



# Hyperchip Inc.

3391 Griffith Ave, St. Laurent, Quebec H4T 1W5  
(514) 906-2400  
(514) 906-2500

## Network Processors: FPGA vs. ASIC

For fixed functionality, dedicated custom hardware (ASICs) beats any kind of programmable hardware, so people new to hardware-programmable Network Processors often ask “How can an FPGA possibly beat a deep-sub-micron-ASIC software-programmable network processor?” Evolving networks requires adaptable equipment, and adaptability requires programmability. FPGAs are inherently reprogrammable which gives an FPGA many advantages:

First, an FPGA doesn't use software, so it doesn't need instruction fetch and decode pipelines, register sets, programmable ALUs, branch/address management, translation buffers, contention management, out-of-order processing, etc., which give a CPU its performance but cost die area and power. Commercial Network Processors dispense with some of these, but pay the price of increased parallel programming headaches. But all software NPs need to dynamically translate each instruction to figure out what to do, and then have a flexible processor do it, while in a hardware-programmable NP each stage is set up to simply do the proper operation.

Second, FPGAs are ideally suited to deep pipelines. Because each pipeline stage executes the equivalent of one instruction, each pipeline stage is very simple, and can be as wide as that particular operation needs. Hyperchip has pioneered an FPGA-optimized pipeline architecture that simplifies deep pipelining where all stages process their individual operations in parallel without requiring parallel software.

In both of these ways hardware-programmable NPs regain much of the efficiency that fixed function ASICs have over CPUs, but with the inherent programmability of FPGAs. A hardware-programmable NP running at under 200 MHz can deliver 40 Gb/s of full-router network processing from a single pipeline. In contrast Cisco's flagship 40 Gb/s SPP network processor requires 188 cores running at 250 MHz to deliver 40 Gb/s of network processing, and burns more power just for the NP<sup>i</sup> than Hyperchip's entire chip-set (including TCAM, table RAMs and buffer RAMs as well as the NP) consumes.

The third way that Hardware-programmable NPs beat software NPs is that FPGAs have a lithography lead of almost two generations, which is a four-year technology lead. While a decade ago FPGAs lagged ASICs, once logic passed DRAM as a driver of Moore's law, FPGAs became the ideal vehicle for pushing lithography because they are the most regular and repetitive logic chips. At 90 nanometers FPGAs passed most ASICs, and at 40 nanometers FPGA lithography passed even Intel's leading edge 45-nanometer CPUs for the first time, and software NP companies are still talking about 65 nanometer NPs.

## **Relative Cost and Power for 40Gb/s (100M packets-per-second) Full-Router Network Processing**

- Ten Quad-core 45 nm servers = \$30,000 and 2500 Watts.
- Three '40 Gb/s' Software NP Chip Sets + PCBs = \$ 6,000 and 150 Watts.
- One Hardware-Programmable NP chip sets = \$ 2,000 and 40 Watts.

### **Other Advantages:**

In addition to being more compact and power efficient any given generation, hardware-programmable NPs have a huge advantage over software NPs in the cost of progress. Software NPs bear the ever-increasing cost of keeping up with Moore's law; the average ASIC cost rose from \$4 million at 180 nm to \$10 million at 130 nm and \$25 million at 90nm<sup>ii</sup>, and high-performance network processors are more complex than average ASICs. In contrast, with hardware-programmable NPs the Moore's Law cost is borne by an FPGA supplier who amortizes it over a billion dollars in FPGA sales.

An additional advantage is the ease of programming. Massively multi-core software NPs are not ordinary CPUs and high-performance parallel programming is a notoriously hard problem. Software NPs compound that challenge with unique, highly-customized instruction sets that only a few people in the world understand well enough to obtain maximum performance with. In contrast, an FPGA-based hardware-programmable NP is a single pipeline, and a high-tech hub will have thousands of fluent FPGA programmers who can add features to it.

### **Specific Comparison:**

To reach 40 Gb/s with commercial network processors would currently require two ingress network processors and one egress network processor to perform full router ingress and egress network processing at full line rate. Hyperchip has confirmed that its ingress and egress network processors will fit *together* in a single mid-sized 40-nanometer Stratix-IV FPGA from Altera.

Hyperchip's FPGA-based network processors include industry-leading features - multi-million-entry forwarding tables, large ACL spaces with layer 2 through Layer 4 fields, million-entry statistics, IPv4, IPv6 and MPLS LER/LSR, Ethernet and even SONet at full line rate, etc. At 40 Gb/s, Hyperchip's 100-million-packet-per-second network processor will be the fastest single-chip network processor on the planet, and if all goes well, the memories around the FPGA will allow Hyperchip to push the pipelines to 60 Gb/s.

