OPENFABRICS ALLIANCE

Fabrics – Why We Love Them and Why We Hate Them

11TH ANNUAL INTERNATIONAL OPENFABRICS SOFTWARE DEVELOPERS' WORKSHOP

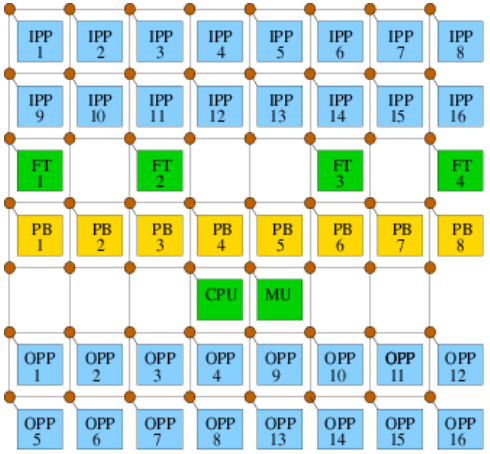
#OFADevWorkshop

Dave Dunning
Intel Corporation

Why we love Interconnect Fabrics, or, “I can design transportation systems”

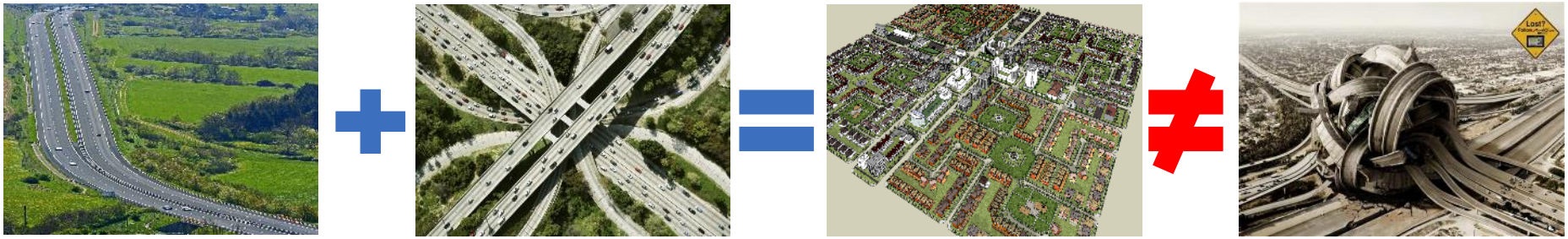
OPENFABRICS ALLIANCE

- They move bits around…
  - Communication between resources
  - Start at some “address” and go to another “address”
    - On-die, fit everything into a nice gridded pattern, at an intersection, go towards your destination
    - Between die, travel along wires, when you get to a fork in the road, pick the path that goes to your destination
- How hard can it be to design?
  - Just copy our transportation systems, but make it better



Fabrics – Often the most "scrutinized" shared resource in a system

- The technology seems simple, can be visualized
  - Switches, arbiters, buffers, wires, tables, counters…
  - Distributed, usually organized, repetitive
  - "Features" can be "invented" by anyone with any background
  - Make them work the way I want to use them

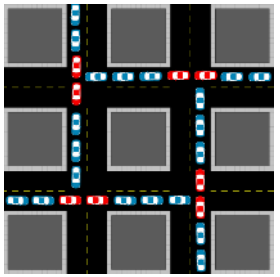


The most common sentence uttered by a Fabric Designer:

"Stop helping me!"

