Extreme Datacenter Specialization for Planet-Scale Computing: ASIC Clouds

Shaolin Xie
University of Washington
shaolx@uw.edu

Scott Davidson
University of Washington
stdavids@uw.edu

Ikuo Magaki
Apple Inc.
ikuo.magaki@icloud.com

Moein Khazraee
UC San Diego
mkhazraee@ucsd.edu

Luis Vega
University of Washington
vegaluis@uw.edu

Lu Zhang
UC San Diego
luzh@eng.ucsd.edu

Michael B. Taylor
University of Washington
prof.taylor@gmail.com

ABSTRACT

Planet-scale applications are driving the exponential growth of the cloud, and datacenter specialization is the key enabler of this trend, providing order of magnitudes improvements in cost-effectiveness and energy-efficiency. While exascale computing remains a goal for supercomputing, specialized datacenters have emerged and have demonstrated beyond-exascale performance and efficiency in specific domains.

This paper generalizes the applications, design methodology, and deployment challenges of the most extreme form of specialized datacenter: ASIC Clouds. It analyzes two game-changing, real-world ASIC Clouds–Bitcoin Cryptocurrency Clouds and Tensor Processing Clouds–discuss their incentives, the empowering technologies and how they benefit from the specialized ASICs. Their business models, architectures and deployment methods are useful for envisioning future potential ASIC Clouds and forecasting how they will transform computing, the economy and society.

CCS CONCEPTS

• **Computer systems organization → Special purpose systems**;

KEYWORDS

Datacenter, ASIC, Accelerator

1 WHAT ARE ASIC CLOUDS

In the last decade, two parallel trends in the computational landscape have emerged. The first is the bifurcation of computation into two sectors: cloud and mobile. The second is the rise of dark silicon [29] and dark silicon aware design techniques [48, 54] such as specialization and near-threshold computation. Specialized hardware has existed in mobile computing for a while due to extreme power constraints, however recently there has been an increase in the amount of specialized hardware showing up in cloud datacenters. Examples include Baidu's GPU-based cloud for distributed neural network acceleration and Microsoft's FPGA-based cloud for Bing Search [49].

At the level of a single node, we know that ASICs can offer order-of-magnitude improvements in energy-efficiency and cost-performance over CPU, GPU, and FPGA. Our recent papers [35, 36, 43] explore the concept of ***ASIC Clouds*** which are purpose-built datacenters comprised of large arrays of ASIC accelerators. ASIC Clouds are not ASIC supercomputers that scale up problem sizes for a single tightly-coupled computation; rather, ASIC Clouds target scale-out workloads consisting of many independent but similar jobs, often on behalf of many users. Notably, our work [1] predicted the Google TPU [33] before Google announced it.

As more and more services are built around the cloud model, we see the emergence of planet-scale workloads. Examples include Facebook's face recognition, Siri answering speech queries, and YouTube transcoding user-uploaded videos to Google's VP9 format. For systems of this scale, the total cost of ownership (TCO) improvements derived from the reduced marginal hardware and energy costs of ASICs could make it a routine business decision to create ASIC Clouds.

In this paper, we overview the emergence of datacenter specialization in different industries and then examine two real-world game-changing ASIC Clouds—Bitcoin ASIC Clouds and Tensor Processing ASIC Clouds for ML—what their incentives were, and the impact they hard in their respective markets. Performing deep analysis of real ASIC Clouds provides valuable insights on ASIC Cloud design and deployment problems. Unlike conventional servers that are assembled from low cost off-the-shelf components, ASIC Clouds involves high mask cost and long time-to-market which prevents fast and incremental design iterations. We summarize the end-to-end system-level optimization techniques and discuss deploying problems that are unique to ASIC Clouds. By considering the chip design, server design, and data center design of an ASIC Cloud in a cross-layer system-oriented fashion, our work [36, 43] develops methodologies which designers can use to create novel systems that optimize the TCO in real-world ASIC Clouds. We conclude by showing the designs of several ASIC Cloud systems that use the aforementioned design methodologies for the following applications: Bitcoin mining, YouTube-style video transcoding, Litecoin, and Deep Learning.

1.1 ASIC Cloud Applications

Hardware specialized in the cloud is used to attain both high performance and power efficiency. Specialization can be done at different

This work is based on an earlier work [35, 36, 43, 53, 55] and in particular: Specializing a Planet's Computation: ASIC Clouds , Issue 3, (14 June) ACM 2017 https://doi.org/10.1109/MM.2017.49

Industries	Workloads	Specialized Level	Representative Specialized Clouds
Artificial Intelligence	Image Recognition, Natural Language Processing, Speech Recognition/Translation	Server, FPGA, ASIC	Google TPU [33], Microsoft BrainWave [22], Neural Network Accel [13, 17, 20, 21, 28, 31, 37, 42, 50, 51, 60]
Financial	Blockchain, High Frequency Trading Real Time Risk Control	FPGA, ASIC	Bitcoin Miner [56], Litecoin Miner [8], Ethereum Miner [6], HFT Accel [40], J.P. Morgan FPGA Accelerator [52]
Internet	Database, Web Search, Social Network	FPGA, ASIC	Bigdata Accel [25], Memcached Accel [26, 38, 41], Graph Accel [12, 27, 47], Microsoft Bing Cloud [49]
Media/Entertainment	Video Transcoding, Live Streaming	Server, ASIC	ASIC Cloud Transcoder [43]
Genomics	Burrows-Wheeler Aligner (BWA), Genome Analysis Toolkit (GATK)	Server, FPGA	Microsoft Genomics Clouds [9], Intel-Broad Genomics Stack [3], Amazon-Falcon Accelerated Genomics Pipelines [7], Darwin Accel [58], GenAx [24], DRAGEN FPGA Bio Platform [5]
Scientific Computing	Physical Simulation, Modular Dynamics	ASIC	GF11 [16], Anton 1 [23], Anton 2 [18], GRAPE8 [44]

Table 1: Specialized Clouds in industry and academia.

levels, from Server, to FPGA, and ASIC. *Server Specialization* uses customized motherboard with dedicated processor (GPU), storage devices, networks and software stacks; *FPGA Specialization* uses FPGAs to accelerator certain workloads; *ASIC Specialization* employ ASIC chip arrays to execute tasks with extreme constrains. Table 1 shows the prominent specialized clouds appearing in industry and academia.

The performance and efficiency of different specialization level increase exponentially, but the cost for design and deployment them also increase exponentially. The specialization level is a trade-off between the investment, time-to-market and total cost of ownership. With sufficient marketing and capital incentives, ASIC specialization specialization can be applied to any cloud applications, turning it into an *ASIC Clouds*. In later sections we will examine when it makes sense to design and deploy an ASIC Cloud and proposes a new rule, the **two-for-two rule** that highlights the link between development cost (NRE) and the speedup or energy improvements the ASIC Cloud must get in order to have a net benefit.

2 ANALYSIS OF TWO GAME-CHANGING ASIC CLOUDS

Among all of the ASIC Clouds, Bitcoin miners and Google's TPU are the most influential. In this section, we review the history, incentives and benefits with corresponding market and technology backgrounds for these ASIC Clouds.

2.1 Bitcoin ASIC Clouds

Bitcoin, since its deployment in January 2009 [45], has experienced explosive exponential growth. As of summer 2018, there are 17.15 million Bitcoins (BTC, or ₿) in circulation with the USD/BTC exchange rate being $6,366. Therefore, Bitcoin's market capitalization exceeds $109B.

Such rapid growth has made Bitcoin the most successful digital currency. Underpinning Bitcoin's success is a series of technological innovations spanning from algorithms, to distributed software, and to hardware. Amazingly, these innovations were not initiated by corporations or governments but rather emerged through a grassroots collaboration of enthusiasts.

In this section, we introduce the hardware systems that maintain the integrity of the Bitcoin blockchain, discuss the relevant economic forces, and then delve into the fascinating hardware ecosystem that has emerged—from GPUs to FPGA to custom ASICs. Greater discussion of Bitcoin's software and user-experience can be found in [45, 53].

The latest round of hardware—dedicated ASICs—was financed, developed, and deployed by Bitcoin enthusiasts which is perhaps an unprecedented event in recent history. As the value of the Bitcoin ecosystem grew, the industry rapidly matured and Bitcoin mining has attained extraordinary scale, equivalent to 3.2 billion high-end GPU's. Bitcoin ASIC hardware has co-evolved with datacenter design and now a majority of the computation is performed in highly specialized ASIC-filled datacenters that collectively form an *ASIC Cloud* [35, 36, 43].

2.1.1 CPU: First Generation Mining. The bitcoin miner source code is on github, and is surprisingly simple (see https://github.com/bitcoin/bitcoin/blob/master/src/miner.cpp). The basic computation,

```
while (1)
 HDR[kNoncePos]++;
 IF (SHA256(SHA256(HDR)) < (65535 << 208)/ DIFFICULTY)
   return;
```

leverages existing high-performance SHA256 hashing libraries. One simple optimization employs a *mid-state* buffer, which hashes the first part of the block's header' which precedes the nonce and has a constant intermediate hash value. More optimizations are discussed in [46].

