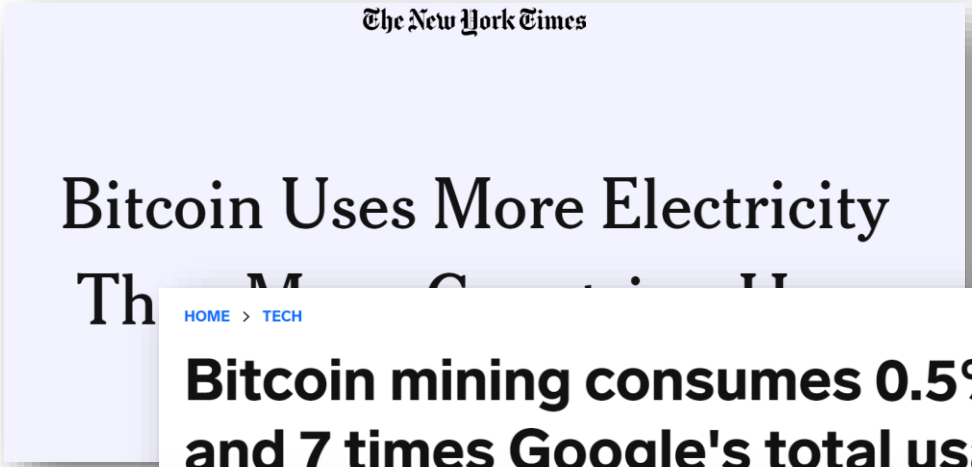
Crypto's Capex Carbon Footprint

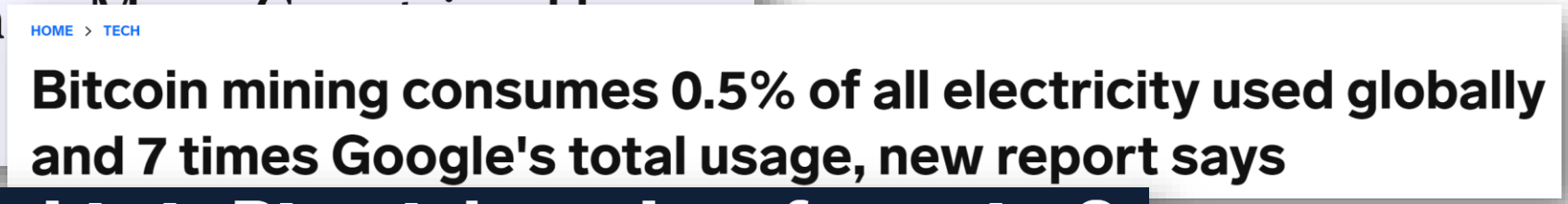
Jaylen's ES91r Project
(w/ the help of Udit)

Agenda

1. Motivation
2. Bitcoin Mining Background
3. Project Goals
4. Bitcoin Mining Workload
5. ASIC Design
6. Estimating Embodied Carbon
7. Future Directions

Motivation


The New York Times
Bitcoin Uses More Electricity

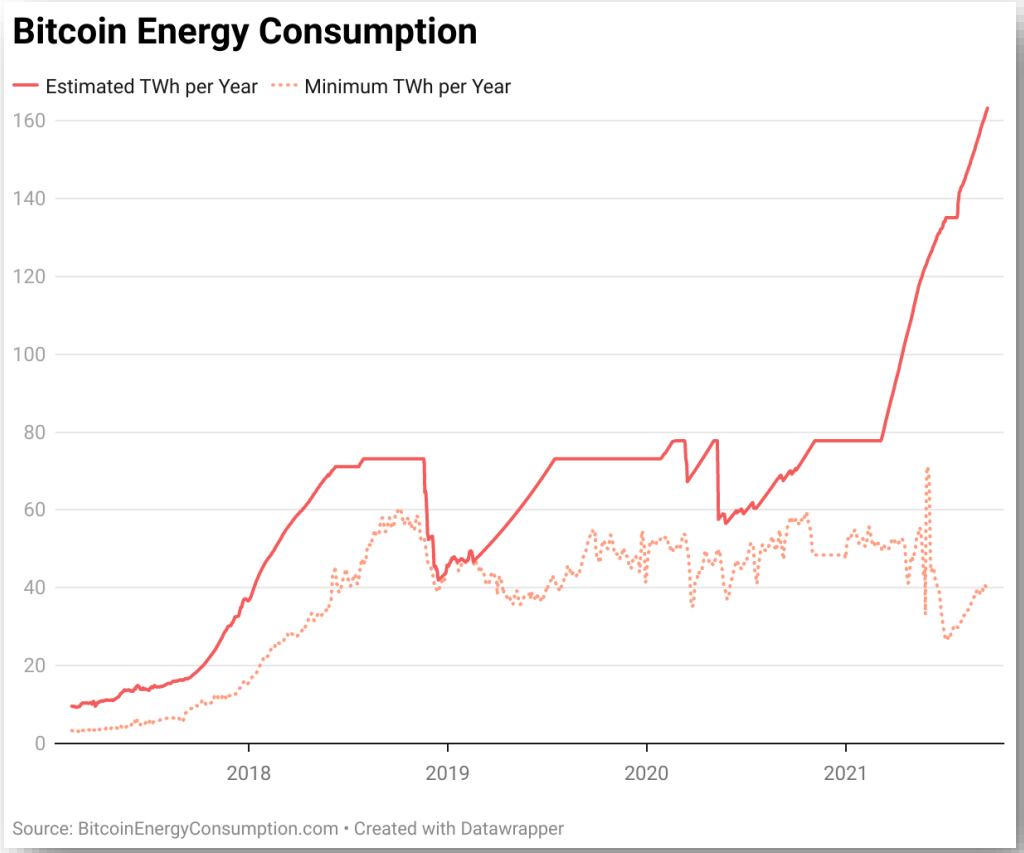

HOME > TECH
Bitcoin mining consumes 0.5% of all electricity used globally and 7 times Google's total usage, new report says


How big is Bitcoin's carbon footprint?
Elon Musk said Wednesday that Tesla would no longer accept bitcoin, citing the cryptocurrency's energy demands.


Climate and Environment
Why the bitcoin craze is using up so much energy

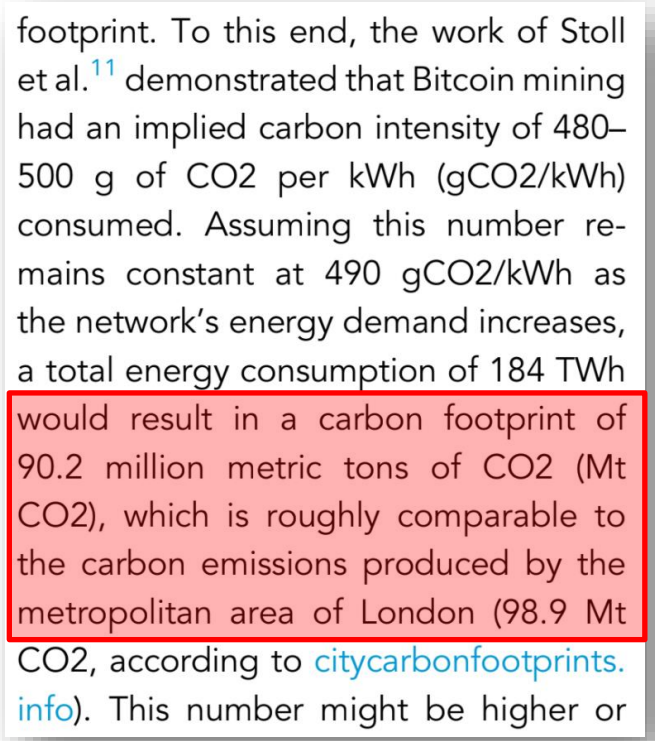
Current State

- Energy devoted to Bitcoin has been on the rise and has surpassed major companies and industries.
- Consumes many times more than Google (~12TWh) or Facebook (~5TWh).



Current State

- Energy devoted to Bitcoin has been on the rise and has surpassed major companies and industries.
- Consumes many times more than Google (~12TWh) or Facebook (~5TWh).
- Carbon emissions have been estimated to be on par with medium-sized countries.

