Check for updates

Chip Measuring Contest

THE BENEFITS OF PURPOSE-BUILT CHIPS

JESSIE FRAZELLE

Alan Kay once said, "People who are really serious about software should make their own hardware." We are now seeing product companies genuinely live up to this value. On August 19, 2021, Tesla showed off Dojo, its new chip used for training neural networks. You might imagine the lead of an article about this something along the lines of, "A company that is not in the business of making chips, made its own chip for its own specific use case, wat!" That part of the announcement was not so shocking because it was something seen before with Tesla and its FSD (full self-driving) computer, with Cisco and its network ASICs, and recently with Apple's M1 chip. In reality the shocking part of the Tesla announcement was not their chip but their humanoid robot, but we'll save that for another article.

Companies such as Tesla and Apple are so serious about their software (and hardware) that they bite off more and more challenging problems lower in the stack to give their customers better products. Additionally, with Moore's law slowing down, chip manufacturers are forced to get more and more creative in their approaches, resulting in diversification among chips. It is an exciting time to be alive when the incumbents known as the chip vendors are being outdone, in the very technology that is their bread and butter, by their previous customers.

It is important to note that it is hard for chip vendors

to stray from general-purpose chips since those are how they can get the most customers and maintain a successful business. That being said, let's dive into some of the interesting bits of these purpose-built chips: the benefits of economics, user experience, and performance for the companies building them.

AI CHIPS

GPUs were originally designed for graphics, hence the name *graphics* processing unit. GPUs are not actually made for neural networks; however, they tend to be used for this solely because they outperform CPUs since they have lots of cores for running computations in parallel. In 2016, Google introduced the TPU (tensor processing unit), which is an ASIC (application-specific integrated circuit) made for neural networks. ASICs made explicitly for neural networks tend to be very good at matrix multiplication and floating-point operations since that is largely what training a neural network is all about. This is why you often see these types of chips advertised by comparing FLOPS (floating-point operations per second). Traditional GPUs focus on calculations for placing pixels; they are also capable of matrix multiplication and floating-point operations but not to the same scale as those made specifically for neural networks.

If you are doing any complex work with neural networks, you have only a few good options for compute. Traditionally, the champion in this space has been Nvidia's A100. A company like Tesla, that competes directly with Google's self-driving car experiments, likely does not want its data in Google's cloud. So the A100 is its only

option. The A100 comes at a steep price, and Nvidia seems to take advantage of its domination in this space. Because of Nvidia's high margins, Tesla could get better unit economics and performance by making its own chips. Because of the cost of designing the chip, building the software, manufacturing, and maintenance, however, Tesla's strategy is likely less a result of economics and more because of vertical integration and the performance benefits of designing to its specific use case.

Startups such as Cerebras, Groq, and Graphcore have entered the space as well. The dominant public opinion in this space seems to be, "Can anyone please compete with Nvidia?" [youtube.com] Chips tend to be made specifically for either training or inference, or both (denoted as general-purpose here).

Training is the process of developing a neural network based on examples. Training neural networks is memory intensive since backpropagation requires storing activations of all intermediate layers; therefore, chips made for training tend to have much more memory.

Inference is like production in that data is fed to a model in order to get a prediction. Inference of models has strong latency requirements because you want to get a prediction as fast as you can. For self-driving cars, a slow prediction could mean the difference between life and death. Tesla's FSD computer is made for inference (it is in your car while you are driving, predicting what your car and other cars should do), while the Dojo D1 chip is made for training.

There is quite a variety of different names for ASICs that are best suited for neural networks. Google calls its ASIC a TPU; Nvidia refers to the A100 and others as GPUs;

Groq uses the term TSP (tensor streaming processor); Graphcore invented the term IPU (intelligence processing unit); and Apple goes with NPU, for neural processing unit. (It would be nice to standardize on the Apple term only because it uses the word *neural*, which implies neural networks, instead of everyone coming up with their own names, but what do I know?)

Table 1 compares the most recent generations of all these chips. Note that all the numbers in the table are taken from marketing materials, not from actual benchmarks.