The SHA256 computation takes in 512 bit blocks and performs 64 rounds of a basic encryption operation involving several long chains of 32-bit additions and rotations, as well as bit-wise functions including xors, majority, and mux functions. An array of 64 32-bit constants are used as well. Each round depends on the previous round, creating a chain of dependencies between operations. Although successive SHA256 rounds cannot be parallelized, every

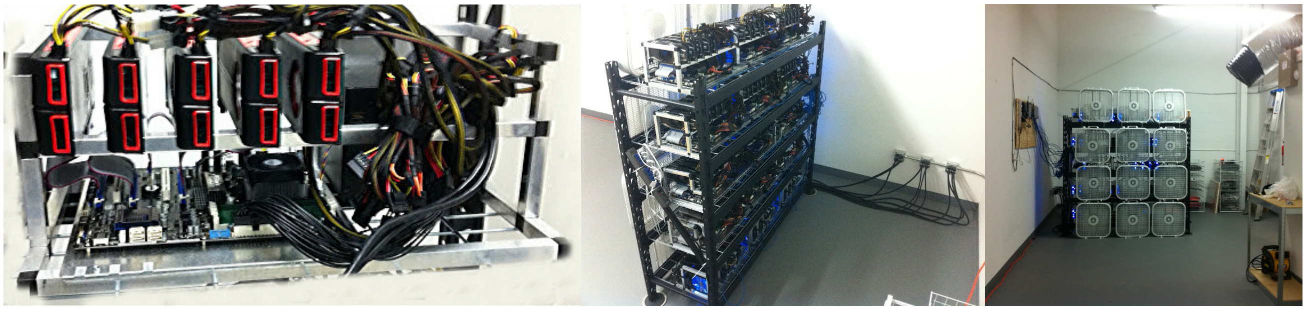


Figure 1: *Left:* An open-air GPU mining rig. Five GPUs are suspended above the motherboards, with riser cables connecting the GPU's PCI-E connector to the motherboard below, and a single high-wattage power supply. *Center,Right:* Two pictures of a homebrew 69-GPU Bitcoin mining datacenter. Note the ample power cabling, left, and the cooling system, consisting of box fans and an air duct, right. Photos Credit: James Gibson (gigavps).

nonce can be tested in parallel with each other making this a classic Eureka-style computation. Furthermore, some operations inside a round are parallelizable. Typical multicore machines have extra hardware optimized for less regular computations, resulting in wasted performance and energy efficiency.

2.1.2 GPU: Second Generation Mining. In October 2010, open-source miner software for GPUs was released on the web. It was rapidly optimized and adapted by several open-source efforts. Typically, this software would implement the Bitcoin protocol as well as GPU voltage/temperature/error control in a language such as Java or Python. The core nonce-search algorithm was distributed as a single OpenCL file (e.g., [10]) that would be compiled down, by installed runtimes , into the GPU's hidden native ISA.

Scaling up GPUs. GPUs proved much more accessible than FPGAs for Bitcoin enthusiasts, requiring only PC-building skills and avid forum-reading but no formal training in parallel programming or FPGA tools. After investing time in building a GPU-based mining rig that is literally minting cash, the natural inclination is to scale it up.

Efforts to scale BTC hash rate through GPUs pushed the limits of consumer computing in amazing and novel ways. As described in [53], a crowd-sourced standard evolved, where 5 GPUs were suspended over a cheap AMD motherboard with minimum DRAM, connected via 5 PCI-E 8x-to-1x extender cables to reduce motherboard costs, and using a large high-efficiency power supply to drive all GPUs. The system was open-air to maximize airflow, as shown in Figure 1-left. These approaches enabled a low-cost motherboard, CPU, and DRAM combination to be amortized across 5 GPUs, improving capital efficiency.

After optimizing per-GPU overhead, the next scaling challenge is the prodigious power and cooling requirements of maintaining many GPUs. With each GPU consuming 300W, the power density exceeded that supported by both high-density datacenters and residential electric grids. Most successful Bitcoin mining operations typically relocated to warehouse space with a large air volume for cooling and cheap industrial power rates. Figure 1-right shows a homebrew datacenter consisting of 69-GPU rack that is cooled by an array of 12 box fans and an airduct.

2.1.3 FPGA: Third Generation Mining. June 2011 brought the first open-source FPGA bitcoin miner implementations. FPGA are inherently good at both rotate-by-constant operations, and at bit-level operations both used by SHA256, but not so good at SHA256's 32-bit adds.

The typical FPGA miner design replicates multiple SHA-256 hash functions and *unrolls* them. With full unrolling, the module creates different hardware for each of the 64 hash rounds, each of which was separated by pipeline registers. These registers contain the running hash digest as well as the 512 bit block being hashed. The state for a given nonce trial would proceed down the pipeline, one stage per cycle, allowing for a throughput of one nonce trial (hash) per cycle.

Hackers developed custom boards that minimized unnecessary cost due to parts like RAM and I/O and focused on providing sufficient power and cooling. These boards attained 215 MH/s rates with Spartan XC6SLX150 parts, and quad-chip boards were developed to reduce board fabrication, assembly and bill-of-materials costs, reaching 860 MH/s at 216 MHz and 39 W, and costing $1060.

Another manufacturer, Butterfly Labs (BFL), based in Kansas, offered a non-open-source version that cost $599 with similar 830 MH/s performance. BFL was by all accounts the most successful commercial FPGA miner vendor.

Unfortunately, FPGAs had trouble competing on cost per GH/s with high-volume GPUs that were on more advanced process nodes and often would go on sale at NewEgg. FPGA had superior energy-efficiency by as much as 5×, breaking-even on total cost of ownership (TCO) after a year or two.

The reign of FPGAs was brief because ASICs arrived soon after, providing orders of magnitude cost and energy-efficiency improvements.

2.1.4 The Race to ASIC: Fourth Generation Miners. Three companies came to market with ASIC miners in very close succession. The designs were based loosely on FPGA-based miners. Because ASICs had such an enormous benefit over prior devices, the emphasis was to get a working, not necessarily optimal, design out as quickly as possible. To understand the relative benefits of GPU and ASIC performance, cost, and energy for Bitcoin, especially as process geometries shrink, refer to [35, 43].

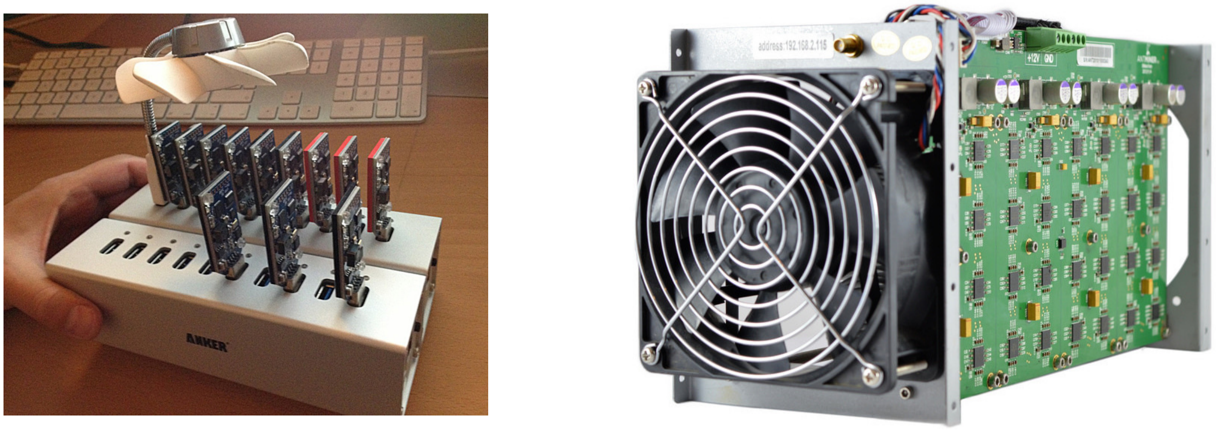


Figure 2: *Left:* A USB hub hosting an array of ASICMINER Block Erupter USB-stick style bitcoin miners, and a USB-powered cooling fan. Each USB-stick uses a 130-nm ASIC that hashes at 330 MH/s, or about half the performance of $450 28-nm AMD 7970 GPU. *Right:* Bitmain Antminer machine with two parallel sea-of-ASICs PCBs. Photo Credits: DennisD7, and dogie of bitcointalk.org.

Butterfly Labs (BFL) BFL was first to announce an ASIC product line, confident from their prior FPGA product line success. BFL took pre-orders in June 2012 for three types of machines; $149 Jalapenos rated at 4.5 GH/s, $1,299 SC Singles rated at 60 GH/s and $30K SC MiniRigs rated at 1,500 GH/s. At these prices, the machines could generate 20-50× more bitcoins per dollar invested versus GPUs. The pre-order funds, which exceeded $250K on day one, presumably covered the considerable ~500K NRE mask costs for BFL's 65-nm GLOBALFOUNDRIES process.

The BFL chip used in all three products contained 16 double-SHA256 hash pipelines. The die was 7.5 mm on a side, and placed into a 10x10 mm BGA 144 package.

Surprises. BFL initially targeted Nov 2012 for product ship date, however the schedule repeatedly slipped after setbacks and delays from the ASIC foundry, packaging and BFL itself. It took almost until Nov 2013 to clear the order backlog. A major cause was that the chip's power consumption was 4–8× expected, requiring a redesign of all ASIC systems. For example, the Jalapenos, slated to use one chip, shipped with two chips to meet the 4.5 GH/s rate, and they typically operated at 30 Watts, close to 6W per GH/s.

ASICMINER

The ASICMINER Bitcoin effort started in early July, after BFL had started taking pre-orders for their machines, and consisted of three Chinese-national founders. A key motivation was to prevent BFL from being the sole Bitcoin ASIC purveyor and controlling the blockchain. Their approach was quite different than BFL's; they intended not to sell hardware initially, but to run an ASIC datacenter that mined Bitcoin on behalf of shareholders. This is arguably the earliest example of an *ASIC Cloud*. This approach, eliminating the need to ship HW to customers, won them the race to large-scale deployment.

Lacking BFL's name recognition, they raised funding through online forums, namely bitcointalk.org, and also some Chinese-language forums. They carefully outlined their plan for developing an ASIC, and responded to hundreds of questions by the online community, regarding their business model, their technical decisions, and their financial trustworthiness. This paper [53] summarizes the openly-posted developments.

