COMP 598 Literature Review: Virtualizing Hardware Accelerators

Elie Climan 260686400

May 14, 2020

Abstract

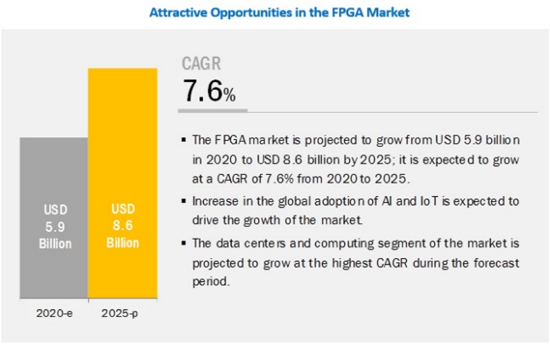
This work seeks to introduce and explore the idea of virtualizing FPGA's and ASIC's in the cloud. An overview of the status quo as well as the feasibility and need for virtual hardware accelerators will be the primary focus along with an analysis of the options that AWS provides. Finally, an analysis of the challenges and opportunities regarding the recent trend of using vFPGA's and vASIC's to increase performance in large scale data centers will be explored.

Introduction

For decades, Moore's law has held since the number of transistors, in fact, doubled every two years [6]. However, as we aim for efficiency increases at orders far smaller than that of, even a decade ago, there is an increasing deviation between Moore's Law and reality. As we push the limitations of hardware to the brink, even Godon E Moore, the co-founder of Intel and creator of Moore's law admits that he expects Moore's law to no longer hold at some point in the 2020s[6]. However, every year we require more transistors to be able to complete the increasingly complex challenges of the modern world[6]. In order to champion those tasks, we developed Field Programmable Gate Array's (FPGA's) and Application Specific Integrated Circuit's (ASIC's) have become increasingly popular with the rise of AI, ML, large scale Data Centers [1] and various other computationally intensive fields that require repetitive tasks that are all very similar in nature. FPGA's and ASIC's allow a developer to accelerate their hardware by adding bare metal circuitry that is optimized to perform very specific tasks. However, because they are bare metal components, they cannot be shared across multiple devices. Furthermore, programming FPGA's and developing ASIC's require knowledge of bitstreams and IP block design which are extremely complex and time intensive to create for even very simple tasks. It then stands to reason that FPGA's and ASCI's are not colloquially developed by individual users but rather companies that can reduce the overall development price per chip by benefiting from the economics of scale. With the rise of cloud computing, this overall benefit via economics of scale has the ability to drastically further bring down those unit prices by virtualizing the components and scaling them across the entire network.

Demand for FPGA's and ASIC's

As per figure 1, the demand for FPGA's and ASCIC's are, unsurprisingly, increasing along with the growth of the IoT and AI.



- The FPGA market is projected to grow from USD 5.9 billion in 2020 to USD 8.6 billion by 2025; it is expected to grow at a CAGR of 7.6% from 2020 to 2025.
- Increase in the global adoption of AI and IoT is expected to drive the growth of the market.
- The data centers and computing segment of the market is projected to grow at the highest CAGR during the forecast period.

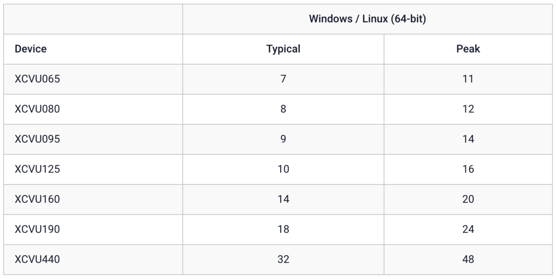
Figure 1: Growth of the FPGA Market

Since, in the case of FPGA's which are encoded by a bitstream, the hardware can be much more efficient than a standard CPU or even GPU because this bitstream allows the user to program the hardware accelerator to maximize efficiency on a specific task which makes it far faster and efficient [15]. Of course, when it comes to AI and IoT which are extremely computationally expensive, this increase in efficiency can be substantial. However, as is the case in computer science, there is a trade-off which is what has driven the demand for ASIC's. A study at the University of Toronto presented the gap between FPGA's and ASIC's. This study showed empirical evidence on the power and speed efficiency improvements that ASIC's offer over FPGA's. However, FPGA's are reprogrammable while, once the ASIC's architecture is imprinted in the silicone, it can never be changed. In short, the trade-off is between flexibility vs efficiency. Either way, the demand for both is increasing.

