LCS

MIT Laboratory for Computer Science

Banked Multiported Register Files for High-Frequency Superscalar Microprocessors

Jessica H. Tseng and Krste Asanović

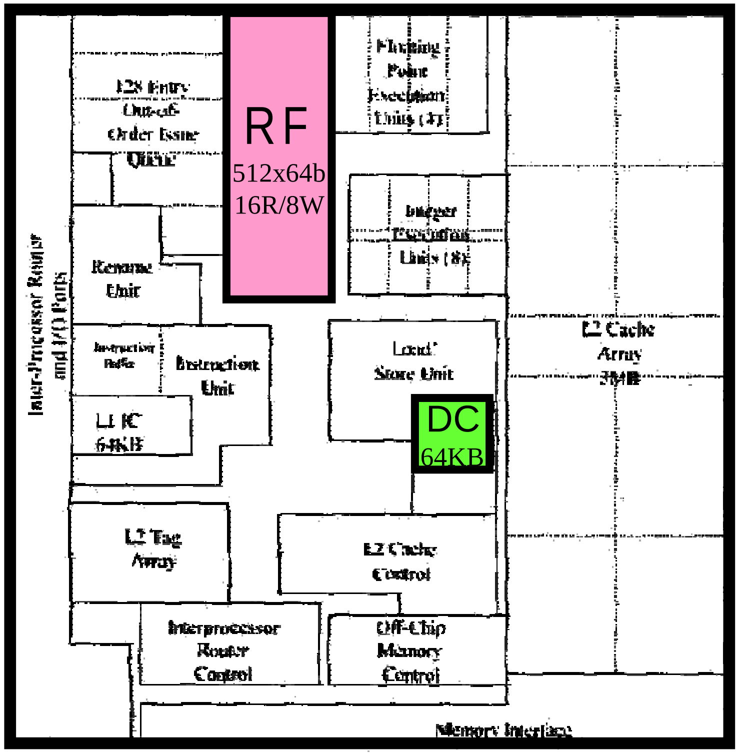
MIT Laboratory for Computer Science, Cambridge, MA 02139, USA

ISCA2003

Motivation

- Increasing demand on number of ports and number of registers in a register file.

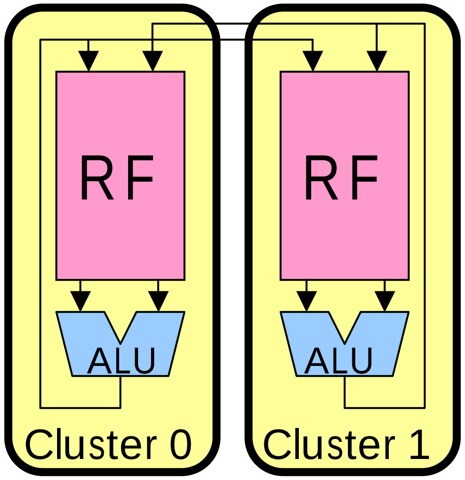
- Growing concerns in access time, power, and die area.
  - Example: Alpha 21464 register file (RF) occupied over **5X** the area of 64KB primary data cache (DC).



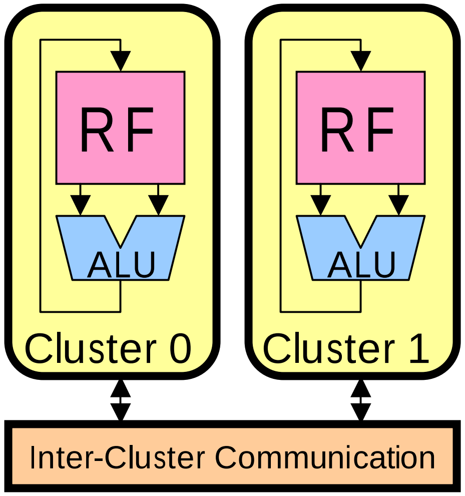
Alpha 21464 Floorplan
ISSCC, 2002

Distributed Architecture

- Duplicated
  - Fewer Read Ports
  - Same Number of Write Ports
  - Twice Total Number of Registers
  - Alpha 21264 & Alpha 21464

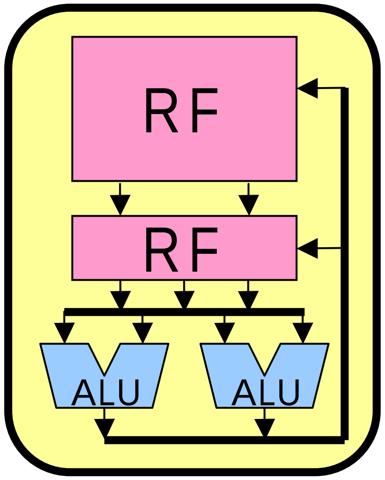


- Non-Duplicated
  - Fewer Read Ports
  - Fewer Write Ports
  - Complex Inter-Cluster Communication

