FPGAs and eFPGAs Accelerate ML Inference at the Edge (WP026)

Achronix
Data Acceleration

May 05, 2021 **White Paper**

Introduction

Machine learning (ML) is poised to transform the world around us. ML has found its applications in a wide range of markets such as autonomous driving, home automation, retail analytics, medical diagnostics, real estate, and image search. With the rapid proliferation of Internet-of-Things (IoT) and billions of connected devices, there is a paradigm shift taking place where big data is not only being processed in the core data center but also at the network edge. Much of the data being generated does not even need to go to the core data center. As such, the need for ML inference at the network edge is growing at an astonishing rate. Pushing inference to the edge imposes severe challenges on the edge hardware due to power, cost and latency constraints. With ML models still in their infancy, there is a need for a flexible architecture that allows the adoption and acceleration of ever changing models. Field Programmable Gate Arrays (FPGAs), sitting at the intersection of performance and flexibility, are a promising solution for deep learning edge inference applications.

Inference Is Moving to Edge

ML is a subset of artificial intelligence that uses trained data models for making predictions or classifications based on the newly observed data. ML involves two phases: training and inference. Training is the process of building models that can perform tasks such as facial recognition and object detection. Training is compute and data intensive. Almost all of the ML training is currently happening at centralized data centers where there is access to a vast amount of data and compute power. Inference is the deployment of the trained models to get insights on the real world data streams. Until recently, inference has been carried out in the data centers. This fared well for non-time-critical data flows, however, it does not meet the latency requirements for mission critical applications such as autonomous driving, AR/VR systems, home automation and telemedicine. With many industries rapidly adopting artificial intelligence to gain insights on the ever increasing data from billions of connected devices, combined with a demand for low latency, there is a growing push to move the inferencing closer to the location where the data is created in order to reduce network dependencies versus a centralized compute implementation which ultimately increases performance and reduces the latency of the workload or application.

Edge inference shifts the information processing close to the point where the information is generated and consumed. Moving inference from the distant centralized data centers to the edge results in faster response times, lower latency, improved security by not having to move data to central data centers over exposed networks, power efficiency and cost savings. Grand view research predicts that the global edge artificial intelligence chip market is expected to grow at a CAGR of 21.3% from 2020 to 2027. One of the main advantages of having the intelligence at the edge is that information can be processed immediately on the premises as it arrives without requiring connectivity to the data center for making insights. Another consideration for edge computing is that not all of the data generated at the end devices flows back to the cloud. Cisco GSI estimates that 850 Zettabytes (ZB) of data will be generated at the network edge by 2021, while only 20.46 ZB of data flows through the global data centers. In many applications, growing concerns on data usage and privacy preclude data transmission off-premises. Data being used for inference might include user sensitive information such as medical records, personally identifiable information (PII) and financial data that need to be handled with utmost security. Users feel safe if their data resides on the local device where they have better control. To make use of all the data generated at the end devices and offer better data protection, edge inferencing emerges as an attractive solution. Let's consider several more use cases which are either dependent on edge computing or greatly benefit from it.

The automotive sector provides a microcosm of changes that can be seen in many sectors. Autonomous vehicles run machine learning models on the data gathered from a multitude of cameras, lidar, radar and ultrasonic sensors to capture any movement or obstacle in the surroundings. These vehicles need to have human-like reflexes to avoid the risk of collision and operate with utmost safety. When the vehicle is traveling at 30, 40 …70mph, milliseconds matter. These vehicles do not have the luxury of milliseconds of delay due to data moving to and from the data center. Edge computing mitigates this delay providing faster response times.

Autonomous driving is only one example of systems needing inference at the edge. Augmented Reality (AR) and Virtual Reality (VR) systems are other applications that have stringent latency requirements. To provide an immersive AR/VR experience, motion-to-photon latency or, stated differently, the delay between when the user moves his/her head and when the system updates the view, should be less than 20ms. If the motion-to-photon delay increases beyond 20ms, there is a possibility that the user will suffer from motion sickness. Various inference applications such as audio or video recognition, object detection and pose estimation running in AR/VR systems are bounded by the above latency requirements and are required to have real time responsiveness to provide a satisfying user experience.