Economies of Scale

The case for FPGA's:

Historically, FPGA's were relatively expensive, costing upwards of 250 USD for brands such as Altera. Over time, that price has come down below 100 USD making the hardware purchases less of an issue. This trend has brought the upfront costs for FPGA's down but it has not addressed a serious barrier to mass adoption, the enormous amount of power that Xilinx and Lattice environments take to run. These environments can exceed 30GB in installs and have RAM requirements that can exceed those that a typical student or professional might have[9]. An example of this is in Figure 2 which shows the typical and peak memory requirements to run Virtex UltraScale, as part of Xilinx.



	Windows / Linux (64-bit)	
Device	Typical	Peak
XCVU065	7	11
XCVU080	8	12
XCVU095	9	14
XCVU125	10	16
XCVU160	14	20
XCVU190	18	24
XCVU440	32	48

Figure 2: System Requirements for Vivado Virtex UltraScale

These memory requirements are beyond what a laptop might have and thus poses a substantial barrier to entry. This is further exacerbated by the reality that, if the hardware requirements are too high then bitstream development remains only accessible to companies and fortunate individuals who can afford the investment. This perpetuates the cycle that the only way to accelerate hardware is to develop hardware accelerators but the most accessible hardware is unable develop said accelerators. This, of course, leads to a natural conclusion that, by virtulizing FPGA's in the cloud and sharing resources such as memory, it can break down some of those barriers that could lead to increased development of bitstreams thus bringing down the overall cost of bitstream development. This is one of the main principles of economics of scale is that, on the aggregate, we can bring down the cost per unit by scaling the supply. The supply, of course in this case being removed from the individual and placed in the cloud. Furthermore, there is also an argument to be made for abstracted hardware accelerators in the cloud. This is to say that the development of bitstreams and the implementation should be handled by the cloud provider so that the user benefits without needing any precursor knowledge in bitstream development.

The case for ASIC's:

As previously explained, if FGPA's are not fast enough, the next and much more expensive option are ASIC's. The trade-off between flexibility and efficiency can be seen in Fig 3.

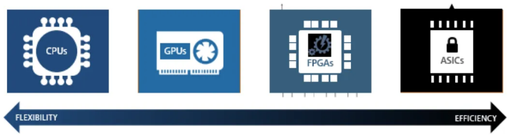


Figure 3: The trade-off between flexibility and efficiency among silicone chips

However, since ASIC's are not re-programmable, in order to virtualize them in the cloud, there's a level of abstraction between the end user and the circuitry of the ASIC. Diagrammatically, this abstraction is represented in Fig 4.

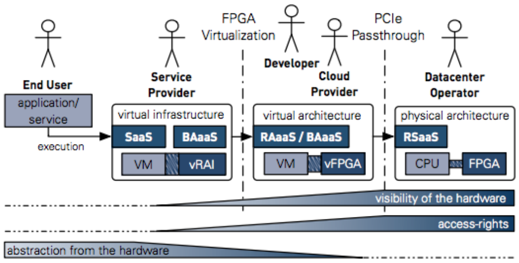


Figure 4: Abstracted cloud based hardware accelaration using FPGA's

Furthermore, due to the high cost and the fact that, once an ASIC circuitry is imprinted in silicone, it can never be changed, this causes the cost per unit, with R&D factored in to be upwards of 100,000 USD [14] , especially relative to the more flexible FPGA's. Amazon's purchase of Annapurna Labs in 2016 in a good example of this which gave Amazon an edge over Google's ASIC counterpart called TPU and Microsoft Azure's FPGA in the machine learning space [18]. This purchase by Amazon also represents a potentially very exciting future for virtualized hardware. By abstracting the process away from the user, Amazon is able to benefit from economics of scale as they continue to develop more complex ASIC's that can be reused and deployed across their entire product line. This allows them to reap the benefits of a substantially decreased marginal cost as the R&D costs become amortized across more products.

Overall, there is a strong case to be made for virtualizing hardware accelerators due to the economics of the cloud. Regardless of which is implemented; Google's TPU, Microsoft Azure's FPGA or Amazon EC2 F1 [8], the computational intensity of tasks are increasing at astonishing rates with the rise of IoT, 5G, AI, ML and various other new paradigms. By virtualizing hardware accelerators in the cloud and thus putting the proverbial ball in the cloud providers court, this could stimulate a competition between cloud providers to then further abstract the development of bitstreams of vFPGA's and vASIC's. This, in turn would provide users with the benefits of virtualized hardware without the prior knowledge necessary to construct their own and provide cloud providers the ability to amortize their development costs across the entire cloud network while giving them an edge over their competitors [18].

