# Title of the Experiment: Side Channel Attacks


## Group Members' Names: Sujan Kumar Saha, Pankaj Bhowmik


**Date: 11/12/2019**

---


**Abstract**

This experiment is about side channel attack. We implemented a DES encryption algorithm in MAX10 FPGA. Then we tried to do power side channel attack to extract the key used in DES encryption. This experiment has two parts. In the first part, we have done the Simple Power Analysis (SPA) where we directly observed the power consumption of the FPGA chip and figured out the key value from the power signal. In the second part, we have done the Differential Power Analysis (DPA) to extract the key. The power signal traces were given. We have done the DPA analysis using Matlab and tried to find the key.


**Experiment Details**

**Goals:** The goal of this experiment is to understand how a power side channel attack can be launched in a system where a cryptographic computation is performed.

**Experimental Setup:** We used the MAX10 FPGA on Hardware Hacking (HaHa) board, USB-blaster, Quartus Prime Software, Analog Discovery and Matlab.

**Experiment Steps:**

1. We run the DES algorithm in Quartus from experiment five. We connected the analog discovery to the HaHa board according to the mentioned procedure and observed the MAX10 FPGA power signal in the channel 2 of analog discovery.
2. We implemented the Matlab code of DPA analysis and run the code to find the key.

**Results and Analysis:**

**Part I:**

1. The current sensing resistor situated in between JP2 and JP3 in the haha board. The values of the resistor in 1 ohm. Figure 6 shows a processes to get the voltage drop against the current sensing resistor. The voltage drop does not reflect the total power consumption by the FPGA. However, the figure 7 shows the proper way to get the actual power consumption by the FPGA. Because, one point of the scope is connected after the current sensing resistor and the other one lies in the ground. If we follow figure 6, it might damage the resistor.
2. The waveform when e is 8'b01110001 is presented in Figure 1. Figure 2 also represents the SPA analysis for 00001111.

3. Figure 5,6, and 7 represents the spa for 1-A.sof, 1-B.sof ,and 1-c.sof files. To guess the values in those files, we implement some other files with different values of e, and they are given Figure 3 and 4. According to the analysis, we can assume that the a_sof, b_sof, and c_sof contains the e value of 1100110011, 11111111 ,and 11100111, respectively.

