

System-on-Chip: An Architecture Perspective

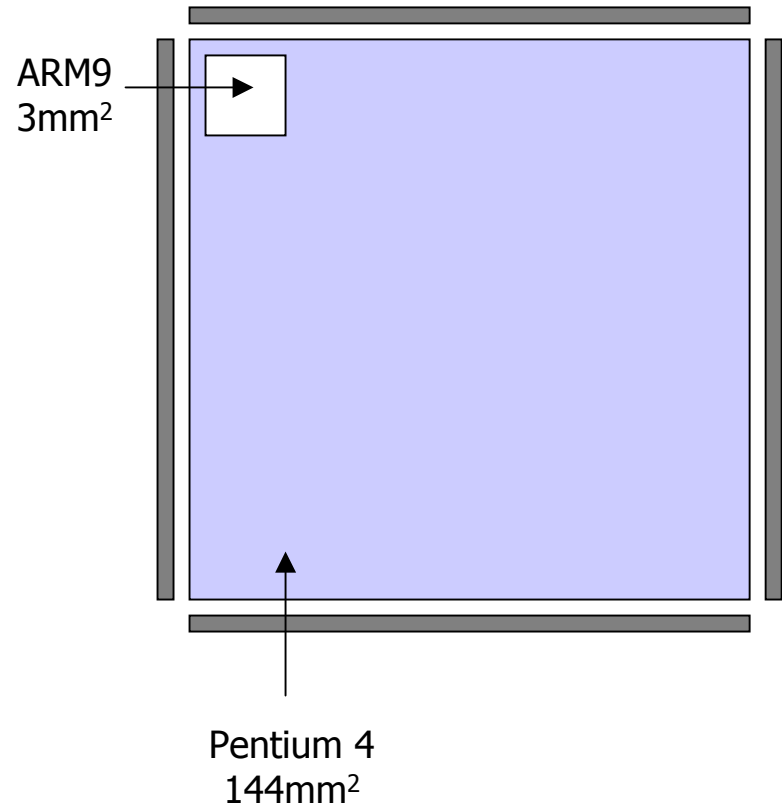
CSE 291E / ECE 260C
Spring 2002



System-On-Chip

(or what am I going to do with a billion transistors?)

- At $0.13\mu\text{m}$
 - ARM9 = 3mm^2
 - Pentium 4 = 144mm^2
 $\sim 50 \times \text{ARM9}$
- By 2007
 - $0.02\mu\text{m}$ (20-nm)
 - 20GHz
 - 1 billion transistors !
 - $\sim 2500 \times \text{ARM9}$

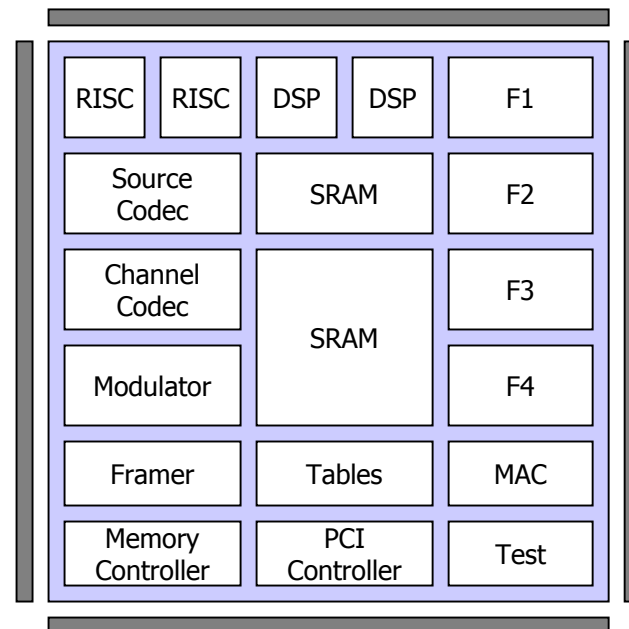


What am I going to do with a billion transistors?