An Overview of AWS Hardware Acceleration Offerings

The virtualized hardware accelerator market is still in its infancy. Microsoft Azure's FPGA, Google's TPU, AWS EC2 F1/Nitro Cards are some of the leaders in the consumer facing hardware acceleration market right now [8]. While the IBM DB2 with BLU Acceleration is an example of a business to business facing hardware accelerator that is designed for large scale data centres (DC's) [26]. That is to say that consumers are aware that they are benefiting from the accelerators and they are consumed by the end-user. An example of this is Google's TPU which is a vASIC (virtualized ASIC) which powers Google's products such as Translate, Photos, Search, Assistant and Gmail which has been made available to end users for Machine learning [10]. This paper will focus on AWS's Nitro Project and AWS EC2 F1.

AWS Nitro Project

AWS's Elastic Compute (EC) cloud is a web service that provides secure, resizable compute capacity in the cloud for web-scale cloud computing [2]. In recent years, AWS has been developing more complex FPGA and ASIC offerings to compete in the increasingly competitive cloud compute space. Two of those products are the AWS EC2 F1 and AWS Nitro cards. Nitro cards are the building blocks of AWS EC2 instances. At Amazon re:Invent 2017, Amazon was quoted saying ”our goal was to make the EC2 instance completely indistinguishable from bare metal” [5]. One of the major hurdles of that is the inefficiency of using a Hypervisor. AWS has tried to mitigate those problems by modularizing their cloud offering which can be seen visually in figure 5.

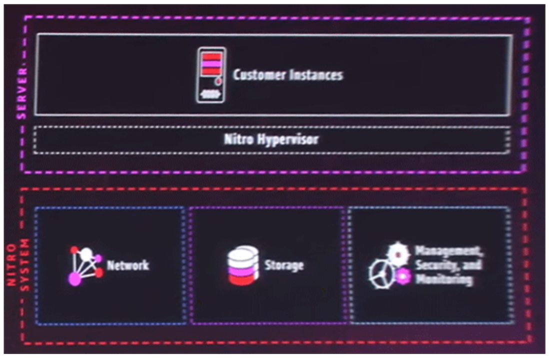


Figure 5: AWS Nitro architecture from AWS re:Invent 2017

By modularizing processes through a hypervisor, AWS allows the user to choose from a series of custom ASIC's called Nitro Cards individually to match their needs. Examples of these Nitro Cards include the Nitro Card for EBC and Nitro Card for Instance storage which all sit above the Nitro Security chip and Hypervisor which manages CPU allocations and memory [16]. In addition to the benefit yielded by the Nitro Cards themselves, the ASIC enabled Nitro Hypervisor also brought considerable benefits that should be noted in Figure 6. One of these benefits is jitter time. Historically, Hypervisors have struggled with real time processing because, as to be expected, there is more software and hardware to traverse before execution so there is substantial latency that could prevent certain software from being virtualized in the cloud.

