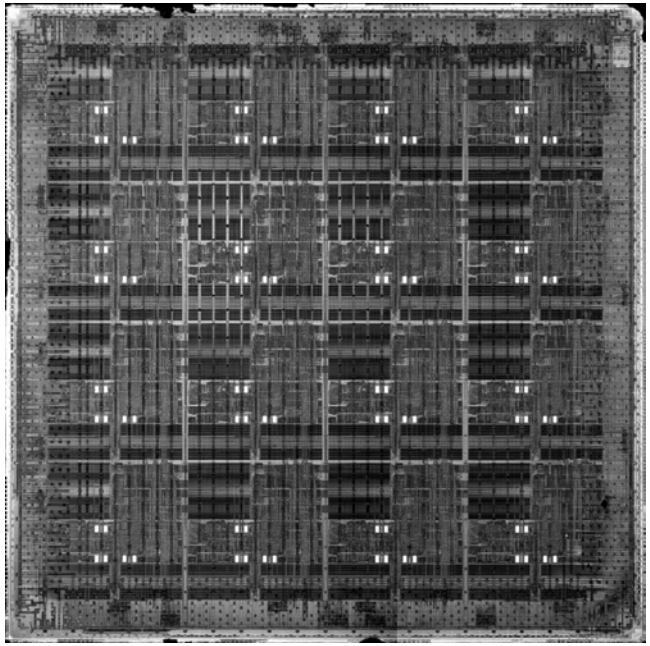
The Raw Architecture

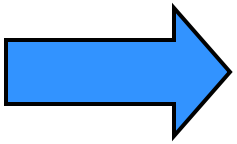
A Concrete Perspective



Michael Bedford Taylor

Raw Architecture Group
Laboratory for Computer Science
Massachusetts Institute of Technology

Outline



Raw Architectural Overview

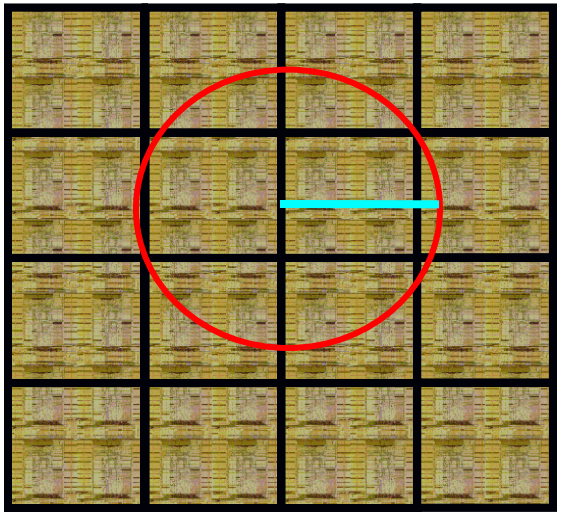
Compute Processor

On-chip networks

Architectural Usage hints

Raw exploration avenues

The Raw Architecture



Divide the silicon into an array of identical, programmable tiles.