- View 1: More logic to make sequential execution faster
 - More & more logic for predictive & speculative execution, “on-the-fly” optimizations & code morphing ...
- View 2: More memory on-chip
 - Currently 70% logic, 30% cache/memory
 - Expect 30% logic, 70% cache/memory

What am I going to do with a billion transistors?

- View 3: Chassis to boards to IC's
 - What used to be in a chassis become on a board
 - What used to be on a board become on a single IC
 - Been doing this for decades
 - Most common definition of "System-on-Chip"

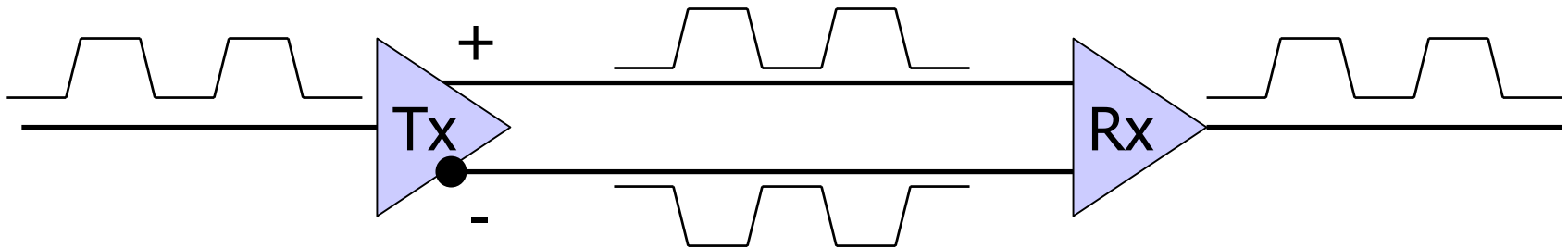


What about I/O to a billion transistor IC?

- Problem
 - # transistors 2x every 18 mos (exponential)
 - Pin count constant, linear at best
- Solution
 - More pins: 1000+ pins possible
 - **SerDes (Serializer/Deserializer) to the rescue!**