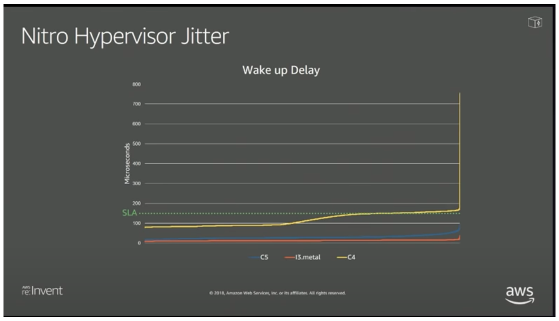


Figure 6: AWS Nitro Jitter graph from AWS re:Invent 2018

The yellow line which denotes AWS C4 represents pre-Nitro wake up delays. This meant that applications that required very low latency - in this case benchmarked at 150 microseconds were not able to run in the AWS cloud. While the C4 remained below the SLA line for the majority of cases, passed the 70th percentile mark, it deviated sharply from I3 metal and Nitro C5. Nitro C5 is actually able to remain quite close in jitter time to bare metal - certainly below the 150 microsecond threshold [5]. This is significant because processes that were once thought impossible because of the latency associated with the cloud were not necessarily so anymore and could be brought into Nitro instances. Ultimately, this allows a whole new branch of hardware accelerated applications to benefit from ASIC designed Nitro Cards. The underlying Nitro Cards (ASIC's) that power these instances include the Nitro Cards for VPC and Nitro EBS for Instance Storage. The Nitro Card for VPC (virtual private cloud) which resembles an NIC (network interface controller) in both appearance and actions provides the link between network adaptors and network connectivity. It also provides the hardware interface between EC2 servers and network connections which supports network packet encapsulations, enforces limits and implements the EC2 security groups [25]. The benefit of having this off the hypervisor and thus, modularizing it, allows customers to fully use their underlying server without impacting network performance. It also enables the Elastic Fabric Adapter which is Nitro Card enabled to provide extremely high powered networking capabilities. Another Nitro card is the Nitro card for EBS (Elastic Block Store) which is an ASIC dedicated to storage acceleration that's designed for both throughput and transaction intensive workloads [3]. The EBS-optimized performance, as of Feb 26 2020 was announced to a provide 36% in performance speed across much of the EC2 suite [4].

AWS EC2 F1

On the FPGA side, AWS offers the EC2 F1 which has both pre-built bitstreams and an interface with Xilinx for developers to create their own. AWS has said that their four major goals regarding FPGA acceleation are; 1. Make FPGA's available and affordable at scale (which ties into the economics of scale above) 2. Speed up the development process 3. Allow developerrs to focus on value-adding acceleration and 4. Provide a marketplace for FPGA applications [23]. EC2 F1 FPGA's handle compute-intensive, deeply pipelined, hardware-accelerated operations while the CPU handles the rest [23]. It does so by transferring the data to and from the FPGA's via PCIe. Figure 6 shows, diagrammatically, how this looks. Essentially, the Hardware Development Kit (HDK) is paired with the Xilinx development environment to create the custom logic which gets AWS then formats into an FPGA shell to create an Amazon FPGA Imagine (AF1), the user can then choose to upload it to the AWS marketplace or attach it to their own F1 instance [23].

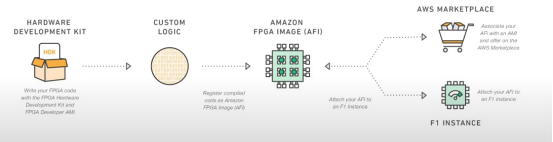
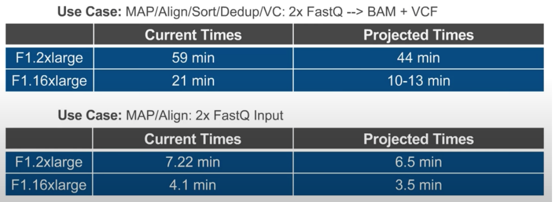

HARDWARE DEVELOPMENT KIT
HDK
Write your FPGA code with the FPGA Hardware Development Kit and FPGA Developer AMI
CUSTOM LOGIC
Register compiled code as Amazon FPGA Image (AFI)
AMAZON FPGA IMAGE (AFI)
Attach your AFI to an F1 Instance
AWS MARKETPLACE
Associate your AFI with an AMI and offer on the AWS Marketplace
Attach your AFI to an F1 Instance
F1 INSTANCE

Figure 7: Diagram of FPGA Programming Flow

The benefits of this process have been well documented and Figure 6 shows the potential benefits on computational biology.



Use Case: MAP/Align/Sort/Dedup/VC: 2x FastQ --> BAM + VCF

	Current Times	Projected Times
F1.2xlarge	59 min	44 min
F1.16xlarge	21 min	10-13 min

Use Case: MAP/Align: 2x FastQ Input

	Current Times	Projected Times
F1.2xlarge	7.22 min	6.5 min
F1.16xlarge	4.1 min	3.5 min

Figure 8: F1 FPGA acceleration data on Edico Genome mapping, alignment and sorting

AWS EC2 F1, however, is not uni-dimensional in that only a single AF1 can be executed on an F1 instance. Similar to a network service chain, the AF1's can be organized to execute dynamically along with the application requirements via the Amazon Machine Image (AMI) [23].

Essentially, AWS has both enabled ASIC hardware accelerators to automatically optimize certain modules within a clients instance as well as offer a user-programmable FPGA option to account for more customer specific use cases.

