Assignment 1 Report

Devansh Agarwal 200316
Utkarsh Aditya 201066
Harshit Singhai 2000434
Antariksh Choudhary 200159
Abhinav Zade 200019

Answers to the Report questions

1. Let's say that we have 64 XOR gates and we are focusing on the i^{th} XOR, then we can write the time taken for the signal to pass from the i^{th} XOR as:

$$t_i^0 = a_i\{x_i\delta_{11}^i + (1-x_i)\delta_{01}^i\} + (1-a_i)\{x_i\delta_{10}^i + (1-x_i)\delta_{00}^i\} + t_{i-1}^0$$

where:

x_i: Output of the $(i-1)^{th}$ XOR and input to $i^{th}XOR$
a_i: i^{th} Input bit
t_i^0: Time taken by the signal to pass i^{th} XOR considering x_0 as 0
$\delta_{00}^i, \delta_{01}^i, \delta_{10}^i, \delta_{11}^i$: time delay taken by the XOR for inputs 00,01,10, and 11 respectively

This equation considers x_0 as 0, and since it is a ring oscillator, x_0 will change to 1 in a single cycle; this will invert every x_i throughout the oscillator. Thus, the total time period of one cycle can be written as:

$$T_i = t_i^0 + t_i^1$$
$$= a_i(\delta_{11}^i + \delta_{01}^i) + (1-a_i)(\delta_{10}^i + \delta_{00}^i) + T_{i-1}$$
$$= a_i(\delta_{11}^i + \delta_{01}^i - \delta_{10}^i - \delta_{00}^i) + \delta_{10}^i + \delta_{00}^i + T_{i-1}$$
$$= a_i\alpha_i + \beta_i + T_{i-1}$$

where:

$$\alpha_i = \delta_{11}^i + \delta_{01}^i - \delta_{10}^i - \delta_{00}^i$$
$$\beta_i = \delta_{10}^i + \delta_{00}^i$$

The counter takes a decision based on the frequency of the Ring Oscillator, i.e., $f = \frac{1}{T_{63}} = \frac{1}{t_{63}^0 + t_{63}^1}$. If the frequency of the top signal is greater, then the counter output is 1. We can also conclude and say that if the Time period of the top signal is smaller, then the output is 1. Hence, the difference of time periods Δ can decide the output of the counter.

$$\Delta_{63} = a_{63} \cdot w_{63} + b_{63} + \Delta_{62}$$
$$= (a_{63} \cdot w_{63} + b_{63}) + (a_{62} \cdot w_{62} + b_{62}) + ... + (a_0 \cdot w_0 + b_0)$$
$$= [w_{63} \quad w_{62} \quad ... \quad w_0] \cdot \begin{bmatrix} a_{63} \\ a_{62} \\ . \\ . \\ a_0 \end{bmatrix} + (b_{63} + b_{62} + ... + b_0)$$

$$= \begin{bmatrix} w_0 \\ w_1 \\ . \\ . \\ w_{63} \end{bmatrix}^T \cdot \begin{bmatrix} a_{63} \\ a_{62} \\ . \\ . \\ a_0 \end{bmatrix} + (b_{63} + b_{62} + ... + b_0)$$

$$\Delta_{63} = w^T \cdot a + b$$

where w_i is the difference between α_i, and b_i is the difference between β_i for two XORs. Thus, we can represent the output as a linear model using this equation:

$$y = \frac{1 + sign(w^T \cdot a + b)}{2}$$

2. An Advanced XORRO PUF is a collection of multiple Simple XORRO PUFs. So we can extend our linear model established above for a Simple XORRO PUF to crack an Advanced XORRO PUF. A Simple XORRO PUF takes 2 XORROs and compares their frequency and an Advanced XORRO PUF consists of 2^S XORROs (where S is the no. of select bits) , thus we will use 2 XORROs at a time in an Advanced XORRO PUF to build a linear model. Similarly, we will get $2^S - 1(2^S - 1)$ linear models for breaking an Advanced XORRO PUF.

   We first use the 2S select bits to select the the 2 XORROs for building a linear model. The first S select bits are read as a binary number and then converted to decimal to get the index number of the first XORRO. The second S select bits are used similarly to select the second XORRO. After obtaining 2 XORROs we use the linear model developed above for a Simple XORRO PUF. We furthermore use $2^S - 1(2^S - 1)$ linear models for an Advanced XORRO PUF and combine all the above obtained linear models in a matrix to get our final model.

3. The code to this problem has been submitted individually.

4.

		Training Time	Test Accuracy
Loss Hyperparameter	Hinge	2.39s	94.02
	Squared Hinge	2.19s	94.75
C Hyperparameter	High	2.78s	93.69
	Medium	3.16s	94.74
	Low	1.05s	93.99
Tol Hyperparameter	High	1.47s	94.74
	Medium	3.14s	94.74
	Low	3.63s	94.74
Penalty Hyperparameter	l1	10.97s	94.59
	l2	3.16s	94.74

LinearSVC

		Training Time	Test Accuracy
C Hyperparameter	High	2.44s	94.94
	Medium	1.77s	93.92
	Low	1.56s	89.63
Tol Hyperparameter	High	1.23s	93.91
	Medium	1.77s	93.92
	Low	1.39s	93.91
Penalty Hyperparameter	l1	8.29s	93.45
	l2	1.77s	93.92

Logistic Regression