Cloud-centric inference requires large amounts of data to be transferred between the data center and the edge device imposing severe challenges on the network bandwidth. This problem is even more pronounced in rich media applications such as video surveillance and security cameras that continuously stream high quality videos to the data centers for making inferences. With the emergence of smart cities and rapid increase in the adoption of IoT devices, bandwidth requirements will increase in the future. This growing demand for bandwidth is another element that supports the push for edge inferencing.

FPGAs for Edge Inferencing

Although inferencing at the edge offers significant advantages in terms of latency, reliability, and security, it presents unique design challenges. On one hand, an edge device typically has a small form factor and limited power capabilities which places tighter constraints on the design. On the other hand, it demands performance similar to what is required in the core data center. Inference models running at the edge cannot simply be a scaled-down version of cloud models as they need to be accurate enough to make meaningful predictions. It is critical to choose the right hardware technology on which the models are run such that latency, power and cost requirements are met under different workload conditions. Inferencing on the edge can be run on a range of silicon hardware devices such as Central Processing Units (CPUs), Graphic Processing Unit (GPUs), FPGAs, and Application Specific Integrated Circuits (ASICs). FPGAs are the best solution that balance performance, power and flexibility for deep learning edge inference applications. The figure below provides a comparison of various processor types for power, performance and flexibility. ASICs will be the fastest, lowest power and most efficient device if no flexibility in the inferencing model is needed which is nearly never the case. CPUs, on the other hand, will offer the most flexibility while being the most inefficient burning significant power. GPUs and FPGAs offer the needed flexibility, however, much like a CPU, GPUs will generally have higher latency, burn more power and will be less deterministic than an FPGA.