Tesla's Dojo training tile packaging leverages TSMC's new InFO_SoW (integrated fan out system on wafer) technology [ieee.org]. Electrical performance, as well as cost and yield, benefit significantly from this packaging. InFO_SoW provides the wafer-scale benefits of low-latency chip-to-chip communication, high-bandwidth density, and low PDN (power distribution network) impedance for greater computing performance and power efficiency with none of the downsides. Those familiar with manufacturing chips might be wary of yield with a wafer-scale chip like Cerebras. For its WSE (wafer-scale engine) and WSE-2 processors, Cerebras disables whole rows and columns that contain broken tiles, which means there are no problems with yield.

The Dojo training tile consists of 25 D1 chips, which makes it easier to compare to the Cerebras WSE-2. The main difference to note between WSE-2 and the Dojo training tile is that WSE-2 is a single wafer. The 25 D1 chips that make up a training tile can be chosen to ensure all the chips are manufactured properly

TABLE 1: **CHIP COMPARISON**

NAME	CEREBRAS WSE-2[1]	DOJO TRAINING TILE[2]	DOJO D1[2]	NVIDIA A100 80GB SXM[3]	GOOGLE CLOUD TPU V4I[4]	GROQ TSP[5]	GRAPH-CORE COLOSSUS™ MK2 GC200 IPU[6]	TENS-TORRENT GRAY-SKULL E300 PCIE[7]
SIZE	46,225 mm^2	< 92,903 mm^2	645 mm^2	826 mm^2	< 400 mm^2		823 mm^2	
CORES	850,000	35,400	1,416[8]	6,912 CUDA + 432 Tensor	1	1	1,472	
BF16/CFP8[9]		9 PFLOPS	362 TFLOPS	312 TFLOPS[10]	138 TFLOPS[11]			
FP64				9.7 TFLOPS[12]				
FP32		565 TFLOPS	22.6 TFLOPS	19.5 TFLOPS			64 TFLOPS	
FP16				312 TFLOPS		250 TFLOPS	250 TFLOPS	
INT8				624 TOPS	138 TOPS	1 POPS		600 TOPS
ON-CHIP MEMORY (SRAM)	40 GBs	11 GBs	442.5 MBs[13]	40 MBs[14]	151 MBs[15]	220 MBs	900 MBs	
DRAM				80 GBs[16] HBM	8 GiBs HBM			16 GBs[16]
MEMORY BANDWIDTH[17]	20 PBs/sec	10 TBs/sec	10 TBs/sec	2.039 TBs/sec	614 GBs/sec	80 TBs/sec	47.5 TBs/sec	200 GBss/sec
FABRIC BANDWIDTH	27.5 PBs/sec[18]	36 TBs/sec	4 TBs/sec	600 GBs/sec[19]	100 GBs/sec	500 GBs/sec[20]	320 GBs/sec	
MAX TDP*	20kW / 15kW	15kW	400W	400W	175W			300W
PROCESS	7nm	7nm	7nm	7nm	7 nm	14 nm	7nm	
TRANSISTORS	2.6 trillion	1.250 trillion	50 billion	54 billion	16 billion	26.8 billion	59.4 billion	
MADE FOR	general-purpose	training	training	general-purpose	general-purpose	Inference	general-purpose	general-purpose
PRICE	$2-3 million+[21]			$20,000+				$2,000

