An Implementation of CMOS Arbiter Physical Unclonable Function with Selecting Modules using Skywater 130 nm [*]

Rhodel Quizon[†]
University of the Philippines Diliman
Diliman, Quezon City
Metro Manila, Philippines
rrquizon1@up.edu.ph

1. INTRODUCTION

Physical Unclonable Functions or PUFs are one of the recent emerging protocols in cryptographic protocols for IoT authentication. It utilizes the manufacturing variations to receive different responses from devices of same designs. It mostly uses variability in production such as delays to receive different responses. These means that PUFs are designed so that two devices with similar designs will have different group of challenge response pairs.

One of these PUFs are the arbiter PUFs which are composed of cascaded multiplexers with a D flip-flop at the end. Two paths will race to reach the D and clock input respectively. This kind of PUF utilizes the delays propagated by different property of silicon to induce different response given similar design and similar inputs.Figure 1 shows sample Arbiter PUF circuit.

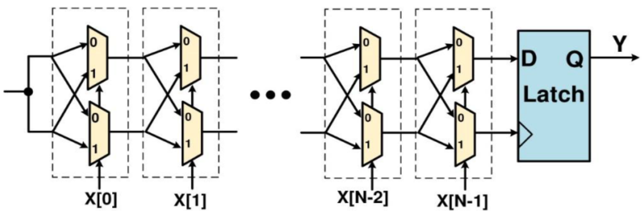

Figure 1: Single bit PUF cell taken from the original paper

This paper will show a different architecture of an arbiter PUF using transmission gates, buffer and two D flip flops as arbiter as done by [1]. This paper will implement study done by [1] on 45 nm architecture to skywater 130 nm architecture.

This paper shows an implementation of the paper [1] by Moradi et al. The original paper implemented their design at 45nm CMOS while this paper implements it at skywater 130 nm architecture. Very close implementation is migrated to skywater 130nm implementation and is then evaluated.

The original paper proposes a weak PUF with a single challenge input to generate an unpredictable response. The paper uses 64 PUF cells using the single bit PUF cell illustrated on figure 2.

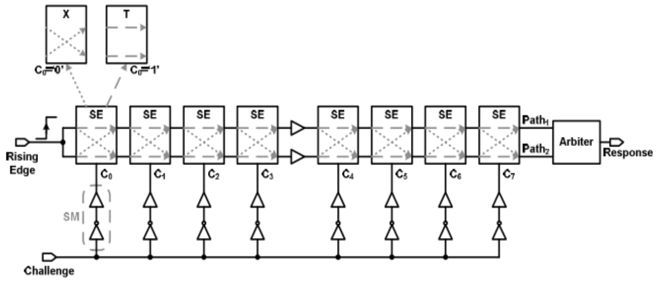

Figure 2: Single bit PUF cell taken from the original paper

2. METHODOLOGY

The PUF is composed of 8 Selecting Modules (SM) and 8 Switching Elements (SE) which are responsible for which path will reach the arbiter first. The output response will depend on which path will reach the arbiter first.

An SE is composed of 4 transmission gates which receives the outputs of the SMs. Figure 3 shows the implementation of the SE on skywater 130 nm.

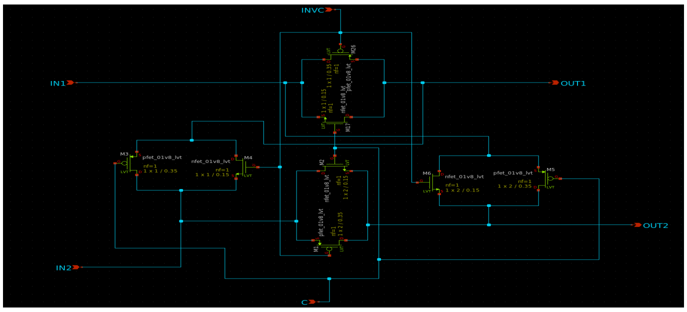

Figure 3: Skywater 130 nm implementation of SEs

One PMOS and one NMOS `nfet_01v8_lvt` and `pfet_01v8_lvt` are used to compose the transmission gates. These are also the MOSFET instances used in the whole project. PMOS and NMOS used for one transmission gates are of the same size. Top and left transmission gate both have PMOS and NMOS width of 1 while bottom and right both have the width of 2. These sizes are based on the ratio used from the original paper.

