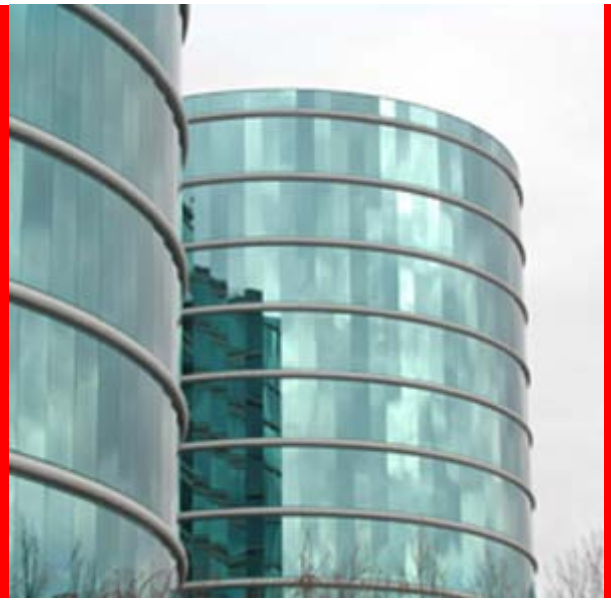
New tools for designing high-performance computing systems

Jon Lexau



ORACLE

New tools for designing high-performance computing systems

Jon Lexau
Consulting Hardware Engineer, VLSI Research Group
Sun Labs, Oracle

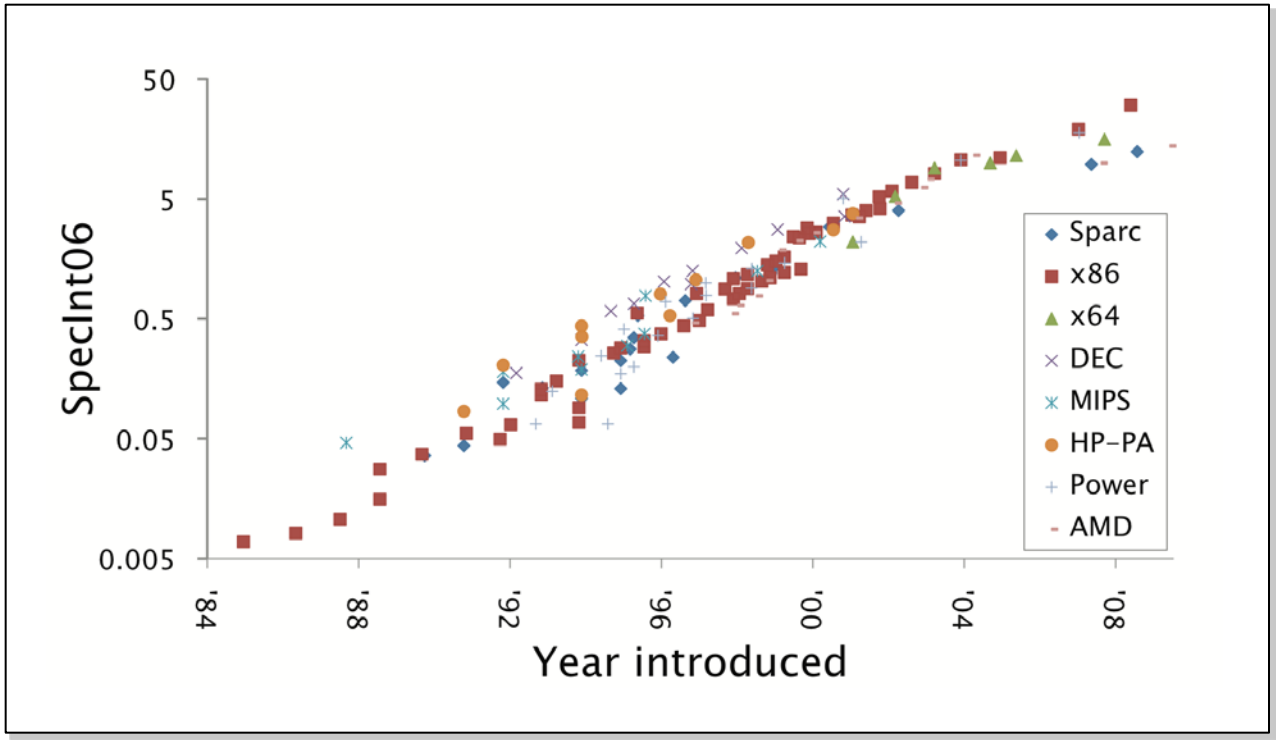
A disclaimer

- This is **not** a talk about upcoming products
- Sun Labs
  - ~100 people in Oracle working on advanced research
  - Looking at hardware, software, OS, languages, DB…
  - Providing a technology toolbox for product groups
- Some of this work was supported by DARPA (HR0011-08-09-0001)

A high-level view: “Life has been good”

Computer system performance has been growing at 35% CAGR

- Can we continue this pace? How?



System integration is the key

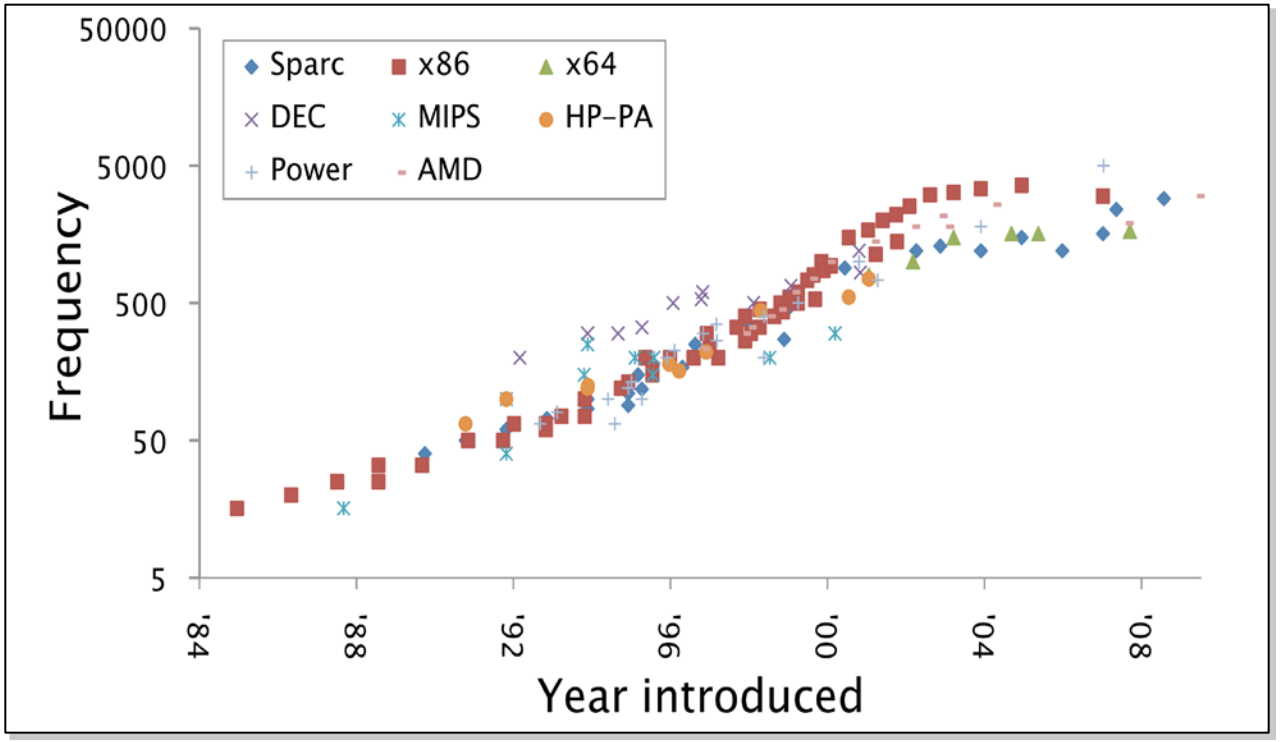
Performance = instructions/cycle * cycles/second

