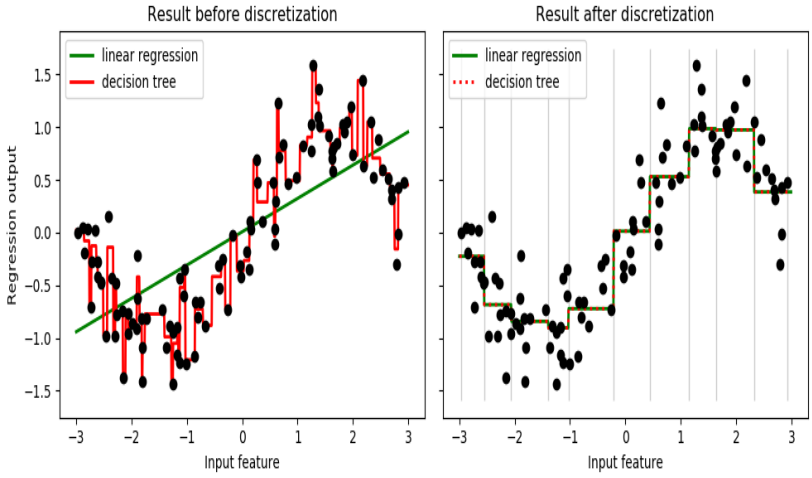


Figure 2.4: Decision Tree Regression

2.5 Modelling using Random Forest Regression

A random forest is a meta estimator that fits a number of classifying or regression decision trees, to improve the predictive accuracy and control over-fitting. Bagging, in the Random Forest method, involves training each decision tree on a different data sample where sampling is done with replacement. Hence, we understand that Random Forest is an improvised model of Decision Trees improving the learning of the model. In our model, we varied the **n estimator** parameter which decides the number of decision tree models used. The errors remain almost constant even with large variations.

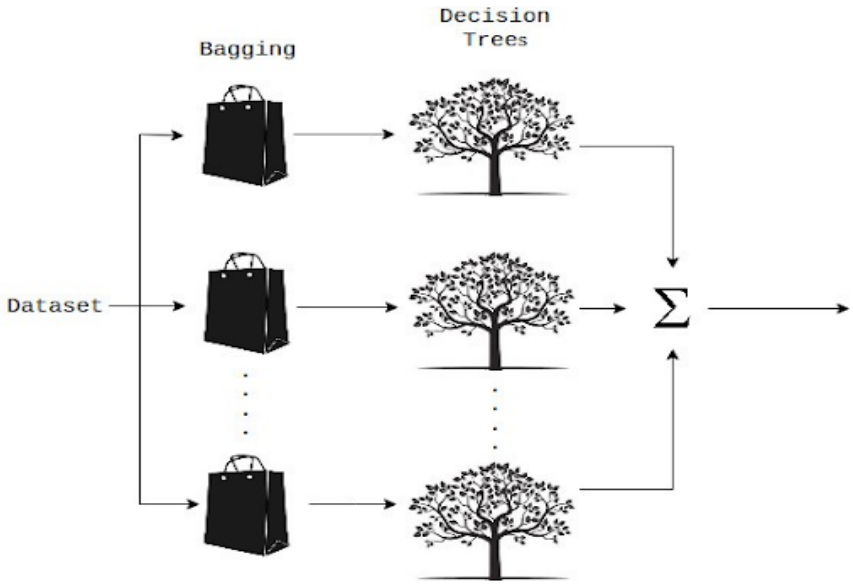


Figure 2.5: Random Forest Regression

Chapter 3

Algorithm Based Project

3.1 Krill Herd Algorithm

The Krill Herd Algorithm belongs to the category of Bio-inspired Optimisation Algorithms. Such algorithms are chosen because they do extremely well on multi-dimensional optimisation problems. They are then extended to work for multiple objectives. As multi-objective Krill Herd does not exist, we reduced all Leakages to a single objective and kept all delays in a strict bound. For Full Adder, we have 8 leakages and 6 delays = 14 objectives. To reduce this to a single objective problem, we take mean of leakages and keep delays in bound.
This algorithm is modelled after Krills which are a kind of fish that try to move towards 2 things :

- Areas of higher food
- Areas of higher krill density

It includes Lagrangian for motion. The step update of Krill position is :

$$X(t+\Delta t) = X(t) + \Delta t \frac{\delta X}{\delta t}$$

For a particle i, its step is given by :

$$\frac{\delta X_i}{\delta t} = N_i + F_i + D_i$$

Here, N_i is the Neighbor induced motion, F_i is the Foraging motion and D_i is the random Diffusion motion of the ith particle.

