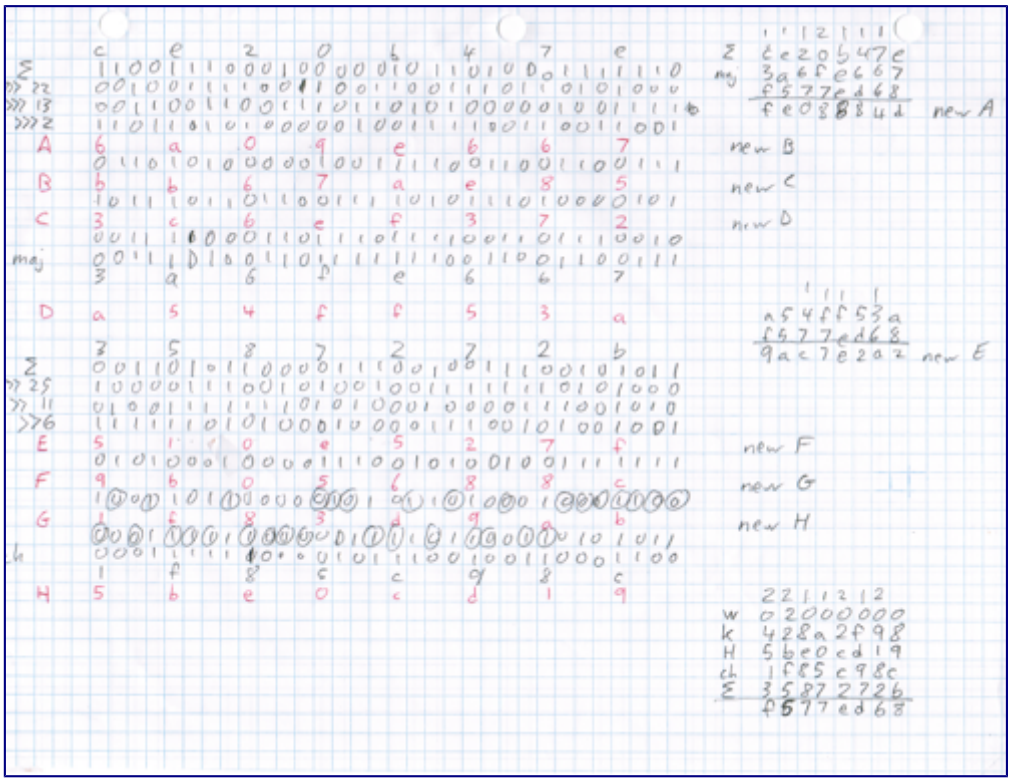
Mining Bitcoin with pencil and paper: 0.67 hashes per day

http://www.righto.com/2014/09/mining-bitcoin-with-pencil-and-paper.html

I decided to see how practical it would be to mine Bitcoin with pencil and paper. It turns out that the SHA-256 algorithm used for mining is pretty simple and can in fact be done by hand. Not surprisingly, the process is extremely slow compared to hardware mining and is entirely impractical. But performing the algorithm manually is a good way to understand exactly how it works.