The IPO closed Aug 27, selling 163,962 shares, roughly equivalent to 160K USD.

By Feb 14, they had 2Th/s chip deployed and hashing. Officially the ASIC Bitcoin movement was in full force! First they sold boards from their datacenter, but later they developed a USB miner stick, the Block Erupter, containing a single ASIC, which sold initially for 2 bitcoin in large lots to be resold by others, and rapidly dropped in price. Figure 2-left shows a USB hub hosting an array of ASICMiner Block Erupter USB-stick style bitcoin miners, and a USB-powered cooling fan. Each USB-stick uses a 130-nm ASIC that hashed at 330 MH/s at 1.05 V and 2.5 W, which is 40× more energy efficient than the 28-nm AMD 7970 GPU, and 4.4× cheaper per GH/s.

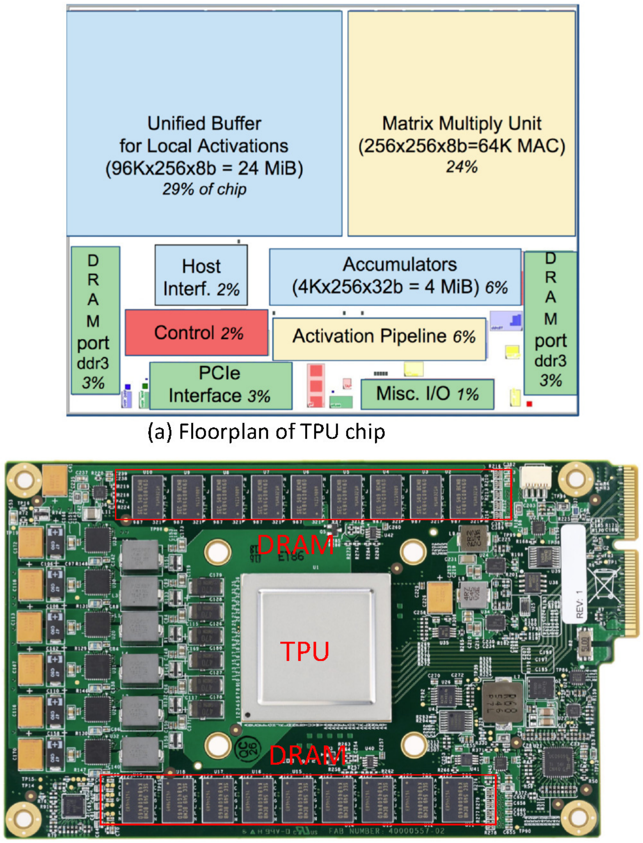
ASICMINER shares reached 4 BTC each in October 2013, signifying a 40× return to the initial investors, in BTC. Of the three efforts, clearly ASICMINER was the most innovative in trying out new ASIC products and business models.

Avalon

The Avalon company was another grass-roots effort that secured funding by direct Internet pre-sales of units via an online store. A key founder, ngzhang, established his reputation with the design of a top Bitcoin FPGA board, Icarus.

They focused on an 110-nm TSMC implementation of a single double-SHA256 pipeline, measuring 4 mm on a side, and packaged 300 chips across 3 blades inside a 4U-ish machine. Like ASICMiner, they were based in Shenzhen, China. They ran pre-order sales for 300 rigs, each selling for $1299 each, or 108 bitcoin at the time, and hashing at 66 GH/s on 600W.

They taped out slightly after ASICMiner, with a target date of Jan 10. On Jan 30, 2013, Jeff Garzik, a Bitcoin developer, was the first customer in history to receive a Bitcoin ASIC mining rig, which earned ~15 BTC the first day.


Unified Buffer for Local Activations (96Kx256x8b = 24 MiB) 29% of chip
Matrix Multiply Unit (256x256x8b=64K MAC) 24%
DRAM port ddr3 3%
Host Interf. 2%
Accumulators (4Kx256x32b = 4 MiB) 6%
DRAM port ddr3 3%
Control 2%
Activation Pipeline 6%
PCIe Interface 3%
Misc. I/O 1%
DRAM
TPU
DRAM

(a) Floorplan of TPU chip

(b) TPU Printed Circuit Board

Figure 3: TPU Architecture. PCIe connected accelerators. Image source: In-Datacenter Performance Analysis of a Tensor Processing Unit$^{\text{TM}}$ [33]

Subsequently, Avalon sold off new machine batches, a 2nd batch of 600 rigs for 75 BTC ($1599) on Feb 2, and a third batch of 600 rigs, also for 75 BTC ($5500) on Mar 25. They sold out almost immediately. Avalon followed up with direct chip sales, selling over 100 batches of 10,000 chips for 780 BTC per batch, or about $78,000, enabling others to design systems around the new chips.

2.1.5 The Massive ASIC War:

Fifth Generation Bitcoin Miners. The next generation of ASICs marked several departures from the first. Since first generation ASICs had proven the value proposition of Bitcoin ASICs, venture capitalists and other investors jumped in to fund a swath of startups, many featuring industry veterans. Moreover, the competition was not easily-beaten GPU's, but rather other ASICs. New ASICs had to beat the previous generation in cost/performance and energy efficiency in order to be competitive and stay ahead of ever-rising difficulty levels. These successive generations had two potential sources of innovation: better architectures and more advanced process nodes. To date, there have been over 37 different ASIC efforts.

Bitfury, with star chip designer Valery Nebesny, reached 55nm first in mid-2013 with a best-of-class full-custom implementation that was in many cases superior to 28nm designs, reaching 0.8 Watts per GH/s, and 2.5 GH/s per chip. 16 chips were placed on a PCB and 16 PCBs went into a backplane. Unlike most other implementations, rather than unrolling double-SHA hashes into long pipelines, the Bitfury architecture used "rolled" in-place hashes that iterate in place. Bitfury also introduced support for string designs, where ASIC power pins are connected serially like Christmas Tree lights, eliminating DC-DC converters, which comprise 20%-40% of Bitcoin server cost. Bitfury's initial 100 TH of chips went to a large datacenter provider that financed the NRE. Later, individual chips were sold, and interesting variants ranging from USB keys to blades were sold by Internet third parties, including on Amazon.com.

Platform	TPUv2	4×v100	4×v100
Clouds	Google Clouds	AWS	AWS reserved instance
Price ($ per hour)	6.69	12.24	8.35
image/seconds	3,186	3,128	3,128
million images/$	**1.71**	**0.92**	**1.35**

Table 2: Cost efficiency of TPUs. Data source: Elmar Hauβmann, Comparing Google's TPUv2 against Nvidia's V100 on ResNet-50 [30]. AWS is Amazon cloud computing service, which is also paid on-demand. AWS reserved instance is a contract plan with discount when renting for 12 months [4]. The performance is gathered via ResNet-50 Tenseflow implementation with synthetic data [11].

Sweden-based KncMiner reached 28-nm by Oct. 2013. Shortly after, SF-based Hashfast and Austin-based Cointerra [14] also came out with 28nm implementations. These miners were much more cost-efficient compared to the Bitfury chips, but energy efficiency was actually worse, > 1.1W/GH/s. These designs placed 4 dies on a shared substrate that reached several hundred Watts and required water cooling.

Since Bitfury had several months to ramp before these products came out, Hashfast and Cointerra were caught off-guard by massive quantities of deployed highly-efficient Bitfury 55nm chips, as well as concurrently shipping 28nm miners. This created a narrow window of usefulness for these machines and contributed to both Cointerra and Hashfast going out of business.

BFL, Spondoolies and Bitmain (see Figure 2-right) also implemented 28nm miners, targeting energy efficiencies that matched or exceed Bitfury (0.7W per GH/s). There is evidence that 21, Inc hit the Intel 22nm node around Dec 2013, but the details are closely guarded secrets.

2.1.6 To the Victors Go The ASICs:

Sixth Generation Bitcoin Miners. The current sixth generation of Bitcoin miners consists of companies that survived the second ASIC wave, and advances to bleeding edge nodes as they came out (e.g. 20-nm and 16-nm). The two primary publicly known contenders are BitFury and Bitmain, which have 16nm parts. Both run at ultra low voltages; Bitfury attains better than 0.07 Watts per GH/s energy efficiencies, an improvement of over 100× over the first 130-nm miners, and over 8,000× over GPUs.

2.2 TPU: Deep Learning ASIC Clouds

Besides Bitcoin miners, Google's tensor flow processing unit (TPU) [33] is another game changing ASIC Cloud. Training for deep-learning involves a tremendous amount of computation which makes general purpose CPUs inefficient. Fortunately, deep learning algorithms are well structured and can be accelerated using specific computer architectures. Recently, many companies like Google, Graphcore, Horizon, Bitmain, and others have been developing dedicated hardware architectures and chips that can train and deploy deep learning models with more than one order magnitude better efficiency when compared with traditional CPUs and GPUs.

While there are many deep-learning accelerators for inference, GPUs have dominated training acceleration until the reveal of the TPU.

Google announced the TPU version 1 in 2016 and published their results at the International Symposium on Computer Architecture (ISCA) in 2017 [33]. The results showed about 15-30× better performance and 30-80× better performance-per-watt when compared to CPUs and GPUs. These improvements enabled the ability to run large neural networks at a relatively low cost and at scale.

Google's TPU was taped out in 28nm and achieves 700MHz while drawing 40W. It was deployed as an accelerator plug-in card and uses a 12.5 GB/s PCIe Gen3 x16 PCIe interface.

2.2.1 *Architecture, Implementation and Software of TPU.*

Most Deep Learning processing is ultimately memory-bound. To increase the memory bandwidth, two arrays of DRAM chips are placed closely to the TPU chip, as shown in Figure 3 (b). The card interacts with the host via a high speed PCIe bus, receiving TPU instructions and data to execute rather than having an on-board standalone CPU.

