

Versatile Tiled-Processor Architectures The Raw Approach

Rodric M. Rabbah
with Ian Bratt, Krste Asanovic,
Anant Agarwal

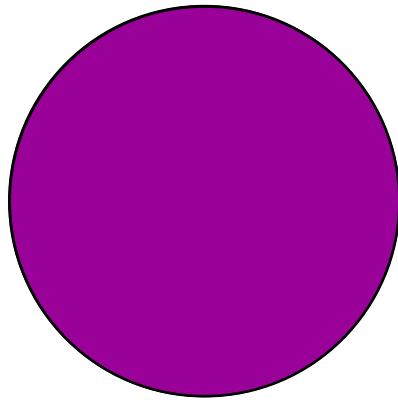


Processor Model

- Stable model for last few decades
 - Von Neumann architecture
 - Sequentially execute instructions
 - Simple abstraction
 - Easy to program

Change Is Around the Corner

- Processor performance not scaling as before
 - Wire delay and power



old view: chip looks small to a wire



chip size



distance signal can travel
in 1 cycle

new view: chip looks much bigger to a wire,
communication is expensive even on chip!

- How to effectively use transistors?

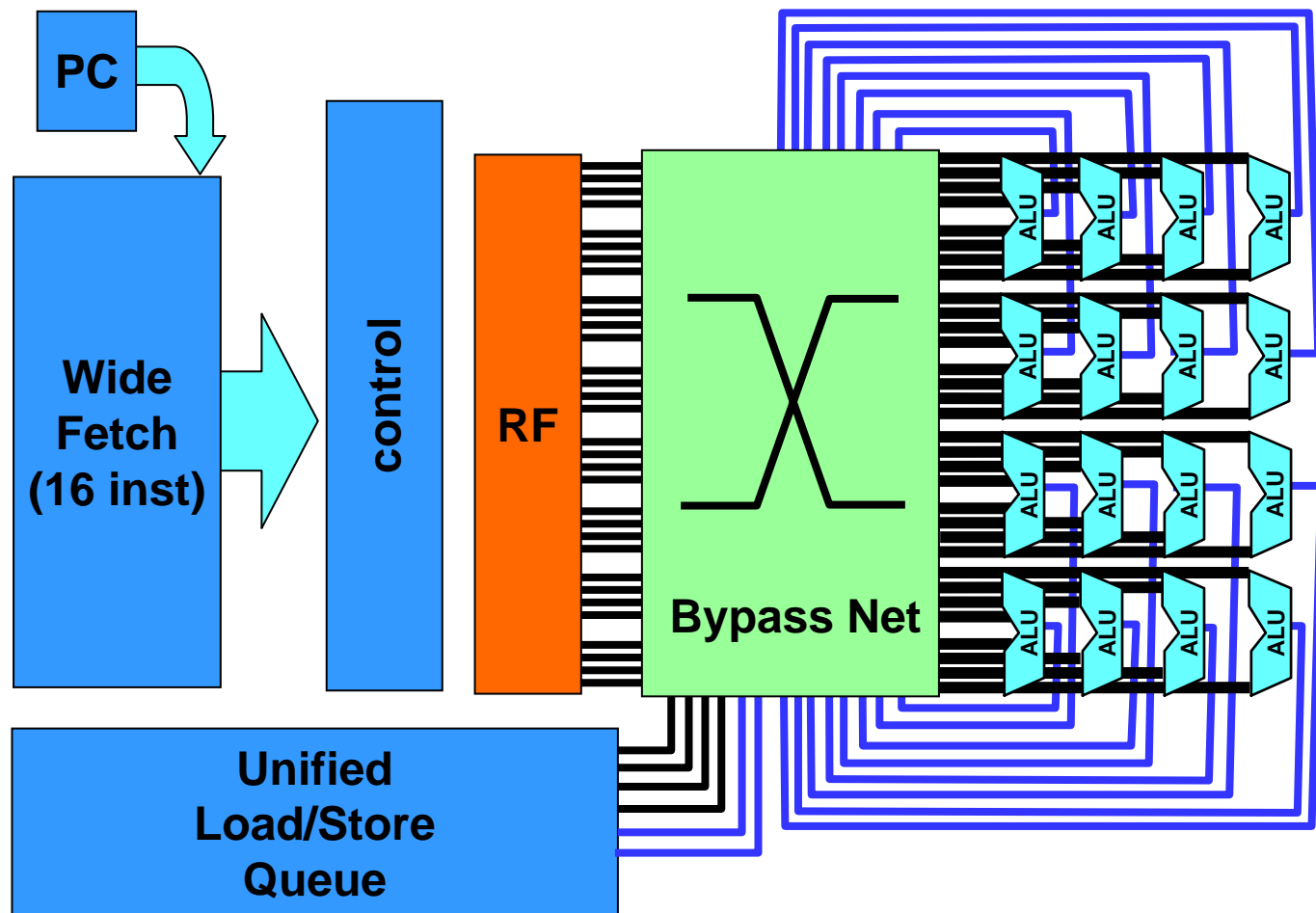
Spatially-Aware Architectures

- Many forward looking architectures are addressing the physical challenges
 - MIT Raw processor
 - MIT Scale processor
 - Stanford Imagine processor
 - Stanford Smart Memories processor
 - UC David Synchronscalar
 - UT Austin TRIPS processor
 - Wisconsin ILDP architecture
 - The original IBM BlueGene processor