Prototype: 4x4

Arch, Sim & Fabric: {4,8,16,32} x {4,8,16,32}

The Raw Tile

Tile

8 stage 32b MIPS-style single-issue in-order compute processor

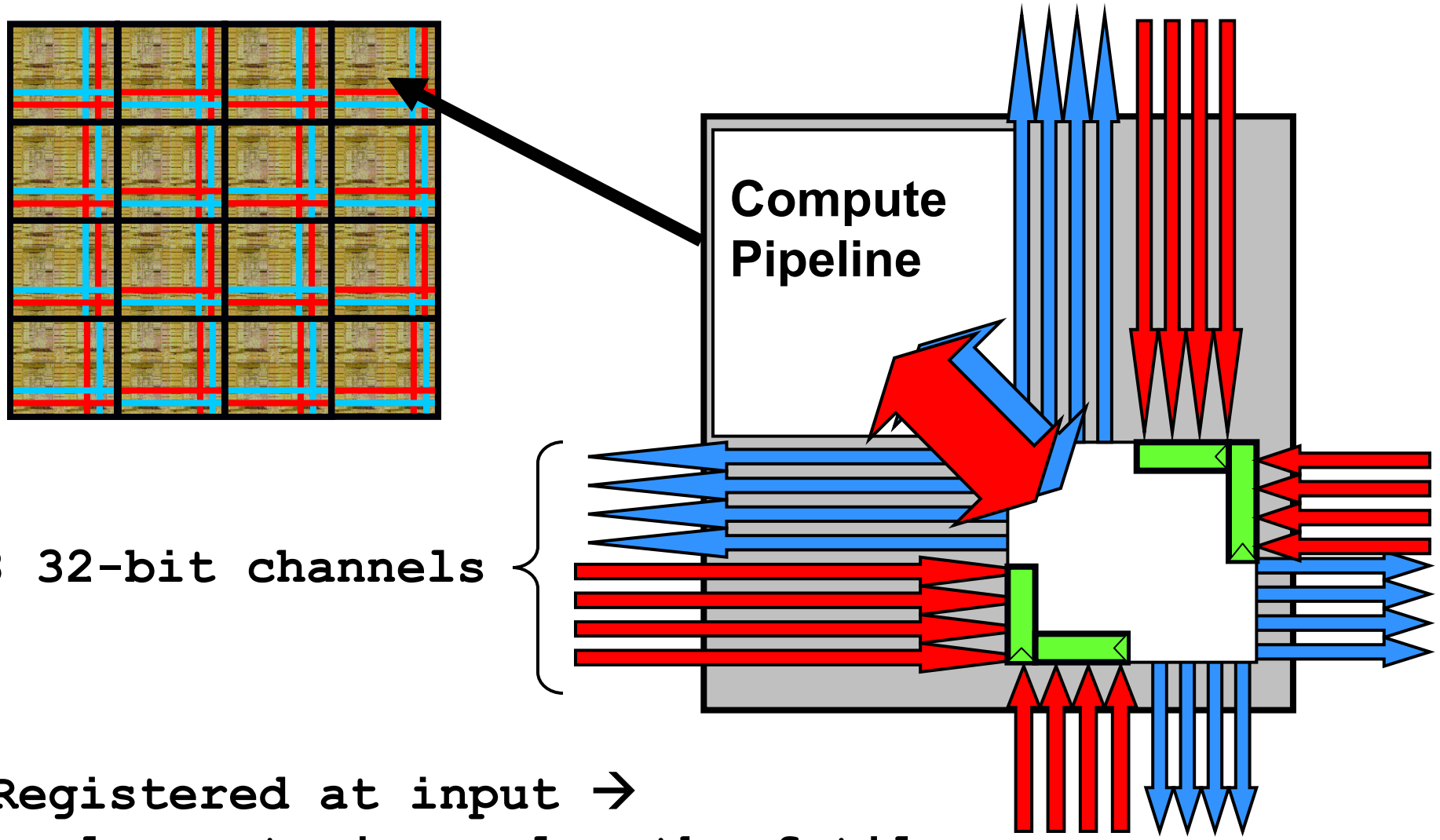
32 KB IMem

32 KB DCache

4-stage 32b pipelined FPU

Routers and wires for three on-chip mesh networks

Raw's four on-chip mesh networks

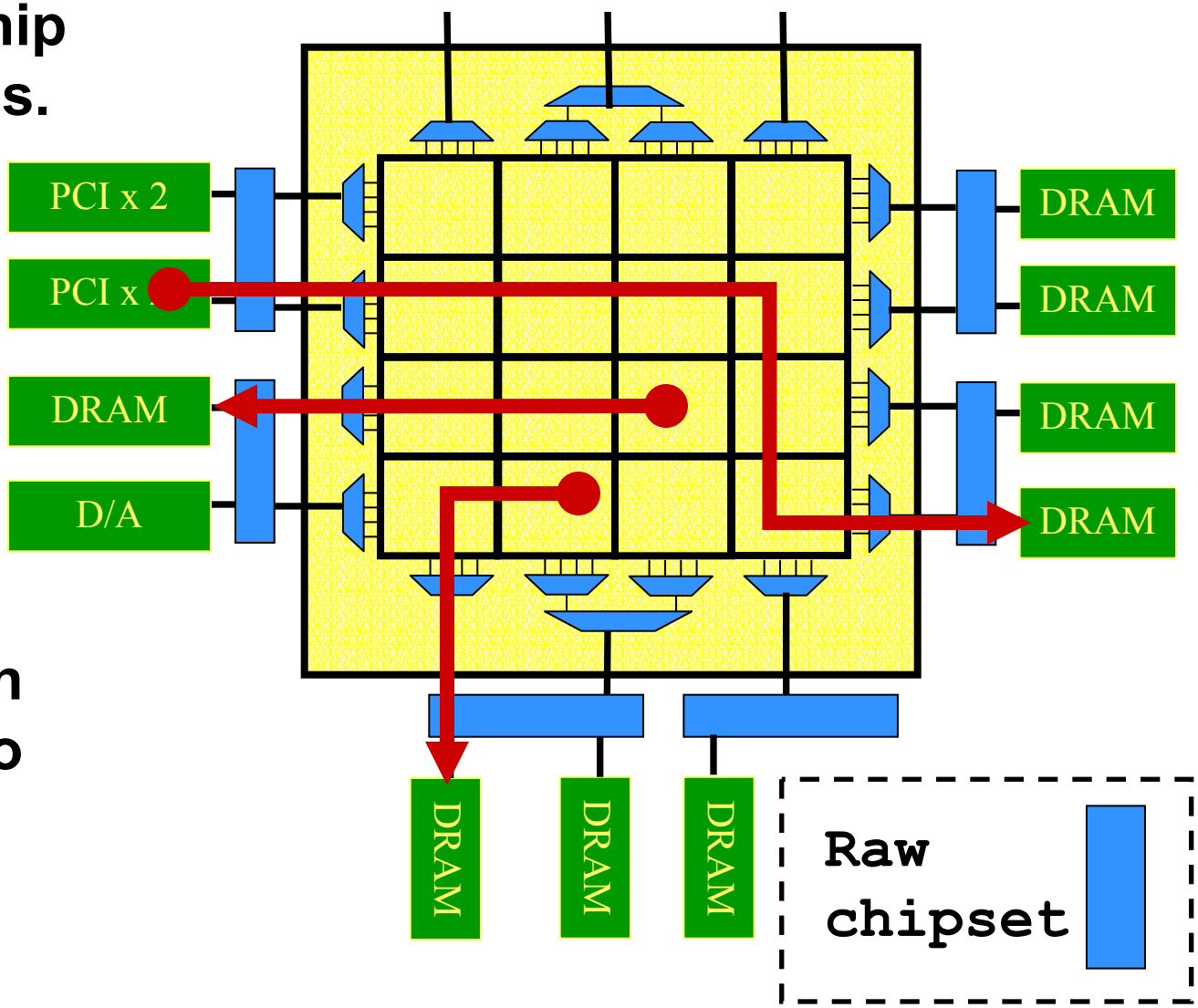


Registered at input →
longest wire = length of tile

Raw's I/O and Memory System

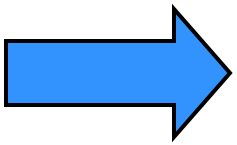
14 7.2 Gb/s channels
(201 Gb/s @ 225 Mhz)

Routes on <u>any</u> network off the edge of the chip appear on the pins.



Direct connection of computation to I/O and DRAM Streams

Outline



Raw Architectural Overview

Compute Processor

On-chip networks

Architectural Usage hints

Raw exploration avenues

Raw compute processor ISA

MIPS-flavour

-- similar instructions, with improvements

-- all but divide instructions are fully pipelined

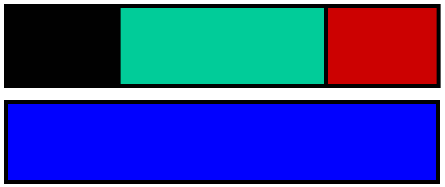
LOAD	3 cycles
ALU	1 cycles
MUL	2 cycles
FPU	4 cycles