Virtualized Hardware Accelerators in Practice: Optimizing Data Centres

Benefits and Driving Factors for their Implementation in Data Centre

Amidst Covid-19, there is no better real world example of the need for virtualized hardware accelerators. Without Covid-19, the wordwide public cloud services market was forcasted to grow by 17% in 2020 [19]. As the world works from home, the need to share and store information is growing exponentially. Furthermore, as our lives become increasingly digital, the number of data points that are stored about each process, individual, company and otherwise are being stored in new and innovative ways daily and our general reliance on mass storage to make everything accessible immediately is growing, irregardless of Covid-19. These types of processes have fallen under the umbrella term "big data". However, as suggested above in regards to Moore's Law, even as our number of data points generated increases, the speed at which CPU's can process and analyze them has stagnated. This stagnation has been a major driving factor towards virtualizing hardware accelerators. Additionally, in conjunction with the growth of IaaS/PaaS [19], cloud storage demand has growth dramatically and there is an increasing drive to push many computationally expensive tasks to the cloud. However these clouds are ultimately bounded by the servers and infrastructures which host them. Therefore, data centres designed for the cloud became a natural place to virtualize hardware accelerators in order to reduce the total cost of ownership of the expensive hardware accelerators. These virtualized hardware accelerators provide two major benefits; speed and power consumption. Quantitatively, a study at Umea University showed a 35% improvement in power/performance ratio when using a mix of multimedia and e-commerce applications [24]. Additionally, traditional data centres, which are dominated by CPU-based computing, have their limitations in terms of cooling, speed, electrical power capacity and rack space. By offloading computationally expensive processes to FPGA's, data centres are able to imporve performance and performance per watt [24]. FPGA's can offer substantial benefits in both of those domains. Examples include "Myrtle"; a deep neural net engine which boasts 165x throughput and 1000x performance per watt relative to a traditional cpu [24]. Another example, directly for databases is rENIAC for Apache Cassandra NoSQL databases which offers 4x throughput compared to its un-accelerated counterpart.

Challenges of Using Hardware Accelerators in Large Scale Data Centres:

One of the major challenges of clouds in general is re-source sharing which is further exacerbated when hardware accelerators are taken into account. In essence, the issue lies in how to efficiently share underlying hardware while enforcing data and performance isolation. However, many of the major challenges fall under an umbrella category that is: multi-tenancy.

1. Multi-tenancy: utilization rates

Until recently, multi-tenancy hardware acceleration was not even an option. That is to say that a hardware accelerator such as an FPGA or ASIC hosted in the cloud could only be accessed by a single user at a time [20]. Ultimately, this means that FPGA/ASIC's could not be amortized as quickly and could not be scaled as effectively. In short, this is a major issue for cloud computing as one of the major benefits of cloud computing altogether is to benefit from economics of scale and to provide the end user with also limitless scaling potential. In order to capture the benefits of the cloud, providers require complex multi-tenancy models to allow users to share the underlying hardware while ensuring data privacy [21]. To date, most of the breakthroughs have been for FPGA's since they afford more flexibility at the cost of efficiency via their re-programmable nature. Multi-tenancy is a very important topic in the hardware accelerated data centre space because, since FPGA's are fixed and partitioned inside the data centre, this can result in low utilization rates. Researchers at Microsoft and Tsinghua University published a paper outlining an OS specifically designed to provide multi-tenancy to a single FPGA [17]. This OS design, called *Feniks* works on each FPGA chip and a Feniks instance divides an FPGA's space into an OS region and one or several application regions. Similar to Apache Hardoops Yarn in that for each application, a service manager requests FPGA resources from a central controller via a lease based model [17]. In order to accomplish dividing the FPGA for multi-tenancy, Feniks uses partial reconfiguration (PR) tools from cloud providers such as intel [11]. A big limiting factor of FPGA's is that to re-purpose them and redesign the logic, there's a substantial downtime associated with resetting and then reprogramming the FPGA. PR allows multiple sets of digital logic, in Intels case called personas to be dynamically swapped into the FPGA while the rest of the FPGA continues to operate, resulting in no downtime between persona swaps and thus allowing for a highly efficient multi-tenancy model [7]. Graphically, this can be seen in Figure 10.