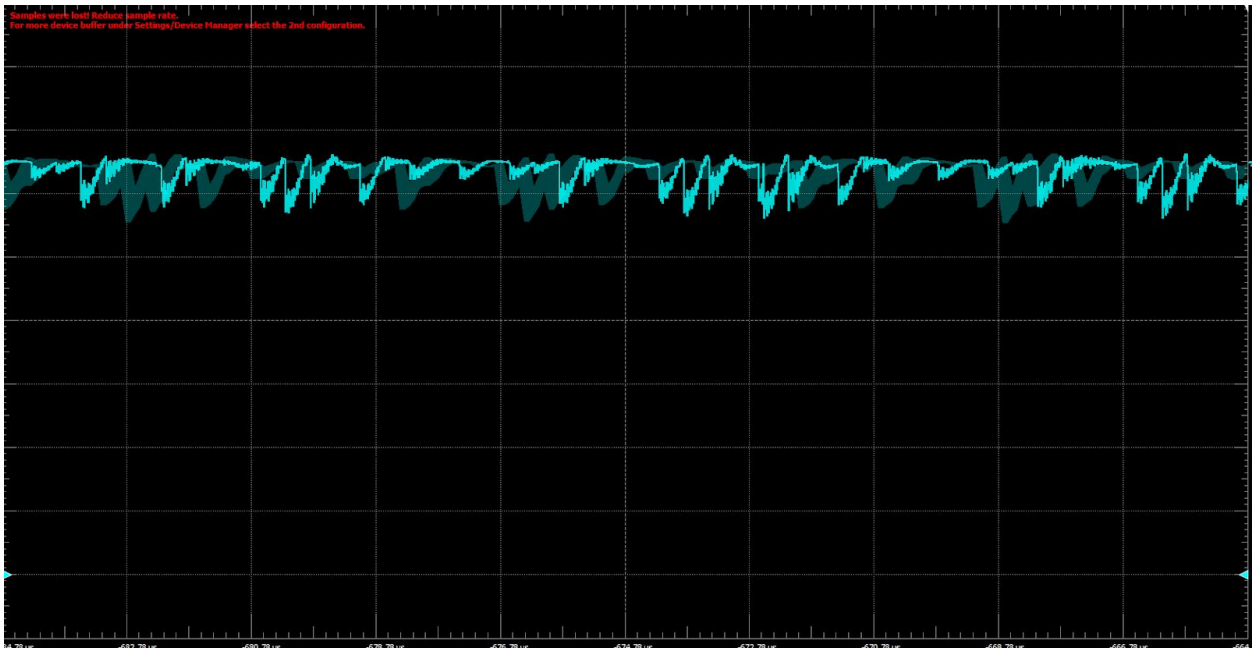


Figure1: SPA trace when the value of e is **01110001**

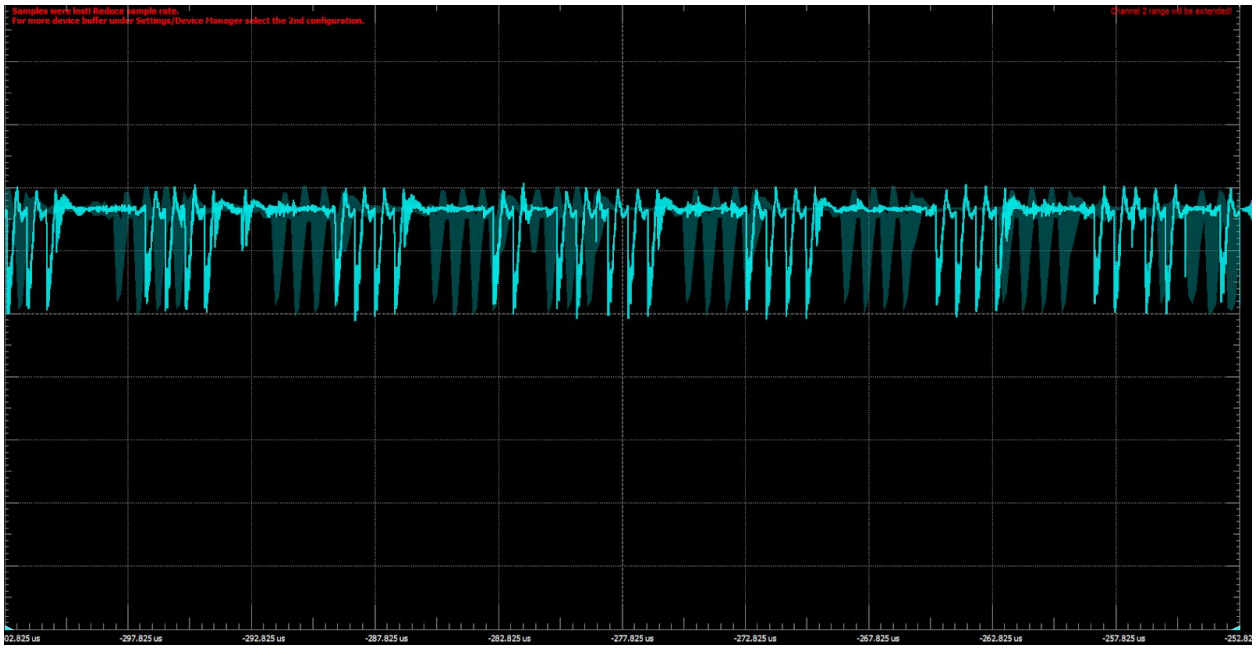


Figure2: SPA trace when the value of e is **00001111**

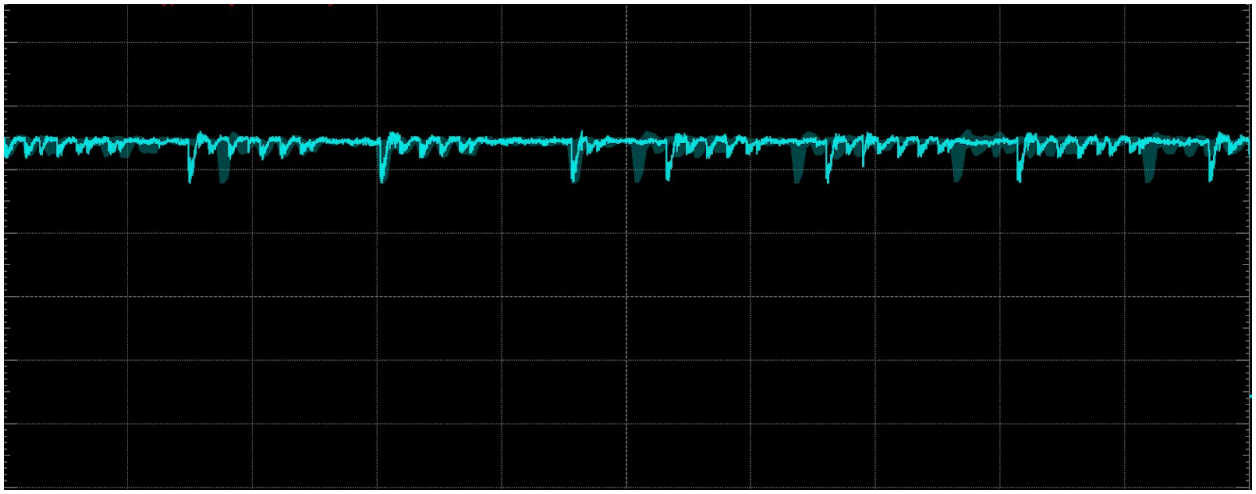

Figure3: SPA trace when the value of e is **00010000**

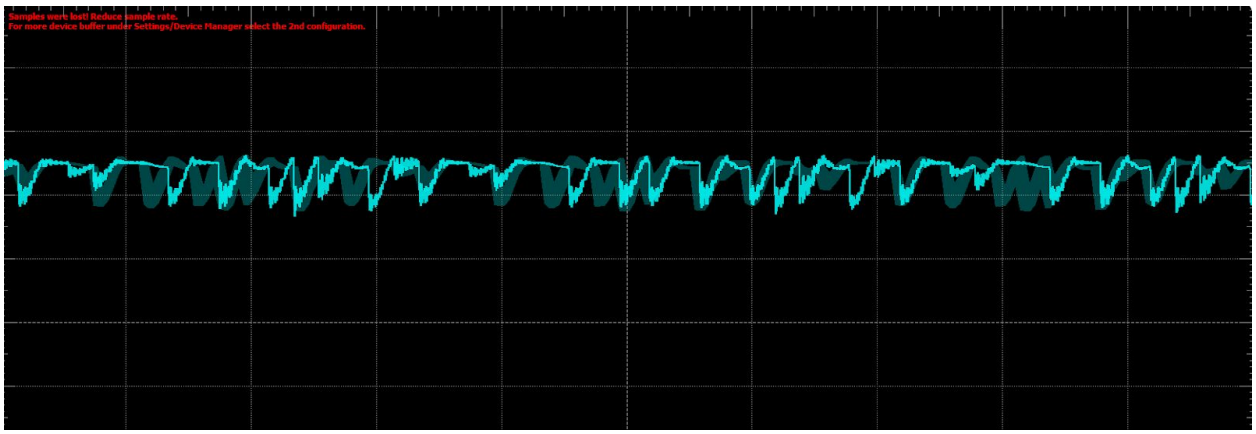


Figure4: SPA trace when the value of e is **01111110**

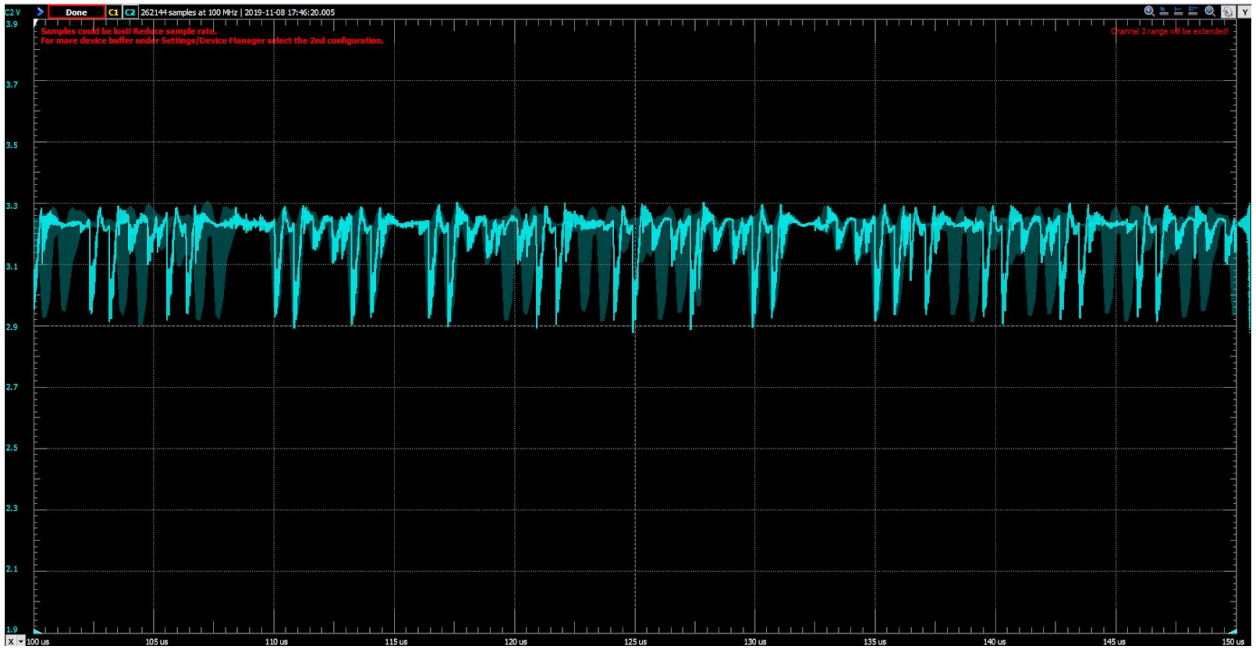