Similar latencies as Pentium-3

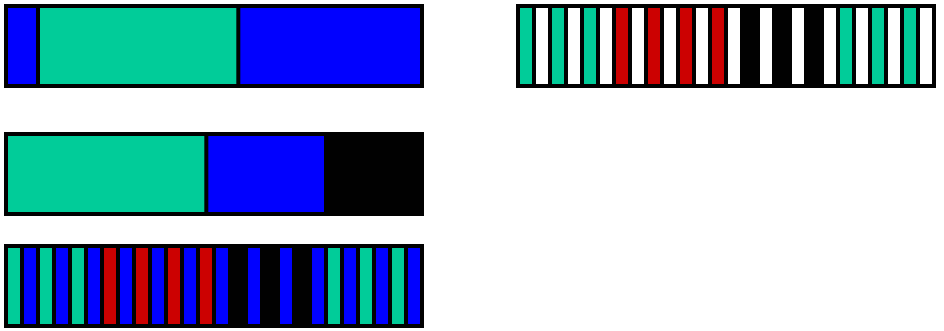
Bit-oriented instructions

PopCount
Count Leading Zero

Rotate and Mask / Rotate, Mask and Insert
- Like PowerPC, but more flexible masks
  Bit-smashing swiss army knife



inputs



sample outputs after 1 cycle

Branches & Divides

1 branch per cycle
- no delay slots
- static branch prediction bit (e.g., bne+ bne-)
- 3 cycle mispredict penalty

Integer Divide	42 cycles
Floating Point Divide	12 cycles

The only non-pipelined instructions in the ISA.
Uses FD and HI/LO registers with full/empty bits.

Proc, IntDiv, FPDiv State Machines run independently
Processor will only stall if you read MFFD, MFLO, MFHI
before dividers have finished.

Bnea

bnea $2,$3, addr

branch not equal, meanwhile, add SPR BR_INCR

minor tweak reduces some loop overheads by %50

```
        li   $4, kCacheSize - (kCacheLineSize << 2)
        li   $5, 0
        mtsri  BR_INCR, (kCacheLineSize << 2)

___cache_invalidate_loop:
        tagswi $0,$5, 0
        tagswi $0,$5, 1
        tagswi $0,$5, 3
        tagswi $0,$5, 2
        bnea+  $5, $4, ___cache_invalidate_loop
```

BR_INCR is callee-saved

Data Cache

Data Cache

32KB per tile, 2-way set associative
allocate-on-write, write-back
tile caches are not coherent

- user manages coherence

Extensive collection of cache management instrs
for coherence management

- tag-index-based for wide address range ops
- address-based for short ranges

More Instructions

Many other instructions including:

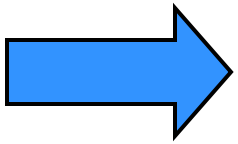
writing to instruction and switch memories
packaging messages
managing interrupts
managing I/O and off-chip memory
adding an immediate value to top 16 bits of word
(useful for $gp-relative calculations)

See the Raw Spec, available off of Raw website.

Outline

Raw Architectural Overview

Compute Processor



On-chip networks

Architectural Usage hints

Raw exploration avenues

Raw's four on-chip networks

General Dynamic Network "gdn"

Memory Dynamic Network "mdn"

Raw's Static Network

Highest performance solution

For predictable communication patterns

Use it:

To route operands

among local and remote ALUs

To route data streams

among tiles, DRAM, I/O ports

Raw's Static Network

Consists of two tightly-coupled sub-networks:

- Tile interconnection network

  For operands & streams between tiles

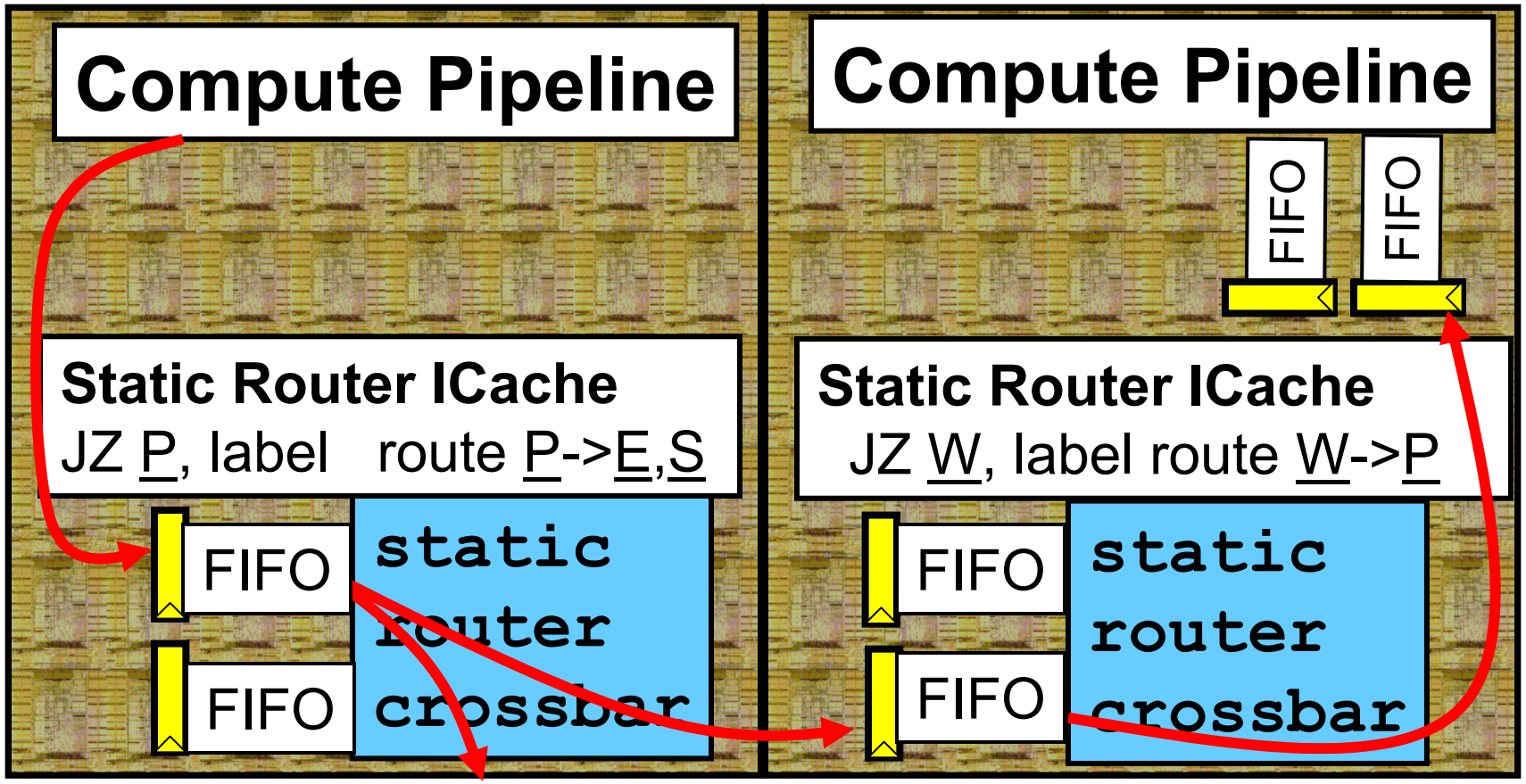
  Controlled by the 16 tiles' static router processors