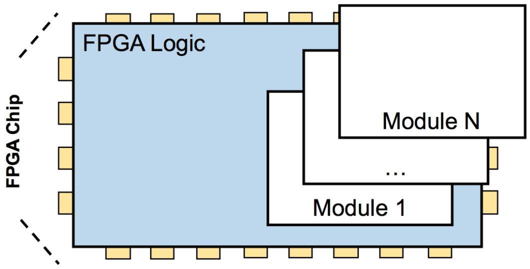


Figure 9: Swapping bitstream logic using PR [22]

2. Multi-tenancy: security

A second major issue regarding hardware accelerator enabled data centres is security. A study at the University of Machester pointed out that it could take less than 10% of the logic resources of an FPGA to draw enough dynamic power to crash a data center FPGA card [13]. In short, security concerns between users can be summarized into a single word: isolation. Isolation in that the provider must ensure one user can't send commands to other users logic or that a user cannot access another users information through the FPGA. Some of these concerns manifest themselves in attackers ability to obtain decrypted FPGA bit-streams via man-in-the-middle attacks and wiretapping. In doing so, adversaries could reverse engineer bit-streams to steal valuable IP from volatile SRAM via cloning [12]. Other common security vulnerabilities are from over-building, hardware trojans and information leaking. To date, there are few elegant solutions to mitigate cloning issues. Identification Friend or Foe (IFF) uses an external safety device with a unique key which is checked before running the FPGA. Another similar approach is Device Identifier Detection which embedds a unique ID into every FPGA. Finally there is watermarking for IP protection. However - none of these solutions address the core multi-tenancy problem where multiple tenants are sharing the same FPGA. These aforementioned solutions only address malicious use of an FPGA, not legitimate use of a shared chip. For this reason, security in multi-tenancy scalable FPGA enabled data centres is still an open issue.

Conclusion

In conclusion, FPGA's and ASIC's have been growing in popularity over the years. However, they have been significantly limited in their widespread adoption because of price, complexity and lack of need for the average person. However, as they are increasingly studied, it has become evermore clear that they provide meaningful results that can extend Moore's law and expand our capabilities with our current transistor densities. Concurrently with the growth of cloud computing, the price barrier has slowly come down as cloud providers are able to amortize the r&d costs as well as upfront capital across their entire cloud, making the venture much more economically viable. As explained above, AWS is an example of this which provides an FPGA and ASIC suite that has been expanded over the years thanks to economics of scale and the increasing demand for computationally expensive tasks in the cloud. Additionally explored is how virtualized hardware accelerators are used in practice to reduce latency, power consumption and increase efficiency in large scale data centres.

References

[1] Putnam et Al. "A RECONFIGURABLE FABRIC FOR ACCELERATING LARGE-SCALE DATACENTER SERVICES". In: (). URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/09/pub_Catapult2014_TopPicks.pdf.

[2] Amazon. "Amazon EC2". In: (). URL: https://aws.amazon.com/ec2/.

[3] Amazon. "Amazon Elastic Block Store Easy to use, high performance block storage at any scale". In: (). URL: https://aws.amazon.com/ebs/?ebs-whats-new.sort-by=item.additionalFields.postDateTime&ebs-whats-new.sort-order=desc.

[4] Amazon. "Announcing 36% faster EBS-optimized performance on additional AWS Nitro System-based Amazon EC2 instances". In: (). URL: https://aws.amazon.com/about-aws/whats-new/2020/02/announcing-36-percent-faster-ebs-optimized-performance-on-additional-aws-nitro-system-based-amazon-ec2-instances/.

[5] Amazon. *AWS re:Invent 2017 - The Amazon EC2 Nitro System Architecture*. Youtube. 2017. URL: https://www.youtube.com/watch?v=02EbskIXC0c.

[6] The Editors of Encyclopaedia Britannica. "Moore's Law". In: (). URL: https://www.britannica.com/technology/Moores-law.

[7] Intel FPGA. "Partial Reconfiguration for Intel FPGA Devices: Introduction Project Assignments". In: (). URL: https://www.youtube.com/watch?v=NAdn39EKWA0.

[8] Karl Freund. "Microsoft: FPGA Wins Versus Google TPUs For AI". In: (). URL: https://www.forbes.com/sites/moorinsights/2017/08/28/microsoft-fpga-wins-versus-google-tpus-for-ai/#79abc4003904.

[9] Xilinix inc. "Memory Recommendations". In: (). URL: https://www.xilinx.com/products/design-tools/vivado/memory.html.