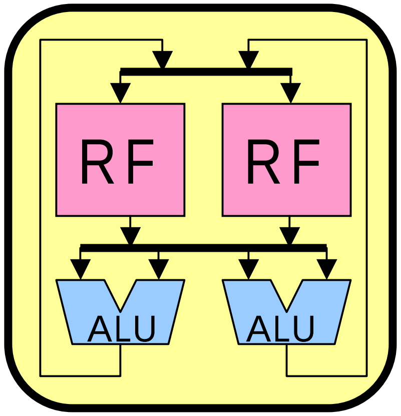


Centralized Architecture

- Multi-Level: Register File Cache
  - Fewer Read Ports
  - Fewer Write Ports
  - Control Logic Complexity
  - Poor Locality



- One-Level Multi-Banked
  - Fewer Read Ports
  - Fewer Write Ports
  - Possible Conflicts
  - Control Logic Complexity
  - Possible Pipeline Stalls



Previous Work

- Use minimal number of ports per register file banks: 1 or 2-read port(s) and 1-write port.

- Avoid issuing instructions that would cause register file read conflicts.
  - Add complexity to the critical wakeup-select loop for the issue logic → slower cycle time

- Resolve register file write conflicts by either delaying physical register allocation until write back stage or installing write buffers.
  - Complex pipeline control logic
  - Possible pipeline stalls

Our Work

- Use more ports per register file bank: 2-read ports and 2-write ports.
- Speculatively issue potentially conflicting instructions.
  - Minimize impact to the critical wakeup-select loop for the issue logic
- Rapidly repair pipeline and reissue conflicting instructions when conflicts are detected after issue.
  - No write buffer requirement
  - No pipeline stalls

