# Octavo: An FPGA-Centric Processor Architecture

Charles Eric LaForest
J. Gregory Steffan
ECE, University of Toronto

FPGA 2012, February 24

# Easier FPGA Programming

- We focus on overlay architectures
  - Nios, MicroBlaze, Vector Processors
    - These inherited their architectures from ASICs
  - Easy to use with existing software tools
  - Performance penalty
  - ASIC architectures poor fit to FPGA hardware!
- ASIC ≠ FPGA
  - ASIC: transistors, poly, vias, metal layers
  - FPGA: LUTs, BRAMs, DSP Blocks, routing
    - Fixed widths, depths, other discretizations

FPGA-centric processor design?

# How do FPGAs Want to Compute?

| Hardware (Stratix IV) | Width (bits) | Fmax (MHz) |
|---|---|---|
| DSP Blocks | 36 | 480 |
| Block RAMs | 36 | 550 |
| ALUTs | 1 | 800 |
| Nios II/f | 32 | 230 |

***What processor architecture best fits the underlying FPGA?***

# Research Goals

1. Assume threaded data parallelism
2. Run at maximum FPGA frequency
3. Have high performance
4. Never stall
5. Aim for simple, minimal ISA
6. Match architecture to underlying FPGA

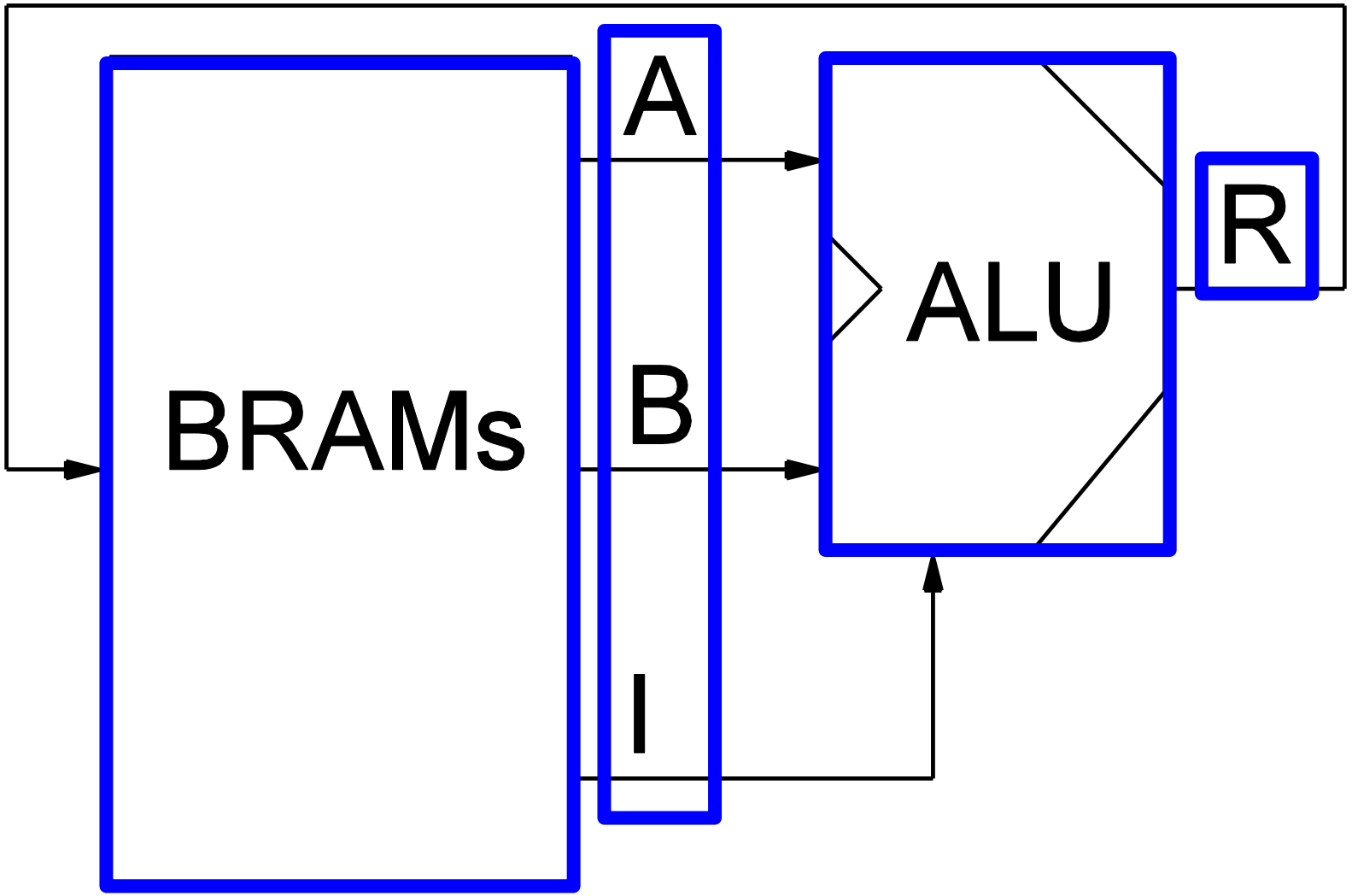
# Result: **Octavo**

- 10 stages, 8 threads, 550 MHz
- Family of designs
  - Word width (8 to 72 bits)
  - Memory depth (2 to 32k words)
  - Pipeline depth (8 to 16 stages)

Snapshot of work-in-progress

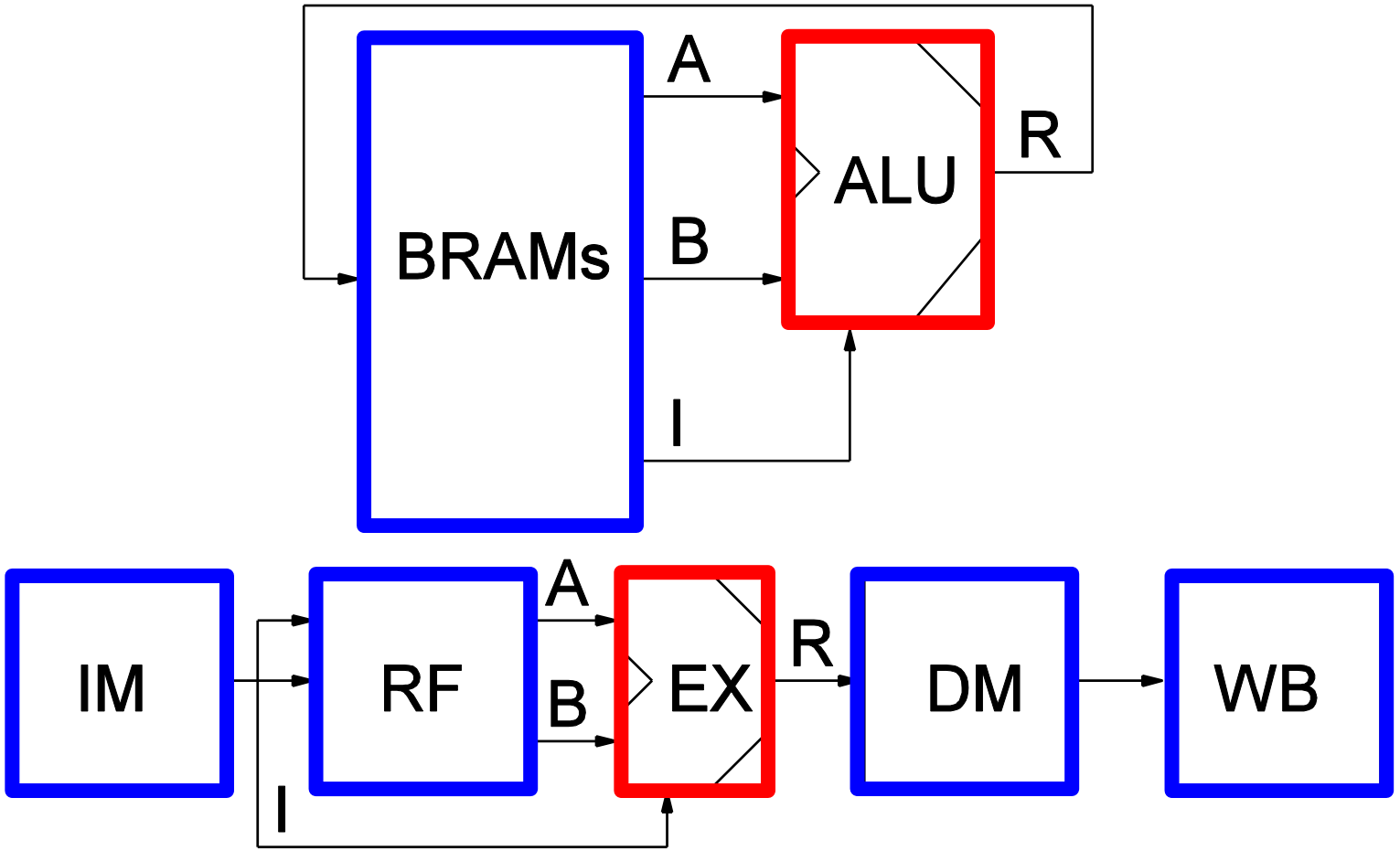
# Designing Octavo

# High-Level View of Octavo




Unified registers and RAM

# Octavo vs. Classic RISC




- All memories unified (no loads/stores)
- How to pipeline Octavo?