***** Thermal Design Power

1 https://cerebras.net/chip/
2 https://www.youtube.com/watch?v=j0z4FweCy4M
3 https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/nvidia-a100-datasheet-us-nvidia-1758950-r4-web.pdf
4 https://ieeexplore.ieee.org/document/9499913
5 https://groq.com/technology/
6 https://www.graphcore.ai/products/ipu
7 https://tenstorrent.com/grayskull/
8 354 units per chip * 4 cores per unit.
9 Configurable floating point 8 (CFP8) only applies to Tesla's Dojo.
10 624 TFLOPS with sparsity. Meaning you can get two times the maximum throughput of dense math for matrices of numbers that includes many zeros or values that will not significantly impact a calculation. Sparsity tends to only be useful for inference.
11 https://www.hpcwire.com/2021/05/20/google-launches-tpu-v4-ai-chips/ Google claims 4096 chips per pod and 1 pod has over one exaflops of floating point performance.
12 The A100s are the only one to advertise this number, some chips might not support, and some might not advertise support.
13 354 units per chip * 1.25 megabytes per functional unit.
14 https://images.nvidia.com/aem-dam/en-zz/Solutions/data-center/nvidia-ampere-architecture-whitepaper.pdf
15 Google advertises this as MiB, but we convert to megabytes for easy comparison to the other numbers.
16 Their website says gigabytes (GB) but likely this is actually gibibytes (GiB).
17 For chips with high bandwidth memory (HBM) / DRAM, this refers to the bandwidth to that memory. Whereas for chips without DRAM/HBM, this refers to the SRAM bandwidth. For chips with both, the SRAM bandwidth is not listed only the HBM/DRAM bandwidth, but you can assume typical SRAM bandwidths.
18 The Cerebras marketing material shows this as 220 petabits, but converted to 27.5 petabytes for comparison to the other numbers.
19 With NVIDIA's NVLink. This is half-duplex, meaning it supports either 600 GB/s out of the chip or into the chip but not both.
20 This is half-duplex.
21 This number is based on the CS-2 systems.

without defects. A single wafer presents more risk of a defect or manufacturing error, but Cerebras claims this is not a problem [cerebras.net]. As shown in table 1, Cerebras clearly overshadows the other chips in terms of bandwidth *because* it is wafer-scale.

Most of these chips integrate into machine-learning frameworks such as TensorFlow and PyTorch with a single line of code. This makes it easy for developers to change the underlying hardware. Some of the newer chips from startups (Graphcore, Groq, and others) are a bit behind in this regard but have roadmaps to get there. Outside the major frameworks, software integration for these specialized chips is a bit more limited, making traditional GPUs more appealing for workloads outside this scope.

BENCHMARKS

Table 2 shows the results of running Andrej Karpathy's minGPT [github.com] and Google's AutoML EfficientDet [github.com] on a few different accelerators in the cloud. (Google's TPU requires a patch since minGPT works only on CPUs or Nvidia's CUDA [github.com] [github.com].) The minGPT results include both the time to train the model and run a single prediction. These are the notebooks in the minGPT repository: `play_math`, `play_image`, and `play_char`. The EfficientDet numbers are only inference because the models are pretrained. End-to-end latency measures from the input image to the final rendered new image, which includes image preprocessing, network, postprocessing, and NMS (non-maximum suppression).

If you are looking to buy a chip like Tesla's, the closest in architecture is Cerebras. Tesla is not the only company to

TABLE 2: **BENCHMARKS**

CLOUD PROVIDER	AWS	AZURE	GCP	GCP	GCP	GCP	GCP
TYPE	p4d.24xlarge[1]	Standard_ ND96asr_ v4[2]	v3-8[3]	v3-32[4]	v3-64[4]	a2-high gpu-8g[5]	a2-high gpu-16g[5]
ACCELERATOR	8 NVIDIA A100s (40GB HBM2)	8 NVIDIA A100s (40GB HBM2)	4 TPU v3 (8 cores)	16 TPU v3 (32 cores)	32 TPU v3 (64 cores)	8 NVIDIA A100s (40GB HBM2)	16 NVIDIA A100s (40GB HBM2)
CPU	96 3.0 GHz 2nd Generation Intel Xeon Scalable (Cascade Lake)	96 2nd-generation AMD Epyc	96 2.0 GHz Intel Xeon	64 2.0 GHz Intel Xeon	128 2.0 GHz Intel Xeon	96 2.0 GHz Intel Xeon	96 2.0 GHz Intel Xeon
ACCELERATOR MEMORY	320 GB HBM + 320 MB SRAM	320 GB HBM + 320 MB SRAM	137[6] GB	550[7] GB	1.10[8] TB	320 GB HBM + 320 MB SRAM	640 GB HBM + 640 MB SRAM
HOST MEMORY	1237[9] GB	966[10] GB	256[11] GB	256[11] GB	256[11] GB	680 GB	680 GB
COST PER HOUR	$32.77	$28	$8 + cost of VM ($1.35) = $9.35	$32	$64	$23.47[12]	$46.94[13]
PLAY _ MATH TIME	Couldn't get quota	1m 47.854s	Too long to care	9m 19.873s	Couldn't get quota	1m 54.273s	3m 55.344s[14]
PLAY _ IMAGE TIME	Couldn't get quota	46m 0.339s	Too long to care	broke the cluster	Couldn't get quota	48m 43.917s	67m 54.672s
PLAY _ CHAR TIME	Couldn't get quota	9m 45.164s	Too long to care	broke the cluster	Couldn't get quota	10m 21.712s	21m 25.199s
EFFICENTDET NETWORK LATENCY TIME	Couldn't get quota	-[15]	0.074245587 10000043		Couldn't get quota	0.14675208 45999985	0.133794982 50000096
EFFICENTDET NETWORK LATENCY FPS*	Couldn't get quota	-	13.468813959 988125		Couldn't get quota	6.81421325 445356	7.47412183 4127769
EFFICENTDET END-TO-END LATENCY TIME	Couldn't get quota	-	0.08260749 860000374		Couldn't get quota	0.08342655 909999622	0.08533461 209999586
EFFICENTDET END-TO-END LATENCY FPS	Couldn't get quota	-	12.105438 573344657		Couldn't get quota	11.986590 490941696	11.718574 90754727

