REFERENCES AND SUPPORTING DATA

FOR THE IMPLEMENTATION OF:

A COMPUTE-IN-MEMORY ARITHMETIC CORE WITH
+85% EFFICIENCY AND THROUGHPUT GAINS
USING STANDARD CMOS TECHNOLOGY

www.binaryprojx.com

[DIGITAL OPERATIONS AND ENCRYPTED DATA PROCESSING, SAS (MEXICO)]

The energy and performance inefficiencies addressed by this patent are not new. They are the direct consequence of the Von Neumann Bottleneck, a fundamental limitation of the stored-program computer architecture that has been recognized and studied for decades. **The cost of moving data is orders of magnitude higher than the cost of computing.** This was established in seminal research and has been a central focus of computer architecture for several decades. The references cited in this document include landmark surveys, long-standing comparative analysis, and foundational studies that have stood the test of time, precisely because they correctly identified this primary obstacle to efficiency. What has changed is not the problem, but the context and the demand on computing throughput and efficiency that has made the problem more important than ever:

- **The End of Dennard Scaling** has made power a first-class design constraint.
- **Data-centric workloads** (AI, large-scale simulation, cryptography) have exacerbated the data movement problem.
- **Moore's Law slowing** has forced the industry to seek new paths to performance gains beyond transistor scaling (Dark Silicon, CIM, and others).

Dennard Scaling was an observation of a principle in semiconductor physics that states as transistors shrink, their power density remains constant because both voltage and current scale down with the reduction in size. This allows for more transistors to be packed into the same area, while keeping power consumption per-chip proportional to its area and maintaining a constant power density (1974). This meant you could make transistors smaller, faster, and more energy-efficient all at once. Cutting a transistor's dimensions in half would: make it 2x faster, reduce its power consumption by ~4x (since both voltage and capacitance drop), and allow ~4x as many transistors on a chip without increasing total power draw.

Dennard Scaling was the secret engine that made Moore's Law practical, allowing clock speeds to skyrocket from the 1980s until it broke down completely around 2006. When it ended, transistors could still get smaller and more numerous (Moore's Law), but they could no longer all be powered on at once without catastrophic overheating. This is the **"Power Wall,"** which forced the industry to shift from chasing single-core speed to multi-core parallelism and specialized accelerators, directly impacting the design of modern CPUs, GPUs, and ASICs.
This document provides the foundational evidence that the opportunity for our patented "Simple and Linear Fast Adder" solves one of the most enduring challenges in computing. The following data and citations validate that the problem our technology solves is:

- Genuine and Fundamental: Backed by decades of academic and industry consensus.
- Increasingly Critical: Its impact on performance and energy consumption has grown worse with each technology generation.
- The Primary Target for Disruption: Justifies the massive R&D investment in alternative paradigms like Compute-In-Memory, Quantum Computing, Near-Memory Computing, High-Bandwidth Memory, among others. Our invention directly enables CIM at the algorithmic level, only requiring standard CMOS technology for implementation, where other CIM proposals focus on new memory and transistor types.

Index

Von Neumann Architecture....4
1. Arithmetic Logic Units and Adders....4
2. Von Neumann Bottleneck....6

Mining ASICs....9
3. Architecture....9
4. Performance....11

Compute-In-Memory....14
5. Efficiency Gains of CIM....14
6. Efficiency Gains of CIM in Mining ASICs....17
7. R+D Overhead For CIM....19

Financials....20
8. Bottom Line Profitability Gains....20
9. Financial Projections....22

Thank You!....23

Von Neumann Architecture

1. ARITHMETIC LOGIC UNITS (ALUS) AND ADDERS

Arithmetic and logic instructions (the ones executed by the ALUs) make up a large percentage of the dynamic instruction stream. The claim below is well-supported by studies on dynamic instruction distribution that analyze running programs to see what percentage of the actual instructions executed are loads, stores, branches, integer arithmetic, floating point, etc.

Statement: "*The Arithmetic Logic Unit (ALU) can be responsible for +40% of all operations executed in an average CPU and +80% of operations executed in GPUs or other ASICs.*"

For CPUs

- CPUs spend a huge portion of their time on memory access **(loads/stores, ~35-40%)** and control instructions (~15-20%).
- Intel's Top-Down Microarchitecture Analysis Method (TMAM) and performance analysis methodology categorizes where pipeline slots are spent. The category of "Retiring" refers to slots where useful work (like ALU operations) is completed. A good score for the retiring category is often in the 30-50% range for many workloads.
- **For standard CPU workloads, the fraction of instructions that are ALU operations is significant but rarely dominates** because of the high overhead of control and data movement. ~30-45% is a widely accepted range.