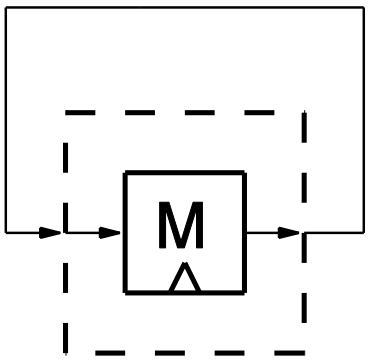
# Design For Speed: Self-Loop Characterization

# Self-Loop Characterization

- Connect module outputs to inputs
  - Accounts for the FPGA interconnect
- Pipeline loop paths to absorb delays
- Pointed to other limits than raw delay
  - Minimum clock pulse widths
    - DSP Blocks: 480 MHz
    - BRAMs: 550 MHz

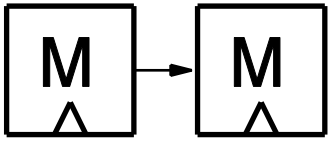
We measured some surprising delays…

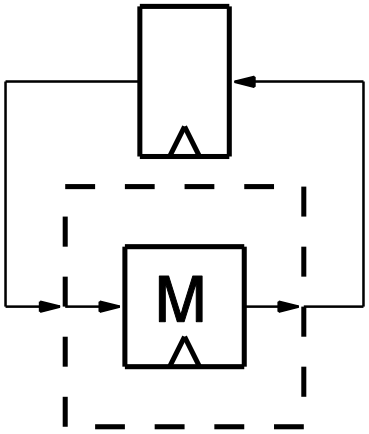
# BRAM Self-Loop Characterization



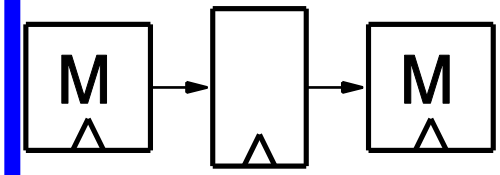

398 MHz

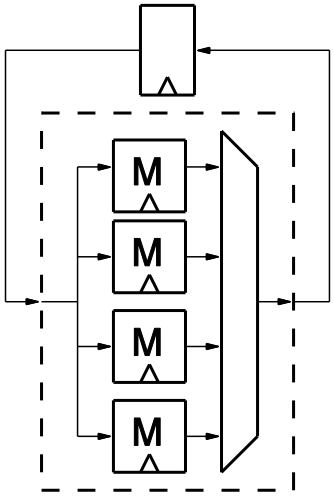
(routing!)