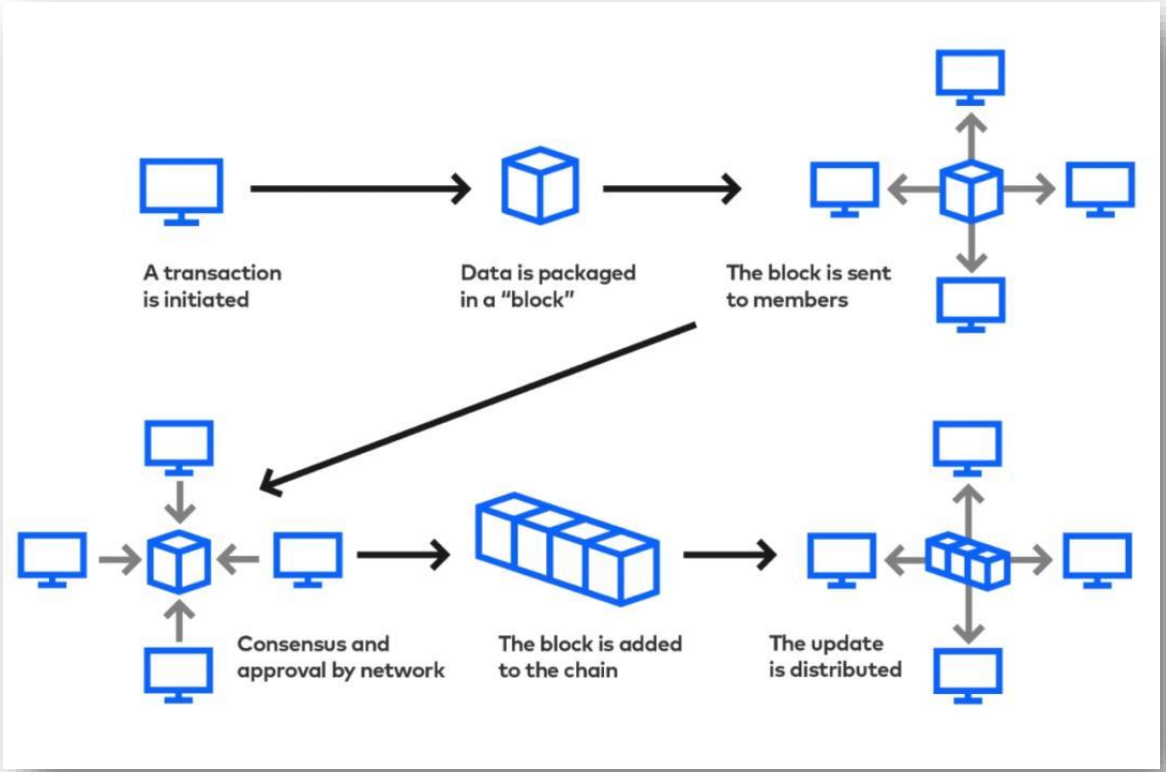


footprint. To this end, the work of Stoll et al.[11] demonstrated that Bitcoin mining had an implied carbon intensity of 480–500 g of CO2 per kWh (gCO2/kWh) consumed. Assuming this number remains constant at 490 gCO2/kWh as the network's energy demand increases, a total energy consumption of 184 TWh would result in a carbon footprint of 90.2 million metric tons of CO2 (Mt CO2), which is roughly comparable to the carbon emissions produced by the metropolitan area of London (98.9 Mt CO2, according to citycarbonfootprints.info). This number might be higher or

de Vries (2021)

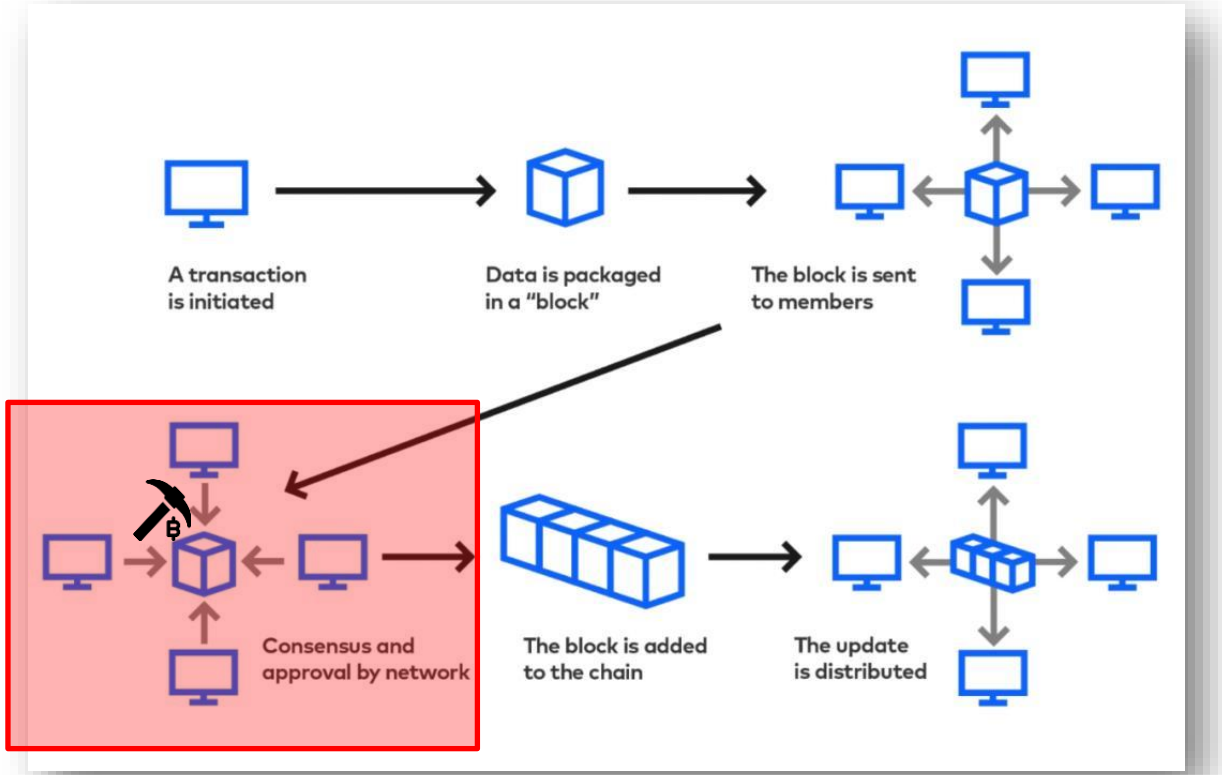
Mining Background

The Blockchain



Source

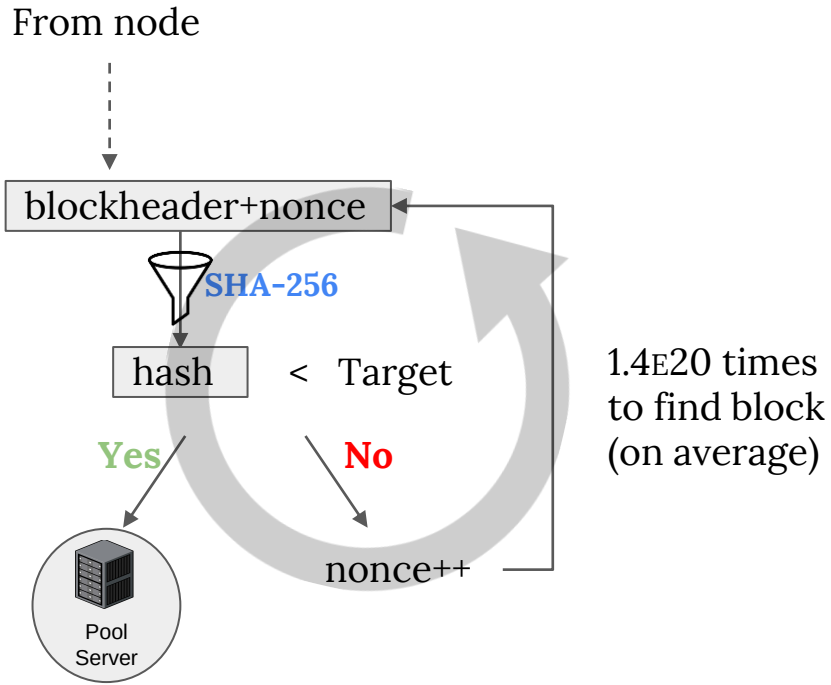
The Blockchain



Source

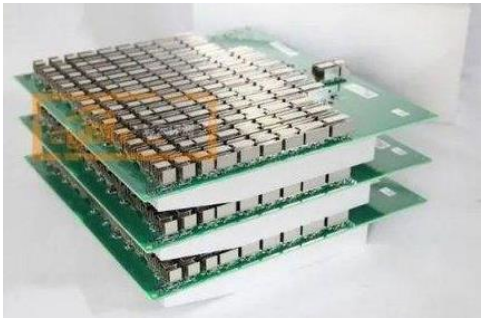
Bitcoin Mining Proof-of-Work

- Miners run a hash function (SHA-256), hashing the “block header” with a different nonce until the output hash is under some threshold.
- The first miner to do this adds to the block and receives compensation in the form of fees.



Mining Hardware

- Almost all hashing is done by specialized mining hardware with multiple SHA-256 ASIC accelerators.
- These machines can run at >2500 Watts → most of the carbon concerns.