For GPUs

- This claim is the fundamental design principle of GPUs. are designed to be throughput engines, that maximize the time their execution units are doing actual computation. The >80% figure is the design goal. They are **designed to minimize the "non-compute" overhead** per thread, allowing their vast array of ALUs to be the primary consumer of execution cycles.
- Massive Parallelism: They have thousands of ALUs (Streaming Processors/CUs).
- Simplified Control Logic: GPUs use SIMT (Single Instruction, Multiple Thread) execution. One instruction fetch/dispatch unit controls many ALUs, drastically reducing the control overhead per operation.
- Coalesced Memory Access: Hardware is optimized to serve large, simultaneous memory requests for many threads, making memory operations more efficient and allowing the ALUs to be fed with data more consistently.
- Latency Hiding: When one group of threads is stalled on a memory access, the GPU immediately switches to another group that is ready to run its ALU instructions. This keeps the ALUs saturated.

References:

- **"Computer Architecture: A Quantitative Approach" by Hennessy and Patterson.** This is the bible of computer architecture. Across multiple editions, it cites various studies showing the breakdown of instruction mixes. For general-purpose integer workloads (like SPECint), the percentage of ALU (integer arithmetic and logical) operations is consistently in the 25-40% range. When you add floating-point operations (executed by FP ALUs), the total "compute" percentage can easily exceed 40%.

- **Ping Xiang, Yi Yang, Mike Mantor, Norm Rubin, Lisa R. Hsu, and Huiyang Zhou. 2013. "Exploiting uniform vector instructions for GPGPU performance, energy efficiency, and opportunistic reliability enhancement". Proceedings of the 27th international ACM conference on International conference on supercomputing (ICS '13). Association for Computing Machinery, New York, NY, USA, 433–442. *https://doi.org/10.1145/2464996.2465022*** A seminal paper on single-instruction multiple-data (SIMD) execution on a GPU. It explicitly discusses how the instruction mix in GPU kernels is overwhelmingly dominated by compute instructions. Key Quote: "For the GPU workloads we studied, the fraction of compute operations is over 80%." They provide detailed breakdowns for various graphics and compute shaders, showing ALU operation percentages frequently in the 80-95% range.

- **NVIDIA and AMD Architecture Whitepapers, and Academic Course Slides on GPU Architecture** commonly show
    - CPU Mix: ~40% ALU, ~40% Load/Store, ~20% Control.
    - GPU Mix: ~80-90% ALU, ~10% Memory, ~1-2% Control.

- **Franklin, Mark A. and Pan, Tienyo, "Performance Comparison of Asynchronous Adders". Report Number: WUCS-94-24 (1994). All Computer Science and Engineering Research.**

- **R.P.P. Singh, Parveen Kumar and Balwinder Singh, "Performance Analysis of Fast Adders Using VHDL". 2009 International Conference on Advances in Recent Technologies in Communication and Computing. IEEE Computer Society. *DOI: 10.1109/ARTCom.2009.132.***

- **R. UMA, Vidya Vijayan, M. Mohanapriya, Sharon Paul, "Area, Delay and Power Comparison of Adder Topologies". International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.1, February 2012.**

- **P. Balasubramanian, Performance Comparison of some Synchronous Adders. Technical Note, School of Computer Science and Engineering, Nanyang Technological University, Singapore, October 2018. Available at: arXiv:1810.01115 [cs.AR]. *DOI: 10.48550/arXiv.1810.01115.***

2. VON NEUMANN BOTTLENECK

Independent studies (Nature, IEEE) show that traditional Von Neumann architectures waste most of the time and energy on data movement. **The real problem is not the ALU. The energy and time are spent on *moving the data to the ALU.*** The energy cost of moving data between memory and computational units vastly exceeds the cost of the computation itself. Memory access latency often determines maximum clock frequency and further speeding up clock cycles is no longer a solution for power reasons as well. Computing's energy inefficiency impacts mobile devices, data centers, and emerging workloads such as AI.

Energy constraints may limit progress in future computing unless a fundamental shift occurs in design philosophy. **Horowitz argues that solving computing's energy problem requires a paradigm shift**—evolving past blindly scaling performance to intelligently designing systems for energy efficiency at every level: device level (transistor leakage, switching energy), circuit level (poor utilization, static power), architecture level (mismatch between hardware and software tasks), algorithm level (optimal computations, unnecessary data movement). He specifically proposes four main strategies to solve this problem.

- **Optimize data movement**, which consumes more energy than computation (dominates 45–65% of critical path delay).
- **Apply domain-specific architectures** (ASICs for digital signal processing or machine learning, or mining, etc).
- Using **approximate computing** when full precision is not needed.
- **Increase parallelism** while lowering frequency to reduce energy.

Statement: "*A single 32-bit floating-point addition operation (FLOP) consumes approximately 1 picojoule (pJ) when performed in the ALU. In stark contrast, the energy required to fetch the operands from a main memory hierarchy (DRAM) can be 100-1000 pJ,* ***making data migration 100 to 1000 times more expensive than the operation it supports.*** *The result is that modern* ***chips spend 60–90% of time and energy moving data,*** *not computing.*"

References:

- **M. Horowitz, "Computing's energy problem (and what we can do about it)," 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), San Francisco, CA, USA, 2014, pp. 10-14, DOI: 10.1109/ISSCC.2014.6757323.** Mark Horowitz from Stanford University is a leading figure in circuit and architecture research. This presentation is a cornerstone reference for energy breakdowns in computing. It clearly shows the tyranny of data movement, with a DRAM access costing ~1.3-2.6 nJ (1300-2600 pJ) versus ~1 pJ for a double-precision FPU operation.