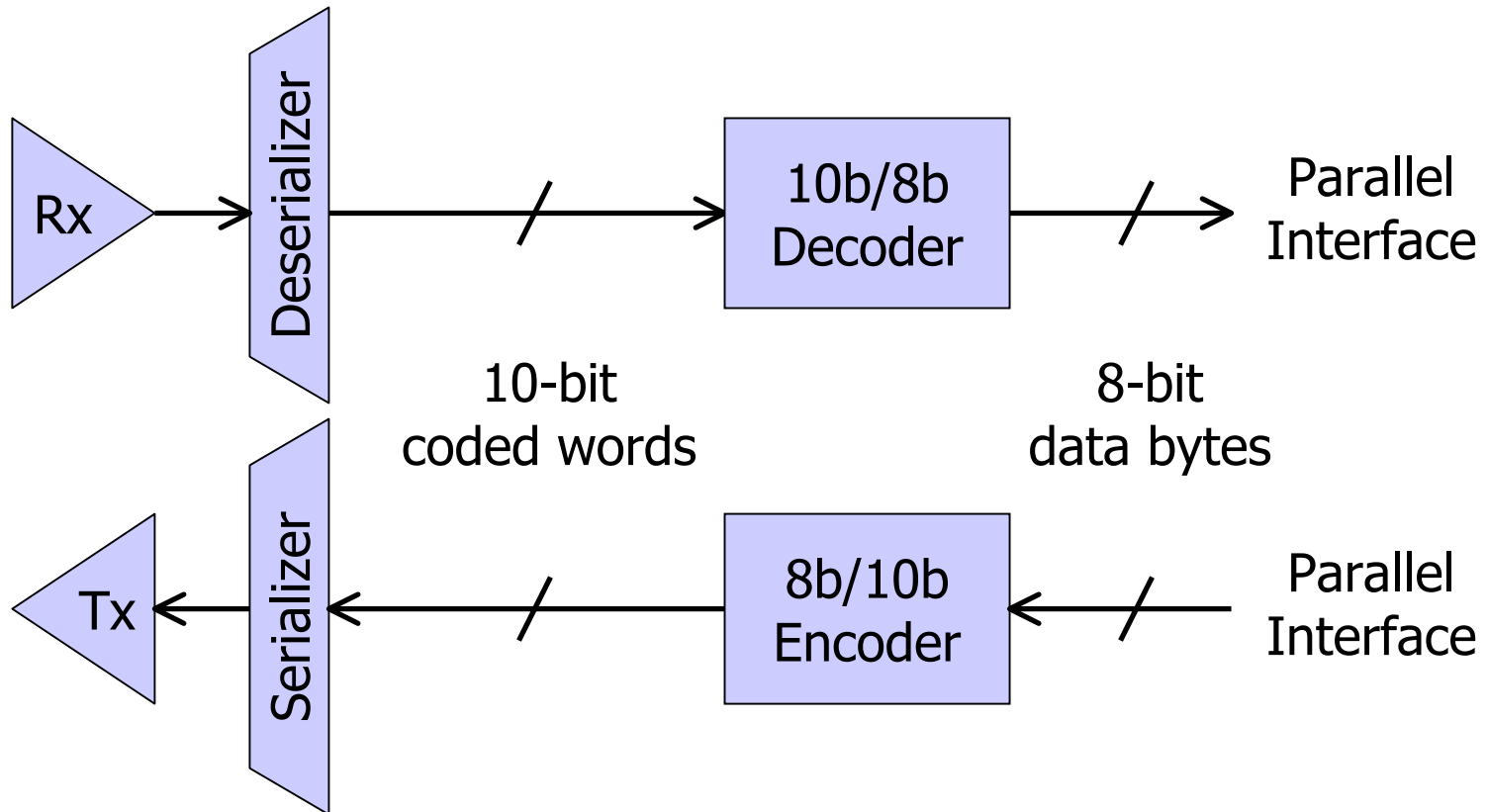
SerDes to the Rescue!

- Based on **differential signaling**
- Can pump 4Gbps per pair of pins
- Internally serialize/deserialize. 8:1 & 16:1 currently common
- Roadmap: 10Gbps per pair of pins, 32:1 in a few years



8b/10b Encoding

- Ensures bit transitions within each 10b word
- Clock can be recovered from the data

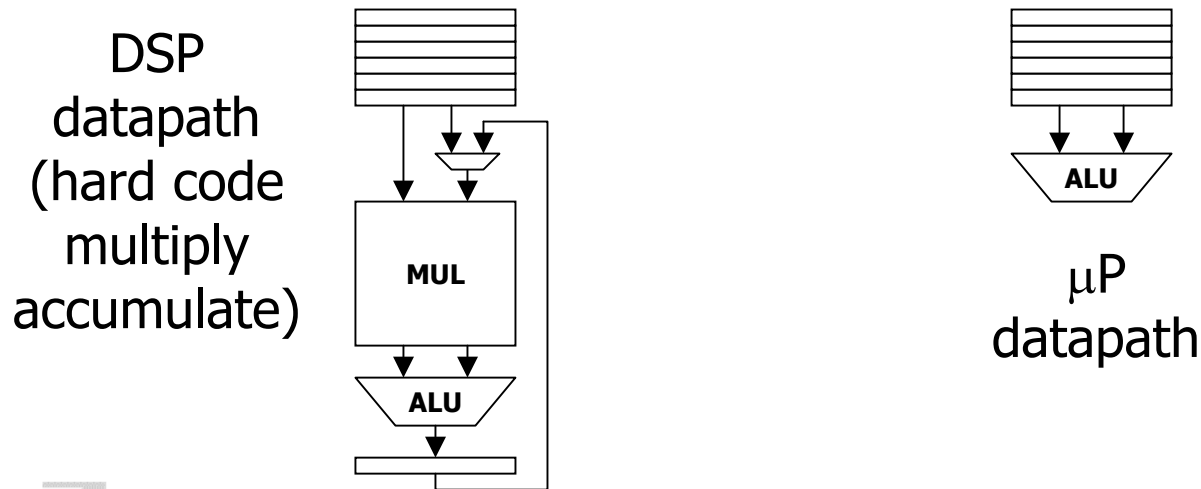


DSP vs. μ P Cores

- Many DSP & digital communications algorithms have this “sum-of-product” computation. e.g. FFT computation