Figure5: SPA trace for the given **a_sof** file

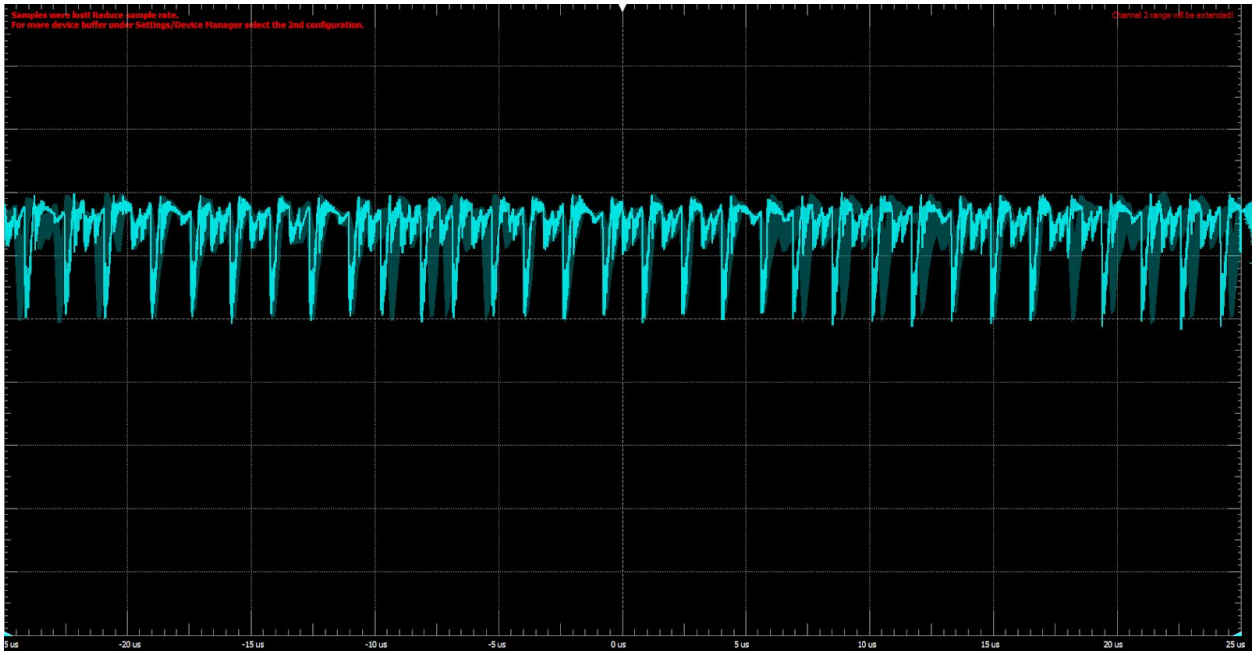


Figure6: SPA trace for the given **b_sof** file

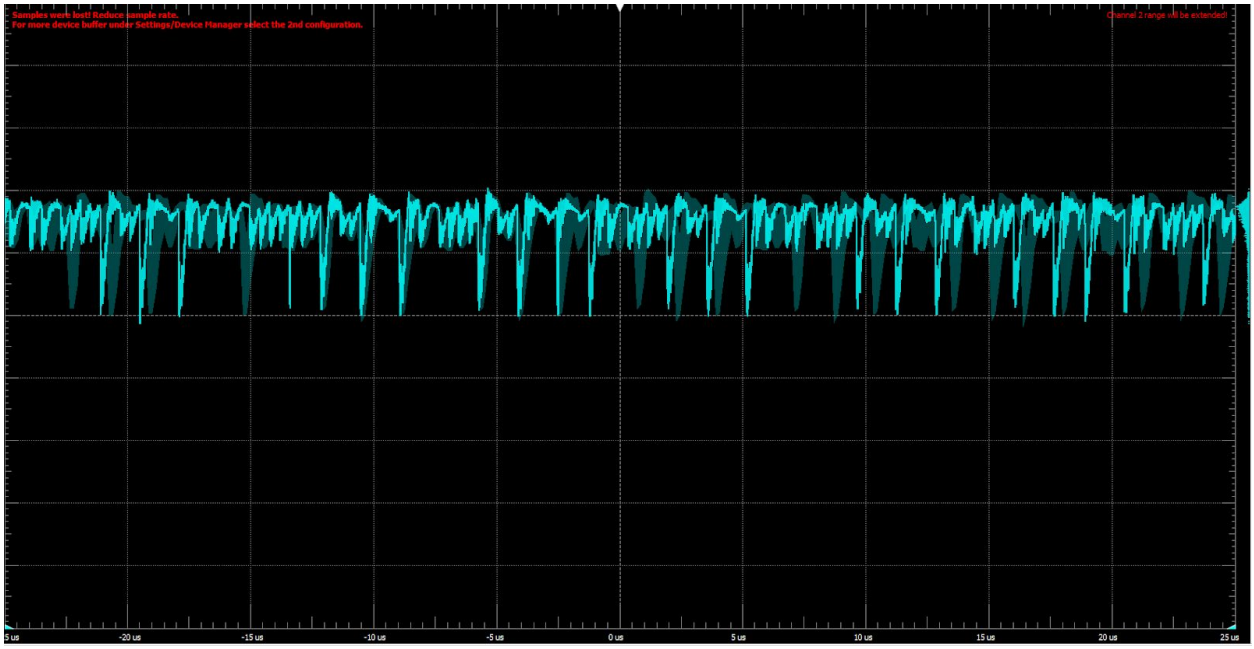


Figure7: SPA trace for the given **c_sof** file

**Part II:**

1. WE implemented the DES in Quartus and measured the power consumption trace. The screenshot is attached below.

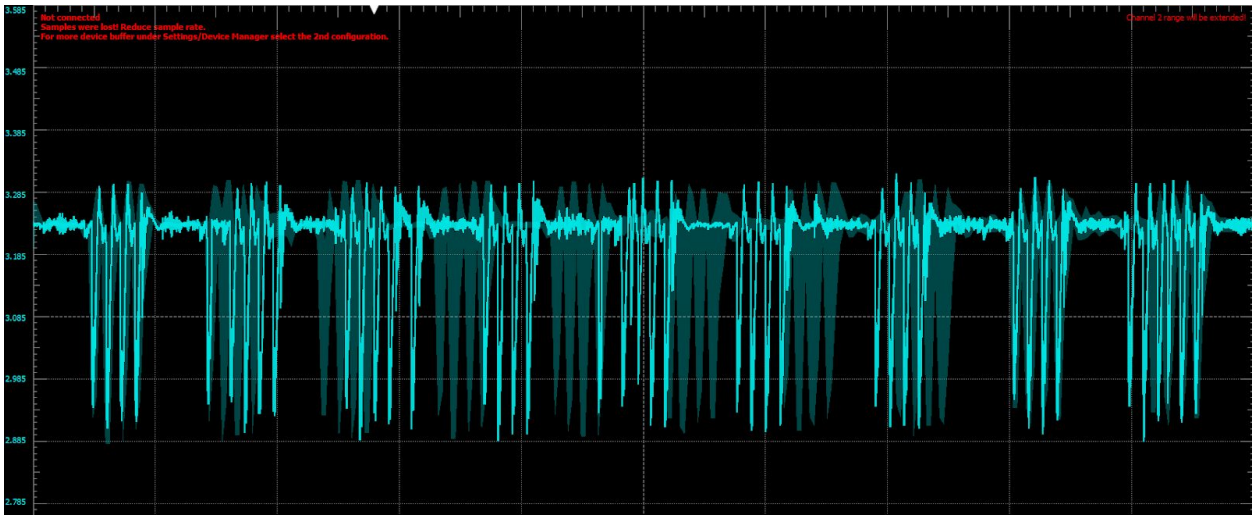


Figure8: Power signal trace for DES implementation for 16 round

2. The first and last round of the DES are targeted by the attackers because the key is provided in those two rounds. So, if they can trace the power trace properly in these two rounds, it is easier for them to get the key.
3. In the DES algorithm, the 56 bits are reduced to 48 bits as only 48 bits are used to encrypt the data. But to get the 56 bits from the 48 bits, we need to know the permutation table

II. Referring to the following link, we know how to use PC-2 table to get the 48 bits. Such as, the first bit of $K_n$ is the 14th bit of $C_nD_n$, the second bit the 17th, and so on, ending with the 48th bit of $K_n$ being the 32th bit of $C_nD_n$. If we do the reverse and know the complete table for permutation, we would get to know the rest of the bits. http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm

4. The 4-bit output of the first sbox does not go to the MSB in the output of the crp module. S[1], S[2], S[3], and S[4] go to the P[9], P[17], P[23], and P[31], location, respectively.

5. The Matlab code is attached with the report where the selection function is implemented.
6. We run the 80000 traces and obtained the difference trace of the average of the two groups for K = 0 to 63. We have attached the screenshot here. Here, some of the signals have spikes but it seems, plot 4 has the highest spike. So, the 6-bit key is 4 for S_box_1.