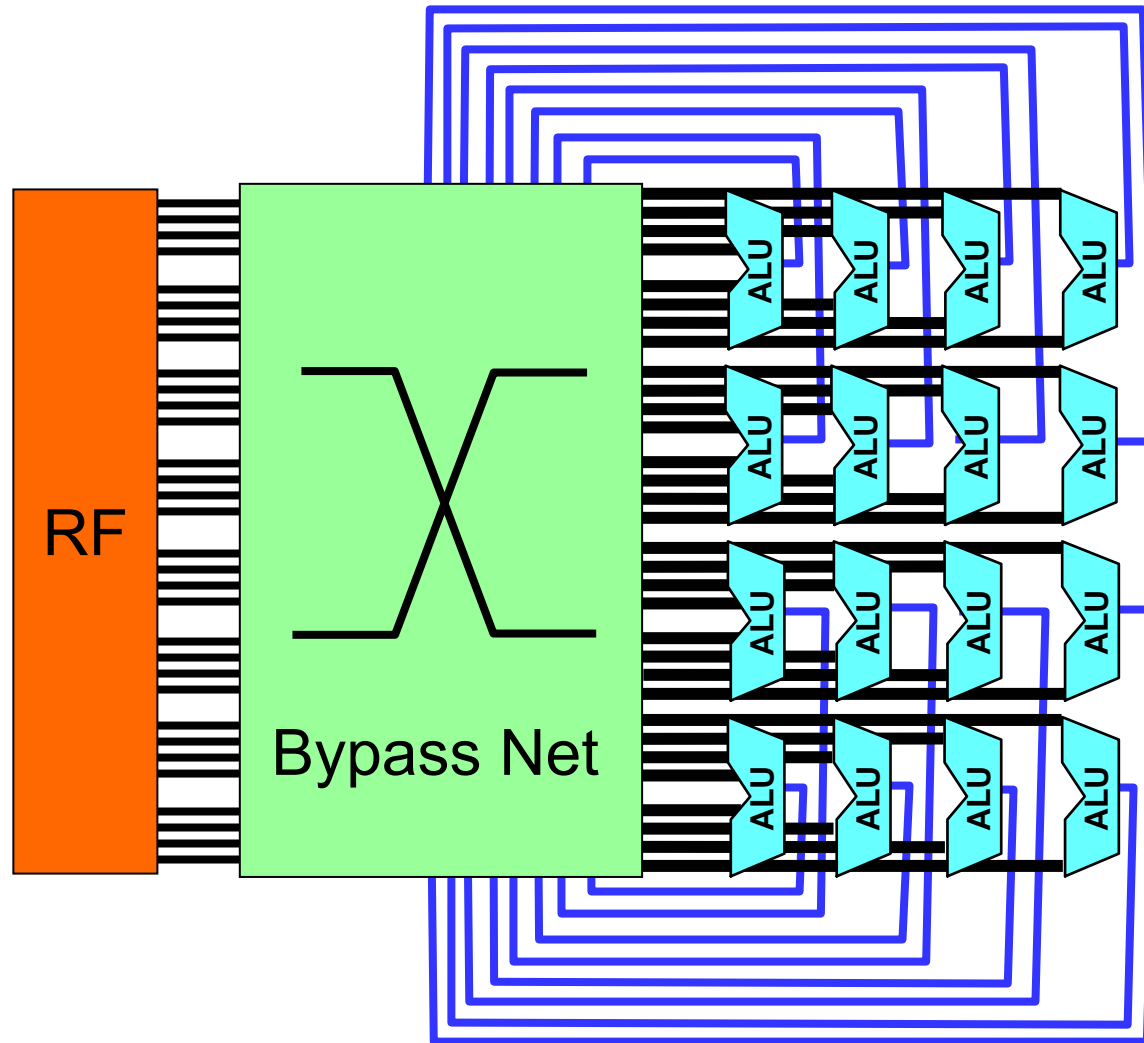
Problems with Monolithic Designs

- Super-wide general purpose processors are no longer practical

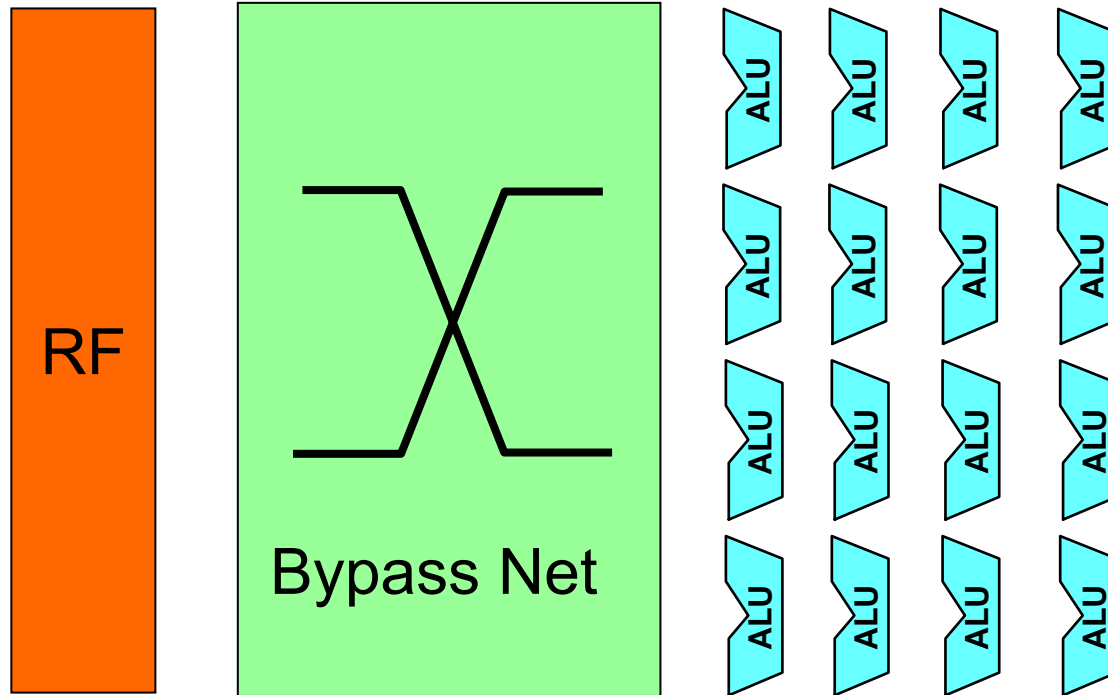
- Centralized control with global operand routing
- Area, power, and frequency concerns



