Building a Bitcoin Miner on an FPGA

Samuel Oliveira, Filipe Soares, Guilherme Flach, Marcelo Johann, Ricardo Reis
{skoliveira, fasoares, gaflach, johann, reis}@inf.ufrgs.br

UFRGS – Universidade Federal do Rio Grande do Sul

Abstract

Bitcoin is decentralized peer-to-peer electronic cash system. Differently from regular currencies, Bitcoin does not rely on any centralized entity like banks or governments to ensure the system reliability. Therefore money is sent directly from person to person without going through a third party financial institution Bitcoins are uniquely generated by a process called Bitcoin mining where miners are rearward for computing power spent to support the network. In this work we discuss the process of Bitcoin mining and develop a new Bitcoin miner on an FPGA.

1. Introduction

Bitcoin is a peer-to-peer electronic cash system which is sent directly from person to person without going through a financial institution. The lack of a centralized entity like banks, clearing houses or governments translates into much lower fees. Moreover Bitcoins can be used anywhere by anyone, accounts cannot be frozen and there are no prerequisite or restrictions to start using Bitcoins.

Launched in 2009 by a hacker self named Satoshi Nakamoto, the Bitcoin network has become one of the most powerful distributed computing networks [1]. At the time this paper was written, 1 Bitcoin (shortened as 1 BTC) was traded by around U$ 5 given to the Bitcoin network a total valuation of U$ 40M [2].

Bitcoins are create all over the Internet by anybody running a program called Bitcoin miner, which works to keep the Bitcoin ecosystem. Each miner tries to find a valid solution for a hard cryptographic problem. Every time a valid solution is found, the miner is rewarded with some amount of Bitcoins. Mining is the unique way to generate Bitcoins and the mining difficulty is adjusted automatically by the Bitcoin network so that the number of Bitcoins generated increases in a steady and predictable way.

In this work we discuss the process of Bitcoin mining and present a new Bitcoin miner implemented on an FPGA. We start by providing a brief introduction of Bitcoin network on Section 2. Next, we present some details of the Bitcoin mining process on Section 3. Then we outline our FPGA implementation of a Bitcoin miner on Section 4. And finally on Section 5 we conclude and present some future work.

2. Bitcoin Network

Running over a distributed peer-to-peer network, the Bitcoin was the first cash system to feasibly solve the double-spending problem which arises when a centralized, trusted third party is not present [3]. The network stores transactions by hashing them into a chronological chain of
hash-based proof-of-work. The proof-of-work concept ensures that the chain could not be modified without redoing the proof-of-work. The longest chain of transaction serves as proof of sequence of events preventing attackers to execute transaction with the same Bitcoins.

As long as the majority computational power is cooperating in support to the Bitcoin network, attackers could not generate a longer chain. As miners work only on the longest chain any alternative chain is discarded by the network. Note that multiple longest chain may exists at same time, however at some point one of them becomes the longest one and the second dis discarded.

2.1. Transaction Chain

The transaction chain is compose by blocks where the transactions are indeed stored. As seen in Figure 1, multiple chains may exist at same time in the network, however the shorter ones are discarded as the time goes by in behalf of the longest one. The longest chain is the one which required the great amount of work to be generated, not necessarily the one with more blocks.

Miners try to extend the longest chain by finding a next block for it (white blocks in Figure 1). The next block is attached to the longest chain and accepted by other participants when its hash has some very unlikely property as discussed on Section 3. The attached block includes a special transaction which generates new Bitcoin rewarding the miner that “solved” that block. Note that several next blocks are tried at same time by different miners around the network. The first one to solve a block is rewarded.

As the longest chain grows transactions become deeply buried into the chain and trying to modify them by starting a concurrent chain becomes impractical. This properties ensures that the Bitcoin network stay reliable as long as an attacker does not control the most nodes on the network.

