TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

DAREN TUZI

CRYPTONIGHT GPU MINING EFFICIENCY

Master of Science Thesis

Examiner: Assistant Professor Billy Brumley
Examiner and topic approved on 29 March 2017

ABSTRACT

DAREN TUZI: Monero Gpu Mining Efficiency
Tampere University of technology
Master of Science Thesis, 53 pages
June 2018
Master's Degree Programme in Information Technology
Major: Pervasive Systems
Examiner: Assistant Professor Billy Brumley
Keywords: monero, mining, bitcoin, cryptocurrency

The purpose of this thesis is to study the efficiency of using graphical processing units in Cryptonight, the proof-of-work system used to mine Monero. By understanding the dependence of Cryptonight in memory, we theorize that by improving read and write delays we can improve mining results.

In this thesis, there is a major focus on the technology behind Bitcoin and Monero since at the time of writing stand to be the most respectable ecosystems. The paper starts by analyzing the history of proof of work and how it has evolved during the past few years. We study the use of CPUs and GPUs to mine during the lifetime of Bitcoin and the eventual development of specialized ASICs. How GPU mining is the current best solution for mining Monero because of its commitment to stay ASIC resistant and why GPU mining is the best way to build a general-purpose miner that has the flexibility to mine different coins and different algorithms.

We look at all the hardware components required to build a GPU miner, how to choose between alternatives and how this affects efficiency. During this writing and testing period many components were burned or damaged so some of the common mistakes in handling hardware will be mentioned. We will take a look at all the hardware modifications that can be made like overclocking, undervolting and modifying bios memory timings to increase mining efficiency measured in hash/watt units.

Major focus is put in understanding memory timings, how changing specific values impacts hashrate, measuring this data to quantify the efficiency benefits that can be used in profitable mining.

This thesis is an attempt to document as much as possible of the knowledge that has been flowing around lately as interest on crypto currencies has increased in the past few years.

PREFACE

Bitcoin is in my personal opinion one of the most important and groundbreaking technological and financial innovations of the last decades. It is the fusion of two of my favorite fields, computer technology and economics. It uses computation to solve the problem of centrally planned monetary policy in a way never seen before. Sound electronic money is the missing piece the internet was lacking. Although early to say, cryptocurrencies could spring a revolution in the way we think of money and decentralized consensus. Good quality money can help people live a better life by giving them independence and more control over their finances by reducing reliability on third parties.

The decentralized nature of crypto makes sure that an attack from external actors could only slow down innovation. The only way to take down crypto currencies is to take down the Internet, which would be very taxing on all global stock markets and the modern interconnected economies. This gives me confidence that crypto is here to stay and that is why I have decided to spend time to learn and research on this topic.

The only missing feature in Bitcoin to be perfect money is fungibility, which is why Monero is my favorite coin and is the focus of this paper. Proof of work, the mechanism by which crypto is created connects the technology to the economics of why these coins have value which is why I will be looking at efficiency improvements in this process.

Tampere, 02.06.2018

Daren Tuzi

CAUTION

All the testing methods described in this paper should be considered purely educational and used cautiously. Getting a higher performance out of a GPU will put more strain than usual usage like multimedia or gaming does. Constant heat generated by the hardware will damage components [23], reduce lifespan, be prone to starting fires and will void the warranties covered by the manufacturer.

CONTENTS

LIST OF TABLES AND FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

280X	An AMD card used in some calculations
7990	An AMD card used in some calculations
AMD	Advanced Micro Devices Inc
AMPS	Plural for ampere, a unit of electric current
ASIC	Application Specific Integrated Circuit
ASUS	AsusTek Computer Inc
CPU	Central Processing Unit
GPU	Graphic Processing Unit
MHz	Mega hertz
MOLEX	A two-piece pin and socket interconnection
PCB	Printed Circuit Board
PCI-E	Peripheral Component Interconnect Express
POS	Proof of Stake
POW	Proof of Work
RAM	Random-access memory
RPM	Revolutions per minute
RX 470	An AMD card used in some calculations
USB	Universal Serial Bus
VRAM	Video RAM, RAM used in GPUSs
W	Watt

DEFINITIOS

Fungibility: being something of such a nature, that one part or quantity is replaceable by another equal part or quantity, something capable of mutual substitution.

Forking: when the nodes of a cryptocurrency disagree on which rules to follow they start building on different branches of the blockchain, this results in two separate networks with a shared past.

ASIC resistance: the ability of a mining algorithm to remain profitably minable only by general purpose computing devices like CPUs or GPUs. ASIC resistance is only temporary since any defined algorithm can be implemented in specialized hardware. ASIC resistance aims to make research and development as costly as possible to economically discourage its development.

1. INTRODUCTION

The motive of this paper is to study the innovation behind proof of work as the method of issuing and securing cryptocurrencies, the efficiency of graphic cards in mining Cryptonight and improvements that can be made to increase performance and profitability while keeping costs low.

In the beginning there were only CPUs, a few guys mining in 2010 with average personal computers. At the time there were no exchanges, no price on the coin so mining was a purely technical experiment. As more people became interested and demand grew, the ability to buy and sell good for Bitcoin slowly increased. Once an economical value was established, an incentive to mine was born. At this point it makes sense economically to earn as much as possible while keeping costs down.

The mining landscape evolved very fast. The first step of this evolution was the use of GPUs. Using so many cores at once allowed for parallel computation that far surpassed what CPUs where doing. In a competitive environment like mining where the supply is fixed but the computation can increase, the introduction of GPUs meant that CPUs were no longer profitable.

Competition spurs innovation and the next step in the mining evolution were ASICs. Private companies started developing special hardware specifically designed to do that computation in hardware. These chips that could do only one job, mine a specific algorithm were very good at that job. The improvements in efficiency compared to GPUs were again so great it made GPUs non-competitive. In between GPUs and ASICs some people started using FPGA boards. This was an intermediate step which had better performance than GPUs but because of the distribution being limited and short lifespan before ASICs it had a smaller effect on the whole ecosystem.

This all happened so fast that most GPU improvements only became relevant again when people started mining Litecoin. A smaller coin that had no ASICs developed at the time. This set the precedent that even though ASICs would develop for bigger coins, there would always be a smaller coin people could switch to. Mining with general purpose hardware would live on as a hobby and as a business [20]. This motivation inspired the research on improving performance and profitability with GPUs in this paper.

2. THEORETICAL BACKGROUND

2.1 Bitcoin

Bitcoin is the first cryptocurrency introduced in 2009 by an anonymous identity known as Satoshi Nakamoto [21]. Bitcoin the network is a decentralized peer-to-peer global payment system. It is open source and cannot be controlled by any single entity. The innovation introduced by Bitcoin is that it removes the need of trusted third parties to facilitate the movement of money. This can be extended to a more general idea of the blockchain that keeps track of records in a decentralized way but as of this time, the practical use as money is the most popular product the market wants.

The blockchain keeps track of spent and unspent outputs. It is a historical ledger that can be trustlessly reconstructed from the genesis block, the first block. A Bitcoin wallet consists of private keys that control your outputs. A public key can be generated from the private key. The public key creates the receiving address that the sender needs to know to send Bitcoin to the recipient. Private keys are used to sign the right to the output to the receiver. They should never be disclosed, a leak of this information leads to unrecoverable thefts.

The network is composed of individual nodes that create transactions, receive, verify transactions from other nodes and rebroadcast these transactions to other nodes. Nodes then put transactions into blocks and append the newfound block after the most recent one, a certain amount of work is needed to do this. This process is known as proof of work.

The nodes performing this proof of work are the only ones that have write ability on the blockchain, non-mining nodes can only read. Non-mining nodes can only contribute to consensus by not broadcasting transactions that they think break the rules. This however does not prevent anything as long as those transactions are accepted by miners. A disagreement between nodes will lead into a fork, where nodes will start building on top of different blocks. This feature allows for freedom of choice to follow the rules that the node believes are the right ones.

2.1.1 Proof of work

Proof of work is an algorithm used to produce data that is costly to produce but easy to verify. Such systems are designed to waste resources in order to reduce denial of service or spam requests to servers, email or other general service providers. The idea was first introduced in a 1993 journal article [7], later reintroduced in a 1997 in Hashcash. [1]

Bitcoin uses this algorithm to generate blocks. The wasted energy secures the system since an attacker willing to generate a longer chain has to waste that much energy to try to find alternative blocks to append to his fork and try to reach a longer chain than the original honest chain.

The proof of work algorithm used in Bitcoin makes use of SHA-256 which is a member of the SHA-2 family. It was designed by the NSA and stands for Secure Hash Algorithm. Hashing is a transformation of arbitrary input data into a fixed size output data. A good hashing function appears to output almost random results so changing a single bit in the input will result in a completely different output. Hashing is a one-way function where it is easy to generate the result from the data but impossible to generate the data from the hash.

Trying to find a block simply means to build a block whose hash is smaller than a certain target. Since it is impossible to generate the block by knowing a certain target hash, miners have to brute force the solution. A block is constructed with some transactions and the hash of the previous block and multiple values of nonce are tried until the blocks hash matches the target condition.

The difficulty of the work done in these systems is dynamic and it depends on the target. Difficulty readjusts once every 2016 blocks and the new target is calculated based on the median block time of the last period [21, 3]. This means that the difficulty is dynamic and can scale up or down depending on the expanding or contracting use of hardware in mining. The target is recalculated to keep the average block time at 10 min.

Since SHA256 does not depend on memory and is rather simple. It was easy to build its logic in hardware and create ASICs for it. It was assumed until 2017 that ASICs for other algorithms would be harder to create but in 2018 Bitmain announced ASICs for most of the largest altcoins, including Monero.

In proof of work, nodes choose the chain with most work. This means that if there are two available chains the one with largest cumulative difficulty is chosen. This is usually referred as the longest chain but this is not entirely correct since a fork can have more blocks produced with lower difficulty thus resulting in lower total work put in to produce it.

2.1.2 Bitcoin as money

The purpose of Bitcoin as introduced by Satoshi is to be peer-to-peer electronic cash. Bitcoin is a very sound form of money and fits most of the properties of money [10].

- Divisible: Bitcoin is divisible. One bitcoin is worth 10,000,000 satoshis.
- Acceptable: Many merchants accept Bitcoin as a form of payment.
- Limited supply: Only 21 million bitcoin will ever exist.
- Uniform: All versions of Bitcoin have the same purchasing power, the consensus mechanism determines what happens when bitcoin forks.
- Portable: Bitcoin is very portable because of its digital nature. Moving bitcoin is simply the act of broadcasting a transation.
- Durable: As long as the Internet survives and the mining nodes are connected, the bitcoin network will function. A global electrical failure could bring the network to a halt.
- Fungible: All units should be interchangeable and indistinguishable for each other. Unfortunately in the current state of Bitcoin this property doesn’t hold due to the ability to trace every transaction and taint them according to its history.

2.2 Monero

In October 2013 Nicolas Van Saberhagen [14], believed to be a pseudonym, introduced the Cryptonote whitepaper. The paper talks about the deficiencies of Bitcoin and proposes solutions. This alternative aims to be a healthy competitor and a better electronic cash system.

The first implementation of the Cryptonote protocol was Bytecoin. It was released in March 2014 and claimed to have been online since 2012. This was later discovered to be fake and 80% of the supply had been insta-mined. This was not received well from the community who considered this move as a scam. Bytecoin code was forked and so BitMonero was born. Bitmonero would later be renamed to Monero.

Monero aims to make all coins equal. To achieve fungibility a third party should not be able to tell the sender, receiver or the amount transacted. This is so important that lives depend on it.

In a hypothetical world where a totalitarian government would come into power. A ruthless dictator could decide that all coins/outputs owned by the political opposition or journalists would be banned from usage in commerce and all merchants accepting that money would go to jail. This would only be possible in a transparent blockchain that lacks fungibility. In another futuristic world where cryptocurrencies are widely accepted, criminals could specifically target rich people by studying the payment history and looking at peoples' balances. Crimes like kidnappings would be facilitated with a transparent financial ledger. Businesses would have all their transactions available for anyone to read, losing competitive advantage and business secrets.

2.2.1 View Keys

The concept of private keys for spending and public addresses for viewing is common from bitcoin and those are sufficient since the bitcoin blockchain is transparent for anyone to see the balance of any address. Since Cryptonote introduces a more opaque blockchain view keys are used as a read only functionality to allow for the owner or other 3rd party to check incoming transactions and their balances but doesn't allow spending of those outputs [5]. This functionality can be useful for accounting or auditing purposes. The information is made available only if the owner decides. This makes Monero private by default, transparent by choice.

2.2.2 Stealth addresses

At the current implementation, Bitcoin lacks fungibility. Having an open and transparent blockchain means that once a receiver publishes an address online, everyone can see all the payments received by that person and link that identity to these other transaction. To go around this problem one could generate a new private-public key pair every time but that would make things difficult to maintain with an ever-increasing pool of keys. Multiple uses of the same receiving address are discouraged and the used of deterministic wallets, also known as HD wallets, is encouraged to help obfuscate the identity of the receiver and its history of previous received payments. Deterministic wallets generate multiple addresses from the same seed or private key. This could be as simple as SHA256(privateString+n) where n is incremented every time a new address is needed. BIP 32 [15] also implements a standard address generation using the 12 word mnemonic seed and doing 100k rounds of sha256 to slowdown dictionary attacks.