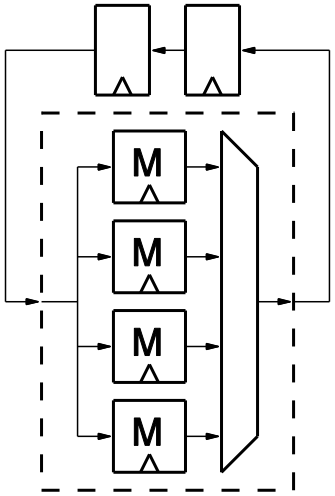




656 MHz

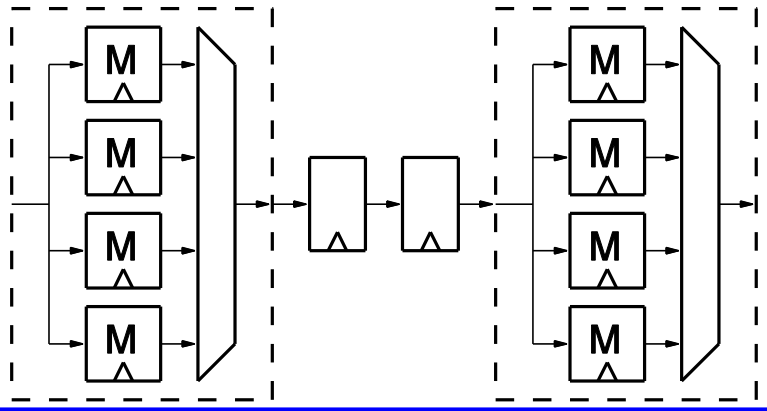



531 MHz



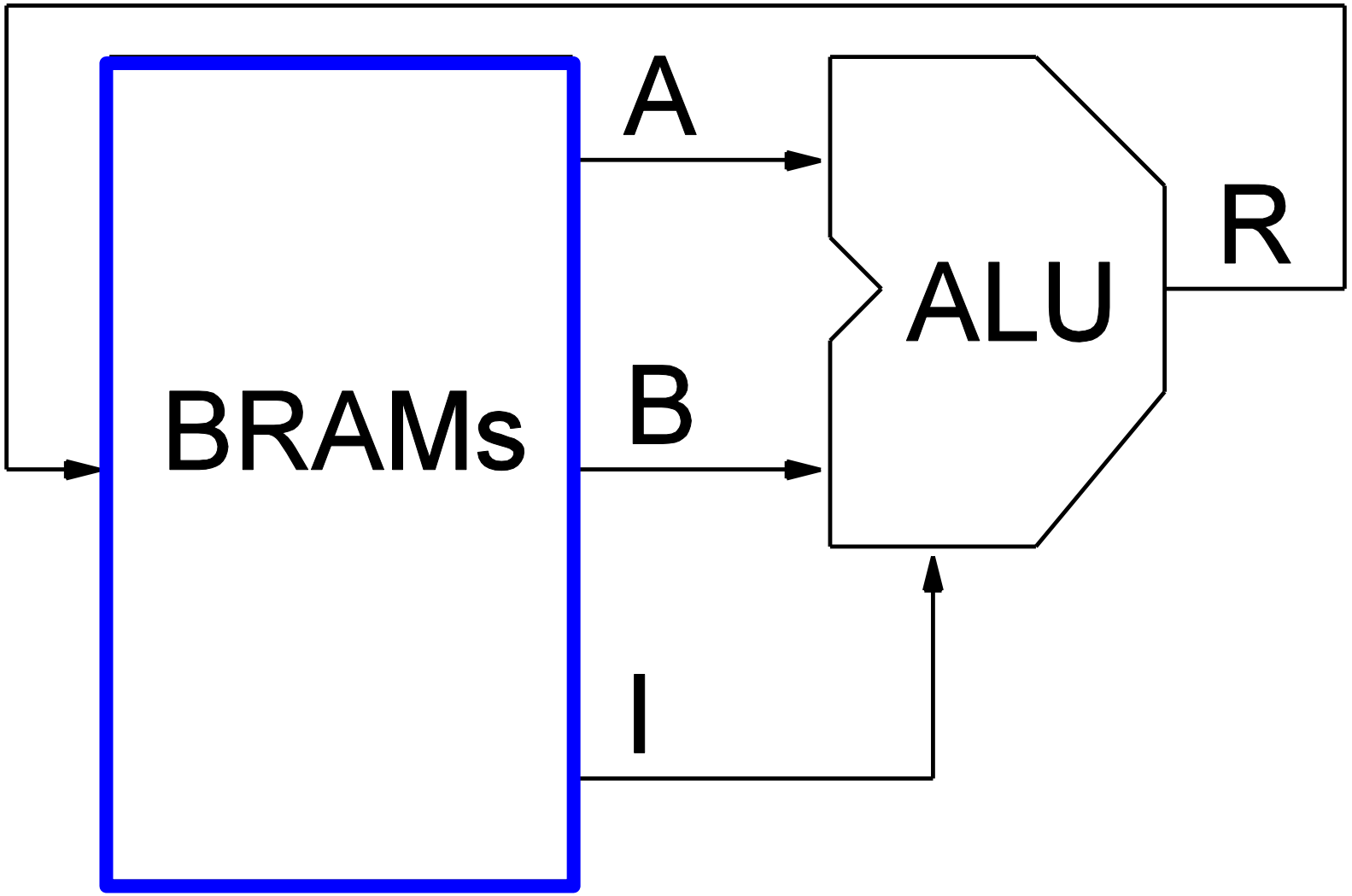

710 MHz



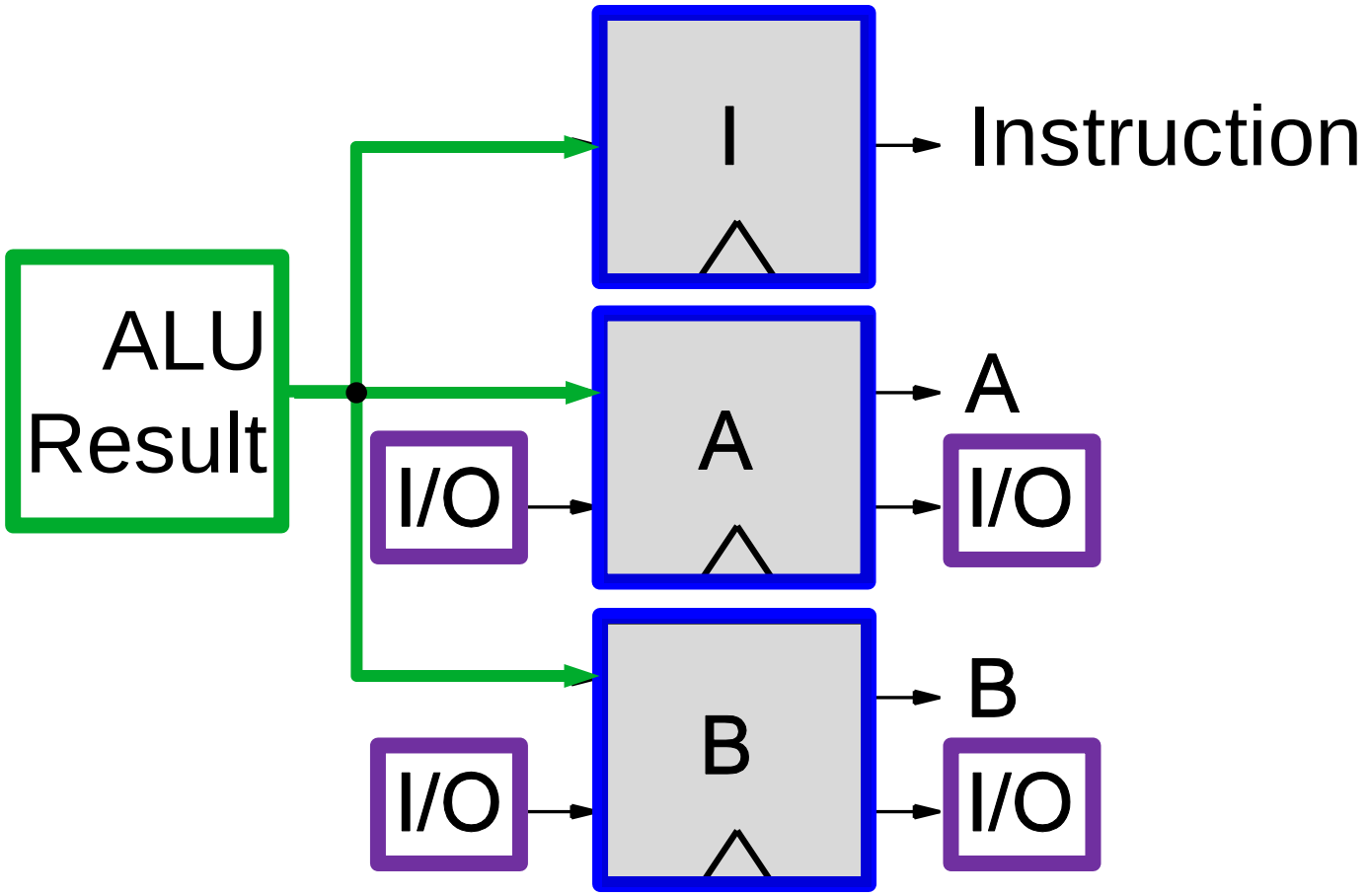

Must connect BRAMs using registers

# Building Octavo: Memory

# Building Octavo: Memory



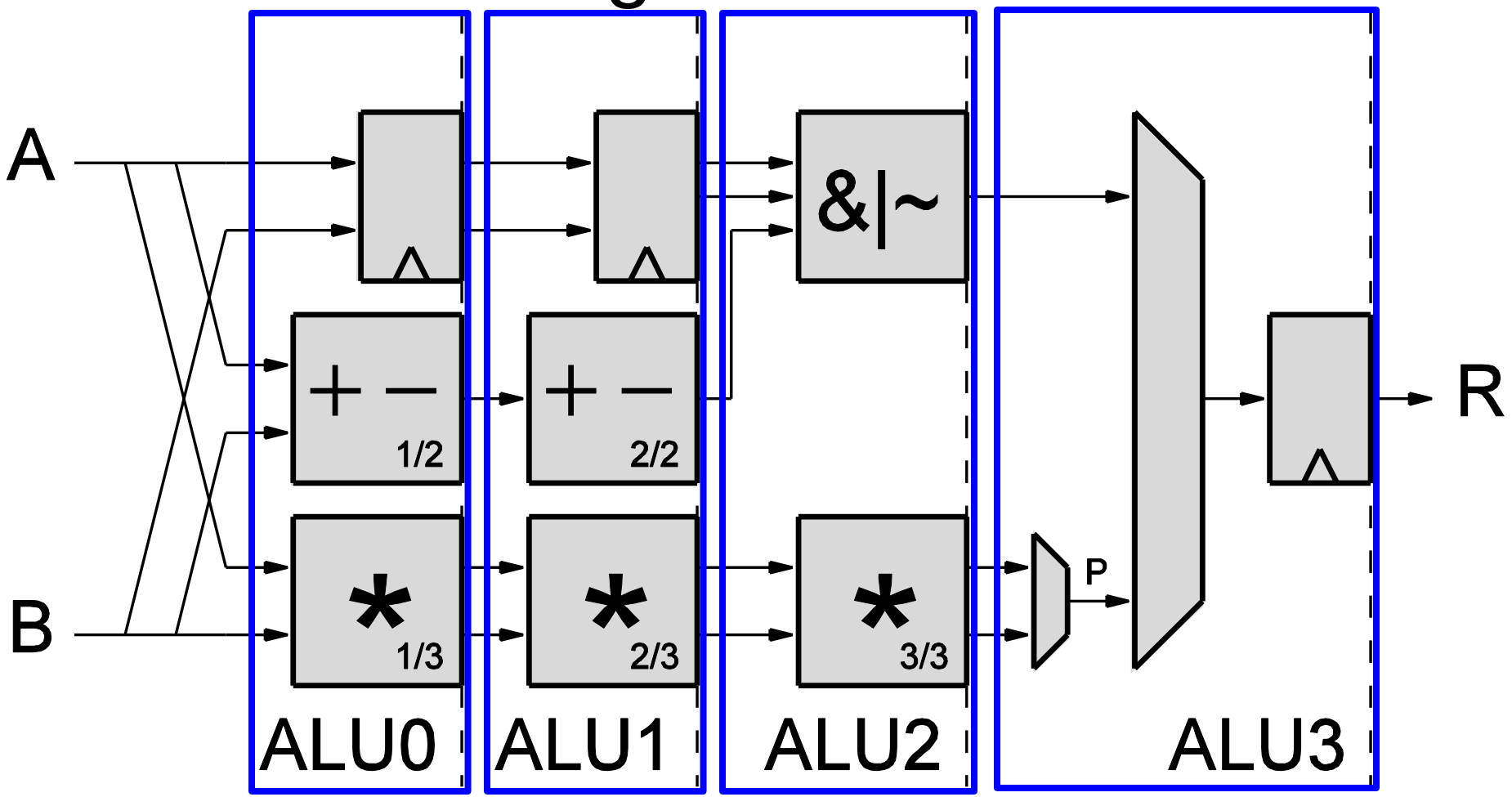
# Memory




Replicated “scratchpad” memories with I/O while still exceeding 550 MHz limit.