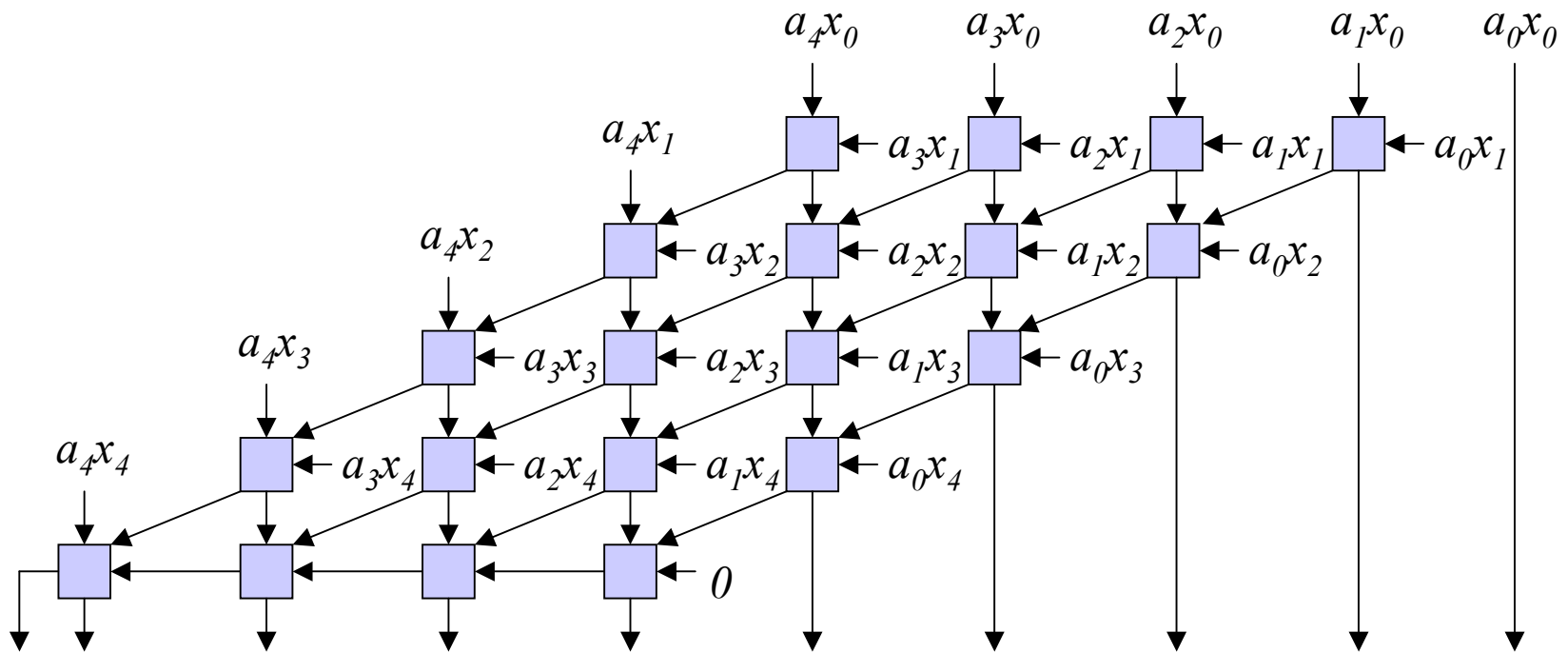
$$H_n = \sum_{k=0}^{N-1} W^{nk} h_k$$

- DSP datapath vs. μ P datapath



DSP vs. μ P Cores

- Try to do multiply-accumulate with just an adder datapath
- Requires many instructions!
- Think of hardware multiply-accumulator datapath as 1st “kernel-accelerator” with broad appeal

