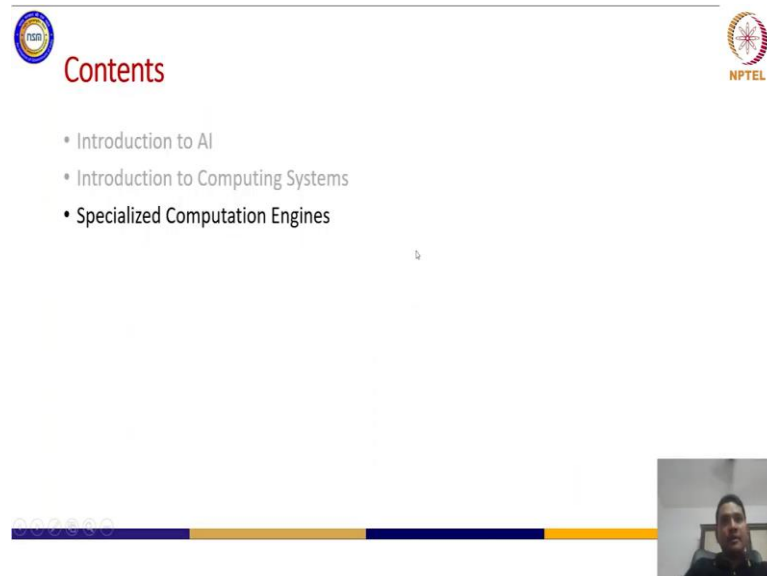


Applied Accelerated AI
Dr. Satyajit Das
Department of Computer Science and Engineering
Indian Institute of Technology, Palakkad

Lecture - 02
Introduction to AI Systems Hardware Part-2


(Refer Slide Time: 00:40)



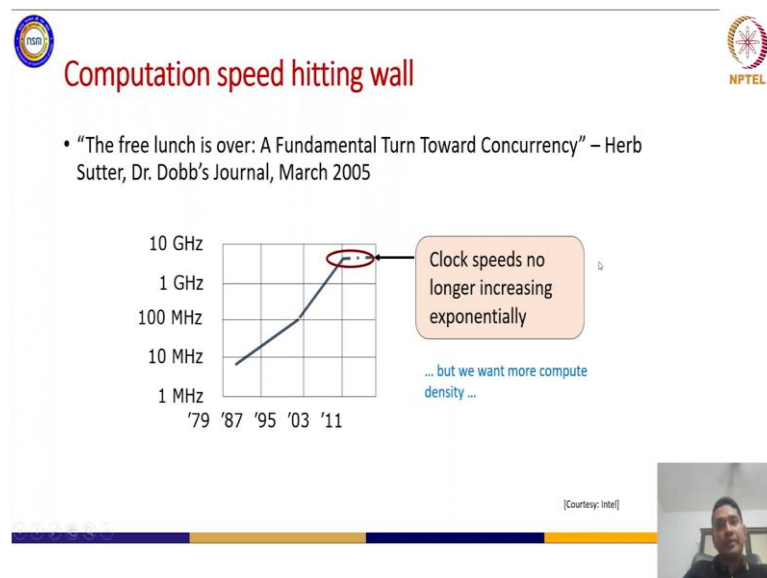
Contents

- Introduction to AI
- Introduction to Computing Systems
- Specialized Computation Engines

Progress bar: 00:40:00



(Refer Slide Time: 00:41)



Computation speed hitting wall

- “The free lunch is over: A Fundamental Turn Toward Concurrency” – Herb Sutter, Dr. Dobbs’s Journal, March 2005


Graph showing clock speed (MHz to GHz) vs. year ('79, '87, '95, '03, '11). The curve shows exponential growth until around 2011, where it plateaus.

Clock speeds no longer increasing exponentially

... but we want more compute density ...

[Courtesy: Intel]

Progress bar: 00:41:00



Now, if you see from the computation engine so, processors ok; so, is the basic computation engine. And, as you know that you cannot increase the clock speed anymore

due to the end of Dennard scaling and the power wall is heating and that's why you cannot get more than a particular fixed clock frequency that is available nowadays in the processors or computation engines.

So, how we will get much more compute density? So, at in terms of speed we want to increase the density as well as computation, because you have to have the flexibility of including more number of transistors in a chip. So, that is fine and in terms of clock speed also you cannot increase the clock speed anymore. So, compute density you cannot increase after a certain limit.