Boards with ~100 custom ASICs producing ~100 TeraHashes/s

Embodied Carbon

- Udit's previous work revealed that a large portion of carbon emissions from computing can come from capex emissions.
- Question: what are the implications of this for Bitcoin?

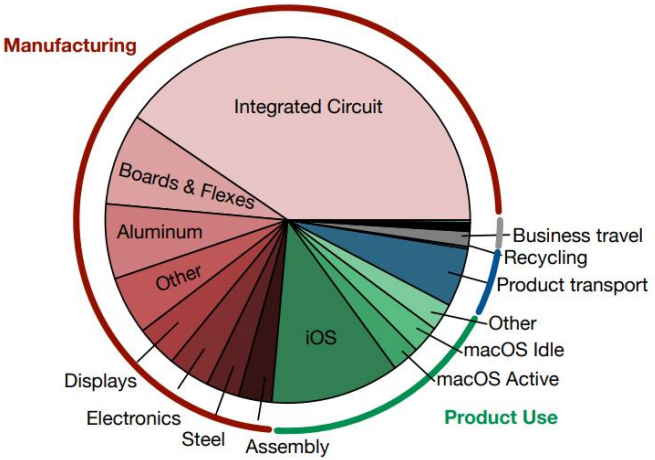


Fig. 5. Apple's carbon-emission breakdown. In aggregate, the hardware life cycle (i.e., manufacturing, transport, use, and recycling) comprises over 98% of Apple's total emissions. Manufacturing accounts for 74% of total emissions, and hardware use accounts for 19%. Carbon output from manufacturing integrated circuits (i.e., SoCs, DRAM, and NAND flash memory) is higher than that from hardware use.

Gupta *et al.* (2021)

Project Goals

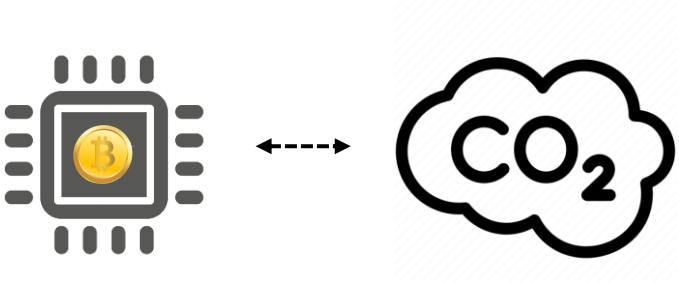
Current Research Landscape

- Previous research has looked at:
  1. Hardware (mining)
  2. Carbon footprint

Current Research Landscape

- Previous research has looked at:
  1. Hardware (mining)
  2. Carbon footprint

  } Need to be considered together
- Hardware research doesn't consider embodied carbon in design
  - What kind of design tradeoffs are there?
- Carbon footprint estimations never include capex costs
  - Beside operational energy costs, what other factors affect the carbon footprint of Bitcoin?



Project Goals

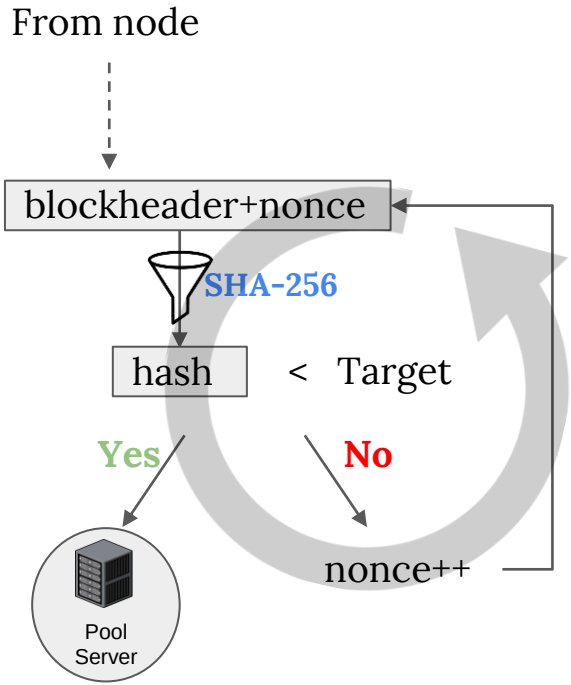
Create a more holistic look at Bitcoin's carbon footprint with the key contributions:

1. **Capex cost** of producing mining hardware
2. Combine this along with previous research to get a more complete **picture of Bitcoin's carbon footprint**
3. **Tradeoffs** that can be made to improve carbon efficiency
4. (Ideally) what are the implications for **crypto as a whole**

Bitcoin Mining Workload

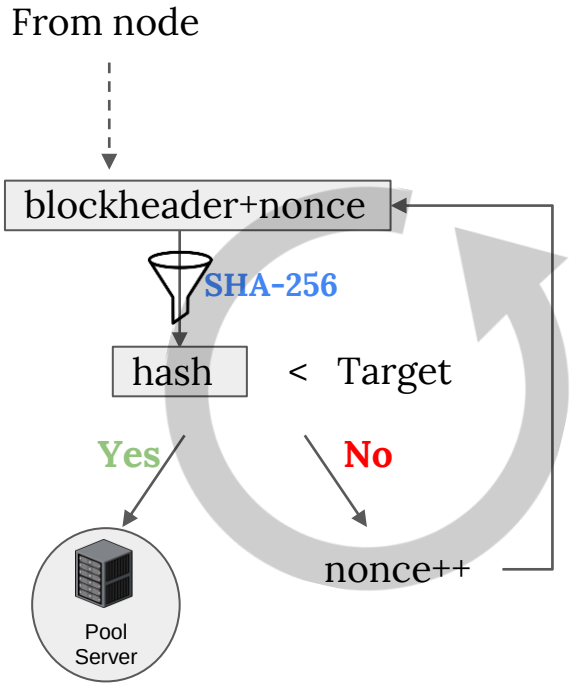
Mining Goal

Job: run SHA-256 hashes on block headers (ie. perform as many hashes per second as possible with as little energy per hash as possible)



Mining Goal

Job: run **SHA-256 hashes** on block headers (ie. perform as many hashes per second as possible with as little energy per hash as possible)



SHA-256 Algorithm

1. Takes in a 512-bit message
2. Then the Message Expander (ME) expands the message into 64 chunks of 32-bit data: W[0...63].
3. Then the Message Compressor (MC) compresses the array to 8 chunks of 32-bit data for the final 256-bit hash.

Algorithm 1 Message Expander (ME)

- For j from 0 to 15 {
  $W_j = M_j$ }
- For j from 16 to 63 {
  $W_j = \sigma_1(W_{j-2}) + W_{j-7} + \sigma_0(W_{j-15}) + W_{j-16}$ }

Algorithm 2 Message Compressor (MC)

(1) Initialization:
$a = H_1; b = H_2; c = H_3; d = H_4; e = H_5; f = H_6;$
$g = H_7; h = H_8$

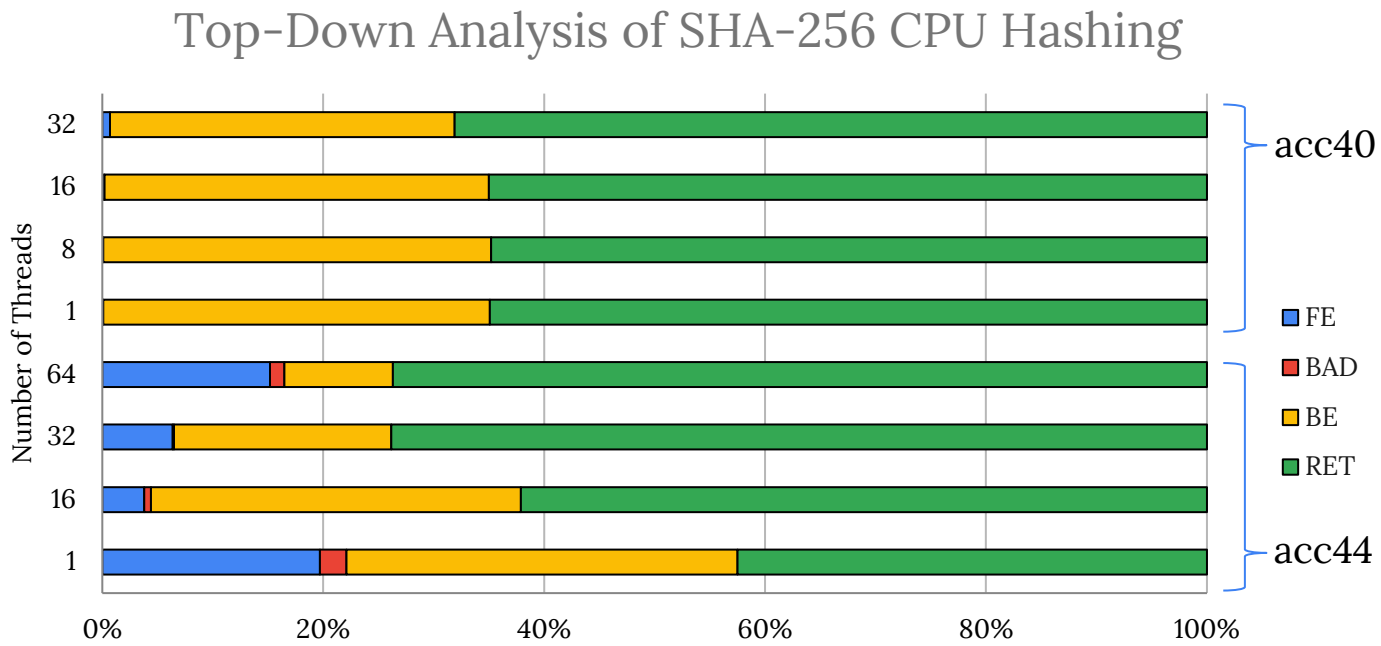
(2) Loop:
For j from 0 to 63 {

- $T_1 = h + \Sigma_1(e) + Ch(e, f, g) + K_j + W_j$
- $T_2 = \Sigma_0(a) + Maj(a, b, c)$
- h = g; g = f; f = e; e = d + T_1; d = c; c = b; b = a; a = $T_1 + T_2$ }