A pencil-and-paper round of SHA-256

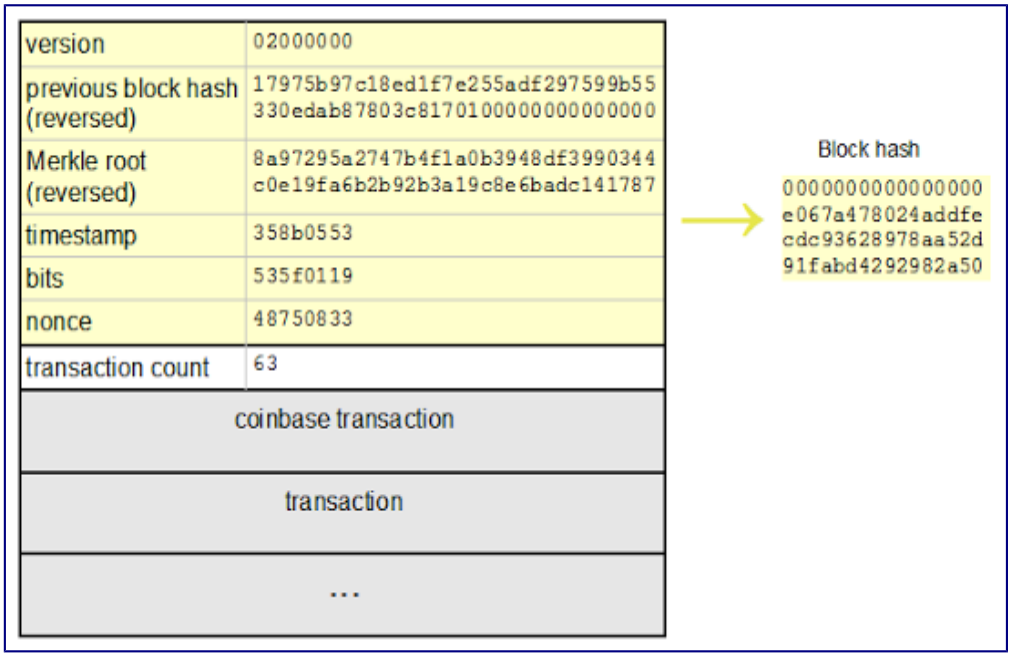
The mining process

Bitcoin mining is a key part of the security of the Bitcoin system. The idea is that Bitcoin miners group a bunch of Bitcoin transactions into a block, then repeatedly perform a cryptographic operation called hashing zillions of times until someone finds a special extremely rare hash value. At this point, the block has been mined and becomes part of the Bitcoin block chain. The hashing task itself doesn't accomplish anything useful in itself, but because finding a successful block is so difficult, it ensures that no individual has the resources to take over the Bitcoin system. For more details on mining, see my Bitcoin mining article.

A cryptographic hash function takes a block of input data and creates a smaller, unpredictable output. The hash function is designed so there's no "short cut" to get the desired output - you just have to keep hashing blocks until you find one by brute force that works. For Bitcoin, the hash function is a function called SHA-256. To provide additional security, Bitcoin applies the SHA-256 function twice, a process known as double-SHA-256.

In Bitcoin, a successful hash is one that starts with enough zeros.[1] Just as it is rare to find a phone number or license plate ending in multiple zeros, it is rare to find a hash starting with multiple zeros. But Bitcoin is exponentially harder. Currently, a successful hash must start with approximately 17 zeros, so only one out of 1.4×10^{20} hashes will be successful. In other words, finding a successful hash is harder than finding a particular grain of sand out of all the grains of sand on Earth.

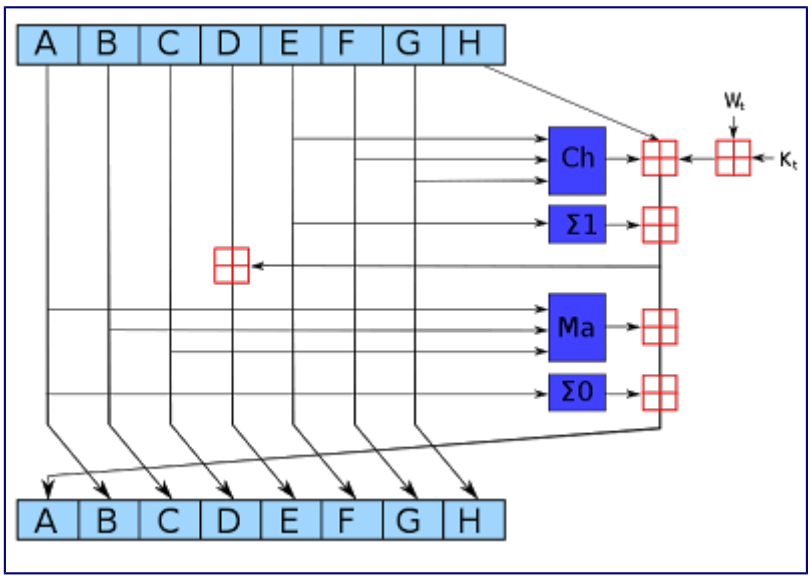
The following diagram shows a block in the Bitcoin blockchain along with its hash. The yellow bytes are hashed to generate the block hash. In this case, the resulting hash starts with enough zeros so mining was successful. However, the hash will almost always be unsuccessful. In that case, the miner changes the nonce value or other block contents and tries again.



