Using Field Programmable Gate Arrays to Find Innovation of Diffusion Characteristics of the Computing Space

An STS Research Paper submitted to the Department of Engineering and Society

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Alexander Tomiak

Fall Semester 2021

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Student

Alexander Tomiak Date 11/08/21

Date: 11/28/21

Richard D. Jacques, Ph.D., Department of Engineering and Society

Introduction

Over the last few decades, the world's computational power has been increasingly taken for granted. There are billions of devices performing data operations to support the demand for data processing driven by the rise of Big Data and Internet of Things, and the current number of Internet of Things devices will triple over the next decade (Holst, 2021). One example of this increased demand is in the artificial intelligence field. Figure 1 shows the findings from an OpenAI study which found "…since 2012, the amount of compute used in the largest AI training runs has been increasing exponentially with a 3.4-month doubling time" (Sastry et al., 2019).

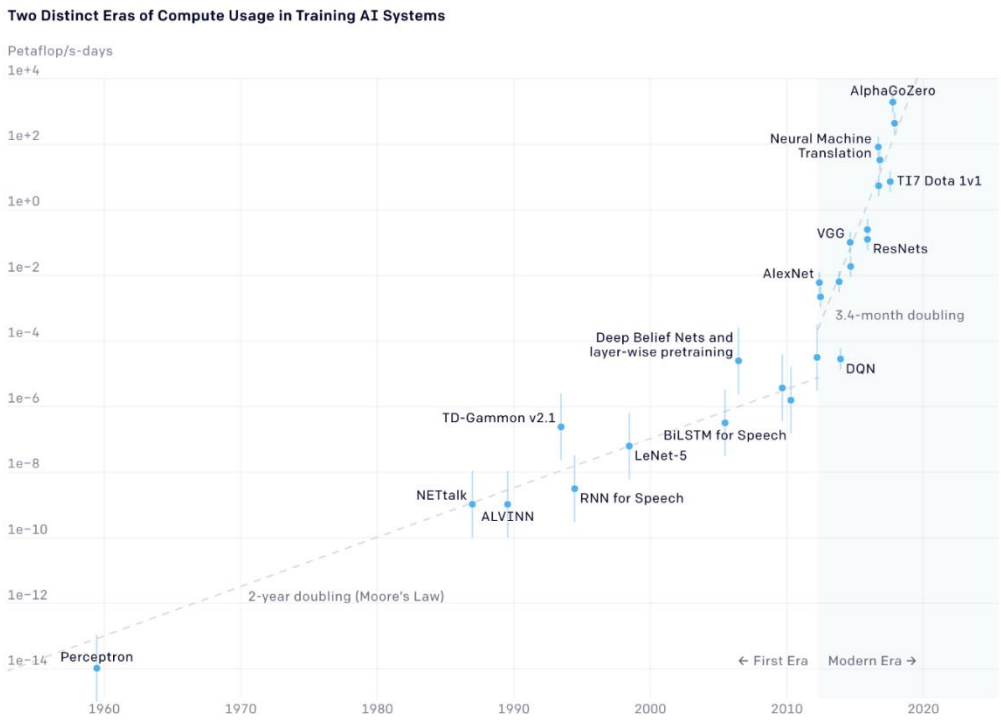

Figure 1: Two Distinct Eras of Compute Usage in Training AI Systems (Sastry et al., 2019). This figure depicts the increasing compute demand in artificial intelligence models.

As the business-facing and consumer-facing performance of devices and systems continues to improve and become a new normal, this demand will eclipse the capabilities of

established computing hardware architectures. This is a reasonably intuitive and straightforward idea—as performance demands change, change demands innovation. In fact, some examples have already emerged in the computing space. One is the usage of a Graphics Processing Unit (GPU) in artificial intelligence applications rather than the traditional Central Processing Unit (CPU) due to the GPU's ability to improve performance by dramatically increasing parallel processing (the performance of multiple operations simultaneously). Another well documented problem in the computing space that has yet to find a widely accepted solution is the Memory Wall. This is the phenomenon in which the processing performance far exceeds the memory access performance to the extent that an operation's time to completion is throttled by reads from memory to fetch new data and writes to memory to store results.

This leads to a sociotechnical problem: given the fact that new computing hardware will need to evolve in order to meet these increasing demands, how can innovators holistically design new computing hardware? What factors should be considered during the design process? This paper will provide answers to those questions by examining Field Programmable Gate Arrays (FPGAs) through a sociotechnical lens. In particular, Diffusion of Innovation Theory will be used to help tell the story of FPGA development and evolution. Through this process, this paper aims to provide insight into how computing technology, and in particular computing hardware, is designed, diffused, and adopted. This insight may be used to help engineers better design these technologies to address the increasing data storage and processing demands of the world.