- Local bypass network

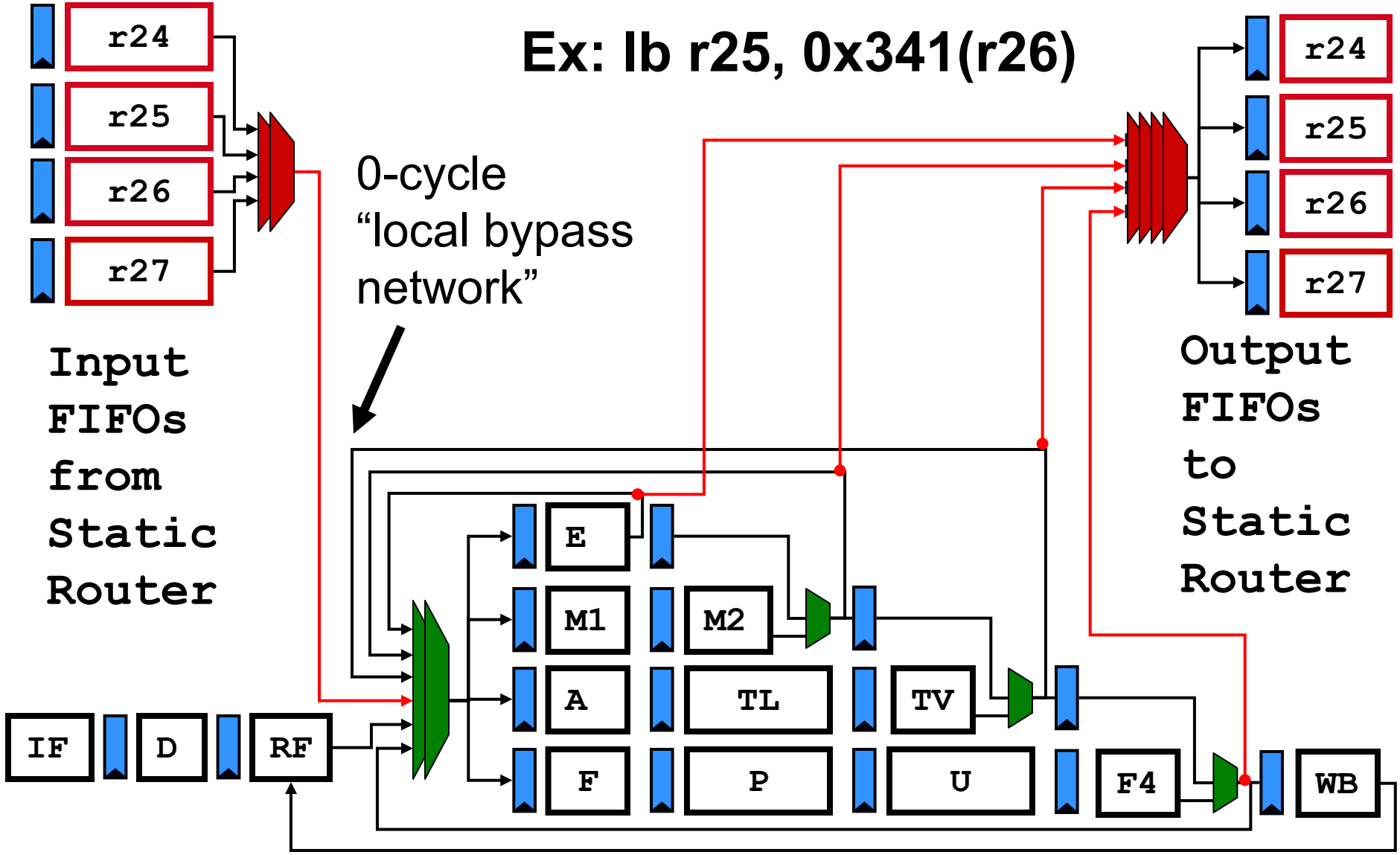
  For operands & streams within a tile

Between tile operand transport

The routes programmed into the static router ICache guarantee in-order delivery of operands between tiles at a rate of 2 words/direction/cycle.



