UDYAM'22

CIRCUIT SUTRA TECHNOLOGIES
Modelling Silicon Dreams

# ICHIP (ROUND-02)

Congratulations !!

Now you know how images work and how you can modify some properties of image to suit your need.

Even if you could not completely finish the previous problem statement, there's nothing to be sad about. We expect that you at least got to know about Image Processing. You can test your understanding in this round as well.

The world we live in is nothing but a huge web of networks. Today, billions of bits are generated and communicated daily.

Communication is a strong and useful tool that connects the entire human race in a single string. Any communication medium depends on a number of publicly active aspects for the transmission of data from one end to the other. The data may be extremely sensitive ranging from a simple account password to information related to nuclear weapons.

As the data needs to travel through different public domains to reach the destination, it becomes important to provide for some mechanism for secure transmission of this data.

The answer to these concerns is simple. What if we **encrypt** the data at the source end and then **decrypt** it (i.e. convert it back into the original form) at the receiver's end? This way the data, while in midst of transmission, is in random and useless form and so communication is secured.

You can read about CRYPTOGRAPHY here :

www.geeksforgeeks.org/cryptography-introduction

There are many schemes to encrypt data but not all can be efficiently modelled as hardware circuits. In this round we will be learning about a famous Stream Cipher that is used for over-the-air communication privacy in the GSM cellular telephone standard.

This Cipher is called A5/1 algorithm.

You can follow the links here to gain knowledge about the working of A5/1 algorithm.

https://en.wikipedia.org/wiki/A5/1

https://www.cryptographynotes.com/2019/02/symmetric-stream-a5by1-algorithm-linear-feedback-shift-register.html

https://youtu.be/LgZAI3DdUA4

You can download the image for this round here :

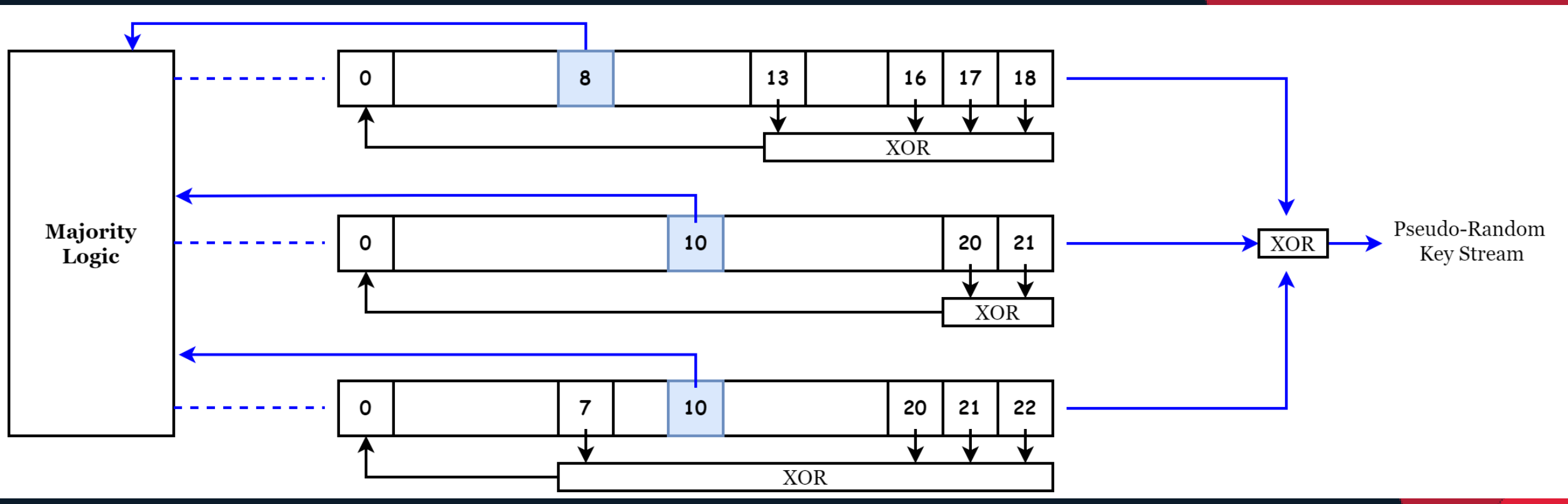
https://drive.google.com/file/d/1mlha9VnlYN2qZ4G-qZpvUsnBzYqnYRBH/view?usp=sharing

UDYAM'22

CIRCUIT SUTRA TECHNOLOGIES
Modelling Silicon Dreams

# TASK :

You have a 256 x 256 grayscale image which you want to encrypt using A5/1 stream cipher.




