JUNIPER NETWORKS | Engineering Simplicity

DAY ONE GREEN:
SQUEEZING EVERY LAST WATT FROM JUNIPER EXPRESS SILICON

The Express architecture delivers high density and power efficiency for transport and core routing applications.

By Sharada Yeluri

Day One Green

Squeezing Every Last Watt from Juniper Express Silicon

In a recent Day One Green paper on "*Improving Network Efficiency with ASIC Architecture and Technology*" Chang-Hong explained how Moore's law started slowing down in the late 2010s. While the density of logic still increased with each generation of the fabrication technology, SRAM densities were not improving at the same pace due to the sensitivity of the SRAM cells to process variations. Improvements in operating voltage, and therefore, intrinsic power consumption also slowed to a stop.

External memory technologies were not doing any better, either. In the past few decades, the gap between the processor and memory performance continued to increase at approximately 50% per year and the gap is now at about 1000 times. External memory density and power improvements have also slowed down significantly.

All of this meant that we could no longer rely solely on the process node advances to double the performance in the same power envelope. There was the need to develop an architecture that relied less on external memory accesses and reduced the data movement within the chip and to the external memories to reduce power consumption.

With that in mind, Juniper set about developing the *Express* architecture whose main intent is to deliver very high density and power efficiency for transport and core routing applications. The first family of Express chips were introduced in 2012 with the PTX Series (Packet Transport Routers). A decade later, we are currently sampling Express 5 (fifth generation) chips with best power density (watts/G) one can obtain for this class of chips.

How did we get such high-power efficiency? A forwarding plane architecture that trades some of the flexibility and scale of our Trio silicon and previous M-series architectures in favor of lower latencies thus lower power consumption. It's an interesting angle for a green engineering perspective and the subject of this paper.

Express PFE

A typical PTX router consists of one or more Express packet forwarding engines (PFEs). When the router contains more than one PFE, they are connected through the Express cell-based fabric as shown in Figure 1. A PFE typically consists of a packet processing complex, WAN interface that receives the traffic from the ethernet links, fabric interface to connect to other PFEs in the router, and a queueing/buffering subsystem (as shown in Figure 2).

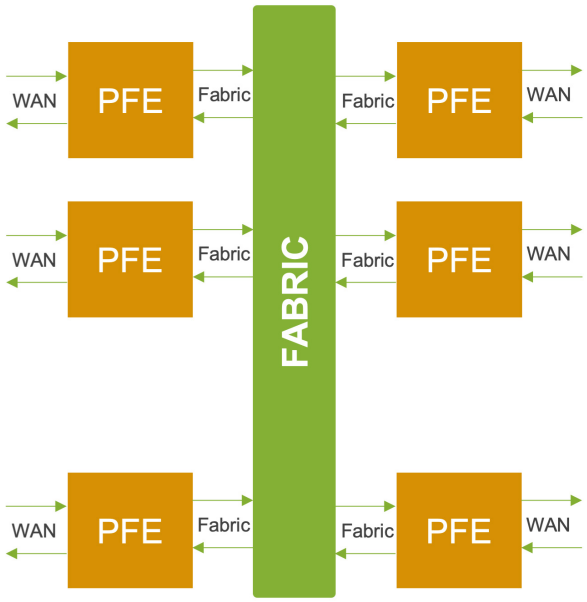

Figure 1 PTX Router

Fixed Pipeline Packet Processing

Express uses fixed pipeline packet processing for ingress and egress processing. These consists of a series of subsequent blocks. The packet headers (typically the first 128B-256B of the packet) flow through these blocks (see Figure 2).

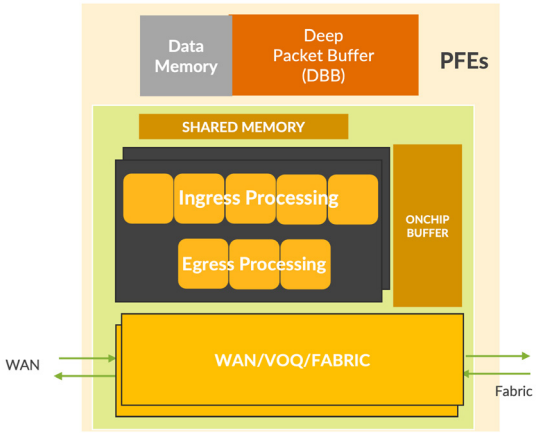

Figure 2 *Express PFE with Fixed Pipeline Packet Processing Complex*

Each block performs a specific function on the packet header and passes on that information to the subsequent block and so on. Compared to the network processor cores that are present in merchant silicon switches, or in the packet processing engines (PPEs) in our Trio family of chips, the Express architecture's fixed-pipeline implementation takes a lot fewer cycles to fully process a header. That's because all the functions are either hard-coded or implemented by executing highly specific microcode engines inside each block.