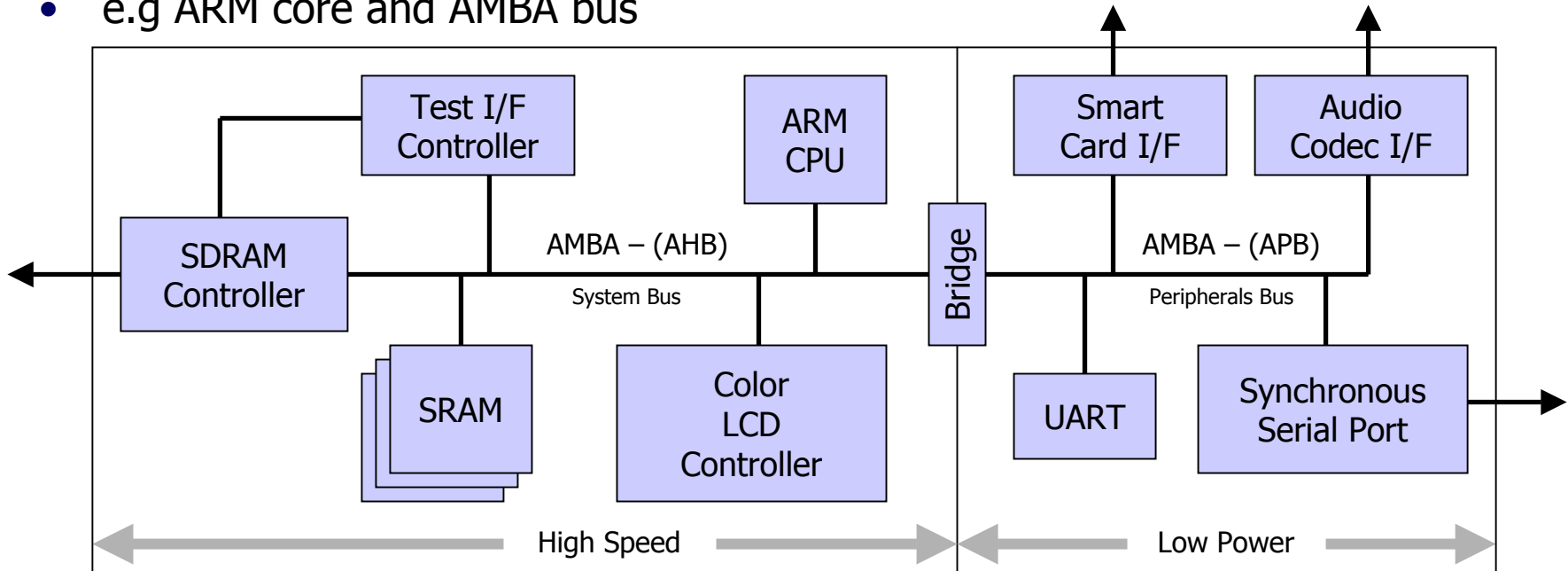


DSP vs. μ P Cores

- Many small differences have been touted between DSP's (e.g. TI-DSP) & μ P's (e.g. ARM)
- Configurable processors (e.g. Tensilica) make multiply-accumulator datapath an "option"
 - With multi-acc = DSP?
 - Without multi-acc = μ P?
 - Distinction between DSP & μ P blurred?
- At least one significant issue
 - Multiply-accumulator delay $\gg$ adder delay
 - Pipelining design trade-offs may be different

SoC Design

- e.g ARM core and AMBA bus



- Analogous to Pentium, Windows/Linux, PCI, memory, graphics card, sound card, Ethernet card ...
- Need standardized bus & driver models (e.g. AMBA bus instead of PCI)
- Need stripped down OS (e.g. WindRiver instead of Windows/Linux) to host software and “drivers” (communication channel) to hardware