When using deterministic wallets the receiver has to generate a new address and give it to the sender. Stealth addresses answer this problem by allowing the sender to generate a different address on behalf of the receiver every time they make a new transaction. Stealth addresses can be implemented in Bitcoin but they are currently not in use. On the other hand they are an integral part of the Cryptonote codebase and currently in use in Monero.

When using a stealth address the receiver can publish only once and still all incoming payments will appear to go to random address on the blockchain [5,6,12]. These generated addresses cannot be linked back to the recipient's identity and only the sender will be aware of that transactions receiver. The receivers view key can be used to scan the blockchain to check what transactions belong to that key.

2.2.3 Ring Signatures

Ring Signatures are a class of schemes that allow a user to sign a message on behalf of a group, making his identity indistinguishable from the other members of the group [6].

Ring signatures protect the traceability of the sender by mixing the spending output with other existing outputs in the blockchain [14]. A ring of dynamic size is created and used to sign the transaction. Outputs are chosen using a triangular distribution method. All outputs appear valid and equal to any outsider observer, who are unable to tell which output was spent [17].

2.2.4 Confidential Transactions

By default Bitcoin transactions are transparent, anyone can check the amount spent from each output. Although this is necessary for nodes in a global distributed network to verify all the amounts mined and sent, there are ways to hide the amounts and still be able to verify the integrity.

Confidential transactions were first introduced by Bitcoin core developers [9] and tested on a testnet sidechain but yet to be implemented in Bitcoin. Confidential transactions were later implemented and are today used in Monero.

Confidential transaction utilize Borromean ring signatures and Padersen commitment schemes to make the amount transacted only visible to the sender and receiver. Outsiders can still sum up all the inputs or outputs to verify no extra coins where created out of nothing. Miner fees are left visible to make sure no miner can award themselves more coins.

The Monero team has then updated the confidential transaction to be ring confidential transactions, making it fit nicely with its ring signatures [22].

To hide the amount transacted commitments are used but to ensure these numbers are positive and do not overflow monero uses range proofs [2]. Range proofs allow anyone to check that a commitment represents a number within a range without revealing its value. These range proofs scale linearly with the number of outputs and they fill the majority of the space in a transaction [24]. Recent work from Bunz, Bootle etc have shown improved ways to handle range proofs, called Bullet proofs [24] .These proofs only scale logarithmically and have significantly smaller size, up to 80% reduction. The are currently being tested and audited and show great promise in improved scalability.

2.2.5 Cryptonight

Cryptonight is memory-hard hash function [19]. It is part of the Cryptonote standard and it is used as a proof of work mining algorithm. It is designed to be inefficient and close the efficiency gap between Cpu and Gpu mining. One of its goals is to disincentivize the manufacturing of ASICs by making it as costly as possible. The algorithm uses a 2mb scratchpad. The scratchpad is populated with semi random data, then read compute write operations are performed over the data more than half a million times. The result is computed by hashing the whole scratchpad.

1. Scratchpad initialization

Initially the input is hashed using Keccak with parameters b=1600 and c=512. The first 32 bytes of the Keccak output are used as an AES-256 key and expanded to 10 round keys. Bytes 64 to 191 (128 bytes) are split into 8 blocks of 16 bytes each. Each block goes through an aes_round(block, round_key[0..9]) using each of the 10 round keys generated previously. Each 128 byte output is written in the scratchpad after the previous one. (128*10) for each block. ((128*10)*8) for all blocks for a total of 10240.

2. Memory hard loop

After initializing the 2mb scratchpad comes the time consuming and most resource intensive part. Before starting the main loop bytes 0 to 31 are XORed with bytes 32 to 63 and the resulting 32 bytes are divided in two 16 byte halves, namely variables a and b are initialized from these values.

```
#convers 16byte to little endian 21bit address
scratchpad_address = to_scratchpad_address(a)
#read value in that address and aes with a, store in C
C = scratchpad[scratchpad_address] =
aes_round(scratchpad[scratchpad_address], a)
# new B takes value from current C, C xor previous B is written in scratchpad
at address A
b, scratchpad[scratchpad_address] = scratchpad[scratchpad_address],b xor
scratchpad[scratchpad_address]
#convers 16byte to little endian 21bit address, this is also address(c) since
b was assigned from C
```

```
scratchpad_address = to_scratchpad_address(b)
#scratchpad[scratchpad_address] means read(C), thts D
# then multiply D with C (thats new b) and add A,
#can be temporarely stored in a since its gonna get rewriten in next step
a = 8byte_add(a, 8byte_mul(b, scratchpad[scratchpad_address]))
#store result of addition (temp a) in D and update a with xor of addition
result (prev a) and D
a, scratchpad[scratchpad_address] = a xor scratchpad[scratchpad_address], a
#latency critical path, can it be multithreaded and have other stuff done at
the same time
```

3. Result calculation

After repeating the loop 524288 (2^19) times. Bytes 32-63 from the Keccak state in part 1 are expanded into 10 AES round keys. Bytes 64 to 191 are XORed with the first 128 bytes of the scratchpad and that is repeated with all round keys similar to the first part. After doing this for the last 128 bytes of the scratchpad, the bytes with index 64 to 191 in the Keccak state are replaced with the result. The resulting Keccak state goes through Keccak-f with parameter b = 1600. The two low order bits of the first byte are used to select one of 4 hash functions:

- 0 Blake 256
- 1 Groestl 256
- 2 JH 256
- 3 Skein 256

The chosen function is applied to the Keccak state and the result is the output of Cryptonight.

2.3 Memory timings

2.3.1 Naming

To understand timings we must first understand naming and definitions. This data comes from different manuals, such as the JEDEC standard [25]. Some of these are explained in detail in section 3.2.

- CAS latency (CL) This is the time it takes for a memory module to have data ready upon request of the memory controller . The number of cycles between sending a column address to the memory and the beginning of the data in response. This is the number of cycles it takes to read the first bit of memory from a DRAM with the correct row already open. Unlike the other numbers, this is not a maximum, but an exact number that must be agreed on between the memory controller and the memory.
- RAS latency (RAS) Row Active Time is the minimum time required for a row to be active to ensure data can be accessed from it. The minimum number of clock cycles required between a row active command and issuing the precharge command. This is the time needed to internally refresh the row, and overlaps with TRCD. In SDRAM modules, it is simply tRCD + CL. Otherwise, approximately equal to tRCD + 2×CL.
- Row Address to Column Address Delay (tRCD) the time it takes to read memory, once the memory is ready. The minimum number of clock cycles required between opening a row of memory and accessing columns within it. The time to read the first bit of memory from a DRAM without an active row is TRCD + CL.
- Row Precharge Time (tRP) the time it takes for memory to have a new row ready for using data.The minimum number of clock cycles required between issuing the precharge command and opening the next row. The time to read the first bit of memory from a DRAM with the wrong row open is tRP + TRCD + CL.
- tRRD is the row to row delay
- tRC is the row cycle time
- tWR is the write recovery time
- FAW is the Four bank active window
- 32AW is the thirty two bank active window
- tWTR is the write to read delay

RAS TIMING include:

TRCDW	Number of cycles from active to write
TRCDWA	Number of cycles from active to write with auto-precharge
TRCDR	Number of cycles from active to read
TRCDRA	Number of cycles from active to read with auto-precharge
TRRD	Number of cycles from active bank a to active bank b
TRC	Number of cycles from active to active/auto refresh

CAS TIMING include:

TNOPW	Extra cycle(s) between successive write bursts
TNOPR	Extra cycle(s) between successive read bursts
TR2W	Read to write turn
TCCDL	Cycles between r/w from bank A to r/w bank B
TR2R	Read to read time
TW2R	Write to read turn
TCL	CAS to data return latency

2.3.2 Working with memory timings

Since latency is calculated in clock cycles in synchronous RAM, bigger latency in clock cycles does not always mean worse performance. The latency in real time is calculated as *clock cycle time* * *number of clock cycles*. This means that higher clock cycles can produce lower latency if the clock is running fast enough even if latencies (ex CAS latency) are looser.

A big contributor and member of the community known as The Stilt, has released over the years [20] several modified gpu bioses, some of which have even been integrated into the bioses by the manufacturers themselves. Comparing some of the default timings vs The Stilts we can see most of the edits are TRC, TRRD.

A term used in frequently is ‘Silicone lottery’, which essentially means that the graphics card has a flawless PCB. This allows it to run at higher clock times, higher voltages and have less errors compared to average cards.

Sometimes changing timings will not result in any performance benefits. Memory timings are interconnected and depend on each other so one has to be careful to update all relevant timings when changing something.

To extract the bios data manually, one has to go thru the bios data with a hex editor and look for particular straps by comparing their hex value in little endian. For example 1500 Mhz is 150,000 in increments of 10 Khz, which is the way the frequency is stored. The decimal 1500 is 249F0 in hex, represented as F0 49 02 in little endian. So searching for ‘F0 49 02’ gives us the start of the 1500 Mhz strap, until the next strap which is 1625 Mhz and stored as C4 7A 02. Luckily there are advanced tools to speed up this process.

After getting the strap values in hex, it’s time to decode and figure out what these values mean. This information is hard to come by but talented members of the community have reverse engineered what each value is. This has been done by studying open source driver code and trial & error experimenting. The most advanced and up-to-date tools are called R_Timings and OhGodATool were used to decode timings during this paper.

To make sure the graphics card does not turn in an expensive paperweight, one has to reduce the risk of irrecoverable failure. Assuming the memory strap used by default is the 1500 Mhz, all timing modifications should be done on 1625 Mhz and higher straps. This way these timings only activate when the card memory is overclocked in that range. In case of errors that cause operation failure, one is able to restart the card and it should fall back to default values.

In optimizing the timings the goal is to achieve the highest memory clock possible with the tightest timings that would work on that speed. If the memory of a specific card can go up to 2000 Mhz and the 1500Mhz timings can only operate normaly until 1900 Mhz and fail to run at 2000 Mhz, it may be better to use a 1625 Mhz strap that runs safely at 2000 Mhz.

Finding the best timings also depends on the memory controller. Newer generation of GPUs like Polaris have better memory controllers with larger queues which allows for multiple memory reads to be grouped together. This leads to lower importance of FAW and 32AW. The big advantage on newer cards comes for higher memory speed in comparison to older generations that would benefit more from tighter timings.

METHODOLOGY AND MATERIALS

2.4 Efficiency goals

The goal of the work that was done in this paper is to improve the efficiency of existing hardware. To quantify changes hashrate per second will be used as primary measure of performance. In addition to raw hashrate, hashrate/second per watt spent will also be used to indicate power usage. Minimal use of power while retaining high hashrate will be the target. Lower consumption of power is related to smaller power supplies, lower temperature, lower electricity bill and ultimately lower cost.

2.5 Hardware

2.5.1 Power supply

Often overlooked the power supply is one of the most important components when building a mining rig. A high quality and reliable power supply will increase the profitability, ease of set up and longevity of all the components, after all it would be a shame to invest into high end hardware and spend time into optimizing it if the power supply was not reliable.

A few elements come into play in choosing a power supply. The most important is the total amount of power it can deliver. For a typical gaming computer the most power consuming components are the cpu and the gpu so a 600 watt power supply is usually enough. In this case a mining rig with multiple gpus will require more than that. During testing 850 and 1000 watt power supplies were used.

Power supplies are rated according to their efficiency into multiple categories. All PSUs with over 80% efficiency are rated 80+ and subcategories such as white at 80%, bronze 82-85%, silver 85-88%, gold 87%-90%, platinum 90-95%. This measurement is important and taken into account when considering total power consumption in watt. A more expensive psu will cost more initially but will reduce power consumption in the long term.

Usually power supplies have peak efficiency at 50% +- 5% range [16]. This is also the starting guideline in calculating which power supply would be most efficient considering the current rig power requirements. Calculating for 6 gpus at 90w each plus cpu a 1200w gold standard psu would be more most efficient.

In addition to this straightforward method, one could also separate the workload to two power supplies. One powering the board, cpu and other peripherals and the second one powering the gpus only. This could be done to make use of existing hardware or buying of less expensive smaller power supplies. For this to work properly both power supplies need get the start signal at the same time. Since a typical motherboard has only one 24 pin connector a simple adapter with a female 24 pin connector and a female 4 pin molex connector can be used to signal the second gpu. In normal operation the first psu would be connected to the motherboard, a molex cable from the first psu connected to the adaptor would signal the second psu to start The 24 pin cable of the second power supply would be connected to the adaptor. In this setup the start and stop signal from the power on/off will simultaneously control both power supplies.

Another aspect relevant to power supply choice is modularity. If one or two power supplies are used both will need the 24 pin cable. Powering the peripherals will require 6 pin to molex or sata cables and of course 6 and 8 pin pci-e cables to power gpus [11]. A power supply with larger power output will usually have more output ports so calculating if one or two power supplies will have enough total ports needs to be considered when choosing the components.

2.5.2 PCI-E connection

Most motherboards have limited space and come with one to three 16x ports and a couple of more 1x pci express ports. Since gpus come with male 16x ports we need 1x to 16x conversion cables to make use of these 1x ports to connect more gpus. In addition to increased gpu number this also allows for creating distance between the gpus and the motherboard and between the gpus themselves allowing for more air flow, cooler temperatures and thus increasing hardware lifetime. Some older motherboards will not recognize the 1x extensions by default so a jump cable is used to connect the first top left pin with the second pin from the bottom right, this technique will not be discussed further here since it was not required with the H81 Pro Btc board.