(Refer Slide Time: 01:38)

The slide is titled "How can we accomplish high computational density" in red text. It features a list of bullet points and a diagram of an HSA Accelerated Processing Unit.

- Through parallel-computing...
- Attempt to speed solution of a particular task by
 - Dividing task into sub-tasks
 - Executing sub-tasks simultaneously on multiple processors and
 - Specialized tasks in **accelerators**

The diagram, titled "HSA Accelerated Processing Unit", shows a central "Dual Core x86 Module" connected to "L2 Cache" and "DDR3 Controller". A large red arrow labeled "GPU" points from the cache area to a green "GPU" block. To the left, a gear icon is labeled "Data Parallel Workloads" and "Serial and Task Parallel Workloads". The bottom right corner of the diagram area includes the text "[Courtesy: AMD]".

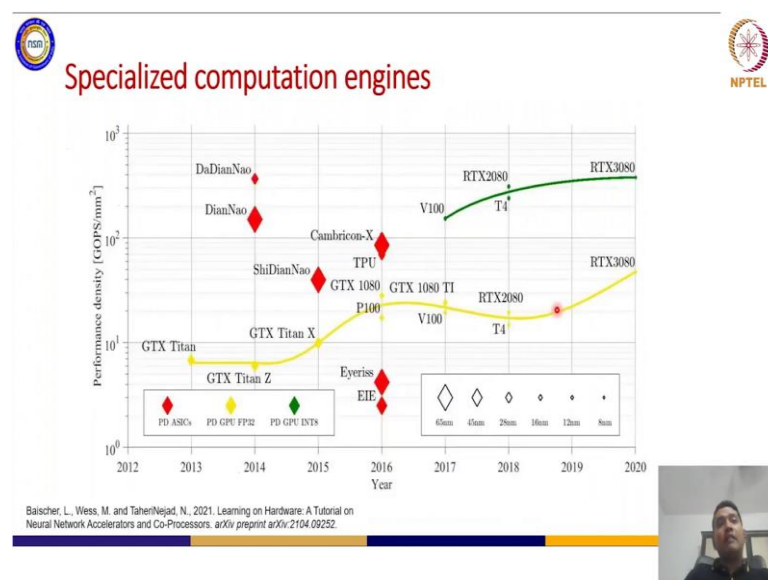
So, what the traditional systems or the modern systems are going towards is that this heterogeneous computing. So, where you have the serial tasks or mostly the parallel workloads ok which are not that much of data intensive; they are being controlled or they are being executed by the processor itself. So, that might be multi-core processors, single core processor. So, in this particular figure you are seeing that if it has a dual core processor.

Now, also in addition to processing engines to get the data parallel workloads executed, we have specialized computing engines and these specialized computing engines are called accelerators. Of course, you will see several accelerators that are being used for particularly AI benchmarks nowadays, but in this slide you can see that one GPU is there to handle these data parallel workloads right.

And, one such accelerator or specialized system that you have seen in the last slide is Cerebras wafer scale engine right. So, now, the tradition is what divide your task into different subtasks and of course, this the data parallel workloads. So, mostly this AI benchmarks that we are talking about. So, from the AI benchmarks for perspective, these AI benchmarks or AI algorithms those will be executed by these accelerators.

So, that is why we are talking about GPUs and because of what that you will see in the subsequent slides. But, the main important thing is that we have now accelerators in the system in the computing system with along with the processors ok. So, that is the main take away from this slide. So, now, what kind of accelerators that are available nowadays.

(Refer Slide Time: 02:58)



So, we talk about specialized computation engines for AI benchmarks right and over the years that you can see here this graph shows the trends from 2012 to 2020 and you can see now different computation engines ok. So, we will discuss them very briefly here. So, what we have? We have ASICs; so, ASICs are ASIC engines or Application Specific Integrated Circuits ok.

So, these ASICs are mostly specialized like highly specialized only for the AI benchmarks and also we have GPUs available in this graph. Now, GPUs can give you the flexibility to run both AI benchmarks as well as let's say video or graphics

benchmarks as well ok. So, now, you can ask you can just try to realize like how much generalized way of computing that can happen.

So, you have the processors which are very generalized. So, maybe the you can say that general purpose computing engine like you can do everything in the computing in your processors. Now, you have ASICs which are highly specialized only for AI benchmarks. Here we will talk about AI based ASICs of course, for any other application domain you can have different ASICs as well.