Describing FPGAs using Diffusion of Innovation Theory

Diffusion of Innovation Theory

In order to properly use Diffusion of Innovation Theory, or DOI, it is useful to explain what it is and how it will be used in this paper. An article by Christopher Sirk nicely summarizes the theory by saying, "Diffusion of innovation theory seeks to explain the adoption of new ideas and technologies" (Sirk, 2020). This theory claims that there are five groups of adopters: innovators, early adopters, early majority, late majority, and laggards. DOI posits that the flow of new ideas begins with innovators, and grows with the early adopters, whose opinions influence the adoption rate in the early majority. The adoption rate being fast enough, "…creates the necessary momentum to get to critical mass. And from there, adoption rate tapers off as late adopters take on the innovation out of necessity" (Sirk, 2020). Sirk indicates that there are four primary elements to innovation diffusion: the characteristics of the innovation itself, communication channels (divided into mass media and social networks), the innovation's compatibility with the social system, and time (in particular, the interval between first knowledge and formation of an opinion). Important characteristics of the innovation itself include complexity, adaptability, compatibility, the relative advantage offered, "trialability" (ability to use the idea on a partial basis or in installments), and how observable the results of the innovation are to others (Sirk, 2020). The five stages of adoption are awareness, persuasion, decision, implementation, and continuation (Sirk, 2020).

In this paper, DOI is used to analyze several products in the computing space. Many of the factors outlined above directly relate to aspects of a computing product's development and evolution. Thus, the sociotechnical analysis will primarily manifest as highlighting particular

points of interest that correspond to the abovementioned diffusion factors while telling the history of the product, and then making assertions about the product's adoption process based on those factors. Both positive and negative factors about the technologies will be highlighted to combat any pro-innovation bias, as this can limit the effectiveness of the model (Abrahamson, 1991).

Introduction to FPGAs

Before diving into the history of FPGAs, it is useful to explain what an FPGA is. FGPAs are a computer processing hardware overshadowed by the more popular CPUs, GPUs and Application Specific Integrated Circuits (ASICs). CPUs, the "brains" of most personal computers and laptops, provide a high amount of flexibility and range in the computations and instructions they can execute. However, this flexibility comes as a performance cost in both optimization for specific tasks and power consumption, each of which cause failures to meet standards for critical systems. In juxtaposition, ASICs are integrated circuits designed to process a specific set of computations on the hardware level. This means they are extremely optimized for a given operation, and also use less power than CPUs, but provide no flexibility or breadth in the kinds of processing they can do without the large financial and temporal costs of remaking the boards. FPGAs are a compromise between these two technologies by allowing for both flexibility and optimization. They are similar to ASICs in that the hardware is designed for specific tasks, but rather than being manufactured in a set way, FPGAs are large collections of logic blocks and memory that the user connects to create a hardware processing system. In short, FPGAs allow the user to program hardware on the fly. This provides the ability to adjust or completely change hardware whenever necessary, providing the flexibility that some tasks may require.

History of FPGAs

Early programmable devices were first designed in the 1980s, and began with Programmable Read Only Memory (PROM). PROM that was able to be programmed by the user was called field programmable, contrasting with mask-programmable (PROM that was mass produced at a factory). One specific type of PROM, called Electronic Erasable (EEPROM), allowed for a user to erase and reprogram its memory multiple times. EEPROM evolved into Programmable Logic Devices (PLDs) by "adding a set of fixed logic OR gates preceded by an array of programmable AND logic gates" (Internet Archive, 2007). These also had mask-programmable and field programmable variants. Finally, in 1985, Xilinx created FPGAs by changing PLDs so that the connections between programmable gates were also programmable. Interest in the technology significantly increased after Adrian Thompson used FPGAs to implement genetic algorithms in order to create a sound recognition device, detailed below.

> Thompson's algorithm allowed the FPGA to decide what the configuration of the chip should be to accomplish the sound recognition task. After repeated generations of the program (approximately 1500) over the period of two to three weeks, the chip not only was able to exceed the requirements set for recognition, but also did it very efficiently. Perhaps the most amazing characteristic of Thompson's voice recognition system was that is used just over thirty of the cells in the FPGA. The best human-designed system used approximately five times that many cells (Internet Archive, 2007).

After their original development, FPGAs have been used and adapted in specific industries. Several of these are written about in an article from *ACM Queue*, summaries of which are presented below.