SoC Design

- Example on-chip bus interconnects
 - ARM's AMBA bus
 - IBM's Core Connect
 - Virtual Socket Interface Alliance group
 - Open Connect Protocol group
- Example processor cores
 - ARM
 - MIPS
 - PowerPC

Standardization is key to creation of SoC value-chain

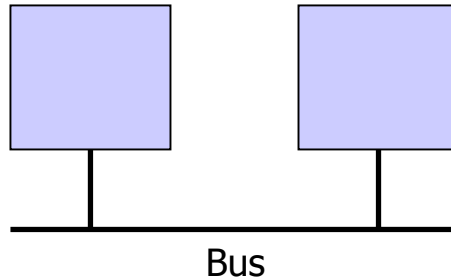
- 3rd embedded software possible
- 3rd party ASIC cores possible (often called IP cores, IP = Intellectual Property, not Internet Protocol)
 - Plug-able to a standardized interconnect
 - Interoperable with embedded software on processors via well-defined drivers ...
- Towards “plug-n-play”
- Effects similar to the Intel-PC value-chain, NPF (Network Processing Forum) based switch/router value-chain ...

See Tality for examples of 3rd party cores

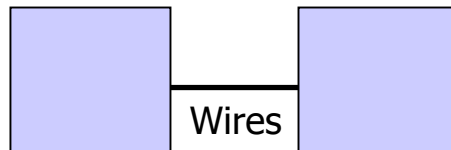
- 10/100 PHY
- 10/100 MAC
- 10G XAUI interface
- 8b/10b encoders
- HomePNA PHY
- 1.25G/2.5G SerDes
- DOCSIS 1.1 PHY
- 802.11a OFDM modem
- 802.11a/b dual-mode MAC
- Bluetooth baseband
- Bluetooth PHY
- USB 2.0 PHY
- SDRAM memory controller
- GSM/GPRS reference platform
- GSM/GPRS data module
- 10bit 50MHz pipelined ADC
- 10bit 300MHz DAC
- Viterbi decoder
- Reed-Solomon decoder
- ...

