

# Four key trends in the networked use of FPGAs

The use of Field-Programmable Gate Arrays (FPGAs) is growing. Their unique mix of configurable programmable logic, memory and network connectivity makes them a serious alternative to traditional microprocessors where large amounts of parallel processing are needed.

They are widely used for video encoding, digital signal processing, neural networks, medical devices, scientific instruments, avionics and much more. Demand is skyrocketing, not least thanks to Amazon offering its EC2 F1 virtual machines with FPGA coprocessors. Microsoft has used FPGAs extensively within Azure. On the networking side, an increasing number of off-the-shelf devices contain FPGAs to perform tasks such as switching, routing, buffering, filtering and more.

FPGAs have the key advantage that they are inherently reconfigurable. Bugs in their application logic can be fixed, features added or platforms reconfigured to implement different applications. From the perspective of FPGA application developers, this results in fewer design compromises as they now have the ability to adapt functionality to changing requirements. This guide introduces four different networked applications that are seeing increasing growth in the use of FPGAs.

### FPGAs enabling software-defined networking

SDN is a concept that essentially decouples the control plane of computer networks from the actual devices implementing the network. SDN has been around in one form or another since the mid 1990's however the inexorable growth of network traffic volume, the trend towards the cloud computing model of shared, easily provisioned, mobile, abstracted services and the need for improved network security has now made it a multi-billion dollar market that is growing rapidly. There are various standards for the SDN control plane with the Open Networking Foundation (ONF)'s OpenFlow probably the most significant. Other proprietary solutions include VMware's NSX for example. The data plane that actually processes raw network packets is traditionally implemented in an application specific integrated circuit (ASIC) microchip. ASICs in this context are microchips designed to switch and/or route Ethernet packets between ingress and egress ports. In the case of a traditional switch or router, these are coupled with a proprietary control plane however in an SDN device, the control plane is abstracted from the switching ASIC which may support one or more of the SDN control plane standards.

A key limitation of the SDN control plane standards is that they can only implement functionality available within the hard-coded data plane implementation. Consequently, if someone wants to create, extend or evolve a network protocol and the data plane is not designed to support it, the SDN network as a whole cannot implement it. Even minor changes could require re-engineering the ASICs implementing the data plane - a lengthy and extremely costly process. As network protocols evolve ever more rapidly to keep pace with new technologies such as the Internet of things (IoT) or new security protocols, a data plane that is fully-reconfigurable is required. Two key initiatives are underway to achieve this:

1. On the ASIC side, more flexible general-purpose programmable networking microchips are coming to market from the larger vendors.
2. On the FPGA side, Xilinx has introduced the concept of "softly" defined networks with a product called SDNet which allows a completely mutable programmable data plane to be defined, updated, re-defined and implemented. SDNet specifies a completely programmable data plane with an equally programmable control plane interface by leveraging the inherent flexibility of the FPGA. For example, rather than being bound by the data plane only processing packet headers, a content aware data plane can be implemented. Rather than the standards in this programmable data plane space diverging, Xilinx have developed a cross-compiler allowing implementers to translate P4, an open-source standard for describing data plane functionality, into SDNet. A number of vendors have implemented off-the-shelf network devices with an FPGA data plane leveraging SDNet for those wishing a completely custom data (and optionally control) plane implementation.

### FPGAs in latency-sensitive automated trading

Modern financial markets are run by automated algorithms which execute rules as to what should be bought and what sold. These automated systems replace the function of the open outcry floor, which used to host day traders and market makers. To achieve profitably, sophisticated proprietary trading and trade execution algorithms are used to decide what position to take at any given time. Trading is time sensitive – wait too long, and the available prices change...take too long to respond and the system is trading on stale data.

Minimising the time between receiving information from the outside world, making a decision to trade and sending an order back to the markets has a direct bearing on the success of the trades. Some trading decisions might be made based on low-frequency events (e.g. the release of key information about an economy, or a company announcement), but others are made based on high-frequency events such as other trades occurring in the market and changing prices. The latter has become synonymous with automated trading, and is called high-frequency trading (HFT).

The majority of HFT trades are triggered based upon incoming information carried over an Ethernet network; often from a stock or derivatives exchange. Trades also go out over an Ethernet network to a trading venue. High-frequency trading is often a matter of receiving Ethernet data, processing it and deciding whether it triggers a trade. If it does, Ethernet data is pushed back out as fast as possible.