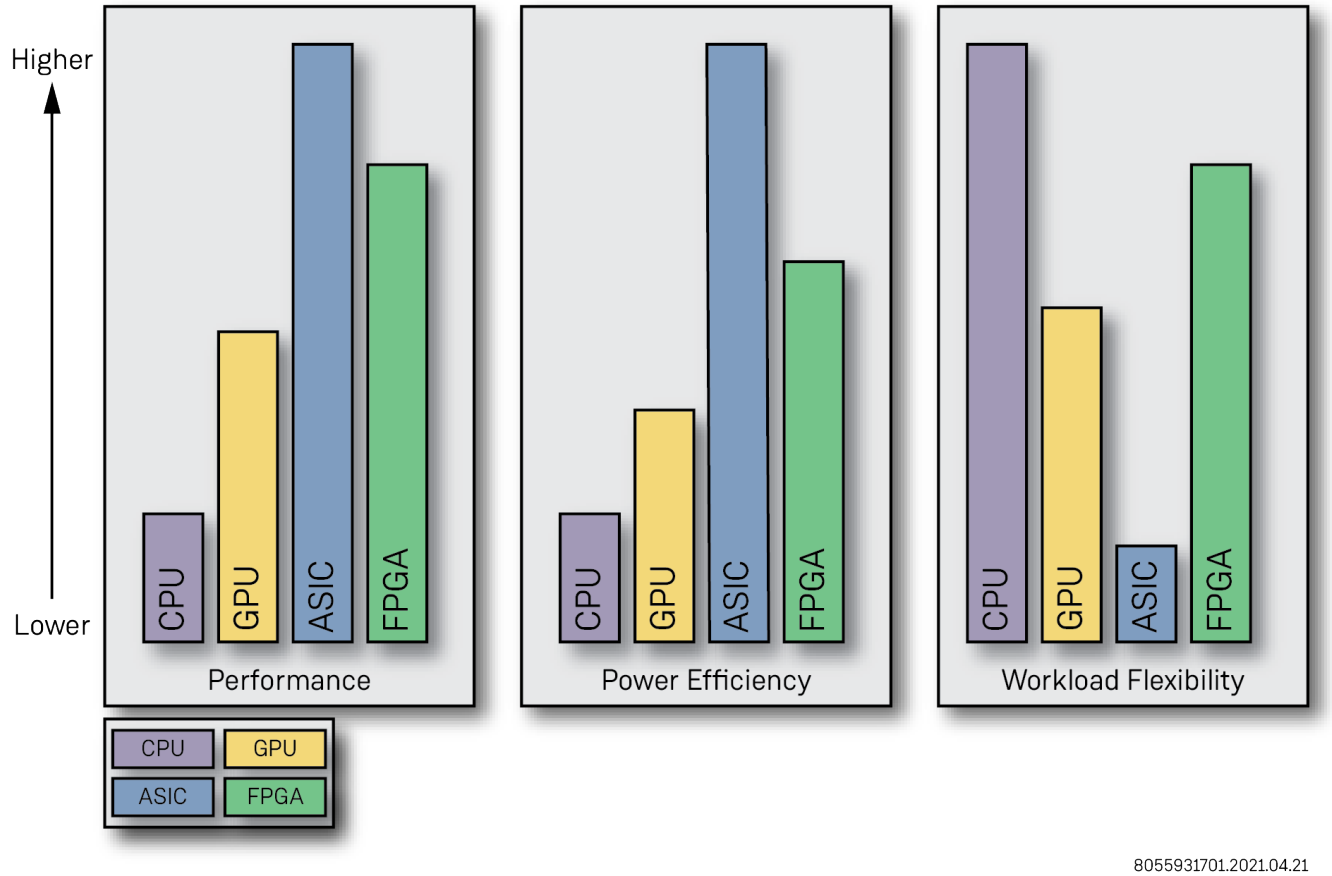

Figure 1: ***CPU vs GPU vs ASIC vs FPGA comparison***

CPUs

A CPU is a general purpose processor that offers high flexibility and is capable of running diverse work loads. However, the flexibility comes at a cost. CPUs, based out of the Von Neumann architecture, execute instructions in a sequential manner. CPUs, because of their architecture, will often perform specific complex tasks much slower than dedicated hardware such as ASICs and FPGAs. Also, the large overhead involved in moving data and instructions around a general-purpose architecture makes CPUs relatively inefficient and power-hungry. While most of the inferencing works at the edge today are carried out using CPUs because of their flexibility and ease of use, their poor performance and high power profiles make them non-ideal for mission critical deep learning edge inference applications.

GPUs

GPUs consist of many small and specialized cores running in parallel offering high throughputs compared to a CPU. They were originally intended to offload highly repetitive compute intensive tasks such as graphic rendering from CPUs. Over time, GPUs found use in deep learning applications especially training because of their massively parallel architectures. While GPUs are widely successful in ML training, they may not be the best hardware choice when it comes to running inference at the edge for various reasons.

GPUs excel in batch data processing, which is often the case in data centers where large chunks of data sitting at the servers are fed into the GPUs for making inferences. At the edge network, data is often streamed from the end devices such as cameras, sensors, and IoT devices. Tighter latency requirements at the edge do not allow for batching of the input data. The data requires processing as it comes in. A GPU does a poor job in handling the stream of data and offers performance lower than that of a FPGA, where the data in and out of the processing engines can be easily pipelined using the custom logic.

GPUs, like CPUs, are power hungry devices and generate a lot of heat. This may not be a major concern for cloud-based training and inference where there is access to a constant power supply and adequate cooling mechanisms are employed. However, it could be a huge problem in the edge environment where mounting cooling systems such as fans is expensive both in terms of the cost and real estate. Additionally, a data center operates in a controlled environment where the temperature and other parameters are strictly regulated. The edge device is deployed in the real world where the hardware could be subject to extreme weather conditions such as heat, dust, and humidity that could negatively affect the performance and functionality of the GPUs. GPUs have a shorter life span of 3 to 5 years. Replacing GPUs on the end device every 3 to 4 years is not cost effective.

ASICs

With CPUs and GPUs on one end of the spectrum offering high flexibility and low performance, ASICs are on the other end of the spectrum, tailored specifically to an end application offering high performance with minimal power consumption. However, the benefits of ASICs come at a cost. An ASIC, once designed, cannot be redesigned without respinning the ASIC. This is a costly and time consuming engineering effort. Also, the initial NRE costs and development time required to develop an ASIC are much higher compared to those of other processor types. The chip volume needs to be high enough to justify the initial R&D costs. With ML models still in their infancy and evolving rapidly, committing to a single architecture that can be relevant until the next ASIC design cycle is not always viable. With a wide range of inference applications requiring a range of solutions, associated costs incurred in developing multiple ASICs will exponentially increase. Programmable silicon provides the ability to update the same hardware on the fly without needing to change the hardware every time there is an update to the model. Unlike ASICs, they can be easily tailored to the evolving applications by merely doing a software update without having to go through an expensive and time consuming process.