The TPU can be seen as a special matrix multiply accelerator. Figure 3 (a) shows the TPU's floorplan. It mainly consists of a 24 MB Unified Buffer, a Matrix Multiply Unit and other auxiliary units. The 24 MB Unified Buffer holds temporal data and takes up a third of the die area. The 24 MB size was chosen to match the dimension of the Matrix Multiply block and to simplify compilation. The Matrix Multiply Unit uses a systolic array micro-architecture which can compute GEMM effectively and takes up a quarter of the die area. Figure 3 (b) shows the TPU mounted on a its PCB which plugs into servers using a PCIe connector.

The TPU software stack is divided into User Space and Kernel Space Drivers. The low-level Kernel driver directly interacts with the hardware and is only responsible for primitive memory access and interrupts. This lightweight interface is designed for long-term compatibility. In contrast, the User Space Driver supports multiple APIs to run the applications that are originally developed for CPUs and GPUs so it changes often. It invokes and checks TPU execution, reshapes data, translates API calls into TPU instructions binaries. The User Space driver is more like a JIT runtime, it compiles a model the first time it is deployed, caching the binaries and transmitting the weight parameters into the TPU; subsequent invocations do not require this overhead.

With a dedicated accelerator architectures, the power efficiency of TPU is about 1 order of magnitude better than that of CPUs (Intel Haswell) and GPUs (Nvidia Tesla K80). The K80 server is 1.7× - 2.9× better than a Haswell server while a TPU server has 17× - 34× better than Haswell server in terms of total-performance/Watt, which makes the TPU server 14× - 16× power efficient than the K80 server [33].

chip	TPUv1	TPUv2	TPUv3
Announced	2016	May-17	May-18
Access	Internal-Only	Service-Beta	Undisclosed
Nodes	28nm	20nm est.	16/12nm est.
Die Size	$300mm^2$	Undisclosed	Undisclosed
Data Precision	INT8/INT16	bfloat16	bfloat16
Performance	92/23 TOPS	45 TOPS	90 TOPS
Memory	8GB DDR3	16GB HBM	32GB HBM
CPU Interface	PCIe3.0 x16	PCIe3.0 x8	PCIe3.0 x8 est.
Power	40W	200-250W est.	200W est.

Table 3: Comparison of TPUs, Data source: Paul Teich, TEARING APART GOOGLE'S TPU 3.0 AI COPROCESSOR [57]. Deep learning is memory centric. New versions have larger memories. The 'bfloat16' is a float point format used only in TPU, which consists of 8 bits exponent and 7 bits mantissa.

2.2.2 *The Economics of the TPU.*

Even though it took long time to design the first TPU, Google quickly released its successors in next two consecutive years. Table 3 shows estimated metrics for three TPU generations.

While the TPU had attained better power efficiency then GPUs, it continued to be improved in the latest generations. The cloud economics greatly benefited from TPU's power efficiency improvements and today, the TPU is available on the Google Cloud via virtual machine instances with relatively low prices.

Table 2 shows the cost of using GPU and TPU. The GPU (Nvidia Tesla V100s) is from AWS (V100s are not yet available on the Google Cloud). Based on the processing ability (images per seconds), we can compute the number of images per $ that can be processed on the different platform. From Table 2 we can see that the Cloud TPU is a clear more cost effective. However, for a certain deep learning model, training is a progress with improved accuracy along time. To reach a certain accuracy, the time needed may vary among these platforms. To compare the cost more reasonably, we assume an acceptable solution at 75.7% for ImageNet [39] (the best accuracy achieved by the GPU implementation), and then we can calculate the cost to achieve this accuracy based on required epochs and training speed.

Figure 4 compares the total cost to achieve this accuracy with GPU and TPU. It is clear that with TPU the total cost is almost halved. Though we are not sure how much is the net profit difference that Google and amazon are making from these cloud services, the TPU clouds are more cost effective in theory.

3 GENERAL ASIC CLOUDS DESIGN METHODOLOGY

We have demonstrated that ASIC Clouds not only increases the performance of the target application but also reduces the TCO. Nevertheless, the high NRE costs of ASIC Clouds and long time-to-market often make it a one-time investment, precluding multiple optimization iterations. In this section, we discuss how can we design an optimal ASIC Cloud with constraints for applications with varying features (e.g, computation bound or memory bound).

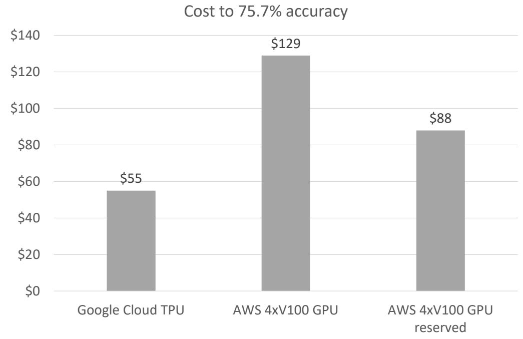


Figure 4: Cost of training with TPU and GPU. Data source: Elmar Haußmann, Comparing Google's TPUv2 against Nvidia's V100 on ResNet-50 [30]. The task is to classify ImageNet dataset into one of 1000 categories, like Hummingbird, Burrito, or Pizza. The dataset consists of about 1.3 million images for training (142 GB) and 50 thousand images for validation (7 GB). The GPU reaches the final accuracy of 75.7% after 84 epochs, while the TPU implementation only needs 64 epochs to reach this accuracy [30].

3.1 General Architecture of an ASIC Cloud

At the heart of any ASIC Cloud is an energy-efficient, high-performance, specialized *replicated compute accelerator*, or *RCA*, that is multiplied up by having multiple copies per ASICs, with multiple lanes of ASICs per server, multiple servers per rack, and multiple racks per datacenter. Work requests from outside the datacenter will be distributed across these RCAs in a scale-out fashion. Figure 5 shows the architecture of a basic ASIC Cloud. Looking inside the racks, each server contains an array of specialized ASICs arranged on a customized printed circuit board along with DC/DC converters . The servers are powered by high-efficiency power supply units (PSU) and contain cooling systems comprised of inlet fans and customized heatsinks on each ASIC. In each customized ASIC, there is a router for off-ASIC communication, a control plane that schedules and distributes off-ASIC workloads across the RCA's, as well as an on-ASIC interconnection network. A control processor or FPGA schedules computation and feeds the ASICs–and DRAMs if applicable–from an off-PCB network .

3.2 Design Metrics and Trade-offs

For scale-out ASIC Cloud servers, the die area, power budget and performance of each individual chip is not critical, so long as the workload's latency, throughput and cost requirements are met . By examining the TCO of the target computation, we can optimize across process nodes and correctly weigh the importance of cost-efficiency, energy-efficiency, and NRE to attain cost savings in an ASIC Cloud workload. The maximum transistors per die metric excludes nodes that do not have sufficient transistor density to fit even a single accelerator. Similarly, the transistor frequency metric serves to filter out process nodes that do not offer the required single-accelerator performance or latency.

Overview. Figure 6 examines various metrics based on data we collected from four sources in order of preference: 1) CAD tool simulations in our lab, 2) disclosure of technical details 3) interviewing industry experts, and 4) using CMOS scaling to interpolate missing data points.

For CMOS scaling, the factor S refers to the ratio of feature widths of two nodes; for example, given 180nm and 130nm, S=180/130=1.38×. Typical scaling factors between successive nodes are often assumed to be S=1.4×. Typically, transistor count increases with S^2, transistor frequency with S, and transistor capacitance (and energy per op at a fixed voltage) decreases with S.

Because of our use of historical and current data rather than predictive scaling theory, our nodes are different than typical scaling theory nodes, reflecting the reality of available process technology. In today's nodes, 40nm has supplanted 45nm, 28nm has supplanted 32nm, and 16nm FinFET has supplanted 20nm.

Although most of the tech node feature widths are spaced by S=1.4×, 65nm and 40nm are spaced by S=1.6×, and 28nm and 16nm are spaced by S=1.75×. Accordingly, we have plotted the data on a log-log plot with the X axis plotting feature width. Thus a straight line with slope of 1 indicates feature-width-proportional scaling. For mask costs, we have standardized on 9 metal layers if the process supports it, and otherwise the maximum number of layers for older processes (i.e. 5 layers for 250nm and 6 layers for 180nm). More metal layers entails more masks, incurring more NRE.

Due to the nature of CMOS scaling, these metrics improve exponentially with more advanced process nodes. At the same time, mask NRE worsens exponentially as nodes advance. The space from 250nm to 16nm spans a **89× range in mask cost**, a **152× range in energy/op**, a **28× range in cost per op/s (558× for non-power density limited designs)**, a **256× range in maximum accelerator size** in transistors, and a **15.5× range in maximum transistor frequency**. Note that the Y axis typically spans two decades of range, but frequency is only slightly more than one decade, and transistor count spans a full three-decades.

Mask Costs. Figure 6-A and Table 6 show mask costs, which range from ~65K for 250nm to almost ~6M for 16nm. Mask cost scaling with feature width actually varies widely, as indicated by the varying slope of the segments. For example, 65nm and 40nm are particularly cheap steps, and 180nm to 130nm is a large step, relative to the previous node. Overall, mask cost multiples are smaller after 90nm than before, possibly because the number of metal layers has stabilized.