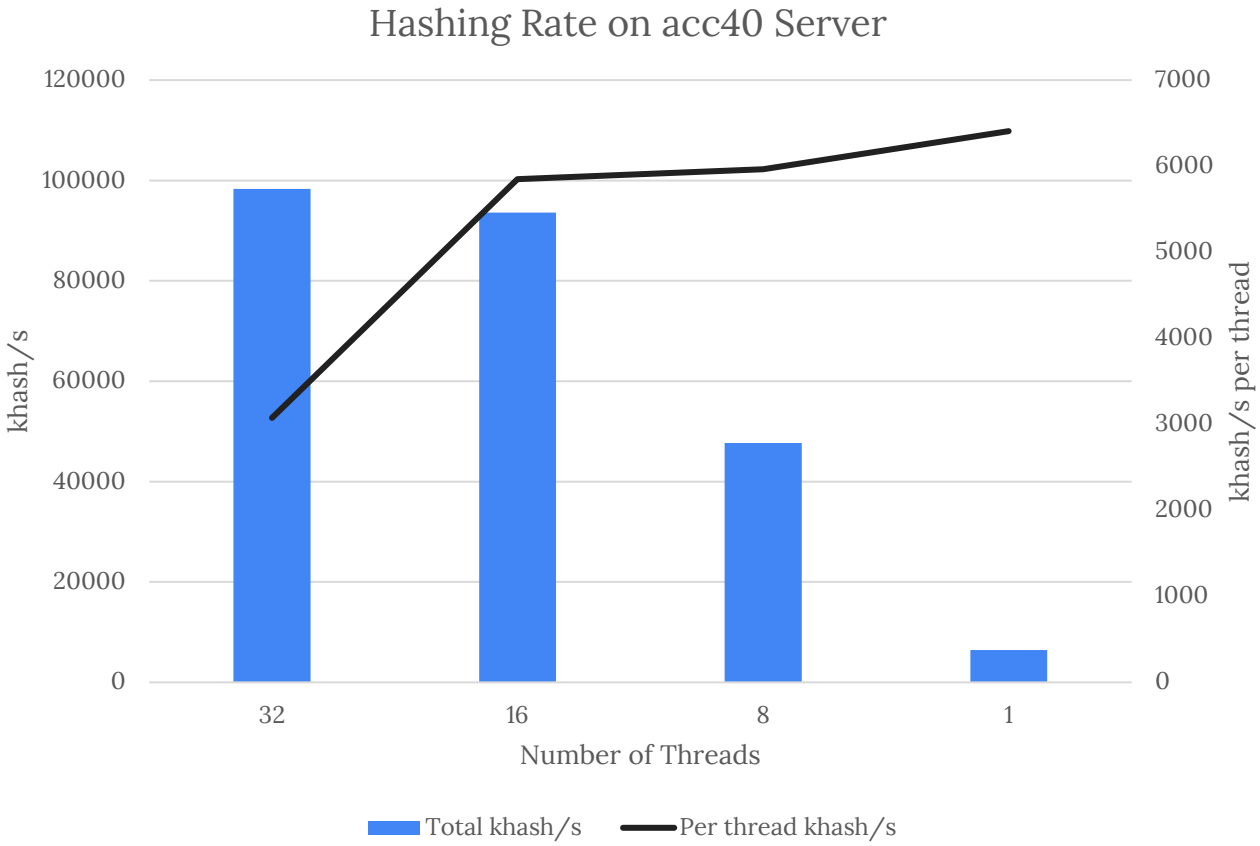
(3) Add:
$HO_1 = a + H_1; HO_2 = b + H_2; HO_3 = c + H_3;$
$HO_4 = d + H_4; HO_5 = e + H_5; HO_6 = f + H_6;$
$HO_7 = g + H_7; HO_8 = h + H_8;$

SHA-256 CPU Profiling

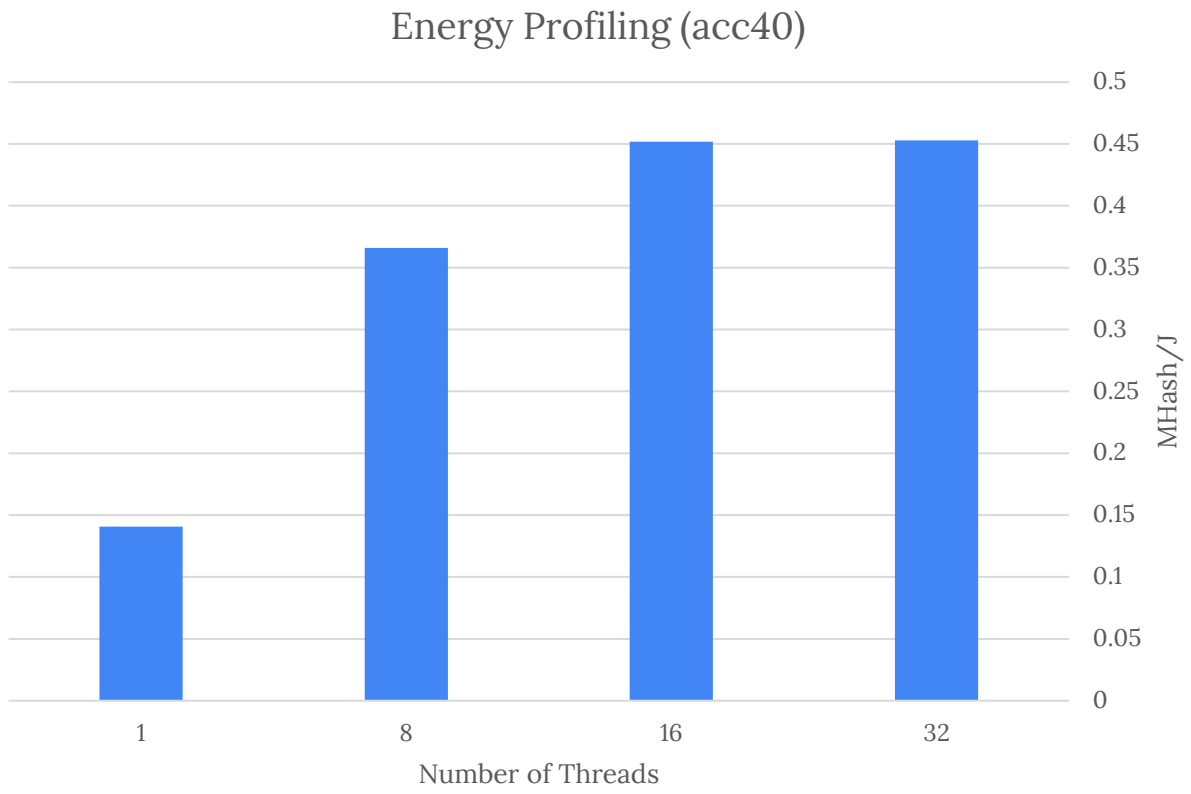
- Performed Top-Down analysis of *cpuminer* (open-source CPU mining)
- Conclusions:
  - **Mostly backend bound → core-bound**, so not enough compute units
  - **Lots of retiring → lightweight operations**, so not a lot of parallelism



Hashing Rate



Energy


Energy Profiling (acc40)
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
MHash/J
1
8
16
32
Number of Threads

Antminer S19 vs. GPU vs. CPU

Hardware	Price	GH/s	GH/J
Antminer S19	2979	95,000	29
RTX 3090	1500	4.85	0.011
Xeon Gold 6242	2529	0.375	0.00125

Source

Source

Source

ASIC Design

Why Design an ASIC?