Why we hate interconnect Fabrics, or Pretending packets are cars is just wrong

- Contexts, threads often don't share resources well
  - Packets can't back up

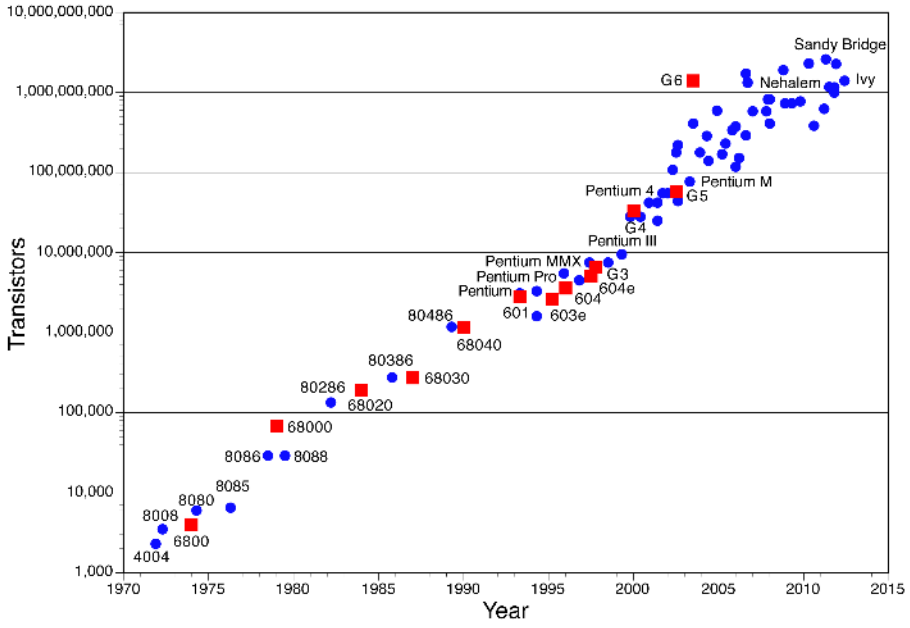


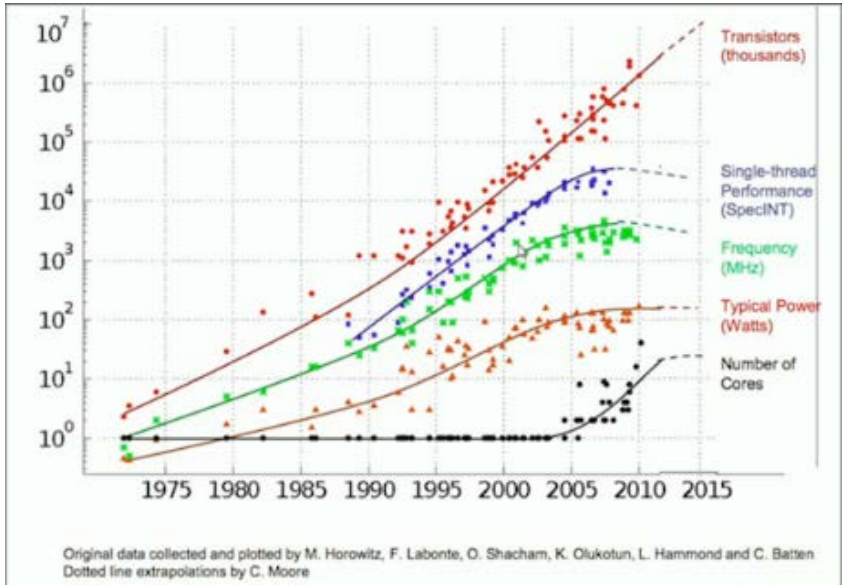
- Different flows often don't mix or coexist very well
  - Order of arrival often matters

- Balancing the use of resources is often difficult
  - What seems like a good idea sometimes isn't

Chip Scaling

- Moore's Law – 2x transistor density increase every 2 years
- Dennard Scaling (MOSFET scaling) – Power density stays constant





Chip Scaling – How this affect Fabrics; A hardware View

- WAS:
  - Compute a scarce resource
  - Simple memory hierarchies
  - I/O was the bottleneck
  - Form factors (FF) – chip counts, chip I/O, connectors
  - Cost: Designing, building, # chips, wires, connectors
- Key Metrics
  - Performance (BW, freq.), cost
- Less Key Metrics
  - Latency, FF, power

- IS:
  - Compute inexpensive
  - Complex memory hierarchies
  - Off-chip I/O faster than on-chip
  - Form factors (FF) – packaging, cooling external I/O interfaces
  - Cost is components, integration/packaging, power
- Key Metrics
  - Performance (BW), Cost, Power, FF
- Less Key Metrics
  - Latency, frequency

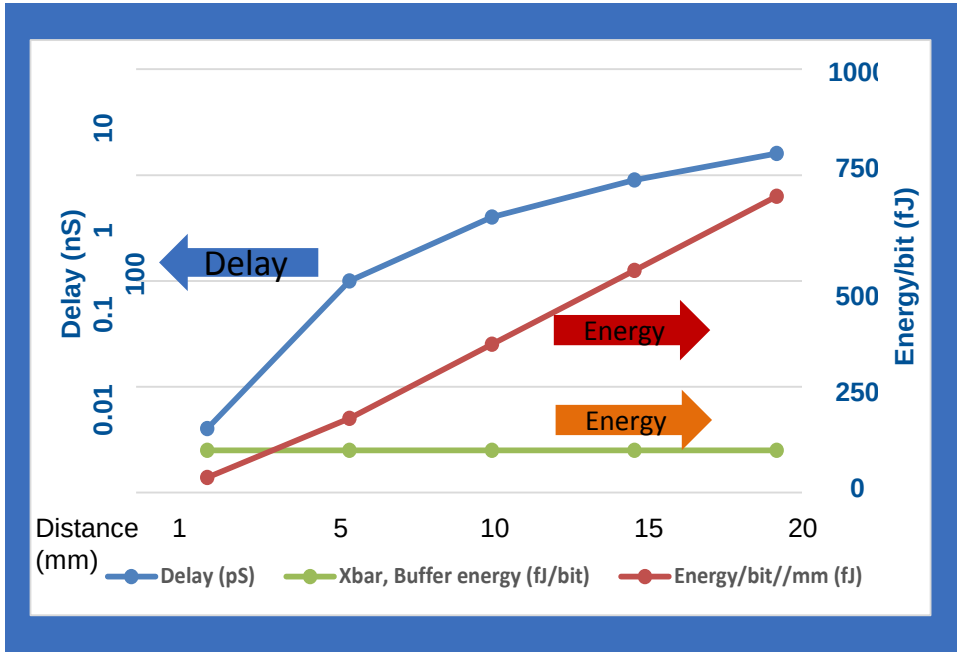
Chip Scaling – How this affect Fabrics; A software View