FPGAs were originally developed as a way to prototype ASICs before sending the final version of the ASIC to be produced. This was because it was much faster and significantly less costly to prototype, debug, and revise circuits on FPGAs than ASICs. However, due to the high per unit price of an FPGA, once a circuit was finalized and deemed acceptable on an FPGA, it was converted to an ASIC for use on the final product (Mencer et al., 2020).

In the telecommunications industry, performance standards constantly change, and building the necessary equipment is difficult. As a result, the company that gets its solution to market first usually secures the largest share of that market. Because ASICs take so long to make, the flexibility of FPGAs makes them desirable to use as a shortcut that can adapt as standards change. They have not completely eliminated ASIC use because of their high per unit cost (Mencer et al., 2020).

In high performance computing and datacenters, a key demand is to provide both extremely optimized performance to handle the large dataflow load and flexibility to handle any needed computational changes. The main obstacle for widespread use is the time it takes to "place and route," or how fast the proprietary FPGA software can map the hardware program onto the physical board. This time can be as long as three days, even with good hardware, making FPGA use difficult. However, the operational cost of FPGAs in these datacenters is significantly lower than CPUs or GPUs, largely because of the significantly reduced power consumption and cooling requirements. Because of this, FPGAs are primarily used by smaller datacenters (Mencer et al., 2020).

In the oil and gas industry in 2007, the time computers took to run simulations to find oil was longer than the physical drilling. Use of FPGAs significantly accelerated the seismic simulations, and led to a lot of success. However, pressure from the ASIC industry and the rise

of the GPU led this imaging to be done on CPUs and GPUs. This ASIC pressure is not isolated, as many ASIC investors see FPGAs as a significant risk (Mencer et al., 2020).

FPGAs are continuing to evolve in response to use cases like these. More and more entities are trying to create FPGAs that are easier to program and use with other technologies. However, many of these methods actually decrease the performance capabilities of the FPGA. FPGAs are in a prime position to be used by engineers working on bleeding edge technology without the financial backing to create ASICs. In particular, there are promising new opportunities in computing around IoT networks, sensors, displays, and artificial intelligence. More progress in availability of software and drivers for I/O devices must be made in order to surmount the barriers to entry in many of the above fields. FPGA developers can also put a greater emphasis on designing new hardware to optimize existing software, rather than creating hardware that is a "blank slate" to run future software on (Mencer et al., 2020).

FPGA Diffusion of Innovation Analysis

Given this information about early development and how FPGAs have evolved to find niche roles in various industries, how are they characterized from a DOI perspective? Almost by definition, FPGAs are extremely adaptable, and provide excellent "trialability" and observability. They have some clear relative performance advantages to other computing hardware, as detailed by an article by David Bacon and others:

> …an FPGA running at a clock frequency that is an order of magnitude lower than CPUs and GPUs is able to outperform them. … When it comes to power efficiency (performance per watt), however, both CPUs and GPUs lag significantly behind FPGAs (Bacon et al., 2013).

FPGAs also take significantly less time to make than ASICs, and have a lower starting unit cost. However, FPGAs also have several relative disadvantages to competitors, some of which result in most software programmers avoiding the technology, "…largely because of the challenges experienced by beginners trying to learn and use FPGAs" (Bacon et al., 2013). These challenges include the poor abstraction, synthesizability, verification, and design and flow tool flow. The focus on low-level flexibility makes it difficult to abstract code in the same way software for CPUs and GPUs has been abstracted. A major factor in all four of these problems is that there is no standardization of features, tools, or libraries in the FPGA field. This creates a clear compatibility problem, and leads to significant per unit pricing issues, explained by one article discussing "their enormous surface area and layers of intellectual property" (Mencer et al, 2020). In summary, the FPGA's innovative characteristics are high complexity, high adaptability, low compatibility, mixed relative advantages, high trialability, and high observability. Given that FPGAs are rarely discussed in broader circles, it is fair to classify their communication channels as being limited to smaller social circles. Their compatibility with the social system is gray; while FPGAs certainly bring benefits to society through performance improvements, they fail to conform to the ease of use that the broader development community seeks.

It is clear that FPGAs have failed to get past the early adopter group. The vast majority of people who are not involved in a specific programming or computing field influenced by or involved with FPGAs are unaware of their existence. Those who are in these fields are either FPGA innovators, FPGA early adopters, or have not adopted FPGAs. The main difference between the last two groups is that for the early adopters, the relative advantages and flexibility offered were enough to make using FPGAs a significantly desirable option in spite of their drawbacks. This idea is directly supported by the use cases outlined above; for example, for

small datacenter owners, the barriers of ease of use and per unit price points are overcome by the flexibility, significantly lowered operational costs, and faster product time to market. In order for FPGAs to become widely adopted, they will need to evolve in a way that reduces per unit pricing or increases accessibility to the average programmer without compromising their impressive performance improvements.