Operand Transport among functional units and the static router



Raw static router ISA

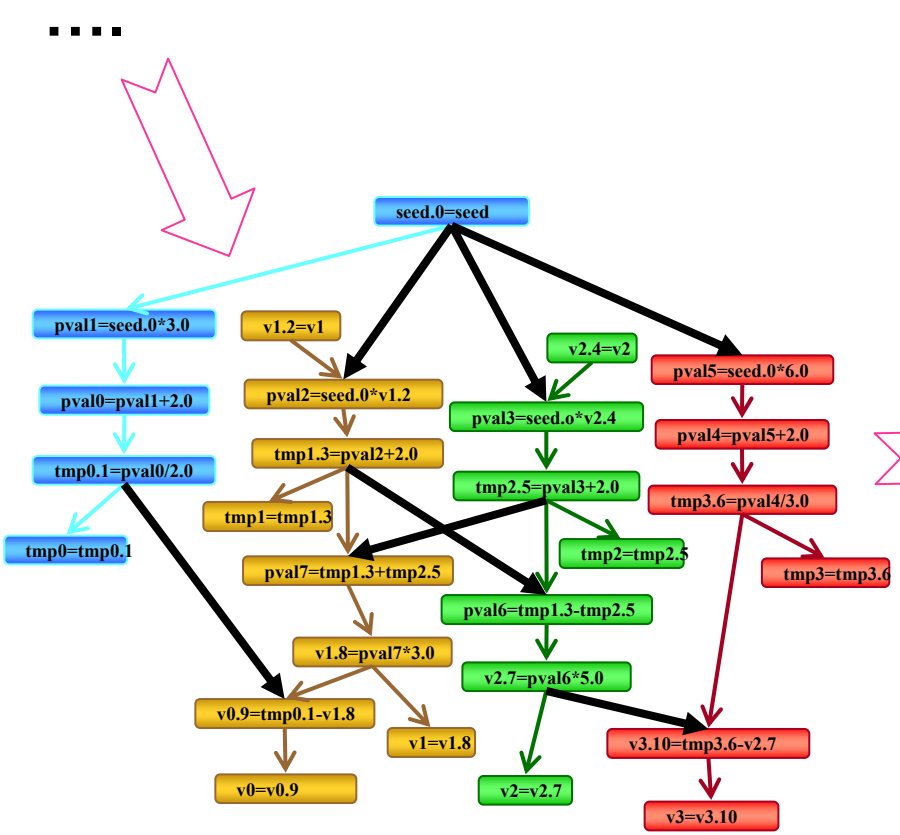
Branches, nops, and up to 13 routes per cycle
Routes support multicast.

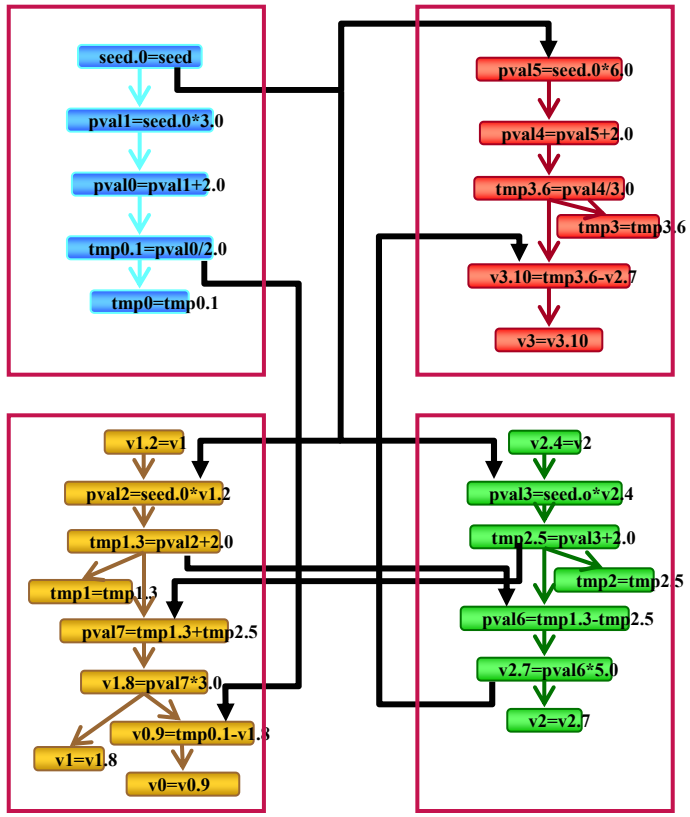
move	$1, $csto	route $csto->$cWo, $csto->$cEo
beqzd-	$1,$1, cleanup	route $csto->$cWo
bnezd+	$1,$1, cleanup	route $csto->$cWo, $cWi->$cNo

Compilation

tmp3 = (seed*6+2)/3
v2 = (tmp1 - tmp3)*5
v1 = (tmp1 + tmp2)*3
v0 = tmp0 - v1
….

Assign instructions to the tiles, maximizing locality.
Generate the static router instructions to transfer Operands & streams tiles.





