Research Paper Presentation

Ramanathan Annamalai

IIT Hyderabad

July 4, 2021

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

A Novel Model for Injecting Error in Probabilistic Gates

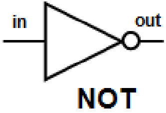
Abstract

1. Using inaccurate models to produce accurate results
2. We will talk about developing an algorithm to a probabilistic gate model.
3. Developing and calculating a metric for analysing accuracy - OPE.
4. Simulating and Comparing simple and complex gates set-up used for the probabilistic model.

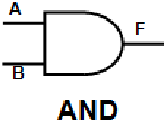
Few Terms

1. Electronic Design Automation (EDA) - Used for designing, simulating and verification of electronic systems such as integrated circuits and printed circuit boards.
2. Stochastic Computational Models (SCM) - Stochastic computing is a collection of techniques that represent continuous values by streams of random bits.
3. PCMOS - Probabilistic complimentary Metal Oxide Semiconductor.

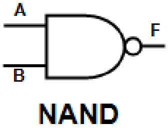
Logic Gates



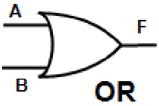
Input	Output
I	F
0	1
1	0



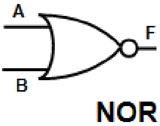
Inputs		Output
A	B	F
0	0	0
1	0	0
0	1	0
1	1	1



Inputs		Output
A	B	F
0	0	1
1	0	1
0	1	1
1	1	0



Inputs		Output
A	B	F
0	0	0
1	0	1
0	1	1
1	1	1



Inputs		Output
A	B	F
0	0	1
1	0	0
0	1	0
1	1	0

Figure: Basic Logic Gates and their Truth Tables

Efficiency

Understanding the output requirements

1. Many features need to be granted in the next computing system according to the target application.
2. To get the better usage of the system, it is important to understand the application's requirements as some applications can tolerate some error in their arithmetic blocks to get low-power consumption

Example

Allowing about 5% loss in classification accuracy for k-means clustering algorithm can lead to 50X improvement in energy saving compared to the fully accurate classification.

EDA

1. The conventional EDA tools employ deterministic methods to simulate the circuits by controlling the inputs and observing the outputs.
2. The evolution in the integrated circuit (IC) industry and computing systems adds more complexity and challenges for EDA as there are many sources of error in the input values in real time.This has to be accomodated in simulations too.

Using Probability in EDA Simulations

1. Using probability and allowing some error in the final result of the simulation will reduce the time taken for each simulation significantly.
2. There are many probabilistic methods used in different areas of EDA tools to measure the circuit reliability, and the process variation on ICs manufacturing.
3. Some computational methods like SCM and PCMOS use probability to arrive at their results.
4. We will mainly be looking at comparing the probabilistic models with the accurate models and analyse the differences and feasibility of usage of such probabilistic methods.

Inaccurate Model

1. The accurate and inaccurate model accepts the inputs $in_1(i), in_2(i)$ from the database.
2. An additional data set pertaining to the error ϵ is taken by the the inaccurate model.
3. This ϵ may or may not be related to the data set or the gates which operate on them. For Probabilistic gates, the ϵ is an independent known probability distribution like uniform, exponential, normal, etc.

Effect of ϵ on Logic Gates

Based on the value of ϵ, each logic gate changes it's behavior as shown in the below truth table.

	Inputs ϵ	B	A	AND	OR	XOR
Accurate	0	0	0	0	0	0
	0	0	1	0	1	1
	0	1	0	0	1	1
	0	1	1	1	1	0
Inaccurate	1	0	0	1	1	1
	1	0	1	1	0	0
	1	1	0	1	0	0
	1	1	1	0	0	1

Figure: Truth Table for AND, OR, XOR gates with ϵ

OPE Model

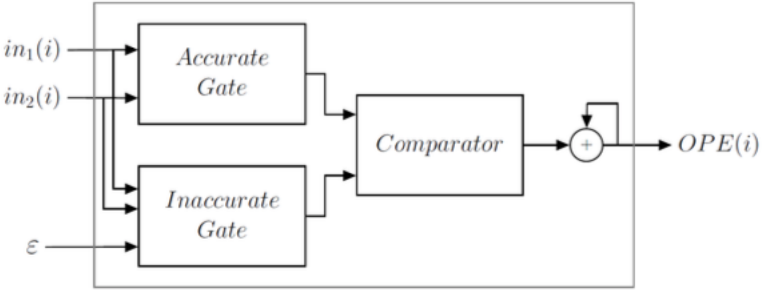


Figure: OPE Calculation Model

OPE Model

OPE

1. OPE is the Output Percentage Error. It is the average error produced in the inaccurate gate.
2. To calculate OPE, we have a comparator device which compares the outputs of the accurate and inaccurate gates and takes the average error over the total inputs of the data set, say N, by using the below formula

$$\mathrm{OPE} = \left(\frac{\sum_{i=0}^{N} |out(i)_{acc} - out(i)_{inacc}|}{N} \right) * 100\%$$

3. ϵ can be considered as a PDF and OPE can be considered as it's equivalent CDF.

