PROJECT PROPOSALS

GROUP MEMBERS

- Aditi Das – 2020CSB1064
- Jugal Chapatwala – 2020CSB1082
- Shruti Sikri – 2020CSB1128

PROPOSAL NUMBER 1

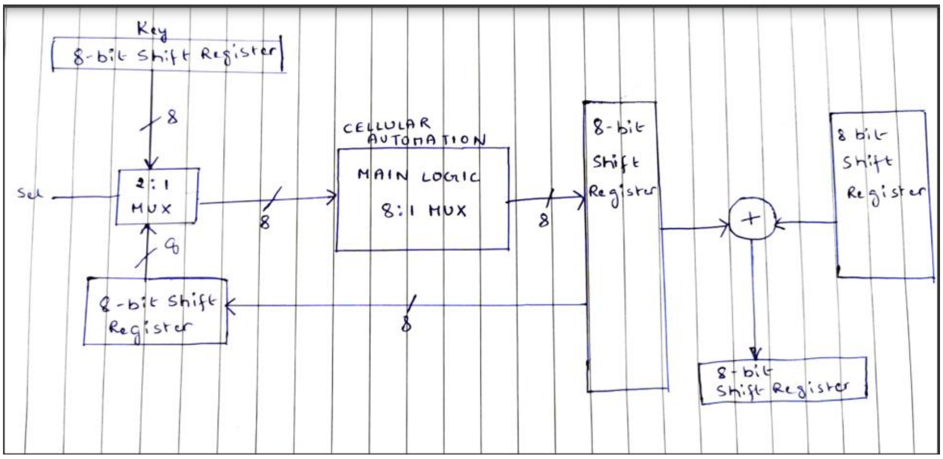
DATA ENCRYPTION AND DECRYPTION SYSTEM

OVERALL OBJECTIVE-

To implement a cryptographic device in Verilog based on cellular automation.

IMPLEMENTATION-

It will be implemented in Verilog using the register transfer logic (RTL).



Here we take an 8-bit array. The initial state of the array will be stored in a parallel-load parallel-out register. The next state of the register is also stored in a parallel-load parallel-out register. The next state is derived from the initial state. So between the registers, we have the logic section that implements the rule 30 of cellular automation by Stephen Wolfram.. Since this is an iterative process, there will be feedback from the next state register to the initial state register.

Before the feedback begins, we must input the required key into the initial state register. This describes the key generation circuit. For the encryption we decided to go for the XOR operation. Data has to be fed in terms of its ASCII values and this data is XORed with the sequence obtained from the key generation circuit. The key generation circuit produces a new sequence for every clock cycle.

This was the basic idea to implement encryption. Decryption is implemented in similar ways.

FUNCTIONALITY-

Encryption: The methodology of converting the original message into cipher text is called Encryption. Cipher text is obtained by converting readable and understandable data into unreadable form.

DATA->TRANSFORMATION->CIPHER TEXT

Decryption: It is the method of converting back the cipher text into plain text. The unreadable form of message is converted back into readable form or the un intelligible form of message is converted into readable or intelligible form.

ENCRYPTED DATA->TRANSFORMATION->READABLE FORM

Encryption and decryption play a vital role in today’s digital world. So we thought of implementing an encrypting and decrypting device here.

PROPOSAL NUMBER 2

IMPLEMENTING A SIMPLE ALARM CLOCK

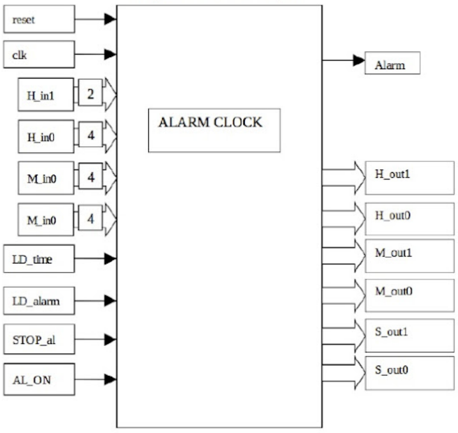
OVERALL OBJECTIVE-

The alarm clock outputs a real-time clock with a 24-hour format and also provides an alarm feature. The clock time can also be set manually.

IMPLEMENTATION-

It will be implemented in Verilog using register transfer logic.

Figure 1: Block Diagram



We take the following inputs:

- A 2-bit input used to set the most significant hour digit of the clock
- A 4-bit input used to set the least significant hour digit of the clock
- A 4-bit input used to set the most significant minute digit of the clock
- A 4-bit input used to set the least significant minute digit of the clock
- LD-TIME : used to set time on the clock
- LD-ALARM: used to set the alarm on the clock
- STOP_al and AL_ON to control the working of alarm output

FUNCTIONALITY-

We use the inputs to calculate the following outputs:

- The most significant digit of the hour. Valid values are 0 to 2.

- The least significant digit of the hour. Valid values are 0 to 9.
- The most significant digit of the minute. Valid values are 0 to 5.
- The least significant digit of the minute. Valid values are 0 to 9.
- The most significant digit of the seconds. Valid values are 0 to 5.
- The least significant digit of the minute. Valid values are 0 to 9.

Finally, the current time is displayed along with the feature of setting alarm. When the current time coincides with the set-alarm time, the alarm signal goes high.

PROPOSAL NUMBER 3

BITCOIN MINER

OVERALL OBJECTIVE-

To use SHA-256 algorithm to calculate the nonce for a bitcoin block.

IMPLEMENTATION-

It will be implemented in Verilog using register transfer logic(RTL).

FUNCTIONALITY-

First we will create a module to calculate the SHA-256 hash of the given input. Then a counter will be used to go through the nonce and validate the block based on the given network difficulty of the bitcoin network.