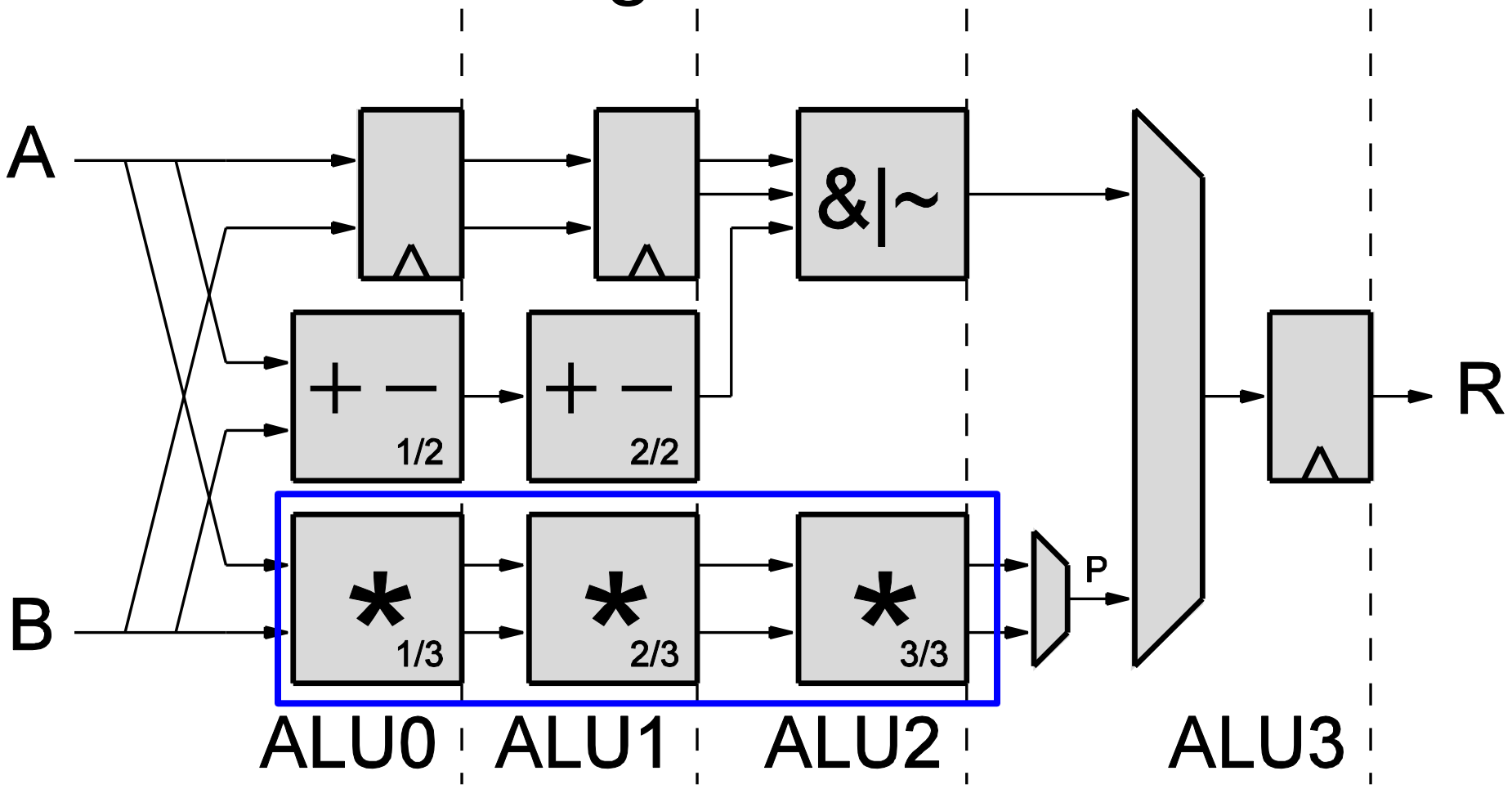
# Building Octavo: ALU

# Building Octavo: ALU




- Fully pipelined (4 stages)
  - Never stalls

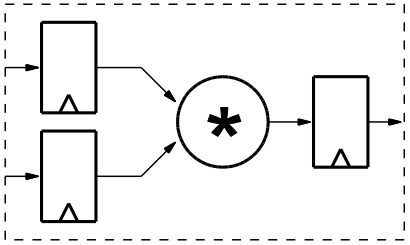
# Building Octavo: ALU




- Multiplication
  - Uses DSP Blocks
  - Must overcome their 480 MHz limit…

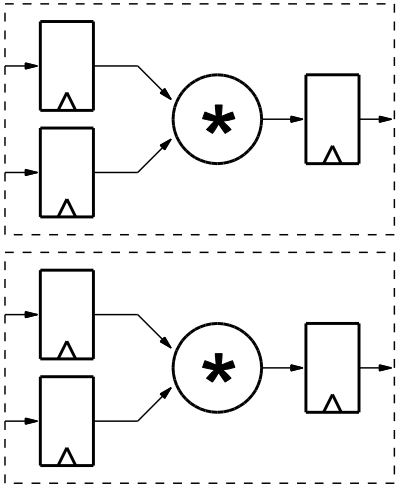
# Building Octavo: Multiplier

- One multiplier is wide enough but too slow




480 MHz

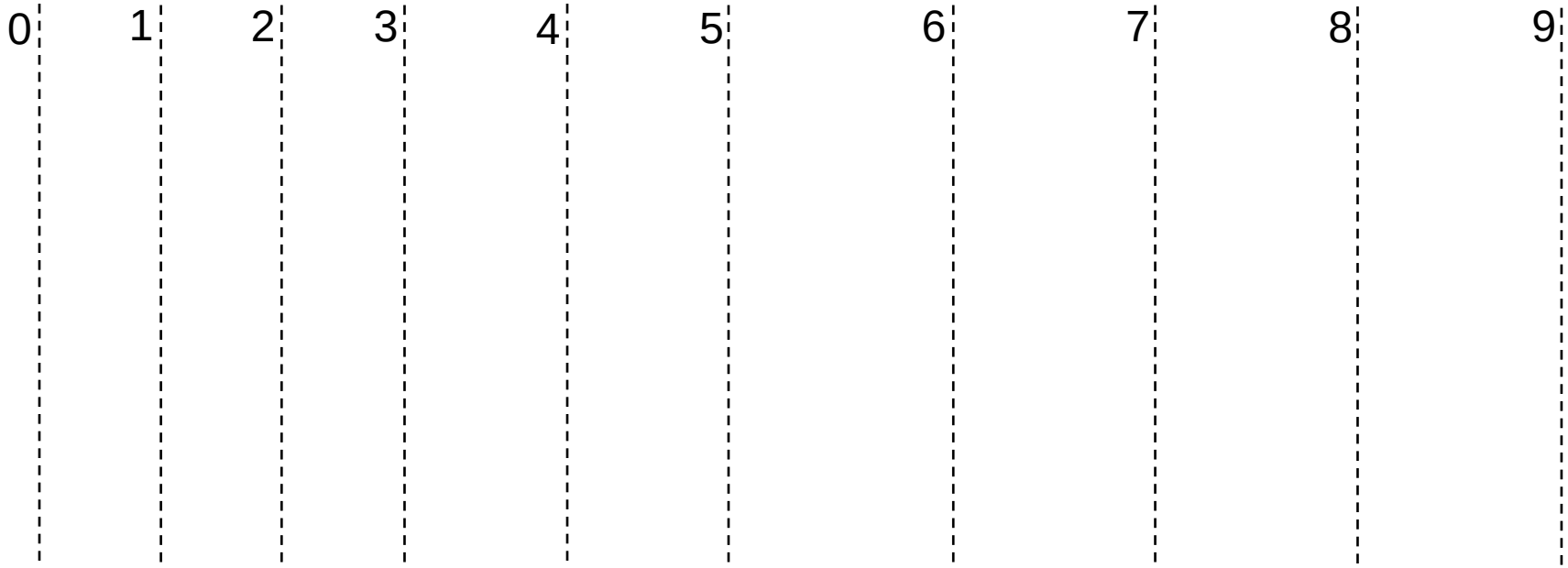
- Two multipliers working at half-speed
  - Send data to both multipliers in alternation