Structure of a Bitcoin block

The SHA-256 hash algorithm used by Bitcoin

The SHA-256 hash algorithm takes input blocks of 512 bits (i.e. 64 bytes), combines the data cryptographically, and generates a 256-bit (32 byte) output. The SHA-256 algorithm consists of a relatively simple round repeated 64 times. The diagram below shows one round, which takes eight 4-byte inputs, A through H, performs a few operations, and generates new values of A through H.



One round of the SHA-256 algorithm showing the 8 input blocks A-H, the processing steps, and the new blocks. Diagram created by kockmeyer, CC BY-SA 3.0.

The blue boxes mix up the values in non-linear ways that are hard to analyze cryptographically. Since the algorithm uses several different functions, discovering an attack is harder. (If you could figure out a mathematical shortcut to generate successful hashes, you could take over Bitcoin mining.)

The *Ma* majority box looks at the bits of A, B, and C. For each position, if the majority of the bits are 0, it outputs 0. Otherwise it outputs 1. That is, for each position in A, B, and C, look at the number of 1 bits. If it is zero or one, output 0. If it is two or three, output 1.

The *Σ0* box rotates the bits of A to form three rotated versions, and then sums them together modulo 2. In other words, if the number of 1 bits is odd, the sum is 1; otherwise, it is 0. The three values in the sum are A rotated right by 2 bits, 13 bits, and 22 bits.

The *Ch* "choose" box chooses output bits based on the value of input E. If a bit of E is 1, the output bit is the corresponding bit of F. If a bit of E is 0, the output bit is the corresponding bit of G. In this way, the bits of F and G are shuffled together based on the value of E.

The next box *Σ1* rotates and sums the bits of E, similar to *Σ0* except the shifts are 6, 11, and 25 bits.

The red boxes perform 32-bit addition, generating new values for A and E. The input W_t is based on the input data, slightly processed. (This is where the input block gets fed into the algorithm.) The input K_t is a constant defined for each round.[2]

As can be seen from the diagram above, only A and E are changed in a round. The other values pass through unchanged, with the old A value becoming the new B value, the old B value becoming the new C value and so forth. Although each round of SHA-256 doesn't change the data much, after 64 rounds the input data will be completely scrambled.[3]