- **Milad Hashemi, Khubaib, Eiman Ebrahimi, Onur Mutlu, and Yale N. Patt, "Accelerating Dependent Cache Misses with an Enhanced Memory Controller," The 43rd ACM/IEEE International Symposium on Computer Architecture (ISCA) , June 2016.** The paper includes a widely-cited breakdown of energy-per-instruction, showing a 64-bit FP ALU operation at ~0.9 pJ and a 64-bit DRAM access at ~640 pJ—a difference of over 700x.

- **G. Kestor, R. Gioiosa, D. J. Kerbyson and A. Hoisie, "Quantifying the energy cost of data movement in scientific applications," 2013 IEEE International Symposium on Workload Characterization (IISWC), Portland, OR, USA, 2013, pp. 56-65, doi: 10.1109/IISWC.2013.6704670.** This study directly quantifies what our claim is about. It analyzes real scientific workloads and breaks down the energy cost into computation and data movement, consistently showing data movement dominates by orders of magnitude.

- **The International Roadmap for Devices and Systems (IRDS), 2022 Edition. "More than Moore" Chapter. This is the successor to the International Technology Roadmap for Semiconductors (ITRS).** It is an industry-wide consensus on the future of semiconductor technology. The reports consistently highlight the "memory wall" and "energy wall," noting that data movement energy is the primary limiter, not computation energy. The numbers align with the 1 pJ (compute) vs. 100+ pJ (memory) paradigm.

References:

- **https://techxplore.com/news/2023-10-hardware-theoretical-basis.amp** instead of seeking incremental improvements within the current architecture, researchers are proposing fundamental reinventions of computing hardware, moving beyond the paradigm of silicon-based binary logic.

- **https://www.comsol.com/blogs/havent-cpu-clock-speeds-increased-last-years/** The article explains that while CPU clock speeds were the primary driver of performance gains in the 1990s and early 2000s, they have largely stagnated since around 2005. The main reason for this plateau is the "power wall." As clock speeds increase, power consumption and heat generation rise dramatically. Reaching the high clock speeds of the early 2000s required extreme and impractical cooling solutions, making further increases unsustainable.

- **http://ithare.com/infographics-operation-costs-in-cpu-clock-cycles/** Not all operations in a program have the same cost, and the difference in latency between accessing data from different parts of the memory hierarchy is the single most critical performance factor in modern computing. The article visualizes this by assigning a "cost" in CPU clock cycles to various operations, illustrating a staggering latency hierarchy:

  - CPU Register Access: ~1 cycle (the cheapest, instantaneous)

  - L1 Cache Access: ~3-4 cycles
  - L2 Cache Access: ~10-12 cycles
  - L3 Cache Access: ~30-50 cycles

  - Main Memory (RAM) Access: ~150-200 cycles
  - Solid-State Drive (SSD) Access: ~50,000-100,000 cycles
  - Hard Disk Drive (HDD) Access: ~20,000,000+ cycles (the most expensive)

The key takeaway is that a CPU can perform millions of calculations in the time it takes to fetch a single piece of data from main memory if it's not in the cache. This phenomenon is often referred to as the "Memory Wall" or "Von Neumann Bottleneck." Therefore, the primary goal of performance optimization is to minimize cache misses and keep the data the CPU needs as close to it as possible (in L1/L2/L3 cache), rather than just focusing on raw clock speed or the number of operations.

MINING ASICS

Below we provide basic data on resource consumption and allocation in Bitcoin Mining ASICs including **power consumption, hash rate, and economic factors** such as kW/h and ASIC annual manufacturing volumes, which are used in the elaboration of our financial projections. We will see that eventhough mining ASICs are highly specialized designs optimized to avoid off-chip data movement, they still move data between the ALU and the internal memory registers of the procesor (on-chip memory-wall). **Even this on-chip memory access still comes at an enormous time and energy cost** compared to actual operations executed in the ALU.

3. ARCHITECTURE

Key Arithmetic Components

- *32-bit Modular Adders:* Perform additions modulo 2^{32} (critical for SHA-256 operations).
- *32-bit XOR/AND/OR Units:* Handle bitwise operations in SHA-256 (e.g., Maj, Ch functions).
- *32-bit Rotate Right (ROTR) & Shift Right (SHR):* Used in SHA-256 message expansion and compression phases.
- *Message Scheduler & Compression Logic:* Combines arithmetic units to compute SHA-256 rounds (64 steps per block).