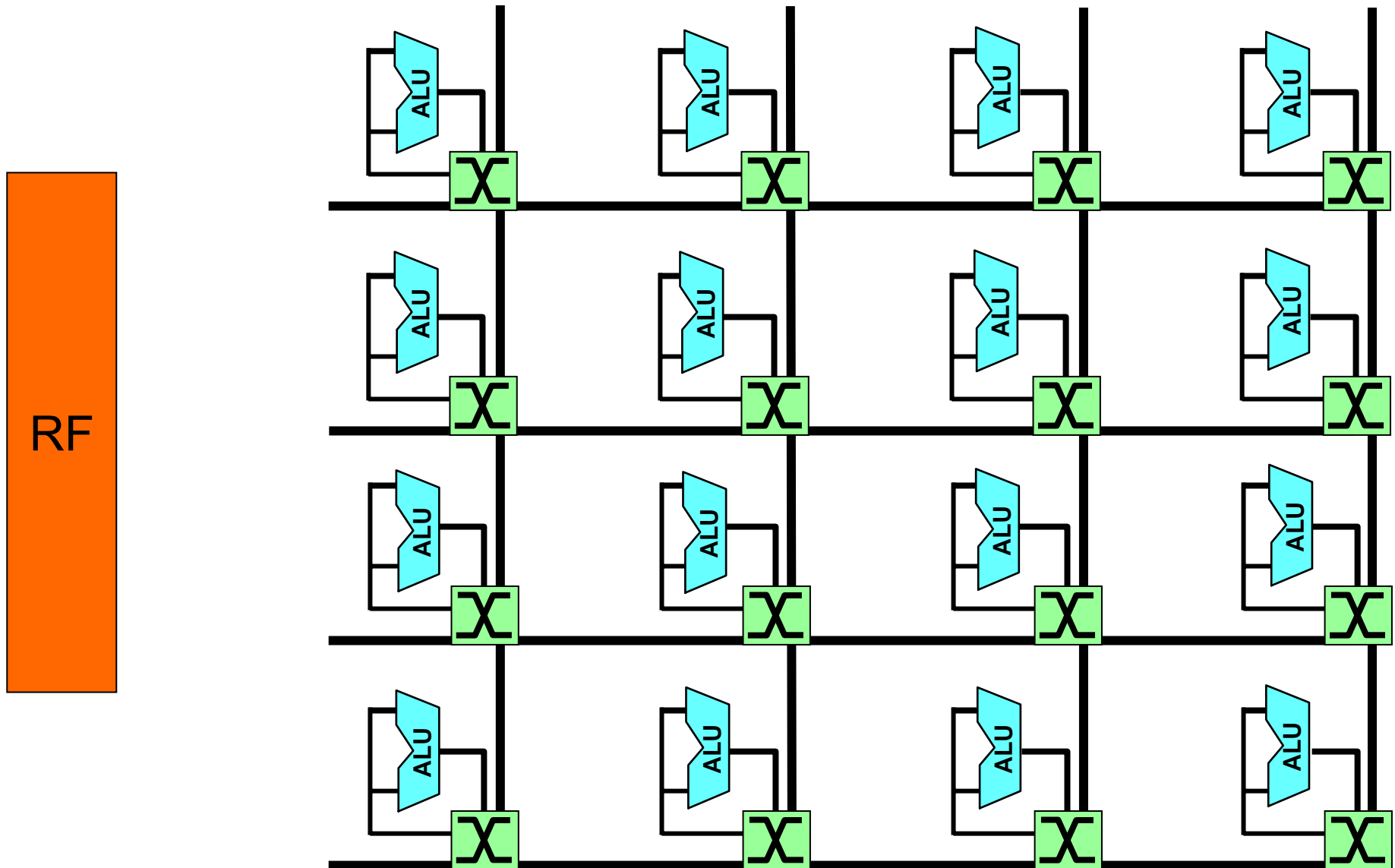
Spatial Architectures



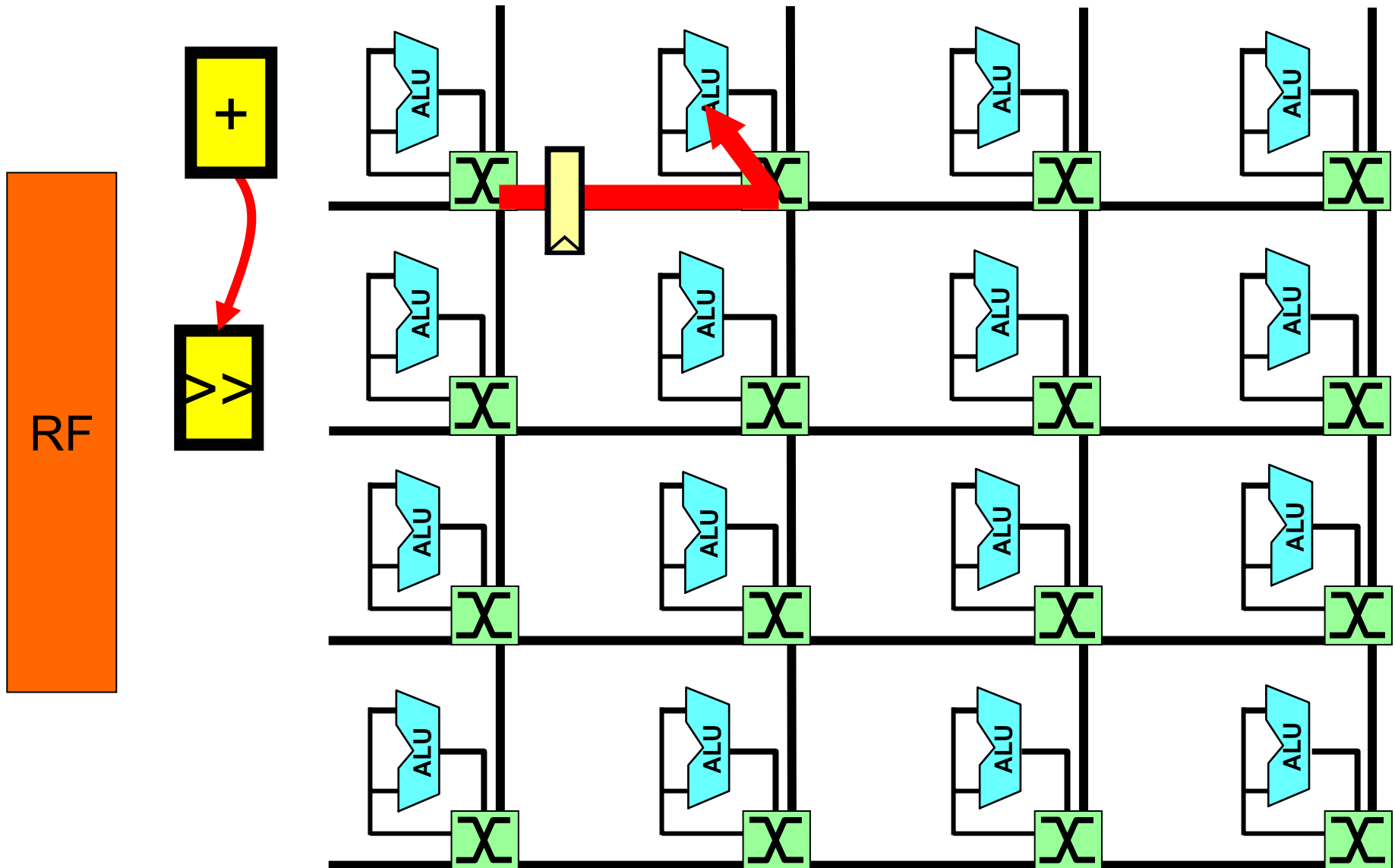
Spatial Architectures



Spatial Architectures



Exploiting Locality



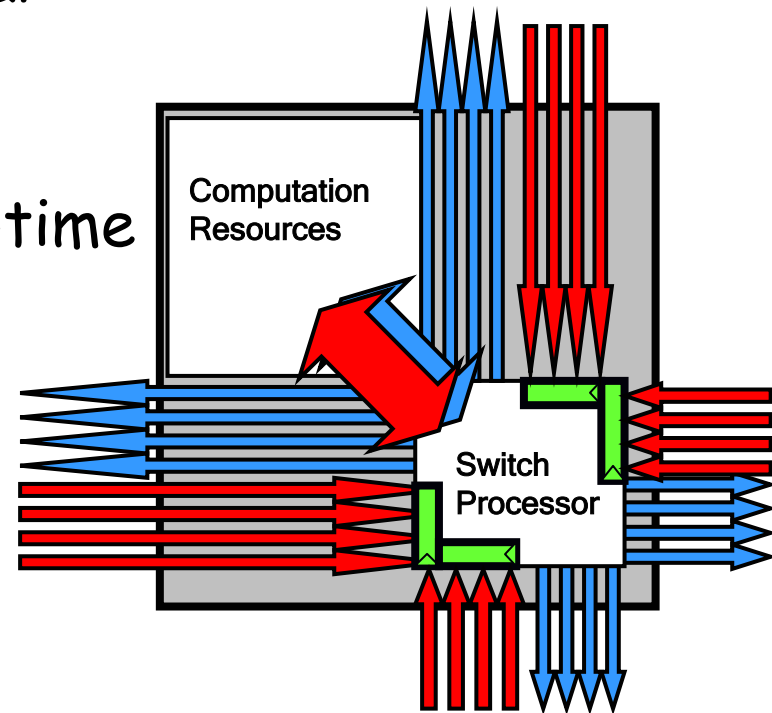
Raw On-Chip Networks

- 2 Static Networks

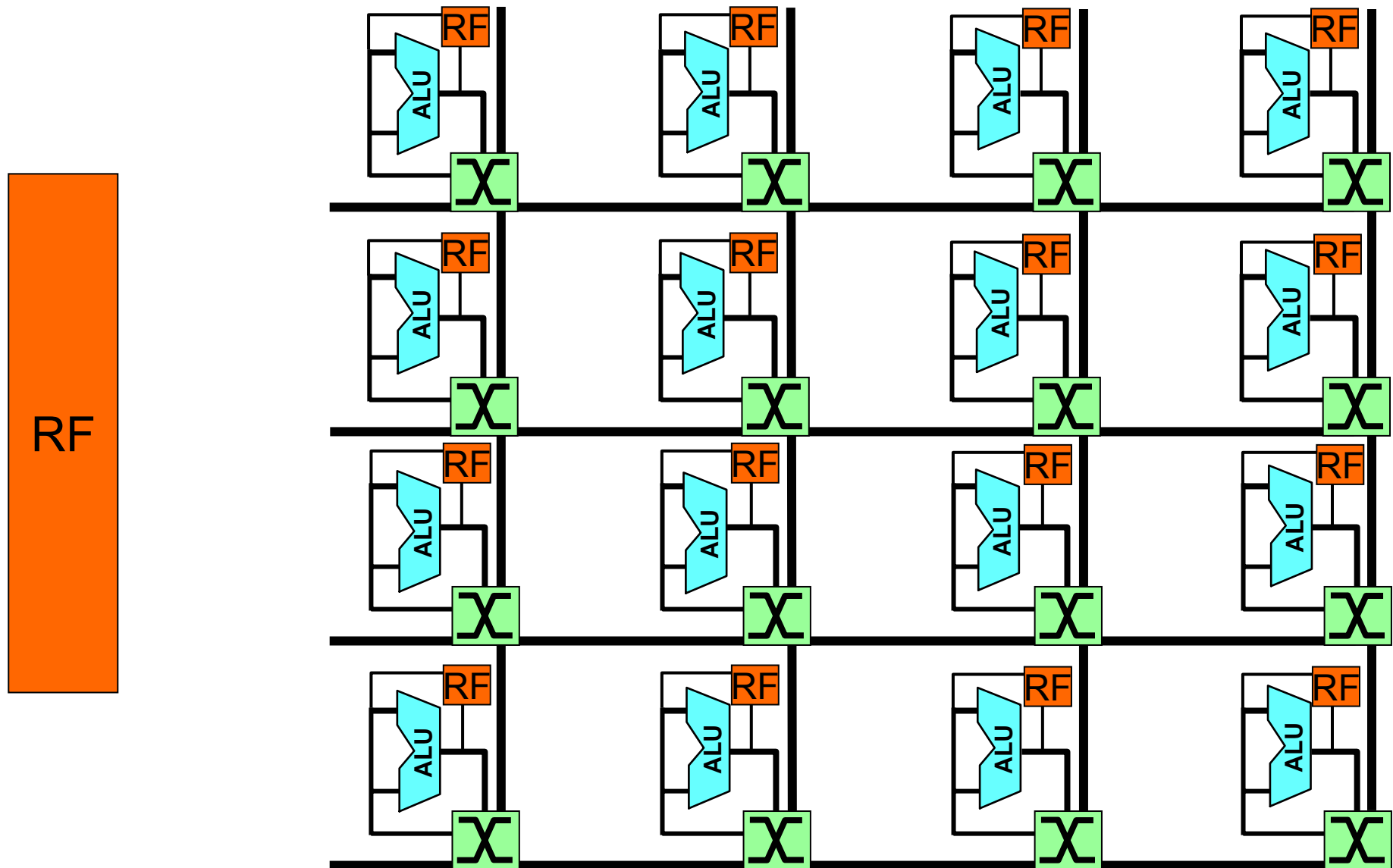
- Software configurable crossbar
- 3 cycle latency for nearest-neighbor ALU to ALU
- Must know pattern at compile-time
- Flow controlled