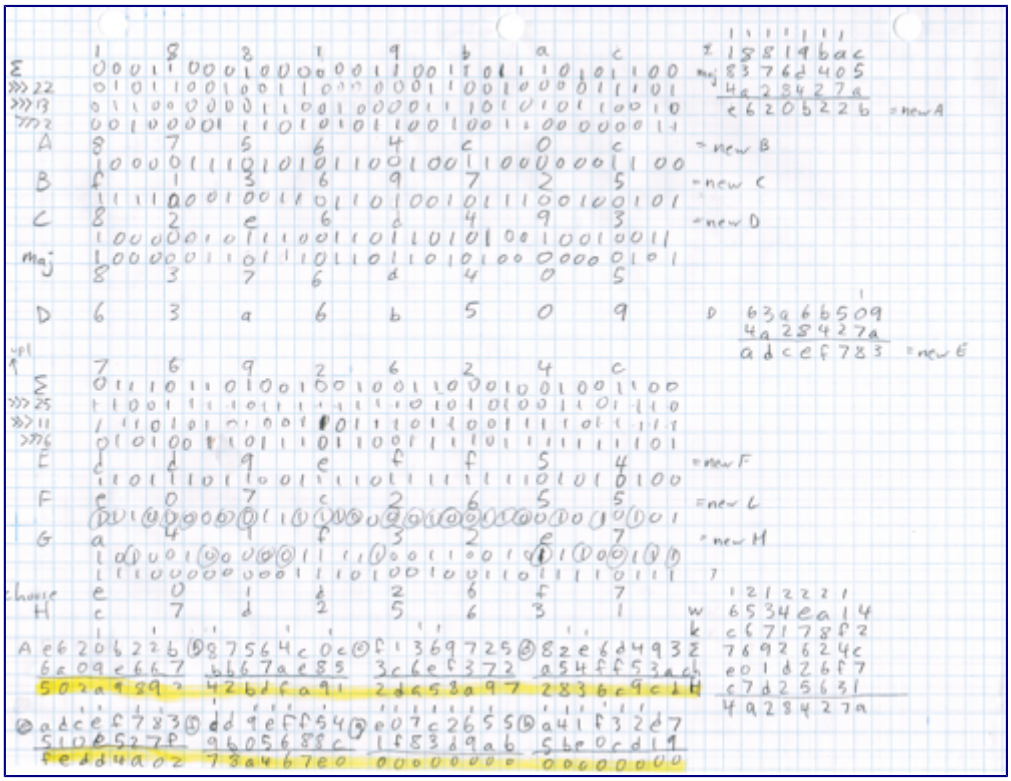
Manual mining

The video below shows how the SHA-256 hashing steps described above can be performed with pencil and paper. I perform the first round of hashing to mine a block. Completing this round took me 16 minutes, 45 seconds.

<<movie>>

To explain what's on the paper: I've written each block A through H in hex on a separate row and put the binary value below. The *maj* operation appears below C, and the shifts and *Σ0* appear above row A. Likewise, the *choose* operation appears below G, and the shifts and *Σ1* above E. In the lower right, a bunch of terms are added together, corresponding to the first three red sum boxes. In the upper right, this sum is used to generate the new A value, and in the middle right, this sum is used to generate the new E value. These steps all correspond to the diagram and discussion above.

I also manually performed another hash round, the last round to finish hashing the Bitcoin block. In the image below, the hash result is highlighted in yellow. The zeroes in this hash show that it is a successful hash. Note that the zeroes are at the end of the hash. The reason is that Bitcoin inconveniently reverses all the bytes generated by SHA-256.[4]

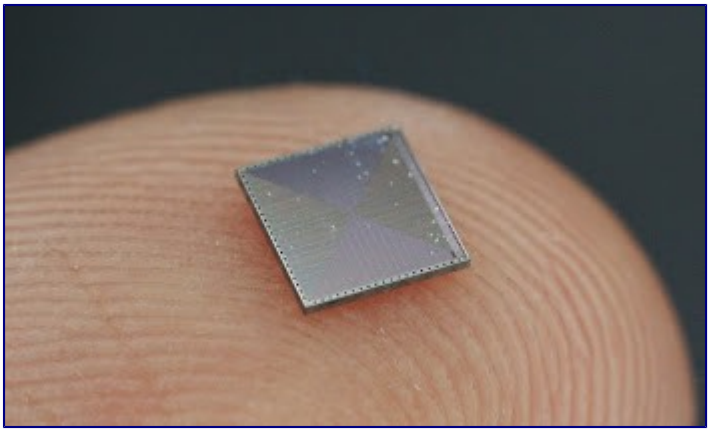


Last pencil-and-paper round of SHA-256, showing a successfully-mined Bitcoin block.

What this means for mining hardware

Each step of SHA-256 is very easy to implement in digital logic - simple Boolean operations and 32-bit addition. (If you've studied electronics, you can probably visualize the circuits already.) For this reason, custom ASIC chips can implement the SHA-256 algorithm very

efficiently in hardware, putting hundreds of rounds on a chip in parallel. The image below shows a mining chip that runs at 2-3 billion hashes/second; Zeptobars has more photos.