And, we have GPUs which are Graphics Processing Unit, graphical processing unit. So, which has now nowadays have the flexibility of ah accelerating your AI benchmarks as well, while that we will see in the coming slides. But, you can see here what are the things available here. So, ASICs, GPUs with FP32 so, what is FP32? That is Floating Point 32.

Now, this is data type. So, now data type and the accuracy that you have seen for the AI benchmarks have very close relations ok. So, that relationship we will talk about and you have your GPU INT also 8, INT 8 means 8 bit integer units ok, GPUs with 8 bit integer units, GPUs with 32 bit floating point units. So, 32 bit floating point units means single precision floating point units, you can have double precision floating point units as well which will be then 64 bits right.

And, these ASICs are a highly customized bit level implementation of computing systems for AI benchmarks. And, that is why you can see that the performance density is much higher in these cases ok; because they are only specialized in running AI benchmarks only for given data type or given precision ok. So, their performance density or GOPS/mm² performance per area is much more higher.

GPU is a much more generalized in terms of it can both accelerate your AI benchmarks as well as your video processing as well, graphics processing. And, it has almost closer or almost similar of performance density that is being achieved nowadays and the trend you can see right. So, this yellow line is the trend for your GPU FP32 which is almost generalized GPUs that you can get in the market nowadays.

And, very few are; so, basically if you see that integer 8 so, basically these are with having these GPUs with RTX2080 T4 V100. So, these are also having the flexibility to

run your fixed point 8 bit right units. Now, what is the relationship I was talking about between the accuracy of the benchmarks, AI benchmarks and the bit precision? Ok. So, the more you have or more precision available in the computation for your AI benchmarks, the accuracy will be higher. So, this is the simple relationship.

Now, how you want to get higher computing density? You can reduce the size of the feature size feature right. So, you can in the in this box here you can see that different sizes of feature map that is available. Now, you want to have much more GOPS/mm² ok. So, that is you want to accommodate much more compute or you want to achieve much more complete density available for your systems.

So, you will go for a lowering your feature size which is let us say 7 nanometer or 8 nanometer that is available that is CMOS technology that is being used to manufacture these chips like a V100 series of NVIDIA GPU. And, also around 28 nanometer is being used for this DaDianNao. This is one ASIC based version of DianNao which was published in the year of 2014.

And, in 2016 TPU was published ok; so, first version of TPU. So, this is Tensor Processing Unit published by Google and that is also ASIC based ok and you have Cambricon, Eyeriss; this was published from the research group of MIT, EIE. So, several ASIC based implementations are there that you can see and their uses they use several feature map size. So, basically the size of this diamonds specify what kind of feature map they are using.

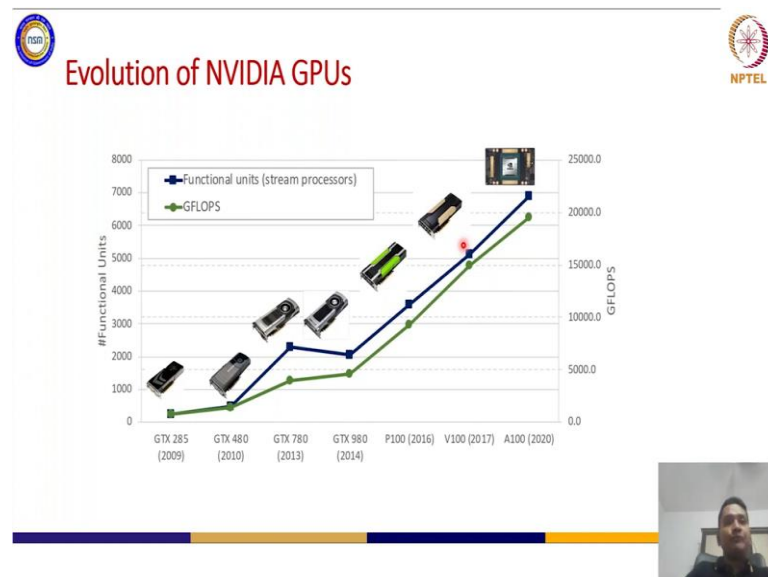
Now, with decreasing the feature maps; that means, you can accommodate more compute units inside your chip right so; that means, you can increase the performance density, GOPS/mm² right. So, how far you can go? Of course, you cannot go beyond certain limit. So, feature map size or technology scaling you cannot do let's say beyond 1nm nowadays you cannot go beyond that.