- 2 Dynamic Networks

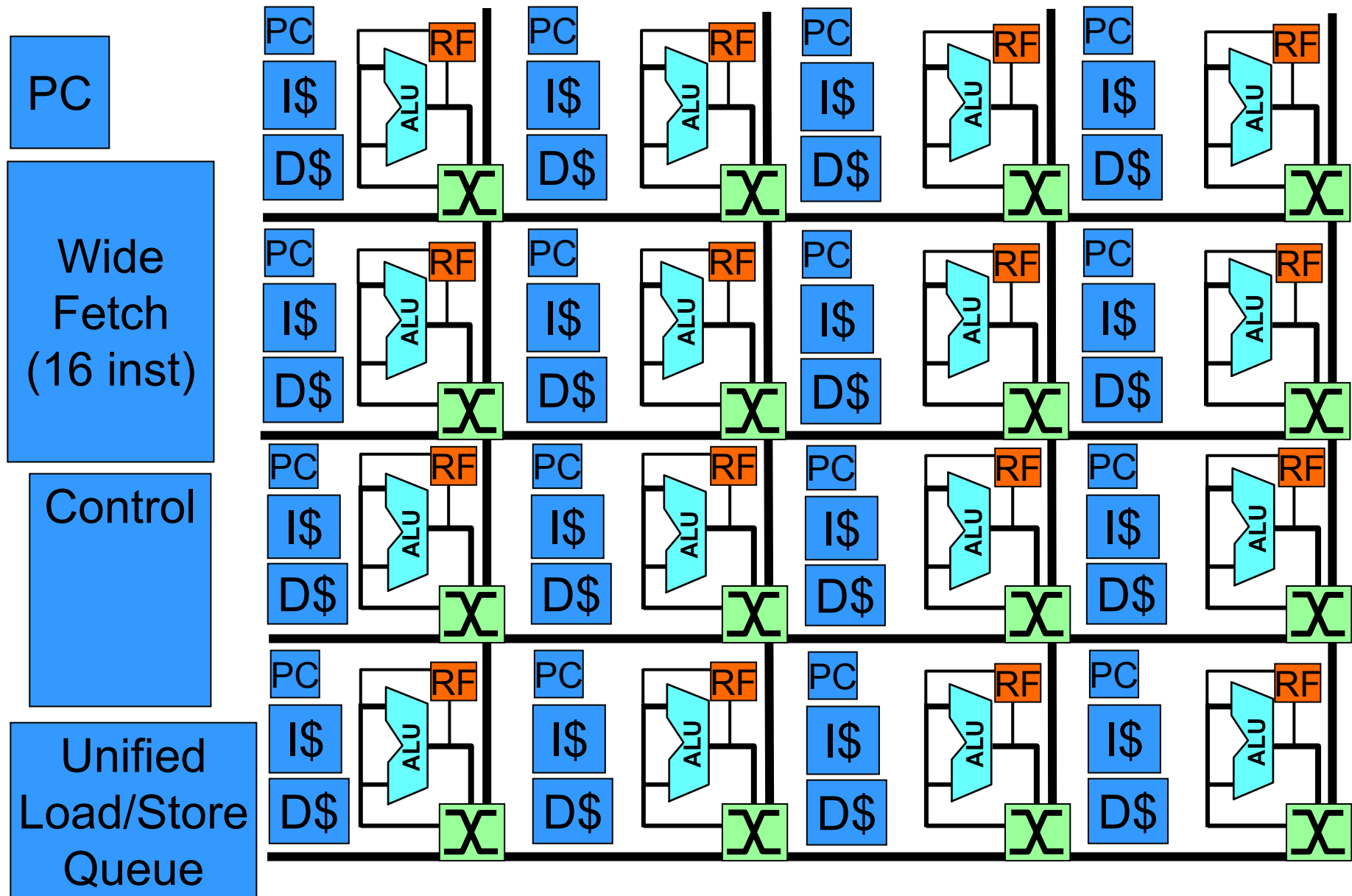
- Header encodes destination
- Fire and Forget
- 15 cycle latency for nearest-neighbor