Simpler and Faster Control Logic

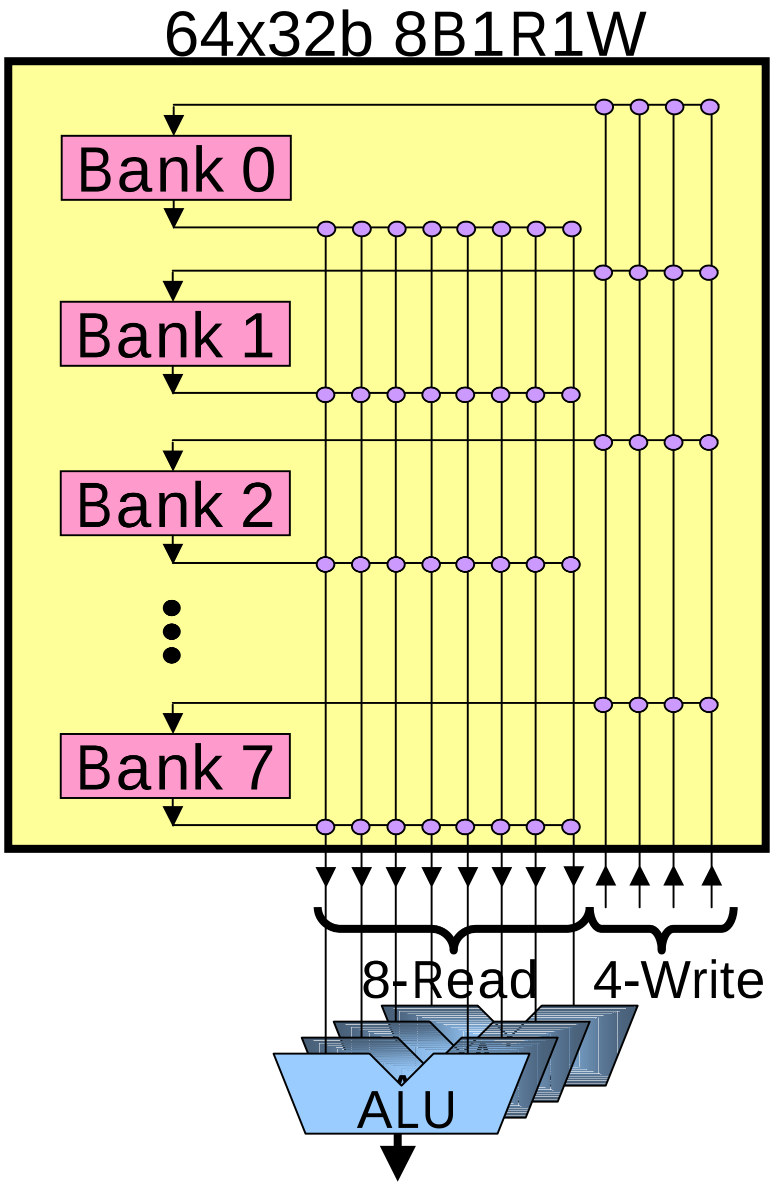
Example

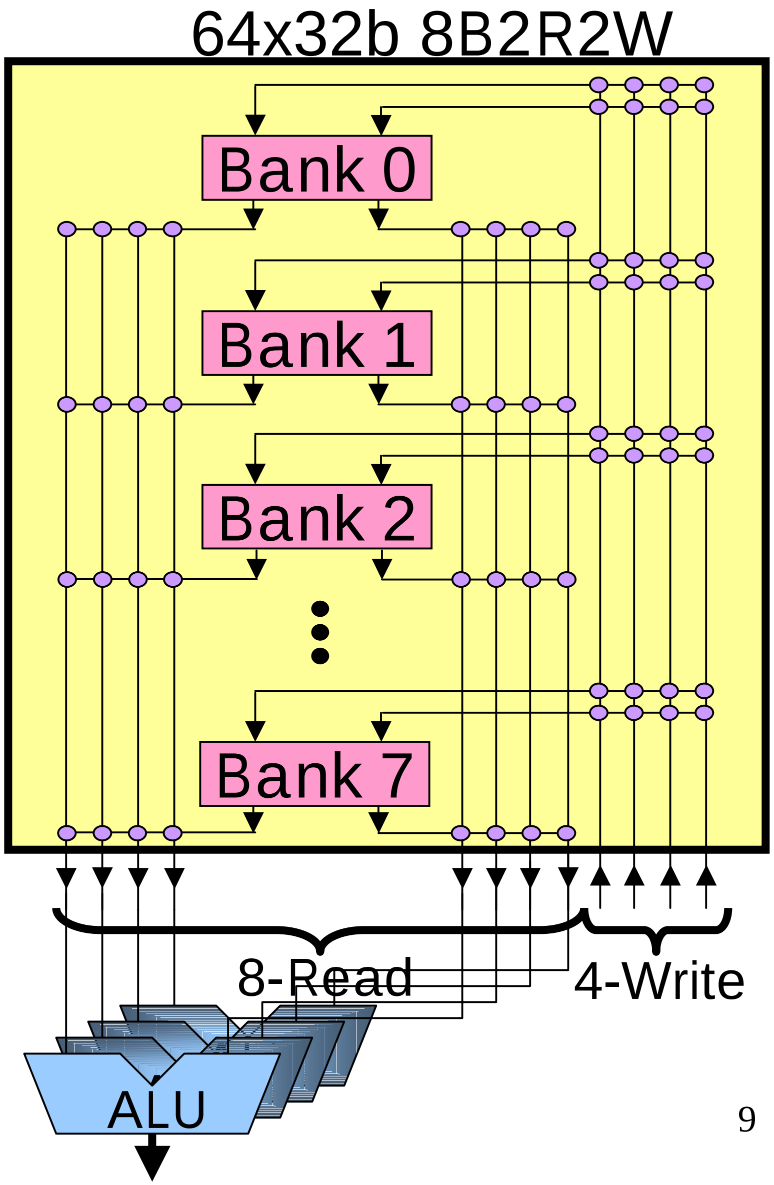
- Four-issue superscalar machine with a 64x32b 8-banked register file.
  - Area Saving: 63%
  - Access Time Reduction: 25%
  - Energy Reduction: 40%
  - IPC Degradation: < 5%

Outline

1. Banked Register File Structure
2. Basic Pipeline Structure and Control Logic
3. Improving IPC
   - Bypass Skip
   - Read Sharing
4. Conclusion