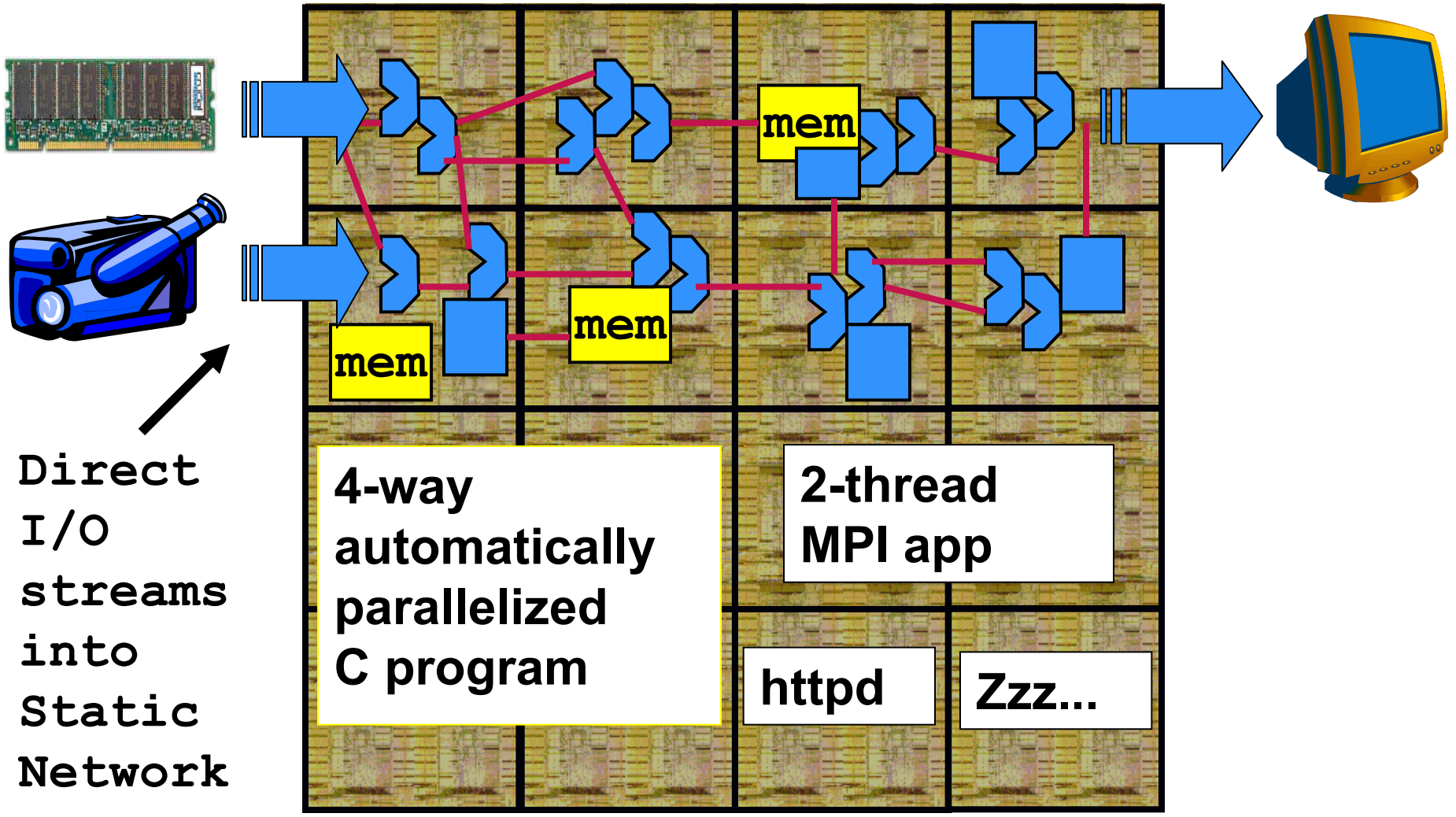
Raw's Memory Network

- Dynamic, dimension-ordered wormhole-routed
  - Insert header, and < 32 data words.
  - Worms through network.
  - Inter-message ordering not guaranteed.
- Trusted clients, stringent usage rules
  - Cache misses
  - I/O Devices <-> Memory DMA traffic
  - Operating System
  - Expert users
- Talk to us if you'd like to use it.

Raw's General Network

- Dynamic, dimension-ordered wormhole-routed
  - Insert header, and < 32 data words.
  - Worms through network.
  - Inter-message ordering not guaranteed.
- User-level messaging
  - Can interrupt tile when message arrives
- Lower performance; for coarse-grained apps
- For non-compile time predictable communication
  - among tiles
  - possibly with I/O devices

A Raw System in Action



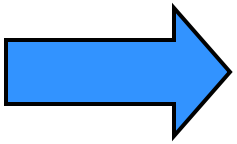
Note that an application uses only as many tiles as needed to exploit the parallelism intrinsic to that application.

Outline

Raw Architectural Overview

Compute Processor

On-chip networks



Architectural Usage hints

Raw exploration avenues

Usage hint #1

If a program operates on linear arrays of data and spends many cycles pulling them in through cache misses:

→ Stream data in/out of DRAMs
via the static network instead of through cache.

Usage hint #2

If the tiles are all stalled, and the DRAM ports are busy.

→ Use more DRAM ports!
Up to 16 DRAM banks in,16 DRAM banks out.

Usage hint #3

If the computation is fine-grained and the code does not use the static network:

→ The static network is the lowest latency, lowest occupancy communication mechanism on Raw. Using it will greatly reduce communication overhead.

Usage hint #4

If tiles receive data from the static network, and store it to the local data memory before using it:

→ The static network guarantees in-order arrival of data. Use the data directly as it comes off the network, to save the latency and occupancy overhead.

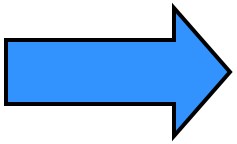
Outline

Raw Architectural Overview

Compute Processor

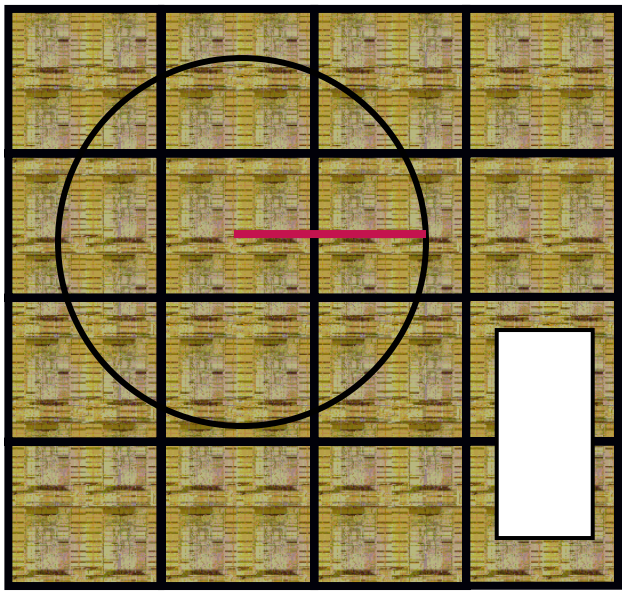
On-chip networks

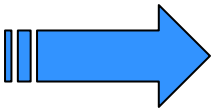
Architectural Usage hints



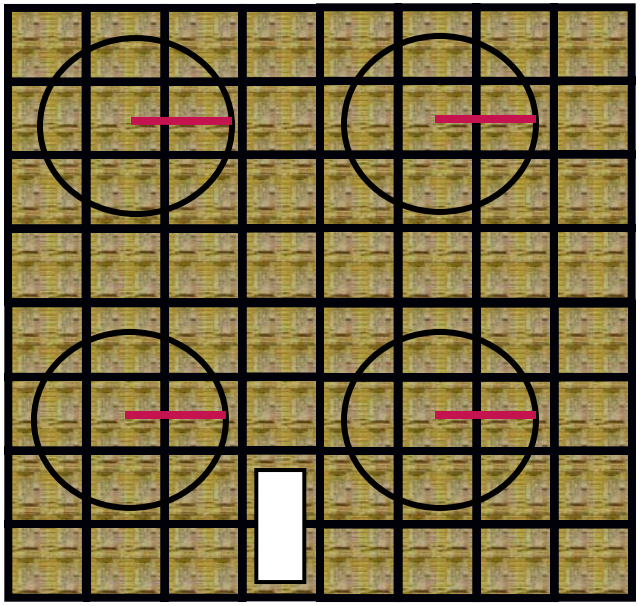
Raw exploration avenues

Looking into the future





1 cycle



180 nm
16-issue

90 nm
64-issue

Just stamp out more tiles!