Current SoC Interconnect Approaches

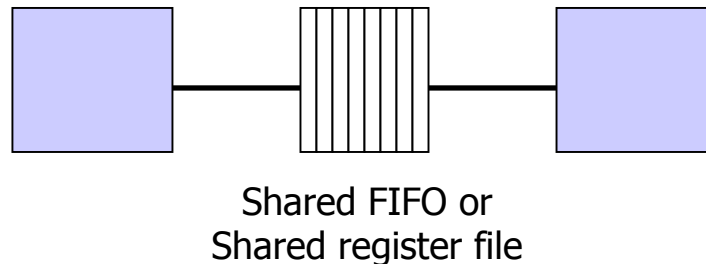
- Standardized busses



- Point-to-point wires



- Shared FIFO's or shared register files

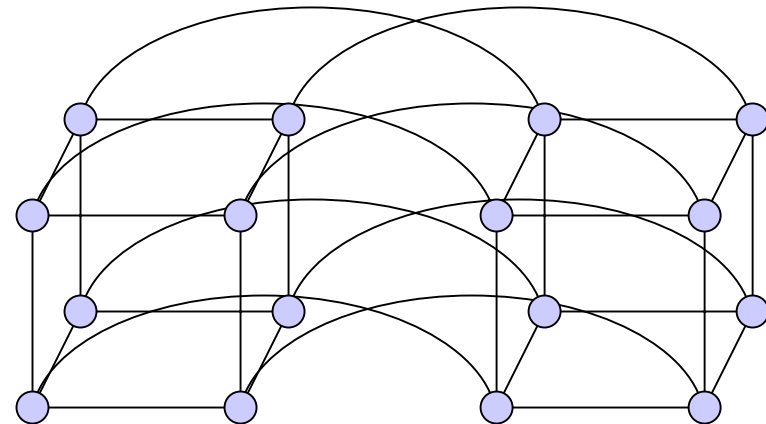
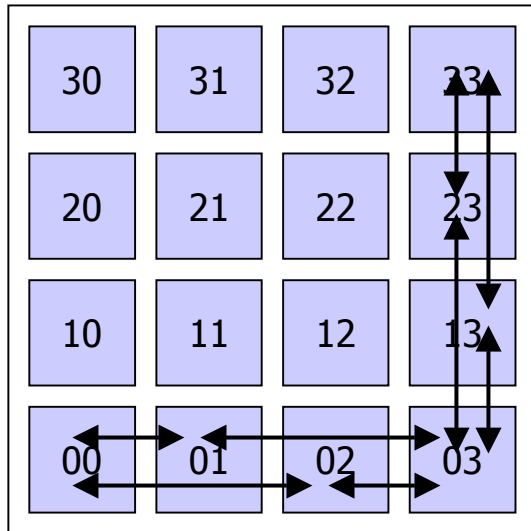


Need Scalable On-Chip Switch Architecture

- Shared busses & local point-to-point connections may not be sufficient as we move into the billion transistor era
- Too many cores to interconnect
 - Complex design & verification effort
 - Lots & lots of wires to route
- Need systematic scalable network architecture
 - Analogous to crossbar-based packet switches at the board level, again with standardized protocols & interfaces needed for a value-chain
 - But on-chip & board-level design trade-offs may be different

Example On-Chip Network Architecture: Hypercube-on-Chip

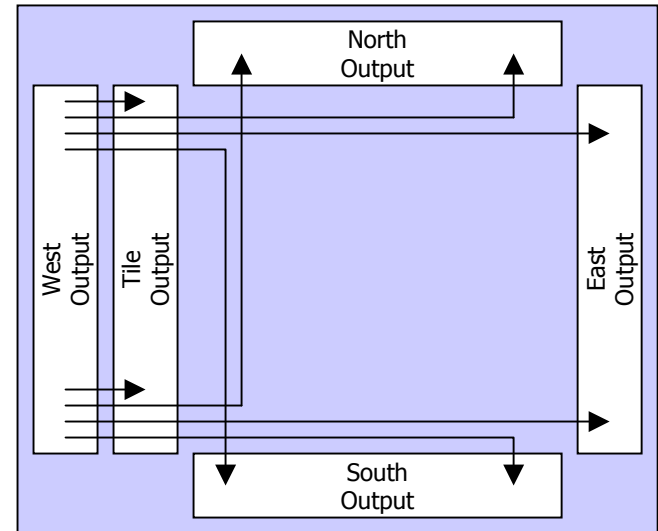
- Stanford's proposal (B. Dally)



- Use Hypercube architecture used in Cray's T3, SGI's Origin ...
- $O(\log N)$ #hops & outgoing links per tile