As an example, a typical Level 3 parsing to identify IPv4/IPv6 headers, perform the length checks, and compute the IPv4 header checksum takes 1-2 pipeline stages through the parsing block of a fixed pipeline. The same Level 3 parsing logic would take 2-4x more cycles with NPU/PPE because the software needs to execute a sequence of generic instructions to extract the various header fields from the Level 3 header and perform the computations.

Fixed pipeline architectures are twice as efficient in the die area and on average take 4-7x less latency to process a header compared to the processor cores.

On-chip Fungible Data Structure (Shared Memory) and the Caches

The Express architecture also carefully avoids accessing external memory for packet processing data structures like Forwarding Information Databases (FIBs), next hops, tunnels, and encapsulation tables, etc. *Accessing external memory for processing each packet not only consumes a lot more power but also increases the latency of processing.*

Even with the advances in memory technologies and with the advent of HBM (high-bandwidth memory), the usable bandwidth from an HBM2E device is around 2.6Tbps. HBM interface takes up a significant beachfront area of the die edge and Express would be severely limited by the amount of throughput it can pack inside each die if every packet were to access the HBM for lookups. Hence, in Express, most of the look-up data

structures are stored in a large fungible on-chip memory (referred to as *Shared Memory*) that can be partitioned between different structures at boot time. Express also allows for some FIB expansion to off-chip data memory that resides in the HBM.

Further, each client implements a lookup cache to store frequently accessed elements closer to where the processing is happening. *These caches additionally reduce data movement which in turn helps conserve power consumption.*

Hash Engines and Bloom Filters

Express reduces the hash/lookup table accesses to the central shared memory by using *bloom filters* that reside within the packet processing blocks. A bloom filter is a space-efficient probabilistic data structure that is used to test whether an element (key) is a member of a set (hash table) or not. Probing a "*key*" in the bloom filter indicates whether it is present in the hash table that resides in either the central fungible data structure or in external data memory. False positives are possible but there are no false negatives. *Using this approach can cut down on memory accesses by 70-80%, which again in turn saves power consumption.*

VOQ Architecture

The Express data path is based on Virtual Output Queue (VOQ) architecture (see Figure 3) which is a significant departure from the Combined Input Output Queue (CIOQ) architecture used in Trio and in many other high-end routing chips.

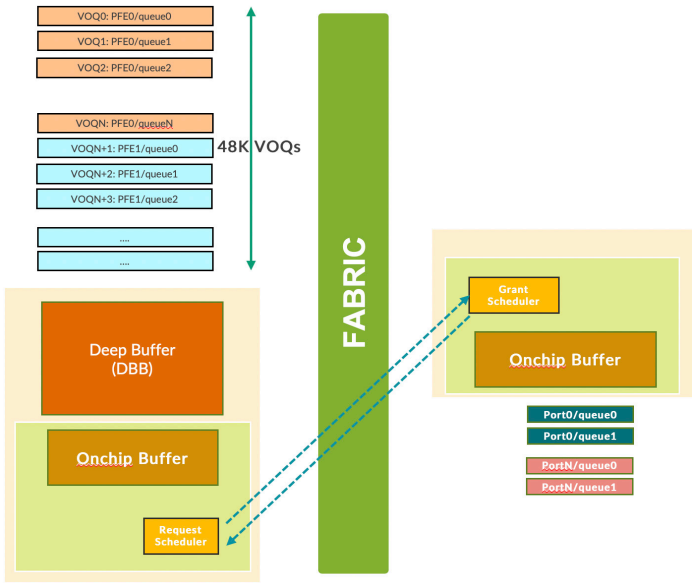

Figure 3 VOQ Architecture

In VOQ architecture, the packet is buffered only once in the ingress PFE after ingress packet processing, in a queue that uniquely corresponds to the final PFE/WAN port/output queue from which the packet needs to depart. These queues in ingress PFEs are often referred to as VOQs or virtual output queues. Every ingress PFE has buffer space for every output queue in the system.

A VOQ stays in the on-chip buffer when it is shallow and moves to the deep buffers in the external memory when the queue starts building up.

The VOQ requester waits for a group of packets to accumulate in a VOQ before it sends a request to the egress scheduler (in the egress PFE) for permission to send those packets over the fabric. The egress scheduler grants the access based on strict scheduling hierarchy and the space in its shallow on-chip buffer. Once a grant is given, the packets move to the egress PFE's on-chip buffer through the fabric and leave the PFE through the egress WAN ports.

This architecture is extremely power efficient for a few reasons:

- Packets are queued only on ingress. They reside in the on-chip buffer for shallow queues and move to the deep external buffer only during the congestion – so there's less data movement.
- The egress buffer is very shallow, and packets are admitted to the egress PFE only when it has space in the shallow egress buffer. So, packets never need to be queued in deep external memory buffers on the egress side – and there's less data movement again.
- Once a packet is accepted on the egress, it can't be dropped anymore. Compare this to CIOQ architecture (Combined Input/Output Queued) where the packets are queued in both ingress and egress PFEs, meaning when egress queues are congested, a packet could get dropped after it has moved to the egress PFE. Again, less unnecessary data movement and less power to operate it.