* Frames per second

1 https://aws.amazon.com/ec2/instance-types/
2 https://docs.microsoft.com/en-us/azure/virtual-machines/nda100-v4-series
3 https://cloud.google.com/tpu/docs/types-zones
4 https://cloud.google.com/tpu/pricing#pod-pricing
5 https://cloud.google.com/compute/docs/gpus
6 128 GiB to GB
7 512 GiB to GB
8 1TiB to TB
9 1152 GiB to GB
10 900 GiB to GB
11 https://cloud.google.com/compute/docs/general-purpose-machines 64 GB * 4
12 https://cloud.google.com/compute/gpus-pricing $2.933908 per GPU * 8
13 https://cloud.google.com/compute/gpus-pricing $2.933908 per GPU * 16
14 I think these are slower since we are doing more memory transfers across different hardware pieces and we didn't have enough training data to make the new threads worth the cost of memory transfers for them.
15 Didn't test but could be considered similar to GCP's 8 A100s.

dip its toes into the water of building its own chips for its own use cases. Let's take a look at Apple's M1.

THE APPLE M1

Apple has created not only a CPU but also a GPU and a bunch of other accelerators making up the SoC (system on a chip) known as M1. In addition to the CPU and GPU,

the M1 SoC includes an image processing unit used to speed up common tasks done by image processing applications; digital signal processor, which handles more mathematically intensive functions than a CPU (for example, decompressing music files); neural processing unit used in high-end smartphones to accelerate AI (artificial intelligence) tasks (for example, voice recognition and camera processing); video encoder and decoder to handle the power-efficient conversion of video files and formats; secure enclave for encryption, authentication, and security; and a unified memory system. Each of these components is designed for the workloads that most Mac users perform. By making its own chips, Apple does not need to rely on the general-purpose chips it was previously buying from Intel and can integrate its hardware fully into its software, making for a complete experience.

As a matter of fact, Apple has now surpassed the capabilities of Intel's fabrication plants (fabs). The M1 uses TSMC's 7nm process, which Intel has yet to catch up to (fabs are covered in depth later in this article). As described in my previous article, "Chipping away at Moore's Law" [acm.org], the smaller the transistor, the less power is required for a chip to function. For Apple, this means better battery life for its devices and power savings for its desktops.

UNIFIED MEMORY SYSTEM

A huge gain in M1 performance over that of general-purpose chips comes from the unified memory system. This allows the CPU, GPU, and other processing units in the SoC to share the same data in memory. General-purpose chips

tend not to do this since they all use some different form of interconnect that does not allow for it. With unified memory, when the CPU needs to give data to the GPU, the GPU can take it from the same bits of memory; it does not need to be copied to the GPU's memory first [eclecticlight.co].

Because RAM is directly embedded in the SoC, an upgrade to more memory is not possible (though that hasn't been possible for quite some time with Apple computers since previously RAM was soldered to the board itself).