3.1.1 Motion Induced By Other Krills

According to theoretical arguments, the krill individuals try to maintain a high density and move due to their mutual effects. The direction of motion induced, α_i, is estimated from the local swarm density (local effect), a target swarm density (target effect), and a repulsive swarm density (repulsive effect). For a krill individual, this movement can be defined as:

$$N_i^{new} = N^{max}\alpha_i + \omega_n N_i^{old}$$

where,

$$\alpha_i = \alpha_i^{local} + \alpha_i^{target}$$

and N^{max} is the maximum induced speed, ω_n is the inertia weight of the motion induced in the range [0, 1], N_i^{old} is the last motion induced, α_i^{local} is the local effect provided by the neighbors and α_i^{target} is the target direction effect provided by the best krill individual.

3.1.2 Foraging motion

The foraging motion is formulated in terms of two main effective parameters. The first one is the food location and the second one is the previous experience about the food location. This motion can be expressed for the ith krill individual as follows:

$$F_i = V_f\beta_i + \omega_f F_i^{old}$$

where,

$$\beta_i = \beta_i^{food} + \beta_i^{best}$$

and V_f is the foraging speed, $_f$ is the inertia weight of the foraging motion in the range [0, 1], F_i^{old} is the last foraging motion, β_i^{food} is the food attractive and β_i^{best} is the effect of the best fitness of the i'th krill so far.

3.1.3 Physical diffusion

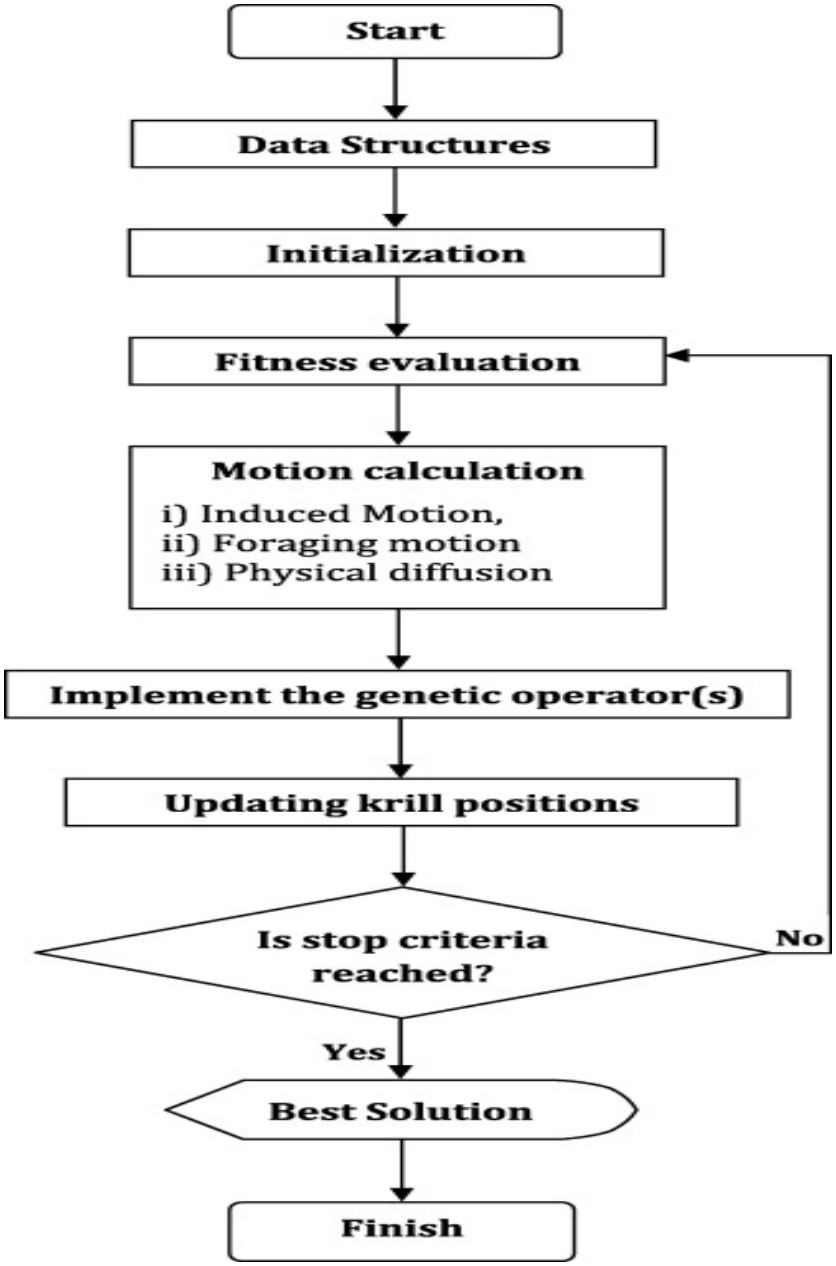
The physical diffusion of the krill individuals is considered to be a random process. This motion can be express in terms of a maximum diffusion speed and a random directional vector. It can be formulated as follows:

$$D_i = D^{max}\delta$$

where D_max is the maximum diffusion speed, and δ is the random directional vector and its arrays are random values between -1 and 1. The better the position of the krill is, the less random the motion is. Thus, another term is added to the physical diffusion formula to consider this effect. The effects of the motion induced by other krill individuals and foraging motion gradually decrease with increasing the time (iterations). This new term linearly decreases the random speed with the time and works on the basis of a geometrical annealing schedule:

$$D_i = D_i^{max}(1 - \frac{I}{I_{max}})\delta$$

3.1.4 Algorithm Flow Chart

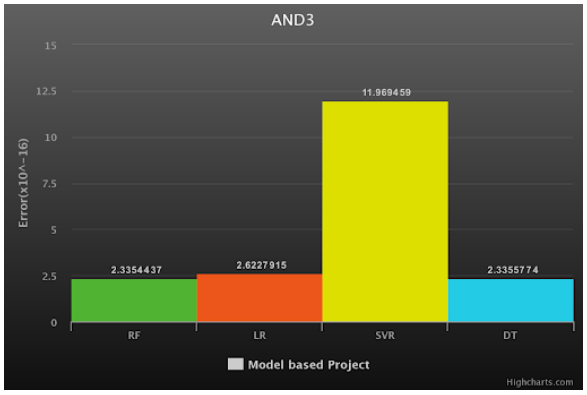


Chapter 4

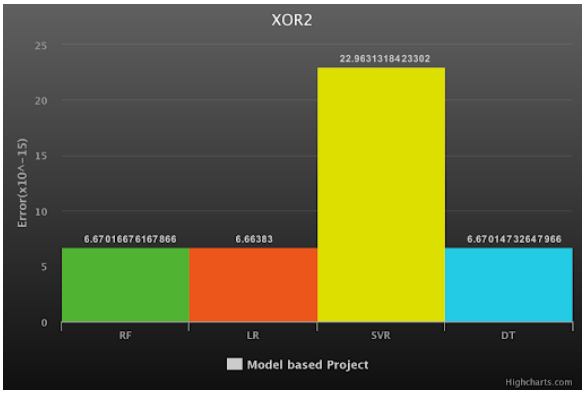
Results

4.1 Model Based Project

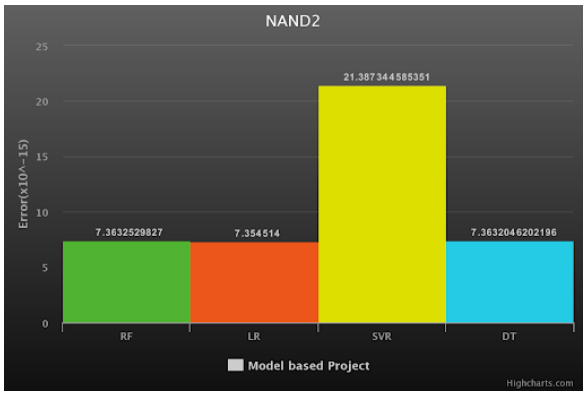
Mean Square Error has been calculated individually for every gate or circuit with different input conditions. For example, AND2 has 00, 01, 10, 11 input conditions. Each model has been train separately for all the conditions to give delays and leakages. Final results were calculated by taking the mean of the all the input conditions for both delays and leakages, to obtain a final error value for a specific circuit from each model.