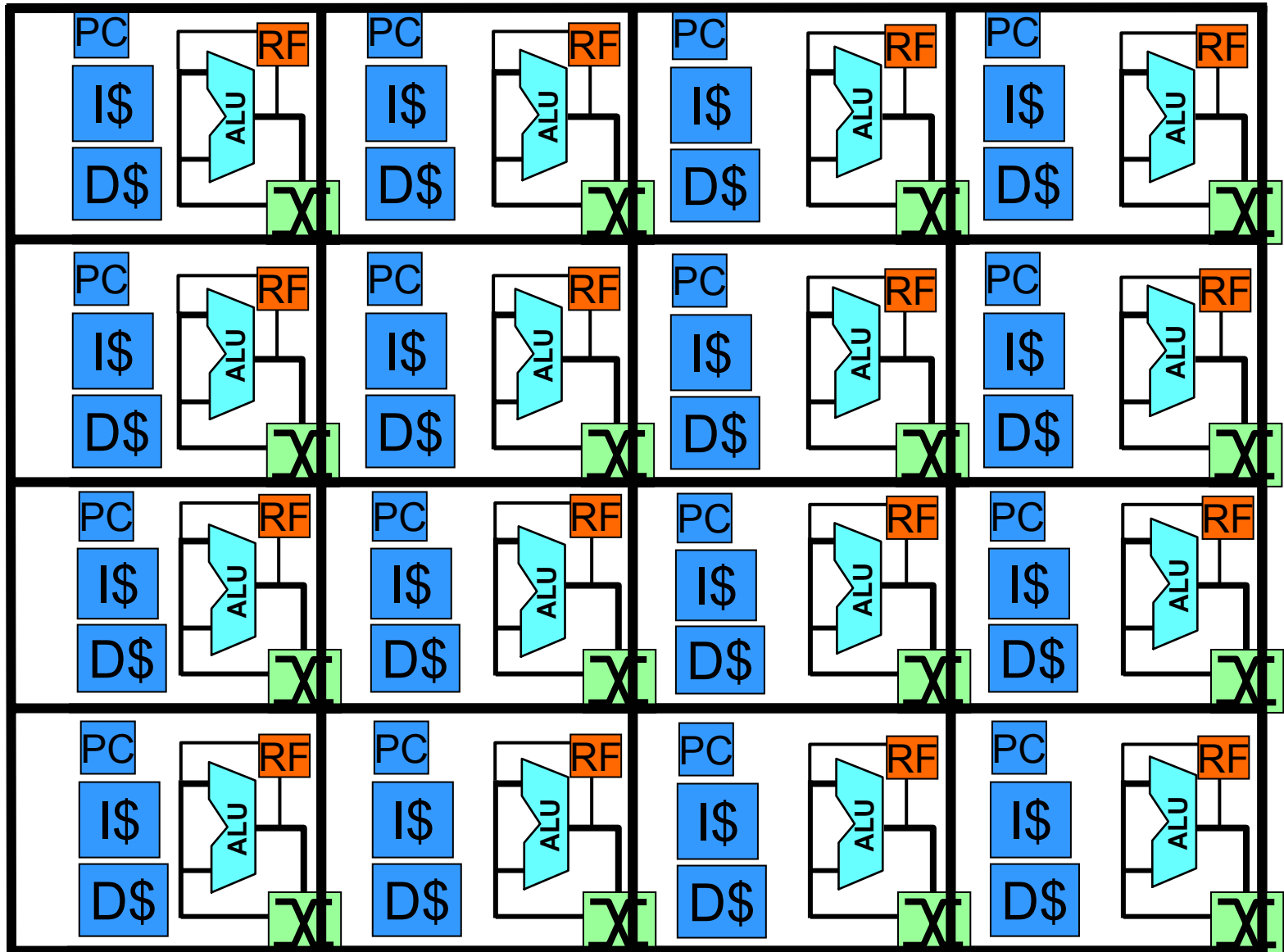
Distribute the Register File



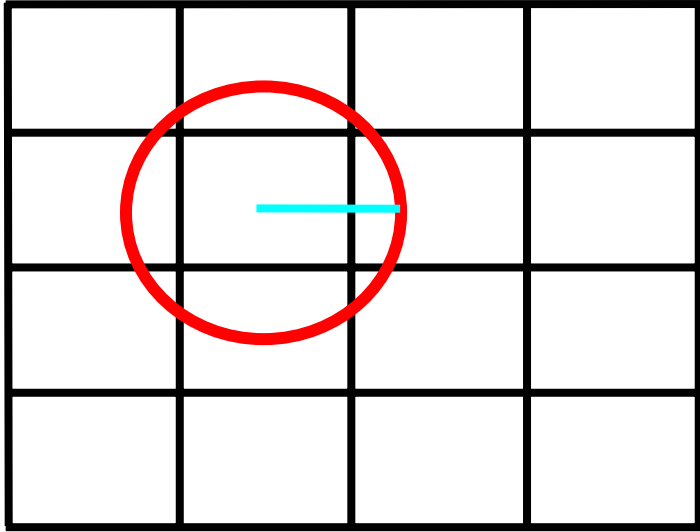
Distribute the Rest



Tiled-Processor Architecture



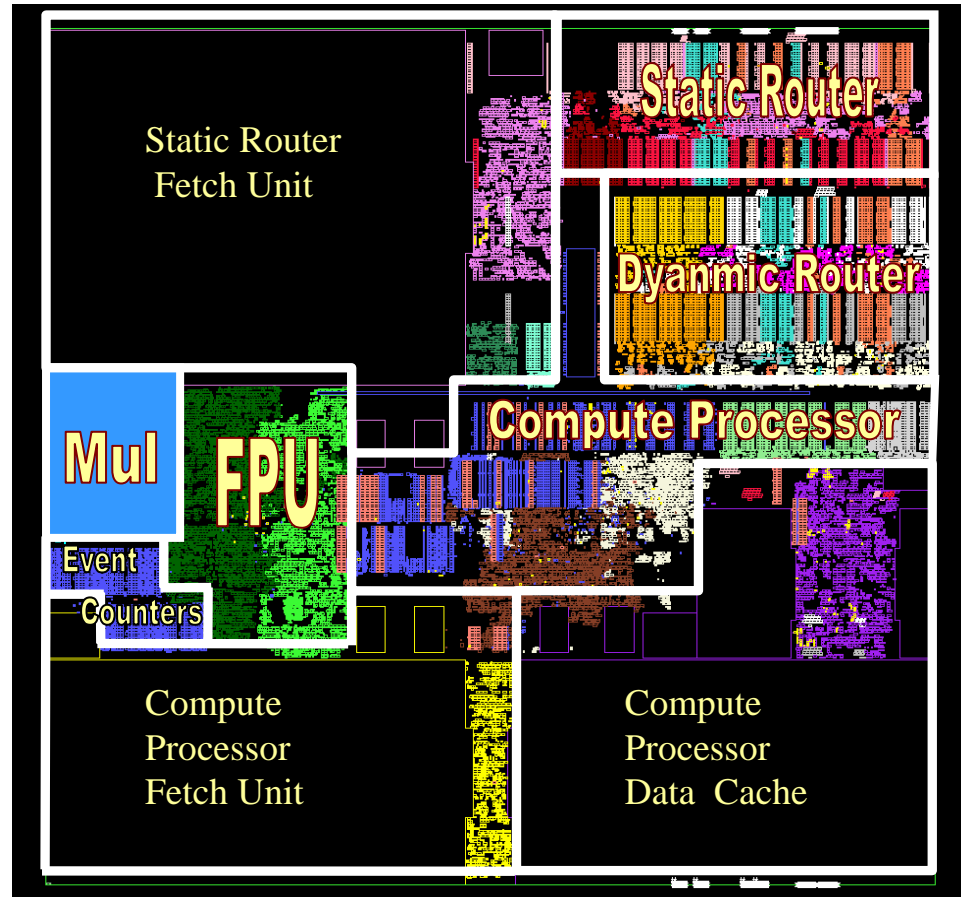
Tiled-Processor Architecture



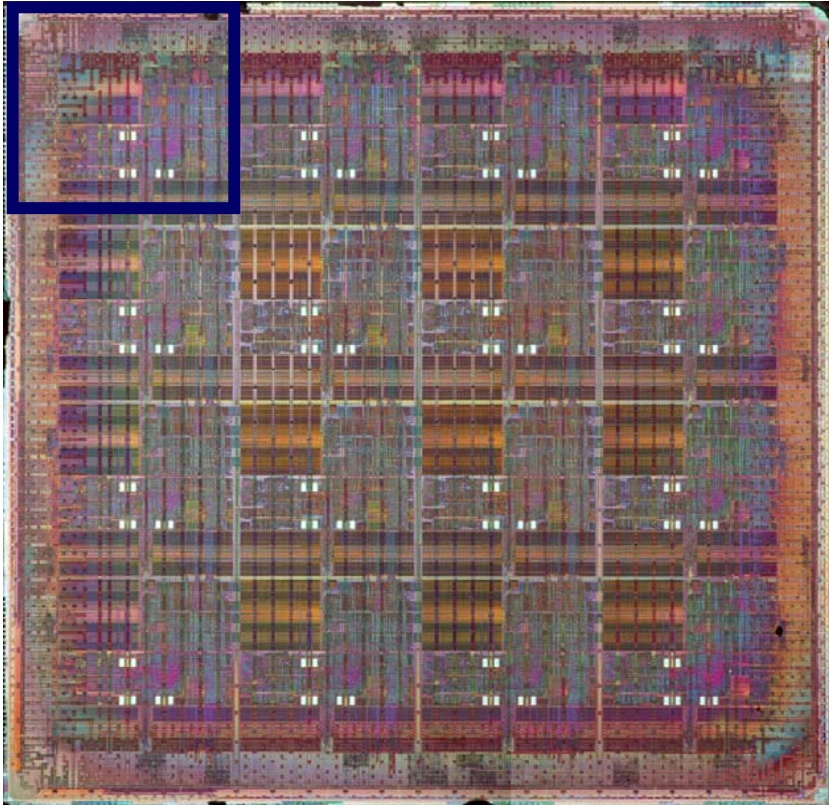
Make a tile as big as you can go in one clock cycle, and expose longer communication to the programmer

- Tile abstraction is quite powerful
 - e.g., power → resources used as necessary
- Easily scalable
 - All signals registered at tile boundaries, no global signals
 - Easier to Tune the Frequency
 - Easier to do the Physical Design
 - Easier to Verify

