Topic : Implementation of bitcoin hash using SHA-256

Team members:

1) Yomi Karthik Rupesh
2) Sreejita Saha
3) Vikas Rao
4) Rajath Bindiganavile

Abstract:

In this project an ASIC capable of performing the SHA-256 Hash Function is presented. A hash function is one that encrypts a Data of arbitrary size into a data of fixed size. For a given input Data, encrypted Hash will always remain the same, but modifying the data by even one bit will completely change the hash output. The aim of our project is to realize BitCoin-Hash which uses SHA-256 (Secure hash algorithm 256) cryptographic hash function belonging to the SHA-2 family. The said algorithm is used in bitcoin mining for maintaining data integrity, hashcach cost-function and deriving block chains by generating verifiably "random" numbers in a way that requires a predictable amount of CPU effort. It compares the result computed by the hash function, with a known and expected hash value.

Introduction and motivation:

A cryptographic hash function essentially takes input data which can be of practically any size, and transforms it, in an effectively-impossible to reverse or to predict way, into a relatively compact string (in the case of SHA-256 the hash is 32 bytes). Making the slightest change to the input data changes its hash unpredictably, so nobody can create a different block of data that gives exactly the same hash. Therefore, by being given a compact hash, you can confirm that it matches only a particular input datum, and in bitcoin the input data being a block-chain is significantly larger than the SHA-256 hash. This way, Bitcoin blocks don't have to contain serial numbers, as blocks can be identified by their hash, which serves the dual purpose of identification as well as integrity verification. An identification string that also provides its own integrity is called a self-certifying identifier.

The hashcash difficulty factor is achieved by requiring that the hash output has a number of leading zeros. Technically, to allow more fine-grained control than Hashcash number of leading 0-bits method, Bitcoin extends the hashcash solution definition by treating the hash as a large big-endian integer, and checking that the integer is below a certain threshold. The hashcash cost-function iterates by perturbing data in the block by a nonce value, until the data in the block hashes to produce an integer below the threshold - which takes a lot of processing power. This low hash value for the block serves as an easily-verifiable proof of work - every node on the network can instantly verify that the block meets the required criteria.

With this framework, we are able to achieve the essential functions of the Bitcoin system. We have verifiable ownership of bitcoins, and a distributed database of all transactions, which prevents double spending

Scope of the project :

In bitcoin mining,each coin is associated with its current owner's public ECDSA key. When bitcoins are

sent from a sender to a receiver,a message the new owner's public key is added along with the private key of the sender. When this transaction is broadcast to the bitcoin network, this lets everyone know that the new owner of these coins is the owner of the new key. The addition of the private key of the sender acts like a signature and verifies the authenticity of the bitcoins. The complete history of transactions is kept by everyone, so anyone can verify who is the current owner of any particular group of coins.
This complete record of transactions is kept in the block chain, which is a sequence of records called blocks. All computers in the network have a copy of the block chain, which they keep updated by passing along new blocks to each other. Each block contains a group of transactions that have been sent since the previous block. In order to preserve the integrity of the block chain, each block in the chain confirms the integrity of the previous one, all the way back to the first one, the genesis block. Record insertion is costly because each block must meet certain requirements that make it difficult to generate a valid block. This way, no party can overwrite previous records by just forking the chain.

To make generating bitcoins difficult the Hashcash cost-function is used. Hashcash is the first secure efficiently verifiable cost-function or proof-of-work function. The beauty of hashcash is that is is non-interactive and has no secret keys that have to be managed by a central server or relying party; hashcash is as a result fully distributed and infinitely scalable. (Hashcash uses symmetric key cryptogaphy, namely a one-way hashcash function - typically SHA-256).

In bitcoin, integrity, block-chaining, and the hashcash cost-function all use SHA256 as the underlying cryptographic hash function.

Design approach :

SHA-256 operates in this manner : The message to be hashed is 1
(padded) with frst its length in such a way that the result is a multiple of 512 bits long, and then
(2) parsed into 512-bit message blocks M1, M2,..... MN .

The message blocks are processed one at a time: Beginning with a fixed initial hash value H , sequentially compute

Hi = Hi + CM (Hi) ;

where C is the SHA-256 compression function and + means word-wise mod 2 addition. HN is the hash of M .

Estimation of design size:

The circuit needed has a few cells with very high requirements (32 input XOR,AND,OR or ALU,64bit shift register,etc) which will require a few trade offs to be placed during design. Hence the total size estimation may not be accurately be possible. As per our reasonable assumption, we are predicting close to 10,000 gates for the implmentation of the stand alone ASIC (with ROM for holding initial data; upto 64-bits of the fractional part of square and cube roots of primes are required).

Estimated tasks and milestones with completion dates:

Third week of november - schematic and 40% library elments layout implementation.
end of november - library for the project fully functional .

first week of december - documentation and fully functional ASIC.

Finally, understanding the size may turn out to be very large, we do hope to fabricate our design.

References :
wikipedia
en.bitcoin.it
description of SHA-256,384,512