# STEPS :

1. Initialize the LFSRs with zeros.
2. Feed the secret key bit by bit ignoring the majority logic.
3. After 64 bit secret key, feed the known 22bit public key (bit by bit) ignoring the majority logic.
4. Step 3 marks the end of initialization step. Now clock the LFSRs 100 times with Majority Logic. Ignore these outputs.
5. Now we start the generation of pseudo random keystream. Clock the LFSRs using Majority Logic and XOR the LSB bits of LFSRs to get single bit key.
6. XOR this generated key with the input bit to get the encrypted bit.

We are providing 2 designs for image encryption. You have to submit any one of them.

*[**NOTE** : Design 1 is simpler and hence has less weightage than Design 2. So, try design 2 if you can, to get more points.]*

## *DESIGN-1*

Each image pixel contains 8 bit of data. You have 256 x 256 = 65536 pixels. You already know how to generate pseudorandom key.

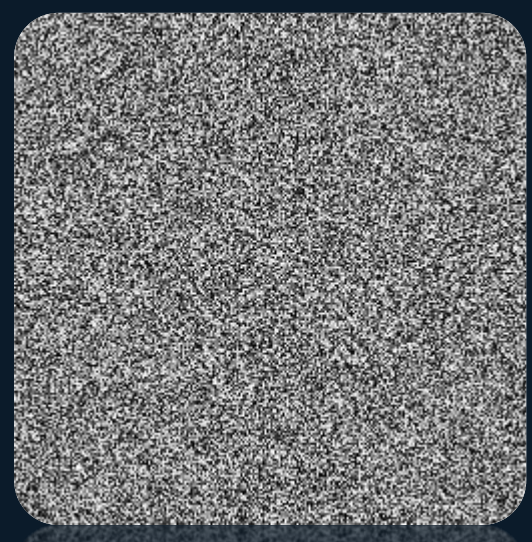
Simply generate 65536 x 8 = 524288 bits of key stream and XOR this bit by bit with the image bits. Store the encrypted bits in some memory (8 bit register array) and then write it to a text file using $writememh(). This will generate the encrypted image.

You can decrypt the image simply by following the exact steps and giving the encrypted image as input.

[*Understand that this design is very poor and is not very secure. This option is provided so that you have at least a working encryption algorithm to submit for evaluation.*]