Comparison to Cloud Computing

In order to ensure the observations and hypotheses about FPGA diffusion are useful for future computing innovation design, this paper will perform a similar analysis on other technologies. The goal of this section is to identify similar Diffusion of Innovation characteristics in Cloud Computing, compare them to the FPGA's DOI characteristics, and then see if any differences can reasonably explain discrepancies between the rate of adoption of Cloud Computing and of FPGAs.

Cloud Computing through DOI

Cloud computing is the delivery of computing capabilities over a network. It began in earnest in the 1990s with the rise of virtual computers, where the Internet was used to purchase and deliver software (Foote, 2017). It expanded as major businesses such as Amazon took advantage of its capability to increase efficiency of their computer's capacity. Other organizations took note and followed suit. In 2006 Amazon launched Amazon Web Services, or AWS, which provides storage and computation to other websites and allows individuals to rent virtual computers to use their own programs (Foote, 2017). In the same year, Google created the Google Docs services after acquiring several smaller companies. This service lets multiple users interact with a program that is compatible with Microsoft products (such as Word). Large

companies continue to release platforms, services, and APIs centered around Cloud Computing, many of which are very popular today.

There are three key features of using a Cloud. Someone using the Cloud rents the services, the Cloud vendor is responsible for all management of the hardware and software, and the Cloud comes with storage, multi-project management, and greater availability to many kinds of users, all at the same time (Foote, 2017).

From a Diffusion of Innovation perspective, the Cloud innovation has entirely favorable characteristics. It is extremely adaptable and compatible, and has high degrees of trialability and observability. A user can try one Cloud service at a time, and the Cloud's impact on performance and ease of use are greatly desirable to businesses and individuals. It has minimal complexity for the user, and manageable complexity for developers. Finally, it has a significant relative advantage over its competitor, since "for many business projects, it is simply faster and easier to use the Cloud" (Foote, 2017). Its communication channels certainly encompass social networks, as it actually provides many platforms for social networks themselves to thrive. Once big businesses saw the upsides of running a Cloud, many were quick to broadcast across mass media channels. Cloud Computing has arguably shaped many values of our present-day social system, such as connections formed and maintained with others through vectors other than physical proximity. Cloud Computing is clearly a widely adopted innovation. Because of the visibility and influence of the early adopters (such as Amazon, Google, and Apple), its rate of adoption was extremely fast and forced most businesses to use it or be left behind. A study done on the adoption of Cloud Computing in Kenya agrees with this conclusion, as it found that "coercive and normative pressures have a significant positive relationship with cloud computing adoption," while "the hypothesis that mimetic pressures have a relationship with Cloud Computing adoption

was not supported" (Oredo and Njihia, 2019). On the individual level, the vast majority of people take advantage of its benefits, and in many ways are unable to live in modern society without interacting with a Cloud in some way.

Comparison to Graphics Processing Unit

Given that FPGAs are a kind of computing chip, it is useful to apply the same analysis as done in the previous section to a competing kind of chip. The GPU is a very popular processing unit primarily because of its relatively modern evolution from a graphics-specific processor into a more general-purpose processor. Thus, comparing its DOI characteristics and its rate of diffusion to the FPGA further validates the analysis results presented.

Graphics Processing Unit through DOI

The Graphics Processing Unit, or GPU, began as a device specifically designed to carry "information from the central processor to the display" (Olena, 2018). The early 1990s was an exciting and crucial time period for the graphics market, where significant improvements in 2D performance and integration were developed. This led to the beginning of the modern GPU, the 3D graphics card. Eventually, in 1999, Nvidia produced the first official graphics processing unit, which they defined as "a single-chip processor with integrated transform, lighting, triangle setup/clipping, and rendering engines that is capable of processing a minimum of 10 million polygons per second" (Olena, 2018). Many graphics features now present in cards are a result of the intense competition between ATI (now part of AMD) and Nvidia. So many features were added that the GPU became capable of computation outside of graphics-specific applications. The GPU is now considered more of a general-purpose processor used when data computations can be parallelized (typically, processes that use a lot of linear algebra). This has led to their

continued use not only in graphics rendering, but also in trending fields like machine learning where huge arrays of data can be processed in parallel.