- Since commercial mining ASIC designs are not accessible, the goal was to use HLS to get something comparable
- Used previous research in SHA-256 accelerators to implement the most common optimizations

Specialized Double SHA-256 Accelerator

- For Bitcoin, you need to hash the block header twice
- The first 512-bit chunk does not change often
- Second part changes frequently (with every nonce)

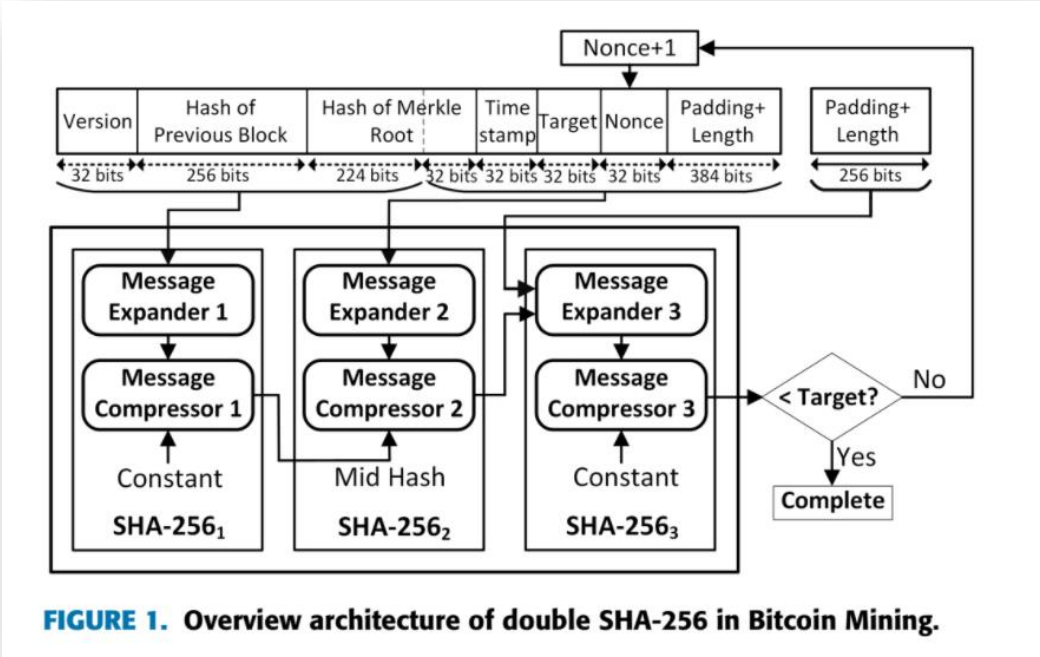


FIGURE 1. **Overview architecture of double SHA-256 in Bitcoin Mining.**

Common Strategies

- Fully unroll the 64-iteration loops and pipeline them → produce one hash per cycle.
- Leave the loops for the first 512-bit chunk rolled.
- Have multiple “engines” or “cores” that produce hashes in parallel (in different nonce ranges).

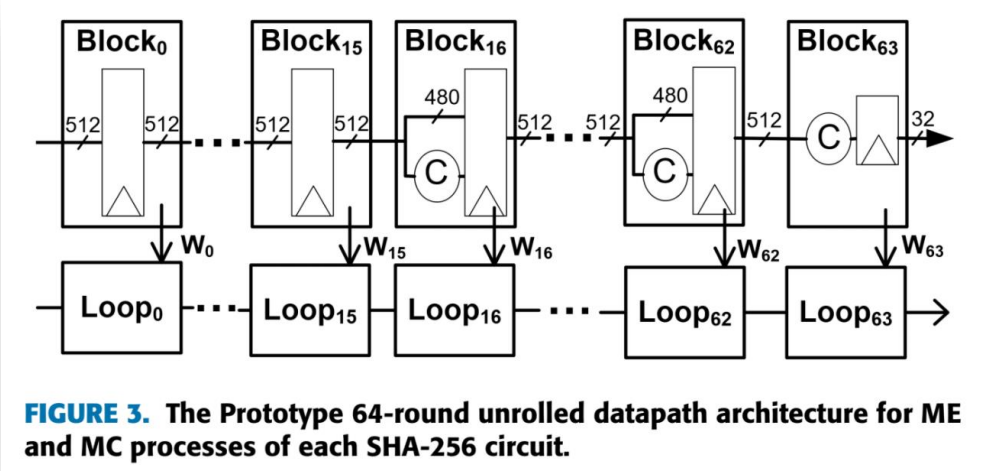


FIGURE 3. The Prototype 64-round unrolled datapath architecture for ME and MC processes of each SHA-256 circuit.

Common Strategies

- Fully unroll the 64-iteration loops and pipeline them → produce one hash per cycle
- Have multiple “engines” or “cores” that produce hashes in parallel (in different nonce ranges).

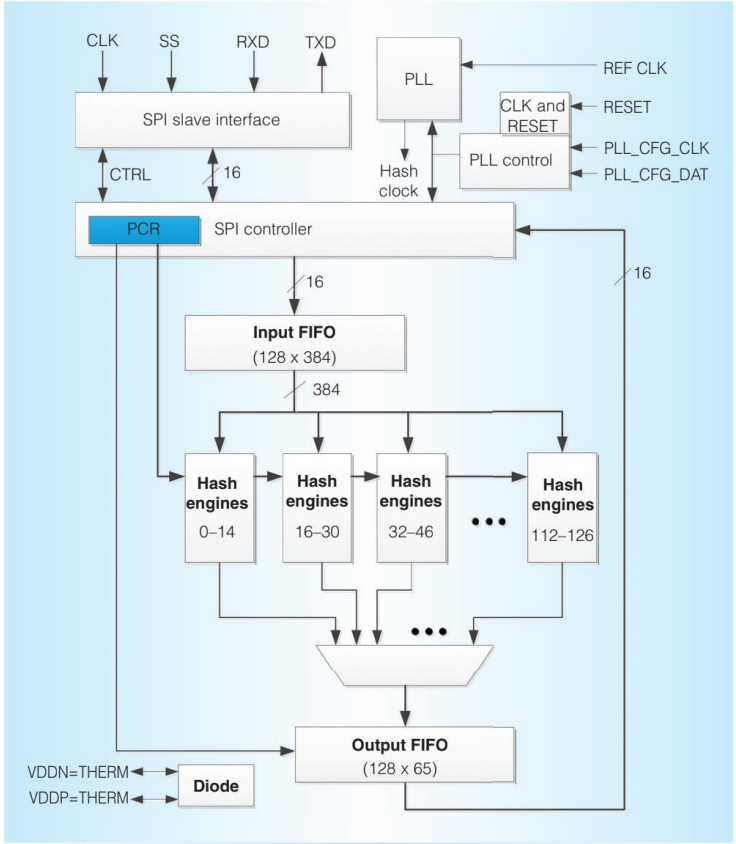


Figure 2. Block diagram of the Goldstrike 1 architecture. Each of the 120 hash engines is working independently on a separate problem.

Common Strategies

- Fully unroll the 64-iteration loops and pipeline them → produce one hash per cycle
- Have multiple “engines” or “cores” that produce hashes in parallel (in different nonce ranges).

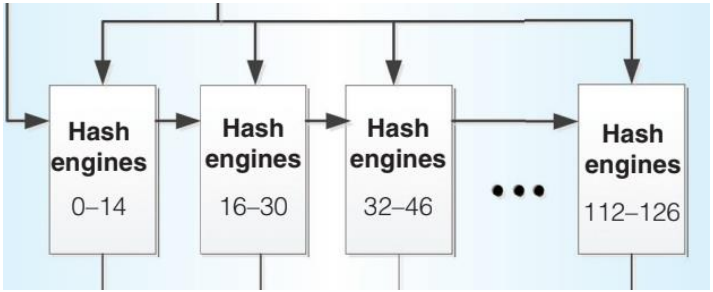


Figure 2. Block diagram of the Goldstrike 1 architecture. Each of the 120 hash engines is working independently on a separate problem.

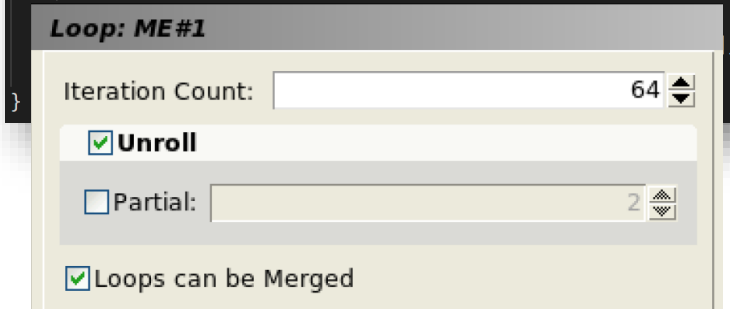
Using Catapult HLS to Generate RTL

- Use Catapult HLS to try to get a reasonable area and performance look at the hashing accelerator to get carbon-footprint.
- Implemented common unrolling/pipelining techniques.

