Disclaimer

ASICs, magic and pro wrestling are all closely guarded secrets
  ▪ Speaking publicly on the topic may lead to long rides in windowless vans

Everything here is public, somewhere

Dates are based on chip/switch announcements

If you make chips, please don’t sue me or get me fired
  ▪ I like my job.
Target Audience

- Designed for enterprise network engineers
  - Some knowledge of ASICs
  - Some knowledge of software
  - Experts in neither

- Not designed for ASIC experts
  - Presentation goal is “good enough” not EE degrees
  - I’m a software guy, go easy

Comments
- @PeteCCDE
My Qualifications

The Bad News

- B- average in Computer Science
- Never took a physics class
- Afraid of electricity
  - Had to call maintenance to replace a thermostat
- Can not spell oscilloscope

The Good News

- Former Cisco TAC Escalation
  - I fixed broken routing hardware
- Webscale DC Design
  - Does the ASIC fit the need?
- ASIC Translator
  - ASIC value/tradeoffs to business and $employer
Agenda

How ASICs are made
CPUs vs ASICs
ASIC Pipelines
Buffers
Chassis Architecture
ASIC Families
Programmable ASICs
Asking Vendors Questions
Making an ASIC

Magic, I assume
I really have no idea
I’m a software guy
Let’s talk about something else

ASICs being made?
CPUs vs ASICs

ASIC = Application Specific Integrated Circuit
- Build a circuit to do a thing
- Anything high speed with lots of ports

FPGA = Field Programmable Gate Array
- Like an ASIC, but you can change it
- ASR1k QFP is an FPGA* (sorta, refer to slide 3)

Spectrum of tradeoffs
- Flexibility vs Speed vs Power (vs Cost)
  No 3.2Tb CPUs
  No ASICs that support all features
Compromises

Everything is a trade off

- Power vs Space vs Heat vs Cost vs Features vs Speed
- Nexus 3548 - Ultra Low Latency (<50ns port to port)
  - Until you turn on L2 forwarding (190ns)
  - Until you turn on L3 forwarding (250ns)
  - Tiny tables (comparatively)
- Broadcom Tomahawk’s 16mb shared buffer
  - But not fully shared, partitioned (no one port gets all 16mb)
- Mellanox Spectrum shared buffer + low latency
  - But only a max of 64 ports

Time to market

- Everyone is pretty close
  - Cisco is losing the clear edge
- 1-3 year gap at most
  - Don’t like the chip? Wait a little bit for the next one
- Everyone optimizing for different things
Pipelines

ASICS have “pipelines”
- The series of circuits that do specific actions

Pipeline determines feature set
- Some features simply don’t exist

Pipeline may be limited to one action
- VxLAN decap or L3 lookup
- GRE or MPLS

Recirculation
- Send it back through the pipeline a second time
- Double the latency (600ns Switch becomes 1200ns)

https://people.ucsc.edu/~warner/Bufs/trident2
Buffering

Buffering is a religion

- Red Sox vs Yankees, Madrid vs Barcelona, UNC vs Duke, All Blacks vs Les Bleus

Buffering depends on ASIC and form factor

- Chassis?
- On chip?
- External “bonus” buffer?

Buffers = Latency

Buffers aren’t evil

- But where and why matters
Buffering Cont’d

When are buffers used?
- Store and Forward
- Egress port congestion (two inputs, one output)
  Consider statistical probabilities
- Burst > Pipeline Speed (packet sizes matter)
- Speed Change (100g into 10g)

When don’t buffers matter?
- Cut through
- Same speeds, no congestion
Single Chip “Shallow” Buffers

On Chip

- Buffer is part of the pipeline*
- Not “deep”. Generally MBs
- Shallow buffers = high speeds

Can also mean low latency

Trident 3 Pipeline from https://people.ucsc.edu/~warner/Bufs/trident3.html
Sidebar: Chassis Architecture (Arista 7300x)
Sidebar: Chassis Architecture (Arista 7300x)

Virtual Output Queues

- Deep buffered chassis
- Prevents fabric congestion
- Packets live in VoQ until given credits to send over the fabric
- Just like a SAN
- Single linecard can still have congestion
- Incast to a linecard still exists
Sidebar: Chassis Architecture

Chassis are spine and leaf networks

- You just don’t know it

Nexus 9500R whitepaper
Single Chip “Deep” Buffers

Family of ASICs with deep buffers
- Measured in Gigabytes

Often buffer off chip
- High speed memory, not L1/L2 cache like shallow buffer ASIC

Nothing comes for free
- Buffer slower than transmit speed
- Consistent congestion doesn’t matter
- Extremely high latency

NCS5500 Buffer Architecture
Pete’s Opinions on Buffers

Deep Buffers

- Long distance transmissions (dark fiber, internet)
  Loss of a packet due to microsecond congestion has BIG impact on high RTT TCP
- You hate money
  Insurance isn’t cheap

Shallow Buffers

- Literally everything else

All buffer marketing is cooked.
This is a religious debate.
I don’t want to talk unless you have real world data.
Broadcom ASIC Families

**Broadcom**

- **StrataSGX** – Datacenter (mostly) 1RU. Named after missiles
  - **Feature Rich**: Trident, Trident2, Trident2+, Maverick, Trident3, Trident4
  - **High Speed**: Tomahawk, Tomahawk+, Tomahawk2, Tomahawk3

- **StrataDNX** – Deep buffers, chassis chips. Named after Israeli cities
  - **Buffers + Medium Speed**: Arad, Qumran, Jericho, Jericho+, Jericho2

**Mellanox**

- **Spectrum** – Low latency, feature rich

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Trident 2 (48x10 + 6x40)</td>
<td>Spectrum (32x100)</td>
<td>Tomahawk (32x100)</td>
<td>Spectrum2 (32x200)</td>
<td>Qumran-MX (48x10+6x100)</td>
<td>Trident 3 (48x25 + 8x100)</td>
<td>Jericho2 (24x100 + 6x400)</td>
<td>Trident 4 (32x400)</td>
</tr>
</tbody>
</table>
Programmable Chips

Not Programmable
- Can’t talk to the SDK
- Fixed Pipeline

Semi-Programmable
- Can’t talk to the SDK
- “Flexible” Pipeline

Fully Programmable
- Same “Flexible” Pipeline
- Full SDK access (i.e., P4)
Let’s Be Honest….

You’re not going to program an ASIC
  - It’s great for vendors

Even with P4, it’s hard

No one builds routing protocols, why data paths?

Some valid use cases
  - Custom metadata (probably not you)
  - Stock feeds trade on stock ticker
Questions to Vendors

Preface:
- VARs (probably) won’t know
- Your vendor SE (might) not know
- Someone knows, make them earn their keep

What does the chip do well?

What does it do poorly?

What trade of was made for $\text{magic\_function}$?
- If it’s low latency or high bandwidth or feature rich, what does it not do?

If I need more speed or more features, what would you position?

Remember: nothing is free. What’s the tradeoff?
Thank you!

Visit us at cumulusnetworks.com or follow us @cumulusnetworks