Close-up of a Single Raw Tile



The MIT Raw Processor



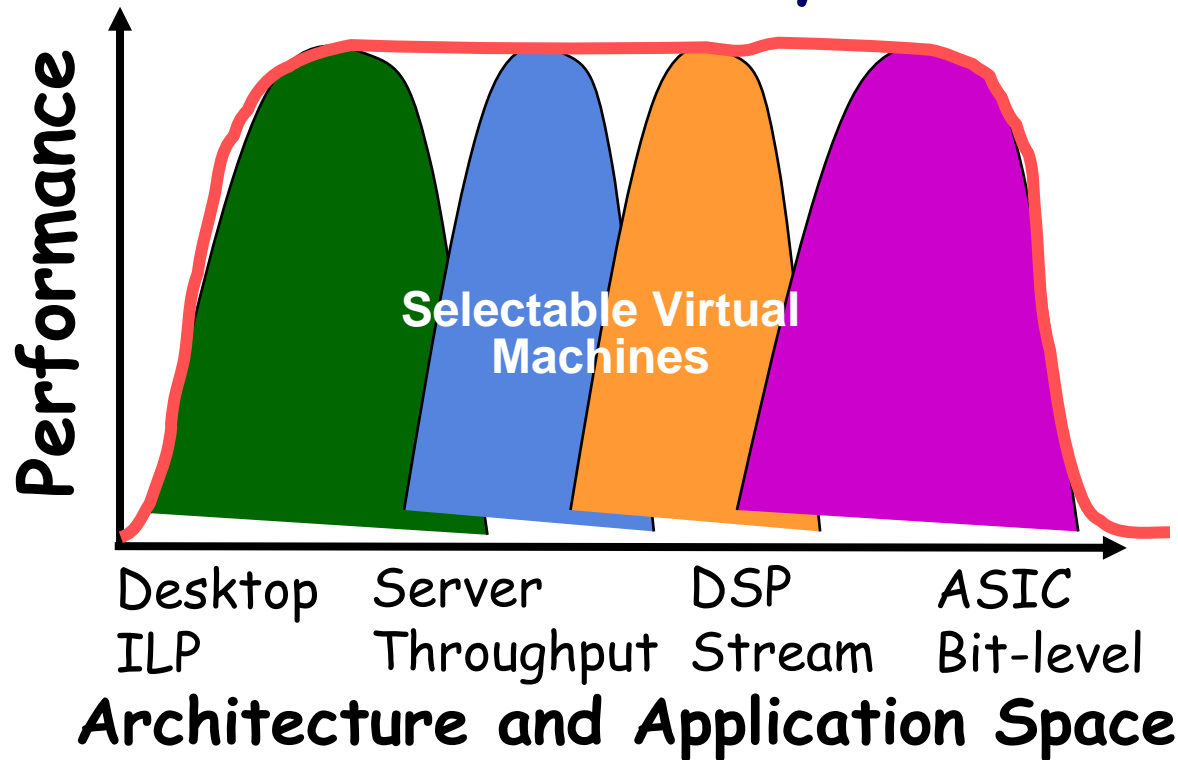
- 180 nm ASIC (IBM SA-27E)
- 16 tiles → 16 issue
- Core Frequency:
 - 425 MHz @ 1.8 V
 - 500 MHz @ 2.2 V
 - Frequency competitive with IBM-implemented PowerPCs in same process
- 18 W (vpenta)

The Raw Goal

- Create an architecture that
 - Scales to 100's-1000's of functional units, memory ports
 - By exploiting custom-chip like features
 - Application-specific routing of operands
 - Is "general purpose" (*Versatile*)
 - Run ILP sequential programs, scientific computations, server-style processing, streaming systems, and bit-level applications
 - Support standard General Purpose Abstractions
 - Context switching, caching and instruction virtualization

The New Performance Goal

Versatility

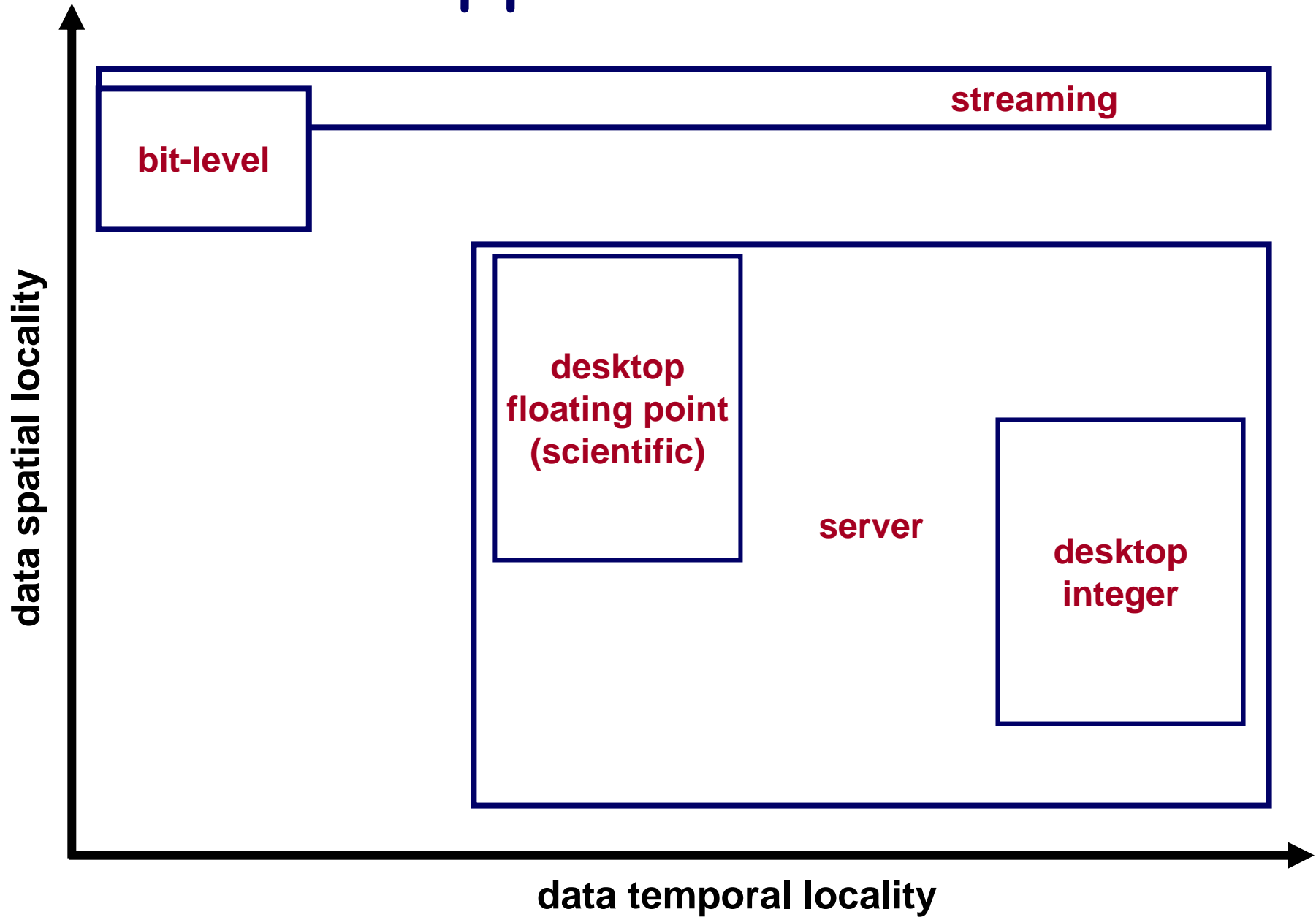


- Raw architecture as an “all-purpose” processor
 - Better SPECmark/Watt across the board
 - Higher SPECmark → think more MIPS compared to some reference machine (e.g., VAX 11/780)

Application Domains

- 5 market-dominant application domains
 - Desktop Integer
 - Desktop Floating (Scientific codes)
 - Server (Throughput Based)
 - Ergonomic simulations, Grid computation, Transaction processing
 - Embedded Streaming
 - Embedded Bit-Level

How Applications Differ



Distinguishing Application Domains

- Five **basis** properties
 - Data temporal locality
 - Quantify address reuse
 - Spatial temporal locality
 - Quantify address adjacency
 - Predominant data type
 - Parallelism
 - ILP, DLP, TLP, etc
 - Instruction temporal locality
 - Inverse of control complexity

Classifying Applications

- Quantitative metrics for the basis properties
 - Measure properties of different applications
- Cluster applications into domains
 - VersaBench

	Data Type	Parallelism	Instruction Temporal Locality	Data Temporal Locality	Data Spatial Locality
Desktop INT	integer	low	low	high	low
Desktop FLT	float	medium	medium	medium	medium
Server	integer/float	high	low to medium	medium to high	low to medium
Streaming	integer/float	very high	high	low to high	very high
Bit-Level	bit	medium-high	very high	low	very high