Algorithm 1 Message Expander (ME)

- For j from 0 to 15 {
  $W_j = M_j$ }
- For j from 16 to 63 {
  $W_j = \sigma_1(W_{j-2}) + W_{j-7} + \sigma_0(W_{j-15}) + W_{j-16}$ }

```
/* Message expander */
ME: for (u32_t i = 0; i < 64; i++) {
    if (i < 16) {
        w[i] = data.slc<32>(480 - i * 32);
}
```



Using Catapult HLS to Generate RTL

- Use Catapult HLS to try to get a reasonable area and performance look at the hashing accelerator to get carbon-footprint.

Algorithm 2 Message Compressor (MC)

(1) Initialization:
$a = H_1; b = H_2; c = H_3; d = H_4; e = H_5; f = H_6;$
$g = H_7; h = H_8$

(2) Loop:
For j from 0 to 63 {

- $T_1 = h + \Sigma_1(e) + Ch(e, f, g) + K_j + W_j$
- $T_2 = \Sigma_0(a) + Maj(a, b, c)$
- h = g; g = f; f = e; e = d + T_1; d = c; c = b; b = a; a = $T_1 + T_2$ }

(3) Add:
$HO_1 = a + H_1; HO_2 = b + H_2; HO_3 = c + H_3;$
$HO_4 = d + H_4; HO_5 = e + H_5; HO_6 = f + H_6;$
$HO_7 = g + H_7; HO_8 = h + H_8;$

```
/* Message compressor */
MC: for (u32_t i = 0; i < 64; i++) {
    /* Temporal Summation */
    u32_t t1 = h + EP1(e) + CH(e, f, g) + k[i] + w[i];
    u32_t t2 = EP0(a) + MAJ(a, b, c);
```

Loop: MC#1

Iteration Count: 64

☑ Unroll

☐ Partial: 2

☑ Loops can be Merged

```
hash[4] += e, hash[5] += f, hash[6] += g, hash[7] += h;
```

Specialized Double SHA-256 Accelerator

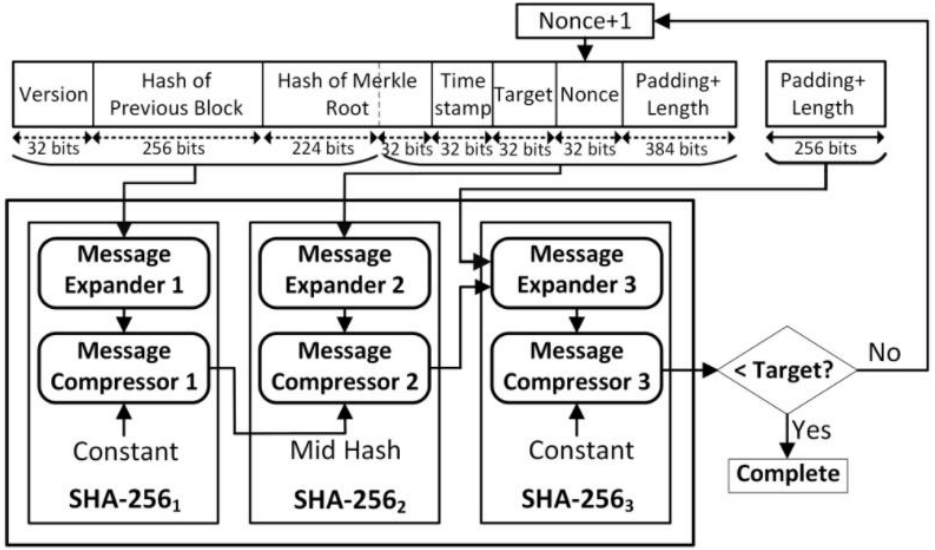
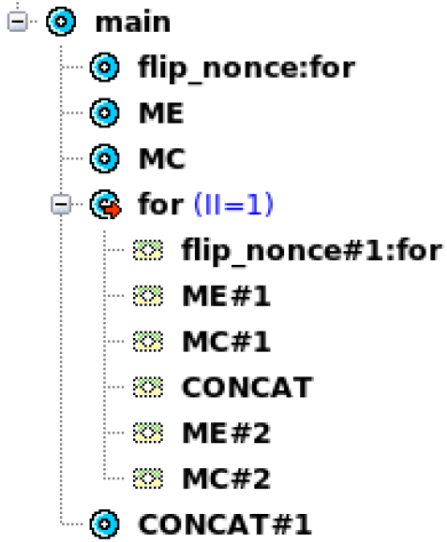


FIGURE 1. **Overview architecture of double SHA-256 in Bitcoin Mining.**



Specialized Double SHA-256 Accelerator

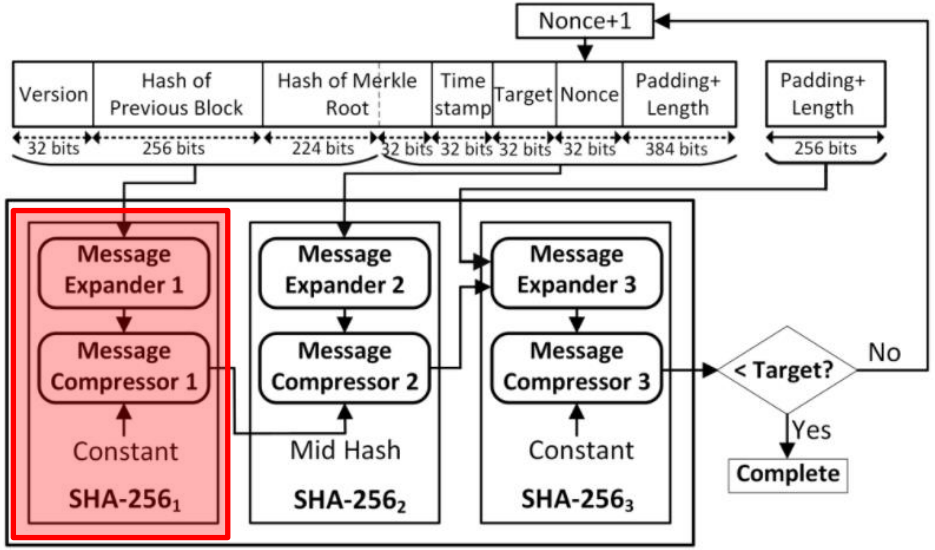
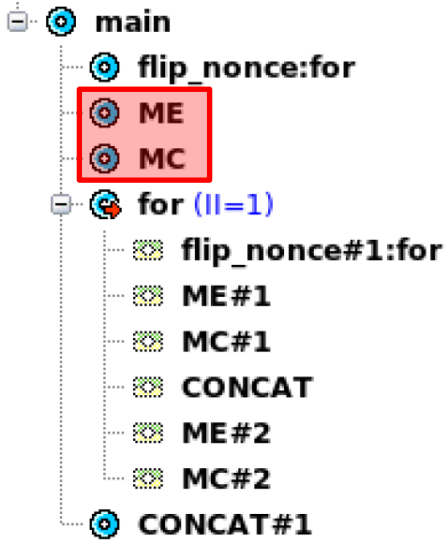


FIGURE 1. **Overview architecture of double SHA-256 in Bitcoin Mining.**



Specialized Double SHA-256 Accelerator

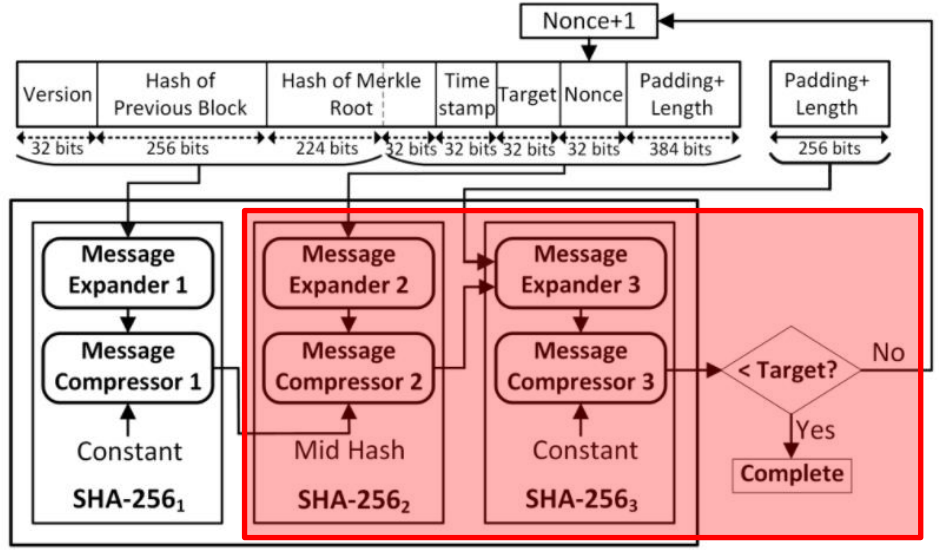
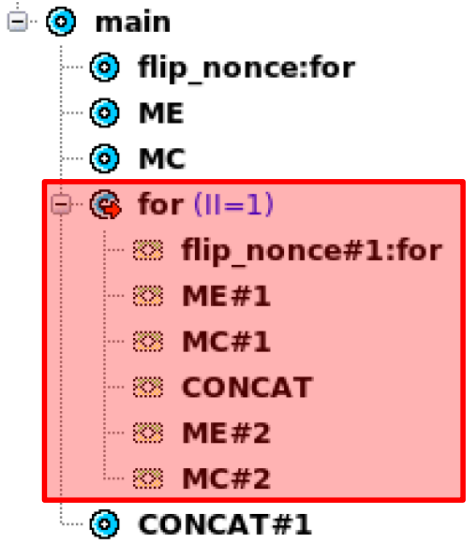