Until around 2010, HFT systems were usually implemented in software running on commodity servers often using network adapters offering the lowest possible latency. These HFT systems could turn around incoming network data and generate outgoing trades in as little as two microseconds. The key factor limiting this turnaround time, or trading latency, was the time taken to get Ethernet packets from the network to the server's microprocessor running the trading algorithm and the resulting trades back out to the network. FPGAs had been around for some years however the overhead in implementing an HFT trading system on an FPGA has always been significantly higher than doing so in software on a microprocessor. So why bother?

As the HFT market matured, the drive to decrease latency intensified, given that, in most cases, doing so significantly increased trading profitability. Simultaneously, the markets have been driven to be more consistent and fair, progressing to tightly controlled co-located data centres. Ultimately this has improved the predictability and stability of the markets. Almost all FPGAs on the market have Ethernet-capable transceivers connected directly to the FPGA fabric. These transceivers generally allow communication between the FPGA fabric and the Ethernet network in low double-digit nanoseconds. Given the progress in trading system latency over the last five years, the significant latency gains achievable via FPGAs over a commodity server have made building trading systems with the critical path entirely in the FPGA an attractive proposition – despite the increased engineering overhead in implementing HFT trading strategies directly on FPGAs. FPGAs were initially used for components of a trading pipeline – for pre-processing a stream of data from the exchange, or for performing the final set of checks and balances on orders being sent to an exchange. For a specific group of low-complexity but latency-critical HFT trades, using networked FPGAs is one of the biggest trends, with most HFT firms and banks employing them extensively as HFT trading platforms.

### FPGAs for network capture and timestamping

The volume of network traffic continues to increase year-on-year driven by such factors as booming smartphone adoption, IoT, faster broadband and increasing internet video. Monitoring, troubleshooting and securing network traffic therefore gets increasingly difficult. FPGAs are a natural fit for capturing network traffic given their large number of Ethernet transceivers and customisable logic. For example, the more powerful models from Xilinx have up to 128 25 GbE-capable transceivers. From a network capture perspective, those transceivers can receive up to 64 full-duplex Ethernet links. As completely programmable devices, FPGAs allow applications to process network traffic arriving on each port in parallel which provides the capacity to filter, buffer, aggregate or otherwise process these huge volumes of traffic.

Moverover network traffic on any given Ethernet link is often bursty, only using on average a fraction of the bandwidth over any given second. Using FPGAs, captured streams can be buffered and frames aggregated into a significantly smaller number of streams to be forwarded to remote analytics or security devices. When configured for this use case, the solution is often referred to as an Aggregation Tap or Packet Broker. Buffering size can vary enormously with some solutions leveraging RAM external to the FPGA and offering tens of gigabytes of buffering.

FPGAs are also capable of implementing logic for timestamping the arrival of each Ethernet packet to nanosecond (or lower) resolution. Why timestamp captured networked packets accurately? When analysing network traffic, the “when” is just as important as the “what”. For example, comparing timestamps on the same packet at different points in the network can allow congestion to be detected before network device buffers overflow, making remediation proactive rather than reactive. From a security perspective, knowing precisely when each network event of interest happened allows the exact causality of an incident to be reproduced.

Almost all of the Ethernet capture solutions on the market are FPGA-based and generally come in one of two forms: either as a pluggable board for an off-the-shelf server or as a specialised platform built around one or more FPGAs offering high port density e.g. 48 Ethernet ports in 1 rack unit (RU) as well as integrated local and remote management capabilities.

## FPGAs for networked video

Though video, streamed digitally over computer networks, has been generally available since the early-1990s, proprietary standards abounded both for the transport protocol and the video and audio that was being transported. Before the first standard for digital broadcast video, SD-SDI, was ratified in 1989, analogue video was the norm for the majority of the broadcast world. On the consumer side, the first consumer camcorders offering adherence to a new Digital Video (DV) standard arrived in 1995. There were also several proprietary versions of DV aimed mainly at professional and broadcast users. Following on the next year, in 1996, the real-time transport protocol (RTP) standard for delivering audio and video over IP networks was ratified providing an alternative to a number of incompatible transport and video/audio payload standards. It took until 2007 for SMPTE 2022, a harmonised standard for the transport of digital broadcast video over IP networks to become available. Standards have also had to keep evolving as resolutions and bitrates have increased - SD, HD, 4K, 8K. Video surveillance and videoconferencing usage are growing rapidly and are bringing their own standards to the mix. As Ethernet networks become ubiquitous, developing a dedicated video cabling infrastructure becomes less and less attractive and thus the preferred transport for digital video.

