PARPRO

Partners in Production and Design

FASTER, BETTER, AND CHEAPER ALTERNATIVES TO FPGA-BASED ACCELERATORS

Achieving Top Tier Performance for a Fraction of the Cost

WHITEPAPER

Table of Contents

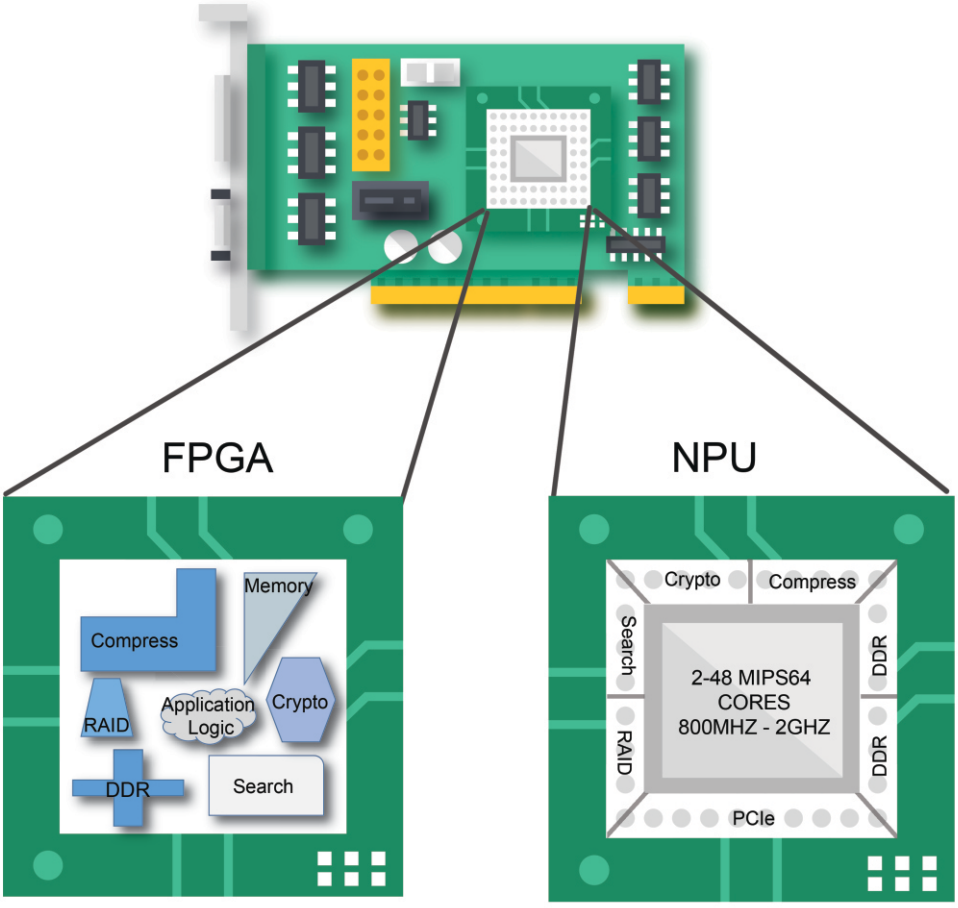
Introduction

Even as the number of cores in general-purpose CPUs continues to climb, many difficulties remain in building systems that are capable of handling large volumes of network data. This is due to several factors, including the ever-increasing rate of network traffic and some fundamental limitations in general-purpose computing architectures. System designers are constantly seeking ways to expand their system capacity beyond such limitations.

FPGA-based "accelerator" cards are commonly used to expand system capabilities beyond those of a general-purpose CPU. Such cards usually contain multiple network interfaces, a high-end FPGA device, and some number of licensed IP cores inside the FPGA for specific tasks such as pattern matching, cryptography, compression, or even storage-related mathematical operations. Connectivity between the general-purpose CPU and the accelerator card is generally done via a multi-lane PCIe connection between the two subsystems

While system design such as this offers significant increases in system performance, this approach has several drawbacks, most notably the long development cycles of FPGA code, limits on certain types of resources in the FPGA, and the expense of the FPGA itself and the licensed IP cores contained within. In addition, the task of the system designer now includes the added responsibility of partitioning the workload in an efficient and effective manner to achieve maximum throughput and minimal latency in the data path. There is, however, an alternative approach.