Encryption

Decryption



# DESIGN -2

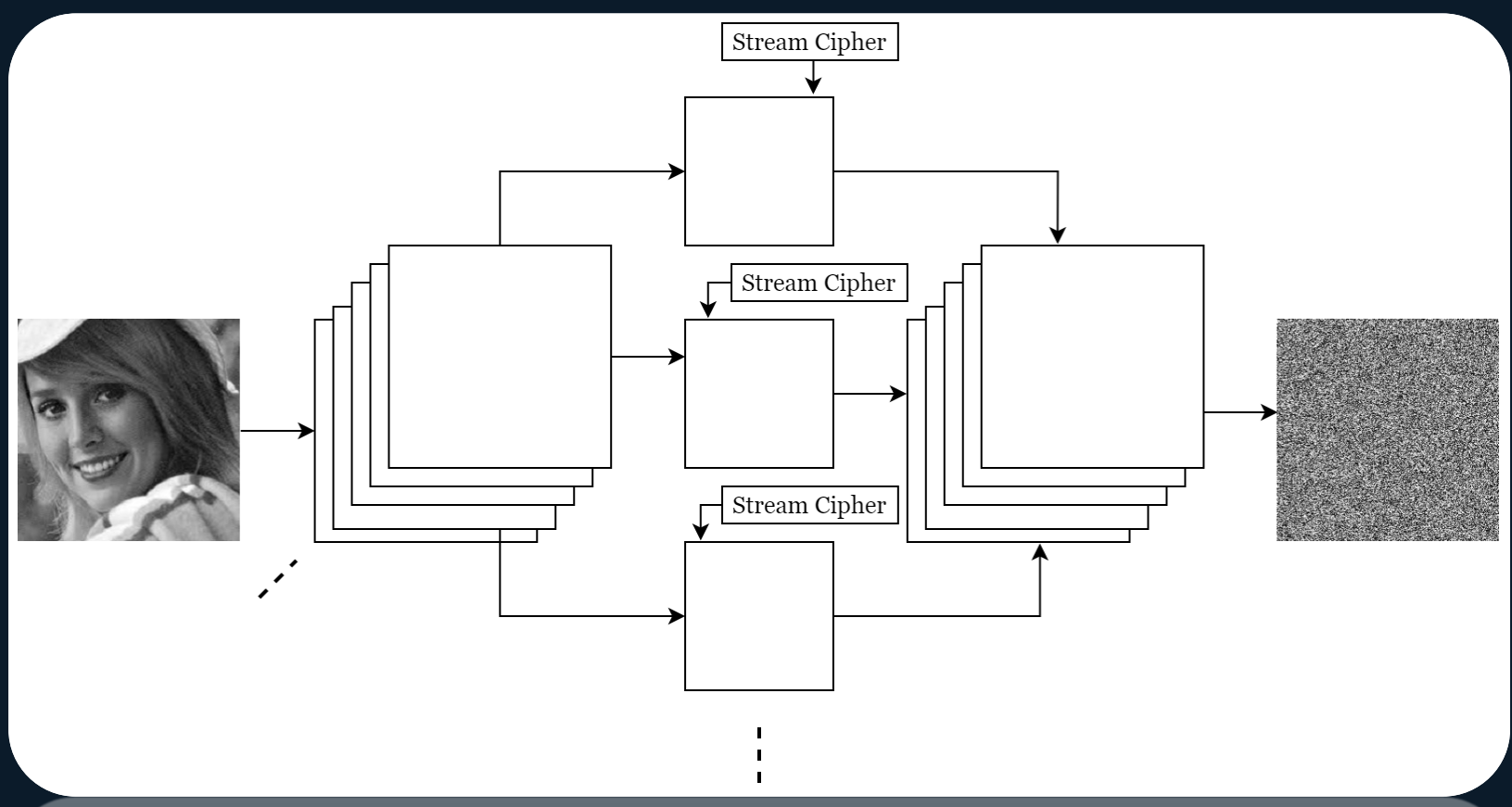
Special care needs to be taken while encrypting images. This is because adjacent pixel values of any image are highly correlated. This can lead to easy attacks and failure of encryption mechanism.

To overcome this problem, we propose an interesting approach. We break the image into 8 bit-planes. Each plane contains 256 x 256 bits.

For example, if the first pixel has value 8'b0111_0011 then the first plane will contain only 1'b0, second plane will contain 1'b1, third again 1'b1 and so on. Similarly for other pixels. Thus, our image will be actually broken into 8 planes of 256 x 256 pixels with each pixel only 1 bit in length.

Now we simply encrypt the image plane by plane. But to make the cipher strong, before the start of a new plane, we re-initialize the LFSRs following the same steps mentioned earlier.

Breaking the image in 8 bit-planes requires proper care and attention should be given on the position of different bits while writing the encrypted image in memory.



UDYAM'22

CIRCUIT SUTRA TECHNOLOGIES
Modelling Silicon Dreams

## CONSTRAINTS :

1. Clock Frequency should be 100 GHz.
2. Design Module and Testbench can be according to your thoughts and your approach.
3. 64 bit secret key = “hardware”
4. 22 bit public key = 22’b11_0100_1110_0001_1001_0001

## SUBMISSIONS :

Create a folder (name same as Team name) with following files :

1. Text files of design module and testbench module.
2. Snapshot of timing diagram
3. Encrypted & Decrypted .jpg images
4. A text file briefly describing your approach

Submit the zip file on D2C before deadline.

## CONTACTS : (WhatsApp Only)

Nishant Kumar Robin : 7478119798
Ujjawal Agrawal : 7394059800

** Any confusion/doubt regarding problem statement should be asked on any of the WhatsApp numbers provided here. No other means of communication will be entertained.**