Make a 2 page writeup of your project , (preferably Time new Roman Size 12)

Abstract of the project :The content should be  the title, abstract , Architecture/ Flow graph of the project  - 1 page (minimum)

Justification of the Project: The related references (links /documents ) and what you have infered in each reference(minimum 5 reference) and modules

 of the project work. - 1 page (minimum)

Write the team number , Register numbers  and name of the complete team , Title in bold.. in a seperate page.

 Use a legible font .

**ABSTRACT**

Blockchain is an ever-growing list of records, called blocks, that are secured using cryptography.

The blockchain consists of blocks connected to each other with the SHA 256 algorithm. The SHA 256 algorithm takes any length of character text as input and outputs a 256-bit text.

In today's blockchain system, it is known that millions of machines simultaneously perform calculations that take hours to satisfy a condition. All these devices can perform these calculations by means of data communication among themselves.

Similarly, in this project, the sharing of common data and parallel computing processes between today's computers will be implemented using threads.

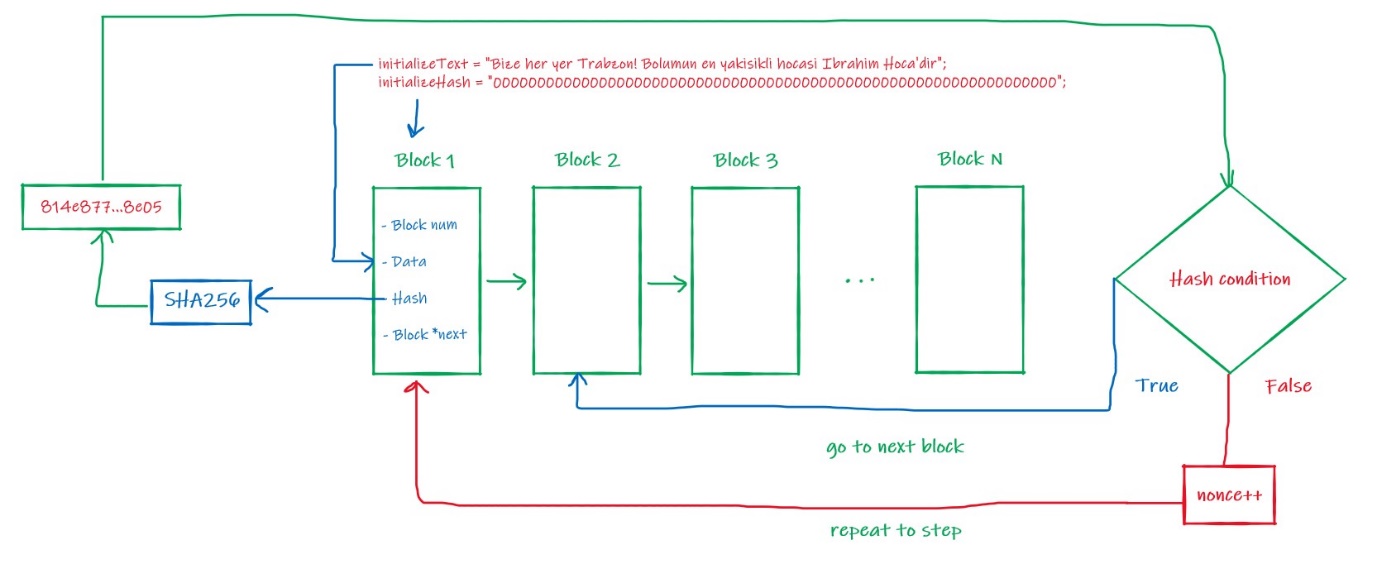
The purpose of the simulation is to implement the blockchain working application with multiple threads/cores instead of multiple machines.

**ARCHITECTURE**

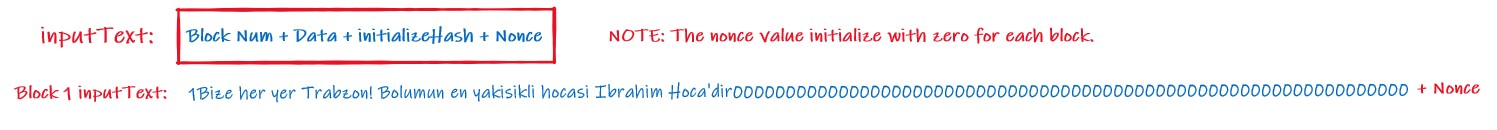
Initialize Hash and Text definitions ( you can change it if you want :) )

Text: “Bize her yer Trabzon! Bolumun en yakisikli hocasi Ibrahim Hoca'dir”

Hash: “0000000000000000000000000000000000000000000000000000000000000000”

[](https://github.com/fbasatemur/openmp_blockchain/blob/main/info/bitminer_design.jpg)

Firstly, Data is created for the current block. With "initilizeHash" and "initializeText" texts, Block 1 "inputText" value is generated as follows. "Data" value for Block 1 is "initializeText" value in initialize time:

[](https://github.com/fbasatemur/openmp_blockchain/blob/main/info/b1_input_text.jpg)

Then the value of "inputText" is given to the SHA 256 algorithm. The resulting value is given to the Hash acceptance condition.

Hash acceptance condition consists of 2 sub-conditions:

1. The resulting hash value must start with 0 as much as the Block number.
2. The character to the right of the characters of zero must be nonzero.

For example, the result "01as8we...." is acceptable for Block 1, while the result "001as8we...." is not.

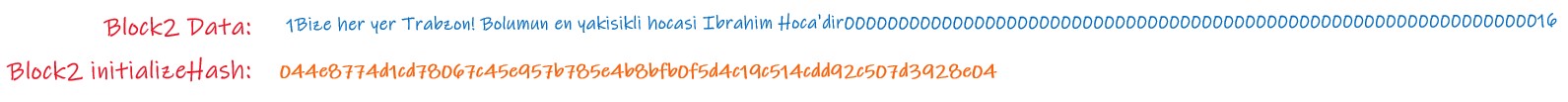
If the condition is met, the next block is passed.

Suppose that the nonce and hash values are found as follows and the condition is satisfied.

Nonce = 16

Hash = 044e8774d1cd78067c45e957b785e4b8bfb0f5d4c19c514cdd92c507d3928e04

In this case, "inputText" and "Data" values for Block 2 are found as follows:

[](https://github.com/fbasatemur/openmp_blockchain/blob/main/info/b2_data_hash.jpg)

According to these values, Block 2 "inputText" value is found as follows:

[](https://github.com/fbasatemur/openmp_blockchain/blob/main/info/b2_input_text.jpg)

MODULES

REFERENCES

1. Zhang, Z., 2016. *A multi-threaded cryptographic pseudorandom number generator test suite*. Naval Postgraduate School Monterey United States.

This paper spoke about generating random numbers suitable for cryptographic purposes using multithreads.