The History, Status, and Future of FPGAs

HITTING A NERVE WITH FIELD-PROGRAMMABLE GATE ARRAYS

OSKAR MENCER

DENNIS ALLISON

ELAD BLATT

MARK CUMMINGS

MICHAEL J. FLYNN

JERRY HARRIS

CARL HEWITT

QUINN JACOBSON

MAYSAM LAVASANI

MOHSEN MOAZAMI

HAL MURRAY

MASOUD NIKRAVESH

ANDREAS NOWATZYK

MARK SHAND, AND

SHAHRAM SHIRAZI

This article is a summary of a three-hour discussion at Stanford University in September 2019 among the authors. It has been written with combined experiences at and with organizations such as Zilog, Altera, Xilinx, Achronix, Intel, IBM, Stanford, MIT, Berkeley, University of Wisconsin, the Technion, Fairchild, Bell Labs, Bigstream, Google, DIGITAL (DEC), SUN, Nokia, SRI, Hitachi, Silicom, Maxeler Technologies, VMware, Xerox PARC, Cisco, and many others. These organizations are not responsible for the content, but may have inspired the authors in some ways, to arrive at the colorful ride through FPGA space described above.

Field-programmable gate arrays (FPGAs) have been hitting a nerve in the ASIC community since their inception. In the mid-1980s, Ross Freeman and his colleagues bought the technology from Zilog and started Xilinx, targeting the ASIC emulation and education markets. [Zilog came out of Exxon, since in the 1970s people were already afraid that oil would run out in

30 years, which is still true today.) In parallel, Altera was founded with similar technology at its core.

An FPGA is a chip that is programmed by a circuit. It is said to “emulate” that circuit. This emulation runs slower than the actual circuit would run if it were implemented in an ASIC—it has a slower clock frequency and uses more power, *but* it can be reprogrammed every few hundred milliseconds.

People who make ASICs started using FPGAs to emulate their ASICs before committing them to a mask and sending them out to the factory to be manufactured. Intel, AMD, and many other companies use FPGAs to emulate their chips before manufacturing them.

HITTING A NERVE IN TELECOM

The telecom industry has been a heavy user of FPGAs. Telecom standards keep changing and building telecom equipment is hard, so the company that ships telecom solutions first tends to capture the biggest chunk of the market. Since ASICs take a long time to make, FPGAs offer an opportunity for a shortcut. FPGAs started to be adopted for first versions of telecom equipment, which initiated the FPGA price conflict. While the price of the FPGA does not matter to the ASIC emulation market, the price of a chip for telecom is important. Many years ago, AT&T and Lucent made their own FPGAs, called ORCAs (optimized reconfigurable cell arrays), but they were not competitive with Xilinx or Altera in terms of speed or size of the silicon.

Today, Huawei is the largest customer for FPGAs. It is possible that the recent tension between the United States

Early on, FPGAs were used for SDRs (software-defined radios), building radios for communication on many different standards at the same time, in essence having a single phone speaking many languages.

and China began with FPGAs from the United States giving Huawei an edge in delivering 5G telecom equipment two years before any of the other vendors around the world got ready to play.

FPGA PRICE HITS A NERVE

Early on, FPGAs were used for SDRs (software-defined radios), building radios for communication on many different standards at the same time, in essence having a single phone speaking many languages. This use of FPGAs hit a huge nerve. There was a split in how SDR technology was implemented. Commercial vendors developed cost-effective solutions, and today every base station on the planet has SDR technology in it. In the defense community, on the other hand, SDRs were built by large defense contractors with profitable legacy product lines to protect. The result was that the price of FPGA-based radio products was so high that part of the U.S. defense market developed a persistent allergic reaction to their use.

Next, FPGAs tried to grow in the DSP (digital signal processor) and embedded markets. FPGAs with small hard microprocessors in the corner started to appear. The pressure to sell these new FPGAs was so high that if customers rejected the new family of chips, they were put on a blacklist, and sometimes even refused service for a few months. Pressure to grow the FPGA market was and still is immense, as is the magnitude of the failures of FPGA companies to conquer new markets, given the impossibility of reducing the price of FPGA products because of their enormous surface area and layers of intellectual property.

HITTING A NERVE IN HPC AND DATACENTERS

For the past few years, FPGAs have tried to grow in the HPC (high-performance computing) and datacenter markets. In 2017 Microsoft announced its use of Altera FPGAs in the datacenter, and Intel bought Altera. In 2018 Xilinx announced its “Datacenter First” strategy, with the Xilinx CEO declaring in front of an audience of analysts that Xilinx is not an FPGA company *anymore*. This may have been a slight dramatization, but historically there is relevance.

In HPC and datacenter usage of FPGAs, the main obstacle today is *place and route*—the time it takes to run the proprietary FPGA vendor software that maps the circuit onto the FPGA elements. On large FPGAs and on a fast CPU server, place and route takes up to three days, and many times even after three days the software fails to find a mapping.