Energy per Op. As can be seen in Figure 6-B, energy per op (e.g. CV^2) improvements are markedly different after 90nm. This coincides with the end of Dennard scaling [59] after 90nm. Prior to 90nm, energy improvements were driven by S voltage scaling and by S capacitance scaling, and in 65nm and later, they are driven by S capacitance scaling and only marginal voltage scaling (about 1.04× per node, post-Dennard scaling, as shown in Table 4). *Thus,*

Tech Node (nm)	250	180	130	90	65	40	28	16
Nom. V_{dd} (V)	2.5	1.8	1.2	1.0	1.0	0.9	0.9	0.8

Table 4: Real nominal supply voltages for each tech node.

"ASIC Clouds: Specializing the Datacenter," Magaki, Khazraee, Gutierrez, and Taylor. In *International Symposium on Computer Architecture*, pp. 178-190., June 2016

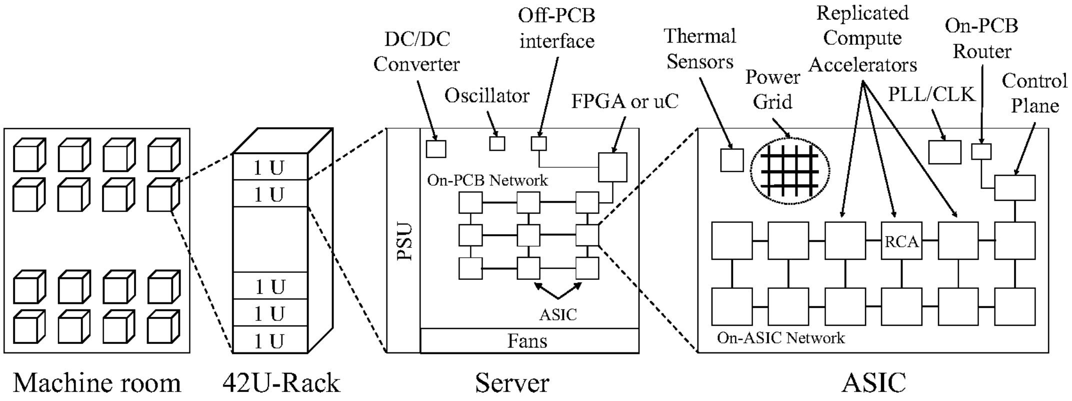


Figure 5: High-Level Abstract Architecture of an ASIC Cloud.

given that energy per op is a major TCO driver, the benefits of nodes after 90nm are much more limited than before 90, when selecting a node for our ASIC Cloud. As a result, the penalty for going to a lower-NRE process is lower than might be implied by the node feature size.

Marginal cost per op/s. Figure 6-C graphs $ per op/s, i.e. the marginal silicon cost of adding computing capacity in a throughput-dominated workload such as typical in scale-out cloud applications. $ per op/s is hurt by exponentially increasing wafer costs, but helped by improvements in wafer size and ops/mm^2 compute density due to transistor frequency and density scaling. Table 6 shows that wafer costs scale approximately with S, but are also related to wafer size. During Dennard scaling, compute density improves as S^3, but below 90nm, it is limited to S by power density. 28nm has higher $ per op/s than 40nm because wafer cost rises faster than usable compute density improves. For applications that are not power-limited in 90nm, scaling continues more as a straight-line continuation of the pre-90nm curve, but bends towards the power-limited case at advanced nodes. In the results section of this paper, many of the accelerators operate the logic at below-nominal Vdd levels (e.g 0.5–0.8V), in order to improve performance within the thermal budgets.

Maximum design size. Figure 6-D graphs the maximum number of logic transistors per die; memories are scaling less well than shown in this graph. Generally speaking, transistors per die mostly places limits on how old a process node can be used before the accelerator does not fit.

Transistor Frequency. Transistor frequency improvements are graphed in Figure 6-E. For post-Dennard nodes that are power-density limited and do not operate at maximal clock rates, this metric still tracks the frequency of SerDes in DRAM controllers and high-speed off-chip interfaces. At older nodes, frequency limits accelerator serial performance, potentially resulting in unsatisfied datacenter latency or Service Level Agreement (SLA) requirements.

3.3 Pareto-and TCO-optimality based design methodology

A classic conundrum since the beginning of energy-efficiency research in computer architecture has been how to weigh energy efficiency and performance against each other. An intermediate solution is the Pareto Frontier analysis, based on Pareto space by the two key metrics: the hardware cost per performance ($ per op/s), and the energy per operation (Watts per op/s, equivalent to Joules per op). To find the most optimal point, TCO analysis is applicable to the space of ASIC Clouds. TCO analysis incorporates datacenter-level constraints including power delivery inside the datacenter, land, depreciation, interest, and the cost of energy itself. The paper applies an improved Barroso et al's [15] TCO model.

We show that joint knowledge and control over datacenter and hardware design allows for the ASIC designers to select the single TCO-optimal point by correctly weighing the importance of cost per performance and energy per op among the set of Pareto-optimal points.

3.4 Practical tools for designing ASIC Clouds

Our prior paper [43] described an ASIC Cloud design space tool, as shown in Figure 7. It takes in a "ball of verilog" that describes the RCA and after extracting critical parameters from RTL-to-GDS tools (area, energy efficiency, frequency), finds the Pareto- and TCO-optimal architecture of an ASIC Cloud for that RCA, including the number of RCAs per ASIC, the numbers of ASICs per lane, the heat sink configuration (materials, fin width, height, and depth) and the fan configuration. The thermals of our systems are validated by simulating airflow through the servers with ANSYS Icepak.

Power density makes a critical connection between an RCA's properties and the number of RCA's that we should place in an ASIC, and how many of those ASICs we should place in an ASIC Server. For example, if power density is high, then less silicon can be placed in a lane within the temperature limits, while divided into more dies. Power density can be tuned by voltage optimization which also determines ASIC Cloud energy efficiency and performance.

Thermal considerations have a great impact on a packaged ASIC's power budget and the server's overall performance. Considering the many ways that ASICs can be arranged on the PCB to optimize thermals, we discovered that duct-style layouts beat the other configurations, employing inexpensive enclosures over each row of ASICs on the PCB gains 93 % improvement in consumable power for the same cooling fans due to much less wasted cool air.

Table 5: CPU Cloud vs. GPU Cloud vs. ASIC Cloud Deathmatch.

Application	Perf. metric	Cloud HW	Perf.	Power (W)	Cost ($)	lifetime (years)	Power/ op/s.	Cost/ op/s.	TCO/ op/s.
Bitcoin	GH/s	C-i7 3930K(2x)	0.13	310	1,272	3	2,385	9,785	20,192
Bitcoin	GH/s	AMD 7970 GPU	0.68	285	400	3	419	588	3,404
Bitcoin	GH/s	28nm ASIC	7,341	3,731	7,901	1.5	0.51	1.08	3.22
Litecoin	MH/s	C-i7 3930K(2x)	0.2	400	1,272	3	2,000	6,360	16,698
Litecoin	MH/s	AMD 7970 GPU	0.63	285	400	3	452	635	3,674
Litecoin	MH/s	28nm ASIC	1,164	3,401	12,620	1.5	2.92	10.8	23.7
Video Transcode	Kfps	Core-i7 4790K	0.0018	155	725	3	88,571	414,286	756,489
Video Transcode	Kfps	28nm ASIC	159	1,654	6,482	1.5	10.4	40.9	87.0
Conv Neural Net	TOps/s	NVIDIA Tesla K20X	0.26	225	3,300	3	865	12,692	8,499
Conv Neural Net	TOps/s	28nm ASIC	235	1,811	2,538	1.5	7.70	10.8	42.6

3.5 ASIC Clouds design examples

Our recent paper [43] examines the design of four types of ASIC Clouds with diverse needs. In this paper we summarize these results. Bitcoin ASIC Clouds require no inter-chip or inter-RCA bandwidth, but have ultra-high power density and have little to no on-chip SRAM. Litecoin ASIC Clouds are SRAM-intensive and have lower power density. Video Transcoding [34] ASIC Clouds require DRAMs next to each ASIC and high off-PCB bandwidth. Finally, our DaDianNao-style [19] Convolutional Neural Network ASIC Clouds make use of on-ASIC eDRAM and HyperTransport links between ASICs to scale to large multichip CNN accelerators.

We evaluated all server configurations spanning the design space for total silicon area per lane, total chips per lane, and all operating voltages from 0.4 up in increments of 0.01V, and pruning those combinations that violate system requirements, e.g., thermals.

In Table 5, we compare the performance of CPU Clouds versus GPU Clouds versus ASIC Clouds for the four applications that we presented. ASIC Clouds outperform CPU Cloud TCO per op/s by 6,270x; 704x; and 8,695x for Bitcoin, Litecoin, and Video Transcode respectively. ASIC Clouds outperform GPU Cloud TCO per op/s by 1057x, 155x, and 199x, for Bitcoin, Litecoin, and Convolutional Neural Nets, respectively.

3.6 When do we go ASIC Cloud?

Given these extraordinary improvements in TCO, what determines when ASIC Clouds should be built? We have shown some clear examples of planet-scale applications that could merit ASIC Clouds. The key barrier is the cost of developing the ASIC Server, which includes both the mask costs (we estimate about $ 1.5M for a 28 nm mask), and the ASIC development costs, which collectively, we term the non-recurring engineering expense (NRE).

We propose the *two-for-two rule.* If the cost per year (i.e. the TCO) for running the computation on an existing cloud exceeds the NRE by 2X, and you can get at least a 2X TCO per op/s improvement, then going ASIC Cloud is likely to save money. Essentially, as the TCO exceeds the NRE by more and more, the required speedup to breakeven declines. As a result, almost any accelerator proposed in the literature, no matter how modest the speedup, is a candidate for ASIC Cloud, depending on the scale of the computation.