System integration is the key

Performance = parallelism * frequency

Improving performance

Performance = parallelism * frequency



Data from M. Horowitz

But growth here is stalled

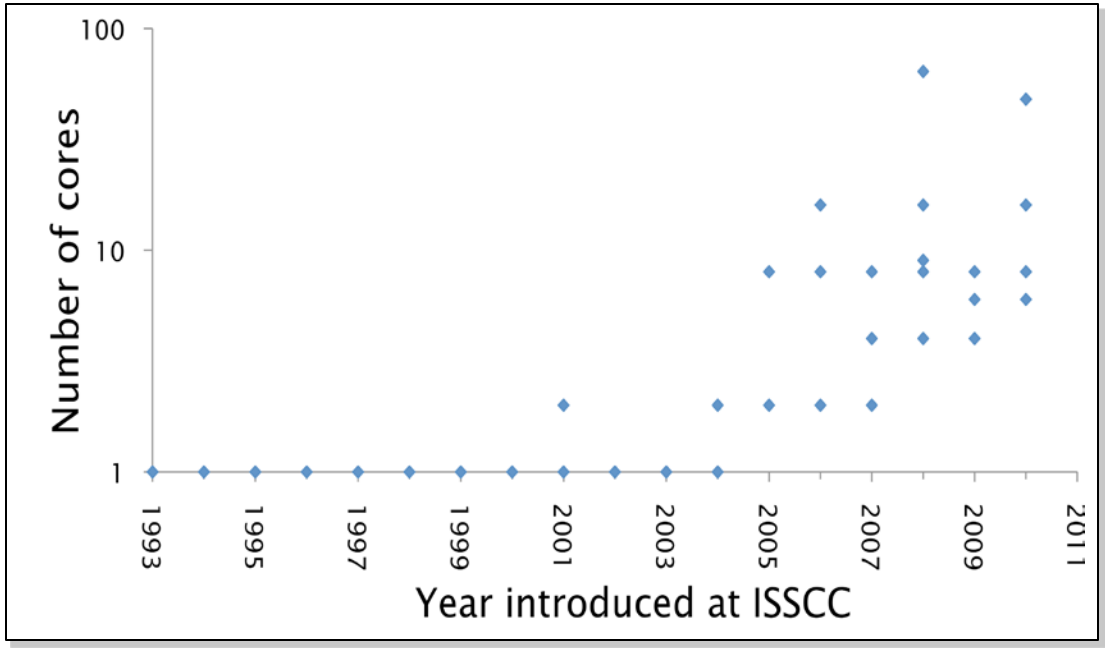
- **Power is a top-line constraint**
- **Complexity must be manageable**
- **ITRS-01 predicted 12GHz CPUs this year…nope!**

Improving performance

Performance = parallelism * frequency

So must push here…and harder!

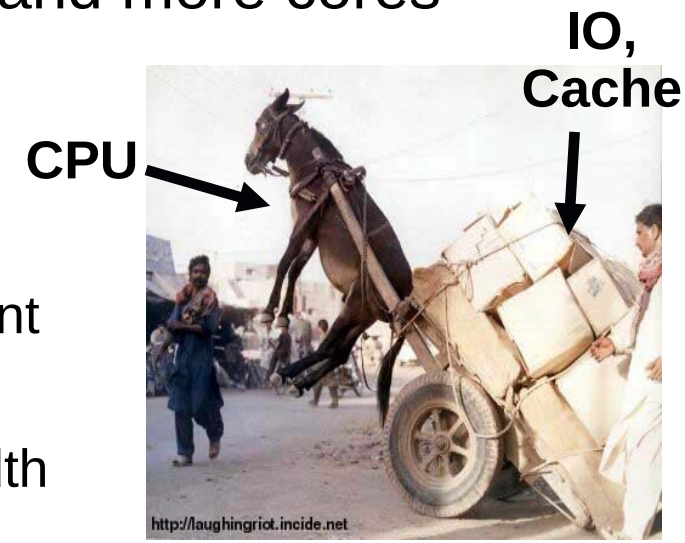
- **Exploit instruction/thread/task parallelism**
- **We need more "stuff" on a chip**
  - **More cores, memories, switches**
  - **More integration (bigger chips)**



A side point on multi-core chips

How to integrate cores and memory

- Chips today are integrating more and more cores
  - What about the cache?

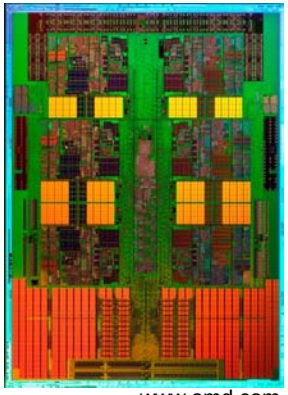


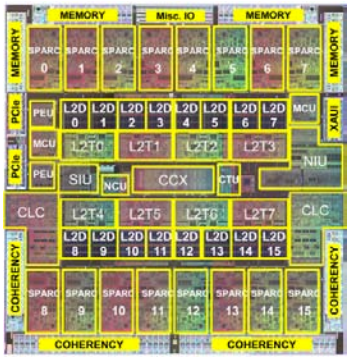
B. Colwell

- If cache per core stays constant
  - Then miss rate per core stays constant
  - Total misses goes up with core count
  - …using up all of the chip I/O bandwidth
- So must increase cache per core with multi-core chips
  - Cache must integrate *disproportionally* faster than cores!
  - Else they load the cart down more than the donkey can take

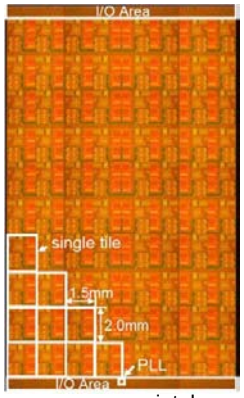
Multicore chips are here to stay

- All the kids are doing it
  - Intel, IBM, Nvidia, Oracle, AMD, …
- 6 to 80 cores now
  - How do we get even more?
  - Plus more memory?

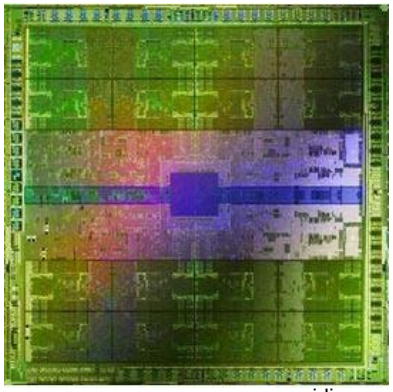

www.amd.com

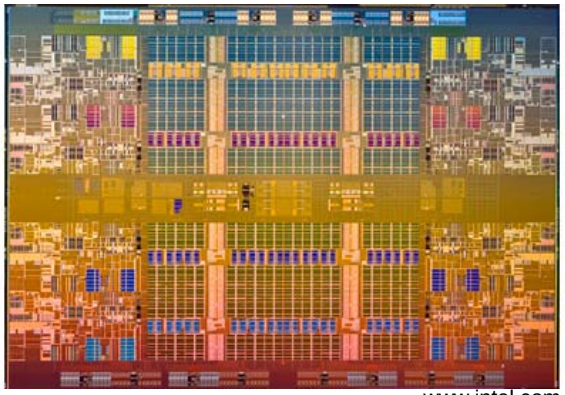


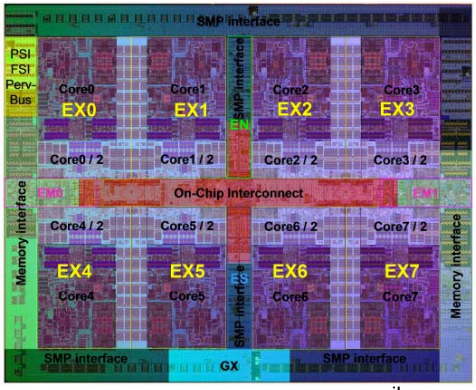
www.oracle.com



www.intel.com


www.nvidia.com

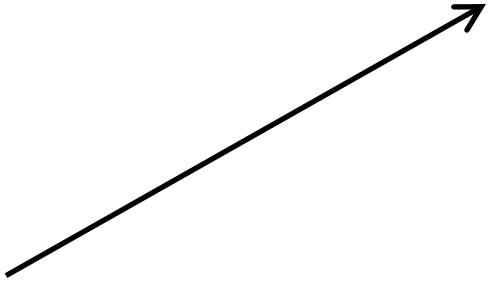

www.intel.com



www.ibm.com

System integration is the key

Performance = parallelism * frequency



So must push here…and harder!

- Exploit instruction/thread/task parallelism
- We need more "stuff" on a chip
  - More cores, memories, switches
  - More integration (bigger chips)

But growth here is stalled

- Power is a top-line constraint
- Complexity must be manageable

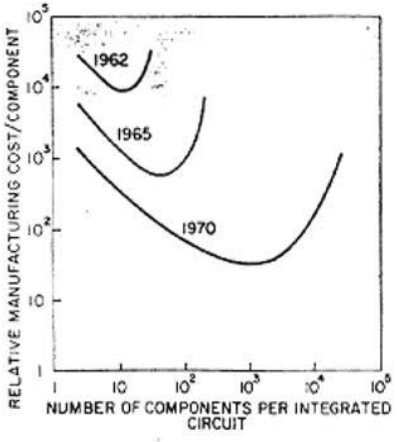
How do we get bigger chips?