Probabilistic Gate Model

Components

1. Accurate Gate - Defines the main required functionality of the model based on its inputs.
2. Comparator - outputs ϵ by comparing the desired probability of error in our model and the estimated physical noise.
   The result of comparator depends on the probability distribution function of the noise.
3. Multiplexer - Outputs either the result of the accurate gate or it's inversion based on the output of the comparator.

Probabilistic Model

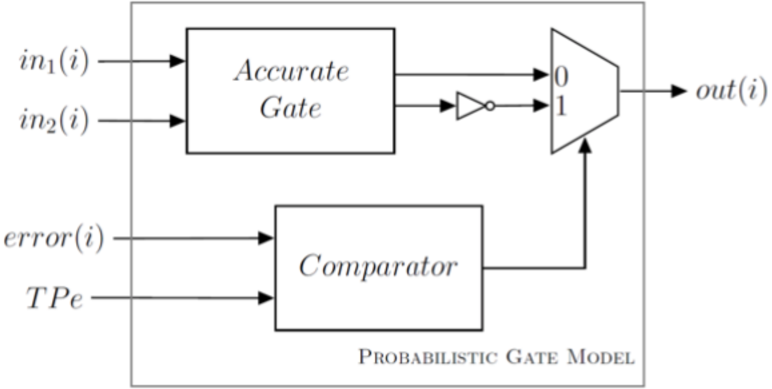


Figure: Probabilistic Gate Model

Probabilistic Gate Model

Error Terms

1. Physical error in the device (error) - Probability distribution function of the physical error.
2. Target Probability Error (TPE) - related to the target number of outputs forced to be incorrect for a set of inputs.
3. Error in Probabilistic Model (ϵ) - Function of physical error and TPE.

Probabilistic Gate Model

Working

1. Within a specific time t, the logic gate has a finite number of inputs and outputs n.
2. According to the error distribution and the probability of error, our model forces the output to be correct or incorrect.
3. The exact series of steps involved is clearly explained in the Algorithm in the following page.

Probabilistic Gate Model

Algorithm

$n \leftarrow$ Number of Samples within Time t

Inputs:

$in_1(i), in_2(i) \in [0, 1] \leftarrow$ Time Space Inputs

$error(i) \in [0, 1] \leftarrow$ Time Space Physical Error

$Pe \in \{0, 1\} \leftarrow$ Probability of Error

$in_{th} \in [0, 1] \leftarrow$ Input Threshold

for$(i \leq n)$ **do** **if** $(in_1(i) \leq in_{th})$ **then**

$crispin_1(i) \leftarrow 0$

else

$crispin_1(i) \leftarrow 1$

end if

Probabilistic Gate Model

Algorithm Contd.

if $(in_2(i) \leq in_{th})$ **then**
  $crispin_2(i) \leftarrow 0$
else
  $crispin_2(i) \leftarrow 1$
end if
$correct(i) \leftarrow crispin_1(i)$**logic operator**$crispin_2(i)$
$incorrect(i) \leftarrow$ **NOT**$(correct(i))$
if $(error(i) \leq Pe)$ **then**
  $out(i) \leftarrow incorrect(i)$
else
  $out(i) \leftarrow correct(i)$
end if
end for

Simple and Complex Gates

Simple Gates

These are single gates producing outputs based on the inputs and errors as mentioned in the probabilistic gate models.

Complex Gates

1. These are combination of many simple gates, each of which behave in a probabilistic manner.
2. The output of the complex gates is the result of inaccuracies in each simple gate.
3. In this presentation we will mainly be focusing on the XOR topology, i.e. the arrangement of gates which produce an identical truth table to that of the XOR truth table.

Complex Gates

Terminologies

1. Topology - A particular arrangement of simple gates.
2. Gate Count - Total number of gates in the set-up.
3. Level - set of all gates which receive a common input.
4. Structure - Either symmetric or non-symmetric based on the inputs of each gate in the same level.

XOR Topologies

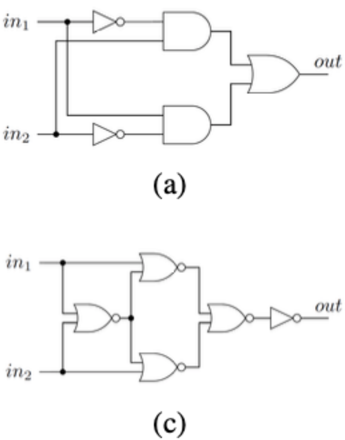


Figure: Few XOR Topologies

Complex Gates

Characteristics of XOR Topologies

Topology	Gate Count	Structure	Levels
a	5	Symmetric	3
b	4	Symmetric	3
c	5	Symmetric	4
d	3	Non-Symmetric	2

Table: Characteristics of XOR Topologies

Simulations

1. Now, we asses the probabilistic gates.
2. We apply uniform and normal distributions by generating sufficient number of random data and fit the probability distribution function to these data.
3. For the normal distribution, the applied data has $\mu = 0.5$ and $\sigma = 0.125$.
4. As there is no control on the physical error in the real device, we sweep $Pe \in [0, 1]$ to get the result of OPE.
5. OPE is calculated over *Pe* range to check the effect of the added noise on simple and complex gates.