The states of the SEs are controlled by the SMs. The SMS are composed of three inverters cascaded to each other. The ratios used in this paper is the same with the ratio of the inverters used in the original paper. Figure 4 shows the implementation of the SMs.

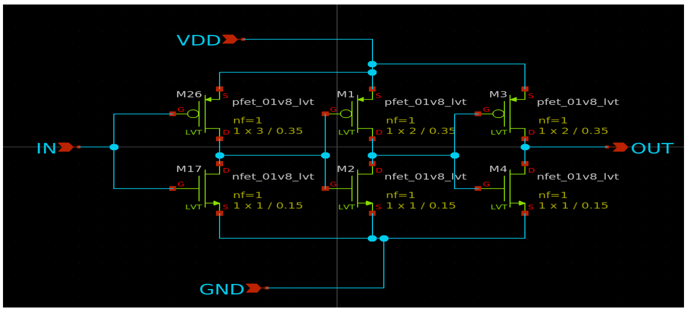

Figure 4: Skywater 130 nm implementation of SMs

The input challenges used for the SMs is VDD/2. This input will cause the output of the SM to be 1 or 0 depending on the variability. The original paper uses 1V as VDD sot heir input challenge is 0.5V for the case for skywater 130nm since the VDD of the MOSFETS used are 1.8V we will be using an input challenge of 0.9V. Figure 5 shows multiple instances of VTC curve of the Selecting Modules.

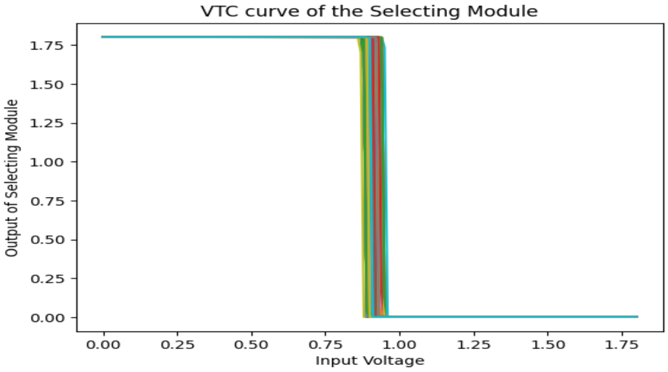

Figure 5: Multiple Instances of VTC curve of the selecting module

It can be observed that on multiple instances, the output of the SMs inverts at different levels, the PUF will take advantage of these variability to make the paths of the rising passing through the SEs random. Median of the switching is around 0.9, making 0.9 an ideal voltage level for causing randomness. Different variations causing Path1 or Path2 to reach the arbiter first. See figure 6 for example path racing to the arbiter.

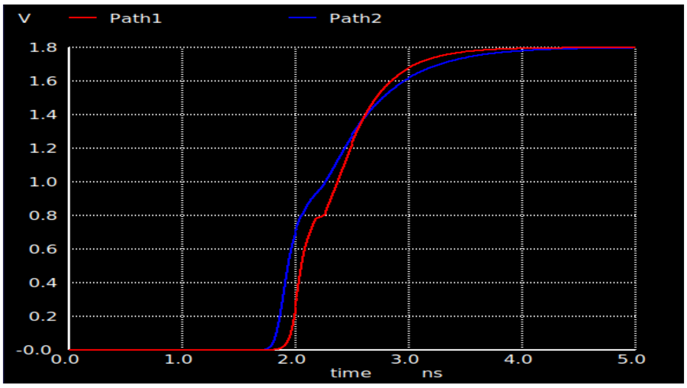

Figure 6: Path1 and Path2 racing to the arbiter

The signal that reaches the arbiter first will still depend on the variability of the transistors used in the arbiter for this example, it can be observed that Path2 reached 0.9V first before Path1.