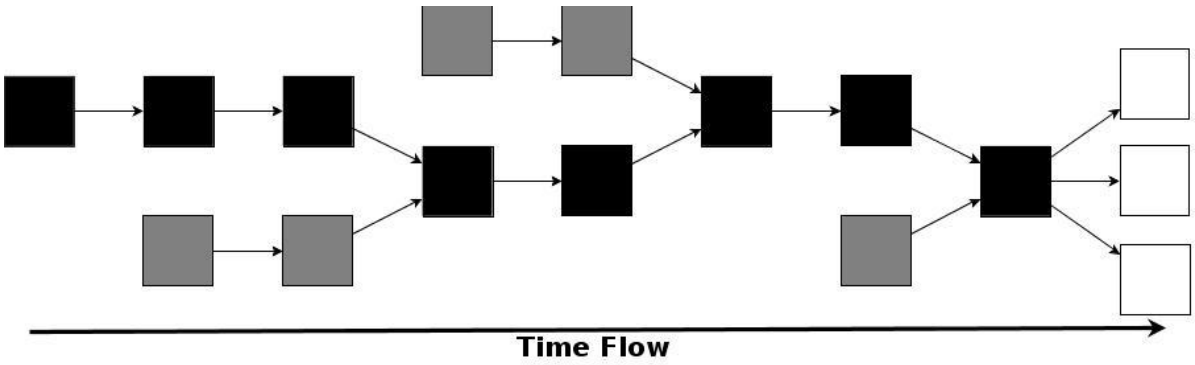


Fig. 1 – Bitcoin Transaction Chain

3. Bitcoin Mining

Bitcoin mining is essentially the process of cryptographic hashing a block header looking for a hash that, when interpreted as big number, has the very unlikely property of being bellow some small target value. Such target is automatically adjusted by the Bitcoin network to keep the generation of Bitcoins in a constant pace. When the computing power of the network increases, the target decreases causing the mining process to become more difficult. On the other hand, if the computing power of the network decreases, the target is increased also increasing the likelihood of finding a hash bellow this target.

Inside each block header, a special field called nonce is reserved for mining purposes only. Miners tries different nonce values looking for a valid hash (a hash bellow the target value). Each time a new nonce is tried, a new hash is obtained. If a nonce generates a valid hash, it is called a golden nonce.

For any good hashing algorithm, there is no known way to find the golden nonce other than by sweeping all possible nonce values. This ensures that in order to find a golden nonce miners have to spend certain amount of computer power bringing out the proof-of-work concept which backs the reliability of the Bitcoin ecosystem.

3.1. Block Header

Data is permanently recorded in the Bitcoin network through blocks. A block is “solved” (published and considered valid by peers) when the hash of the block header is below the current target. The block header consists of 640 bits (80 bytes) as shown in the Table 1. Most of fields are constants, but miners can play with one of them: nonce. Some miners also alter the timestamp field as a workaround to allow more trials per block header, however, in this work, we concentrate only on the nonce field.

Table 1 - Bitcoin Block Header

Field Size	Description	Data type	Comments
4	version	uint32_t	Block version information, based upon the software version creating this block
32	prev_block	char[32]	The hash value of the previous block this particular block references
32	merkle_root	char[32]	The reference to a Merkle tree collection which is a hash of all transactions related to this block
4	timestamp	uint32_t	A timestamp recording when this block was created (Limited to 2106!)
4	bits	uint32_t	The calculated difficulty target being used for this block
4	nonce	uint32_t	The nonce used to generate this block… to allow variations of the header and compute different hashes

3.2. Hash Function

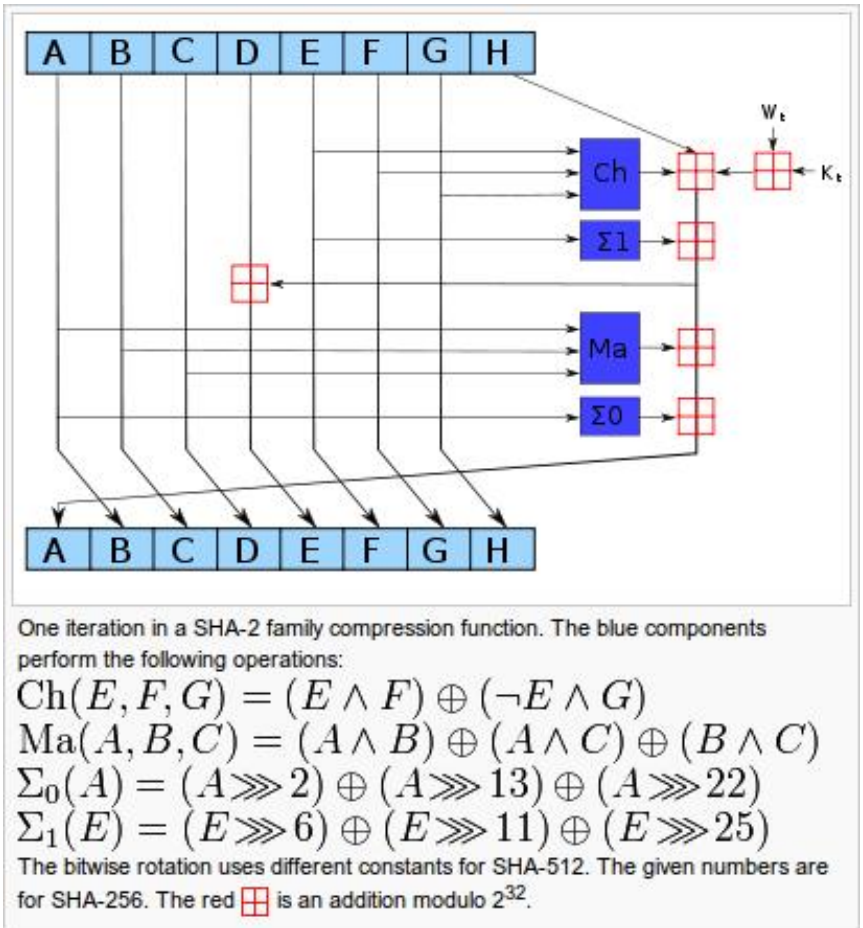
Bitcoin network uses the SHA-256 hash function [5] as the core of its own hash function as shown bellow.

$$hash_{bitcoin}(m) = SHA\text{-}256(SHA\text{-}256(m))$$

3.2.1. SHA-256

SHA-256 is a member of SHA-2 cryptographic hash function family. It has a digest of 256 bits hence its name. So far no collision (two different messages with same hash) has been found.