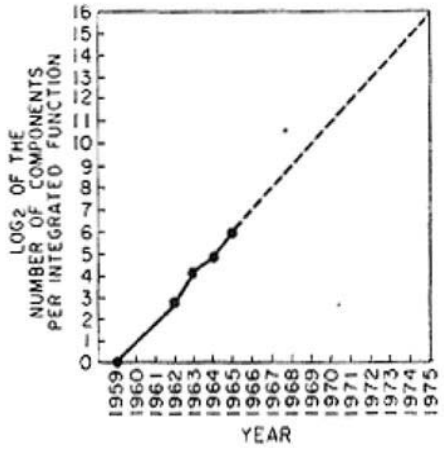
- Size is limited by silicon yield
- Cannot simply aggregate lots of chips
  - Performance is limited by I/O

The obligatory Moore's Law slide

"It's the economy, stupid"

- Fundamentally about economics, not performance
  - Want as many devices as possible on a chip (amortize costs)
  - Want small chips, too (avoid yield hit of large chips)





- 2x # transistors every 12…24…18 months
  - Can't speed this up to compensate lesser scaling elsewhere

The end of Moore's Law

"Any industry reliant on exponential growth will eventually be disappointed"

- Moore's Law **will** end relatively soon. Why? Money.
  - Clearly limited global financial investment in semiconductors
    - No longer an infinitely elastic market
    - Next-gen fabs cost $5B to build (> 1% of the market)
  - No exponential is forever (Moore, ISSCC 2003)
- But wait. Didn't this drive scaled performance?
  - Parallelism: duplicate functional blocks, cores, memories
  - Integration: make deeper global communication structures
- A roadblock on the path to really big chips?

Big chips *are* good

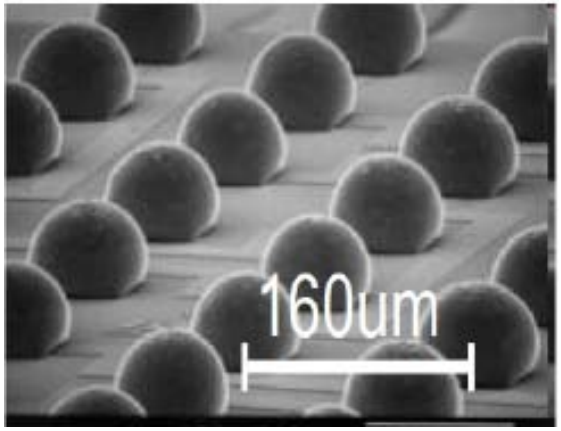
More hardware == more performance

- Pack together lots of functional units
  - 1000s of cores and memories on a chip
  - Can we combine heterogenous technologies? (Flash, DRAM)
- Cut global communication latency
  - “Bandwidth is about money; latency is about God.”
- Cut total system power dissipation
  - Need less power for I/O circuits

- Too bad we can’t make them!
  - Yield losses are prohibitive

How do we get bigger chips?

- Simply aggregating chips together is not a good plan
  - Chip-to-chip I/O is expensive: 0.15mm pitch for area solder
    - And you need 2+ balls per I/O
  - So we amortize the cost of this I/O
    - **Ser**ialize/**des**erialize the channel
    - Run 7x-10x higher than chip clock
    - Pay a large power cost to do this
    - Around 5 pJ/b for serial I/O

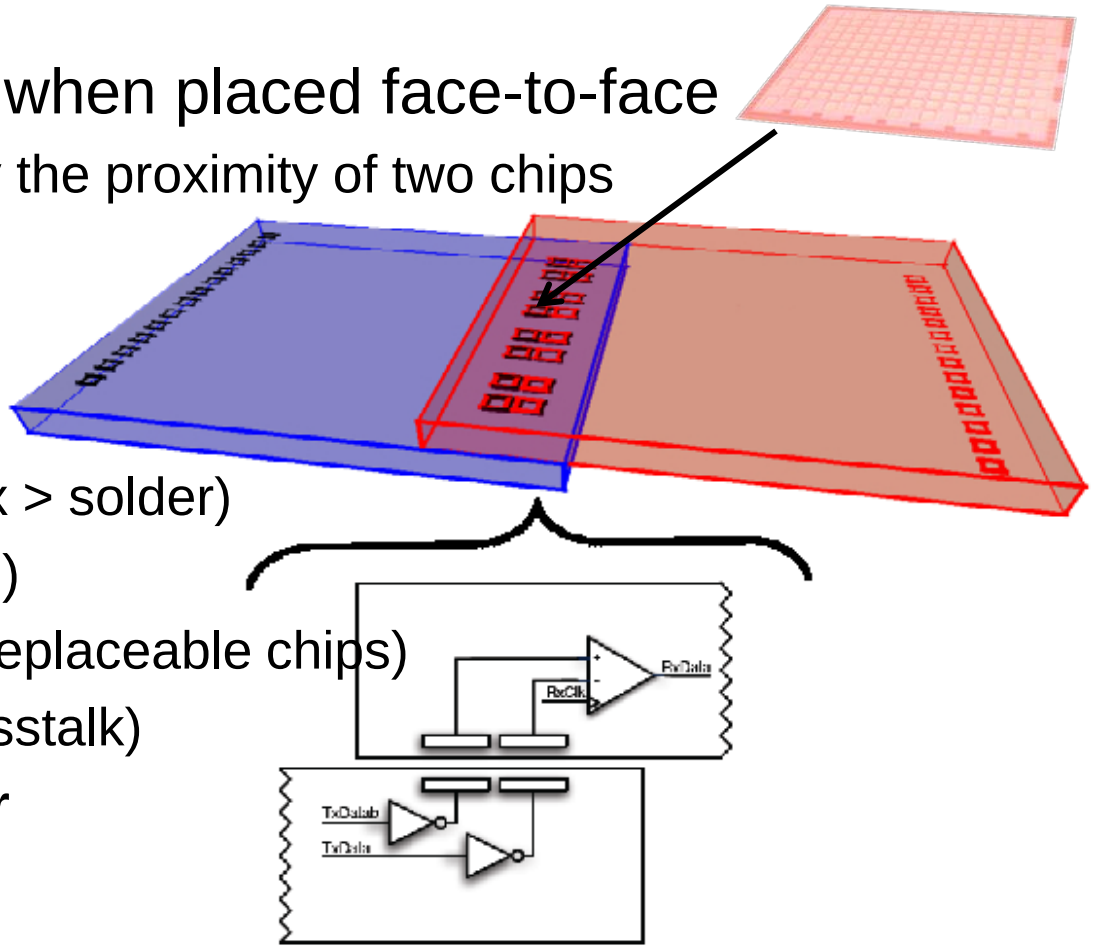


www.flipchips.com

- So sure, we'd rather have really big chips
  - (…or at least the moral equivalent of really big chips)

Proximity Communication

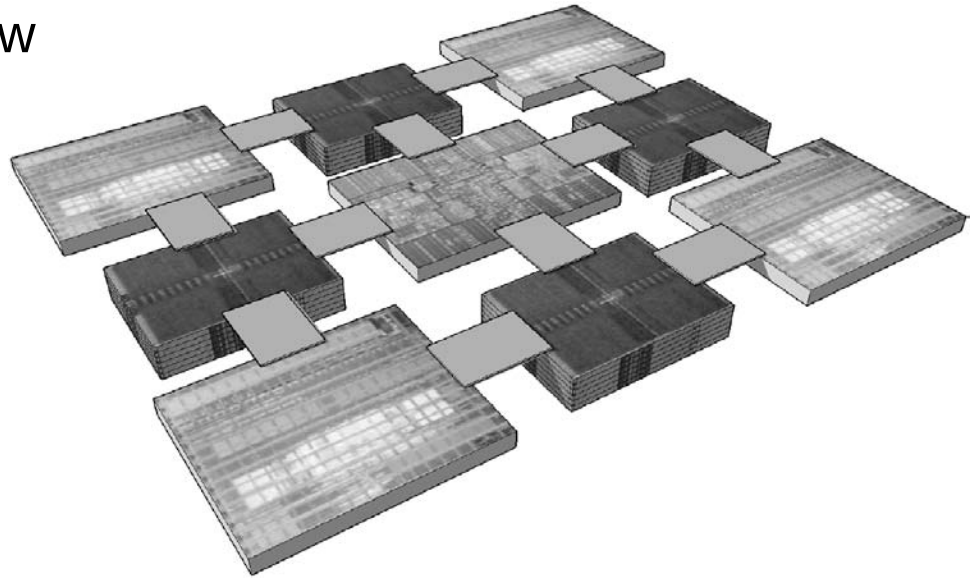
- Chips communicate when placed face-to-face
  - Capacitors formed by the proximity of two chips
  - No solder required
- This I/O provides
  - High BW density (40x > solder)
  - Low power I/O (1pJ/b)
  - High-yield systems (replaceable chips)
  - High fidelity (little crosstalk)
  - **Inter-chip I/O similar to on-chip wires!**