[10] Alphabet inc. "Cloud TPU". In: (). URL: https://cloud.google.com/tpu/?utm_source=google&utm_medium=cpc&utm_campaign=na-CA-all-en-dr-skws-all-all-trial-p-dr-1008076&utm_content=text-ad-none-any-DEV_c-CRE_291265731334-ADGP_Hybrid+%7C+AW+SEM+%7C+SKWS+%7C+CA+%7C+en+%7C+PHR+~+ML/AI+~+TPU+~+Tpu-KWID_43700036255170036-kwd-386224822759&utm_term=KW_tpu-ST_Tpu&gclid=Cj0KCQjw4dr0BRCxARIsAKUNjWRVey8xmFVnEuF-P0Wdhwjz7pNdQqRvEvQ2z2IqpGOZVB651gTpdosaAiQXEALw_wcB.

[11] Intel. “Partial Reconfiguration”. In: (). URL: https://www.intel.com/content/www/us/en/programmable/products/design-software/fpga-design/quartus-prime/features/partial-reconfiguration.html.

[12] Gang Qu Jiliang Zhang. “14Recent Attacks and Defenses on FPGA-based Systems”. In: (). URL: https://dl.acm.org/doi/pdf/10.1145/3340557.

[13] Tuan La et al. “Invited Tutorial: FPGA Hardware Security for Datacenters and Beyond”. In: (Dec. 2019). DOI: 10.1145/3373087.3375390.

[14] Ian Lankshear. “The Economics of ASICs: At What Point Does a Custom SoC Become Viable?” In: (). URL: https://www.electronicdesign.com/technologies/embedded-revolution/article/21808278/the-economics-of-asics-at-what-point-does-a-custom-soc-become-viable.

[15] Jyrki Leskelä. “FPGA VS GPU”. In: (). URL: https://haltian.com/news/fpga-vs-gpu/.

[16] MIKE MACKRORY. “AWS Nitro—What Are AWS Nitro Instances, and Why Use Them?” In: (). URL: https://www.metricly.com/aws-nitro/.

[17] Jiansong Zhang Yongqiang Xiong Ningyi Xu Ran Shu Bojie Li Peng Cheng Guo Chen Thomas Moscibroda. “The Feniks FPGA Operating System for Cloud Computing”. In: (). URL: https://www.microsoft.com/en-us/research/uploads/prod/2018/09/Feniks-APSys17.pdf.

[18] Janakiram MSV. “How An Acquisition Made By Amazon In 2016 Became Company’s Secret Sauce”. In: (). URL: https://www.forbes.com/sites/janakirammsv/2019/03/10/how-an-acquisition-made-by-amazon-in-2016-became-companys-secret-sauce/#14e8070d2f67.

[19] Steve Ranger. “Cloud computing: SaaS, IaaS or PaaS - which is growing fastest?” In: (). URL: https://www.zdnet.com/article/cloud-computing-saas-iaas-or-paas-which-is-growing-fastest/.

[20] Anca Iordache Guillaume Pierre Peter Sanders. “High Performance in the Cloud with FPGA Groups”. In: (). URL: http://www.globule.org/publi/HPCFV_ucc2016.pdf.

[21] Zsolt Istvan Gustavo Alonso Ankit Singla. “Providing Multi-tenant Services with FPGAs: Case Study on a Key-Value Store”. In: (). URL: https://people.inf.ethz.ch/asingla/papers/multes-fpl18.pdf.

[22] Prof. Jakub Szefer. “Lecture: Multi-Tenant Cloud FPGAs”. In: (). URL: https://caslab.csl.yale.edu/courses/EENG428/19-20a/slides/eeng428_lecture_017_multi_tenant_cloud_fpgas.pdf.

[23] AWS Online Tech Talks. *Deep Dive on Amazon EC2 F1 Instance - 2017 AWS Online Tech Talks*. Youtube. 2017. URL: https://www.youtube.com/watch?v=R_Wxc8y7lb0.

[24] Selome Kostentions Tesfatsion Julio Proaño Luis Tomás Blanca Caminero Carmen Carrión Johan Tordsson. “Power and performance optimization in FPGA-accelerated clouds”. In: (). URL: https://doi.org/10.1002/cpe.4526.

[25] M. Vdirona. “AWS Nitro System”. In: (). URL: https://perspectives.mvdirona.com/2019/02/aws-nitro-system/.

[26] Shawn G. Tooley Whei-Jen Chen. “IBM DB2 with BLU Acceleration”. In: (). URL: https://www.redbooks.ibm.com/abstracts/tips1204.html.