So that means, there is a certain level of performance that you can achieve. So, there is a limit that you can achieve ok; so, that is the main idea that you will get from this slide. So, the take away is; so, what is the trend? You can see the trend of different computation engines in terms of GPUs here you can see, what are their performance density in terms of GOPS/mm² and also the systems that are available with different

feature map sizes ok. So, higher compute density will be achieved by lower feature map sizes.

So, that is the intuition, intuitive idea that is very easy to understand right. So, now you want to achieve, you have seen from the previous section that your compute density is kind of exponentially increasing ok. And, your computation engine is kind of limited with this feature map sizes, you cannot put much more than that resource can allow you right. So, these are the two more much more important conclusions that we will take away after this slide.

(Refer Slide Time: 12:44)



Now, next what we will do is that we will go into some details of evaluation of different GPUs from the mainstream NVIDIA GPUs that are available in market nowadays and their functional number of units that are available. So, increase number of functional units. So, you can see that you are increasing the performance density and number of GFLOPS or gigaflops per second that you are actually much more.

So, currently we have ampere series of NVIDIA GPU which has several thousands of functional units that are available inside these GPUs ok. So, we will see a much more clear array like in a very coarser array like what are the things available and how you can program them in the next lecture. But, we will see some abstraction of these or some features of these modern GPUs which are available and also modern ASICs which are available in their performances how they can run ok.

(Refer Slide Time: 13:58)

The slide compares the NVIDIA V100 and A100 GPUs. It features two side-by-side architectural diagrams. The left diagram represents the V100 architecture, showing a grid of 80 cores. The right diagram represents the A100 architecture, showing a grid of 108 cores. The slide includes the following text:

- NVIDIA-terminology:**
 - 5120 stream processors
 - "SIMT execution"
- Generic terminology:**
 - 80 cores
 - 64 SIMD functional units per core
 - Tensor cores for Machine Learning

[NVIDIA, "NVIDIA Tesla V100 GPU Architecture. White Paper," 2017]

- NVIDIA-speak:**
 - 6912 stream processors
 - "SIMT execution"
- Generic speak:**
 - 108 cores
 - 64 SIMD functional units per core
 - Tensor cores for Machine Learning
 - Support for sparsity
 - New floating point data type (TF32)

<https://developer.nvidia.com/blog/nvidia-ampere-architecture-in-depth/>

URLs at the bottom: <https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf> and <https://images.nvidia.com/sem-dam/en-us/Solutions/data-center/nvidia-ampere-architecture-whitepaper.pdf>

So, in 2017 NVIDIA released V100 that is before in the series of NVIDIA GPU and ampere series of GPU was released in the year of 2020 and you can see that NVIDIA in NVIDIA's terminology this V100 comprises of around 5000 stream processors so, basically 80 cores with 64 SIMD functional units ok.

And, the ampere series having this A100 having around 7000 of stream processors, which can be interpreted as 108 cores with 64 SIMD functional ok so, these are the processing cores or this SM stream processors that you can see here it is having L2 cache also and it has also dynamic memory or DRAM. So, which is this HBM2 that you are seeing here and the their interface that is available on this particular systems.

Now, the basic difference of these two you can see here of course, in the micro architecture level there are differences that you will see. But from the memory point of view you can see that L2 cache is now in the ampere series it is kind of banked into 2 banks ok. So, L2 cache 1 and L2 cache 2 and here in the volta series, the previous series having only one cache memory. And, that is just to increase the throughput and reduce the latency of memory accesses ok.