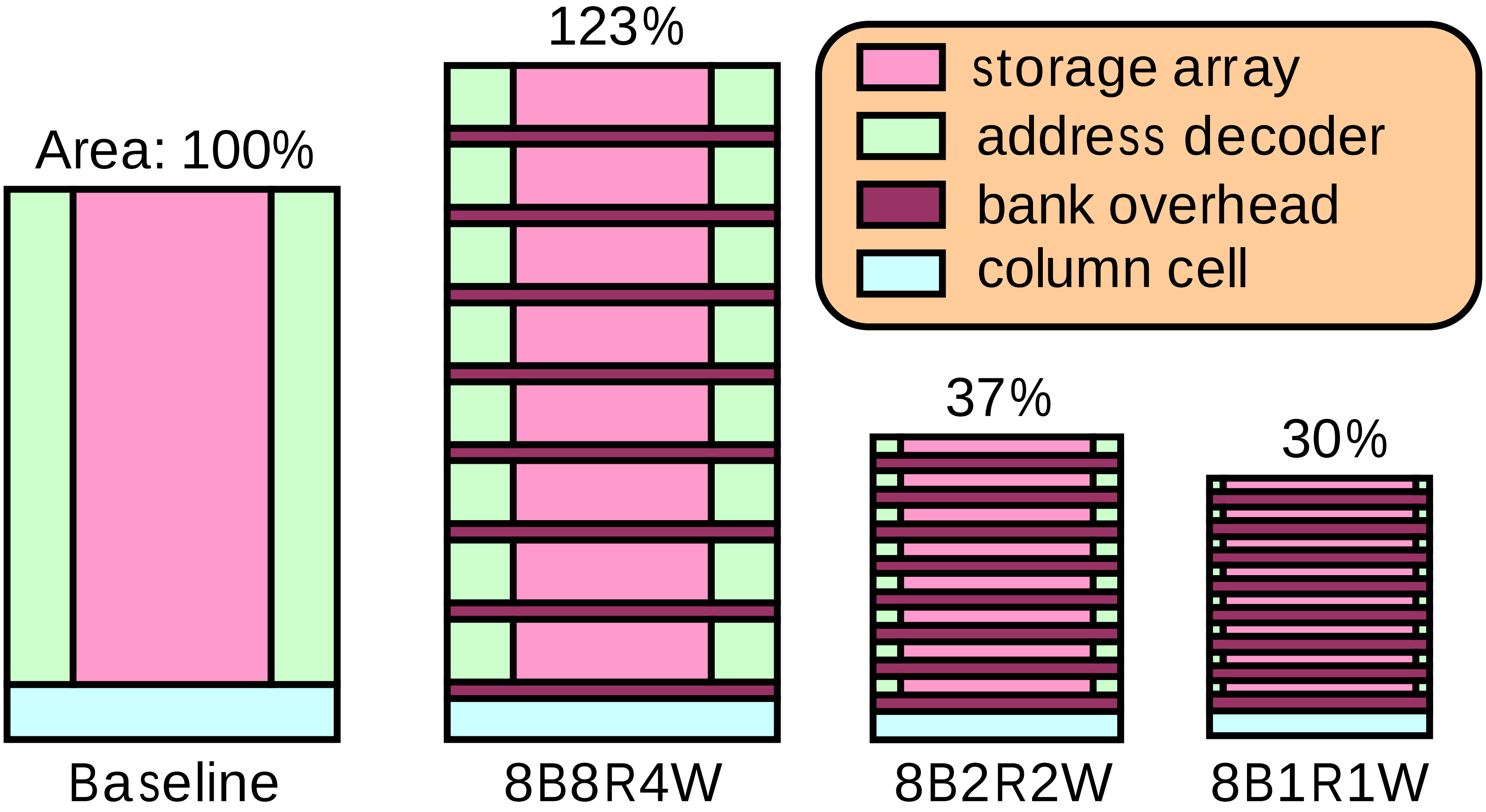
Banked Register File Structure





Register File Floorplan

64x32b 8-Read Ports & 4-Write Ports



Area: 100%

123%

37%

30%

Baseline

8B8R4W

8B2R2W

8B1R1W

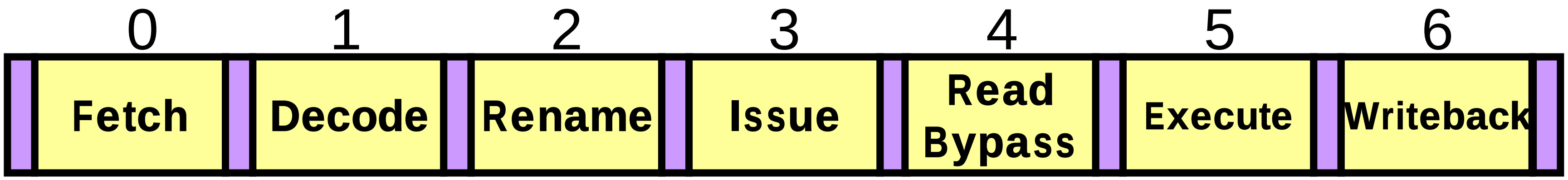
🟪 storage array

🟩 address decoder

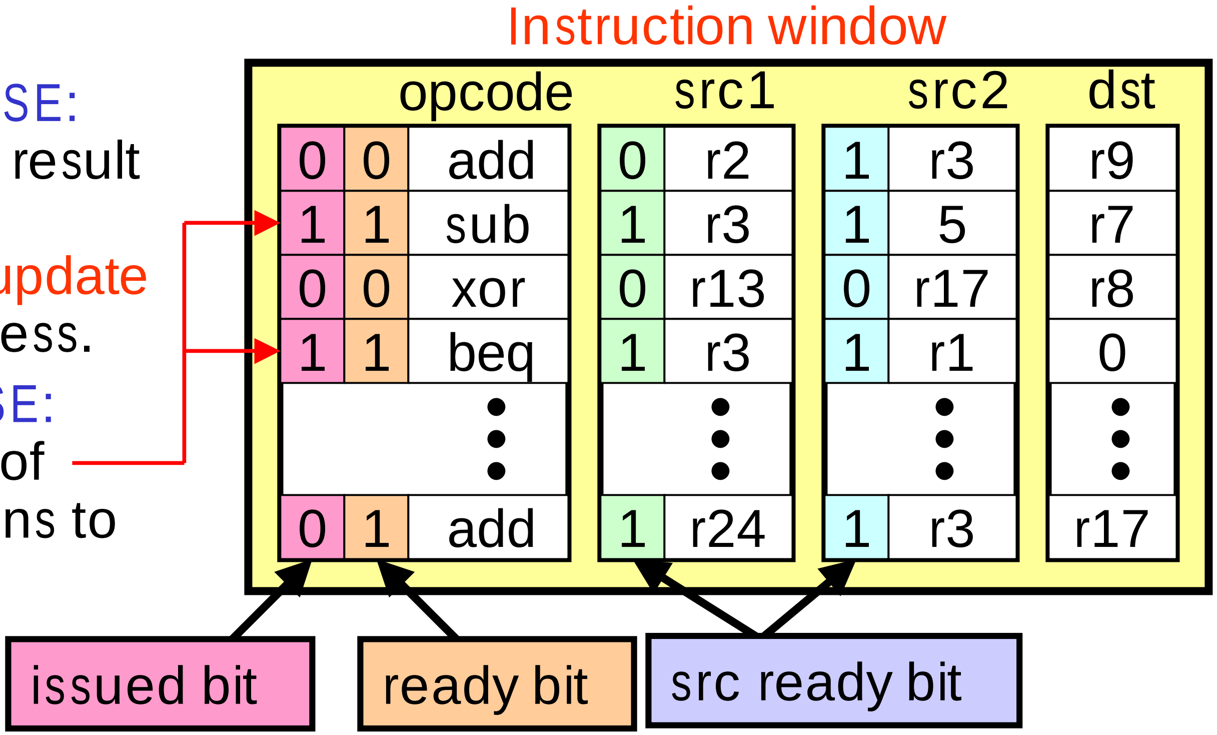
🟥 bank overhead

🟦 column cell

Baseline Pipeline Structure