The promise of TCO reduction via ASIC Clouds suggests that both Cloud providers and silicon foundries would benefit by investing in technologies that reduce the NRE of ASIC design, including open source IP such as RISC-V, in new labor-saving development methodologies for hardware and also in open source backend CAD tools. With time, mask costs fall by themselves, and in fact older nodes such as 40 nm are likely to provide suitable TCO per op/s reduction, with half the mask cost and only a small difference in performance and energy efficiency from 28 nm.

4 GENERAL ASIC CLOUDS DEPLOYMENT CHALLENGES

The feasibility of an ASIC Cloud for a particular application is directly gated by the ability to manage the Non-Recurring Engineering (NRE) costs of designing and fabricating the ASIC such that it is significantly lower (e.g. 2×) than the TCO of the best available alternative.

In this section, we show that technology node selection is a major tool for managing ASIC Cloud NRE and allows the designer to trade off an accelerator's excess energy efficiency and cost performance for lower total cost. We explore NRE and cross-technology optimization of ASIC Clouds. We address these challenges and show large reductions in the NRE, potentially enabling ASIC Clouds to address a wider variety of datacenter workloads. Our results suggest that advanced nodes like 16nm will lead to sub-optimal TCO for many workloads, and that use of older nodes like 65nm can enable a greater diversity of ASIC Clouds. Although research in accelerators has been widespread, translation of these accelerators into commercial practice has proven challenging for two key reasons: *deployment friction* and *Non-Recurring Engineering (NRE)* costs. In this section, we discuss recent trends that have reduced deployment friction and then examine minimizing all costs required to create and deploy an ASIC accelerator.

4.1 The Friction in deploying ASIC Clouds

We employ the term deployment friction to refer to the difficulty of deploying these accelerator designs into a real-world computing ecosystem. For accelerators that target client devices, deployment of a researcher's accelerator often requires convincing Apple, Intel, or Qualcomm to add the accelerator to their high-volume SoCs, a difficult technology transfer problem with complex social and economic aspects. Beyond the standard organizational barriers, these companies must be convinced that customers will pay extra money to provide sufficient additional profit over the increased cost across a large number of price-sensitive parts. In many cases, emerging applications may not have achieved sufficiently wide-spread use

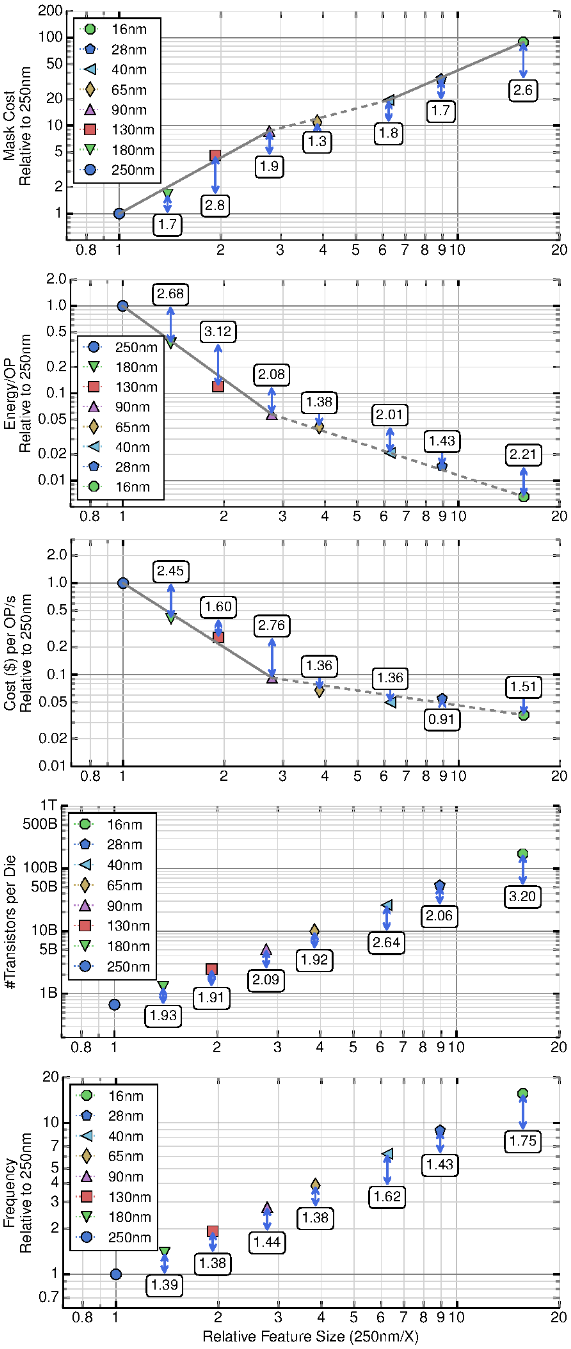


Figure 6: Node Technology trade-offs, normalized to 250nm. #'s indicate multiplicative benefits as node advance. Lines in mask cost indicate different regimes of mask cosk scaling. The dotted lines in energy and cost per op/s graphs indicate the post-Dennard slowdown in voltage scaling.

to make 1-accelerator-to-1-device deployment economically appropriate, eliminating the incentive to place the accelerator on the die. Moreover, the customer may not use the application enough to create a perceivable benefit in terms of battery life or productivity.

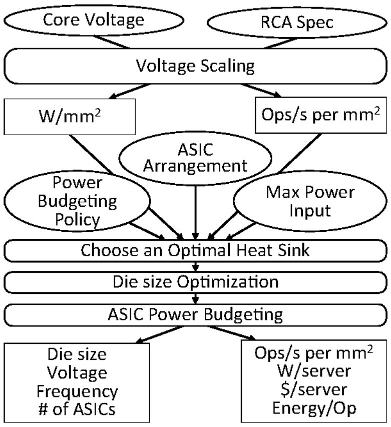


Figure 7: ASIC Server Evaluation Flow. The server cost, per server hash rate, and energy efficiency are evaluated using RCA properties, and a flow that optimizes server heat sinks, die size, voltage and power density.

The Cloud Reduces Deployment Friction. The cloud, on the other hand, provides intriguing possibilities for deployment of discrete accelerators. Because software and hardware are vertically integrated in many cloud contexts, companies like Google, Facebook, Microsoft, Apple, and Amazon can custom-design their hardware–from server to PCB to chip–for recurring workloads that impart significant total-cost-of-ownership (TCO) to their business units.

4.2 NRE Cost breakdown: Mask dominated

NRE is another challenge in deploying ASIC Clouds. Higher NRE not only require large early investment, but also greatly increases the investing risk. These two factors obviously rise the barriers for startups. In this section, we describe our model that incorporates principle components for NRE in ASIC development: mask costs, labor, package design, CAD tool, and IP.

Mask Costs. Table 6 show mask costs, which range from ~65K for 250nm to almost ~6M for 16nm.

Packaging costs. Flipchip package design and tooling costs contribute about $105K to NRE, shown in Table 7.

Labor costs. Labor costs include application-to-architecture design time, frontend development (e.g. Verilog) and testing costs, backend design and verification costs (known as Verilog-to-GDS), IP validation costs (the significant cost of adopting somebody else's IP) and non-ASIC costs like PCB design, system-level interface development, and Cloud API coding. ASIC frontend design mainly involves *IP qualification*, *design specification*, *RTL implementation*, *module integration* and *functional testing*. The backend process consumes human time in *floorplanning*, *power and clock networking*, *placement*, *routing*, *timing closure*, *design rule verification*, and a few more marginal tasks before signing-off the chip layout. System NRE includes the system level code development for interfacing the ASICs with outside world, including firmware for the server's FPGA controller, FPGA code for job distribution across ASICs, and software modifications to an existing cloud to use ASIC Cloud servers. Also, ASIC Cloud servers require a custom PCB design.

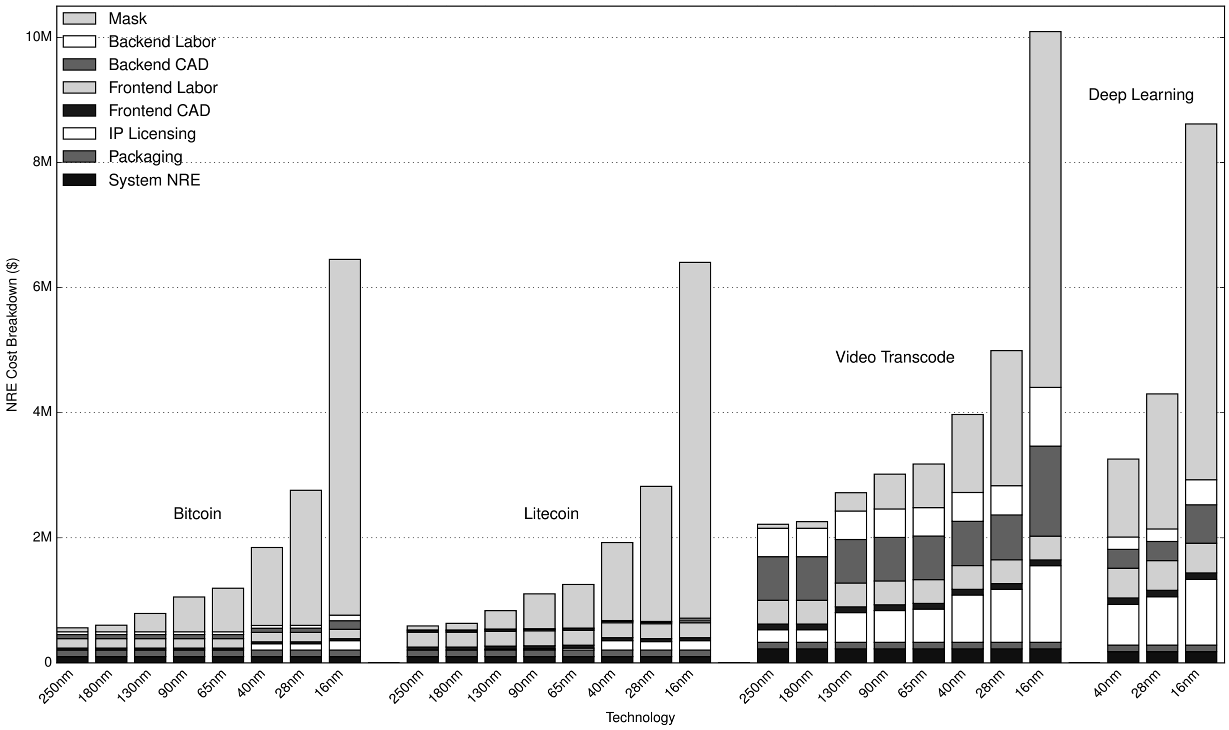


Figure 8: NRE Cost Breakdown Across Tech Nodes. Mask costs rise rapidly for newer technology nodes and become the dominant part of NRE. IP, CAD Tool and Labor Costs are application-dependent but can dominant mask costs in older nodes. Frontend labor and CAD is constant across nodes. IP costs for DDR and PCI-E/HyperTransport for newer nodes rises quickly.