FPGAs

An FPGA, with programmability like software-based solutions and the speed, low latency and deterministic characteristics of ASIC solutions, offers the best of the both worlds. It consists of millions of configurable logic cells, memory elements and arithmetic units connected through programmable switches. It can be configured to implement any functionality by applying user-created patterns to each of the logic cells and selectively combining them using the programmable switches. The massive array of logic cells can be effectively leveraged to perform parallel computations increasing the overall throughput. FPGAs, with their reconfigurable hardware and highly parallel architecture, lend themselves well to the AI/ML applications.

FPGAs typically run at lower clock frequencies compared to GPUs or CPUs. However, one should not associate lower clock frequencies with lower performance. Unlike GPUs and CPUs which execute instructions in software, FPGAs execute instructions in the hardware. By mapping the algorithm directly to the hardware, FPGAs offer low latencies by eliminating much of the instruction processing overhead associated with general purpose processors (GPP). FPGAs, with their custom built data logic, eliminate the data access bottleneck associated with GPPs, more than make up for the difference in clock speeds and enable the device to run algorithms in parallel unlike the serial approach of a GPP. The architectural differences of FPGAs offer higher performance per watt compared to GPPs.

The general purpose nature of CPUs/GPUs warrants continuous data movement between compute and storage, leading to significant power consumption. With FPGAs, the data and control flow can be custom built into the logic that is specific to the application, resulting in significant power savings. ML requires a lot of memory to store weights associated with the models. FPGAs have higher on-chip storage compared to GPUs and offer power savings and latency improvements by reducing unnecessary external memory accesses.

Another big advantage of using FPGAs for ML inferencing applications is their ability to support a variety of data types. ML training operates on standard or double precision floating point (FP32) arithmetic as it demands high precision. ML inference, unlike training, can tolerate a significant loss of precision and can operate with compact data types such as INT8, INT16, FP15, FP24 or BF16, binary, ternary, and even custom data types. Working with lower precision arithmetic dramatically lowers the power, storage and bandwidth requirements. GPUs work exceptionally well with certain native data types such FP32 and FP16, but suffer in performance with low precision custom data types. FPGAs are extremely customizable and can be architected to suit any data type, making them highly desirable for edge inferencing.

While programmable silicon looks promising to address the emerging requirements of the edge inference applications, traditional FPGA architectures have some inherent limitations. Traditional FPGAs continue to rely solely on a bit-wise, conventional routing architecture to move data around the device. While the bit-wise routing in FPGAs is very flexible, it has the downside in that each segment adds delay to any given signal path. Signals that need to span long distances in the FPGA will incur the delays of each of the connecting segments, slowing the performance of the function. In addition, the routing adds congestion and uses extra logic resources so you spend more FPGA resources on routing vs. value added IP generation. Also, conventional FPGAs typically have on-chip RAM blocks distributed across the fabric that are placed at some distance from the processing engines. This choice was an effective architecture for typical FPGA designs but it imposes additional and unnecessary routing and latency overhead in an AI context.

Achronix FPGAs

Achronix is the only FPGA vendor offering high end FPGA and eFPGA IP solutions. Both solutions are supported by the same tool suite which simplifies platform development.

Figure 2: ***Achronix Speedster 7t FPGA and Speedcore eFPGA***

Focusing on their 7nm flagship platform, the Achronix Speedster7t FPGA family sets itself apart from the competition as the fastest FPGA in the industry with highest speed external interfaces, an industry-best 2-dimensional network on chip (2D NoC) which routes external and internal data and offers ML-optimized machine learning processors (MLPs). To get data into and out of the chip, the Speedster7t FPGA offers up to four 400GbE ports, both GDDR6 and DDR5 memory interfaces and PCIe Gen5 interfaces. The combination of these I /O solutions is unparalleled in the FPGA space. MLPs are optimized multiply and accumulate units which natively support various number formats including block floating point, floating point and integer making them ideal for machine learning applications which require number format flexibility. The MLPs provide over 80 tera operations per second (TOps) to power next generation AI/ML applications. Connecting all of the I/O, MLPs and other functional blocks on the Speedster7t FPGA is the 2D NoC which supports more than 20Tbps of bi-directional bandwidth. Any combination of the I/Os and the internal functional blocks can be interconnected which minimizes latency and provides maximum bandwidth.