- WAS:
  - Throw the HW "over the wall", SW will find the performance
  - Maximize compute utilization
  - Users don't worry about Memory, let the OS handle it
  - Cost: Time to released code, limited "legacy" code validation
- Key Metrics
  - Performance, cost, user interfaces
- Less Key Metrics
  - Power (system management)

- IS:
  - SW complexity due to sharing, parallelism
  - Compute abundant
  - Efficient memory usage to save power
  - Cost: Time to market, backwards compatibility design time
- Key metrics
  - Performance, power, user interfaces, cost
- Less Key metrics
  - None – SW rules the systems

Process (Chip) Scaling; Building Fabrics

- On-die bandwidth
  - Lots of traces available
  - Tools must mature
- Energy (to move bits)
  - Linear with distance
  - Energy to move bits is your enemy
- Latency
  - RC product
  - Increases exponentially with distance



Cross die wires will need to be buffered
Logic (switches) at clock cycle intervals
Local BW is cheap, cross die BW will cost time and energy; optimize for locality

Process (Chip) Scaling Is Helping You, but less so moving bits

22 nm	pJoules	8 Bytes	Description	pJ/bit
FP Mul	6.4	A = B * C	8B/Operand	0.10
FP Add	8.1	A = B + C	8B/Operand	0.13
FMA	10.5	A = B * C + D	8B/Operand	0.16
Xbar Switch	0.86	8B per port	12 ports	0.01
On-die Wire	11.20	8B per 5 mm	50% toggle	0.18
Phys Reg File	1.2	8B R/W	2KB, 3 ports	0.02
SRAM	4.2	8B R/W	Small (8KB)	0.07
SRAM	16.7	8B R/W	Large (256KB)	0.26
In pkg DRAM	192	Stacks	64B accesses	3.00
Off Pkg DRAM	640	DDR	64B accesses	10.00
In pkg Wire	19.2	< 20 mm		0.30
Off pkg wire	128	< 200 mm		2.00
In Cab wire	320	< 2 m		5.00
Optical	640	> 2 m	Cost and area	10.00

Will scale well with process and voltage

More difficult to scale down

Will scale well with process, less well with voltage

Can move to 2-4 pJ/bit range; Depends on demand, volume, leading to cost

Most challenging technologies to scale going forward

Power = Energy * Frequency + Leakage

Pulling it Together in the Future

- Formerly scarce resources are now plentiful
  - Compute, logic, on-die memory, wires
    - Technology can support very large bandwidths
  - Moving bits will dominate the power consumed
  - Memory: DRAM still scaling, lagging logic, other technologies may mature
  - Electrical wires remain the low cost choice, optical provides distance and minimizes cables, but requires power and $
    - Optical 5 – 15x higher $ per Gb/s than electrical for distances <2 meters
- Specialized building blocks will proliferate
  - Simpler, faster, lower energy, power gated (off) when not used
  - Sea of resources
  - OS problem, managing, sharing the resources

A Software View

- SW engineers must become system engineers
  - HW engineers have hidden many system challenges from SW engineers
    - SW engineers need to rethink what they are willing to “pay” for features
    - Do you want heavy weight Oses?
    - Memory management, Scheduling, Execution model
    - Methods to exploit parallelism
    - General purpose vs. specialization
- Standardization
  - Standardize when time-to-market is reduced
  - Cost reduction due to increased volume
  - Independent of implementation

Conclusions

- Communication Packets are not cars
- Process scaling Moore's Law continues to provide more transistors/silicon area
  - Compute, logic trends to inexpensive
  - Communication trends to more expensive
- More specialization (accelerators, fixed function)
- SW evolving from performance to locality-based system management
- Standardize for time to market, cost between die, unclear on-die

Thank You

OPENFABRICS ALLIANCE

11TH ANNUAL INTERNATIONAL OPENFABRICS SOFTWARE DEVELOPERS' WORKSHOP