RISC

The M1 is ARM-based, meaning it is a RISC (reduced instruction set computer) architecture. The Intel chips Apple used previously were x86, a CISC (complex instruction set computer) architecture. This switch is important to note for a few reasons. One question Apple had to answer was if, by switching architectures, it would make changes that broke the programs its user base runs. For this reason, Apple introduced an emulator known as Rosetta, which enables a Mac with M1 silicon to use apps built for a Mac with an Intel processor.

Switching from x86 to ARM was not Apple's first rodeo in switching instruction set architectures. From 1984 to 1994, Apple predominantly used Motorola's 68x CISC series processors. In 1994, it switched to the PowerPC RISC series processors. In 2006, it moved to Intel's x86 processors, followed in 2020 with the switch to its own ARM RISC processors [chipsetc.com]. While Apple likely had the *courage* [theverge.com] to make the switch sans experience, it also had the experience to back it up.

RISC architectures have fewer instructions but are

more like Legos: They have all the building blocks for the complex instructions a CISC architecture provides, while also having the flexibility to build whatever the user wants. In a RISC-based system, since there are fewer instructions, more of them are required to do complex tasks; however, processing them can be more efficient. For a CISC-based architecture, it is harder to be as efficient because of the number of instructions and their complexity. (Intel started marketing its processors as RISC by adding a decoding stage to turn CISC instructions into RISC instructions [medium.com]. The advantages of RISC persist because of the fixed length; CISC still has to figure out the length of the instructions.) Using a RISC architecture leads to better power efficiency and performance.

One design detail of the M1 processor to point out is the large number of encoders and decoders. This can be accomplished only with a RISC-based architecture because of the fixed-length instructions. CISC-based architectures have variable-length instructions and lots of complex instructions. It is a bit of a meme that no one knows all the instructions available in x86 [twitter.com], but there are ways of discovering hidden instructions [github.com]. The fixed length of instructions means that RISC-based architectures require a simpler decode leading to less circuitry, heat, and power consumption.

The M1 takes advantage of OoOE (out-of-order execution) as a way to execute more instructions in parallel without exposing that capability as multiple threads. While you might be thinking, [yawn] "Intel and AMD do that as well," there is a core difference with the M1 chip. For OoOE to spread its wings and fly, a large buffer of

micro-operations is needed; then the hardware can more easily find instructions to run in parallel. Decoders convert the machine-code instructions into micro-ops to pass off to the instruction buffer. Intel and AMD processors typically have four decoders. M1 has eight decoders and an instruction buffer three times larger than the industry norm. This means the M1 processor can more easily find instructions to run in parallel.

Now you might be thinking, Why don't AMD and Intel add more decoders? Because CISC-based architectures have variable-length instructions, it is nontrivial for the decoders to split up a stream of bytes into instructions because they have no idea where the next instruction starts. CISC decoders have to analyze each instruction to understand how long it is. AMD and Intel deal with this by brute force. They attempt to decode instructions at every possible starting point, making the decoder step too complex to add more decoders.

It seems like a no-brainer for Apple to build its own processors in terms of user experience, economics, and performance. Not only has it made an efficient CPU, but all the other specialized chips included in the SoC are based on the workloads of Mac users. Apple can integrate all the specialized chips into its software and create nice user experiences for its customers. It has definitely blown Intel out of the water in making better chips for its users and is freed from the obligation of giving Intel a cut of its margins.

FOUNDRIES

If you are an Apple, Tesla, or other "fabless" company (one without its own fabrication plant) that has designed its

own chip, where do you go to have it manufactured? Well, TSMC, of course. TSMC is the trusted fab with advanced processes such as 3nm/5nm/7nm to make these chips. Even Intel uses TSMC instead of its own fabs for some of its most advanced chips. Apple, Tesla, Intel, and AMD must compete for capacity at TSMC. Samsung has processes for 5nm and 7nm, but TSMC appears to outperform Samsung in yield, cost, and density [semiwiki.com], making TSMC the trusted fab among the big-name customers. Tesla does use Samsung for its FSD chip and TSMC for Dojo.