Gpus are designed to use the 16x since it allows up to 16 parallel connections to the CPU and the 1x have only one connection. This affects gaming performance mostly since throughput is important but not monero mining since the hashing is done in the devices internal vram. Depending on the year of production of the motherboard, pci express can have different speeds. The board used during testing had a version 2 port which allows for 500MB/s on 1x ports more than enough for the data transfer needed to supply the gpu with the next job.

There are two types unpowered and powered cables. Unpowered cables only do data transfer and could be enough if only 1-2 gpus are connected. Since maximum efficiency is our goal, up to 6 gpus will be connected to the same board thus power consumption will increase. Depending on the motherboard this power requirement will be too much. There are cases where the motherboard has molex connectors that can be connected to the power supply but it is better to offload the power to the 6-pin, 8-pin pci-e power cables that connect directly to the gpu and use powered risers to reduce stress on the motherboard thus increasing longevity. Modern powered risers come in 4 components: the 1x male connection with a USB output, a USB to USB cable, a female 16x board with an USB input and molex power port. It is recommended that the molex power is plugged directly from the power supply and not thru a sata cable to the motherboard. This will increase power load on the motherboard and put stress on the sata cables which can deliver only 9 amps compared to molex at 22 amps. [18] During the testing done in this paper molex powered USB risers were used in the testing rig.

2.5.3 Processor

When considering gpu mining and ignoring CPU mining, the processor doesn’t do any hashing itself but works on delivering the work to the gpus and keeping the operating system running. This means that a powerful CPU will have little to no effect on the hashrate. During these tests, a processor with only two cores was used and it was enough to have the mining rig produce the same results it did on other boards with stronger processors. A lower limit could be the number of threads a CPU can handle for each GPU connected on the board. This can be addressed by adjusting workload and giving the gpu higher difficulty shares.

2.5.4 Memory

The 2 Mb scratchpad used in Cryptonight mining usually fits in the L3 cache in most modern cpus The most time consuming part is iterating over the scratchpad half a million times in read/write operations. If the mining was being done in an old cpu and the scratchpad had to be stored in Ram then the frequency, latency of the memory would matter.

Considering that most gpus have large memory and that all operations are done on this memory, the amount or speed of physical ram on the motherboard would not affect performance.

In systems where multiple gpus are connected in the same motherboard, the operating system will require sufficient physical and virtual memory to operate its processes efficiently.

2.5.5 Motherboard

To achieve maximum hashrate/watt with minimal hardware cost a motherboard with maximum number of pci-e slots is ideal. Being able to connect as many gpus as possible to the same motherboard reduces cost of purchasing additional processors, memory and other peripherals.

There are multiple motherboards with six and seven pci-e slots but some of these will require additional work to get all 6 gpus working since the hardware software compatibility wasn’t optimized. One reliable board which was used also during these tests was the h81 pro btc. This board has 6 slots and worked without any troubles.

When choosing boards one could also consider the version of the ports since gen v1, v2 and v3 will have different bandwidth and latency. These differences are relevant in gaming or rendering but become irrelevant in cryptonight mining since the amount and frequency of the data transferred to the gpu is minimal. Cards connected to generation 1 pcie ports will have the same hashrate as if they were connected to gen 3 ports. The same principle applies to 16x and 1x ports. A 1x pcie port will be enough to connect a gpu using a 1x to 16x extension cable. When choosing the motherboard one has to keep in mind the supported ram frequencies match purchased ram.

By the time this paper was written new motherboards like the ASUS B250 and the ASUS H370 have been released supporting up to 20 GPUs at a time. A great improvement in hardware management and cost reduction.

2.5.6 Graphic cards

The graphics card, mostly referred as gpu or gpus (multiple) during this paper is the main and most important component of a mining rig purely for being the main hardware where the computation is performed on. All the other main and peripheral parts are ultimately serving the gpu. All hardware and software optimizations discussed are aimed at the graphics cards and much less onto the processor or ram.

When choosing a card to use one of the most important factors to consider is the vram/memory. Since the cryptonight algorithm is a memory hard problem it depends on fast random access of memory. This means that most of the time is ‘wasted’ on reading and writing data on the memory. Improving memory timing will significantly change performance of the card for this algorithm. This is so significant that even cards with more core processors or higher clock rate will perform less than a graphics card with a better memory latency.

It is common for different cards of the same model to perform slightly different because of the memory used in their design. Different brands will use memory produced from different manufacturers like Hynix, Elpida or Samsung. According to my measurements the difference between vram producers can result in hashrates with up to 12% difference. The highest results for cryptonight are usually achieved with Hynix memory. This could be helpful when purchasing second hand cards because of the possibility to check the type before the purchase. When buying a brand new card from the store, there is usually no definitive way to tell what kind of memory they have. Boxes usually do not indicate this and it is common for the same brand to use memory from different producers in the same product line. Nevertheless the are patterns and ways to estimate the probability of a card having a certain memory producer by looking at online reports of other peoples purchases in gaming forums.

Another main aspect to consider is the power consumption of a graphics card. Lower power requirements mean the card will consume less electricity resulting higher hashrate per watt, in lower electric cost, lower temperature and overall increased lifespan.

When different manufacturers make products based on the same model core, they build around it PCBs that might have different design and different components. Having quality components will extend the lifespan of a card and allow larger flexibility to modify its performance. Choosing cards with 8 pin pcie power cables over 6 pin cables will also reduce the load on each individual cable and increase power upper limit.

Graphic cards with larger heat sinks will take more space but are great for dissipating heat and getting lower temperature. Cards that have fans with larger diameter will also move more air and produce lower noise. As an example a gpu with 2 large fans may be better than one with 3 small ones since the smaller fans will need to have a higher rpm to achieve the same air flow/temperature.

Another important feature that sometimes is overlooked can be the ability to dual bios switch. Some modern cards come with two BIOSes instead of one. This could be incredibly helpful when testing bios modifications in case one goes wrong. A second bios will allow one to safely boot the card again after the first one is corrupted. A similar mistake would render a single bios card directly unusable.

2.5.7 Damaged Hardware

While testing on new cards there's always a slight chance that the hardware is flawed or has physical internal defects that originate from the production process. To make sure this is the case one needs to perform several tests of the other parts to make sure the fault does not lie with the existing components.

This happened with a graphics card during hardware assembly of one of the rigs. The card was new and the existing hardware was used with some other cards previously. Upon connecting this card the monitor would work correctly with only default Microsoft drivers and upon updating to the latest drivers for this card the resolution would update to highest possible and the card was successfully recognized by Afterburner and GPU-Z. In idle mode the card would run at 300 Hz but once the card was in use by a game or mining software it would crash the system. Since this particular card was tested in several different computers it would give different problems in different setups, in the first it would blue screen with "thread stuck on device driver" exception which would indicate most probably driver problems, in the second the operating system would say that the driver .dll files were corrupted and the computer would freeze and needed restart, in the third the software would say that OpenCL calls would fail and the operating system would be unresponsive. To tackle these cases and test the root cause these are the steps that were taken:

Tested two different power supplies an 850 watt and a 1000 watt both of gold quality at 90% efficiency with no change in outcome. Both power supplies were tested with multiple 8-pin pci-e cables in different psu ports to make sure there are no cable defects or inadequate power delivery that could cause such gpu behavior.

While testing the processor and gpu temperatures were monitored to see if a dramatic increase in temperature would cause operational failures but the temperature remained constant and within normal operation range.

To test memory, 20 GB of virtual memory were allocated by the operating system, physical memory was tested with memtest and physical ram sticks where inserted one by one making sure there are no damaged parts.

All components where tested in two motherboards with multiple mining software and both rigs have proven themselves to work with multiple gpus at the same time so it's safe to conclude that card that was being tested had physical defects.

2.6 Software

There are many different miners written by different people. These miners support amd or Nvidia cards in Windows or Linux. According to some tests there is no perfect or best program among these as it depends on the card that is being used. One has to test different programs and see which one performs best for own card.

- Xmr-stack
- Ccminer-cryptonight
- Claymore
- Xmrig
- sgminer

When using a custom bios one has to be watchful of which drivers they are using. Old drivers up to 16.11.5 allowed custom bios to run on windows but later drivers would not work properly. Amd has since released 'Blockchain drivers' specifically made for mining.

3. RESULTS AND ANALYSIS

Although a GPU works fine in mining on its own default settings. Improvements can be made to increase hashrate, reduce power consumption thus increasing efficiency.

3.1 Overclocking

Graphic cards come with a standard core clock speed per model. This frequency can be different for models produced by different manufacturers even when using the same chip. The core and memory clock are chosen to produce the best results and be reliable in common usage by the average user. Since an out of the box card will be used by a variety of users the default settings are the most tested doing production and will guarantee stability, reliability and longevity.

Having said that, there is always space for tweaking certain things to personalize hardware usage to improve efficiency in specific tasks. Such tweaks like overclocking core or memory frequency are commonly used by the gaming community to increase frames per second in resource heavy games. This includes casual gamers aiming for a 5% increase and those who are simply trying to find the limits of the hardware [23] by cooling the chip with liquid nitrogen to keep it from starting a fire.

3.1.1 Core clock frequency

The first and most straightforward test to increase mining hashrate is to overclock the core. This will allow it to do more cycles and in theory do more hashes per second. Chart 3-1 demonstrates this.

The x-axis is the core clock frequency. This particular card comes with a default core clock at 1000 MHz and memory clock at 1500MHz. The y-axis represents the hashrate this card alone produces with the default settings.

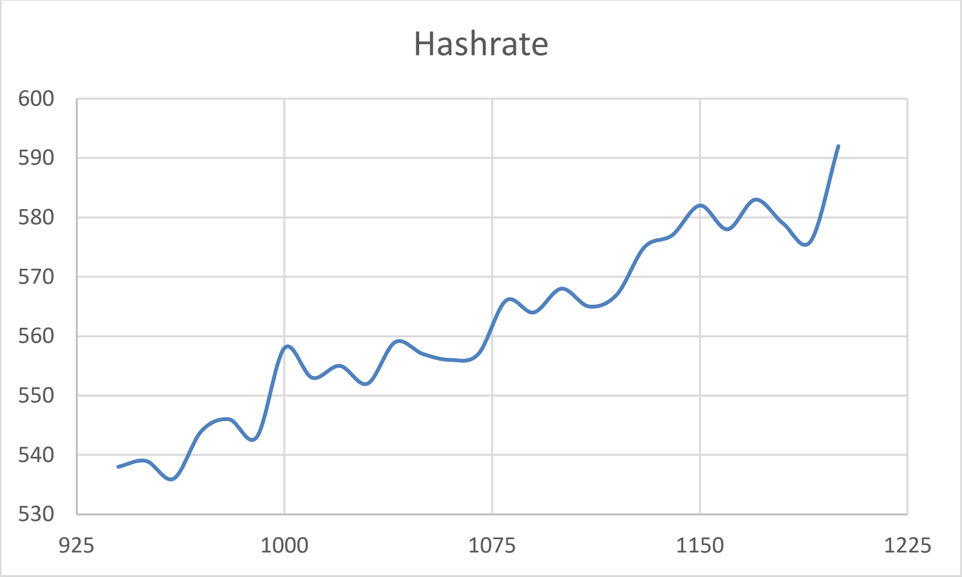


Figure 3-1

An increase of 15% from 1000 MHz to 1150 MHz produced an increase of 4.3% from 558 to 582.

An increase of 20% from 1000 MHz to 1200 MHz produced an increase of 6.1% from 558 to 592.

Such a gain comes with a penalty. Increasing the core clock will requre more power consumption and greater heat output. To maintain safe temperature ranges, higher fan rotation frequency will be needed. This will result in increased noise and a need for better airflow.

The increased core frequency will require suficient power draw from the board. Higher power requirement will put stress on the cables, power suply and increase electrical cost overall.

Chart 3-2 shows the increase in power consuption during the overclocking of the same card. The default core voltage is 1144 mV and was kept constant during all clock frequencies. Overclocking the core at 1190 Mhz caused ocasional flickering and corrupted graphics at the default voltage and completely crashed at 1200 Mhz. The core voltage was increased at 1200mV to keep the core running at 1200 Mhz.

The x-axis is the core clock frequency. The y-axis represents the power in watt drawn while the card is operational. This was calculated by measuring the total power in watt consumed by the rig at any clock frequncy and subtracting the total power consumed when the rig was idle, not mining.

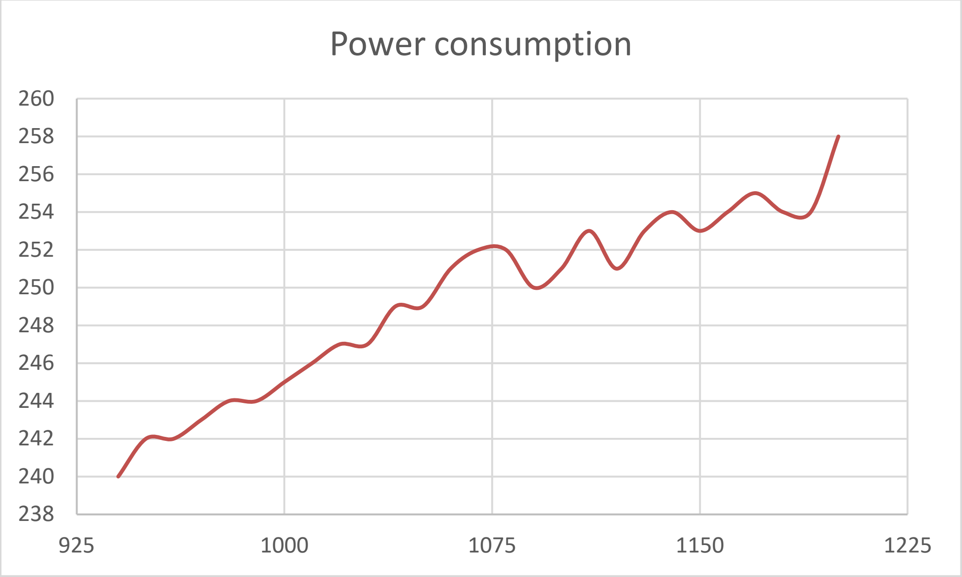


Figure 3-2

An increase of 15% from 1000 MHz to 1150 MHz produced an increase of 3.2% from 245 to 253.