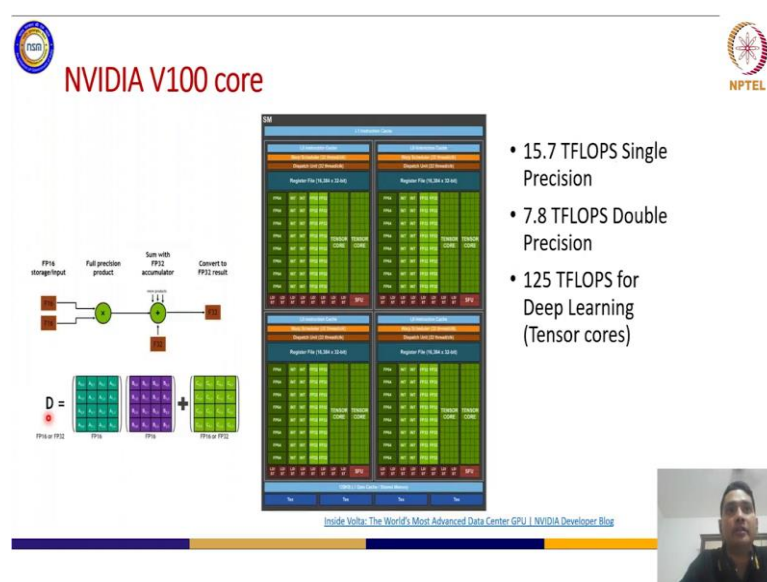
So, now, you can see that how much increase in the number of cores that can be accommodated into one particular system ok. So, from 80 cores to 108 cores in 2 years, you can imagine like how much progress that is happening and how these systems are

getting scaled. Now, from the point of view of precision you can see that the tensor this series of GPUs having tensor cores ok.

So, tensor cores for machine learning is available in your Volta series in the NVIDIA V100 as well as your ampere series, but these having new floating point data type as TF32 ok. So, that gives a bit more flexibility in terms of model training or benchmark training that there will be discussing in the coming classes, but it supports also sparsity in the machine learning or on the AI benchmark.

So, these two are very important things to understand from the algorithmic point of view ok. So, you can accommodate more number of computing cores and now your cores also having tensor cores which are specialized in machine learning.

(Refer Slide Time: 17:20)




Now, if you see the NVIDIA core, one core itself you can see that. So, this is the core of V100, you can see that these are having this are floating point 64. So, this is double precision unit integer units; so, basically these are integer units for MAC operations so, all these are MAC units.


So, with the you can see here floating point 16 who will be multiplied and accumulated with the 32 bit of floating point then it will generate 32 bit of data right. So, this these are floating point 32 bit unit as you can see here and then along with these SIMD processing


So, basically 4x 4 one matrix multiplication will be done in one side, 1 clock cycle to be precise. Now, why matrix multiplication is necessary and why we are telling or why we are calling them tensor core? So, basically the data is this tensor cores are employed or designed specifically to run AI benchmarks which are deep neural network based or convolutional neural network based to be precise.

So, basically these two sets of data you can see 16×16 plus you can accumulate another 16 which is already there. So, in the accumulator and you will get the data of 16 or entire. So, basically 16 MAC operations you are doing in the one side. And, that is how you can increase the throughput in many fold and that is the purpose of these tensor cores, that are available in modern computing engines.



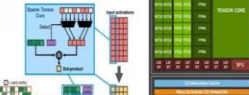
NVIDIA A100 Core





The diagram illustrates the data flow within a tensor core. It starts with 'Input weights' (a grid of colored squares) and 'Input activations' (a column of colored squares). These are processed by a 'Tensor Core' block, which performs a 'Dot Product' operation. The result is then passed through a 'ReLU' activation function to produce 'Output activations' (a column of colored squares). The diagram also shows a 'Tensor Core' block with 'Input weights' and 'Input activations' as inputs, and 'Output activations' as the output.

- 19.5 TFLOPS Single Precision
- 9.7 TFLOPS Double Precision
- 312 TFLOPS for Deep Learning (Tensor cores)



The screenshot shows the NVIDIA A100 TensorRT-LLM performance benchmark results. It displays a 4x4 grid of performance metrics for various configurations. The metrics include 'Performance (TFLOPS)' and 'Performance (GOPS)' for different data types (FP32, FP16, INT8) and precision levels (Single, Double). The results show that the A100 achieves high performance across all configurations, with the highest performance observed in the FP32 Single Precision configuration.

<https://images.nvidia.com/amd-dan/en-zz/Solutions/data-center/nvidia-ampere-architecture-whitepaper.pdf>

Now, we are talking about tensor cores. So, that was core available in the previous generation of NVIDIA GPUs which are V100 and then we now have ampere series of GPUs. And, here you can see in the cores, you can see this register file is there, shared register file and then you have this SIMD of integer 32 then floating point 32, floating point 64.


So, it supports all these different precisions of computations you can see ok. So, all these units are basically this 32 unit, 32 bit fixed point, 32 bit floating point, 64 bit floating point; all these units are mostly used in accelerating your video processing and graphics benchmarks and tensor cores are mostly used for your AI benchmarks ok. So, that is why the modern GPUs that are coming with more number of tensor cores.