600 MHz

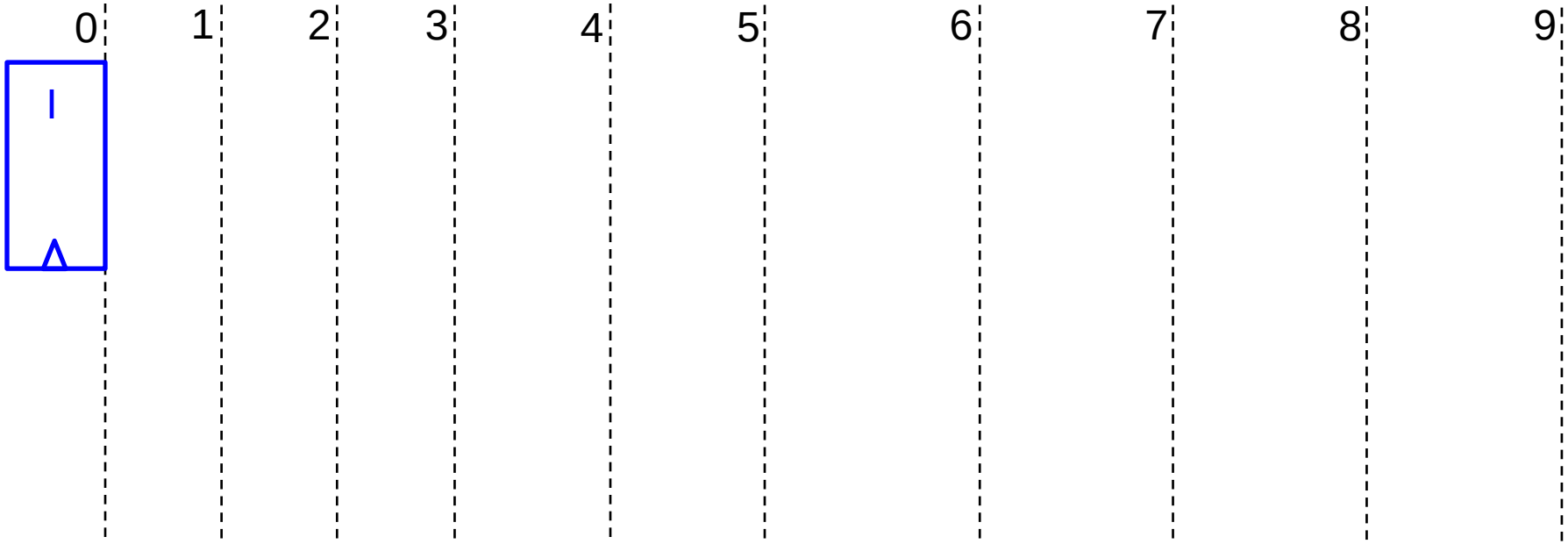
# Octavo: Putting It All Together

# Octavo



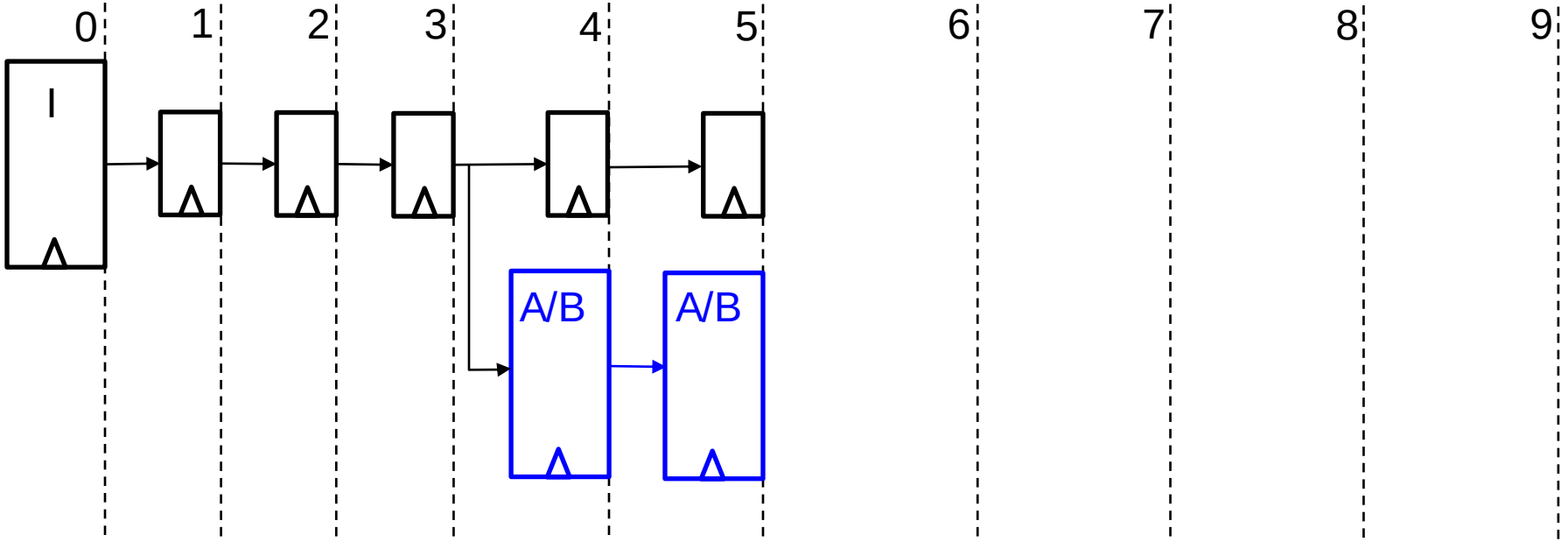

- Pipeline
  - 10 stages
    - Actually 8 stages with one exception (more later)
  - No result forwarding or pipeline interlocks
  - Scalar, Single-Issue, In-Order, Multi-Threaded

# Octavo



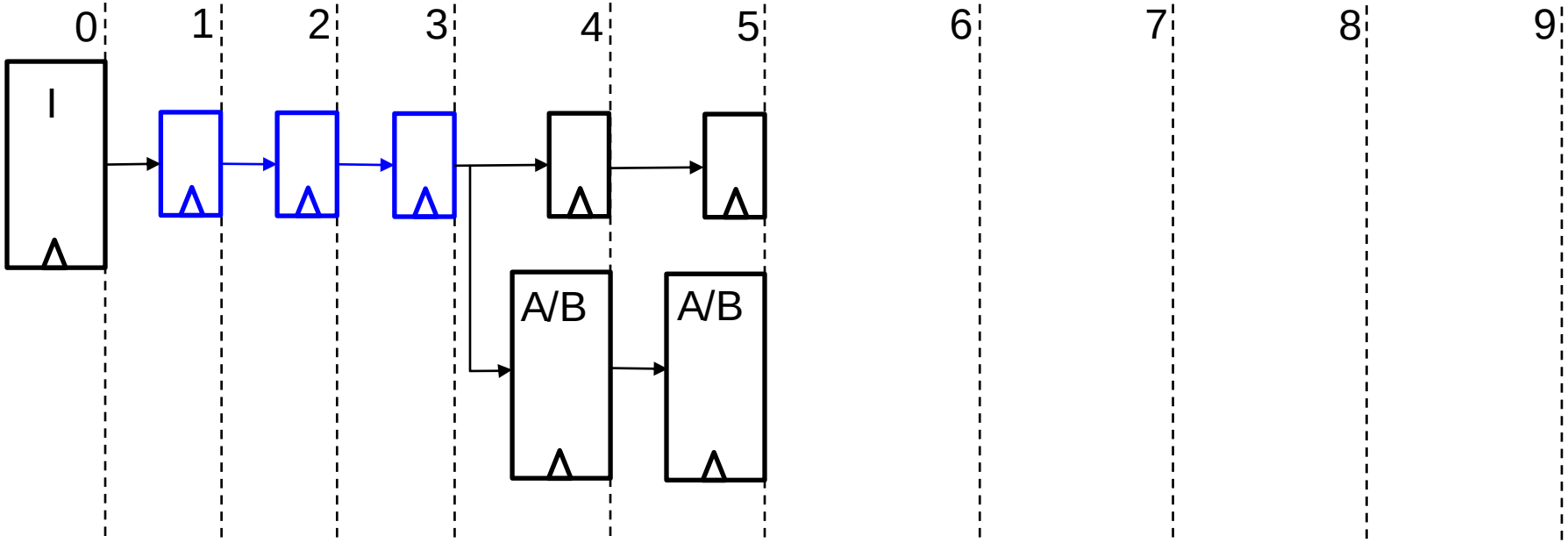

- Instruction Memory
  - Indexed by current thread PC
  - Provides a 3-operand instruction
  - On-chip BRAMs only

# Octavo



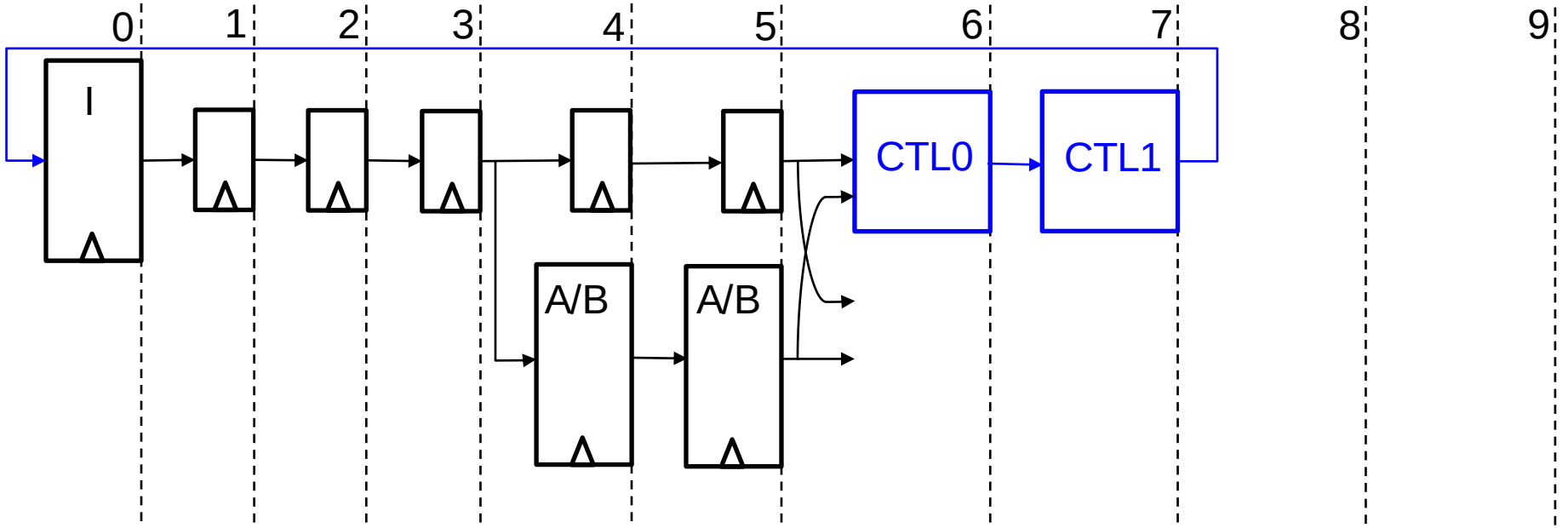

- A and B Memories
  - Receive operand addresses from instruction
  - Provide data operands to ALU and Controller
    - Some addresses map to I/O ports
  - On-chip BRAMs only