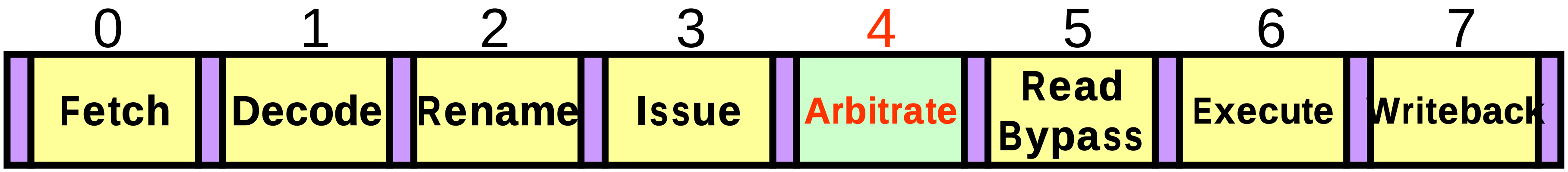


- Issue
  - WAKEUP PHASE: Broadcasts the result tags of issued instructions to update operand readiness.
  - SELECT PHASE: Picks a subset of ready instructions to issue.



		opcode		src1		src2	dst
0	0	add	0	r2	1	r3	r9
1	1	sub	1	r3	1	5	r7
0	0	xor	0	r13	0	r17	r8
1	1	beq	1	r3	1	r1	0
		⋮		⋮		⋮	⋮
0	1	add	1	r24	1	r3	r17