VersaBench Status

- 15 total benchmarks
 - 3 per category
 - Drawn from SPEC INT/FP, Raw, StreamIt, DIS (AAEC), USC ISI
 - Manageable size, encourages evaluation using the entire suite
 - Available online at <http://cag.csail.mit.edu/versabench>
- Benchmarks selected systematically
 - MIT Technical Memo 646, June 2004
 - Rabbah, Bratt, Asanovic, Agarwal

Proposed Metric: Versatility

Versatility (VersaBench)

Geometric Mean of Speedup relative to **best** performing machines

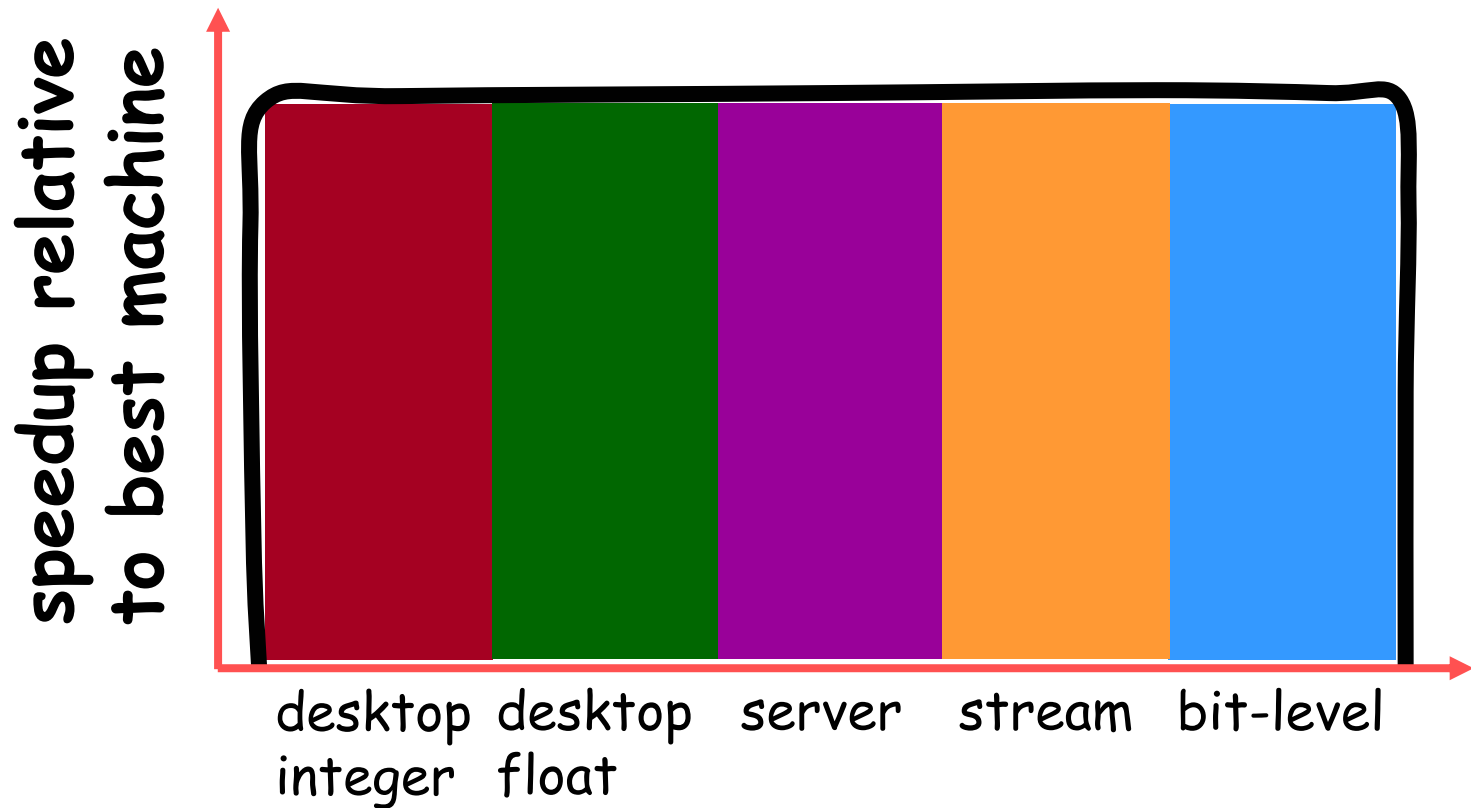
SPECmark (SPEC)

Geometric Mean of Speedup relative to a **single** reference machine

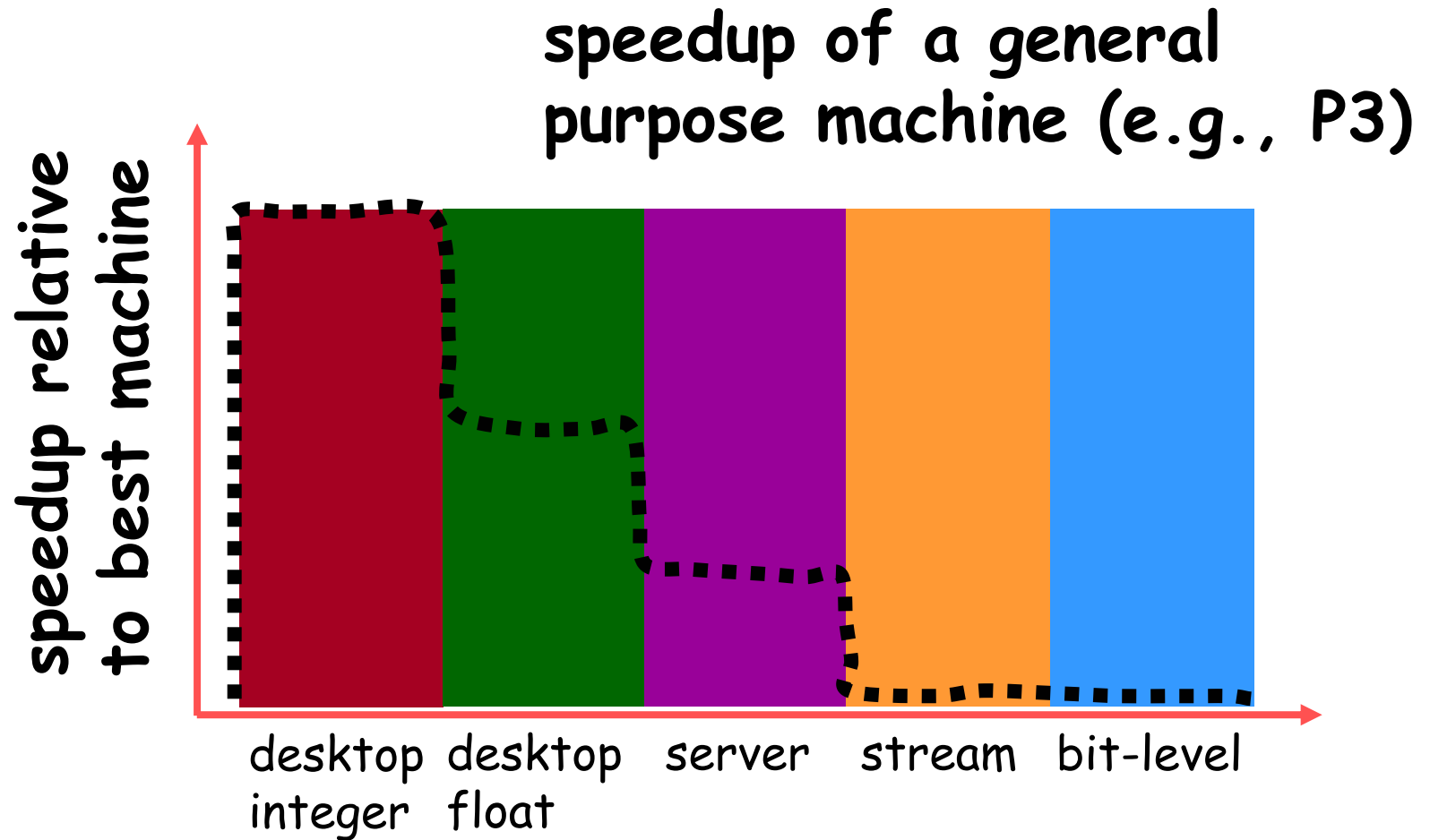
- Normalization to the best performing machines identifies areas for improvements
 - This is especially important → **VersaGraphs**
 - Not another mean over N benchmarks
- High Versatility mark implies architecture is good across the board