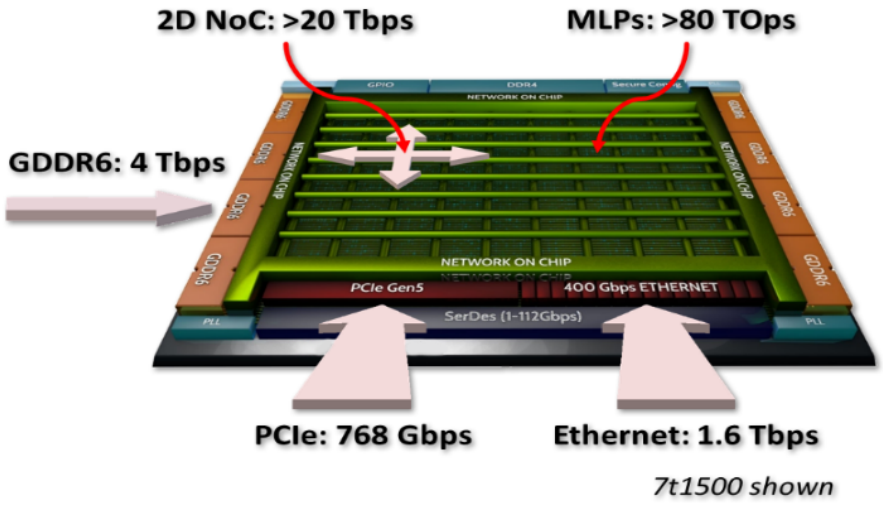

Figure 3: ***Achronix Speedster AC7t1500 High-Performance FPGA Architecture***

In the Speedster7t FPGA architecture, each MLP is associated with a 72kb dual-port block RAM (BRAM72k) and a smaller 2kb dual-port logic RAM (LRAM2k) that can act as a tightly-coupled register file. Employed in AI applications, the BRAM can act as a memory for values that are not expected to change on each cycle, such as neuron weights and activation values. The LRAM is more suited to storing temporary values with only short-term data locality, such as a short pipeline of input samples and for accumulated values for tensor contractions and pooling activities. The co-located architecture of the onboard memory minimizes latency resulting in higher throughput of the inference algorithms.

Aside from having the highest speed FPGA in the industry including the optimized MLPs, Achronix provides an additional benefit over other FPGA vendors with its Speedcore embedded FPGA (eFPGA) solutions. Supported in TSMC's FinFET 16nm, 12nm and 7nm libraries, and soon to be supported in 5nm, Achronix offers its FPGA IP to users who want to develop their own ASIC or SoC solutions. Achronix works closely together with users to determine the logic, memory, DSP, MLP and/or 2D NoC resources required for their solution. The IP is customized for that use case and delivered to the user for integration into their own silicon solution. This combination maximizes the cost effectiveness of a flexible ASIC solution, provides the possibility to address moving standards, workloads and requirements and possibly extends the life of the ASIC with the added flexibility.

Conclusion

With lots to gain by adding intelligence at the edge, ML inference has migrated to the edge network creating demand for accelerated hardware solutions that address the speed, latency and deterministic needs of today's workloads and applications. The edge hardware should offer high performance with minimal power consumption while retaining the ability to adapt to the changing ML landscape. The Achronix speesdster7t FPGA overcomes the limitations of conventional FPGAs through innovations in high speed interfaces, the 2D NoC, the MLP for arithmetic operations and optimized FPGA fabric. The Speedster7t FPGA and Speedcore architecture provides a solid foundation for AI inferencing at the edge.

Achronix Semiconductor Corporation

2903 Bunker Hill Lane
Santa Clara, CA 95054
USA

Website: www.achronix.com
E-mail : info@achronix.com

Notice of Disclaimer