Modified Pipeline Structure

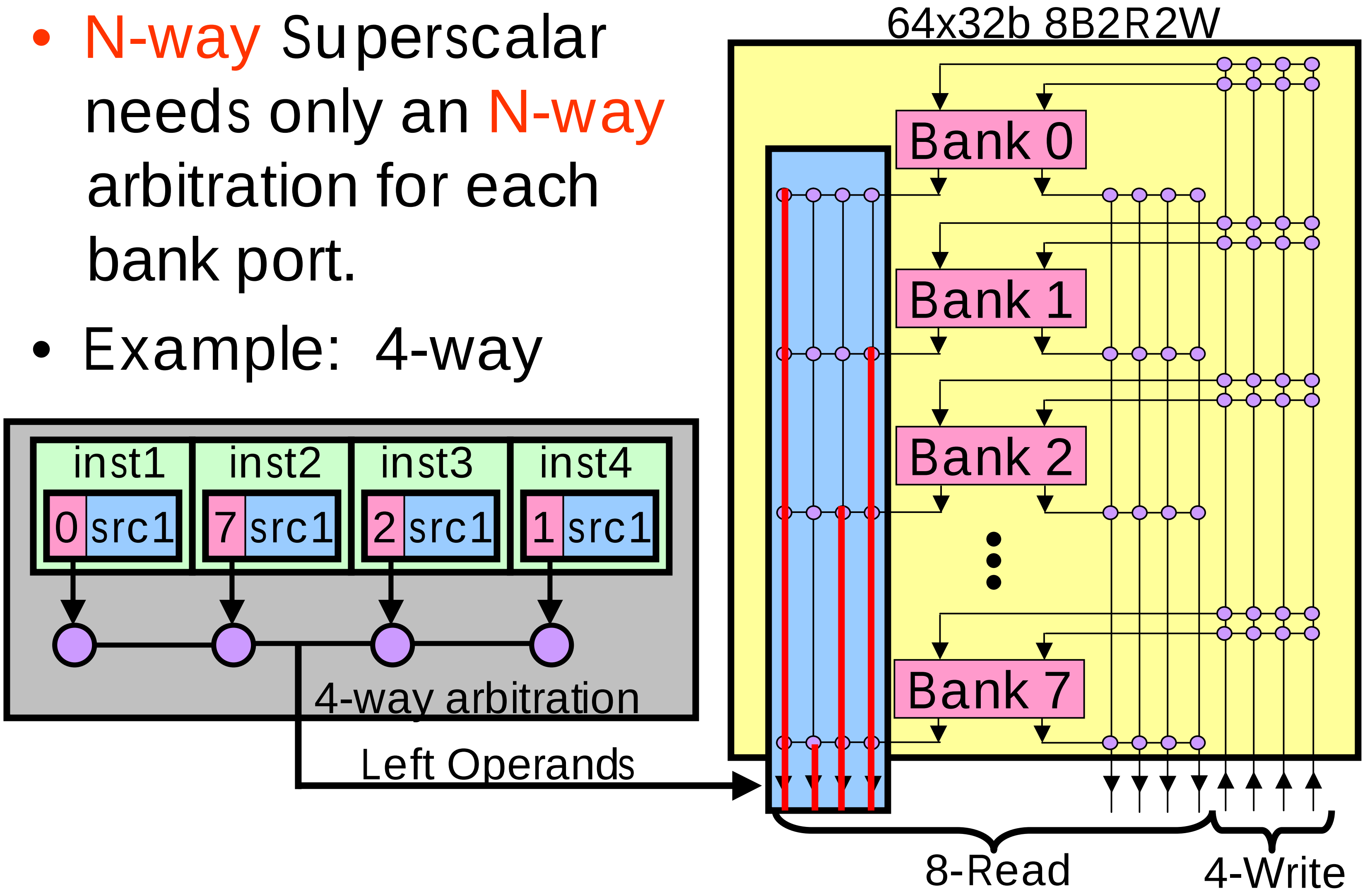


0	1	2	3	4	5	6	7
Fetch	Decode	Rename	Issue	Arbitrate	Read Bypass	Execute	Writeback

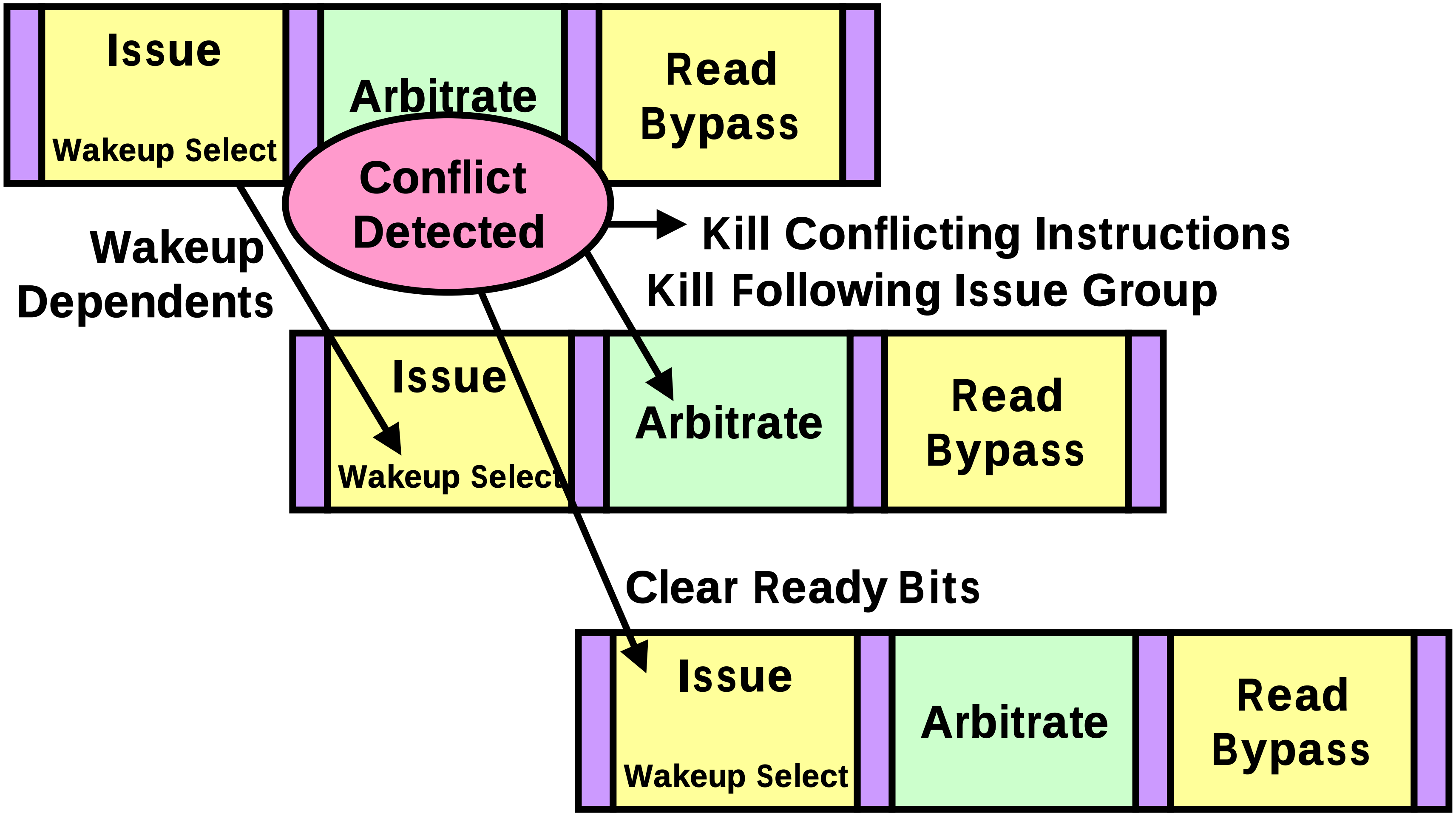
- Speculatively Issue Potentially Conflicting Instructions: Same Wakeup-Select Loop
- Additional Arbitration Pipeline Stage
  - Detect read and write bank conflicts when too many instructions try to read from or write to the same register file bank.
  - Mux operand addresses into available register file ports.
  - Adds a cycle to branch misprediction latency.