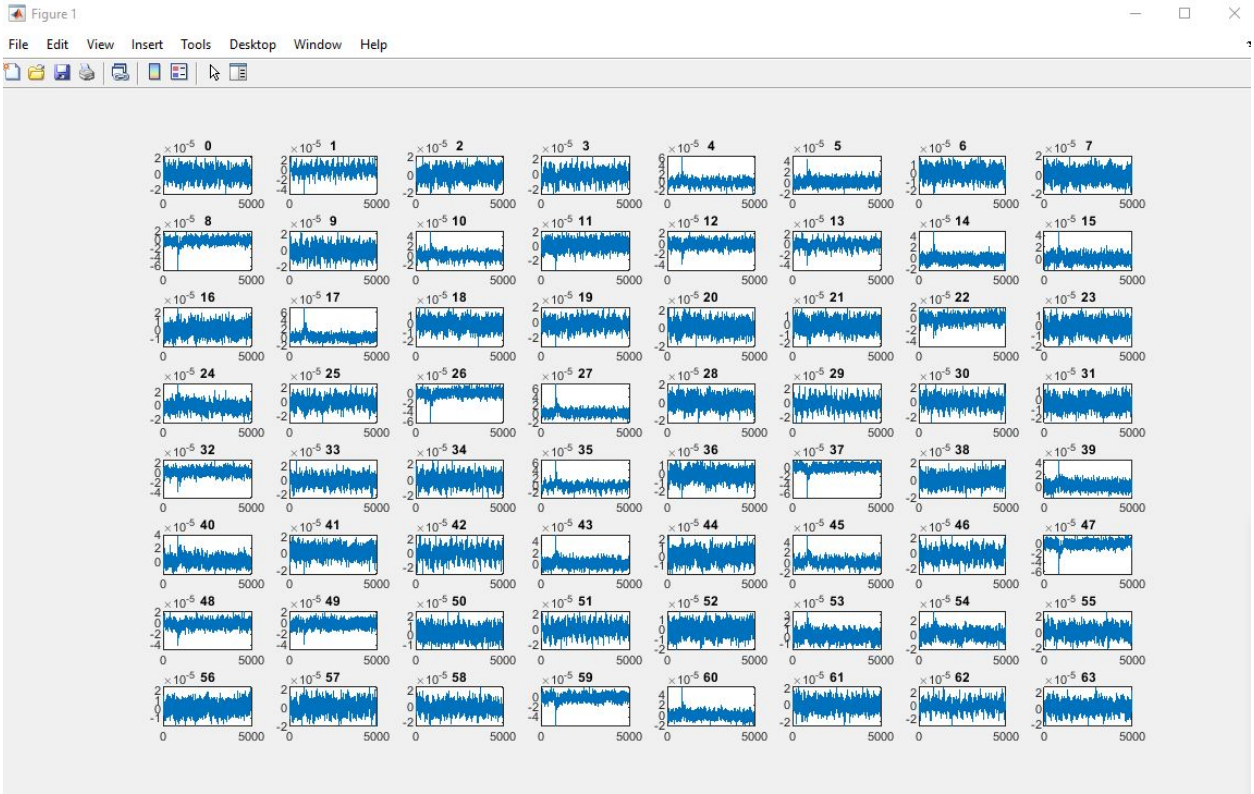


Figure 9: DPA analysis for 6 bit key of S_Box 1

7. We tried to run the code multiple times but as it takes 7-8 hrs to finish, we were able to get two more runs and keys for S_Box 2 and 3. The screenshots are attached below. From figure 10, we see that plot 33 and 43 both have the higher pick. So, 33 or 43 is the key for S_Box 2. From figure 11, we see the plot 15 has the highest spike. So the key is 15 for S_Box 3.