An increase of 19% from 1000 MHz to 1190 MHz produced an increase of 3.7% from 245 to 254.

An increase of 20% from 1000 MHz to 1200 MHz produced an increase of 5.3% from 245 to 258.

The core voltage change from 1190 to 1200 can be seen here causing a sudden increase in power consumption. Section 4.3 goes more in depth into voltage control.

If the goal were to simply get the most hashrate then overclocking would be fine, assuming adequate cooling is provided and power consumption is not an issue. This rarely is the case though and to study efficiency we can combine this data and calculate hashrate per watt.

Figure 3-3 is calculated by diving the hashrate produced in figure 3-1 with the power consumed by the same figure 3-2. The x-axis is the core clock frequency in MHz and the y-axis is the hashrate per watt.

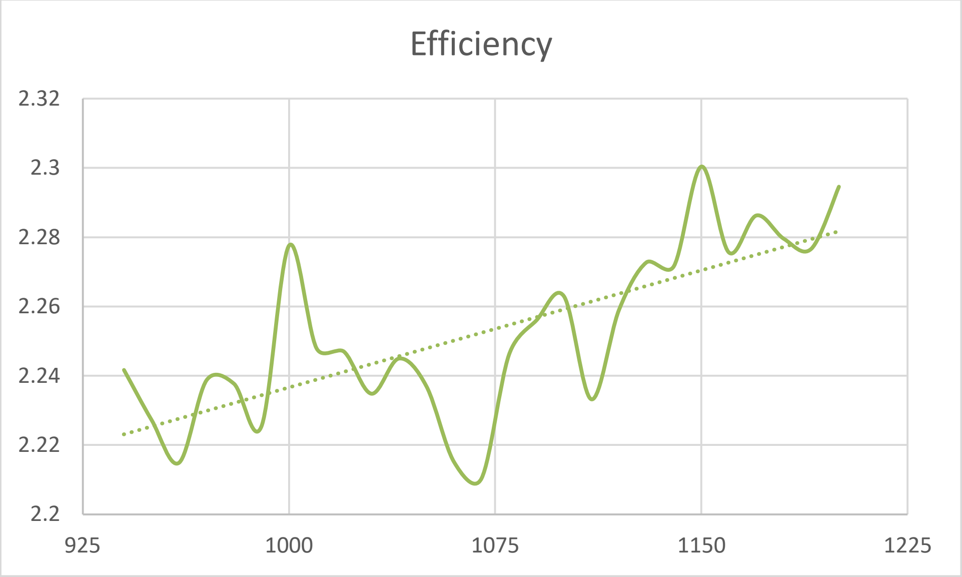


Figure 3-3

We can see here that the core clock at 1000 MHz, which is also the default value, has an efficiency of 2.28 hashes per watt. Overclocking to 1150 MHz seems to be the best efficiency at 2.3 hashes per watt. This shows that any other overclock frequency even if producing a bigger hashrate would consume more power compared to the default clock. At 1200 MHz, the card starts behaving in a non-stable way, will produce bad shares, crash and reduce card longevity.

Although the trend line seems to have a positive slope assuming no core clock limitations this is not always true. Figure 3-4 shows the same efficiency calculation done in another card. This card is a newer model with a different chip and allows for higher clock frequencies. The memory clock was kept constant at 1500 MHz.

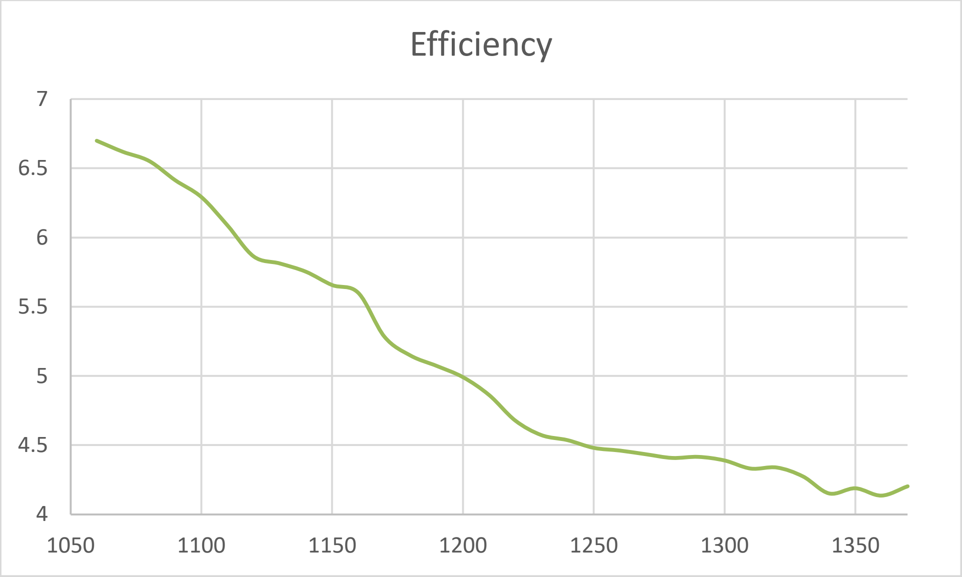


Figure 3-4

What we see in Figure 3-4 is that even if we increase the core clock its efficiency will decline because the power consumption rises faster than the hashrate produced, as shown in Figure 3-5. The power consumption has a higher positive slope while the hashrate does not increase as much. The y-axis higher limits are set to highest values and lower limit to 50% of the high to show a more accurate comparison of % change in values.

Now the bottleneck is the memory. Even though the core is more powerful, the memory clock is not fast enough to serve it in time. Especially in an algorithm like Cryptonight where memory latency is so important, the effect of this bottleneck can be seen more clearly.

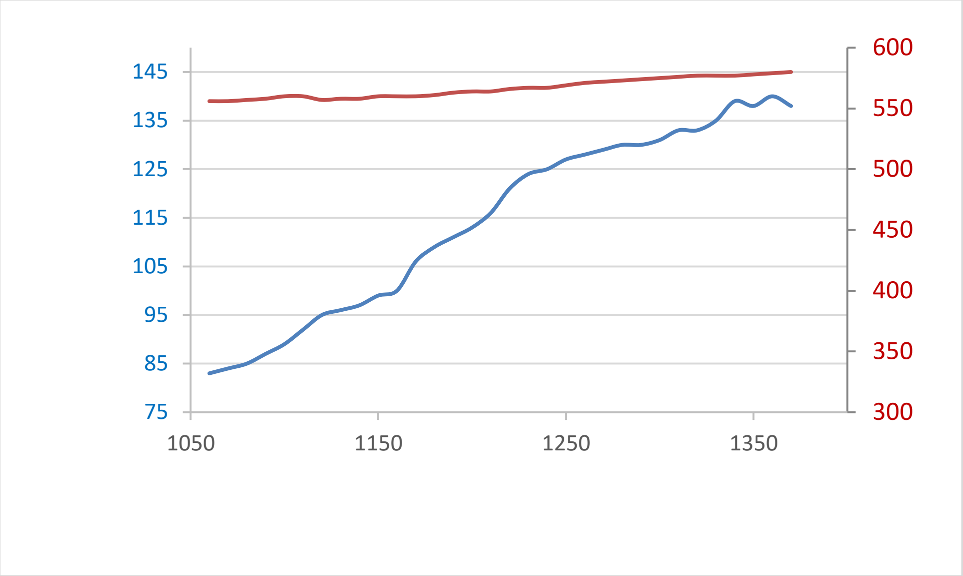


Figure 3-5

3.1.2 Memory clock frequency

The memory clock is responsible for determining how frequently data is fetched or written in memory. Increasing the memory clock means more data will be transferred to and from vram, which increases bandwidth and lowers latency. Since Cryptonight is a memory hard algorithm, cutting down on the time spend to read/write data will improve efficiency.

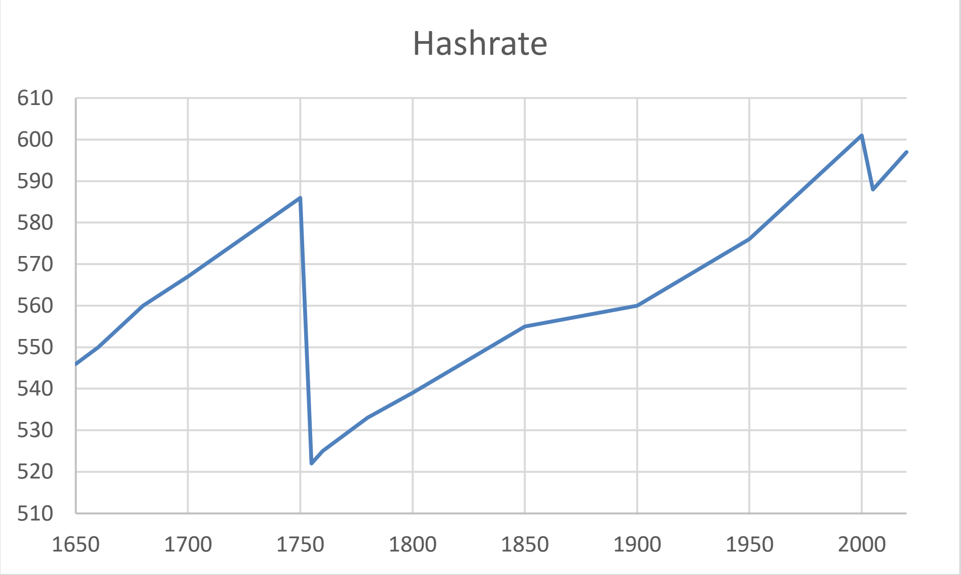


Figure 3-6

Figure 3-6 shows the increase in hashrate seen when overclocking the memory clock while keeping the core clock stable at 1250 MHz. The gpu used here comes with a default 1650Mhz memory clock.

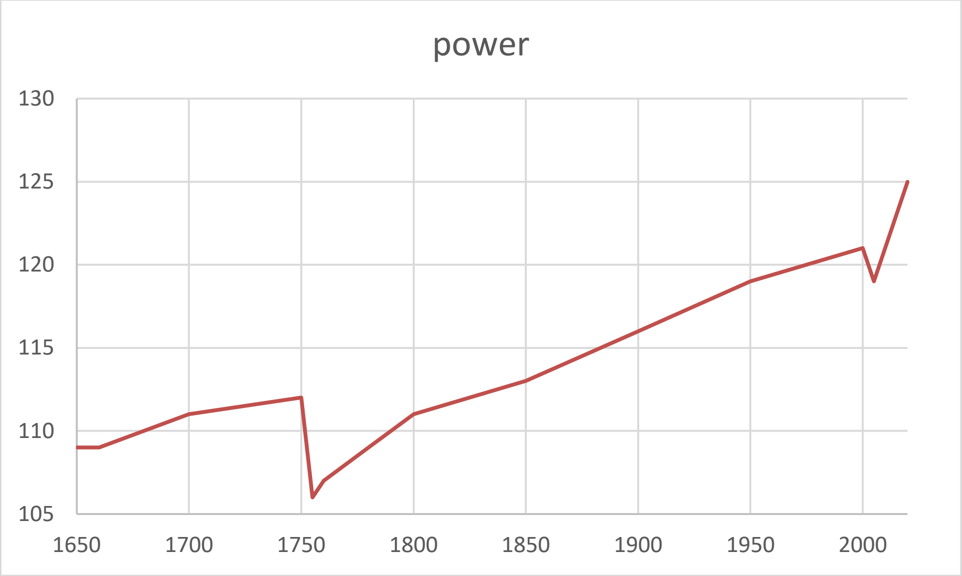


Figure 3-7

The increased memory frequency also required an increased power consumption, which can be seen in Figure 3-7. Since the rate of increase in power consumption is higher than the hashrate produced, this results in decreased efficiency, shown in Figure 3-8.