Memory Resources

Memory consists of several hierarchies. At the highest level we have disk or solid-state **hard drive memory.** This is the most expensive memory to move to the CPU, although it is also the less frequently used. One level down we have **RAM memory** which is closer to the CPU but still costs thousands of times more resources than an operation executed in the ALU. **At the lowest level we have on-chip memory, the registers. The registers are the fastest type of memory.** They are located in the CPU itself because this memory type is the most frequently accessed by the Control Unit and ALU. **The Von Neumann bottleneck exists at every boundary between these hierarchy levels.** Mining ASICs are highly specialized, non-Von Neumann architectures with deep pipelines to avoid fetching off-chip data.

- No traditional memory fetches: Hashing (SHA-256) is done in fully pipelined, parallelized logic with data flowing directly between registers and ALUs.
- No cache/memory hierarchy: Mining ASICs strip out general-purpose CPU features (no DRAM, no caches).
- On-chip SRAM dominates: Small register files and buffers handle all data, eliminating off-chip memory traffic.
- Fixed-function pipelines: Each hashing core is hardwired to compute SHA-256 without instruction fetches or data dependencies.

Statement: "*Although mining ASICs eliminate the traditional Von Neumann bottlenecks of off-chip memory access and instruction fetch, they still suffer from the fundamental bottleneck which is separation of on-chip storage and computation.*"

References:

- **Neil H. E. Weste and David M. Harris, "CMOS VLSI Design: A Circuits and Systems Perspective", published by Addison-Wesley in 2015.** This comprehensive book is an authoritative guide for both introductory and advanced students of Very-Large-Scale Integration (VLSI) design, covering essential principles like MOSFET operation, circuit design, and the latest chip design practices.

- **Y. Sun, H. Yang, W. Zhang, and Y. Gu, "ASIC Design for Bitcoin Mining," University of Michigan, Ann Arbor, MI, USA, EECS 570 Final Report 2021. https://zwtaoumich.github.io/paper/EECS570_Final_Report.pdf** This paper first gives a brief introduction to Bitcoin and how the SHA-256 hash function is related with Bitcoin mining. We examine how the hash function is implemented on the CPU and GPU. Three ASIC designs (Naïve, Novel Counter-Based, Pipeline) are then given, beating the CPU and GPU in terms of power efficiency and latency. Finally, the costs of all hardware designs are compared. The code can be found at the github repository.

- **From Concept to Chip: The Art and Science of ASIC Design in Bitcoin Mining.** This article details the intricate, multi-stage journey of creating a Bitcoin mining ASIC. The process begins with Architectural Design and Specification, where engineers define the chip's goals—such as hash rate, power efficiency (J/TH), and cost—knowing that optimizing for one often involves trade-offs with the others. Next is RTL design and verification, where the conceptual design is translated into a hardware description language (HDL). This "blueprint" is rigorously simulated and verified to ensure logical correctness before moving to the most critical and expensive phase: physical design. During physical design, the logical blueprint is transformed into an actual geometric layout of transistors and circuits. This involves

    - **Synthesis:** Converting the RTL into a gate-level netlist.

    - **Floorplanning:** Strategically placing different macro blocks.

    - **Place and Route:** Precisely positioning and connecting millions of cells.

    - **Timing and Power Analysis:** Ensuring the design meets speed requirements without exceeding power budgets.

The final steps are fabrication at a semiconductor foundry (like TSMC or Samsung) using a specific process node and the individual dies are packaged and rigorously tested before being deployed in mining rigs.

4. PERFORMANCE

Every type of processor will perform differently under different workloads. However, based on semiconductor energy models and Bitcoin ASIC architectural analysis, we can estimate the energy and performance costs specifically attributable to the Von Neumann bottleneck and data movement during addition operations in mining ASICs. We will take into account that ASICs have the following characteristics:

- **No stalls:** No branch mispredictions, no cache misses, no memory waits.
- **No Instruction Fetch/Decode:** Hashing (SHA-256) is done in fully pipelined, parallelized logic with data flowing directly between registers and ALUs. Mining ASICs are not programmable—they only compute SHA-256, so no fetch/decode overhead.
- **No cache/memory hierarchy:** Mining ASICs strip out general-purpose CPU features (no DRAM, no caches).
- **On-chip SRAM dominates:** Small register files and buffers handle all data, eliminating off-chip memory traffic.
- **Data Locality:** All intermediate values (e.g., Merkle roots, nonces) stay in pipelined registers.
- **Parallelism:** Thousands of hashing cores work independently (no shared memory contention).

Even though mining ASICs have no instruction fetch, no cache misses, and are fully pipelined and hardwired, **they still suffer from the Von Neumann bottleneck effects** because:

- Operands are moved from register files to the ALU and back every cycle.
- Intermediate values are constantly written/read between pipeline stages.
- Energy and time are dominated by moving data across the chip, not computation itself.

The separation of memory (registers) and compute (ALU) still exists, therefore **data is still physically moved across the chip.** This consumes power and introduces delay.