Simulations

Simple Gates

1. $P(0)$ is the probability of correct '0' output, and $P(1)$ is the probability of correct '1' output.
2. These probabilities are proved to be exact to the accurate gate in case of $Pe = 0$, and inverted for $Pe = 1$.
3. The results of OPE are found to be identical with CDFs of fitted PDFs for the injected noise either as uniform or normal distribution.
4. OPE for simple gates is independent of the functionality of these gates, as it is only effected by the noise distribution in the gate.
5. OPE equation can be re- written as:

$$\text{OPE} = \left(\sum_{i \le Pe} \frac{|error_i|}{N} \right) * 100\%$$

Simulations

Simple Gates Graph

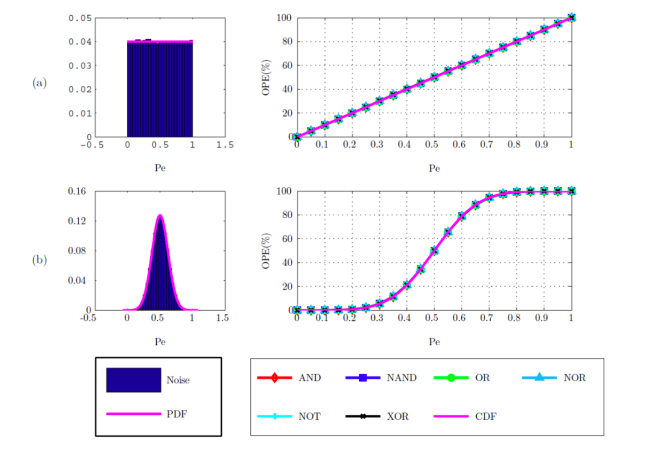


Figure: OPE for simple gates, and CDF of the injected noise (on the right) and Histogram of the injected noise with the fitted PDF of it (on the left) are plotted for, (a) Uniform Distribution, and (b) Normal Distribution

Simulations

Complex Gates

1. The results for complex probabilistic XOR gates show great improvement in OPE for $Pe \geq 0.5$ and little degradation in the performance for $Pe \leq 0.5$.
2. First and second topologies have the same OPE despite the difference constellation cause of OPE is identical between each level in first topology with its analogous level in second topology as shown in both distributions.
3. Third topology which has the largest level of gates and the highest count has the best OPE among other topologies because of the last NOT gate in the constellation which has the great impact on OPE improvement.

Simulations

Observations in Complex Gates

1. Topologies with even levels as topology c,d show better OPE rather than that with odd levels as topology a,b.
2. All levels of gates in the XOR topologies have the same OPE.

Simulations

Complex Gates (XOR) Graph

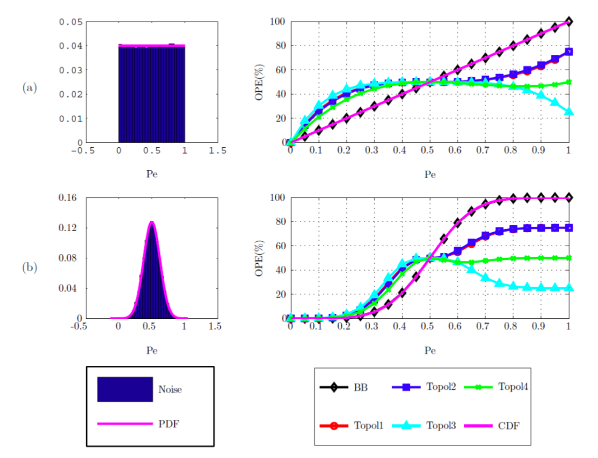


Figure: OPE for different XOR topologies, and CDF of the injected noise (on the right) and Histogram of the injected noise with the fitted PDF of it (on the left) are plotted for, (a) Uniform Distribution, and (b) Normal Distribution

Summary

1. The primary purpose of this paper is developing an algorithm to model probabilistic gates which is implemented by a device with an error that can be extracted as a probability distribution function.
2. This can be useful in modeling probabilistic gates in EDA tools.
3. This model is investigated for simple and complex gates by applying uniform and normal random distribution as error for these gates.

Summary

1. Using the OPE for numbers of inaccurate samples as a metric to measure our model's efficiency proves validation of the model, as OPE for simple probabilistic gates matches CDF of injected noise for uniform and normal distributions.
2. Integrating these simple gates to implement four different complex topologies of XOR gate, and study the effect of simple probabilistic gates on XOR functionality shows OPE improvement in topologies with even levels.