An NPU-based architecture offers many of the development benefits of traditional CPU platforms while also offering many of the performance benefits of the FPGA approach. And with Linux a prevalent operating system on both CPUs and NPUs, the task of migrating workload from the CPU to a more flexible NPU offers dramatically easier development and test cycles, which in turn accelerates your time-to-market.



"Typical architectures of FPGA and NPU based accelerator cards. Whereas FPGA architectures have complex workload partitions, the NPU approach is more flexible for a faster development cycle."

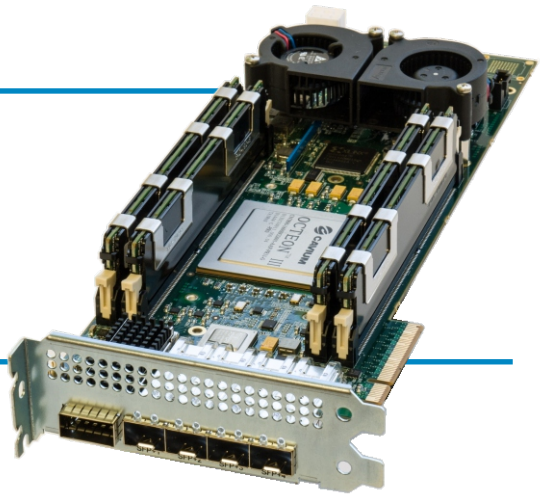
Faster Time to Market

For those applications which are very well suited to processing in an FPGA, nobody will argue that a general-purpose CPU can perform as fast. The example of signal processing applications, which is much more suited to FPGAs or DSPs than almost anything else, comes immediately to mind. However, there are many applications which are deployed onto FPGA-based platforms that are only moderately-well suited to that approach. For example, financial industry applications will use FPGAs as a way to get lower latency than an x86 processor, even though managing a large dataset like an Order Book isn't necessarily well suited to a platform which does not have cache or has limited memory bandwidth. The FPGA has become a go-to solution, but at the significant cost of development time.

The task of developing FPGA code is a very different one from developing code for a CPU or NPU. While there are many common elements, FPGAs present an entirely new realm of issues which need to be accounted for in the design and tested for during the quality assurance phase of any project. Aside from obvious concerns such as timing closure and limited quantities of certain types of resources (registers, clocks, etc.), more broad issues such as the difficulty of performing upgrades in the field and proper design of the data path pipeline mean that the development cycle for FPGAs are very long. In the modern era of rapid release cycles, a fast time to market is a critical aspect of any project and the FPGA approach doesn't easily support that.

Meanwhile, the benefits of the NPU approach include programming in high-level languages (C or C++, most commonly), the availability of both source-code and assembly-code debuggers, and a large collection of pre-existing libraries for routine tasks (i.e. sending and receiving packets) – all of which are familiar to existing application developers. Partitioning application workload often comes down to separating various application functions into separate programs, and then simply re-compiling those application components on the NPU with minimal changes – leading to a much faster development timeline.

The PARPRO O3E-110 Accelerator Card is a high-performance network processor PCI Express card designed for use in PCI Express compliant systems



A Better and More Flexible Platform

There is no dispute that, in terms of combinatorial logic, nothing can beat the performance of an FPGA. The FPGA code directly translates to gates and registers with specific timing constraints to guarantee that the output is correct. But, that fine-grained level of control in the FPGA code also reveals a weakness – more complex applications are exponentially more difficult. Often, common IP blocks such as PCIe or DDR3 come in pre-packaged blocks which are licensed from 3rd party vendors, which limits the flexibility of such components. Making a small change in a FPGA design can have massive impacts on timing constraints and resource allocations throughout the design.

At the core of an NPU, however, is a RISC microprocessor – often 64-bits with dozens of cores and running at speeds approaching 2.0GHz. Like its x86 CPU counterpart, it is surrounded with flexible I/O blocks that use standardized interfaces (PCIe, XFI, SGMII, etc.) and relatively large amounts of memory (in some cases as much as 256GB of DRAM). With that microprocessor comes the capability to easily add or change tasks and data path processing without fear of running out of a constrained resource like FPGA registers. Different firmware modules can be loaded and unloaded to provide different behaviors on-the-fly, and software upgrade is trivial (especially when compared to re-programming a flash device for an FPGA). Major interfaces, such as DRAM and PCIe, are built-in to hard logic gates in the ASIC and are ready-to-use in your application.