Messages are sliced in chunk of 512 bits and data is processed as 32-bit big-endian words. The partial digest found for each chunk depends on the previous one (for the first one a constant digest is used). The main structure of the hash function is shown in Figure 2, which is executed 64 times.



One iteration in a SHA-2 family compression function. The blue components perform the following operations:

$$\mathrm{Ch}(E,F,G) = (E \wedge F) \oplus (\neg E \wedge G)$$
$$\mathrm{Ma}(A,B,C) = (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C)$$
$$\Sigma_0(A) = (A \ggg 2) \oplus (A \ggg 13) \oplus (A \ggg 22)$$
$$\Sigma_1(E) = (E \ggg 6) \oplus (E \ggg 11) \oplus (E \ggg 25)$$

The bitwise rotation uses different constants for SHA-512. The given numbers are for SHA-256. The red ⊞ is an addition modulo 2^{32}.

Fig. 2 – SHA-2 family compression function [6]

4. Bitcoin Miner on an FPGA

All the FPGA implementation was done using the hardware description language VHDL. As a golden model, a software Java implementation was also created.

To calculate the hash, we divide the block header in 16 slices with 32 bytes each and use them as serialized input in the same block of logic. This way we reduce the area needed for the implementation with the cost of extra clock cycles and the need of a state machine to control the process

4.1. Serial Interface

In order make the communication between the FPGA board and the computer, we developed both a Java application and a VHDL module that uses the RS-232 serial protocol to send and receive data as depicted in Figure 3. We opted for the serial interface because it is one of the simplest communications protocol available and its low transmission speed does not interfere with the overall system performance. The serial communication consists on sending 44 bytes to the FPGA (32 bytes for the midstate and 12 for the 2nd chunk) and receiving the answer from the FPGA with the 5 bytes of processed data (4 bytes for the nonce and 1 byte to state its validity).

The Java application is also responsible for requesting the block header from the server and submitting the proof-of-work.

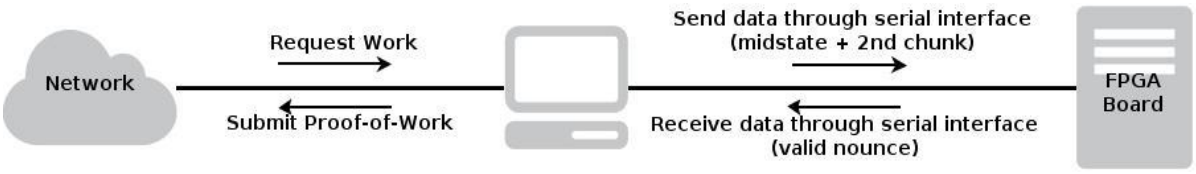


Fig. 3 – Miner Infrastructure

4.2. FPGA Parallelism

One advantage of working with an FPGA board is that we can create as many parallel units of the hash calculation as we can fit on the board. this units are totally independent and each will take its own time to calculate the right hash. with the Virtex-II PRO used in our tests we were able to fit 8 units in the chip but this number could easily increase with the use of other boards with more logic.

5. Conclusion and Future Works

In this work we discussed the overall Bitcoin mining process and built a Bitcoin miner on an FPGA. Our goal was to understand deeply the mining process and create a functional but not high optimized miner. Although the current miner performs only 6 Mhash/s (compared to some 100 Mhash/s GPU miners), the mining framework developed will allow us to easy test new miner designs. We expect to reach 20 Mhash/s using the same FPGA board creating a more compact design and applying some known SHA-256 hardware optimization. Moreover we are studding new ways to further optimize the SHA-256.

Although our first intent is to build a faster Bitcoin miner, any SHA-256 hardware optimization technique is important as the SHA-2 is used in many applications as TLS and SSL, PGP, SSH, S/MIME, and IPsec.

6. References

[1] Reuben Grinberg. “Bitcoin”. Available on http://www.milkeninstitute.org/publications/review/2012_1/22-31MR53.pdf.

[2] http://bitcoincharts.com/

[4] Satoshi Nakamoto. “Bitcoin: A Peer-to-Peer Electronic Cash System”. Available on http://bitcoin.org/bitcoin.pdf.

[5] R. Chaves, G. Kuzmanov, L. Sousa, S. Vassiliadis, "Cost-Efficient SHA Hardware Accelerators", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 16, NO. 8, pp 999-1008, August 2008

[6] Wikipedia. “SHA-2”. Available on http://en.wikipedia.org/wiki/SHA-2.