FIGURE 1. **Overview architecture of double SHA-256 in Bitcoin Mining.**



Specialized Double SHA-256 Accelerator

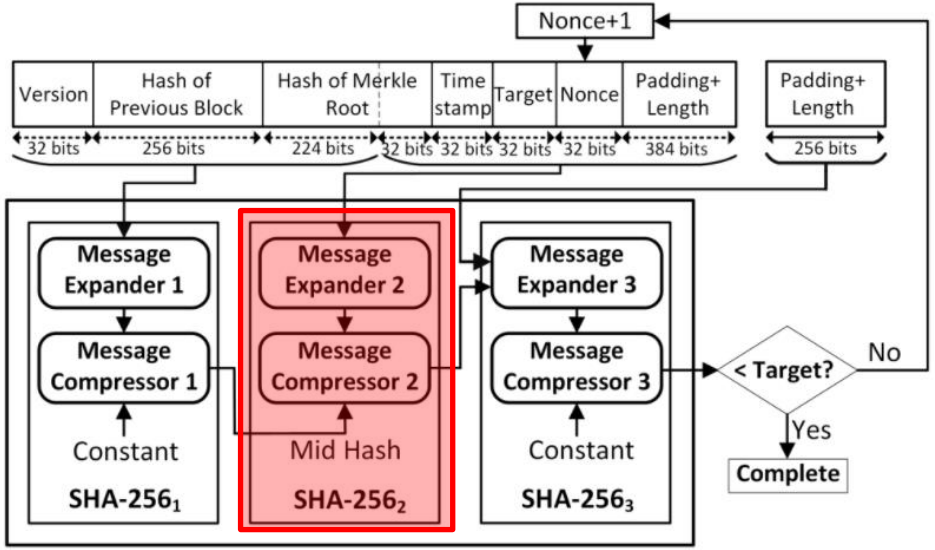
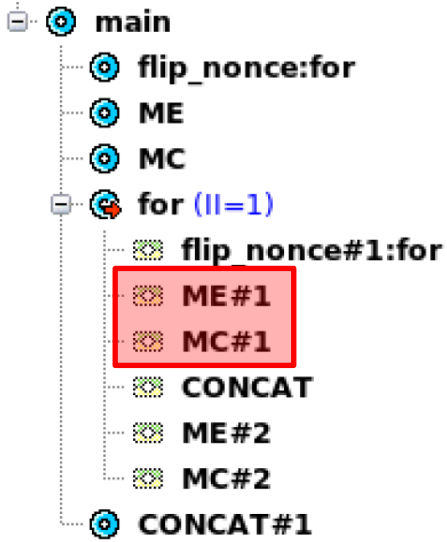


FIGURE 1. **Overview architecture of double SHA-256 in Bitcoin Mining.**



Specialized Double SHA-256 Accelerator

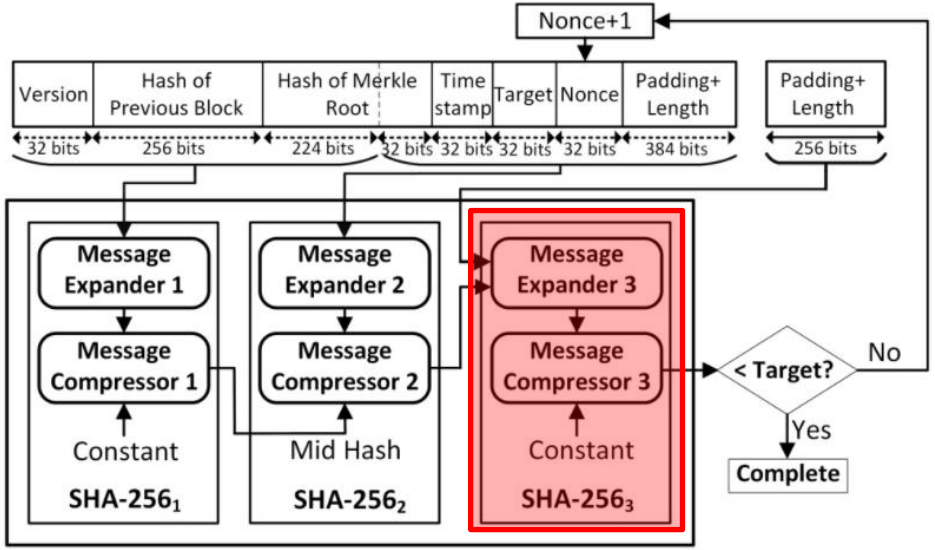
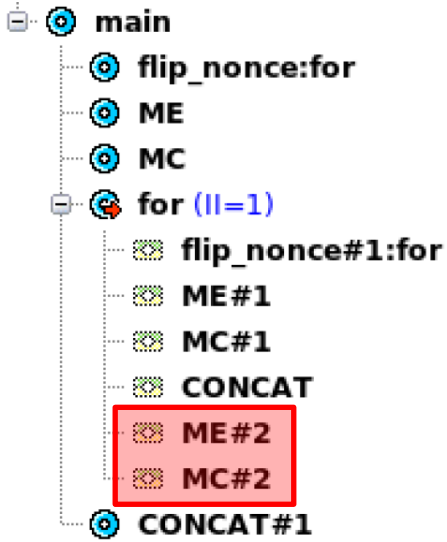
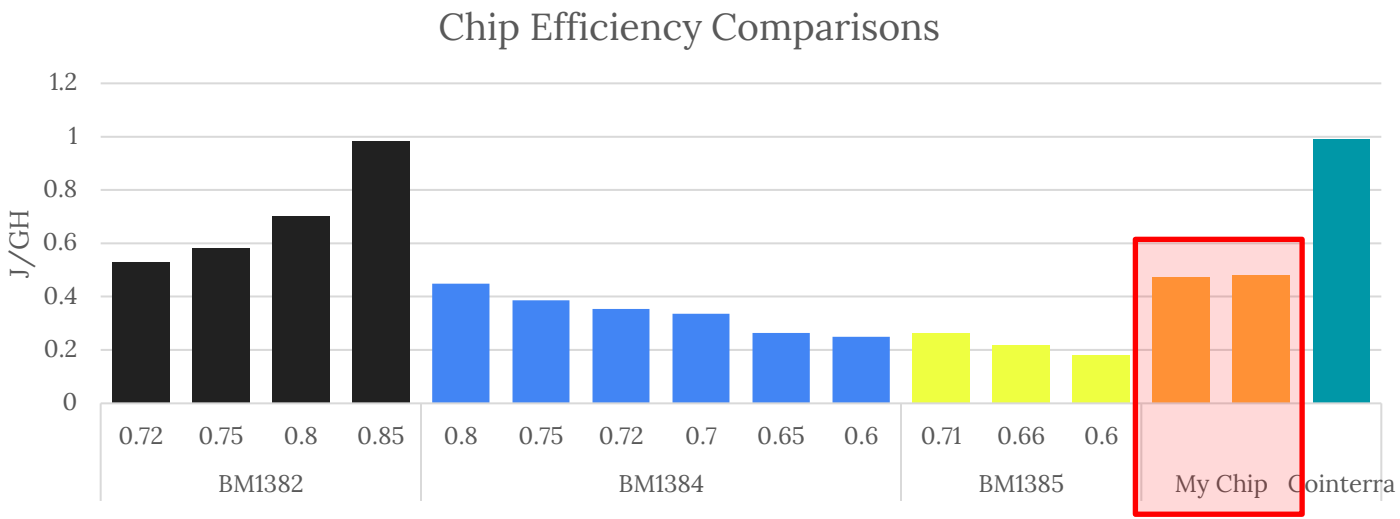