Proximity Communication

Why this matters

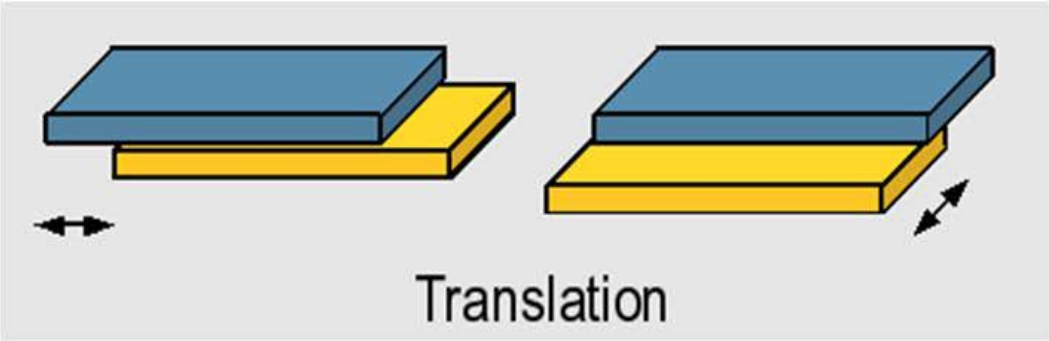
- Gives chip-to-chip I/O that looks like on-chip RC wires
  - Density, power, and latency
- Construct inexpensive big (virtual) “chips”
  - More transistors per “chip”
  - Generations of Moore’s Law scaling without buying new technologies
  - Break reticle limit
  - Mix and match processes
- A “obvious” first step
  - CPU + stacks of DRAM

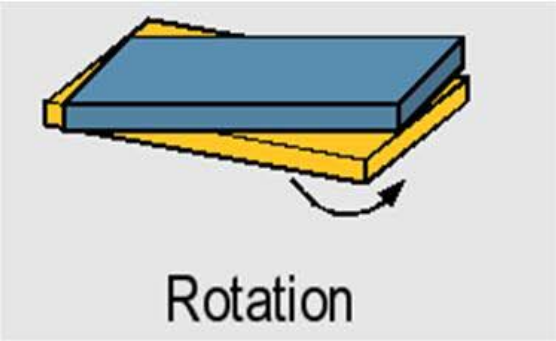


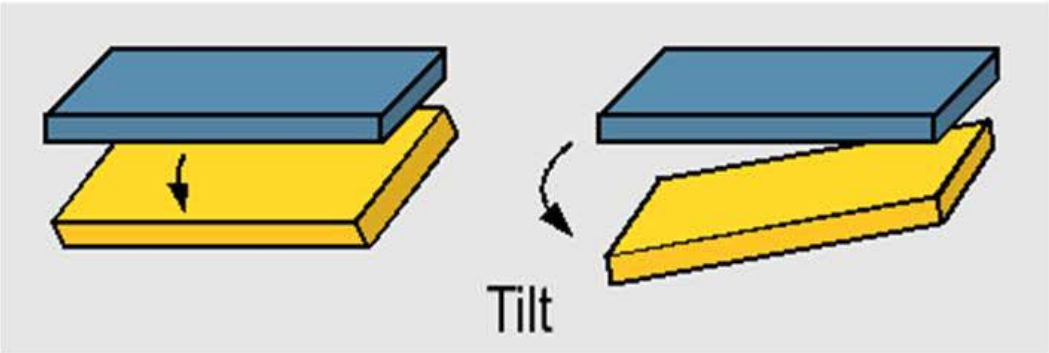
Proximity Communication

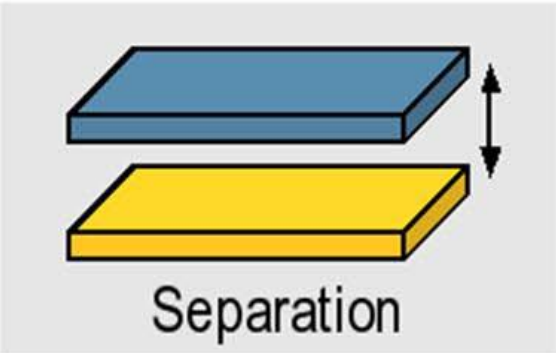
What makes this hard

- Good chip-to-chip alignment is critical
  - At assembly and under dynamic loads (i.e. thermal expansion)





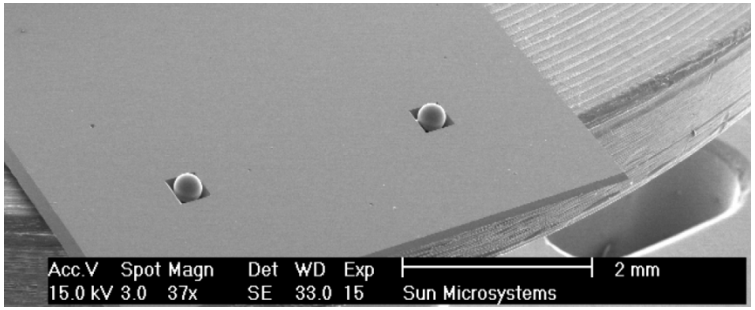


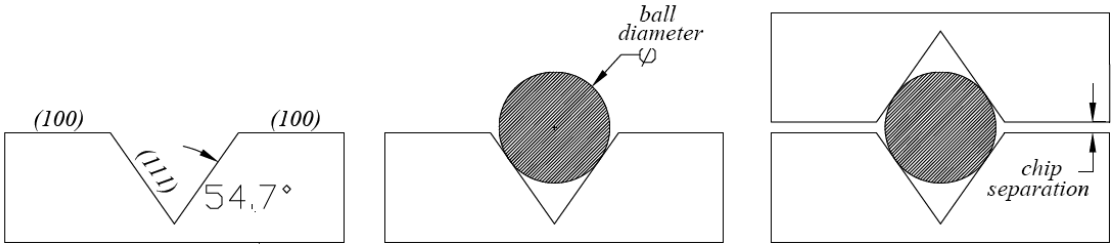


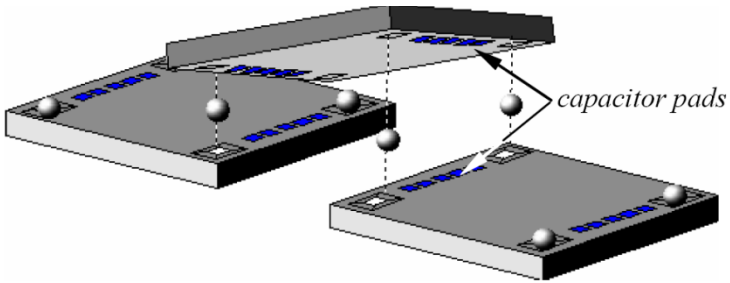
Proximity Communication

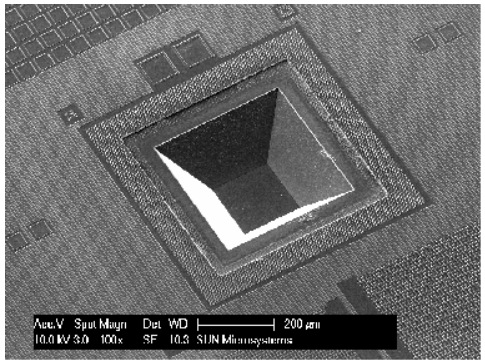
Packaging technologies can get us close ….