HITTING A NERVE IN OIL AND GAS

In oil and gas implementations, however, around 2007 a niche opened up. The time it took classical computers to simulate drilling holes in the earth to find oil was longer than the actual building of a drilling site and the drilling itself. The use of FPGA accelerators dramatically changed this upside-down timing. The first FPGAs in the datacenter of an oil company, computing seismic images, were built by Maxeler Technologies and delivered to Chevron.³

The use of FPGAs in oil and gas expanded for a few years, until pressure from the ASIC industry led to a return to standard CPU technology. Today prediction and simulations in oil and gas are still important, and seismic

imaging is mostly done on CPUs and GPUs, but the FPGA opportunity still exists. We are reminded that “today’s new stuff is tomorrow’s legacy,” and, of course, today’s new stuff is AI and a focus on data.

Despite all of this, FPGAs remain a quick way to market, a simple way to obtain competitive advantage, and an indispensable technology for many mission-critical situations—even though they are expensive on a per-chip basis compared with ASICs. In HPC and the datacenter, however, FPGAs have significantly lower operational costs than running software on CPUs or GPUs. Fewer FPGAs are needed, requiring much less cooling than both CPUs and GPUs. FPGAs make for smaller datacenters, hitting a nerve with operators who fear their datacenters might shrink.

ASIC VS. FPGA

Another way to use FPGAs is to complement ASICs. ASICs are built to hold fixed functionality while adding FPGAs to provide some flexibility for last-minute changes or adaptivity of the products to different markets.

Modern FPGAs are integrating more and more hard functionality and becoming more and more like ASICs—while ASICs are sometimes adding a bit of FPGA fabric into their design for debugging, testing, in-field fixes, and flexibility in adding little bits of functionality as needed.

Nevertheless, ASIC teams always fight the FPGA concept. ASIC designers ask, “Which functionality do you want?” and are impatient if the answer is, “I don’t know yet.”

One such new battleground is the autonomous car industry. Since algorithms are constantly changing, and laws could change when cars are in the field, requiring

driver updates, the solution needs to be flexible. Since FPGAs consume less power and are physically smaller (and have much smaller heat-sinks because of their lower clock frequency of FPGAs) than CPUs and GPUs, they are the obvious choice. Nevertheless, GPUs are easier to program and do not require a three-day place and route.

Moreover, it is critical to be able to run the same code in the car and in the cloud (primarily for simulation and testing), so FPGAs would have to be available in the cloud before they could be used in the car. For these reasons, many developers prefer GPUs.

EVOLUTION OF FPGAS

FPGAs are evolving. Modern interfaces are trying to make FPGAs easier to program, more modular, and more cooperative with other technologies. FPGAs support AXI (Advanced Extensible Interface) buses, which make them easier to program but also introduce enormous inefficiencies and make FPGAs less performant and ultimately much less competitive. Academic work, such as Eric Chung's paper on dynamic networks for FPGAs,¹ helps with the routing problem, but such advanced ideas have not yet been picked up by industry.

How are FPGAs connected? For HPC workloads with large flows of data, you can use PCI Express and deploy communication-hiding techniques. But how about small workloads, such as found in NFV (network function virtualization), serving a large number of users at the same time. For NFV and acceleration of virtual machines in general, the FPGA must connect directly to the CPU, possibly using cache coherency as a communication

The key barrier to entry for new CPUs and FPGAs in the data center is not just speed and cost, but also the availability of software and drivers for all possible I/O devices.

mechanism, as investigated these days by VMware. Of course, a key feature is the ability to crash the FPGA without crashing the CPU, and vice versa. Hyperscalar technology companies are rediscovering requirements from IBM mainframe days, driving more and more complexity into standardized platforms.

There are also opportunities for the masses. In offering FPGA platforms, organizations without the budgets for ASIC development and without knowledge of the latest silicon fabrication challenges and solutions can develop circuits and build competitive advantage into their products, such as the newly emerging opportunities for computing at the edge of the IoT (Internet of things) network, close to sensors, displays, or just in-line at the wire, as data flows through.

Meanwhile, FPGA companies are pushing vertically up the stack and into the CPU socket, where Intel is dominating the market, including, for example, special instructions for NFV. The key barriers to entry for new CPUs and FPGAs in the datacenter are not just speed and cost, but also the availability of software and drivers for all possible I/O devices.

Key to making FPGAs work in the datacenter is to make them easier to use—for example, with automatic tools that drive the use of FPGAs without place and route difficulties. Microsoft pioneered the use of FPGAs in a hyperscalar datacenter for accelerating Bing, NFV, and AI algorithms. Microsoft also built abstractions, domain-specific languages, and flexible hardware infrastructures. Commercially, the main problem with FPGAs is the go-to-market strategy.

Building new chips and *then* starting to think about the software is too late. How do you extract value from existing software by adapting the hardware to serve the software? This also brings an opportunity to rethink FPGA architecture. A word of warning, however: The silicon industry devours cash. Building ASICs is a poker game with minimum bets rising over the years. It's a winner-take-all game, and any threats such as FPGAs get eliminated early in the race.

FPGAs are creating additional and undesirable risks for silicon projects.