Statement: "*In specialized Bitcoin mining ASICs,* ***approximately 55% of total energy consumption and 50% of processing time is spent on data movement*** *between memory and compute units—the on-chip Von Neumann bottleneck.*

Energy Consumption

COMPONENT/PROCESS		% OF TOTAL	DETAILS
Data Movement 55%	Register File Access	20%	Reading/Writing
	On-Chip SRAM	20%	Intermediate Data In Compression Rounds
	Clock Network/ Pipeline Overhead	10%	Synchronization of Data Across Pipeline Stages
	Control Overhead	5%	Instruction Scheduling/ Pipeline Management
Computation 45%	SHA-256 Logic	20%	Orchestration, Data Management, Iteration, Comparison
	32-Bit Adders	15%	Addition Operations
	Bitwise Units	5%	XOR, AND, OR, NOT
	Bit Shift/Rotate	5%	Rotate Bits

Time Latency

COMPONENT/PROCESS		% OF TOTAL	DETAILS
Data Movement (50%)	Data Fetch	25%	Reading
	Result Writeback	25%	Writing
Computation (50%)	ALU Computation	40%	Arithmetic
	Pipeline Overhead	10%	Synchronization

Time and Energy Resources in Mining ASICs

COMPONENT/PROCESS	% OF ENERGY	% OF TIME
Data Movement	55%	50%
Computation	45%	50%

References:

- **Antonopoulos, A. M. (2017). Mastering Bitcoin: Programming the Open Blockchain (2nd ed.). O'Reilly Media. (Chapter 8: Mining and Consensus).** Provides an authoritative, technical explanation of the Bitcoin mining process and the "hardware arms race" that led to the development of ASICs.

- **Vranken, H. (2017). Sustainability of bitcoin and blockchains. Current Opinion in Environmental Sustainability, 28, 1-9.** A good academic source that discusses the energy consumption of ASIC mining, linking the hardware's architecture directly to its environmental impact.

- **Gencer, A. F., van Renesse, R., & Sirer, E. G. (2018). Service-Oriented Sharding for Blockchains. Financial Cryptography and Data Security.** Papers like this contain analysis of the mining ecosystem and the centralizing role of powerful ASIC farms.

- **Jingming Li, Nianping Li, Jinqing Peng, Haijiao Cui, Zhibin Wu, Energy consumption of cryptocurrency mining: A study of electricity consumption in mining cryptocurrencies, Energy, Volume 168, 2019, Pages 160-168, ISSN 0360-5442. https://doi.org/10.1016/j.energy.2018.11.046** Cryptocurrency is a relatively new combination of cryptology and currency in financial areas and is increasingly frequently used worldwide. Blockchain applications are expected to reshape the renewable energy market. However, there is a lack of studies covering the power usage of digital currencies. Therefore, this study ran experiments on mining efficiency of nine kinds of cryptocurrencies and ten algorithms. A comparison of statistical analysis of data in a benchmark and experiment results of Monero mining was conducted. Thereafter, this study provided an estimation of global electricity consumption of the Monero mining activity. The results indicated that the hashing algorithm mainly determines the mining efficiency. Data analysis and experiments and estimated Monero mining electricity consumption in the world and its carbon emission in China as a case study. In 2018, Monero mining may consume 645.62 GWh of electricity in the world after its hard fork. The Monero mining in China may consume 30.34 GWh and contribute a carbon emission of 19.12–19.42 thousand tons from April to December in 2018. Although cryptocurrency mining and blockchain technology are promising, their influence on energy conversation and sustainable development should be further studied.

Compute-In-Memory

5. EFFICIENCY GAINS OF CIM

Even though In-Memory computing is not commercially viable at a large-scale, because of the fundamental challenges in restructuring manufacturing processes and technology, it does exist and it has been benchmarked against existing computing paradigms, and other theoretical computing schemes. Reports for CIM systems have been produced at the academic and industrial level in the last decade, demonstrating the true potential of CIM systems.

One of the most well known technologies with potential for CIM is ReRAM. Unlike other memories that just store data (0 or 1), a ReRAM cell's core property—its resistive state—can be used to natively perform the most critical operation in neural networks: matrix-vector multiplication. By applying a voltage (the input vector) along the rows of a ReRAM crossbar array (which stores the matrix weights as conductance values), the resulting current measured at the columns naturally computes the multiplied sum (Ohm's Law and Kirchhoff's Law). This happens in a single step, in a massively parallel, analog fashion. ReRAM isn't just a memory that can be used for computation; its physical structure is a computational fabric. This makes it fundamentally more efficient for CIM than memories like SRAM or DRAM, which lack this native analog computing property.

Statement: "*Compute-In-Memory architecture eliminates the power wall by performing computations directly within the memory, offering transformative improvements. CIM systems reduce data migration costs by more than 90%.*"

However, ReRAM and other options still suffer from fundamental challenges that do not allow large-scale manufacturing. The biggest constraints against the full commercial implementation of ReRAM are:

- **Manufacturing Challenges:** Difficulty in achieving high yield and uniformity at scale. ReRAM cells can be inconsistent, which is a major hurdle for mass production.
- **Endurance:** ReRAM cells have a limited number of write cycles before they wear out, which is a problem for applications requiring frequent data updates.
- **Integration with CMOS:** Creating reliable and cost-effective hybrid chips that combine ReRAM cells with standard silicon-based CMOS logic is complex.
- **Established Competition:** Competing non-volatile memory technologies, like 3D NAND Flash and the emerging 3D XPoint, already have massive commercial infrastructure and cost advantages.