The arbiter circuit used two D flip flops (DFF) instead of just one. The two D flip flops are connections are shown in figure 7:

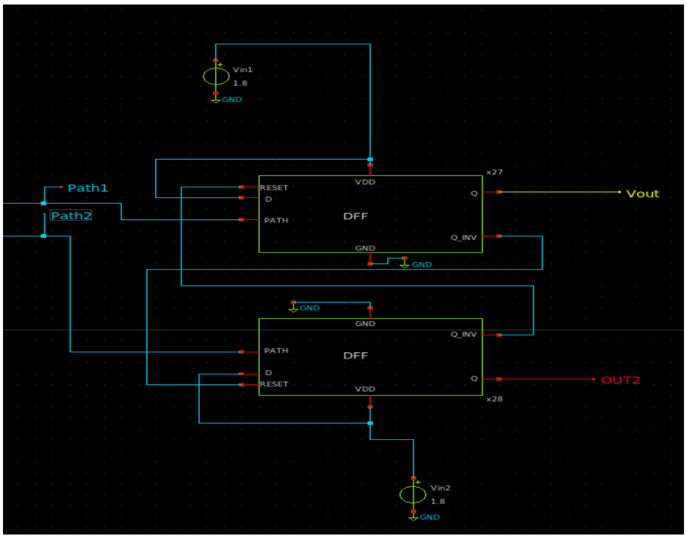

Figure 7: Connections of 2 D flip flops used as Arbiters

The output Vout is the main output of the PUF cell. The PATH input of the DFF is the clock input. The data input are always shorted to level high. Whichever path reaches the their DFF will have an output of 1. This will result to an output of 0 in the inverted output of the said DFF. The inverted output of each flip flops are connected to the active low reset of the other DFF. This means that when one path already reached a DFF and shifted 1 to the output, the other DFF will be held in reset causing it to not affect the output of the first triggered DFF.

3. RESULTS AND DISCUSSION

Evaluation Methodology

This PUF implementation is simulated using NGSPICE and the skywater 130 nm technology. The statistics of PUF responses are analyzed using python. The main target of this paper is to implement the reference paper using the indicated technology. For Uniqueness and Randomness, Monte Carlo simulation of 6400 instances of the PUF cell was done for and the output bits are grouped by 64 to simulate 64 different instances for one silicon. The simulation do not factor layout and is focused on the design variation for the PUF cell.

For Reliability 640 instances of the PUF cell is used to sweep different voltages and temperature. *Randomness*

Randomness of PUF responses are judged by the proportion of 1s and 0s. To be declared random, the proportion of 1s and 0s should be close to 50%. For this current implementation, the average proportion of 1s are 45.64%. Figure represents the responses for this implementation. Figure 8 shows that the PUF does not show any systematic pattern or any noticeable correlation. The randomness of the PUF are calculated using the Shanon Entropy H show in eq 1 was used to measure entropy. This was also used in the original paper. The average entropy calculated for this implementation 0.9834 which is very close to the ideal value of 1. This means that there are no correlations among bits despite it only having 45.64% of 1s.

$$H = -p\log_2(p) - (1-p)\log_2(1-p) \tag{1}$$

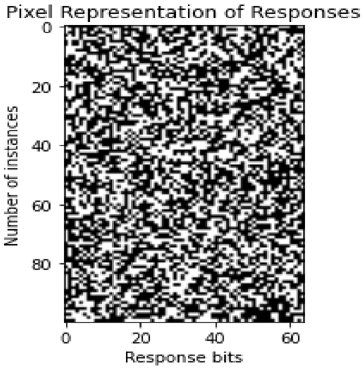

Figure 8: Pixel Representation of 64 bit PUF responses

Uniqueness

Uniqueness is one factor that is used for performance evaluation of PUFs. It measures how different are different responses to each other. It is measured using pairwise Hamming distances of different responses among each other. Ideally, hamming distances should be exactly half of response bits to have perfect identification. The average of all the hamming distances is computed to identify uniqueness. Figure 9 shows the histogram of pairwise hamming distances.