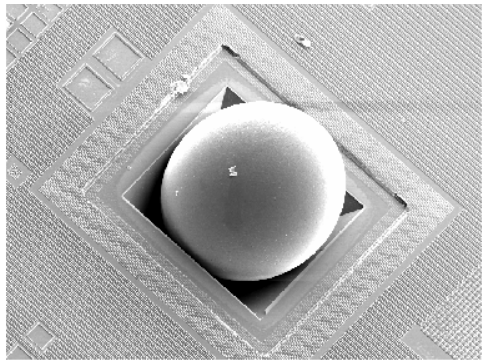
- Can align chips to within 1um accuracy
  - Silicon etching gives inverted pyramid “pits” in substrate
  - Use a sapphire sphere as a “key” to lock two chips
  - An inexpensive, high-volume-manufacturing-compatible flow







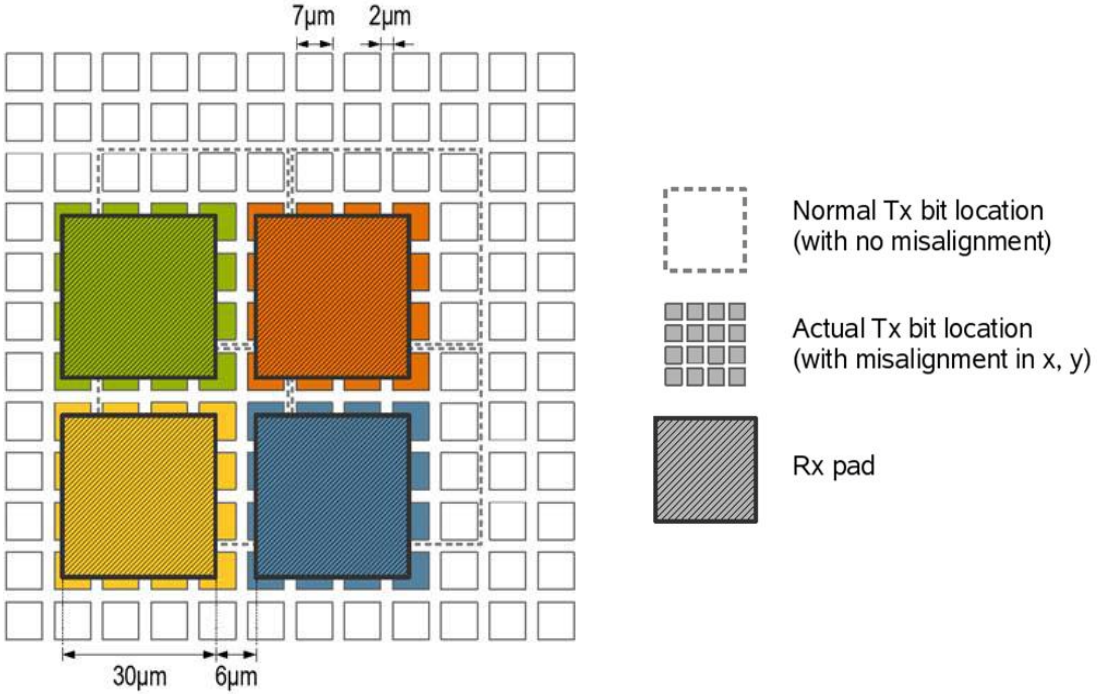




Proximity Communication

… circuits can get us closer

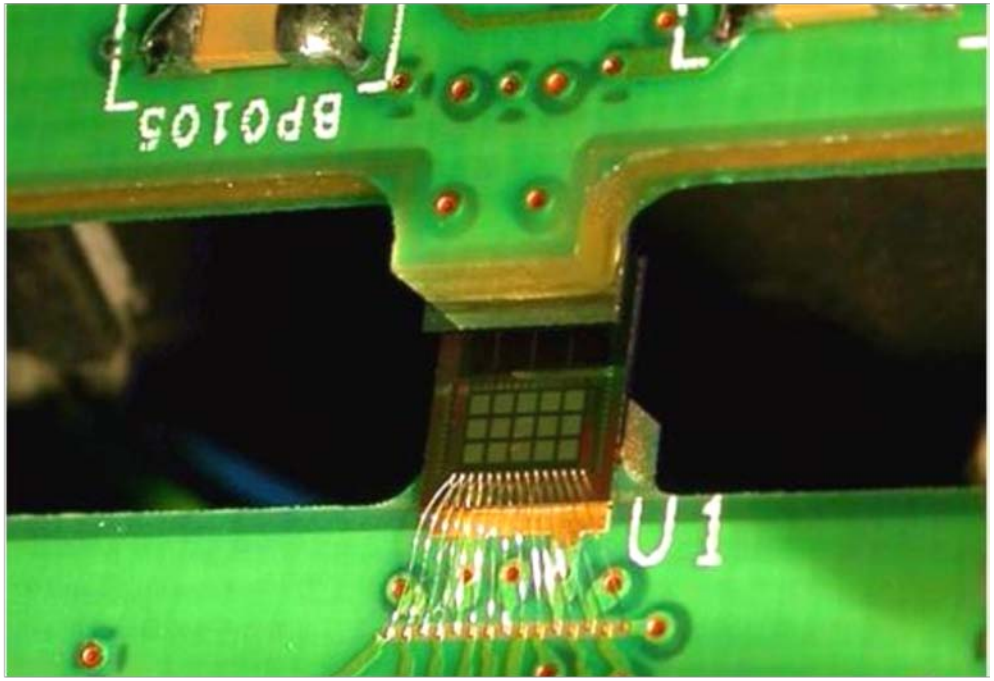
- Steer the data appropriately to overcome misalignment
  - Calculate misalignment by using side capacitors
  - Lots of muxes, but we have efficient circuit techniques to help

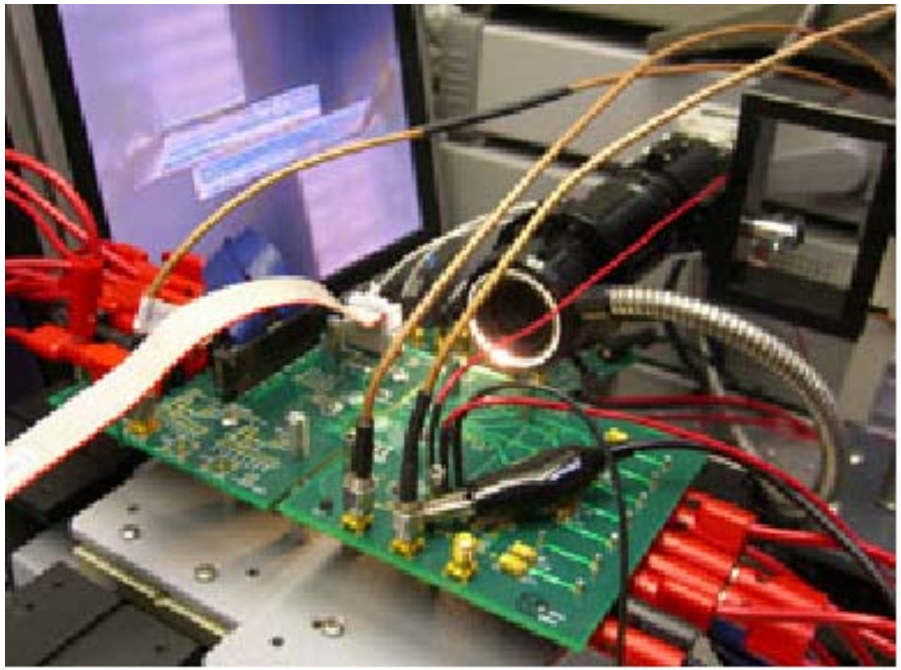


Proximity Communication

Testing chips

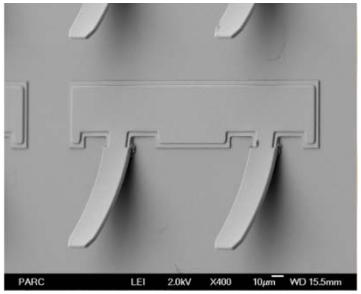
- Intentionally distort alignment during operation
  - One chip is fixed, other is on a 3D positioner
  - Move 0.1μm in X, Y, Z; move 2μrad in θx, θx, and θz