From a DOI perspective, the clash between ATI and Nvidia, two industry leaders, developed factors that led to an extremely high rate of adoption. This competition added so many features to the GPU that it became extremely adaptable and compatible with many other systems and applications. It has good trialability, since one can choose to use a GPU just for graphics or for more generalized purposes. It has excellent observability, best demonstrated by the improvement in visual graphics over the last two decades. Its complexity on the user side has increasingly dropped, as for the casual user they are automatically integrated into most personal computers, and for the developer, APIs continue to be built out (although the most notable, OpenCL and CUDA, are known for being difficult to debug). It has an excellent relative advantage over other chips, as its combination of parallelism and easy integration with other systems makes it a very popular choice for the data intensive roles it is considered for. Its communication channels include both social networks and mass media because of the abovementioned intense competition between two computing heavyweights and its recent rise in popularity with machine learning.

This assessment provides an extremely useful comparison to the FPGA analysis. Both devices show similar levels of adaptability, trialability, observability, and computation speed performance (FPGAs have significantly better power consumption performance). However, the GPU has significantly more favorable levels of compatibility and complexity; in fact, the GPU has become mainstream in areas that the FPGA could but does not largely because of these factors. For example, the GPU is used as an accelerator in most personal computers instead of an FPGA because it integrates well at a significantly cheaper price point. Because of its ability to

cleanly and cheaply enhance performance of a wide variety of systems, the GPU has been widely adopted often at the expense of FPGAs.

Conclusion

Through the Diffusion of Innovation analysis of FPGAs, and subsequent comparisons to similar analyses of other computing technologies, several factors have been made clear as being important for widespread adoption in the computing space. All three studied innovations are highly adaptable, which plays a key role in their diffusion. This is because the computing space evolves so rapidly that any one product must be able to adapt to a new field or risk becoming obsolete. The innovation should provide a significant relative advantage without compromising a high level of compatibility with other technologies and without providing a high level of complexity for both developers and users. This is most likely the biggest takeaway from the evolution of FPGAs: despite the impressive flexibility and performance advantages over other chips, their failure to be accessible from both financial and knowledge standpoints significantly reduces their rate of adoption in the computing industry. If a future chip can be adaptable to new trends, offer power consumption and processing performance improvements, limit complexity to developers and users, and be compatible with other technologies, it will have a high rate of adoption. According to a *Forbes* article, "People have started realizing that in the age of the digital economy, computing power is the most important and innovative form of productivity," but the same article says that the world has insufficient computing power (Tsou, 2020). Perhaps the next chip is a redesigned board that combines elements from CPUs and FPGAs, or maybe it is something entirely new. Whatever its form, there is no doubt it will exhibit the innovative characteristics found in this paper to be crucial to diffusion and adoption.

References

Abrahamson, E. (1991). Managerial Fads and Fashions: The Diffusion and Rejection of Innovations. *Academy of Management*, *16*(3), 28. https://doi.org/10.2307/258919

Amodei, D., et al., (2018, May 16). *AI and Compute*. OpenAI. Retrieved from https://openai.com/blog/ai-and-compute/

Bacon, D. F., Rabbah, R., & Shukla, S. (2013). FPGA Programming for the Masses. *Acmqueue*, *11*(2), 13. https://queue.acm.org/detail.cfm?id=2443836

Field Programmable Gate Array Chips: History. (2007, April 12). Internet Archive. https://web.archive.org/web/20070412183416/http://filebox.vt.edu/users/tmagin/history.htm

Foote, K. D. (2017, June 22). A Brief History of Cloud Computing. *DATAVERSITY*. https://www.dataversity.net/brief-history-cloud-computing/

Holst, A. (2021, October 29). *IoT connected devices worldwide 2019-2030*. Statista. https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

Mencer, O., et al., (2020). Hitting a nerve with field-programmable gate arrays. *Acmqueue*, *13*(3), 12. https://queue.acm.org/detail.cfm?id=3411759

Olena. (2018, February 22). A Brief History of GPU. *Altumea*. https://medium.com/altumea/a-brief-history-of-gpu-47d98d6a0f8a

Oredo, J., & Njihia, J. (2019). Adoption of Cloud Computing by Firms in Kenya: The Role of Institutional Pressures. *The African Journal of Information Systems, 11*(3), 25. Retrieved from https://www.researchgate.net/publication/334083570_Adoption_of_Cloud_Computing_by_Firms_in_Kenya_The_Role_of_Institutional_Pressures

Sirk, C. (2020, August 21). *Diffusion of Innovation: How Adoption of New Tech Spreads*. CRM.Org. https://crm.org/articles/diffusion-of-innovations

Tsou, S. (2020, January 24). *Council Post: The Need For Computing Power In 2020 And Beyond*. Forbes. https://www.forbes.com/sites/forbesbusinesscouncil/2020/01/24/the-need-for-computing-power-in-2020-and-beyond/