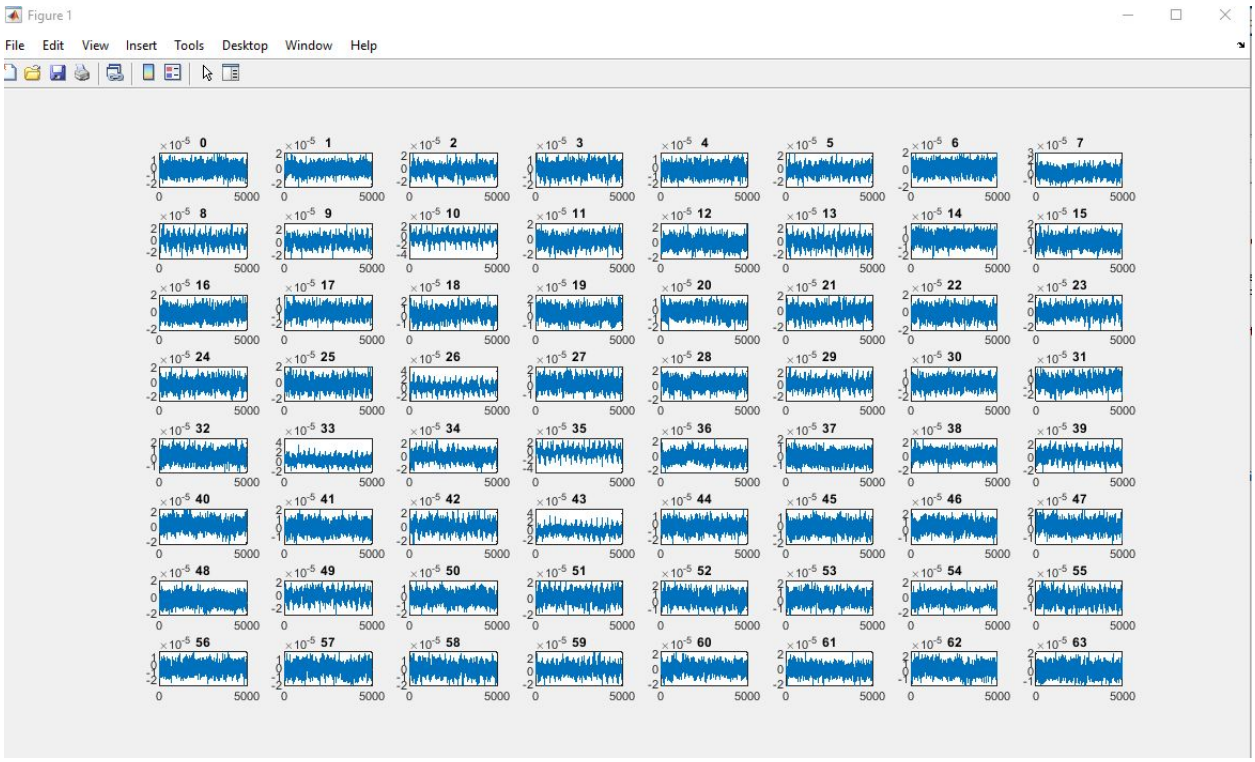


Figure 10: DPA analysis for 6 bit key of S_Box 2

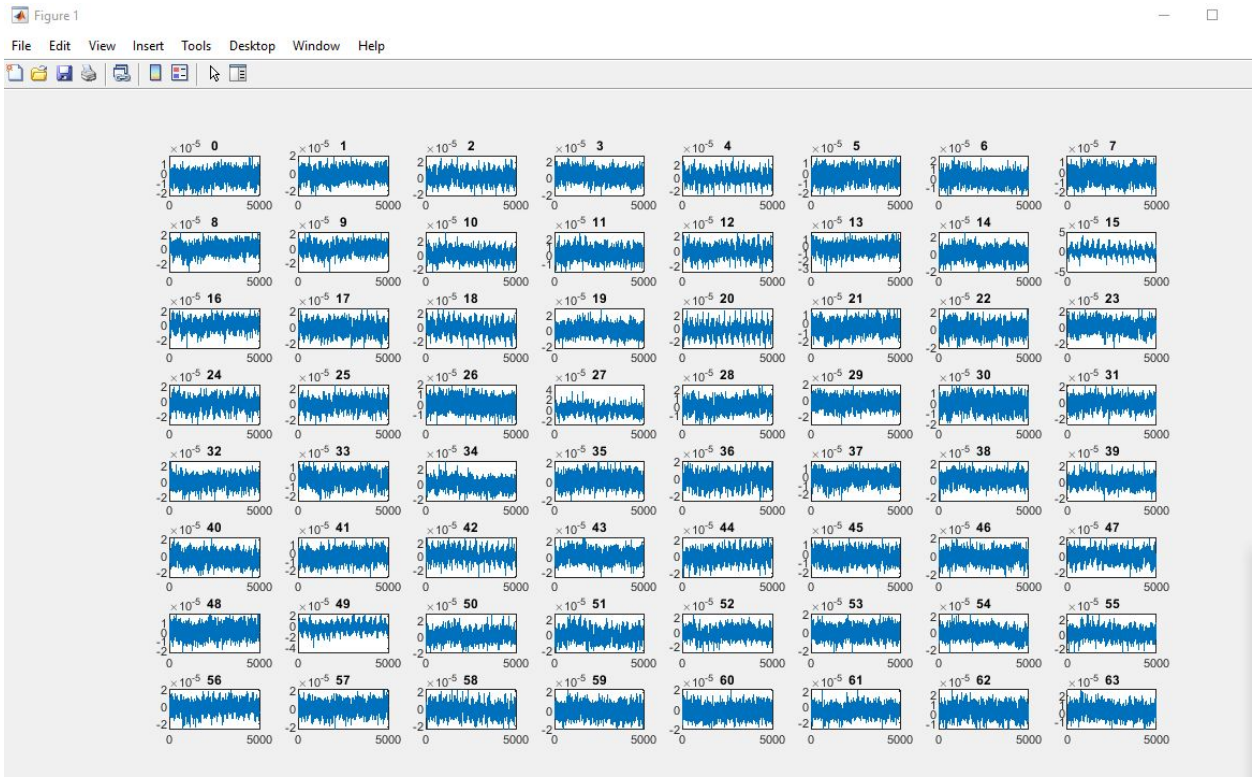


Figure 11: DPA analysis for 6 bit key of S_Box 3

**Discussion and Summary:**

In this experiment, we did the Simple Power Analysis (SPA) and Differential Power Analysis (DPA) to do the side channel attack on DES encryption. We face several issues to do the experiment. First, we connected the analog discovery channel 2 pin and ground pin in a wrong way and burnt the current sensing resistor. Our TA gave us a new resistor and we did the experiment properly. Second, in DPA analysis, after writing the Matlab code, we run the code for 80000 traces and it took forever!!! We saw that it takes approximately 7-8 hrs on our computer. That's why we were able to run only three times to get only 3 6-bit keys. Unfortunately we were not able to finish the rest of the bits. But if we get enough time, we could get the complete 48 bit key. Overall, it is a good learning experiment.