And the tensor core in the ampere series is even much more flexible than the earlier series. So, there is a notion of sparsity inside your training or inference in your AI benchmarks. So, what is sparsity? Sparsity means you can have multiple weights or parameters which are very close to 0 can be interpreted as 0. And, if you think from computation point of view; so, if you have let us say 4x4 matrix multiplier.


And, if your let's say half of the data is let's say 0, then you do not need to compute those particular half number of multiplication. So that means, of course, that will be manifold for your 2D matrix multiplication, but you get the idea right. So, basically for where one operand is 0, you just don't do the multiplication.

So, in terms of energy efficiency in terms of throughput you can increase it many fold ok. So, that is why the sparsity and the matrix multiplication is introduced in this tensor cores that are available in ampere series ok.

(Refer Slide Time: 22:16)




Overview of GPU based accelerators



Name	Area [mm ²]	feature size [nm]	Quantization	Bit width	Tensor unit	Throughput [TOPS] ^(a)	Freq. [MHz]	Power [W]	$\frac{GOPS}{mm^2}$ ^(b)
V100 ¹ [37]	815	12	float	64		7.8	1530	300	9.57
			float	32		15.7	1530	300	19.26
			mixed	32-8	X	125	1530	300	153.37
T4 ¹ [38]	545	12	float	32		8.1	1590	70	14.81
			float	16	X	65	1590	70	119.26
			fixed	8	X	130	1590	70	238.53
			fixed	4	X	260	1590	70	477.06
RTX 2080 ² [38]	545	12	float	32		10.6	1710	225	19.45
			float	16	X	84.8	1710	225	155.6
			fixed	8	X	169.6	1710	225	311.19
			fixed	4	X	322.2	1710	225	591.2

Baischer, L., Weiss, M. and TaheriNejad, N., 2021. Learning on Hardware: A Tutorial on Neural Network Accelerators and Co-Processors. *arXiv preprint arXiv:2104.09252*.



So, in overall idea like Volta series that the T4 series, the RTX series. So, what kind of performance or density they are getting with particular power envelope that you can see here in this table. Again this table is taken from this reference and of course, I will give the name of this reference that you can go through after this.

Plus, that the main important thing is that you can increase the density; here you can see only take particular for V100 and for V100 you can see that almost 10 times increase of performance density you can get just by using mixed precision. So, mixed precision means in the whole computation engine you can have 8 to 32 bits of multiply and accumulate.

And, with full precision, full means the double precision you can see the GOPS and one order magnitude you can get of more performance density in 32 bit and even more you can get it if you go for mixed precision. So, precision compute density, the energy the you can see all they are in the same power envelope ok. So, in the same power envelope without either without the inducing more energy you are actually getting much more performance density ok.

But of course, again just to connect to that graph that we saw before is that you cannot increase beyond certain point; because the features size you cannot decrease beyond certain ok. If you could decrease beyond certain point then you will go to atomic level of feature map size and, but that is just theoretical ok.

So, beyond 1nm it is very difficult, because the energy, the temperature that will be generated by the processing engines will be much higher and it will be hard to contain the amount of temperature that will be generated ok. So, that's why you cannot go beyond a certain level of feature size scaling.

(Refer Slide Time: 24:45)

Overview of FPGA based accelerators

Name	Area [mm ²]	feature size [nm]	Quantization	Bit width	Throughput (GOPS) ^(a)	Frequency [MHz]	Power [mW]	(GOPS/mm ²) ^(b)
DuDianNao [7]	4335*	28	fixed	16	1586288*	606	48380*	366*
EIE [18]	40.8	45	fixed	16	102	800	590	2.5
Cambricon-X [61]	6.38	65	fixed	16	544	1000	954	85.26
Eyeriss [8]	16	65	fixed	16	67.2**	200	278	4.2
TPU [25]	331***	28	fixed	8	92000	700	40000	69.48

DCNN Accelerator

Baschiet, L., Weiss, M. and TaheriNegad, N., 2021, Learning on Hardware: A Tutorial on Neural Network Accelerators and Co-Processors. arXiv preprint arXiv:2104.06232

Cloud TPU v2, Cloud TPU v3, Cloud TPU v2 Pod, Cloud TPU v3 Pod


Now, in terms of FPGA based accelerators, FPGAs are fully reconfigurable and this is the abbreviation of Field Programmable Gate Arrays which are field programmable means these are mostly bit level configurable devices. Now, in the FPGAs of course, you can employ different ASIC level accelerators, because it is fully configurable.