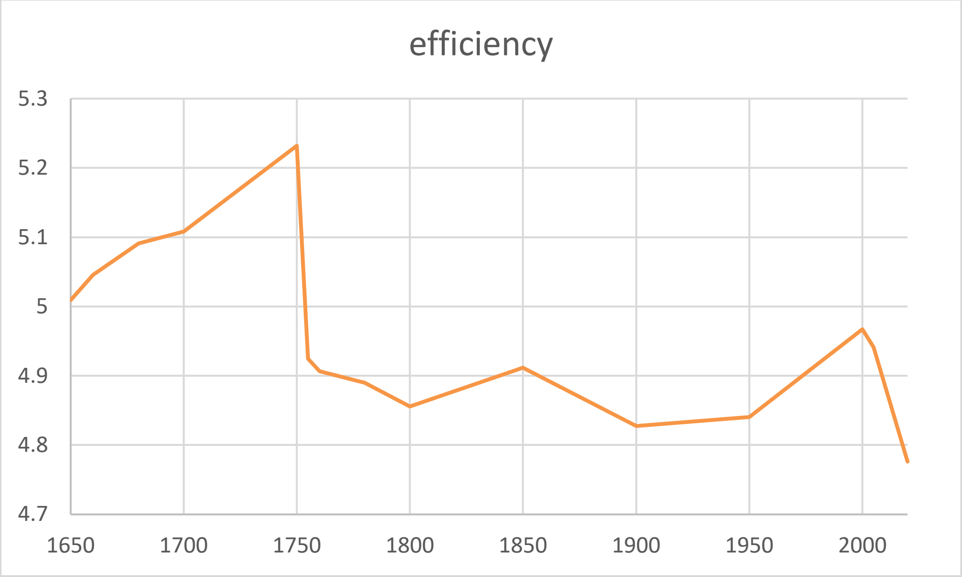


Figure 3-8

Notice the sharp decline of hashrate right after 1750. This is caused by changing memory straps. The data stored in the bios of this gpu reveals, where exactly these changes occur. At 400, 800, 900, 1000, 1125, 1250, 1375, 1425, 1500, 1625, 1750 and 2000 MHz. this gpu runs at different memory timings. A memory strap like 1626-1750 means that if the card is operating at a frequency that falls in that range, specific values will be used. Once the frequency goes over that range, the timings of the next strap will be used. With increasing frequencies, manufacturers tend to loosen the memory timings and relax them for reliability and lower probability for error. Lower memory clock frequencies usually have tighter timings. This is the reason for the decrease in hashrate seen right after 1750. Even though the frequency is higher, the clock is not synchronized which leads to wasted time.

This card has Hynix vram and the default bios memory timings are

1500 MHz

7770000000000000022339D00CE516A3D9055111230CB4409004AE600740114206A8900A00
2003120150F292F94273116

1625 MHz

9990000000000000022559D0010DE7B4480551312B78C450A004C0601750414206A8900A002
00312018112D34A42A3816

1750 MHz

9990000000000000022559D0031627C489055131339CDD50A004C06017D0514206A8900A00
200312019123037AD2C3A17

2000 MHz

BBB0000000000000022889D0073EE8D53805515133ECF560C004E26017E0514206A8900A00
20031201C143840C5303F17

To test this, the memory timings of 1500 MHz are copied into 1625, 1750 and 2000 MHz and flashed into the card. Hashrate is recorded again at default 1250 MHz core clock speed.

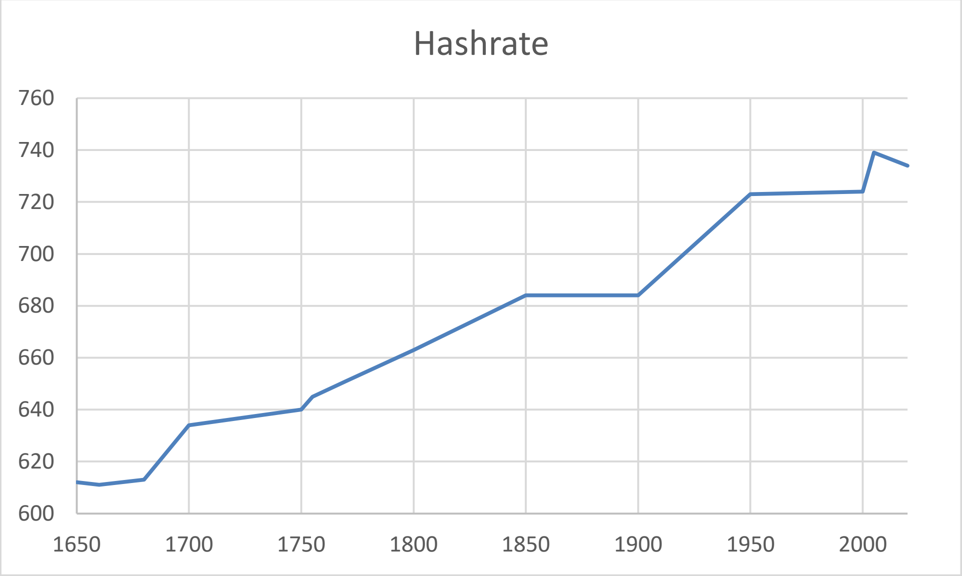


Figure 3-9

Figure 3-9 shows the hashrate achieved with the new memory timings modification. The hashrate increase is more consistent and has no sharp falls caused by different memory timings. The power consumption is also recorded in Figure 3-10.

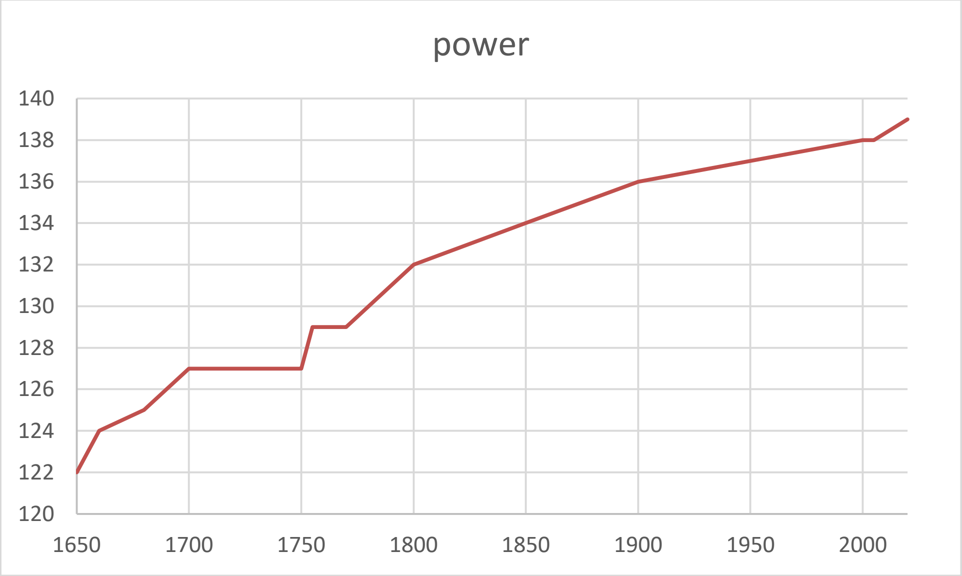


Figure 3-10

Calculating for efficiency, we can see an uptrend in Figure 3-11. Overclocking the memory clock frequency from 1650 to 2000 MHz results in an increase of **18.3%** in hashrate from 612 to 724 and an increase of **4.5%** in efficiency from 5.02 to 5.25.

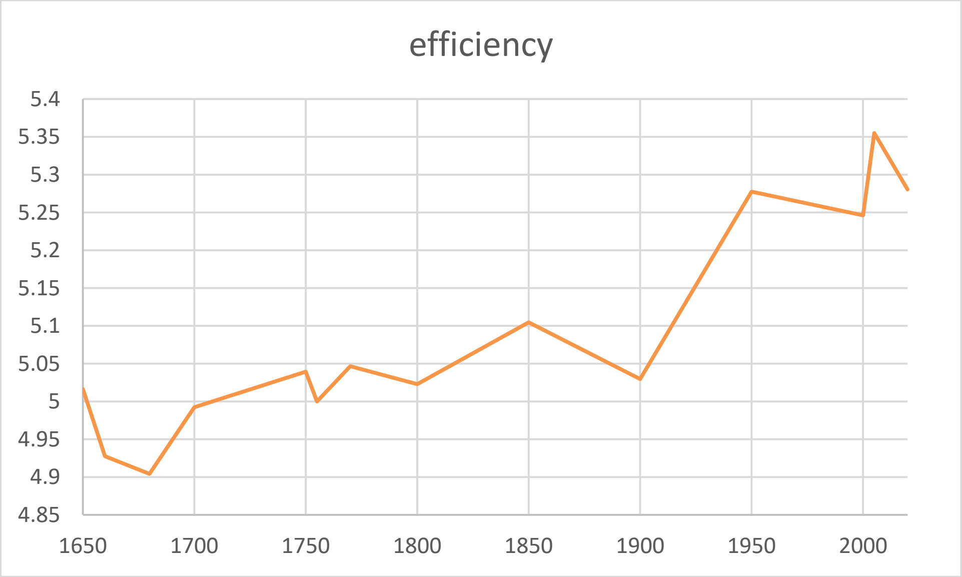


Figure 3-11

3.2 Custom memory timings

The first step towards editing the memory timings is understanding them. To do this we start by reading and decoding the default bios. AtiFlash is a tool for windows used to read and write/flash a new bios on a gpu [13]. This tool is used initially to read the default bios of the card. The file is saved as .rom . Next, PolarisBiosEditor is used to read the rom file. This tool, designed specifically for Amd cards that have the Polaris Architecture, is used to read the rom file of our Rx 470 card with Hynix memory. This is done for convenience and timesaving. In the absence of these kinds of tools, early bios reads where done manually by dumping the hex code and trying to understand its parts. The PolarisBiosEditor gives us a general overview of what's inside the bios, such as device and vendor id for the card, fan control information, power control and limits, core clock voltage control, memory straps etc.

The memory straps have encoded values such as:

```
777000000000000022339D00CE516A3D9055111230CB4409004AE600740114206A8900A002003
120150F292F94273116
```

for the 1500 Mhz strap.

The next step is to decode these straps into individual values. Information like this is not readily available and hard to find but thanks to dedicated people in this space who did the reverse engineering and test, tools have been developed to decode these values. OGodATool and R_Timings are popular ones used to decode these timings. The information used to understand the encoded values comes from different sources:

- The linux kernel files
- Jdec standard
- Hynix, Samsung and Elpida documentation

Decoding the default 1500 strap presented above with R_Timings outputs a result like this:

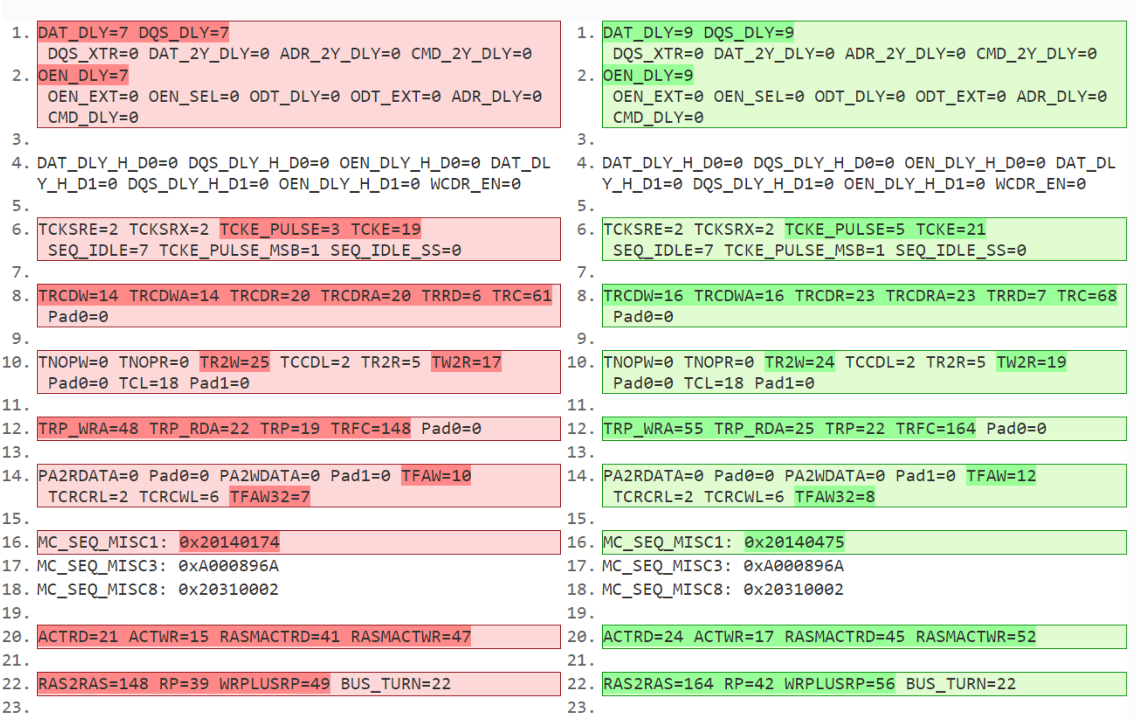
```
####SEQ_WR_CTL_D1####
DAT_DLY = 7
DQS_DLY = 7
DQS_XTR = 0
DAT_2Y_DLY = 0
ADR_2Y_DLY = 0
CMD_2Y_DLY = 0
OEN_DLY = 7
OEN_EXT = 0
OEN_SEL = 0
ODT_DLY = 0
ODT_EXT = 0
ADR_DLY = 0
CMD_DLY = 0

####SEQ_WR_CTL_2####
DAT_DLY_H_D0 = 0
DQS_DLY_H_D0 = 0
OEN_DLY_H_D0 = 0
DAT_DLY_H_D1 = 0
DQS_DLY_H_D1 = 0
OEN_DLY_H_D1 = 0
WCDR_EN = 0

####SEQ_PMG_TIMING####
TCKSRE = 2
TCKSRX = 2
TCKE_PULSE = 3
TCKE = 19
SEQ_IDLE = 7
TCKE_PULSE_MSB = 1
SEQ_IDLE_SS = 0

####SEQ_RAS_TIMING####
TRCDW = 14
TRCDWA = 14
TRCDR = 20
TRCDRA = 20
TRRD = 6
TRC = 61

####SEQ_CAS_TIMING####
TNOPW = 0
TNOPR = 0
TR2W = 25
TCCDL = 2
TCCDS = 5
TW2R = 17
TCL = 18

####SEQ_MISC_TIMING####
TRP_WRA = 48
TRP_RDA = 22
TRP = 19
TRFC = 148

####SEQ_MISC_TIMING2####
PA2RDATA = 0
PA2WDATA = 0
TFAW = 10
TCRCRL = 2
TCRCWL = 6
T32AW = 7
TWDATATR = 0

####ARB_DRAM_TIMING####
ACTRD = 21
ACTWR = 15
RASMACTRD = 41
RASMACTWR = 47

####ARB_DRAM_TIMING2####
RAS2RAS = 148
RP = 39
WRPLUSRP = 49
BUS_TURN = 22

####MC_SEQ_MISC####
MC_SEQ_MISC1 = 0x20140174
MC_SEQ_MISC3 = 0xA000896A
MC_SEQ_MISC8 = 0x20310002
```

Now we can start comparing different straps to see how these values change from lower to higher clock speeds. Most of the time values in higher straps have larger values which mean more time is given for the completion of certain jobs. This is done to increase stability and reduce errors at higher speeds.