# Octavo



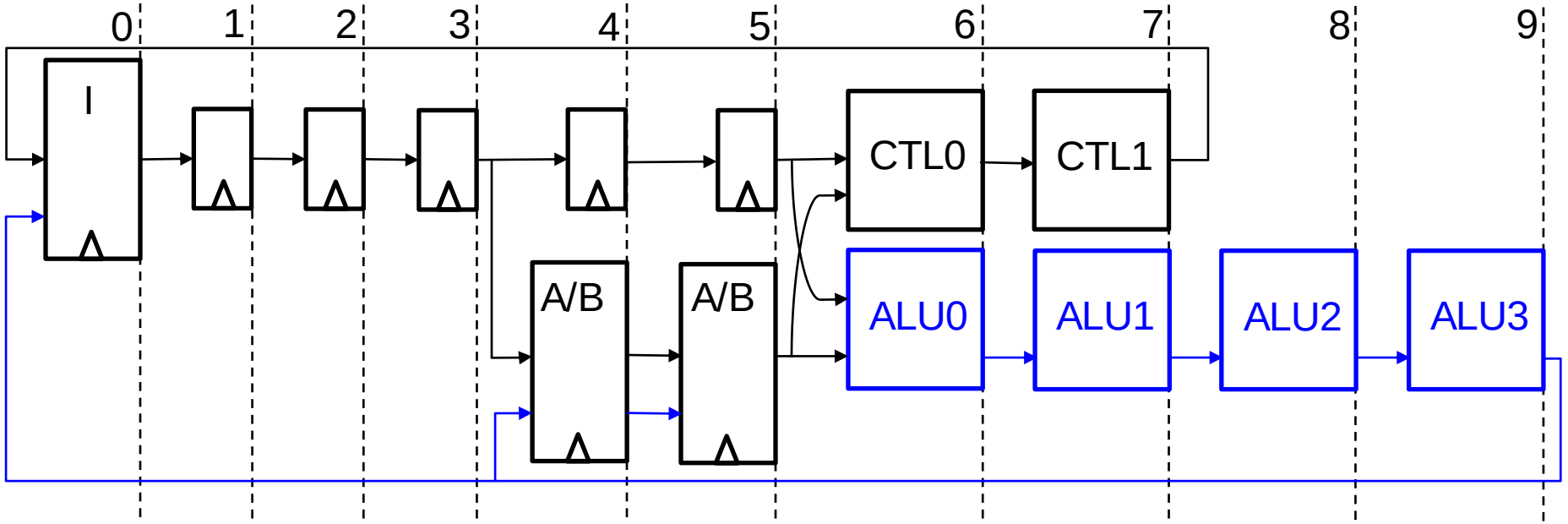

- Pipeline Registers
  - Avoid an odd number of stages
  - Separate BRAMs for best speed
    - Predicted by BRAM self-loop characterization
    - Unusual but essential design constraint

# Octavo



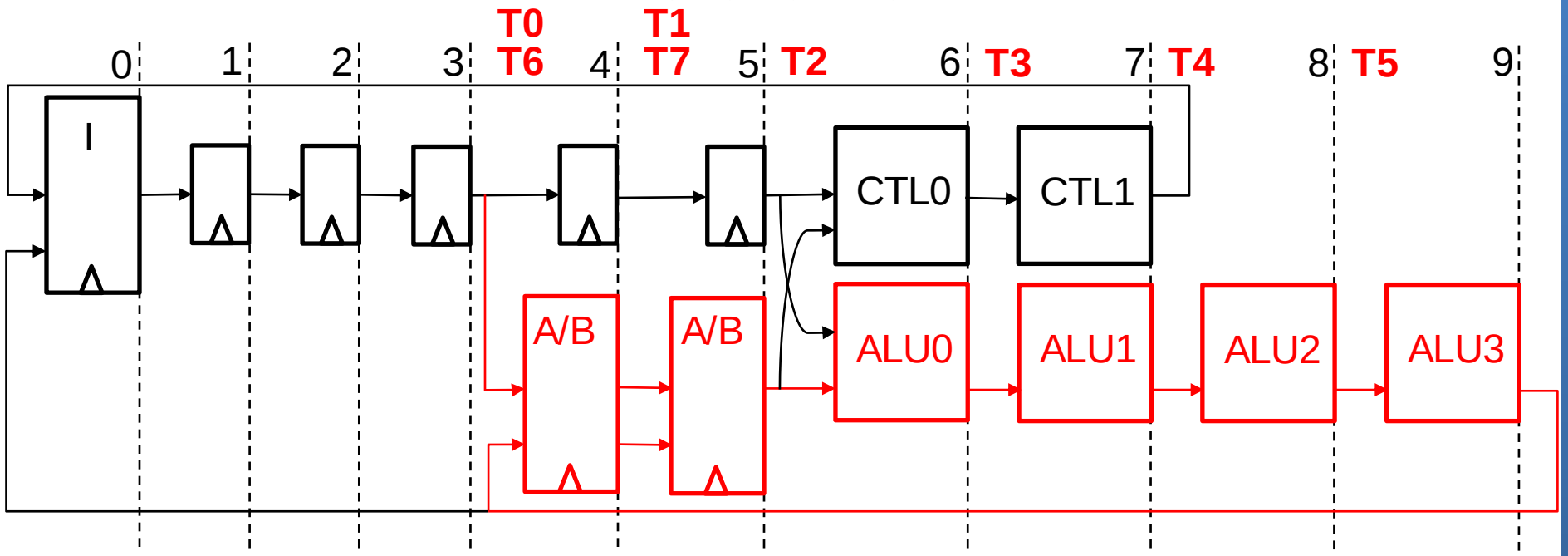

- Controller
  - Receives opcode, source/destination operands
  - Decides branches
  - Provides current PC of next thread to I memory

# Octavo



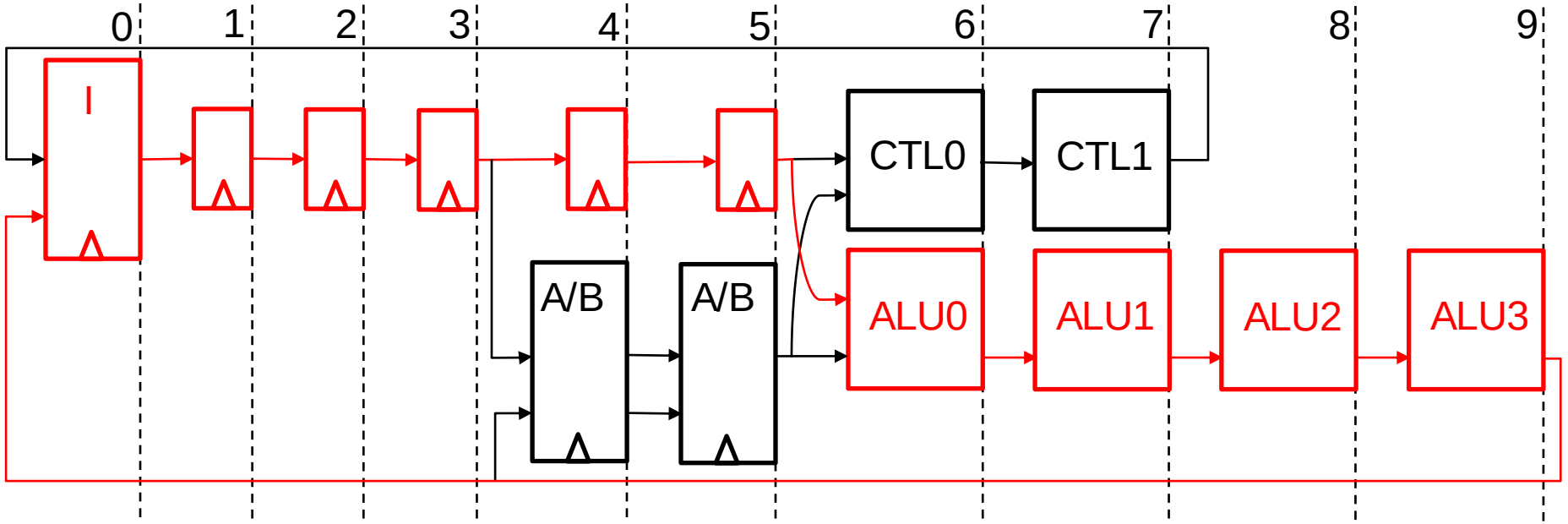

- ALU
  - Receives opcode and data
  - Writes result to all memories

# Octavo




- Longest mandatory loop: 8 stages
  - Along A/B memories and ALU
  - Fill with 8 threads to avoid stalls