Tech	250nm	180nm	130nm	90nm	65nm	40nm	28nm	16nm
Mask cost ($)	65K	105K	290K	560K	700K	1.25M	2.25M	5.70M
Cost per wafer ($)	720	790	2,950	3,200	3,300	4,850	7,600	11,100
Wafer diameter (mm)	200	200	300	300	300	300	300	300
Backend labor cost per gate ($) [32]	0.127	0.127	0.127	0.127	0.127	0.129	0.131	0.263

Table 6: Wafer and mask costs rise exponentially with process node in early 2017. Backend cost per gate jumps with double-patterning.

Based on our analysis, frontend labor costs do not vary much with technology node, and relate more to design complexity (as measured imperfectly in lines of code, functional blocks, or gates.) For backend labor costs, costs scale with the number of unique design gates being mapped to the die, and by the complexity of the target node. Advanced nodes like 16nm that employ double-patterning suffer an additional multiplier based on greatly escalated back-end design costs. Since the ASIC Clouds employ regular arrays of accelerators on-die connected by a simple NoC (Network on Chip), we assume a hierarchical backend CAD flow that scales with RCA complexity rather than raw instance count on the die. A fixed gate count overhead is considered for I/O and NoC at the top-level of the chip. Frontend and backend labor salary rates as well as top-level overheads are shown in Table 7. 65% overhead is assumed for employee benefits and supplies.

Tool Costs. The tool costs include the frontend tools (e.g. Verilog Simulation and Synthesis), backend tools (e.g. RTL-to-GDS tools like Synopsys IC Compiler or Cadence Innovus), and PCB design tools. Of these tools, the backend tools are by far the most expensive. The model described in [32] gives the total backend labor cost in terms of gates. To calculate the required man-months for backend CAD tools, we divide the backend cost by the backend labor salary.

IP Costs. Each application's IP licensing cost depends on that application's specific IP requirements. Almost all accelerators will need standard cells (e.g. VLSI layouts for the gates, and basic LVCMOS I/O cells) and generator programs for making SRAMs. Typically, these are provided free for nodes at 65nm and older, and cost $100K or so for advanced nodes at 40nm & up. Designs that use fast (> 150 MHz) clocks need an internal PLL. For systems that use DRAM, two IP blocks are required: a DRAM controller, and a DRAM PHY, the mixed-signal block that does high-performance signaling outside the chip. Similarly, for high-speed interfaces like PCI-E or HyperTransport, a controller and PHY IP block are required. Simple applications like Bitcoin may not need any IP beyond the standard cells, while a video transcoder might require a DRAM PHY, and a neural network ASIC Cloud might require a PCI-E or HyperTransport block. These IP costs greatly escalate the NRE of these accelerators. Table 8 shows typical IP licensing costs.

Frontend Labor Salary [2]	$/yr	115K
Frontend CAD Licenses	$/Mm	4K
Backend Labor Salary [2]	$/yr	95K
Backend CAD Licenses	$/month	20K
Overhead on Salary		65%
Top-level gates		15K
NRE, flip-chip BGA package	$	105K

Table 7: Node-independent NRE parameters in San Diego, CA in late 2016. Mm=man-month. Backend Tools are more expensive than the people using them. Flip-chip packages add significant NRE.

IP Cost Correlation with Nodes. In our investigation illustrated by Figure 9, **we have found that IP costs rise rapidly as the technology node increases**, and that the most expensive IP blocks in general are PHY blocks found in PCI-E and DDRs. For 180nm and 250nm, no DDR DRAM blocks are available, and so a free SDR controller suffices. At advanced nodes like 16nm PCI-E and DDR cost almost $1M.

Based on the aforementioned NRE model, we now can estimate the NRE cost for different ASIC Clouds. NRE cost breakdown across nodes and applications is shown in Figure 8. The trend clearly shows that the overall NRE cost rapidly increases as technology node advances and that mask costs for newer nodes become the dominant part of NRE. Labor, tool costs, IP costs and system NRE vary widely between applications but with the exception of backend labor in 16nm and PHY IP, is relatively constant across nodes. Figures 9 and 8 show how IP prices scale across tech nodes.

Tech Node (nm)	250	180	130	90	65	40	28	16
DRAM Ctlr	NA	NA	125	125	125	125	125	125
DRAM PHY	NA	NA	150	165	175	280	390	750
PCI-E Ctlr	NA	NA	90	90	125	125	125	125
PCI-E PHY	NA	NA	160	180	325	375	510	775
PLL	15	15	15	20	30	50	35	50
LVDS IO	7.5	7.5	0	150	90	36	40	200
Standard Cells, SRAM	0	0	0	0	0	100	100	100

Table 8: IP Licensing Costs increase with advancing Technology Nodes. Commonly used IP licensing costs across tech nodes, in late 2016, thousands of USD. Costs generally rise with node, but there are some irregularities.

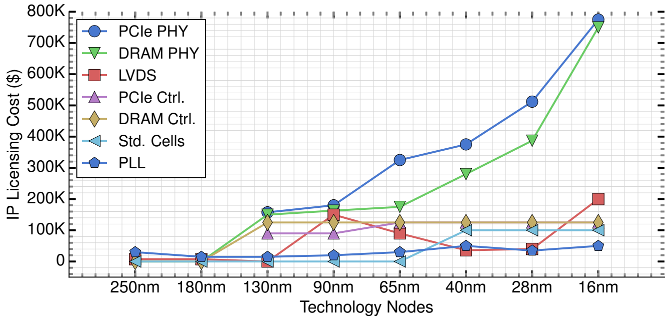


Figure 9: IP Licensing Costs increase with advancing Tech Nodes. High-speed I/O blocks rise exponentially.

4.3 Mask Cost reduction

The rich menu of available nodes means that we have an equally rich tradeoff space that links *mask cost* NREs, *energy efficiency* (i.e. joules per operation), *cost efficiency* (i.e. $ per op/s, a function of frequency, transistor count and wafer cost), *maximum transistor count* per accelerator, and *frequency* (i.e. serial performance per accelerator).

Our data suggests that using the latest node (e.g. 16nm) for an emerging datacenter accelerator can be a mistake. For example, our results show that 16nm was optimal only for TCOs starting at $805M for Litecoin and reaching a geomean of $6.36B across all four applications. Effectively, by choosing an advanced node to do a study, a researcher is setting too high of an NRE on the technology, preventing a prospective company or investor from adopting the technology. Rather, the optimal node must provide *just enough* TCO improvement over the baseline. Moreover, reduced NREs allow an ASIC Cloud to be more agile, updating ASICs more frequently to track evolving software.

5 CONCLUSION

Datacenter and ASIC co-development has sparked the interest for researchers and industry for several reasons. First, the datacenter provides a low-friction deployment surface for ASIC developers; eliminating the worry about varying customer environments (temperature, customs and certifications, making the system 220V/110V compatible, setup guides, tech support, shipping, returns, warranties...) and enabling new kinds of optimizations for cost, energy efficiency and performance. Second, the time to market for an ASIC is significantly reduced if the product does not have to be packaged, troubleshooted and shipped to the customer. Third, quicker time time market means earlier profits which is critically important with highly competative markets such as cryptocurrencies where the network hashrate is increasing exponentially and the bulk of the profits are early on in a machine's life. Finally, meeting strict performance targets in an ASIC design is challenging and must be met before shipping which leads to product delay and a reduction in the ASICs time-to-live.

Bitcoin mining and the TPU are two examples of the emerging class of planet-scale computations being optimized on ASIC clouds. Companies like Apple, Facebook and Google are deploying planet-scale applications like Facebook Live, Siri, and Google Brain. Like Bitcoin, these applications scale-out as the number of people using these systems increases. Ultimately the total cost of ownership (TCO) of the computation becomes so large that it makes economic sense to build specialized ASICs to reduce hardware cost and power consumption. Recent ASIC Cloud research [35, 36, 43] extracts lesson from the history of Bitcoin miners and shows how these same ideas can apply to other planet scale workloads. The future of ASIC Clouds is bright, in part due to the many pioneers who took financial, legal and technical risks to accelerate the Bitcoin hardware ramp and design an entirely new class of planet-scale hardware.

ACKNOWLEDGMENTS

This work was partially supported by NSF Awards 1228992, 1563767 and 1565446, and by the Applications Driving Architectures (ADA) Center, one of six JUMP centers, a Semiconductor Research Corporation program co-sponsored by DARPA.