The silicon die inside a Bitfury ASIC chip. This chip mines Bitcoin at 2-3 Ghash/second. Image from Zeptobars. (CC BY 3.0)

In contrast, Litecoin, Dogecoin, and similar altcoins use the scrypt hash algorithm, which is intentionally designed to be difficult to implement in hardware. It stores 1024 different hash values into memory, and then combines them in unpredictable ways to get the final result. As a result, much more circuitry and memory is required for scrypt than for SHA-256 hashes. You can see the impact by looking at mining hardware, which is thousands of times slower for scrypt (Litecoin, etc) than for SHA-256 (Bitcoin).

Conclusion

The SHA-256 algorithm is surprisingly simple, easy enough to do by hand. (The elliptic curve algorithm for signing Bitcoin transactions would be very painful to do by hand since it has lots of multiplication of 32-byte integers.) Doing one round of SHA-256 by hand took me 16 minutes, 45 seconds. At this rate, hashing a full Bitcoin block (128 rounds)[3] would take 1.49 days, for a hash rate of 0.67 hashes per day (although I would probably get faster with practice). In comparison, current Bitcoin mining hardware does several terahashes per second, about a quintillion times faster than my manual hashing. Needless to say, manual Bitcoin mining is not at all practical.[5]

A Reddit reader asked about my energy consumption. There's not much physical exertion, so assuming a resting metabolic rate of 1500kcal/day, manual hashing works out to almost 10 megajoules/hash. A typical energy consumption for mining hardware is 1000 megahashes/joule. So I'm less energy efficient by a factor of 10^16, or 10 quadrillion. The next question is the energy cost. A cheap source of food energy is donuts at $0.23 for 200 kcalories. Electricity here is $0.15/kilowatt-hour, which is cheaper by a factor of 6.7 - closer than I expected. Thus my energy cost per hash is about 67 quadrillion times that of mining hardware. It's clear I'm not going to make my fortune off manual mining, and I haven't even included the cost of all the paper and pencils I'll need.

2017 edit: My Bitcoin mining on paper system is part of the book The Objects That Power the Global Economy, so take a look.

Follow me on Twitter to find out about my latest blog posts.

Notes

[1] It's not exactly the number of zeros at the start of the hash that matters. To be precise, the hash must be less than a particular value that depends on the current Bitcoin difficulty level.

[2] The source of the constants used in SHA-256 is interesting. The NSA designed the SHA-256 algorithm and picked the values for these constants, so how do you know they didn't pick special values that let them break the hash? To avoid suspicion, the initial hash values come from the square roots of the first 8 primes, and the K_t values come from the cube roots of the first 64 primes. Since these constants come from a simple formula, you can trust that the NSA didn't do anything shady (at least with the constants).

[3] Unfortunately the SHA-256 hash works on a block of 512 bits, but the Bitcoin block header is more than 512 bits. Thus, a second set of 64 SHA-256 hash rounds is required on the second half of the Bitcoin block. Next, Bitcoin uses *double-SHA-256*, so a second application of SHA-256 (64 rounds) is done to the result. Adding this up, hashing an arbitrary Bitcoin block takes 192 rounds in total. However there is a shortcut. Mining involves hashing the same block over and over, just changing the *nonce* which appears in the second half of the block. Thus, mining can reuse the result of hashing the first 512 bits, and hashing a Bitcoin block typically only requires 128 rounds.

[4] Obviously I didn't just have incredible good fortune to end up with a successful hash. I started the hashing process with a block that had already been successfully mined. In particular I used the one displayed earlier in this article, #286819.

[5] Another problem with manual mining is new blocks are mined about every 10 minutes, so even if I did succeed in mining a block, it would be totally obsolete (orphaned) by the time I finished.