Since graphic cards are produced for general purpose computation they have safe default timings that can assure proper function in different tasks. We can try to tighten up these timings for our specific use, that of running the cryptonight algorithm.

Since cryptonight is heavy on random access reads of the memory the thesis is that reducing that delay as much as possible should increase the performance of our card.

We start by looking at the difference between these straps to identify patterns and learn more about how the default values change from each other. Below is the difference between the 1500 Mhz and 1625 Mhz straps.



```
DAT_DLY=7 DQS_DLY=7
 DQS_XTR=0 DAT_2Y_DLY=0 ADR_2Y_DLY=0 CMD_2Y_DLY=0
OEN_DLY=7
 OEN_EXT=0 OEN_SEL=0 ODT_DLY=0 ODT_EXT=0 ADR_DLY=0
 CMD_DLY=0

DAT_DLY_H_D0=0 DQS_DLY_H_D0=0 OEN_DLY_H_D0=0 DAT_DL
Y_H_D1=0 DQS_DLY_H_D1=0 OEN_DLY_H_D1=0 WCDR_EN=0

TCKSRE=2 TCKSRX=2 TCKE_PULSE=3 TCKE=19
 SEQ_IDLE=7 TCKE_PULSE_MSB=1 SEQ_IDLE_SS=0

TRCDW=14 TRCDWA=14 TRCDR=20 TRCDRA=20 TRRD=6 TRC=61
 Pad0=0

TNOPW=0 TNOPR=0 TR2W=25 TCCDL=2 TR2R=5 TW2R=17
 Pad0=0 TCL=18 Pad1=0

TRP_WRA=48 TRP_RDA=22 TRP=19 TRFC=148 Pad0=0

PA2RDATA=0 Pad0=0 PA2WDATA=0 Pad1=0 TFAW=10
 TCRCRL=2 TCRCWL=6 TFAW32=7

MC_SEQ_MISC1: 0x20140174
MC_SEQ_MISC3: 0xA000896A
MC_SEQ_MISC8: 0x20310002

ACTRD=21 ACTWR=15 RASMACTRD=41 RASMACTWR=47

RAS2RAS=148 RP=39 WRPLUSRP=49 BUS_TURN=22
```

```
DAT_DLY=9 DQS_DLY=9
 DQS_XTR=0 DAT_2Y_DLY=0 ADR_2Y_DLY=0 CMD_2Y_DLY=0
OEN_DLY=9
 OEN_EXT=0 OEN_SEL=0 ODT_DLY=0 ODT_EXT=0 ADR_DLY=0
 CMD_DLY=0

DAT_DLY_H_D0=0 DQS_DLY_H_D0=0 OEN_DLY_H_D0=0 DAT_DL
Y_H_D1=0 DQS_DLY_H_D1=0 OEN_DLY_H_D1=0 WCDR_EN=0

TCKSRE=2 TCKSRX=2 TCKE_PULSE=5 TCKE=21
 SEQ_IDLE=7 TCKE_PULSE_MSB=1 SEQ_IDLE_SS=0

TRCDW=16 TRCDWA=16 TRCDR=23 TRCDRA=23 TRRD=7 TRC=68
 Pad0=0

TNOPW=0 TNOPR=0 TR2W=24 TCCDL=2 TR2R=5 TW2R=19
 Pad0=0 TCL=18 Pad1=0

TRP_WRA=55 TRP_RDA=25 TRP=22 TRFC=164 Pad0=0

PA2RDATA=0 Pad0=0 PA2WDATA=0 Pad1=0 TFAW=12
 TCRCRL=2 TCRCWL=6 TFAW32=8

MC_SEQ_MISC1: 0x20140475
MC_SEQ_MISC3: 0xA000896A
MC_SEQ_MISC8: 0x20310002

ACTRD=24 ACTWR=17 RASMACTRD=45 RASMACTWR=52

RAS2RAS=164 RP=42 WRPLUSRP=56 BUS_TURN=22
```

Figure 3-12

3.2.1 tFAW & t32AW

The next step is to start looking at individual values in all straps. Below are the values extracted for tFAW and t32AW.

According to the bank restrictions section (7.6) of the JEDEC standard, to ensure the ability of the board to provide instantaneous current, there needs to be a limit to the number of activities in a rolling window. FAW defines the short-term capability of the device to provide current. On the other side, 32AW defines the same capability for the longer term. No more than 32 banks can be activated in a rolling 32AW window.

An example value of 8 for FAW means that if at time T an ACTIVE command is issued, no more than 3 additional ACTIVE commands can be issued at clocks T+1 until T+7.

By controlling FAW, we can control the amount of instant currents in that short window. The advantage of this is higher throughput, which leads to higher hashrate, on the other hand it can lead to errors on older hardware that cannot handle such strain.

Strap	FAW	32AW
1000	8	6
1125	8	6
1250	8	6
1375	8	6
1425	8	7
1500	10	7
1625	12	8
1750	12	8
2000	14	9

Table 3-1

Looking at the data gathered from the default bios we can see that higher hashrates have bigger FAW windows. To try out these default values, custom bioses with custom timings containing these FAW and 32AW pairs were made. In addition (0,4) and (0,0) custom pairs were added to test the effect of 0 value FAW and 32AW. All these values were tested in a constant clock speed of 1650 Mhz. The results are presented in the chart below. We can see that larger FAW windows like 12 really restrict the amount of ACTIVE commands that can be issued. This has a negative effect in cryptonight due to its heavy dependence on random reads.

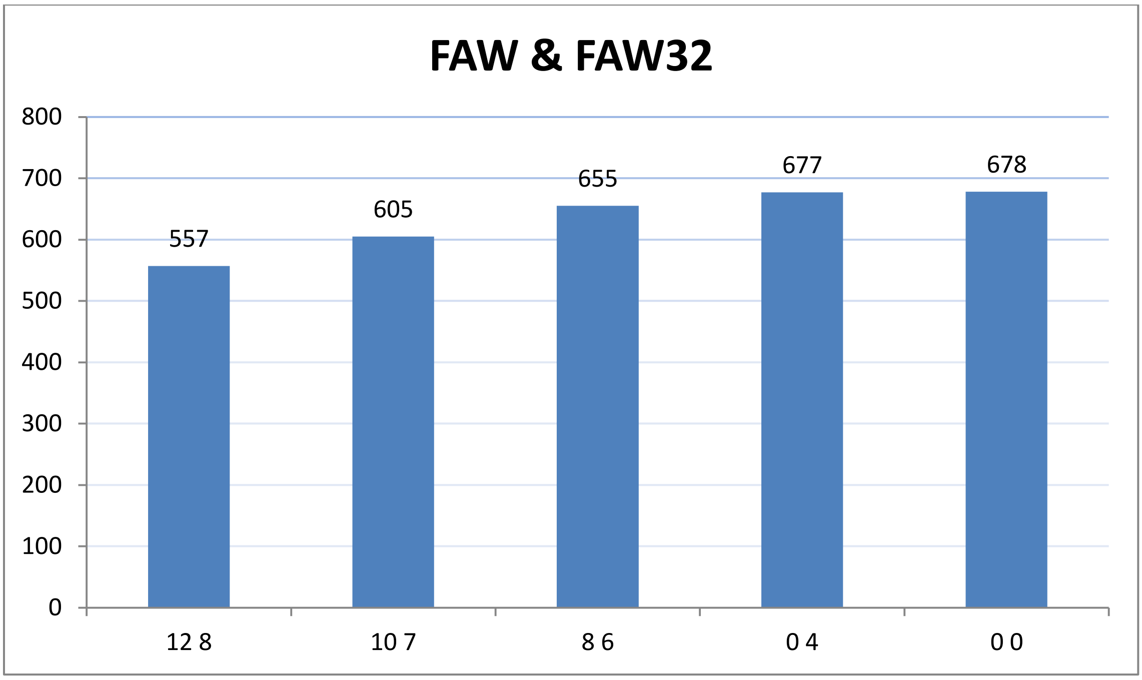


Figure 3-13

3.2.2 tRRD

Whenever we need to read or write, an ACTIVE command has to be issued to the row in a specific bank. To activate another row in the same bank we need to wait for the first row to close but this is not necessary if the row is in another bank. This brings us to tRRD, the minimum time interval between two consecutive ACTIVE commands on different banks. In the case of cryptonight where random access is important, the hypothesis is that tRRD will have a great impact in performance. Accessing the next row for reading without having to wait longer directly affects throughput.

We can see the default values range from 4 up to 8 for the fastest clock. Custom bioses with custom timings for each tRRD values are tested at 1650 Mhz.

Strap	TRRD
1000	4
1125	5
1250	5
1375	5
1425	6
1500	6
1625	7
1750	7
2000	8

Table 3-2

We can see that tRRD has a great impact on performance and lowering it gives better results. At 1650 clock speed we can achieve the highest value of 705 h/s with a tRRD of 4 but when testing this value at 2000 Mhz, the card fails to operate in a stable normal way so we can rule out this value and use tRRD of 5 as a tight enough but still stable value.

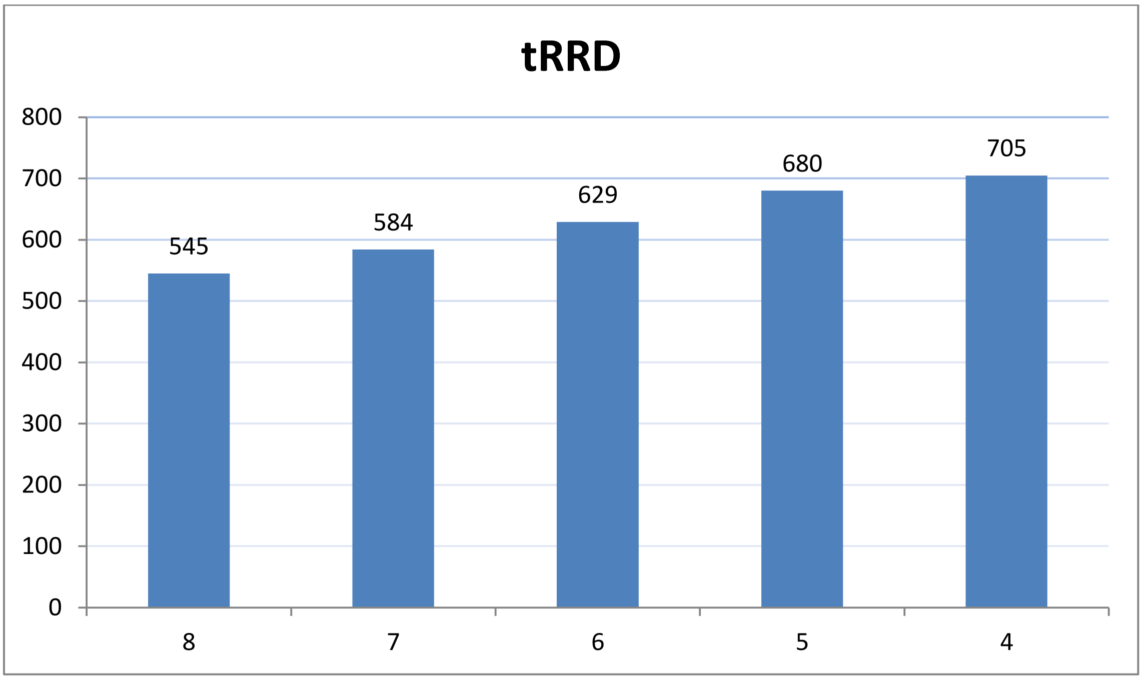


Figure 3-14

3.2.3 tR2W & tW2R

An interesting pair to watch was the tR2W (read to write turn) and tW2R (write to read turn). These have unusual values on the default straps. tR2W increased to 25 until 1500 Mhz back to 24 at 1625 Mhz, 25 at 1750 Mhz and back again to 24 at 2000 Mhz. Unusual compared to the linear increase seen on other values.