Given the plethora of digital video formats and resolutions, both standardised and proprietary, working with them and interfacing between them is extremely difficult. Digital video formats vary, they may or may not be compressed or security encoded high definition multimedia interface (HDMI). Digital video transports can be cables, digital files or via digital networks yet they all need to interoperate. Uncompressed HD bitrates today exceed 1 Gbps with 8K taking them up to 24 Gbps for a single stream. The real-time, deterministic processing required for live video switching or editing thus becomes a difficult technical problem to solve. The SDI standards and SMPTE 2022 also allow multiple simultaneous "streams" of video, audio, timecode and ancillary data such as closed captioning, making processing them far from straightforward. These challenges have proved to be an extremely good fit for FPGAs.

They come with the ability to interface with Ethernet networks as well as digital serial lines and provide precisely the type of digital signal processing (DSP) required for converting video, audio and metadata from one format to another - including digitising analogue video - at whatever rate is required, or switching between synchronised streams. It is particularly difficult to interface broadcast and consumer products where the adherence to the consumer standard may be incomplete or imperfect. FPGA vendors offer off-the-shelf FPGA software modules that know how to "speak" the most popular broadcast standards such as SMPTE 292M (HD-SDI) and SMPTE 2022-5,6 (video over IP) and consumer standards such as HDMI, making designing custom digital video processing solutions much easier. Their main competition is ASIC-based solutions. Though ASICs also excel in this sort of processing, their programmed inflexibility is at odds with the need to support the hundreds of implementations still being used and the constant evolution of both existing proprietary and ratified standards. The programmable nature of FPGAs, on the other hand, is an extremely good fit to support this evolution of standards. FPGAs can be found in consumer video devices such as digital video cameras, DVD and Blu-ray players and LCD televisions. They can also be found in studio cameras and most professional video processing equipment.

## Conclusion

FPGAs are playing a steadily growing role in a number of networked application areas. In this article, we have looked at their ability to offer a fully-programmable data plane for SDN, their use in offering financial trading firms the lowest possible latency, their preeminence in the booming network capture and timestamping market, providing the capability to offer high-density network capture and aggregation and finally their ubiquity in both the professional and consumer video market.

For networked applications requiring extremely low-latency, deterministic processing without the commitment of locking the application into hardware, FPGAs are often the best fit.

### **Santa Clara—Corporate Headquarters**

5453 Great America Parkway,  
Santa Clara, CA 95054

Phone: +1-408-547-5500

Fax: +1-408-538-8920

Email: [info@arista.com](mailto:info@arista.com)

### **Ireland—International Headquarters**

3130 Atlantic Avenue  
Westpark Business Campus  
Shannon, Co. Clare  
Ireland

### **Vancouver—R&D Office**

9200 Glenlyon Pkwy, Unit 300  
Burnaby, British Columbia  
Canada V5J 5J8

### **San Francisco—R&D and Sales Office 1390**

Market Street, Suite 800  
San Francisco, CA 94102

### **India—R&D Office**

Global Tech Park, Tower A & B, 11th Floor  
Marathahalli Outer Ring Road  
Devarabeesanahalli Village, Varthur Hobli  
Bangalore, India 560103

### **Singapore—APAC Administrative Office**

9 Temasek Boulevard  
#29-01, Suntec Tower Two  
Singapore 038989

### **Nashua—R&D Office**

10 Tara Boulevard  
Nashua, NH 03062



Copyright © 2018 Arista Networks, Inc. All rights reserved. CloudVision, and EOS are registered trademarks and Arista Networks is a trademark of Arista Networks, Inc. All other company names are trademarks of their respective holders. Information in this document is subject to change without notice. Certain features may not yet be available. Arista Networks, Inc. assumes no responsibility for any errors that may appear in this document. Dec. 20, 2018