References:

- **Ping Chi, Shuangchen Li, Cong Xu, Tao Zhang, Jishen Zhao, Yongpan Liu, Yu Wang, and Yuan Xie. 2016. PRIME: a novel processing-in-memory architecture for neural network computation in ReRAM-based main memory. In Proceedings of the 43rd International Symposium on Computer Architecture (ISCA '16). IEEE Press, 27–39. https://doi.org/10.1109/ISCA.2016.13** PRIME is a computer architecture that tackles the "memory wall" problem by enabling Processing-In-Memory (PIM). It proposes using Resistive RAM (ReRAM), a non-volatile memory technology, not just for storing data but also as a physical substrate for performing computations directly within the memory chip. However, it comes with the drawbacks of integrating analog systems into digital systems.

- **https://mythic.ai/technology/compute-in-memory/** Another prpoposal to solve the memory wall is to divide the memory into smaller arrays, and add a local compute unit next to each memory array. However, this solution is only suited specific kinds of AI workloads and is not solving the problem at its core. The biggest drawbacks (precision, scalability, complexity) come from the fact that it is an analog system.

- **Y. -H. Chen, T. Krishna, J. S. Emer and V. Sze, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," in IEEE Journal of Solid-State Circuits, vol. 52, no. 1, pp. 127-138, Jan. 2017, doi: 10.1109/JSSC.2016.2616357.** This constitutes one of the few examples of CIM implemented with standard SRAM cells. However, this chip is designed for being an ultra-low-power, high-speed Binary Neural Network inference at the edge. Meaning that it is not capable of executing any other task, making it of low commercial value and reserved for very specific instances (addressable market is extremely niche).

- **"Hardware Accelerators for Artificial Intelligence" by S M Mojahidul Ahsan, Anurag Dhungel, Mrittika Chowdhury, Md Sakib Hasan, Tamzidul Hoque. Available at arXiv:2411.13717 [cs.AR] (or arXiv:2411.13717v2 [cs.AR] for this version) https://doi.org/10.48550/arXiv.2411.13717** This book chapter provides a comprehensive overview of the specialized hardware designed to meet the massive computational demands of modern AI. It includes a detailed analysis of GPUs, FPGAs, ASICs (like Google's TPU), and Neuromorphic ICs (like Intel's Loihi), comparing their strengths and weaknesses and an in-depth look at seminal research architectures, including the DianNao family, PRIME (a ReRAM-based processing-in-memory architecture), Eyeriss, and Neurocube.

References:

- **Åleskog, Christoffer, Håkan Grahn, and Anton Borg. 2022. "Recent Developments in Low-Power AI Accelerators: A Survey" Algorithms 15, no. 11: 419. https://doi.org/10.3390/a15110419** This survey systematically reviews architectural strategies for developing low-power AI accelerators, essential for edge computing. It identifies three primary approaches to energy efficiency: parallel computing architectures (e.g., GPUs, TPUs) that leverage massive core counts; in- or near-memory computing (e.g., using ReRAM) that minimizes data movement energy by performing calculations within the memory array itself; and approximate computing techniques that exploit the error-resilience of neural networks to reduce computational precision.

- **Yiran Chen, Yuan Xie, Linghao Song, Fan Chen, Tianqi Tang, A Survey of Accelerator Architectures for Deep Neural Networks, Engineering, Volume 6, Issue 3, 2020, Pages 264-274, ISSN 2095-8099, https://doi.org/10.1016/j.eng.2020.01.007** This survey provides a comprehensive overview of accelerator architectures for deep neural networks (DNNs), categorizing them by their foundational approach to tackling the "memory wall" problem. It details three key architectural paradigms: Spatial Architectures (like Google's TPU), which use a fixed arrangement of processing elements to optimize data reuse through systolic arrays; Dataflow Architectures, which dynamically reconfigure the execution path to minimize data movement; and In-Memory Computing Architectures, which use technologies like ReRAM to perform computations directly within the memory array, thereby eliminating data transfer altogether. The analysis concludes that while spatial and dataflow architectures offer a practical balance of performance and efficiency for current CMOS technology, emerging in-memory computing represents a revolutionary, long-term path to extreme energy efficiency by fundamentally collapsing the traditional von Neumann bottleneck.

6. EFFICIENCY GAINS OF CIM IN MINING ASICS

Most of the operations involved in a hashing round can be performed with our FAU. These include bit wise AND,XOR and addition, as well as rotations/shifts. We can achieve at least 80% efficiency in the on-chip bottleneck, with 40% efficiency gains in time and energy resources. We expect 30–40% overall system improvements in terms of both time latency and power efficiency. Our CIM architecture eliminates the final on-chip bottleneck by combining SRAM registers and computing transistors in a linearly scalable circuit, reducing power consumption and improving performance. Based on the SHA-256 data flow analysis and Bitcoin mining ASIC architecture, we provide an estimated overall power savings projection CIM architecture would achieve in a mining ASIC. As demonstrated in prototypes like Samsung's GDDR6-AiM (90% energy reduction) and Mythic's analog CIM (95% reduction), a very significant reduction in resources is expected. But, instead of developing new special types of transistors, or analog devices that require new manufacturing processes, we use standard SRAM edge-triggered registers.