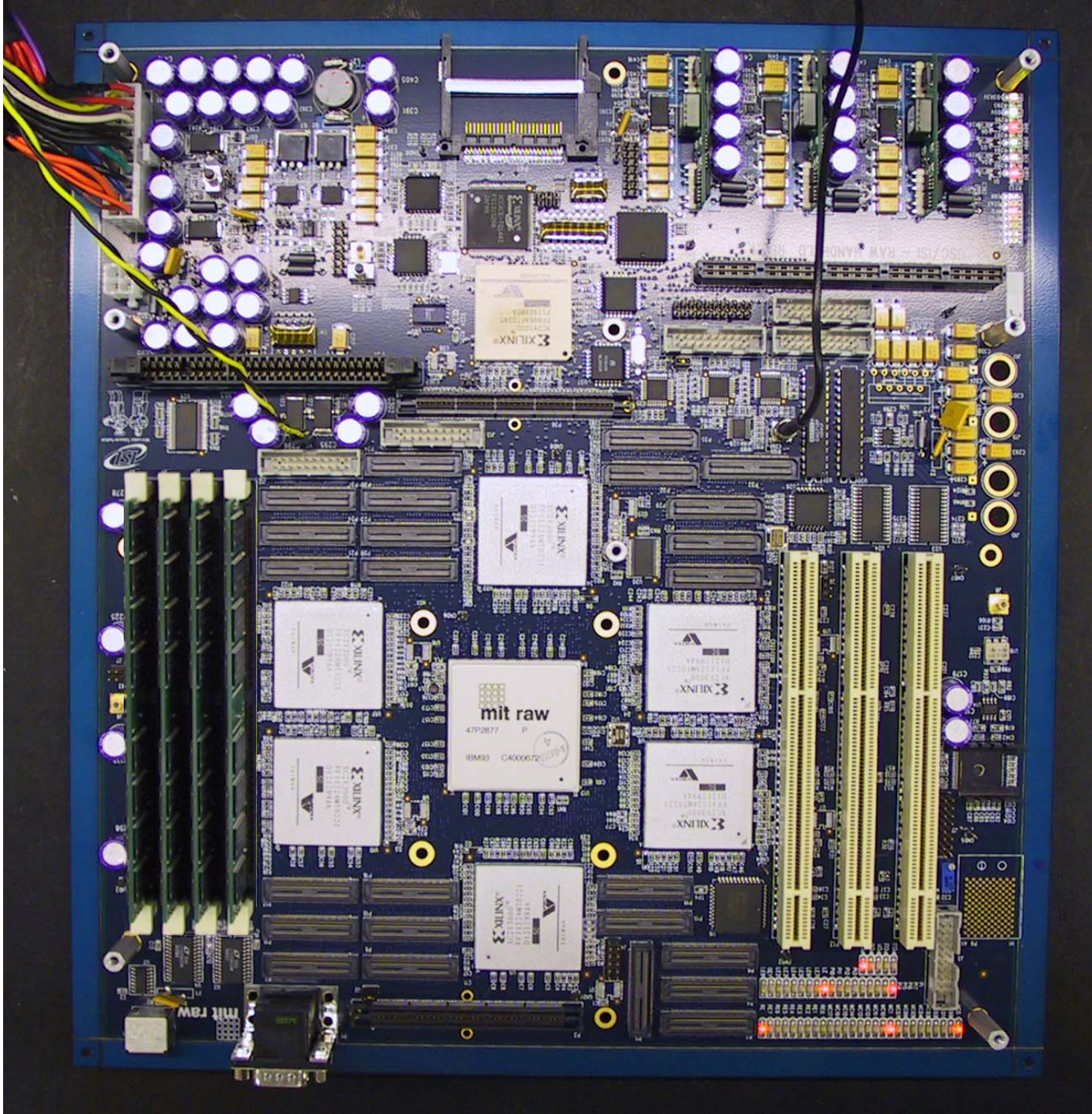
Longest wire, frequency, design and verification complexity all independent of issue width.

Architecture is backwards compatible.