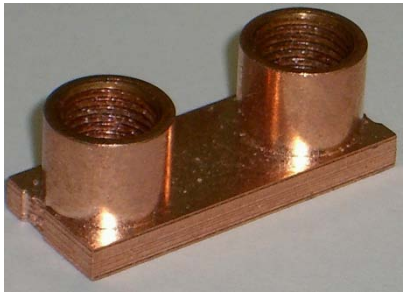


Proximity Communication

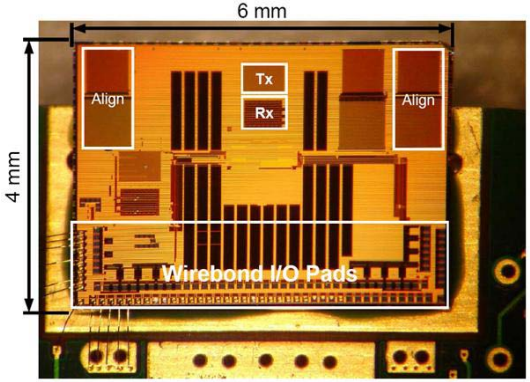
Putting it all together

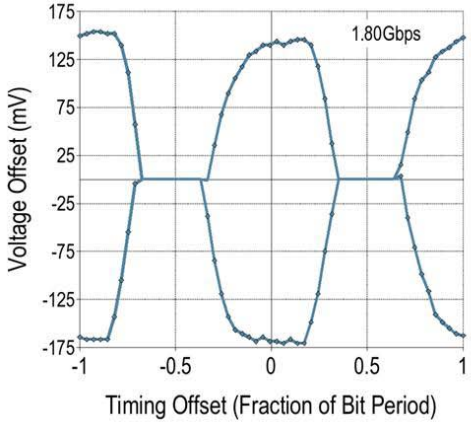


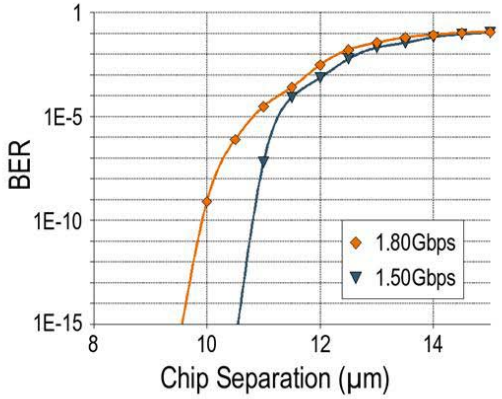
- Supporting packaging technologies
  - "Claw" non-soldered power delivery
  - Fluid cooling solutions



- Test chips and measured results (2005)
  - Today's targets: >1Tb/s/mm^2 , <1pJ/b, <<10^{-15} BER, <10 cycles



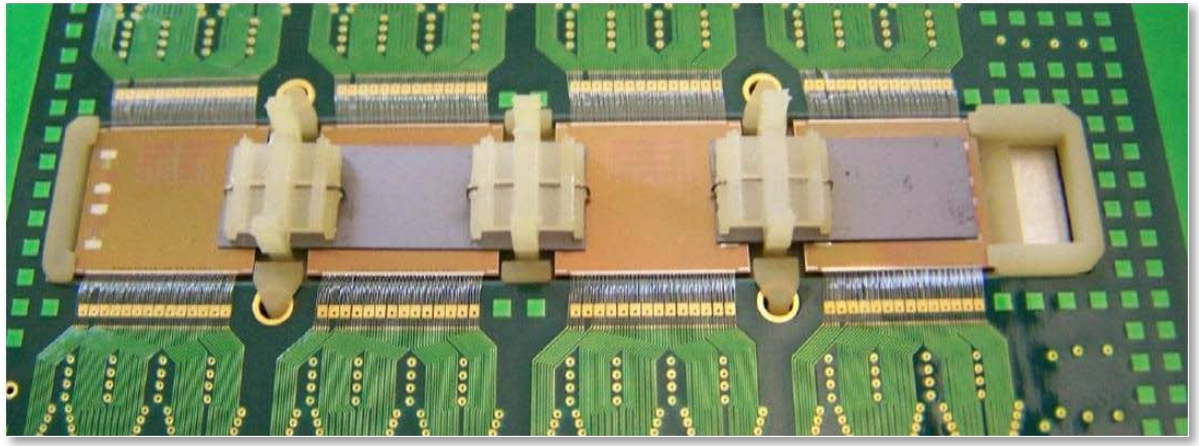


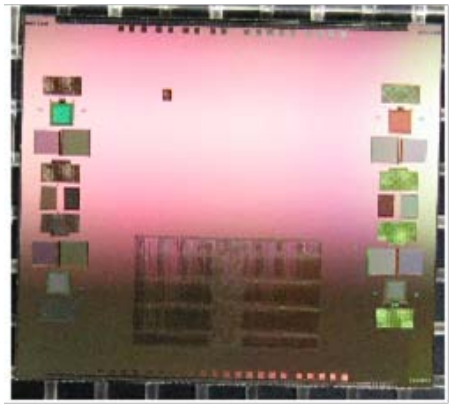


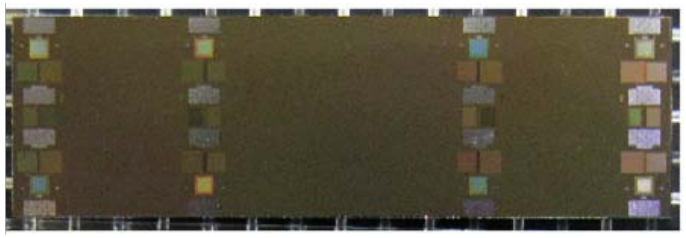
Proximity Communication

Other demonstrations

- A switch, on a 1x4 vector package, 10W/chip, 40W total
  - Each island chip is 12x14mm, package is 70x20mm
  - FR4 substrate, very inexpensive precision injection-molded plastic clips
  - I/O wirebonded directly to board (736 signals)



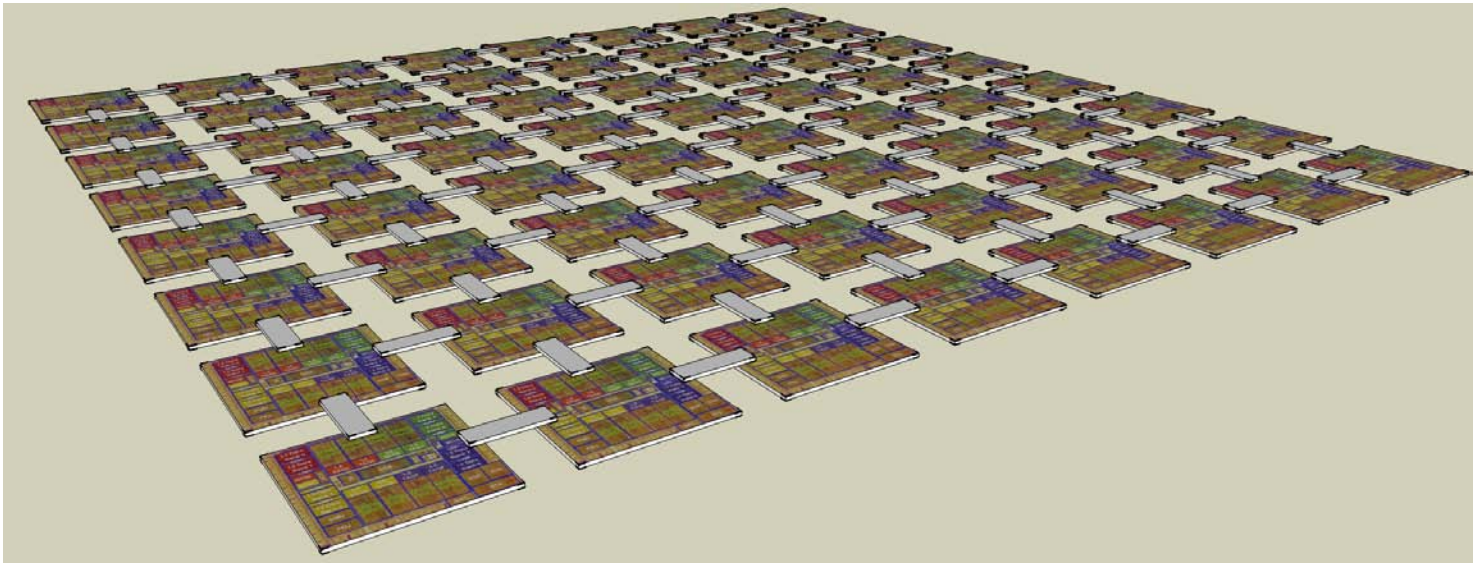




So, what are the hardware limits?

How big of a virtual chip can we actually build?

- Packaging/assembly would allow this:

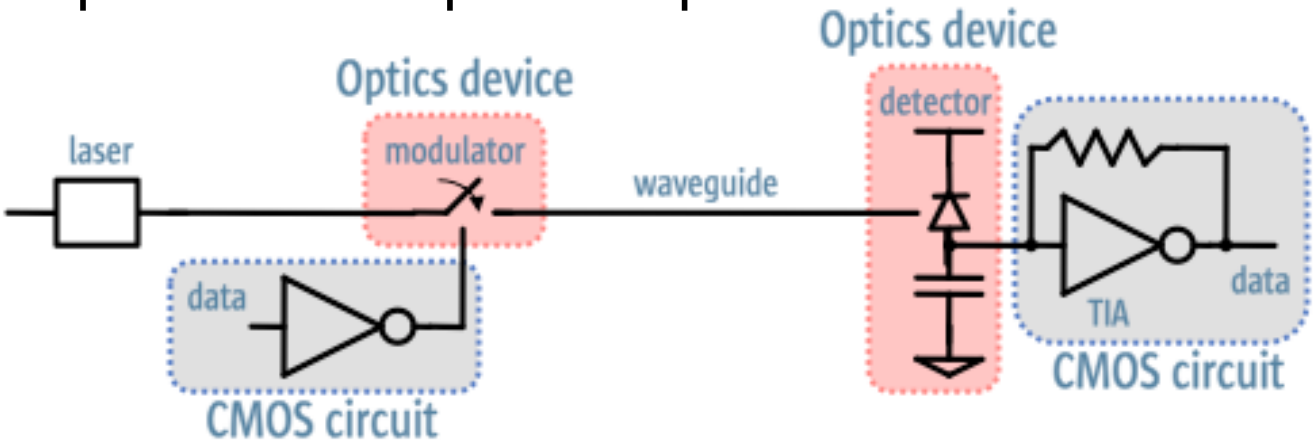


- But recall: All data communication is over RC wires
  - Wires are slow (velocity of c/20); voltage “diffuses” down wire
  - For large arrays, cross-system latency would be untenable

Can optics help?