The flexibility of a programmable microprocessor would normally come with the penalty of lower performance, but NPU vendors address that problem by providing “offload engine” blocks within the NPU. Much like the way the CPU offloads tasks to the accelerator card, the NPU microprocessor can offload tasks to these offload blocks for increased performance. Common block include compression and decompression, encryption and decryption, key generation, pattern matching / search, and sometimes even RAID calculations for storage applications. This combination of small blocks of dedicated logic and the more generally programmable microprocessor give tremendous flexibility to the accelerator while retaining much of the performance gains that an FPGA-based design provides.

“The task of developing FPGA code is a very different one from developing code for a CPU or NPU. While there are many common elements, FPGAs present an entirely new realm of issues which need to be accounted for in the design and tested for during the quality assurance phase of any project.”

A More Cost-effective Approach

The price of electronic components is a constantly shifting landscape, but even within that FPGAs are notoriously difficult to price. Pricing is often done on a per-engagement basis, with list prices that are so high as to be downright obscene only to have discounts of up to 90% applied, depending on the specific program. Furthermore, developers often initially plan to use one speed grade of FPGA, only to discover mid-way through the development effort that a faster speed grade will be required to meet all timing constraints, at a significantly higher price.

Then, of course, there is the question of purchased IP for the FPGA. Such blocks are often licensed with a significant up-front cost and a per-unit charge as well, impacting both the development budget and the product's per-unit cost.

By comparison, NPU pricing is significantly more reasonable. While NPUs are known to be “more expensive”, that is in comparison to x86 platforms which have significantly higher global production volumes, and thus better economies of scale. By comparison to FPGAs, however, NPUs are the same price or cheaper – low-end NPUs can be as little as $15 with high-end parts under $1k, significantly less than high-end FPGA devices.

Conclusion

Fundamentally, the concept of an offload accelerator is a good one. But while the FPGA approach may be a common choice for an accelerator platform, an NPU should be considered. Many applications, especially those which deal directly with network traffic, can be efficiently offloaded into NPU-based architectures with a design which gives a faster time to market, a better and more flexible solution, and does so at a lower price point than the FPGA alternative.

"This combination of small blocks of dedicated logic and the more generally programmable microprocessor give tremendous flexibility to the accelerator while retaining much of the performance gains that an FPGA-based design provides."

PARPRO

Partners in Production and Design

About PARPRO

PARPRO is a full service design and manufacturing company with an emphasis on ODM solutions. We offer a comprehensive engineering-rich hardware solution with low-to-high volume manufacturing and integration/test capabilities, and pride ourselves on delivering simple to complex solutions making our manufacturing offerings competitive at virtually any volume and with any sourcing strategy.

We serve customers in the aerospace, gaming, telecommunications and industrial markets providing time savings and cost optimization by minimizing margin stacks throughout the value chain.

PARPRO Embedded Systems, a business unit of PARPRO, provides next-gen multi-core and switching platforms. We team with our customers and technology partners to deliver innovative embedded computer hardware in application-specific platforms. Whether you need a custom appliance, PCI Express, AMC, or ATCA, PARPRO can help you respond quickly to business opportunities.

Matthew Dharm
CTO, PARPRO
Embedded Systems

Author

Matthew Dharm is the Chief Technology Officer of PARPRO's Embedded Systems group. Matthew has worked in the embedded computer industry since 1998 and is an experience software and systems designer with special emphasis on single board computers across multiple platforms and architectures and high-performance mixed-solutions. His career has touched on such diverse markets as mobile communications, defense, medical, datacenter, and many others. Matthew graduated from Harvey Mudd College with a B.S with Distinction and was a founder of JumpGen Systems, which was later acquired by PARPRO to become their Embedded Systems group.

PARPRO

Partners in Production and Design

N. AMERICA HEADQUARTERS
PARPRO Embedded Systems (CA)
2772 Gateway Road, #102
Carlsbad, CA 92009
Tel: +1 (760) 931.7800

NEVADA - US
PARPRO Nevada
7390 Eastgate Road, #160
Henderson, NV 89011
Tel: +1 (702) 331.2700

MEXICO
PARPRO Mexico
Periférico Sur No. 1 Col. Obrera
Tijuana, Baja California México
C.P. 22180
Tel: +1 (664) 637.5602

TAIWAN
PARPRO TAIWAN
67-1, Dongyuan Road
Chungli Industrial Park, Taoyuan
Taiwan 32063
Tel: +886 3 452.5535

For General inquires, please contact
1-844-PARPRO-1 or +1 (760) 931-7800
Email: sales@parpro.com