Exploration Avenues for External Users

Prototype

Raw Prototype



Raw ASIC

IBM SA-27E .15u 6L Cu

18.2 mm x 18.2 mm

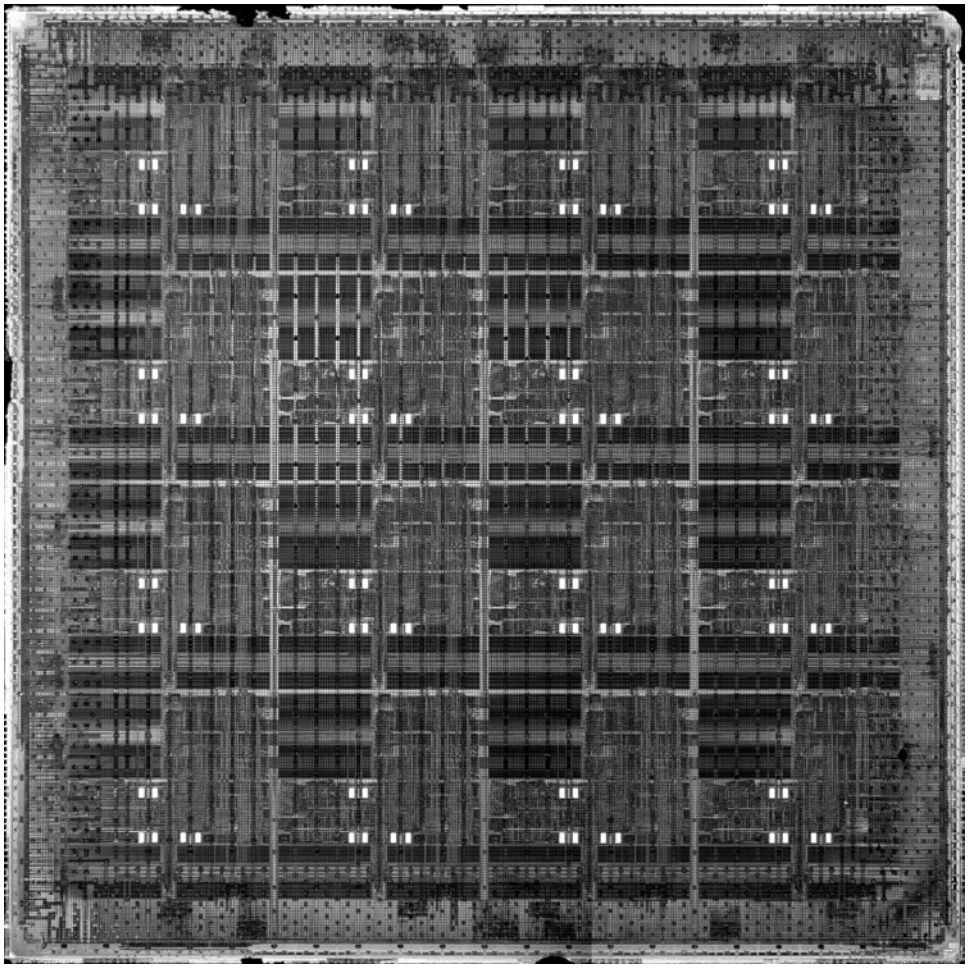
16 Flops/ops per cycle

208 Operand Routes / cycle

2048 KB L1 SRAM

1657 Pin CCGA Package

1080 HSTL core-speed signal I/O



@ 225 MHz Worst Case (Temp, Vdd, process):

3.6 Peak GFLOPS (without FMAC)
230 Gb/s on-chip bisection bandwidth
201 Gb/s off-chip I/O bandwidth

Exploration Avenues for External Users

```
Prototype       +   real world experience, fast
                +   practical apps
                -   static configuration
                      # tiles, I/O & memory ratio

Simulator
```

Exploration Avenues for External Users

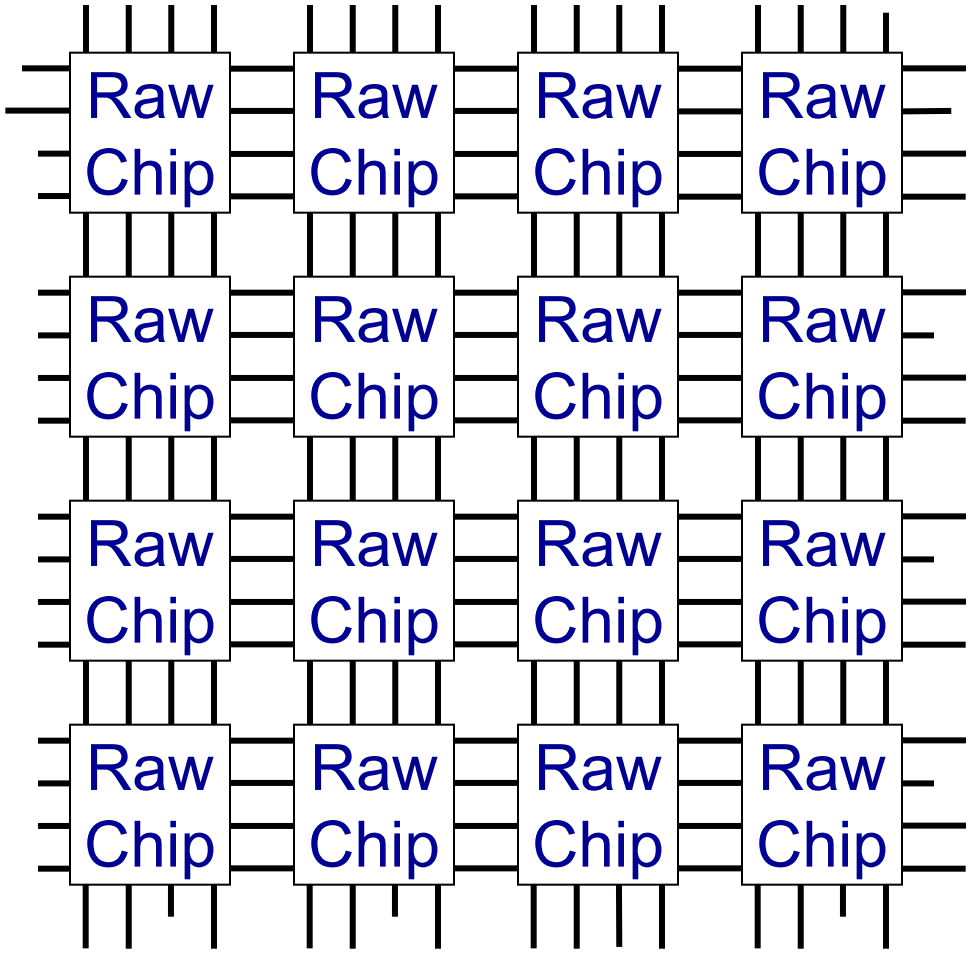
Prototype
+ real world experience, fast
+ practical apps
- static configuration
  # tiles, I/O & memory ratio

Simulator
- "magic", slow simulation speed
+ scalability exploration
  (64 tiles is feasible in 90nm)
  lots of flexibility in
  I/O and memory experimentation

Raw Fabric

Raw chips gluelessly connect to form larger virtual chips up to 32x32 tiles.

This 16 chip array would approximate a 256 tile Raw from a 45 nm process.



Exploration Avenues for External Users

Prototype	+	real world experience, fast
	+	practical apps
	-	static configuration
		# tiles, I/O & memory ratio
Simulator	-	"magic", slow simulation speed
	+	scalability exploration
		(64 tiles is feasible in 90nm)
		lots of flexibility in
		I/O and memory experimentation
Raw Fabric	+	positive aspects of both
	-	not exact simulation of future
	-	$$$, not avail. yet

Outline

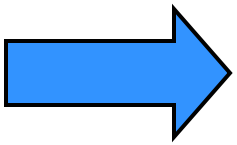
Raw Architectural Overview

Compute Processor

On-chip networks

Architectural Usage hints

Raw exploration avenues



End comments

Ultimate Guide: Raw Specification

Definitely need to read it to get the best performance.

Chapters 1-14 “Raw Architecture Manual”

Read this to understand Raw’s components, how they work, and how to program them.

In a few cases, later sections supercede earlier ones.

Ignore the constants – they’ve changed over time.

Chapters 15 “Prototype Specific Data”

Updated continuously.

Ultra dense presentation of constants and implementation.