Intel has plans to make more advanced chips and even sell the capacity at its foundries to customers such as Apple, but history is not in its favor [theverge.com]. Intel is still trying to get the 7nm process up and running as TSMC works on 3nm. Customers like Apple aren't interested in Intel's 12nm or 14nm processes; they are looking for 3nm or smaller. Will Intel be able to catch up?

It's important to understand that the name of the process (5nm, 7nm, etc.) has become more of a marketing term than a description of the transistor size. Traditionally, naming came from the L_{eff} (the minimum effective length of a transistor channel). When comparing processes, it is better to compare the density of the transistors. For example, Intel claims its unproven 7nm process is comparable in density to TSMC's 5nm process [hardwaretimes.com], should Intel get the process up and running. This might help its odds of catching up.

Interestingly, Intel CEO Pat Gelsinger stated during an investor briefing [intc.com] on March 23, 2021, that the company foresees Apple as a future customer of its foundries, while simultaneously running a series of

advertisements that were anti-Apple [youtube.com]. Ironically, the ads poke at features of PCs versus Apple computers that have nothing to do with the underlying processors, leading to some funny YouTube comments. Overall public opinion on the ads was not in Intel's favor and might have actually given AMD a marketing boost.

Suppose, however, that the global shortage of processors and fab capacity continues and Intel manages to catch up to TSMC. In that case, lots of customers would undoubtedly be relieved that there is more than one fab that can be trusted to manufacture advanced chips. Intel has a long way to go to catch up, however, while TSMC is investing $100 billion in its own expansion [bloomberg.com].

EXTREME ULTRAVIOLET LITHOGRAPHY

EUV (extreme ultraviolet) lithography is used to etch the tiniest nanoscopic features into silicon wafers with light. One of the early limitations of EUV lithography was that pellicles were not ready. A pellicle is a thin, transparent membrane that protects an expensive photomask from particles falling on it during the chip production flow. If a particle were to fall on the photomask, the scanner could print repeating defects on the wafer. This would have a catastrophic impact on yield, not to mention that EUV photomasks are priced around $300,000 [semiengineering.com]. (ASML makes the $150 million EUV machines that power the leading-edge manufacturing of chips. Intel, Samsung, and TSMC have all invested in the company.)

As a result of these limitations, Intel decided to walk away from EUV and try to develop in a different direction. TSMC and Samsung moved forward with EUV despite the

lack of pellicles and came up with their own solutions for the problem. TSMC also has an advantage in that Apple, Qualcomm, and AMD's 7nm designs have a relatively small die size. Photomask dimensions can be around 20 times those of the resulting EUV die; however, the masks for those customers' ICs (integrated circuits) are still relatively small. Unfortunately, Intel is still on large monolithic dies, so an attempt to use any pellicle-less EUV solution would likely end in terrible yields. Intel had to either change its die size, requiring massive architecture changes, or wait for pellicles.

This is why Intel got left behind TSMC and Samsung in terms of advanced processes and EUV. Samsung was the first to get EUV into the production of its 7nm process [semiwiki.com], with TSMC following soon after. Samsung seems to have suffered from yield problems [semiwiki.com], perhaps as a result of trying to do EUV without pellicles. In July 2020, TSMC had manufactured one billion 7nm chips using EUV [tsmc.com]. It wasn't until March 2021 that pellicles were ready, finally allowing Intel to consider using EUV [semiengineering.com].

THE FUTURE

Not only are general-purpose chips getting better, but also multiple companies that previously were not in the business of making chips are now making their own. Doing so seems to pay dividends in terms of user experience, economics, and performance. It will be interesting to see who joins this club next. Long live the engineers who are so serious about software that they make their own hardware. Technology is better off because of it.

Acknowledgments

Huge thanks to James Bradbury, Ben Stoltz, Todd Gamblin, Nils Graef, Ed West, and Thomas Steininger for their feedback on this article.

Jessie Frazelle *is the cofounder and chief product officer of the Oxide Computer Company. Before that, she worked on various parts of Linux, including containers, as well as the Go programming language.*

Copyright © 2021 held by owner/author. Publication rights licensed to ACM.