N-way Arbitration

- N-way Superscalar needs only an N-way arbitration for each bank port.
- Example: 4-way



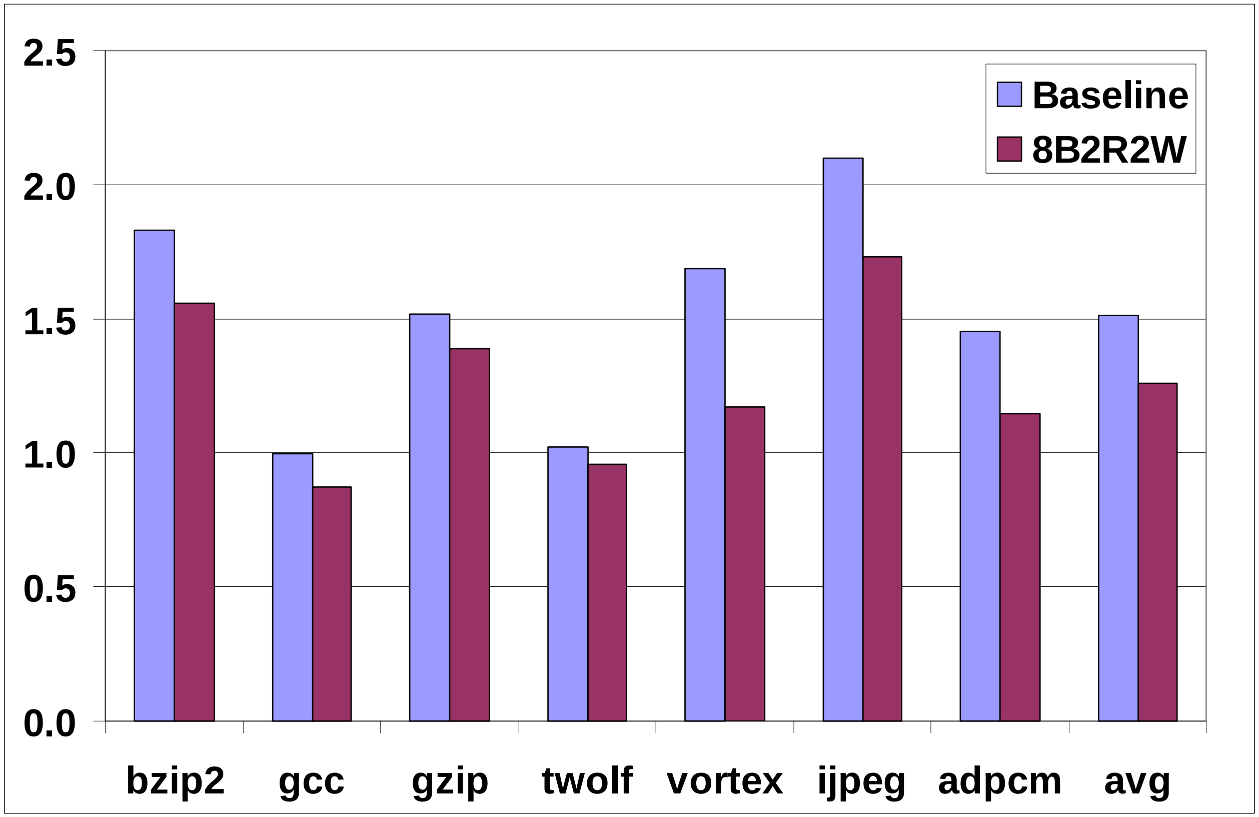
Pipeline Repair Operation



Evaluating IPC Impact

- IPC degradation simulation: modify Simplescalar simulator to keep track of a unified physical register file organized into banks.
  - Shorter access time of banked register files may lead to higher processor clock rate.
- Benchmarks: Use a subset of SPEC2000 and Mediabench benchmarks that cover a range of different IPCs.