In mining ASICs we will take the percentage of addressable time and energy waste from the VNB to be around 50%, in contrast to the +85% that has been observed in other types of processing units. Below we will provide the underlying data and a rough breakdown of the variables of time and energy efficiency of mining ASICs. Although mining ASICs eliminate instruction fetch and cache misses, they still spend ~50–60% of their energy and ~40–50% of their time moving data between registers and ALUs — a direct consequence of the Von Neumann architecture at the processor level. Intermediate values are calculated and stored, with several values stored and reused across different rounds. The SHA-256 algorithm maintains 8 main working variables (a,b,c,d,e,f,g,h) that are updated each round (64 rounds per block), stored between rounds in pipeline registers, and refetched for the next round's calculations.

Our approach eliminates these intermediate data movements by keeping values in memory where they are computed, through direct in-place updates of the working variables. Our CIM architecture reduces the data movement overhead, enabling up to 50% power reduction while simultaneously providing 40–50% performance improvement. This represents the single largest efficiency leap in mining ASIC history. The key drivers of savings are:

1. Data Movement Elimination

Register file access energy
Pipeline register overhead

2. Computation Efficiency

Adder energy
Bitwise operations

SHA-256 hashing involves extensive intermediate value storage and refetching of working variables (a-h), message schedule, and temporary values. The SHA algorithm is filled with intermediate value storage—primarily the 8 state variables (a-h) that are constantly read, modified, and rewritten across 64 rounds. This constant data movement between computational units and pipeline registers represents the bottleneck that our CiM architecture eliminates. This makes our technology valuable since it addresses a real, measurable inefficiency in current mining ASICs.

Below we provide a different way of separating the baseline power distribution in traditional ASIC Architectures, based on architectural analysis of modern Bitcoin mining ASICs (e.g., Bitmain BM1398, MicroBT M60). It is an alternative method to quantify the power consumption in a Mining ASIC, reflecting other industry reports.

CIM Efficiency Gains for Mining ASICs

COMPONENT/PROCESS		POWER	% OF TOTAL	CIM SAVINGS POTENTIAL
Hash Cores 80% (4,200 W)	Data Movement	1,580 W	35%	80%
	Computation Logic	1,300 W	20%	50%
	Clock/Pipeline Overhead	790 W	15%	30%
	Control Logic	525 W	10%	30%
Everything Else 20% (1,050 W)	Memory Controller	320W	15%	50%
	I/O Interface	280W	2.5%	-
	Power Management	250W	2.5%	-
TOTAL		**5,250 W**	**100%**	**53%**

A 60-80% improvement on the 53% total addressable bottleneck cited in the table above translates into an overall system efficiency and performance increase of 30-40%.

Statement: "*Our CIM architecture eliminates the on-chip bottleneck, promising 30-40% improvements in system latency and power efficiency for mining ASICs.*"

Our CiM architecture enables ~40% overall SoC power reduction while simultaneously delivering ~40% performance improvement—a combination that would fundamentally reset Bitcoin mining economics and establish a new efficiency benchmark for the industry, which would instantly make all existing miners obsolete.

Traditional ASICs:
Compute → Store in Register → Read Register → Compute → Store in Register...

Our CIM ASIC:
Compute in Memory → Compute in Memory → Compute in Memory...

7. R+D OVERHEAD FOR CIM

Although industry leading names have succesfully designed and developed CIM schemes, these are not manufacturable at commercial scale. This is in large part due to the fact that the development of new memory and transistor types doesn't just require investment into R+D, it also requires huge investment into new manufacturing processes and technology. For these reasons CIM proposals have not been brought to market and most probably won't be ready for mass production in this decade. Below we provide an estimate of **research budgets into CIM over the last decade** (estimates based on public information).

IBM: *$400 Million USD*
Intel: *$1 Billion USD*
Samsung: *$500 Million USD*

These are estimates of investment into CIM research and development. The **investment required to actually mass produce CIM devices would be much greater** and close to a decade of platform migration in order to reaccomodate the entire manufacturing process with new factories, new lithography technology, formation of human resources, restructuring new supply chains, etc.

References:

- Samsung's annual financial reports consistently show R&D spending in the tens of billions for its Device Solutions Division (which includes semiconductors). A small fraction dedicated to advanced memory-centric computing over a decade easily reaches our cited figure. Announcements about developing MRAM and other embedded non-volatile memory technologies, which are enablers for CIM, represent significant, directed investment.

- IBM's investment is heavily channeled through its AI Hardware Center at the Albany Nanotech Complex, which is explicitly focused on next-generation AI hardware, including CIM and analog AI cores.