Strap	tR2W	tW2R
1000	19	14
1125	20	15
1250	22	15
1375	24	17
1425	24	17
1500	25	17
1625	24	19
1750	25	19
2000	24	21

Table 3-3

Testing different pairs revealed that a tR2W of 24 was most stable as decreasing it too much produced lower results while a lower tW2R down to 17 or 19 only resulted in a 1% increase.

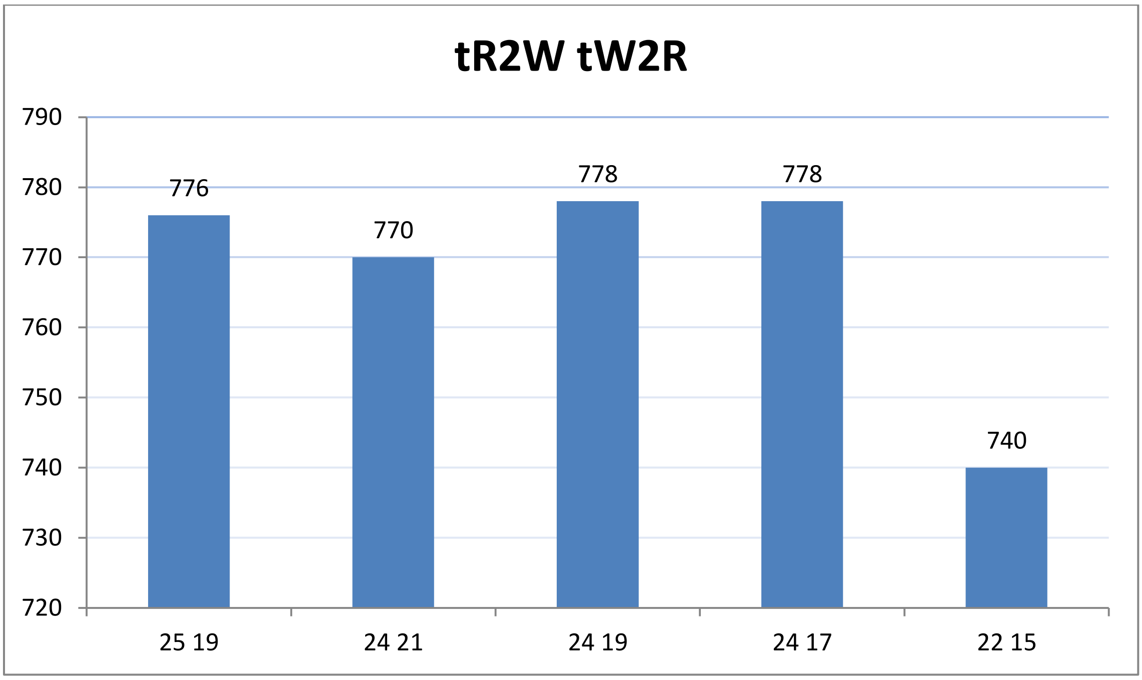


Figure 3-15

3.2.4 tCRCWL & TCRCRL

tCRCWl is CRC Write Latency.

tCRCRL is CRC Read Latency.

The EDC or Error Detection Code [25 p80] is an error detection mechanism provided in GDDR5 to improve system reliability. A checksum is generated for both read and write data. Result is returned to the controller who decides if an error happened and whether to retry the command. This data is the returned CRC.

The total EDC latency depends on CAS latency. The exact formulas are:

EDC Read Latency tEDCRL = tCL + tCRCWL

EDC Write Latency tEDCWL = tWL + tCRCWL

Strap	TCRCWL	TCRCRL
1000	2	1
1125	5	2
1250	5	5
1375	6	2
1425	6	2
1500	6	2
1625	6	2
1750	6	2
2000	6	2

Table 3-4

The expected impact of these values was minimal since the EDC has a dedicated transfer pin. The results however show that lowering the latency when operating at 2000 MHz to values similar to lowest straps can increase hashrate by about 2.3%

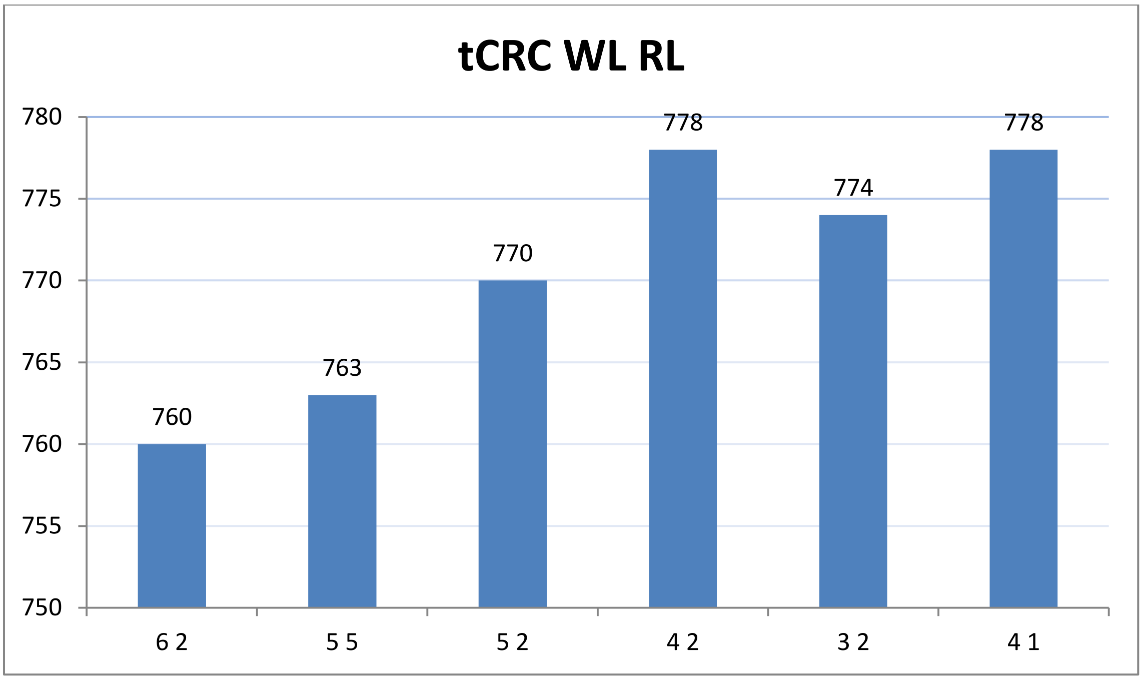


Figure 3-16

3.2.5 tRC

tRC is the delay between ACTIVE to ACTIVE commands.

Whenever a row has been opened by an ACTIVE command, another row in the same bank cannot be opened until the first one is closed (precharged). The time between these two ACTIVE commands in the same bank is defined as tRC. Similar to tRRD, tRC also affects delay between read commands, potentially affecting hashrate.

The table below shows the data extracted from the default bios. The table also includes ACT and RASMACT since tRC value seems to be equal to ACT+RASMACT+1.

In most documentation [26] it's stated that $T_{RC} = T_{RAS} + T_{RP.}$

tRAS is the time between opening (ACTIVE command) and closing (precharging) a row. tRAS on its own is equal to tRCD+tCL where RCD is minimum time between opening the row and accessing columns within it (Row address to Column Address Delay) and CL is the number of cycles between sending the column address and the result data to be available. During tRCD the row signal settles enough for the charge sensor to amplify it.

tRP or row precharge time, is the time it takes between the precharge command (closing) and the next ACTIVE command. During this time the sense amps charge and the bank is activated [26].

In the decoded data these actions are represented by ACT which stands for ACTIVE and RASMACT which is the RAS to ACTIVE time. Both these timings have different values for READ and WRITE commands.

The table below contains the decoded values from the default bios for most straps.

Strap	TRC	ACT R	ACT W	RASMACT R	RASMACT W
1000	41	14	10	28	32
1125	47	16	12	32	36
1250	52	18	13	35	40
1375	57	20	15	38	43
1425	59	21	15	39	45
1500	61	21	15	41	47
1625	68	24	17	45	52
1750	72	25	18	48	55
2000	83	28	20	56	64

Table 3-5

To test the performance of these values, 5 different combinations were tested at a constant memory clock.

- V1 is the 1425 strap.
- V2 is the 1375 Mhz strap.
- V3, V4, V5 is the 1500 Mhz strap with tighter tRC, slightly less than the formula suggests.

Strap	TRC	ACT R	ACT W	RASMACT R	RASMACT W
V1	59	21	15	39	45
V2	57	20	15	38	43
V3	57	21	15	41	47
V4	55	21	15	41	47
V5	52	21	15	41	47

Table 3-6

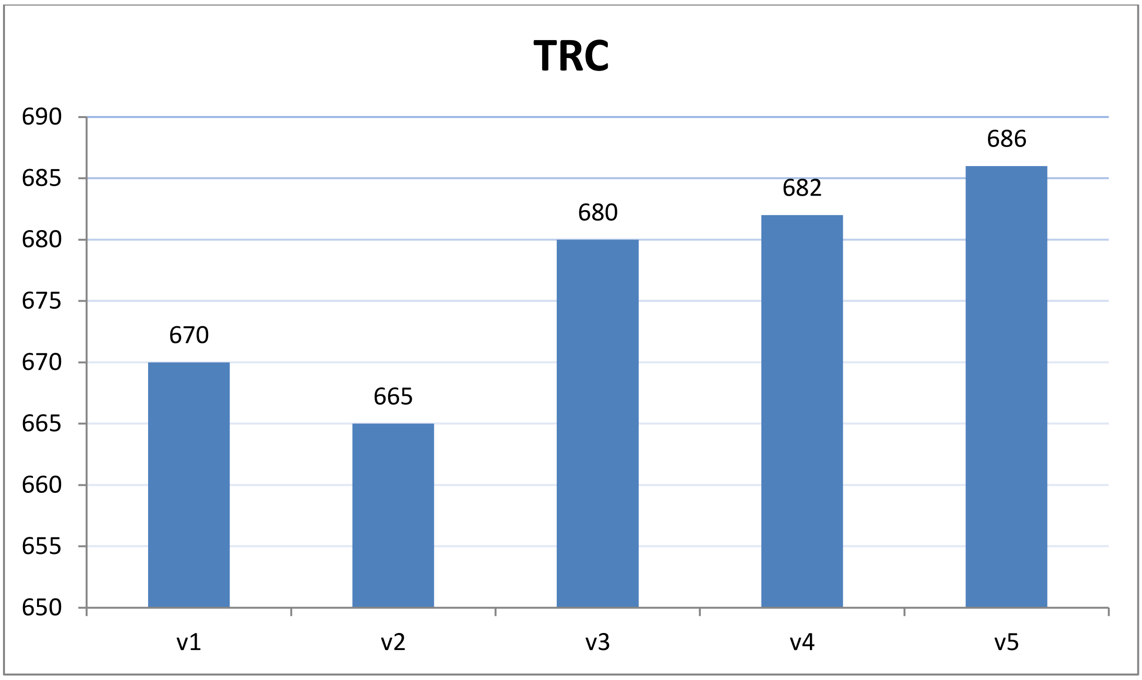


Figure 3-17

Using the new custom memory timing in the 1625, 1700 and 2000 MHz timings results in an even bigger hashrate and efficiency improvement. Figure 3-18 show the new hashrate in red compared to the one achieved with the default 1500 MHz timings in blue.

- At 1650 MHz: from 612 to 693, a 13.2% increase
- At 1750 MHz: from 640 to 763, a 19.2% increase
- At 2000 MHz: from 724 to 844, a 16.6% increase

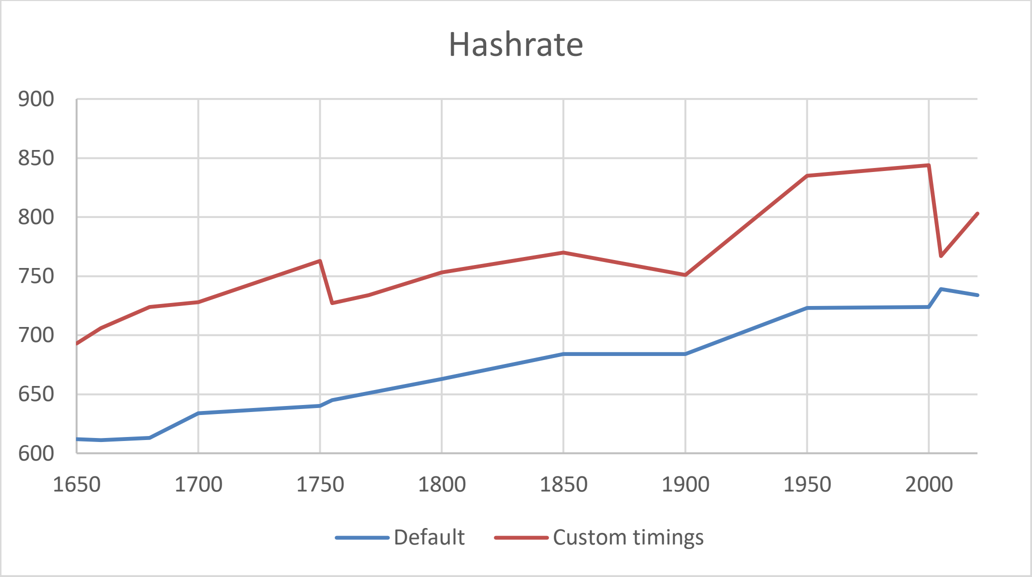


Figure 3-18

A similar trend can be seen in increased efficiency in Figure 3-19.