# Octavo




- Special case longest loop: 10 stages
  - Along instruction memory and ALU
  - Does not affect most computations
    - Adds a delay slot to subroutine and loop code

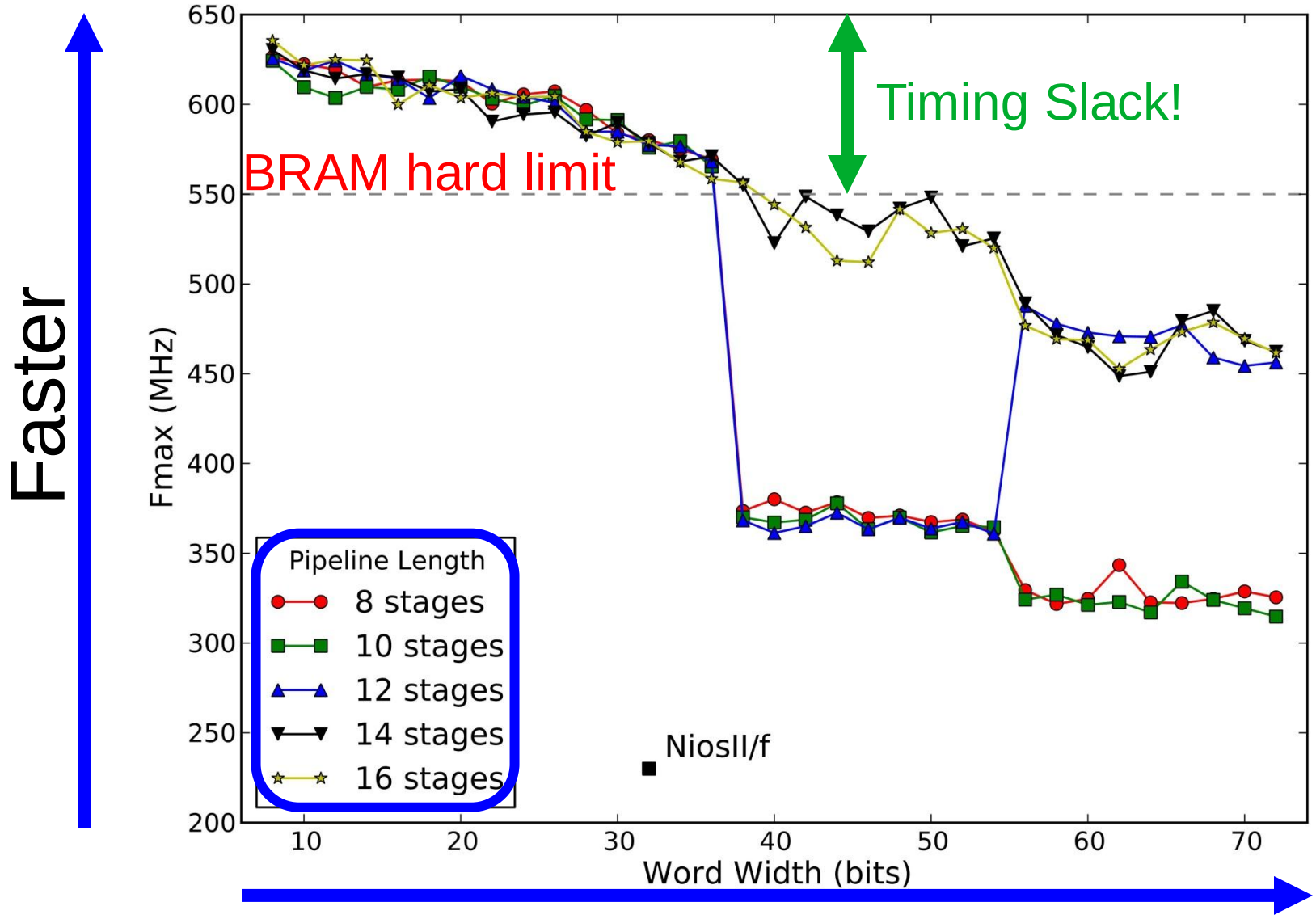
# Results: Speed and Area

# Experimental Framework

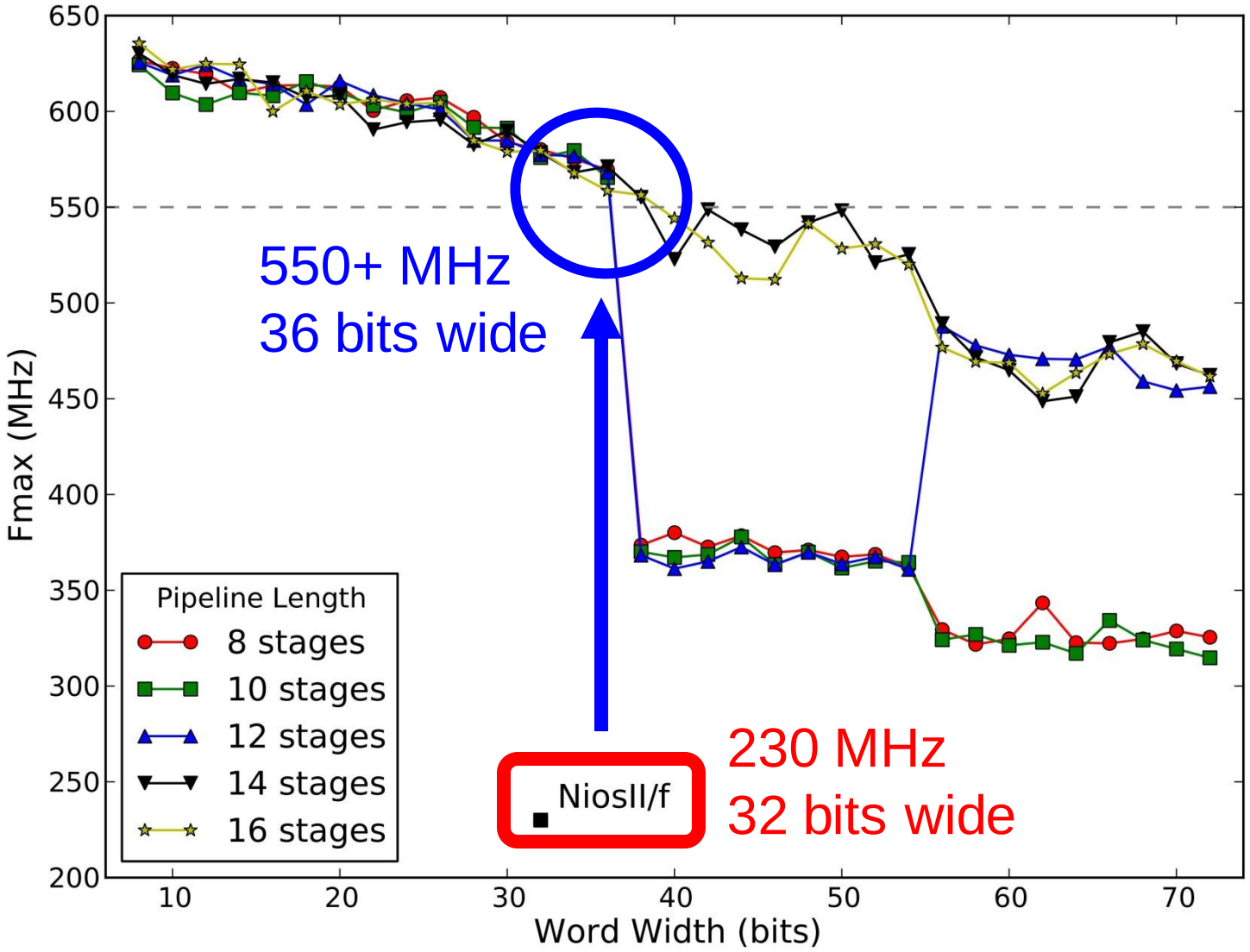
- Quartus 10.1 targeting Stratix IV (fastest)
  - Optimize and place for speed
  - Average speed over 10 placement runs
- Varied processor parameters:
  - Word width
  - Memory depth
  - Pipeline depth
- Measure Frequency, Area, and Density