- Intel's acquisition of Movidius (2016) and Habana Labs (2019) for ~$2B. While not exclusively for CIM, these represent massive bets on specialized AI accelerators and the architectural principles that lead to CIM. R&D Programs Point to the decades-long funding of Intel Labs and the specific development of the Loihi neuromorphic research chip. Neuromorphic computing is a direct form of CIM. The long-term funding for hundreds of researchers and fabrication costs for multiple generations of a highly experimental chip easily runs into the hundreds of millions.

Financials

8. BOTTOM-LINE PROFITABILITY GAINS

- Our architecture is particularly powerful because it eliminates the most expensive boundary—between registers and the first level of cache/SRAM—which is where the highest-frequency data movement occurs.

- In the section dedicated to CIM gains in mining ASICs, we provided a breakdown of power and time resources. We were able to extrapolate data from other CIM implementations, together with industry reports from mining ASICs to identify the total addressable Von Neumann Bottleneck for the on-chip movement in hashing cores.

- **We identified the total addressable time and energy resources to be around 53% of consumption.**

- We also know that most CIM concept chips improve data movement by more than 90%.

- If we set an achievable goal of improving the addressable fraction in mining ASICs by 60-80%, we can achieve an overall system efficiency and performance improvement of 30-40%. In other words, **eliminating 60-80% of the 53% of time and energy resources spent in the bottleneck, represents a time latency and power efficency improvement of 31.8-42.4%.**

To calculate the bottom-line profitability increase of our mining architecture with respect to existing state-of-the-art mining ASICs we use the following underlying variables on mining costs and throughput.

- **Hash Rate:** Each ASIC has a throughput of 335 TH/s. At the current Total Network Hash Rate this gives a total productivity of **0.000137 Bitcoins/day.**
- **Power Consumption:** Power resources are **128 kWh/day** per ASIC.
- **Price of Elecrticity:** Globally competitive mining farms have to operate in locations where the rates for electric power is close to **$0.05 USD/kWh** in order to ensure profitability at current mining turn-out rates and market prices.

We have calculated the Bitcoin output from 1,000 mining ASICs (BitS21 Hydro) from operating with the aforementioned cost of electricity. This information is used to further calculate the bottom-line profitability.

Yearly Profits per 1,000 ASICs

SCENARIO	BITCOINS GENERATED	ENERGY COST (MUSD)	PROFIT IN BITCOINS	PROFIT INCREASE
Current ASICs	50	2.34	27	-
+30% T&E Efficiency	65	1.7	48	+75%
+40% T&E Efficiency	70	1.4	56	+100%

Statement: "*Efficiency and performance gains of 30-40% translate directly into a* ***75-100% increase in Bitcoin mining bottom-line profit.***"

Mining metrics are calculated based on a **network hash rate of 1,116,498,453,900 GH/s** (equivalent to 1.1 ZH/s) and using a **BTC - USD exchange rate of 1 BTC = $ 100,000 USD.** These figures vary based on the total network hash rate and on the BTC to USD conversion rate. Network hash rate varies over time, this is just an estimation based on current values. **Block reward is fixed at 3.125 BTC.** The **average block time used is 1175 seconds.** Future block reward and hash rate changes are not taken into account. The electricity price used in generating these metrics is **$ 0.05 per kWh.**

References:

- CoinDesk's suite of complimentary research reports deliver high quality, trusted and unbiased data-driven analysis into key digital asset trends and narratives. https://data.coindesk.com
- https://cryptocompare.com
- Online Crypto Calculators

9. FINANCIAL PROJECTIONS

The potential income projected from in-kind royalties is calculated on the basis of manufacturing volume of the three biggest mining ASIC manufacturers, the average price of bitcoin and standard royalties percentage.

- **ASIC manufacturing volumes:** The three largest mining ASIC manufacturers have annual produciton that is estimated to range between **100k-300k units,** depending on several factors.

- **Price of Bitcoin:** The 1-year average on Bitcoin oscillates close to **$100,000 USD.**

- **Royalties:** Royalties on a strong IP portfolio with similar characteristics to our underlying technology and market positioning can range in the 1-5% of revenue. For our long term projections we have used a **2.5% in-kind royalties** on manufacturing yields.

References:

Annual Spending % on Microprocessors per Year and Market Sizes are derived from studies and industry reports from Gartner, IDC, McKinsey, MarketsandMarkets, Precedence Research, and on chip procurement trends (e.g., NVIDIA earnings, TSMC demand, Bitcoin mining farm expenditures).

Juan Pablo Ramirez
Architect, Project Manager & CEO
jramirez@binaryprojx.com

Pablo César Vázquez Estrella
Chief Financial Officer
pablo.vazquez@binaryprojx.com

Sergio Adrián Trujillo González
Senior Software Engineer
sergio.trujillo@binaryprojx.com

Héctor Alejandro Galvez López
Consultant in Elec. Eng.
University of Guadalajara
hgalvez@binaryprojx.com

THANK YOU!

www.binaryprojx.com