- At 1650 MHz: from 5.02 to 5.68, a 13.1% increase
- At 1750 MHz: from 5.04 to 6 a 19% increase
- At 2000 MHz: from 5.28 to 6.12, a 16% increase

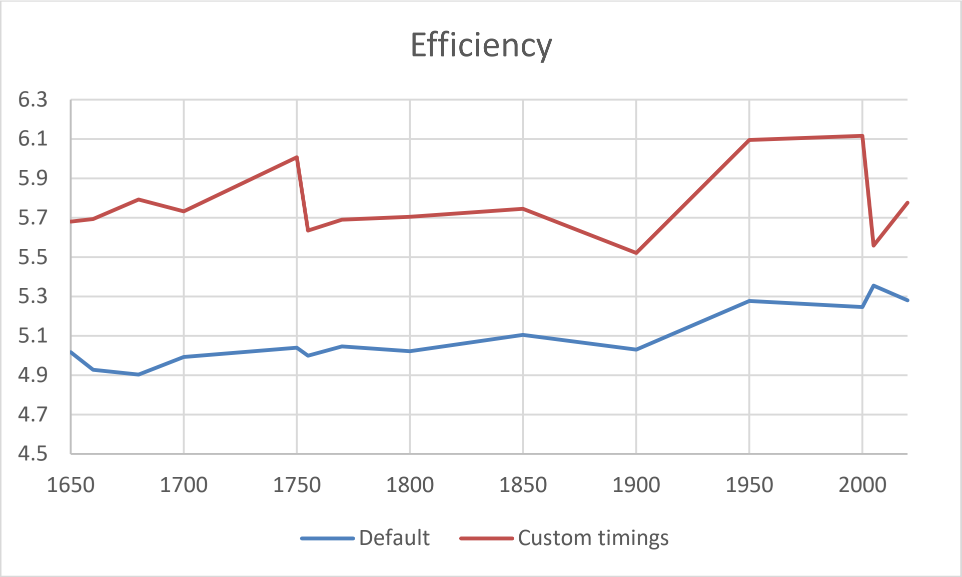


Figure 3-19

Putting all these changes together in the same card we get stable results over 860 h/s compared to the default 546 h/s.

This results in an improvement of around 57.5% in hashrate.

3.3 Undervolting

Another technique used in increasing efficiency is undervolting. Undervolting reduces the voltage given to the graphics card. Whenever we overclock the card, more energy is required and thus increasing voltage is needed to sustain normal work. Once a stable core and memory clock is found a step-by-step reduction of the voltage is applied to find the least amount of energy needed to operate correctly. Reducing the voltage too much will cause the card to stop working and at that point, a lower limit has been reached.

The benefits of undervolting include lower power consumption thus reducing electrical costs. Reduced heat produced from the chips, which leads to longer card longevity and lower fan usage, reducing electrical consumption and noise.

A 7990 card with a default 1000 MHz core clock and 1500 Mhz memory clock is given 1200 mv and reaches 90° performing cryptonight pow. Reducing the voltage by 10 millivolt at a time shows that we can go down to 1030 mv and still have a normal operation producing a constant hashrate. Doing this also reduces the temperature down to 74°. This is a significant improvement to power usage, heat output and noise. Note that results depend on the mining algorithm since different mining algorithms make use of core calculations and memory access in different ratios.

Undervolting can also be used to reduce power consumption at the expense of hashrate produced in specific situations where energy is limited. Reducing the core clock of a 7990 from 1000 MHz to 830 MHz reduces to hashrate output from 560 h/s to 500 h/s and allows for an undervolt down to 900 mv. A 10% reduction in voltage results in a similar 12% in hashrate.

3.4 Asic development and future outlook

Technology moves at a fast pace and during the writing of this paper new graphic cards have been introduced to the market. These newer versions have a base performance of more than twice of the cards that were used to do the testing in previous sections. These new more efficient cards can outperform the older ones and make them obsolete from a profit point of view but it does not mean that the same process of optimizing efficiency cannot be applied to these new cards. Which brings us back again to the purpose of this paper, aiming for the best efficiency of general purpose cards.

The Monero community has long been a firm believer of ASIC resistance. The idea to have an algorithm that best performs on general-purpose hardware easily accessible to the masses, not because ASICs are inherently evil but because of their current state of distribution. Cryptocurrencies at an early stage are vulnerable to a theoretical hostile ASIC producer who has a monopoly in the production of the hardware. Some argue that this scenario is rare since this actor is economically incentivized to play along with the majority of the consensus otherwise the community will fork the coin and all the investment in research, development and mined coins will go to zero. To protect against these scenarios the community is currently in consensus to fork the mining algorithm if necessary. In the near future, where mining reaches a larger adoption phase in the hardware industry more producers will join the competition and a free market develops where multiple suppliers are available, there is no reason to be against specialized hardware that improves efficiency. Multi suppliers of mining hardware increases the decentralization of mining power thus reducing risks of hostile miners or other political forces having majority in the network.

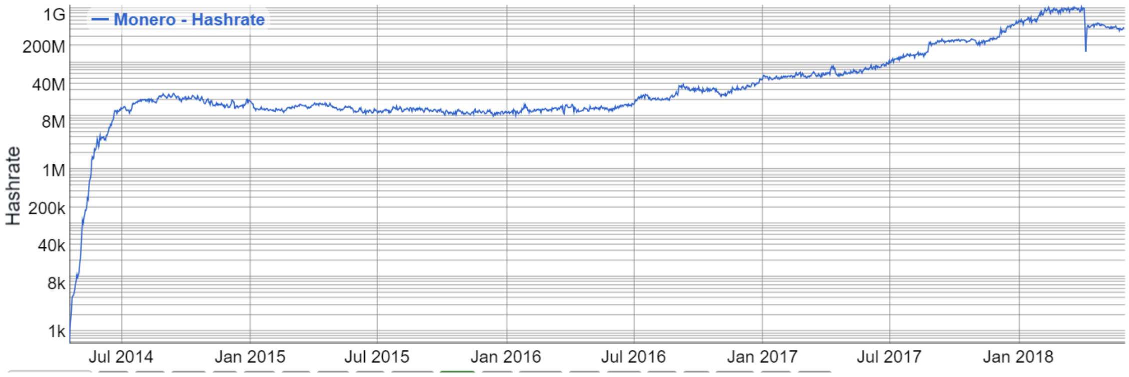


Figure 3-20

Monero hashrate since creation, log chart

In February 2018 rumors spread of the development of the first cryptonight ASIC. Until this point most mining was done with CPUs and GPUs and the rise in total network hashrate (Figure 3-20) was mostly attributed to new GPUs coming to market and new investments in mining led by the appreciation in price of the Monero coin which lead to higher profits. Some speculated that the rise in hashrate was due the first ASICs being developed. These rumors where finally confirmed in March when the company Baikal (source) and Bitmain (course) introduced the Baikal giant N and Bitmain Antminer X3 ASIC miners. The Giant N with a hashrate of 20Kh/s +/- 10% consuming 60W +/- 5% and the AntMiner X3 doing 220 kh/s using 550W. These machines perform 40 times better than the most efficient GPUs.

In the middle of the GPU-ASIC battle, a small part of the community started to experiment with FPGA boards. Being programmable, they allow for more flexibility in case the POW algorithm changes periodically. Hardware availability and the programming skill required kept this kind of mining specialized and not very popular. Their efficiency is believed to be better than GPUs and less than ASICs but no public information on their performance is available yet. Due to their competitive nature, these mining operators have not disclosed performance reports. Programmable hardware might be the future of mining and its worth to keep an eye on.

The development of such efficient hardware is truly remarkable and deserves and applause but this doesn’t mean that GPUs are now obsolete. As discussed above the social consensus of forking to achieve temporary ASIC resistance has lead the community to change the cryptonight algorithm to make these ASICs unusable to mine Monero. The fork happened at height 1,539,500 and from that point these machines cannot be used to mine Monero but can still be used to mine other coins that will still use the original cryptonight as the mining algorithm. This is a perfect example displaying the usefulness of general purpose graphic cards that have the ability to adapt to new algorithms or completely change to other mining algorithms.

4. CONCLUSIONS

During the writing of this paper a lot has changed. New GPUs with more than double the performance of old versions were announced. New motherboards with four times the amount of PCI-E lanes. Monero ASICs were developed and the community forked to remain resistant. This is a very fast evolving space and although old cards may be less competitive now, the exact numbers don't matter. What matters is the % change in hashrate or efficiency that we are able to add on top of that base line. The methodology and experiments done in this paper apply to new versions also. Even if a specific coin like Monero ends up accepting ASICs in the future, there will always be new mining algorithms that will be profitably minable with GPUs.

In this paper we saw how the choice of hardware affects the performance of miners. How to choose hardware and maintain it for a long time without burning everything.

Techniques like overclocking do improve hashrate slightly but at the cost of energy consumption. This results in lower efficiency but might be desirable if electricity costs are negligible.

Most importantly, we demonstrated how by understanding CryptoNight memory heavy property we could leverage memory timings modifications to increase efficiency by up to 57% more than stock settings.

REFERENCES

[1] A. Back, A partial hash collision based postage scheme, 1997, Available: http://www.hashcash.org/papers/announce.txt

[2] A. Werner, K. Sugihara, Montag, Ardolabar, Tereno, A.M Juarez, CryptoNote Transactions, Cryptonote Standard 4, 2012, Available: https://cryptonote.org/cns/cns004.txt

[3] A. Werner, M. Pliskov, Montag, CryptoNote Difficulty Adjustment, Cryptonote Standard 10, 2014, Available: https://cryptonote.org/cns/cns010.txt

[4] A. Werner, Montag, Ardolabar, Tereno, A.M Juarez, CryptoNote Blockchain , Cryptonote Standard 3, 2012, Available: https://cryptonote.org/cns/cns003.txt

[5] A.M Juarez , A. Werner, Neocortex, O. Norton, CryptoNote Keys and Addresses , Cryptonote Standard 7, 2012, Available: https://cryptonote.org/cns/cns007.txt

[6] B. Hawking, Pacific_skyline, Yggdrasil, J, CryptoNote Technology, Cryptonote Standard 9, 2013, Available: https://cryptonote.org/cns/cns009.txt

[7] C. Dwork, M. Naor, Pricing via Processing, 1993, Lecture Notes in Computer Science No. 740. Springer: 139–147

[8] Dave, Minting Money with Monero ... and CPU vector intrinsics , 2014, Available: https://da-data.blogspot.com/2014/08/minting-money-with-monero-and-cpu.html

[9] G. Maxwell, Confidential Transactions, Available: https://people.xiph.org/~greg/confidential_values.txt

[10] J. Desjardins, The properties of money, 2015, Available: http://money.visualcapitalist.com/infographic-the-properties-of-money/

[11] J.C Perraux, A Fisher, Serial ATA power connector pinout and connections, 2013, Available: http://pinouts.ru/Power/sata-power_pinout.shtml

[12] Luigi1111, Understanding Monero Cryptography Privacy, 2016, Available: https://steemit.com/monero/@luigi1111/understanding-monero-cryptography-privacy-part-2-stealth-addresses

[13] N. Ralph, Advanced Tonga Bios Editing, 2016, Available: http://nerdralph.blogspot.com/2016/09/advanced-tonga-bios-editing.html

[14] Nicolas van Saberhagen, Cryptonote v2.0, 2013, Available: https://cryptonote.org/whitepaper.pdf

[15] P. Wuille, Hierarchical Deterministic Wallets, 2012, Available: https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki

[16] Phaedrus2129, On PSU Efficiency, 2011, Available: https://hardforum.com/threads/on-psu-efficiency.1575419/

[17] S. Noether, Ring Multiginature, 2016, Available: https://web.archive.org/web/20161023005318/https:/shnoe.wordpress.com/2016/03/22/ring-multisignature/

[18] Schoenborn, Zale, Board Design Guidelines for PCI Express Architecture, 2004, 19-22, Available: http://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/639/7851.PCIe_5F00_designGuides.pdf

[19] Seigen, M. Jameson, T. Nieminen, A.M Juarez , Neocortex, CryptoNight Hash Function, Cryptonote Standard 8, 2013, Available: https://cryptonote.org/cns/cns008.txt

[20] Stilt, Lard, Tahiti Memory Timings, 2015, Available: http://www.overclock.net/forum/67-amd-ati/1554360-tahiti-memory-timings-patch-hynix-vram.html

[21] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008, Available: https://bitcoin.org/bitcoin.pdf

[22] S. Noether, A. Mackenzie, Monero Core Team, Ring Confidential Transactions, 2016, Avaialble: https://lab.getmonero.org/pubs/MRL-0005.pdf

[23] TiN, Extreme Overclocking, 2010, Available: http://forums.xtremelabs.org/viewtopic.php?f=18&t=827

[24] W. Peter, B. Bunz, J. Bootle, D. Boneh, A. Poelstra, G. Maxwell, Bulletproofs: Short Proofs for Confidential Transactions and More, 2017, Available: https://web.stanford.edu/~buenz/pubs/bulletproofs.pdf

[25] JEDEC STANDARD, Graphics double data rate, JEDEC Solid State Technology Association, 2016, Available: https://www.jedec.org/standards-documents/docs/jesd212c

[26] W1zzard, tRAS, tRCD, tRP, tRC, 2004, Available: https://www.techpowerup.com/articles/overclocking/64