VersaGraph Example

speedup of an ideal machine

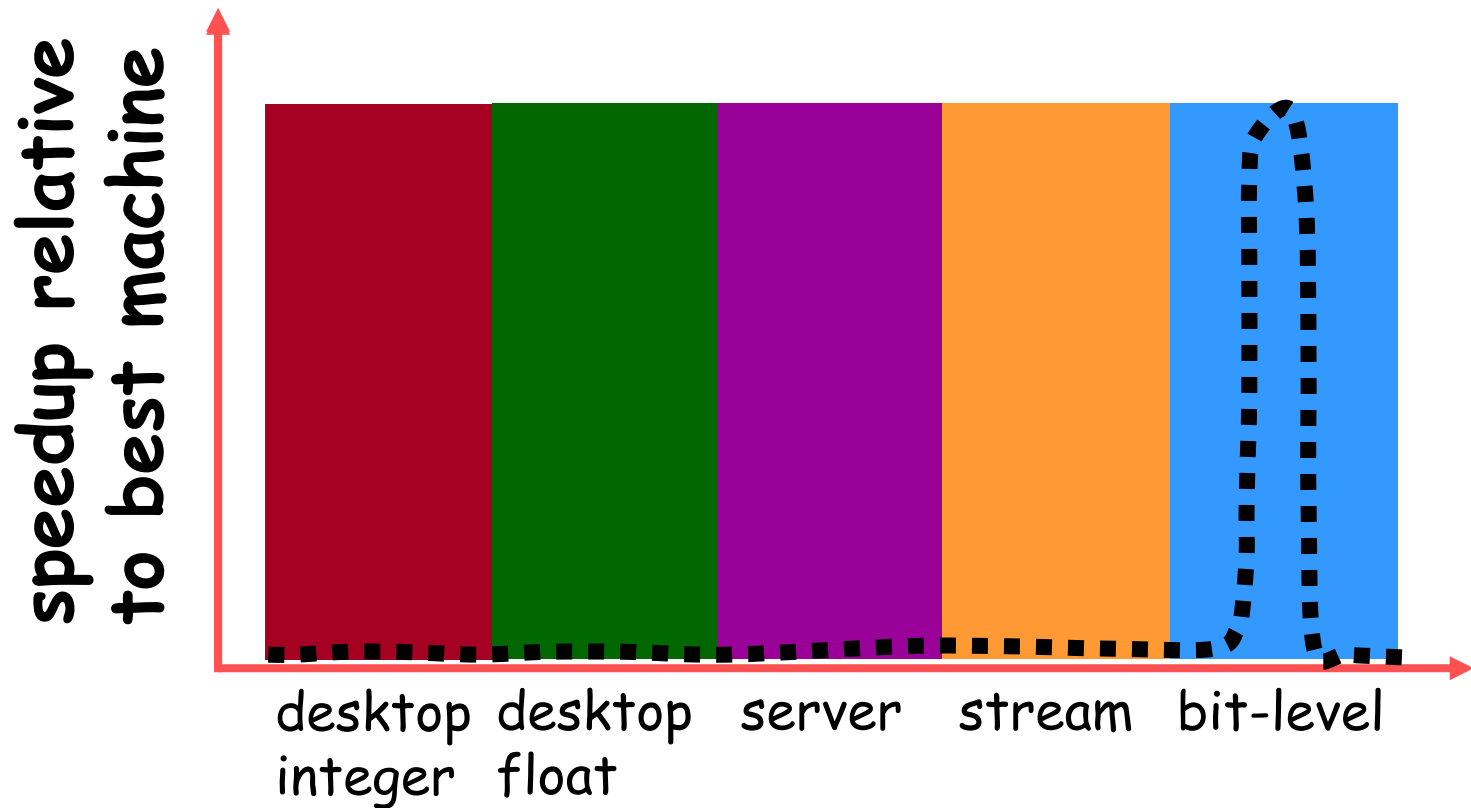


VersaGraph Example



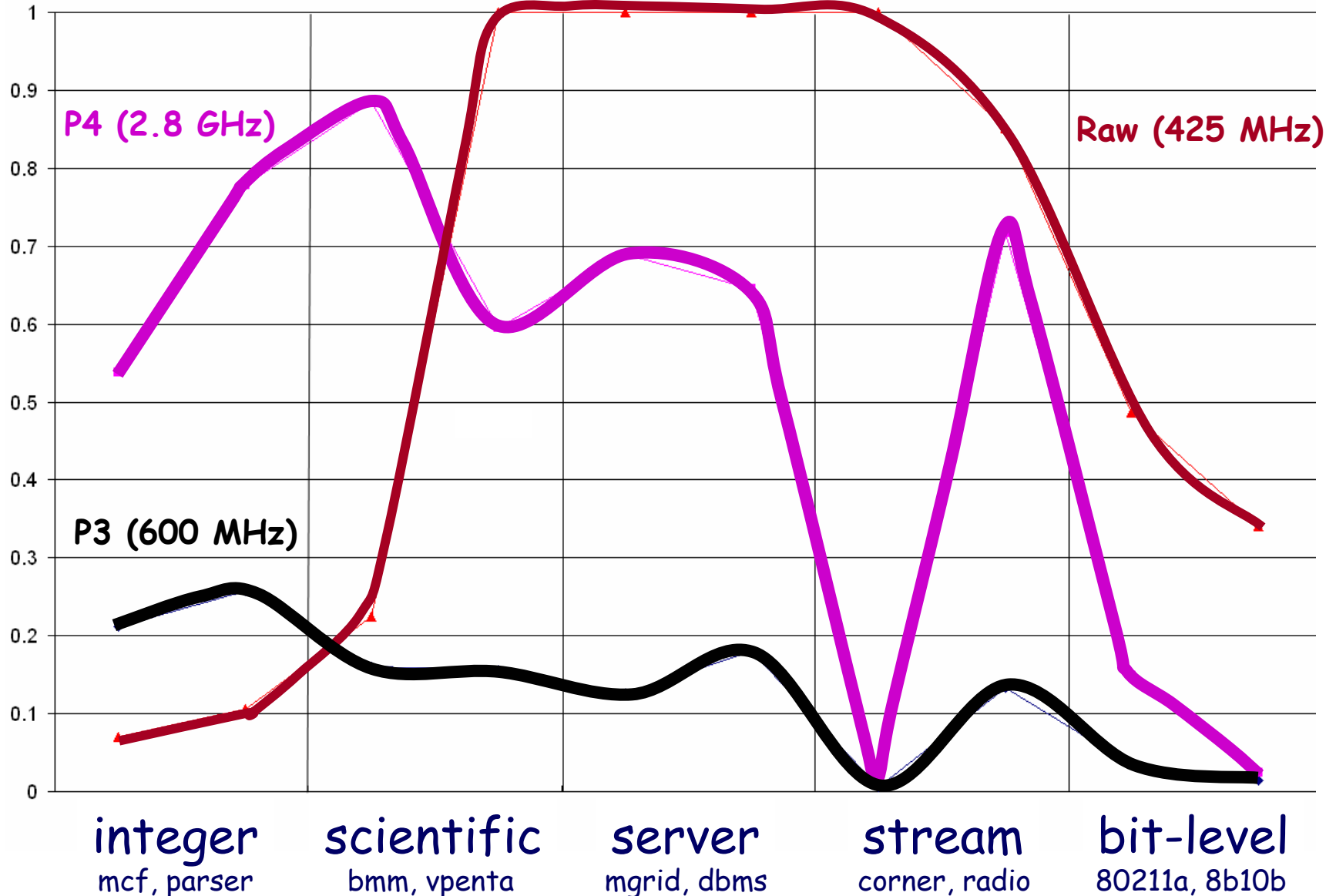
VersaGraph Example

speedup of an ASIC



VersaGraphs For Real Architectures

Also compared against Athlon 64 and Itanium 2



Raw Homepage

<http://cag.csail.mit.edu/raw>

download papers, benchmarks, ...