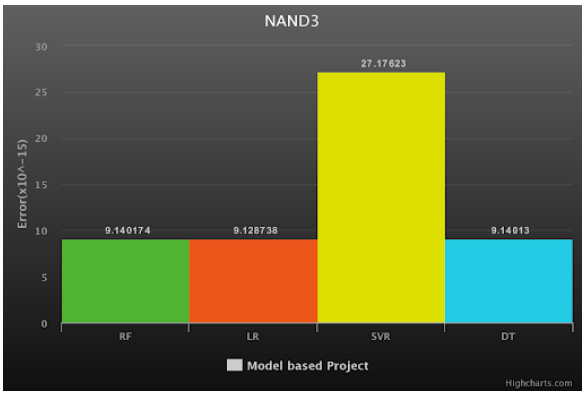
(a) AND3



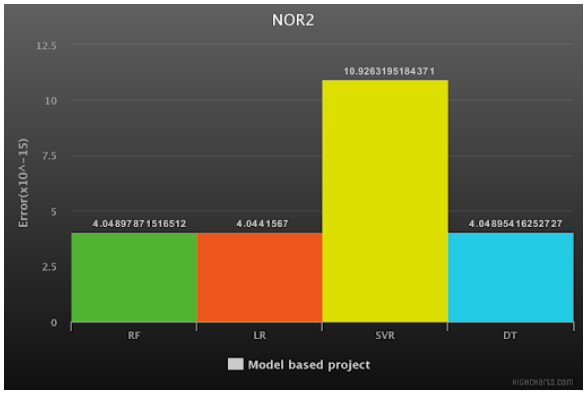
(b) XOR2



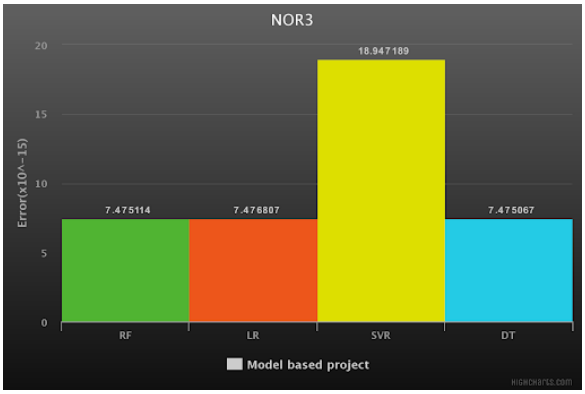
(a) NAND2



(b) NAND3



(a) NOR2



(b) NOR3

4.2 Algorithm Based Project

Delay	**Old**	**New**
delay lh nodeaco	1.00558×10^{-11}	8.69043×10^{-12}
delay hl nodeaco	1.17463×10^{-11}	2.71432×10^{-11}
delay lh nodebco	8.47425×10^{-12}	7.14982×10^{-12}
delay hl nodebco	1.18857×10^{-11}	2.89149×10^{-11}
delay lh nodecco	1.03265×10^{-11}	1.21012×10^{-11}
delay hl nodecco	1.13746×10^{-11}	1.10815×10^{-11}

Leakage	**Old**	**New**
000	3.65835×10^{-06}	2.14053×10^{-06}
001	3.69198×10^{-06}	2.00234×10^{-06}
010	3.04658×10^{-06}	1.77634×10^{-06}
011	3.00985×10^{-06}	1.84428×10^{-06}
100	2.98732×10^{-06}	1.79932×10^{-06}
101	2.69092×10^{-06}	1.57589×10^{-06}
110	2.99784×10^{-06}	1.88214×10^{-06}
111	2.66278×10^{-06}	1.68353×10^{-06}