Hyperchip has put a tremendous amount of work into memory efficiency to enable the whole hardware-programmable FPGA Network Processor chip-set fits on an industry-standard single-width AMC module, and in a microTCA Hub for an industry-standard MicroTCA chassis. This allows producing an industry-standard chassis system 17 times more compact and five times more power efficient than the closest comparable offering from Cisco, which is based on software NPs.

## History:

The first routers were based on off-the-shelf computers with commercial processors, and did all forwarding operation in software. However while this worked for slow-speed interfaces, network speeds have been increased far faster than processor speeds, creating the need for custom-built routing equipment.

The second generation of routers used dedicated fixed-function ASICs for forwarding, but this required need hardware every time new functions were added. Customers got tired of having to upgrade network engines four or five times during the life of a piece of equipment – the equivalent of doing four or five heart transplants, with the need to take equipment out of service, upgrade it, test it, and then bring it back into a network.

The network equipment then started moving to programmable network processors, or NPs, to regain the flexibility that CPUs had offered, but at higher performance than commercial CPUs. Many large companies, and quite a few start-ups as well, started building software-programmable NPs – basically custom build CPUs with custom instruction sets optimized specifically for network processing.

Although the network equipment industry wanted off-the-shelf network processors as fast as ASICs and easy to program as commercial CPUs, building software programmable NPs was no easy feat, and many companies pursued this exciting challenge. In addition to a host of startups, even processor giants IBM and Intel launched NPs.

But no commercial supplier was initially able to deliver NPs that work at the full line rate of the fastest telecom interfaces, 2.5 Gb/s and the then-upcoming 10 Gb/s. Most start-ups failed to provide enough memory bandwidth, causing their NPs to only deliver about ¼ of their expected performance, and even IBM, who understood memory bandwidth from its server business, only managed to reach half of their anticipated performance.

In a reaction to this, the network processor market fragmented. The high-end network equipment companies, such as Cisco, Juniper, Alcatel, and all of the terabit router startups, needed line-rate performance and so they started building their own dedicated NPs, specific only to the exact needs of one product family to allow maximum performance for that family. Even Hyperchip started down this path, and gave its top FPGA team, who had just finished successfully prototyping what would become the Matrix ASIC, the task of prototyping a network processor.

The Network Processor startups split into three further fragments; the first simply targeted lower-performance markets and worked on cost reduction through Moore's law. The second targeted more complex processor architectures, such as deep-pipeline Very-Long-Instruction-Word (VLIW) processors. And the third started designing massively multi-core processors, putting the burden of parallel processing on the on programmers.

At Hyperchip the FPGA team quickly realized that there was no way that a software-programmable NP could fit in a single FPGA. So they set out to emulate the initial feature set as fixed-function logic in the FPGA, reasoning that as new features were required, they could simply reprogram the FPGA with new features as well.

At the time FPGAs lagged custom ASIC by one lithography generation, and 0.25 micron FPGAs could not compete with 0.18-micron ASICs, so this was just for prototyping, with the expectation that we would then convert it to a software-programmable NP. But FPGAs were progressing very rapidly, and by the time Juniper had the industry's first 10 Gb/s full line rate software programmable NPs in 0.18 micron ASICs, Hyperchip had 10 Gb/s full line rate NPs in the newest 0.13 micron FPGAs, and without the millions of dollars in ASIC development costs. And the FPGAs were more efficient, too, requiring only two NP FPGAs to a half a dozen ASICs for competitors, making Hyperchip hardware twice as dense as any competitor at that time. And so the hardware-programmable network processor was born!

After the telecom industry crash, the competitive landscape changed. IBM and Intel sold off their network processors, which joined the low-performance camp. The VLIW architectures added more cores, and became a medium performance camp, claiming high bandwidth, but only if one did not actually do much processing at that bandwidth. And the multi-core architectures become massively multi-core, with Cisco building NPs with 188 cores working in parallel. And Hyperchip has optimized its FPGA pipeline architecture to where it can once again leapfrog software programmable NPs by a huge margin in performance, space, and power efficiency.

---

<sup>i</sup> On the Use of General-Purpose Multi-Core Processors in Networking Devices

Patrick Crowley and Jon Turner, Department of Computer Science & Engineering Washington University in St. Louis

<sup>ii</sup> IBS Handel-Jones Report 2004