But, this slide presents different performances of different ASICs that are available. So, what are different ASICs that are available? We have the TPU which is Tensor Processing Unit from Google. So, v1 is essentially only published for inferencing, but now it is v2, v3 and other versions are employed in Google data centers for a large scale deep neural networks training.


And, all these computation engines that you can see here, they are essentially array of processing elements or they are called systolic array based engines. And, these processing engines as you can see, these are just array of multiply and accumulate engines. And, to feed the huge number of processing engines you can see 12x14 multiplication engines that are being employed in one; this is the first version of the TPU we are talking about.

And, one scratch card memory is deployed to just feed these data hungry processing elements ok and then you have the off-chip DRAM to offload the data to your scratch pad. So, this is the kind of architectural ASIC based accelerators or ASIC based specialized AI engines that are available in markets. And, this is one picture of DaDianNao that you are seeing here and you can see that the GOPS that can be achieved is much more higher with of course, with you can see that how much power it is consuming.

(Refer Slide Time: 27:10)




Overview of GPU based accelerators



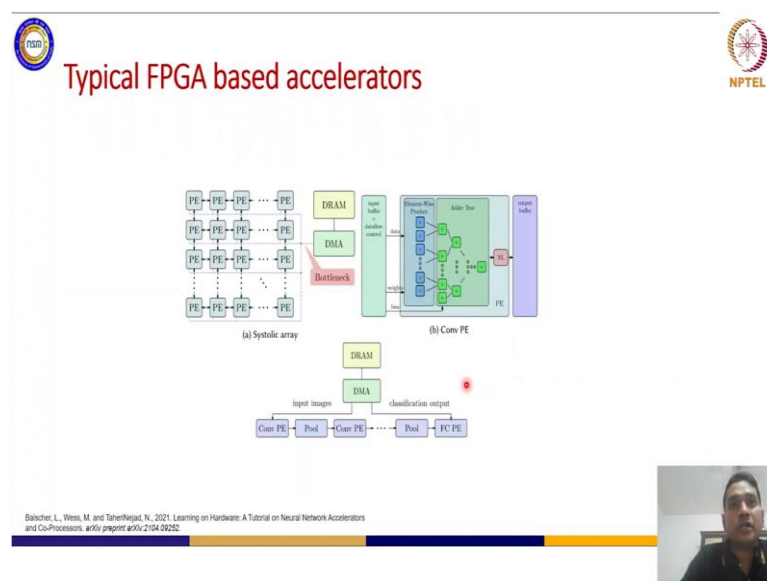
Name	Area [mm ²]	feature size [nm]	Quantization	Bit width	Tensor unit	Throughput [TOPS] ^(a)	Freq. [MHz]	Power [W]	$\frac{(\text{GOPS})^2}{\text{mm}^2}$
V100 [37]	815	12	float	64		7.8	1530	300	9.57
			float	32		15.7	1530	300	19.26
			mixed	32-8	X	125	1530	300	153.37
T4 [38]	545	12	float	32		8.1	1590	70	14.81
			float	16	X	65	1590	70	119.26
			fixed	8	X	130	1590	70	238.53
			fixed	4	X	260	1590	70	477.06
RTX 2080 ² [38]	545	12	float	32		10.6	1710	225	19.45
			float	16	X	84.8	1710	225	155.6
			fixed	8	X	169.6	1710	225	311.19
			fixed	4	X	322.2	1710	225	591.2

Baischer, L., Weiss, M. and TaheriNejad, N., 2021. Learning on Hardware: A Tutorial on Neural Network Accelerators and Co-Processors. *arXiv preprint arXiv:2104.09252*.



But of course, these are in terms of milliwatt whereas, the power consumed by these GPUs are in terms of watt as you can see because of course, they are much more generalized in terms of computing systems. And, these are much more specialized because this fixed precision or mixed precision multiply and accumulate engines only for these processing engines and that is why they consume much more much less power as you can see here.


(Refer Slide Time: 27:45)




And of course, the FPGA based accelerators as I was mentioning that these arrays that you saw in a in the ASIC based accelerators, they can be actually configured or programmed into the FPGA; they emulate one compute engine ok. So, this is basically the figure of that array of multiply and accumulate processing engines and memory hierarchy and these engines are basically this convolution engines.