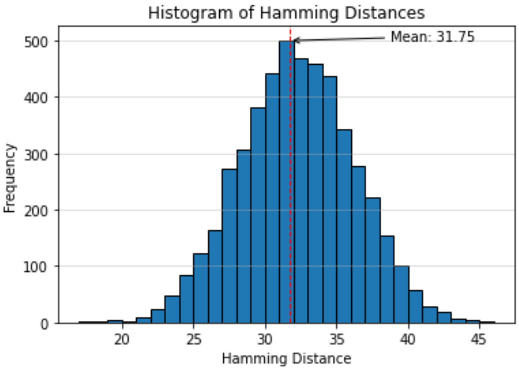

Figure 9: Hamming Distance of 64 bit responses

Figure shows that most of the instances have differences around 30 to 32. There are 501 instances of 31 and 470 for 32. Measuring the mean of all the pairwise distances, we get 31.75 bit difference. This means that there is a uniqueness of around 49.61% on this implementation of this PUF architecture.

Reliability

The reliability is measured by running an instance of the PUF cell and sweeping VDD voltage and temperature. For

VDD, the ground truth is considered at 1.80V while the voltage sweep is sweep at 1.70V and 1.90V. The temperature is measured at 0 to 100 degC with temperature at 20 degC as ground truth. 640 instances of the PUF cell are used for this test. Each instances' VDD and temperature are swept and the output voltage is checked if it is still above or below threshold 0.9V. Figure 10 shows the reliability based on temperature and voltage.

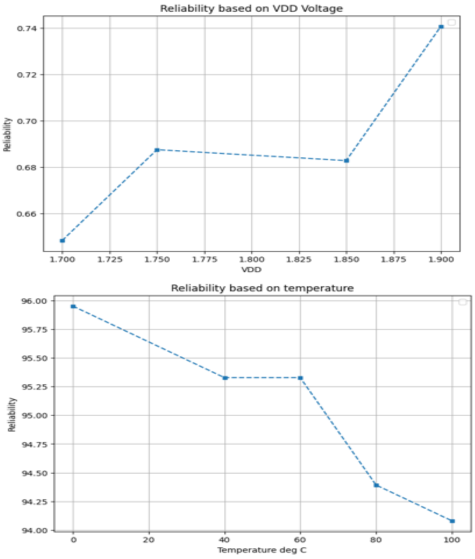

Figure 10: Reliability performance

It can be observed that at VDD sweep, the PUF is highly unreliable with highest reliability at 1.90V only 1t 74% and lowest reliability at 65%. When temperature sweep was done, the PUF architecture is reliable with highest reliability at 96 % at 0 deg C and around 94 % at 100 deg C. This means that skywater 130nm is highly reliable to temperature with the PUF architecture while highly unreliable for VDD changes.

4. CONCLUSION

This paper shows an implementation of [1] using skywater 130nm. Exact ratios indicated in the original paper was followed. It is found that the skywater 130nm is both reasonably random and unique. It has median 45.64% ratio of 1s compared to 0s for each 64 bit response and has an entropy of 0.9834 which is close to 1. Also, hamming distance is calculated to check the uniqueness of the design and it is found that calculated average hamming distance is 31.75 bits which is very close to the ideal 32 for 64 bit response.

However, it is found that the PUF design is not very reliable at skywater 130 nm at variable VDD voltage. It is only reliable at max 74% and minimum 65% when VDD voltage is swept from 1.70V to 1.90V. While when temperature is swept, it is more reliable with maximum reliability of 96% and minimum reliability of 94%.

Some tweaking and adjustments could still be done to reach higher reliability even at different VDD voltages.

5. REFERENCES

[1]M. Moradi, Reza Faghih Mirzaee, and S. Tao, "CMOS Arbiter Physical Unclonable Function with Selecting Modules," Aug. 2020, doi: https://doi.org/10.1109/cads50570.2020.9211853.