After all, they **do** run at the speed of light…

- An optical link is quite simple



- What does this buy you?
  - Low latency
  - High bandwidth density (wavelength division multiplexing)
  - Potentially very low energy cost
  - A “clean” channel that can run very long distances

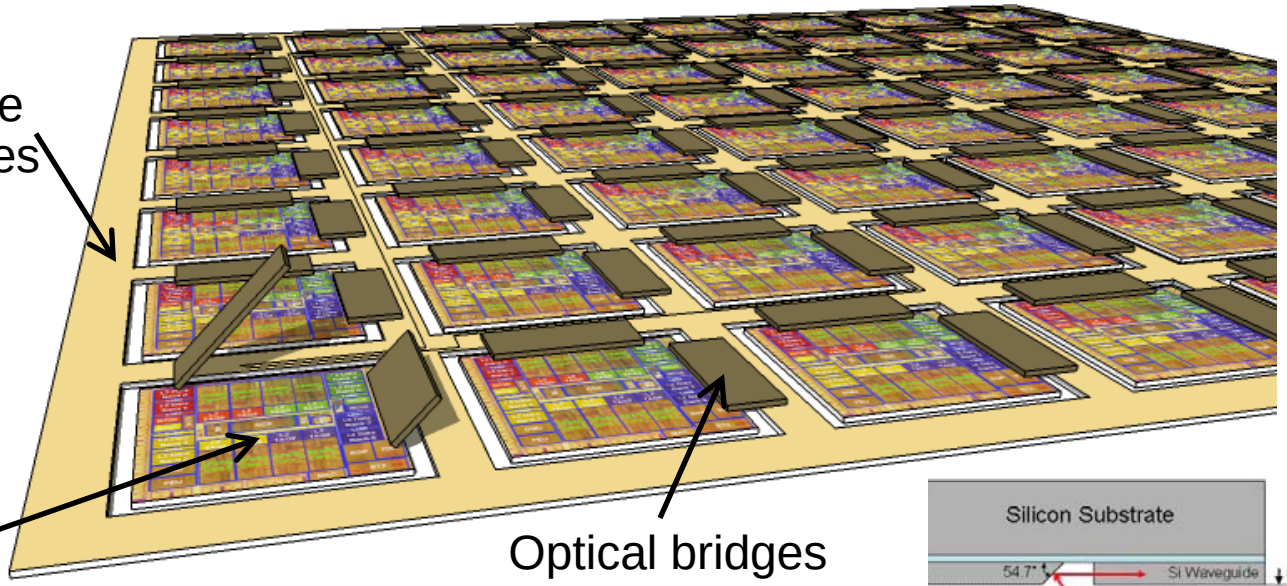
Making optics viable for VLSI systems

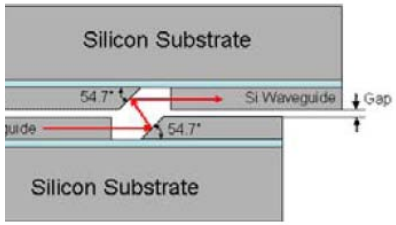
- Reduce energy: pJ/bit == mW/Gbps
- Reduce the link overhead
  - Use very small optical devices, ~10μm on a side
  - Share waveguides: 8-16x wavelength division multiplexing
  - Minimize the number of opto-electronic conversions
- Minimize any yield impact
  - Separate optical devices from CPUs
  - But stay compatible with high-volume CMOS processing

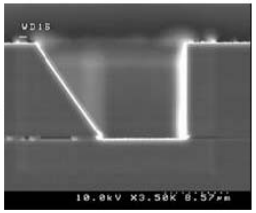
The “macrochip”: a technical vision

An attention-focuser to promote device, circuit, and architectural studies

- Silicon lattice carrying CPUs/DRAMs
  - With a fully-connected pt-to-pt network of optical waveguides
  - Bridges with optical PxC performs opto-electric conversion



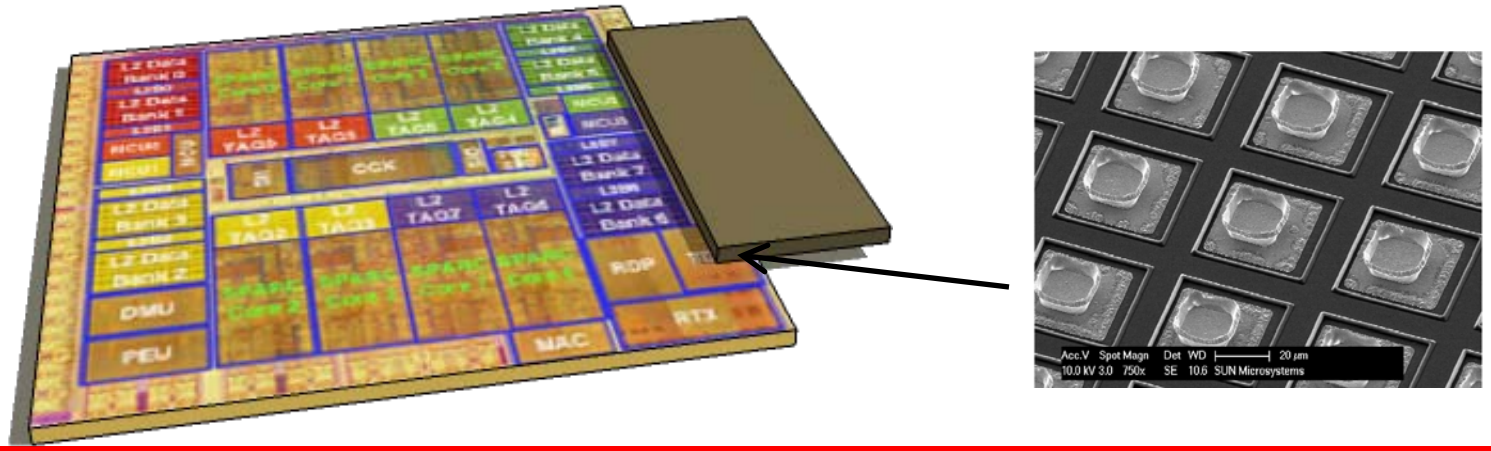




Hybrid bonded optical devices

A way to separate CMOS logic chip and optical devices

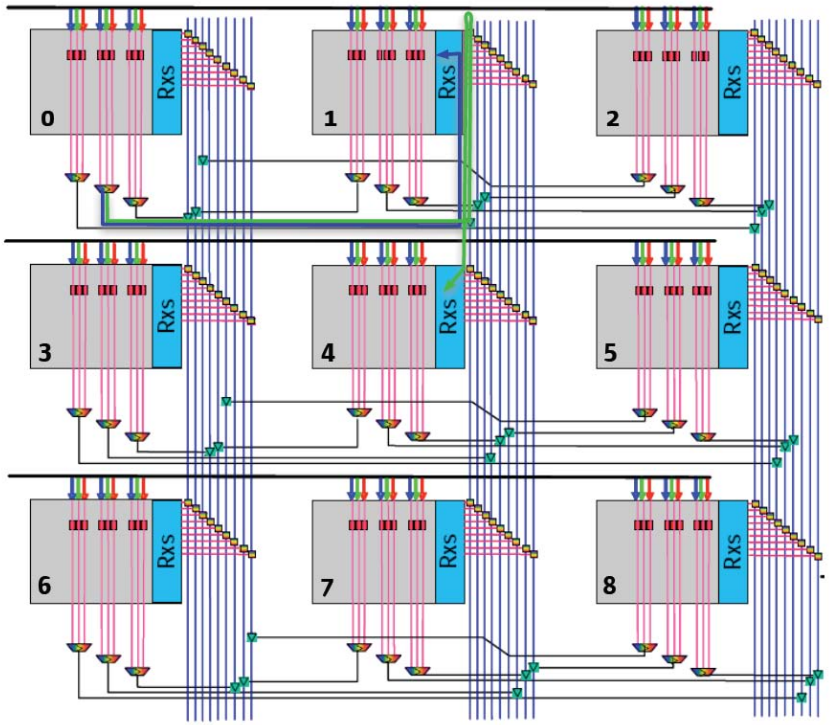
- Put optical devices on a separate “bridge” chip
  - Avoid complicating the CPU
  - Allows SOI for optics, bulk CMOS for CPU
- Bridge is passive; all active circuits live on CPU
  - Can be connected using PxC or dense microsolder
  - Waveguides on bridge can talk to other waveguides or fibers