FIGURE 1. **Overview architecture of double SHA-256 in Bitcoin Mining.**



Performance Comparison

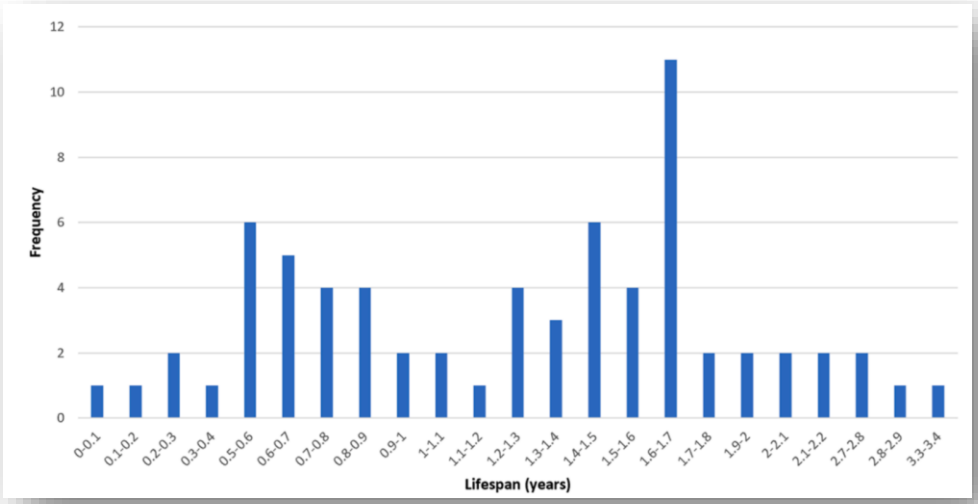
- Used node scaling numbers to scale down the area and power of my design (45nm) to do a direct comparison.
- Iterated on the design until it was in the same ballpark as previous accelerators.



Estimating Embodied Carbon

How to Get Carbon Numbers? (opex)

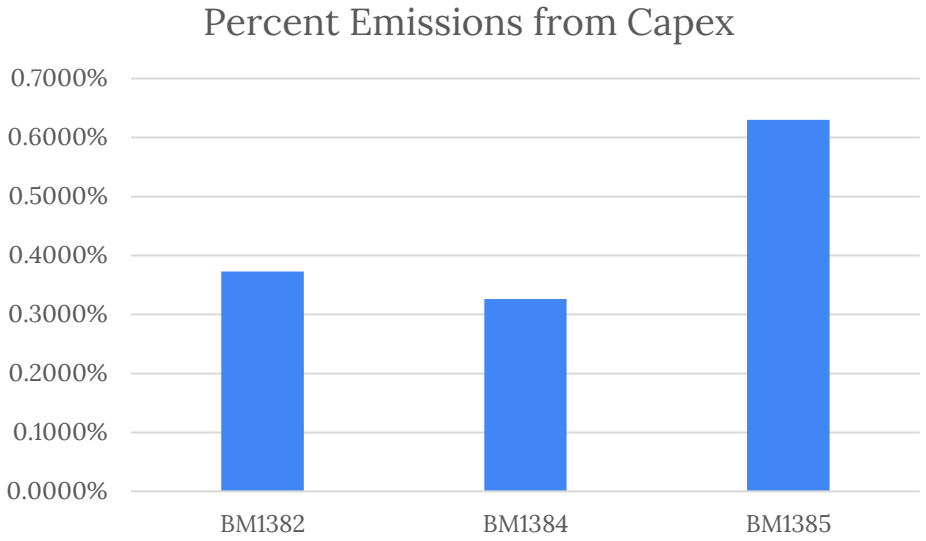
- For opex, used power combined with previous data.
  - Previous research found carbon intensity to be around 490 g CO2/kWh for average bitcoin miner.
  - Previous research found conservative estimate of lifetime of device to be 1.3 years
  - Used reported power consumption of commercial mining ASICs.



de Vries and Stoll (2021)

How to Get Carbon Numbers? (capex)

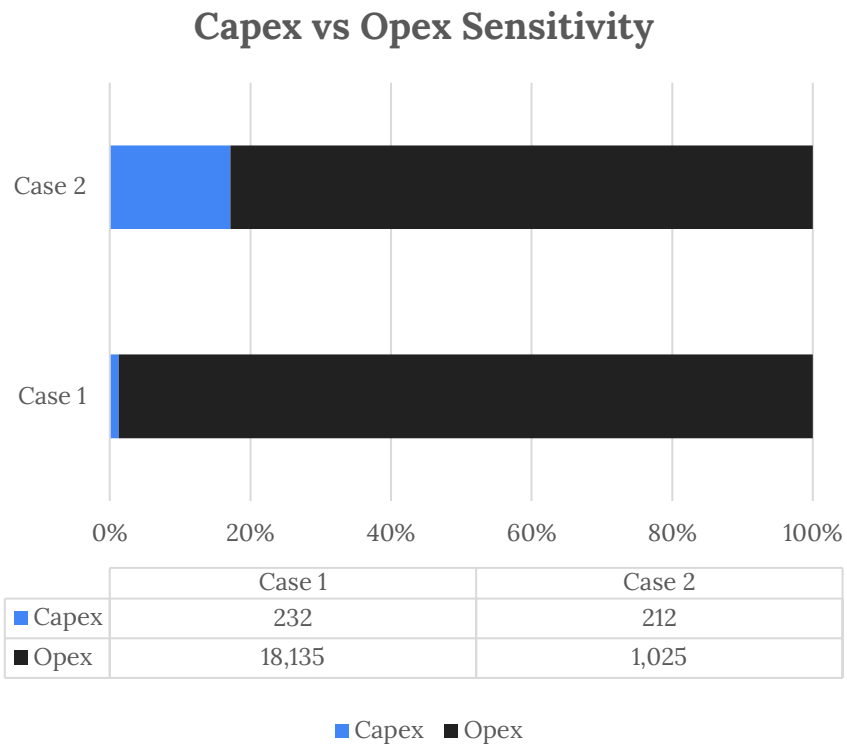
- For capex, numbers from *Bardon et al.* (2020), with my area estimates and node size as inputs to get g CO2/cm^2.
- Normalized all chips to have the same hash rate (scaling up area and power linearly) and then calculated opex and capex numbers.



Extremely opex dominated

What if we consider the whole machine?

- For now, included rough carbon numbers for chassis, memory, PCB, PSU, and fans:



	Case 1	Case 2
Capex	232	212
Opex	18,135	1,025

	Case 1	Case 2
Carbon intensity (g CO2/kWh)	490	50
Utilization (%)	100	90
Lifetime (yrs)	1.3	1
TOTAL kg CO2	18,367	1,267

Takeaways

- Opex dominates Bitcoin mining ASICs because:
  - The power density is very high (almost all compute and full pipeline utilization)
  - Very few other ICs for the miner (very little external memory or other compute requirements beyond hashing)
  - Utilization at almost 100% (mining is a constant workload)

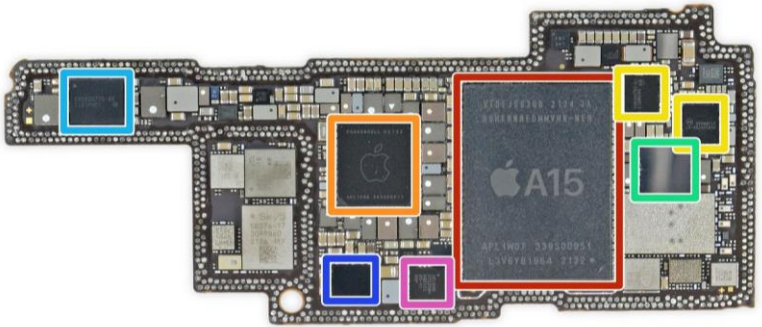
Takeaways

- Opex dominates Bitcoin mining ASICs because:
  - The power density is very high (almost all compute and full pipeline utilization)
  - Very few other ICs for the miner (very little external memory or other compute requirements beyond hashing)
  - Utilization at almost 100% (mining is a constant workload)

	Area (mm^2)	Power (W)	Power Density (W/mm^2)
Apple A15	107	5	**0.047**
Antminer BM1385	15	10	**0.67**

Takeaways

- Opex dominates Bitcoin mining ASICs because:
  - The power density is very high (almost all compute and full pipeline utilization)
  - Very few other ICs for the miner (very little external memory or other compute requirements beyond hashing)
  - Utilization at almost 100% (mining is a constant workload)



Future Directions

Other Proof-of-Work Algorithms

- Ethash: requires very large amounts of memory (random accesses to a 4 GB DAG)
  - Manufacturing costs of memory can make up a substantial portion of capex costs
- Supposedly is “ASIC-resistant”
  - GPUs are much more competitive due to the high memory bandwidth

ethereum

Thank you for listening!
Questions? Feedback?

Also – big thanks to Udit for helping at each step of the way :)

Please reach out with any questions (or to chat about whatever!)
Can find me over Slack or email me at jaylenwang@college.harvard.edu