# Maximum Operating Frequency

# Maximum Operating Frequency

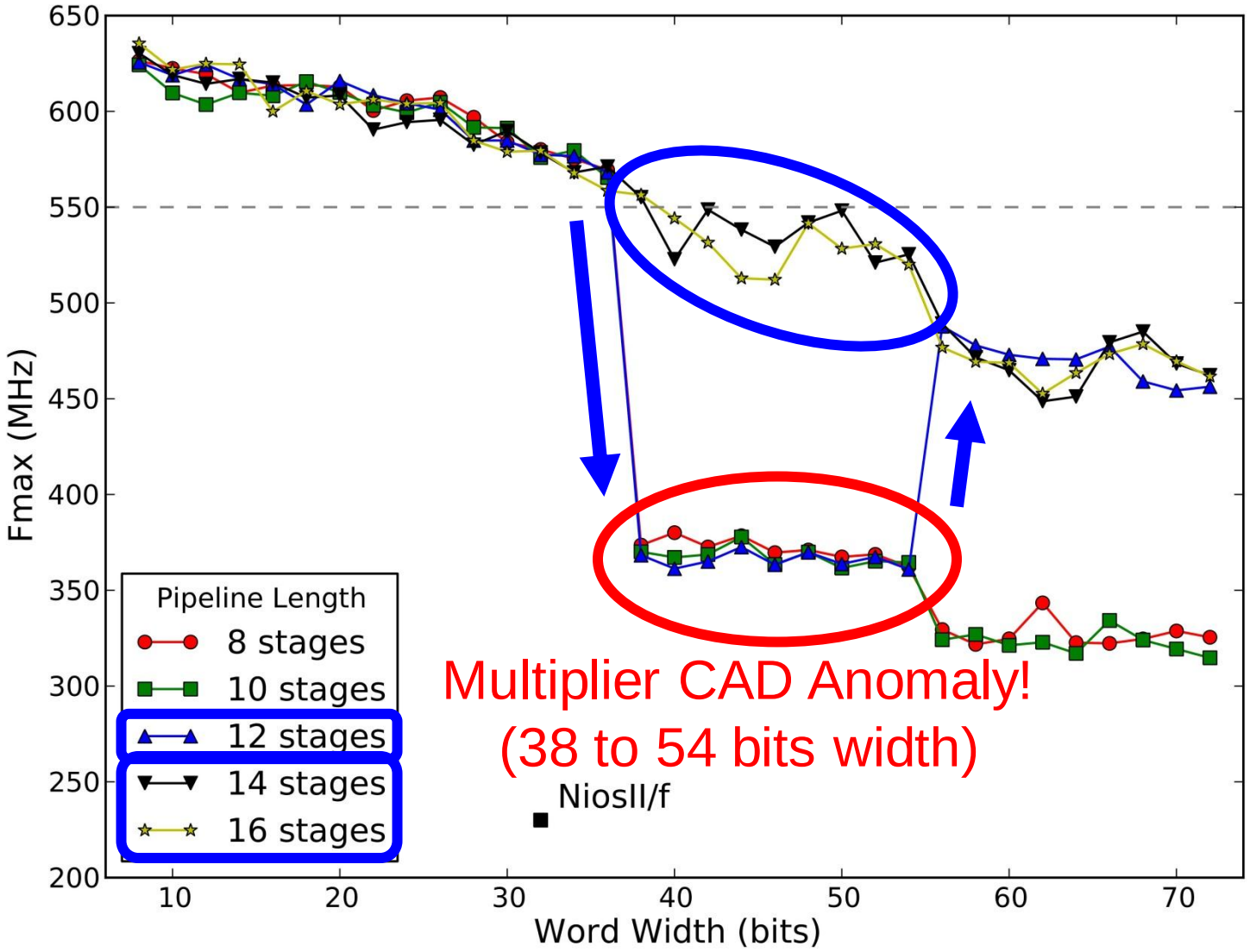


# Maximum Operating Frequency




2.39x faster, but not a fair comparison

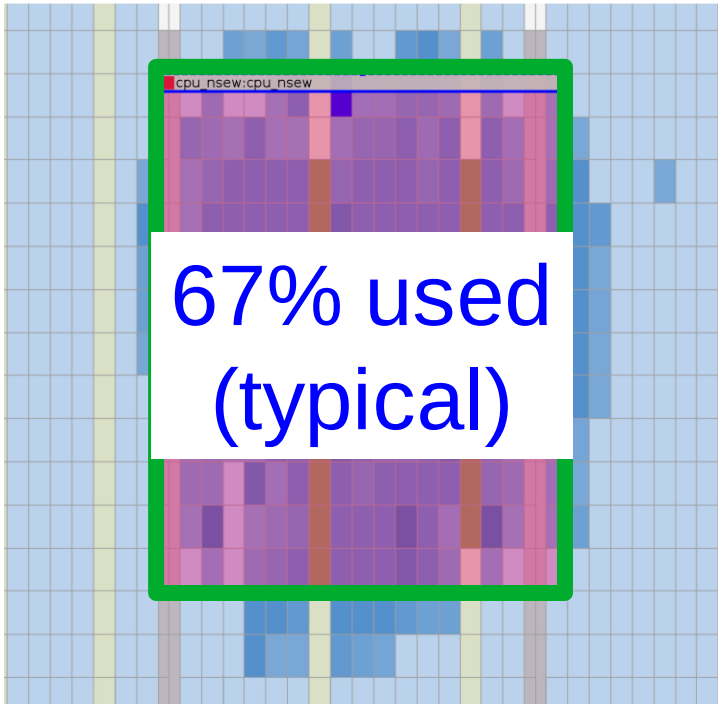
# Maximum Operating Frequency




Enough pipeline stages bury the inefficiency

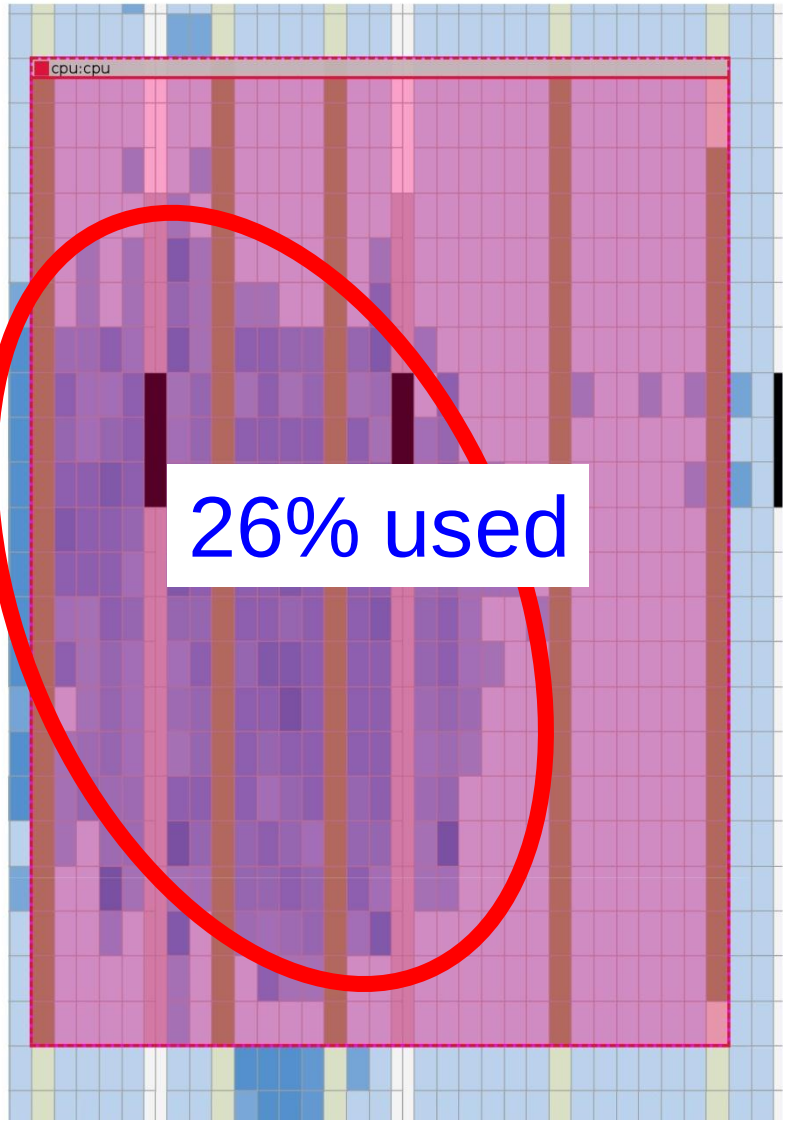
# Area Density

# Area Density




"Sweet spot"

72 bits, 1024 words




72 bits, 4096 words

# Designing Octavo: Lessons & Future Work

# Lessons

- Soft-processors can hit BRAM Fmax
  - Octavo: 8 threads, 10 stages, 550 MHz
- Self-loop characterization for modules
  - Helps reason about their pipelining
  - Shows true operating envelopes on FPGA
- Octavo spans a large design space
  - Significant range of widths, depths, stages

Consider FPGA-centric architecture!

# Future Work