These are problems and then you have multiply and accumulate then you have final at the tree of multiply and accumulate engines and then you have the final output ok. So, all these AI benchmarks are mostly to train your deep neural network benchmarks to be precise.

(Refer Slide Time: 28:36)

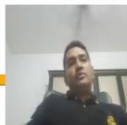


Market share of different technologies




- As of the third quarter of 2017
 - GPU
 - Nvidia represented 72.8%
 - with the rest by AMD
 - FPGA
 - Xilinx 53%
 - Altera 36%
 - Microsemi 7%
 - Lattice Semiconductor (3%)

Hwang, Tim, "Computational power and the social impact of artificial intelligence," Available at SSRN 3147971 (2018).




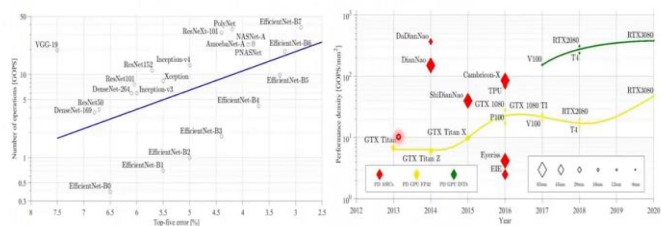
Well, from the market share point of view. So, what kind of market shares in these companies have in terms of different engines like GPU and FPGA based? So, GPU based of course, market is fastly dominated by Nvidia 72.8 percent and rest of the market is. So, these are the stocks of after 2018 financial year. So, now, the ratio might have changed slightly, but overall you can get the idea and then we have in the FPGA domain we have Xilinx, Altera, Microsemi and Lattice semiconductors, which is very few share. Well, these are the systems that are available nowadays.

(Refer Slide Time: 29:30)




The Gap...





Bleicher, L., Weiss, M. and TaheriNegad, N., 2021. Learning on Hardware: A Tutorial on Neural Network Accelerators and Co-Processors. arXiv preprint arXiv:2104.06252



Now, we are talking about a gap. So, this gap I have several times mentioned in the previous lessons, we will just again go through this gap just to summarize whatever we have studied so far right. So, what is this gap about? The gap is about; so, to get particular accuracy level; so, if you want to get linear increase in accuracy ok. So, you need to exponentially increase the computation density ok. So, this is the take away, this was the take away from the left hand side graph.

And, the systems that are available we know that mixed precision and different precision we can employ and we can have much more performance density. But of course, there is some limit and we cannot go beyond certain limit of performance density right. So, the trend to if you see the computation requirement frame that is going exponentially and the resources that are available the trend is kind of getting saturated right, because this the feature size is almost going to your nanometer level right.

So, now how we can bridge this gap? We want to accelerate the AI; that means, the benchmark that we talked about. So, these benchmarks like all these different benchmarks that we have seen and few of these we will see in details in the coming classes, because we need to employ or you need to implement them for a particular target device right. Now, the gap is there ok. So, this computational requirement or the requirement of the computational density is going high exponentially and the performance density is getting saturated.

So, how we can accelerate some beyond certain limit right; so, that is this course all about. We will learn how to accelerate the AI benchmarks that we have seen with the systems that are available nowadays with us and how to employ several techniques from algorithmic to different configurations in training, how we can with different libraries, with different SDKs how we can actually employ efficiently these benchmarks onto these systems that is the goal of this course. And, we will see the implementation or how we can implement on these systems that are available in the coming lectures.

(Refer Slide Time: 32:40)



References and further reading

- [Kim & Mutlu, "Memory Systems," Computing Handbook, 2014.](#)
- [Baischer, L., Wess, M. and TaheriNejad, N., 2021. Learning on Hardware: A Tutorial on Neural Network Accelerators and Co-Processors. arXiv preprint arXiv:2104.09252.](#)



So, to conclude we have these references. So, you can see that we have the reference from Professor Mutlu. So, you can learn about the memory systems, the state of the memory technologies that are available nowadays. And of course, about the basics of these neural networks, the specialized computation engine, their performance and how this scaling happening from that algorithmic innovation point of view as well as from the system level point of view.

And, how we can actually merge them to get better computation and energy efficient computation, performance wise at high accuracy computation, for these benchmarks you refer this paper. Well, that is all about for today.