Example On-Chip Network Architecture: Hypercube-on-Chip

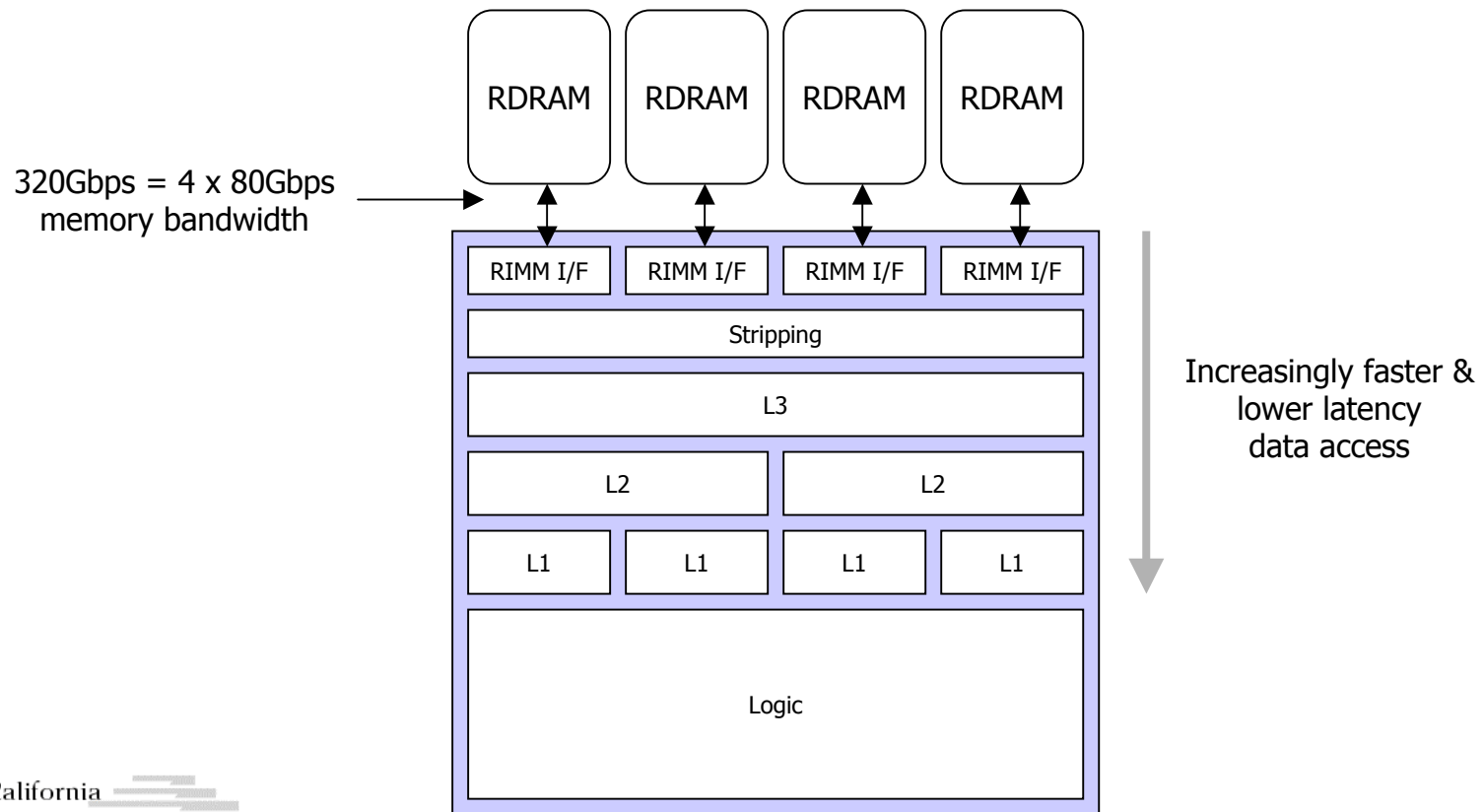
- Claims
 - 6.6% area overhead using layer 6 & 7 metal that route over tiles (cores)
 - Regularized routing enable faster data rates over wires than global clock (e.g. using double/quad data rates or serial differential signaling approaches)
- Alternative architectures should be investigated
 - e.g. Crossbar, mesh, quad trees, token rings ...



On-Chip Memory Hierarchy

Another Major Issue

- Need to move lots of data in-and-out of chip!
- Many open challenges to on-chip memory hierarchy design
- Cache coherency across multiple processing nodes on-chip?



Cross-Fertilization of SoC and CMP Research

- Many challenges and ideas in the Chip-Multi-Processor (CMP) research are applicable to System-on-Chip (SoC) research
- Many challenges and ideas in System-on-Chip (SoC) research are applicable to Chip-Multi-Processor (CMP) research
 - Digital communications, network processing, and video processing guys have been building chip multiprocessors for years

Questions?

University of California
San Diego

The logo of the University of California San Diego, featuring a stylized gray graphic of a building or landscape to the right of the text.