REFERENCES

[1] May 8, 2016. ASIC Clouds: Specializing the Datacenter . https://csetechrep.ucsd.edu/Dienst/UI/2.0/Describe/ncstrl.ucsd_cse/CS2016-1016.

[2] Retrieved 2016. Glassdoor salaries, 2016. https://www.glassdoor.com.

[3] Retrieved Jun, 2018. Accelerate Genomics Research with the Broad-Intel Genomics Stack. https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/accelerate-genomics-research-with-the-broad-intel-genomics-stack-paper.pdf.

[4] Retrieved Jun, 2018. Amazon EC2. https://aws.amazon.com/ec2/.

[5] Retrieved Jun, 2018. DRAGEN Bio-IT Platform. http://edicogenome.com/dragen-bioit-platform/.

[6] Retrieved Jun, 2018. Ethereum Miner pool. https://ethermine.org.

[7] Retrieved Jun, 2018. Falcon Accelerated Genomics Pipelines. https://aws.amazon.com/marketplace/pp/B07C3NV88G.

[8] Retrieved Jun, 2018. Litecoin Miner pool. https://www.ltcminer.com.

[9] Retrieved Jun, 2018. Microsoft Genomics Acceleration. https://www.microsoft.com/en-us/research/project/genomicsacceleration/.

[10] Retrieved Jun, 2018. OpenCL miner for BitCoin. https://github.com/Diablo-D3/DiabloMiner/blob/master/src/main/resources/DiabloMiner.cl.

[11] Retrieved Jun, 2018. Tensorflow CNN Benchmarks. https://github.com/tensorflow/benchmarks/tree/a03070c016ab33f491ea7962765e378000490d99/scripts/tf_cnn_benchmarks.

[12] Junwhan Ahn et al. 2015. A scalable processing-in-memory accelerator for parallel graph processing.

[13] Jorge Albericio et al. 2016. Cnvlutin: Ineffectual-neuron-free deep neural network computing. In *International Symposium on Computer Architecture (ISCA)*.

[14] J. Barkatullah et al. 2014. GOLDSTRIKETM 1: COINTERRA's FIRST GENERATION CRYPTO-CURRENCY PROCESSOR FOR BITCOIN MINING MACHINES. In *Hot Chips: A Symposium on High Performance Chips (HOTCHIPS)*.

[15] Luiz André Barroso, Jimmy Clidaras, and Urs Hölzle. 2013. The datacenter as a computer: An introduction to the design of warehouse-scale machines. *Synthesis lectures on computer architecture* (2013).

[16] John Beetem et al. 1985. The GF11 Supercomputer. In *International Symposium on Computer Architecture (ISCA)*.

[17] Mahdi Nazm Bojnordi et al. 2016. Memristive boltzmann machine: A hardware accelerator for combinatorial optimization and deep learning. In *International Symposium on High Performance Computer Architecture (HPCA)*.

[18] J. Adam Butts et al. 2014. The ANTON 2 chip a second-generation ASIC for molecular dynamics. In *Hot Chips: A Symposium on High Performance Chips (HOTCHIPS)*.

[19] Yunji Chen et al. 2014. DaDianNao: A Machine-Learning Supercomputer. In *International Symposium on Microarchitecture (MICRO)*.

[20] Yu-Hsin Chen et al. 2016. Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. In *International Symposium on Computer Architecture (ISCA)*.

[21] Ping Chi et al. 2016. PRIME: a novel processing-in-memory architecture for neural network computation in ReRAM-based main memory. In *International Symposium on Computer Architecture (ISCA)*.

[22] Eric Chung et al. Mar 2018. Serving DNNs in Real Time at Datacenter Scale with Project Brainwave. *IEEE Micro* (Mar 2018).

[23] Martin M Deneroff et al. 2008. Anton: A specialized ASIC for molecular dynamics. In *Hot Chips: A Symposium on High Performance Chips (HOTCHIPS)*.

[24] Daichi Fuijiki et al. 2018. GenAx: A Genome Sequencing Accelerator. In *International Symposium on Computer Architecture (ISCA)*.

[25] Boncheol Gu et al. 2016. Biscuit: A framework for near-data processing of big data workloads. In *International Symposium on Computer Architecture (ISCA)*.

[26] Anthony Gutierrez et al. 2014. Integrated 3D-stacked Server Designs for Increasing Physical Density of Key-value Stores. In *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*.

[27] Tae Jun Ham et al. 2016. Graphicionado: A high-performance and energy-efficient accelerator for graph analytics. In *International Symposium on Microarchitecture (MICRO)*.

[28] Song Han et al. 2016. EIE: efficient inference engine on compressed deep neural network. In *International Symposium on Computer Architecture (ISCA)*.

[29] Nikos Hardavellas, Michael Ferdman, Babak Falsafi, and Anastasia Ailamaki. 2011. Toward dark silicon in servers. *IEEE Micro* (2011).

[30] Elmar Hauβmann. Retrieved Jun, 2018. Comparing Google's TPUv2 against Nvidia's V100 on ResNet-50. https://blog.riseml.com/comparing-google-tpuv2-against-nvidia-v100-on-resnet-50-c2bbb6a51e5e.

[31] Yu Ji et al. 2016. NEUTRAMS: Neural network transformation and co-design under neuromorphic hardware constraints. In *International Symposium on Microarchitecture (MICRO)*.

[32] H Jones. 2014. Whitepaper: strategies in optimizing market positions for semiconductor vendors based on IP leverage. *International Business Strategies. Inc.(IBS). Google Scholar* (2014).

[33] Norman P. Jouppi et al. 2017. In-Datacenter Performance Analysis of a Tensor Processing Unit. In *International Symposium on Computer Architecture (ISCA)*.

[34] Chi-Cheng Ju et al. 2015. 18.6 A 0.5 nJ/pixel 4K H. 265/HEVC codec LSI for multi-format smartphone applications. In *International Solid-State Circuits Conference (ISSCC)*.

[35] Moein Khazraee et al. 2017. Moonwalk: NRE Optimization in ASIC Clouds. In *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*.

[36] Moein Khazraee, Luis Vega, Ikuo Magaki, and Michael Taylor. 2017. Specializing a Planet's Computation: ASIC Clouds. *IEEE Micro* (May 2017).

[37] Duckhwan Kim et al. 2016. Neurocube: A programmable digital neuromorphic architecture with high-density 3D memory. In *International Symposium on Computer Architecture (ISCA)*.

[38] Onur Kocberber et al. 2013. Meet the walkers: Accelerating index traversals for in-memory databases. In *International Symposium on Microarchitecture (MICRO)*.

[39] Alex Krizhevsky et al. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*.

[40] Christian Leber et al. 2011. High frequency trading acceleration using FPGAs. In *Field Programmable Logic and Applications (FPL)*.

[41] Kevin Lim et al. 2013. Thin servers with smart pipes: designing SoC accelerators for memcached. In *International Symposium on Computer Architecture (ISCA)*.

[42] Shaoli Liu et al. 2016. Cambricon: An instruction set architecture for neural networks. In *International Symposium on Computer Architecture (ISCA)*.

[43] Ikuo Magaki et al. 2016. ASIC Clouds: Specializing the Datacenter. In *International Symposium on Computer Architecture (ISCA)*.

[44] Junichiro Makino et al. 2012. GRAPE-8–An accelerator for gravitational N-body simulation with 20.5 Gflops/W performance. In *High Performance Computing, Networking, Storage and Analysis (SC)*.

[45] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).

[46] Courtois Nicolas et al. 2014. Optimizing sha256 in bitcoin mining. In *International Conference on Cryptography and Security Systems (CCS)*.

[47] Muhammet Mustafa Ozdal et al. 2016. Energy efficient architecture for graph analytics accelerators. In *International Symposium on Computer Architecture (ISCA)*.

[48] A. Pedram et al. 2016. Dark Memory and Accelerator-Rich System Optimization in the Dark Silicon Era. *IEEE Design and Test* (2016).

[49] Putnam et al. 2014. A Reconfigurable Fabric for Accelerating Large-scale Datacenter Services. In *International Symposium on Computer Architecture (ISCA)*.

[50] Brandon Reagen et al. 2016. Minerva: Enabling low-power, highly-accurate deep neural network accelerators. In *International Symposium on Computer Architecture (ISCA)*.

[51] Ali Shafiee et al. 2016. ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars. *International Symposium on Computer Architecture (ISCA)*.

[52] Stephen Weston. 2011. FPGA Accelerators at JP Morgan Chase. Stanford Computer Systems Colloquium, https://www.youtube.com/watch?v=9NqX1ETADn0.

[53] Michael Taylor. 2013. Bitcoin and the Age of Bespoke Silicon. In *International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES)*.

[54] Michael Taylor. 2013. A Landscape of the New Dark Silicon Design Regime. *Micro, IEEE* (Sept-Oct. 2013).

[55] Michael B. Taylor. 2012. Is Dark Silicon Useful? Harnessing the Four Horesemen of the Coming Dark Silicon Apocalypse. In *DAC*.

[56] Michael Bedford Taylor. 2017. The Evolution of Bitcoin Hardware. *Computer* 50, 9 (2017), 58–66.

[57] Paul Teich. Retrieved Jun, 2018. TEARING APART GOOGLE'S TPU 3.0 AI COPROCESSOR. https://www.nextplatform.com/2018/05/10/tearing-apart-googles-tpu-3-0-ai-coprocessor/.

[58] Yatish Turakhia et al. 2017. Darwin: A Hardware-acceleration Framework for Genomic Sequence Alignment. *bioRxiv* (2017).

[59] Ganesh Venkatesh et al. 2010. Conservation cores: reducing the energy of mature computations. In *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*.

[60] Shijin Zhang et al. 2016. Cambricon-X: An accelerator for sparse neural networks. In *International Symposium on Microarchitecture (MICRO)*.