NICHE TECHNOLOGY

While a software designer will always say, "If it *can* be done in software, it *will* be done in software," the ASIC designer will say, "If it *can* be done in an ASIC, it *will* be done in an ASIC." Most interestingly, "If it can be done in software, you don't have to deal with the guy who thinks like an FPGA." FPGAs have a tiny community of sometimes eccentric, programmers, compared with the armies needed to make ASICs and with the world population of software programmers. The FPGA companies are small. The FPGA community is small.

Intel is driving FPGAs for flexibility. It is the most successful company following the principle of building the hardware to run existing software.

FPGAs can be faster than CPUs and GPUs, but the hard lesson from industry and the investment community is that most of the time during a computer's existence, speed does not matter, and realtime does not matter. Therefore, buying a computer for speed alone is rare. It happens, but

it's more of a random event than a market on which to build a business. In addition, FPGAs have no standard, open source, enjoyable programming model—and, therefore, no standard marketplace for FPGA programs that work on all FPGA chips or can be easily cross-compiled. Maxeler Technologies has a high-level solution to provide such an interface, but wide industry adoption requires trust. To go from early adopters to benefiting everyone, trust requires alignment and support from established vendors in the datacenter space.

Applications people in the real world say, “I don't care what it is, just give me a way to do what I want to do.” What are the possible application areas for FPGAs that have not been widely explored yet? For realtime computing, there is manufacturing. For computer vision on drones, it's the weight and power advantage of FPGAs. On a satellite it is very expensive to do hardware upgrades, so FPGAs provide long-term flexibility that can be critical. FPGAs need to find a product that resonates, and they need to be easy to program. It's not just the hardware or software, it's the ecosystem. It's the complete solution.

One way to expand beyond current market confines is realtime compilation and automatic FPGA program generation. This is easier said than done, but the opportunity is growing with AI tearing up the application space. These days, everything is done with AI; even traditional algorithms such as seismic imaging for oil and gas are incorporating AI. A science and engineering solution is needed to deal with AI blocks. FPGAs might be a good starting point, maybe initially to connect the AI blocks and then to incorporate them into the FPGA fabric

such as the next-generation chips from Xilinx—with AI fabric, CPUs, 100G interfaces, and FPGA cells all in the same 7-nm chip.

From another perspective, with AI chips producing and consuming vast amounts of data, FPGAs will be needed to feed the beast and move outputs away swiftly. With all the new ASICs for AI processing coming out, FPGAs could provide differentiation to AI chip companies.

PREDICTIONS

Could the following developments have been predicted 10 or 25 years ago?² While the world changes, the predictions seem to stay the same.

1. There will be successful CPU+FPGA server chips, or FPGAs with direct access to the CPU's cache hierarchy. Some say yes, and some say no.
2. SoC (system on a chip) FPGA chips will grow and expand, driving the medical, next-generation telecom, and automotive industries, among others.
3. Developers will use FPGAs to do amazing things and make the world a better place but will have to hide the fact that there is an FPGA inside.
4. The FPGA name will remain, and chips called FPGAs will be built, but everything inside will be completely different.
5. As we forego (dataflow) optimization in order to make FPGAs easier to program, the performance of FPGAs will be reduced so they are no longer competitive with CPUs, which will always be easier to program.
6. There will be FPGAs with dynamic routing, evolving interconnect, and runtime-flexible data movement.

7. Place and route software, as well as the complete software stack on top of FPGAs, will be open source. There are already initial efforts with Yosys and Lattice FPGAs.
8. All semiconductor architectures will be combined into single chips with combinations of TPUs, GPUs, CPUs, ASICs, and FPGAs. Some may be combinations of the whole of each. Others will be combinations of parts of each.
9. More chips will be focused on limited application spaces, and fewer on general-purpose chips. In a way, everything is becoming an SoC.

Related articles

FPGA Programming for the Masses

The programmability of FPGAs must improve if they are to be part of mainstream computing.

David F. Bacon, Rodric Rabbah, Sunil Shukla

<https://queue.acm.org/detail.cfm?id=2443836>

FPGAs in Data Centers

Expert-curated Guides to the Best of CS Research

Gustavo Alonso

<https://queue.acm.org/detail.cfm?id=3231573>

Reconfigurable Future

The ability to produce cheaper, more compact chips is a double-edged sword.

Mark Horowitz

<https://queue.acm.org/detail.cfm?id=1388771>

FINAL COMMENT

How many conflicts are resolved with this article, and how many new ones are created? In this sense, a conflict is a challenge to an existing way of doing things. Such an existing way of doing things may have implications for the way people think, and, therefore, for the way they act. But maybe more importantly, there will be implications for how we developers earn a living.

References

1. Chung, E. 2011. CoRAM: An in-fabric memory architecture for FPGA-based computing. Ph.D thesis, Carnegie Mellon University.
2. Field-programmable Custom Computing Machines. 2012. FCCM predictions; <https://www.fccm.org/past/2012/Previous.html>.
3. Nemeth, T., Stefani, J., Liu, W., Dimond, R., Pell, O., Ergas, R. 2008. An implementation of the acoustic wave equation. In *Proceedings of the 78th Society of Exploration Geophysicists Meeting*, Las Vegas.

Copyright © 2020 held by owner/author. Publication rights licensed to ACM.