The “macrochip”: a technical vision

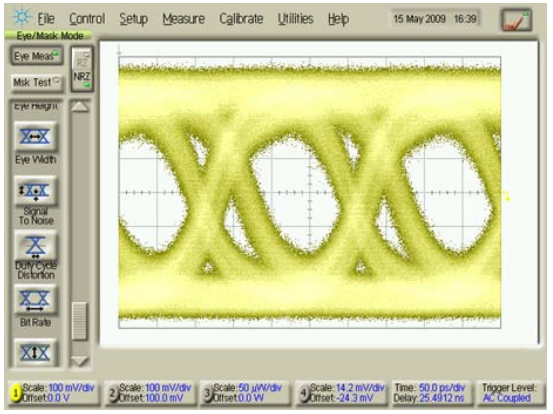
And a platform for network exploration

- Collaborative work driven by colleagues in Austin, TX
- Compare point-to-point network with other options
  - If you’ve made bandwidth enough to waste …
  - … then a “wasteful” over-provisioned network is the right choice
  - Move switching complexity from the network and into the sites
  - Minimizes optical device component counts, and hence link loss, and hence total energy cost

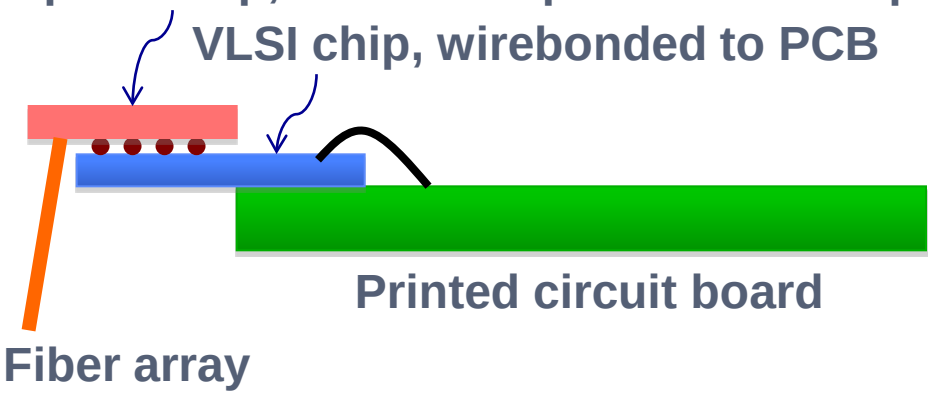


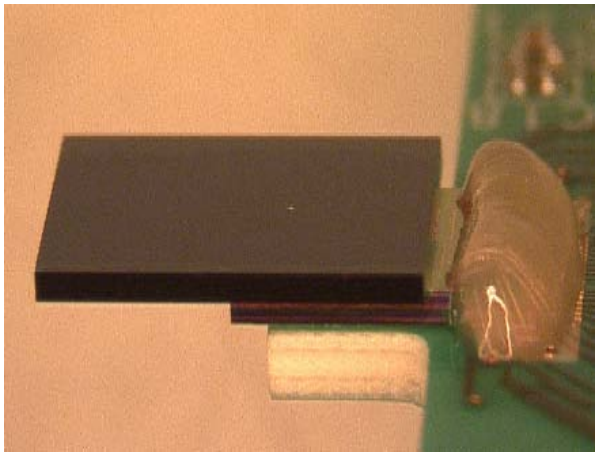
Progress thus far

In year 2 of a 5+ year research program



- Year 1 results include several “firsts” results
  - OPxC, record low-loss waveguides and splitters, transceivers
  - 1.2 pJ/bit for transmit + receive at 5 Gbps
  - 90nm CMOS VLSI chip and an SOI optics chip

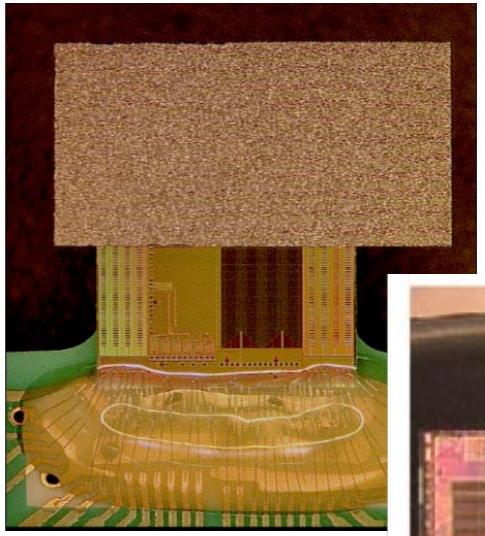


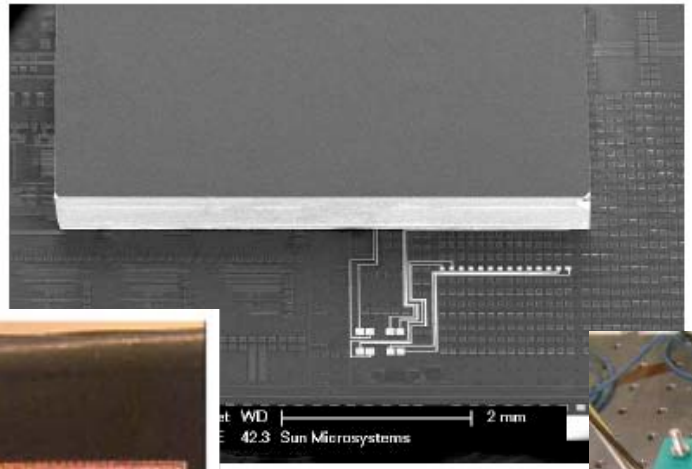


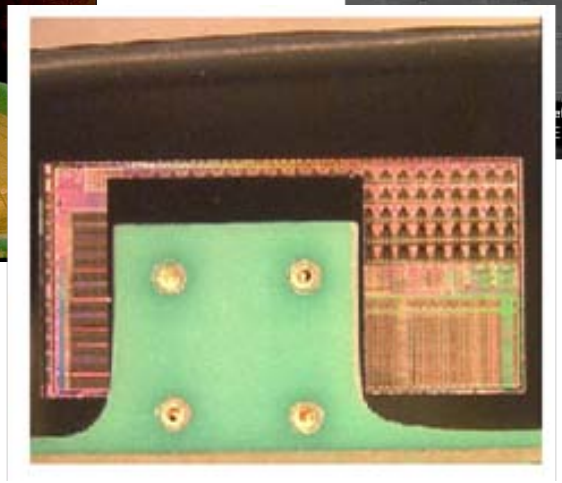
Progress thus far

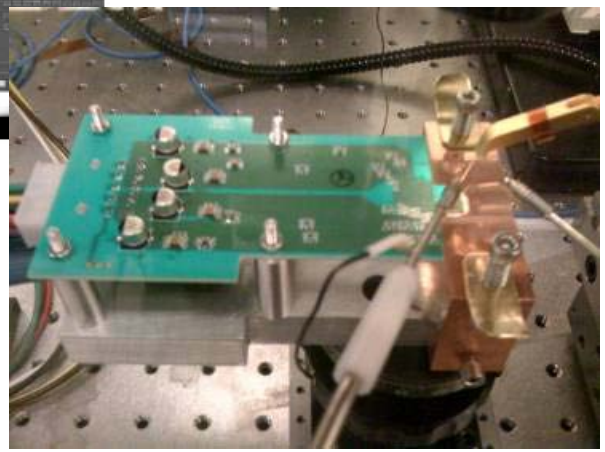
Challenges ahead

- Future challenges are legion!
  - Devices, circuits, packaging, testing, system analysis…









Final thoughts

What can we do for you?

- New hammers for an architect's tool box
  - Create the "big chip" you know you want
    - PxC for modest sizes
    - Optics for larger systems
- What to do with these tools is up to you
  - We'd love to hear about your ideas

Questions?