Cell-Based Switch Fabric

Once an egress PFE grants an ingress PFE admission for a group of packets, the ingress PFE 'cellifies' the packets in that group, attaches sequence numbers to these cells, and sprays them over available fabric links. On the egress side, these cells are put back in order and the packets are assembled. By chopping the packets into cells and spraying them across the links, Express can achieve >95% utilization on these links connecting PFEs to the fabric. The Day One Green paper, by Harshad Agashe, "Connecting Multi-Terabit Packet Processing ASICs Using High Throughput Multi-Terabit Fabric ASICs", explores how cell-based fabric is superior in power and performance to Ethernet switch-based fabrics used by many other network vendors.

Multi-Slice Architectures

In the last two generations of Express chips, we squeezed in multiple PFEs inside the same die by sharing the packet processing data structures and the on-chip packet buffering between the PFEs. *This enables us to have a smaller SRAM footprint on the die and not only improves the area efficiency but also saves the leakage power associated with these SRAM structures.*

Power Optimization Techniques during Implementation

While a good packet processing and data path architecture that reduces the data movement and decreases the processing latency can offer significant savings in power efficiency, power can further be optimized by advanced implementation choices.

The total power consumption of a chip consists of static and dynamic power. Static power is usually the leakage power of the logic gates and the SRAMs, and it is directly proportional to the voltage of operation and the process node. Leakage power is becoming more and more prominent in the latest process nodes, and it is preventing some vendors from lowering the operating voltages on their devices. In Express implementations, we focus on reducing the dynamic power of the chip as that directly relates to the switching activity.

The dynamic power of an integrated circuit consists of switching power and short circuit power:

Dynamic Power = Pswitching + Pshort-circuit

Pswitching = $a.f.Ceff.Vdd^2$

Pshort-circuit = Isc.Vdd.f

You can see that switching and short circuit power are directly proportional to the clock frequency (f) of operation. Supply voltage is often the lowest voltage recommended by the vendor for the process node. Reducing the voltage affects the performance of the SRAMs and the logic gates and can push the minimum frequency of operation to a lower number, which in turn reduces the packets per second and the bits per second you can achieve with a given piece of silicon. In Express chips, we usually keep the operating voltage at the recommended setting by the vendor for that process corner.

Optimizing the Frequency of Operation

While it seems obvious that reducing the frequency of operation would reduce power consumption, it can also reduce the performance (power/gigabit). Then, to get the same overall throughput from the PFE or the system, you would have to add more logic in the PFE or add more PFEs in the line card/system. Both would add to the power consumed by the ASIC.

A network chip with tens of terabits per second of bandwidth, with central buffers and data structures, has many wide buses that need to be routed across a large die. Operating frequency of the chip decides the width of these buses. A wide bus is often required at lower frequencies to get the same bandwidth. And routing a wider bus involves more repeaters and therefore more power consumption.

Another factor to be considered is the re-use of the existing IP components which might not scale for higher frequencies. Similarly, SRAM performance might not scale with frequencies, so to realize a logical memory you would be forced to use multiple stammer SRAM structures.

For each generation of Express chips, Juniper carefully considers the process node, wiring congestion, IP re-use, SRAM scaling to select the frequency of operation that reduces the overall system power.

Clock Gating

We provide the ability for software to clock gate (or turn off the clock) for functions that are not used or enabled for the users. By turning off the clock to large chunks of logic, you can save the clock tree as well as the switching power (as the frequency component goes to zero).

For example, the clock network for the logic/functions associated with unused WAN ports is turned on/off by the software as the user attaches/detaches the cables to the WAN ports in system.

We also implement dynamic clock gating. Here, if the output of a flip-flop is not used in a specific cycle, the clock could be turned off for that flop to prevent the output from switching in that cycle. Dynamic clock gating is inferred by the EDA tools during the synthesis (conversion of the Verilog behavioral RTL code to gates) when the designer writes the code for the flip-flops in a specific format. *Express uses advanced EDA tools and methodologies to proactively identify and fix all the clock gating opportunities that the designer missed. Our designs achieve >90% efficiencies in turning off the clocks to the flops when their outputs are unused/not changing.*

Power Optimization in Placement and Routing

Lastly, Express uses advanced power-driven placement and routing tools and methodologies to optimize power consumption even further. This is the topic for a future paper. Stay tuned.

Summary

Express ASICs are all about switching and transporting packets in core/peer and transport routers as fast as possible with the least possible power consumption. With a novel fixed-pipeline VOQ-based architecture, advanced techniques to reduce the data movement within and across the ASICs, 2.5D packages with HBM memories inside the package, and by using the latest EDA tools and methodologies to reduce the dynamic power even further, Express has not left any stone unturned in achieving the lowest power per gigabit of traffic. Express 5 boasts some of the most efficient silicon in the market with 28.8Tbps of throughput from a single package.

Speak to your Juniper Networks' account manager or Professional Services rep about Express silicon in the PTX Series of Packet Transport Routers.

© 2022 by Juniper Networks, Inc. All rights reserved. Juniper Networks and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo and the Junos logo, are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners. Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.