IPC Comparison (1)



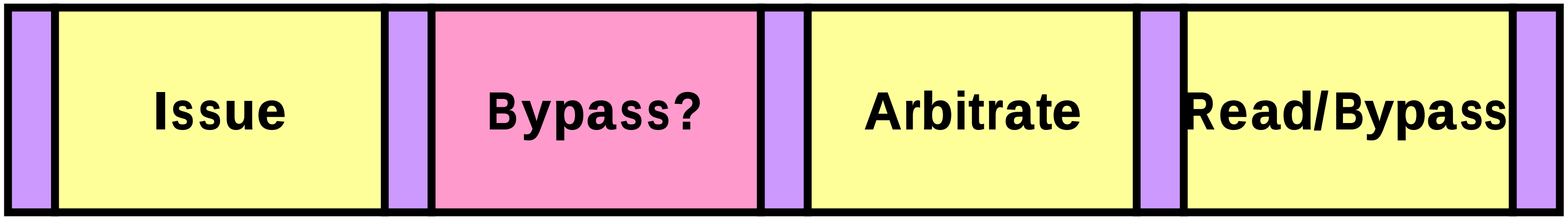
- IPC degradation ranges from 0.1 (9%) to 0.5 (31%) with an average of 0.3 (17%).

Improving IPC

- Avoid contending for register file read ports when it is possible.
  - Bypass Skip: Operands that will be sourced from the bypass network do not compete for access to the register file.
  - Read Sharing: Allow multiple instructions to read the same physical register from same bank.
- Suggested in previous work [Park et. al. MICRO-35, Balasubramonian et. al. MICRO-34]

Bypass Skip Implementation

- Need to determine bypassability before the arbitration for register file read ports.



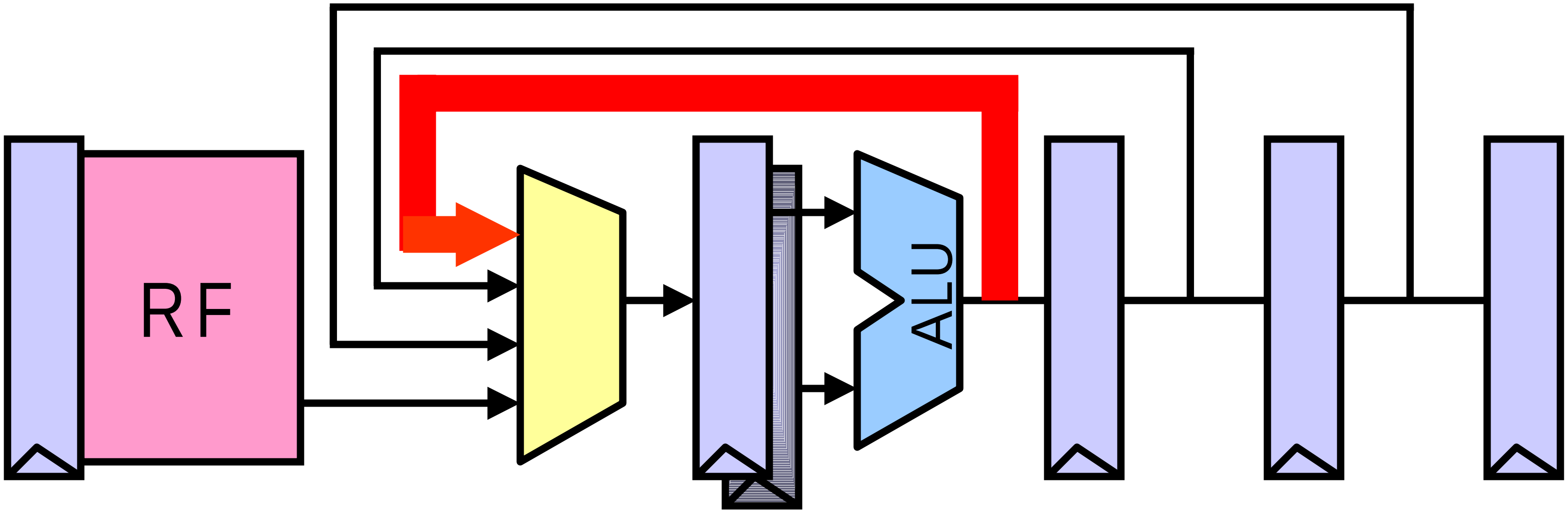
Issue | Bypass? | Arbitrate | Read/Bypass

  - Problem: Extra pipeline stage, possible latency increase

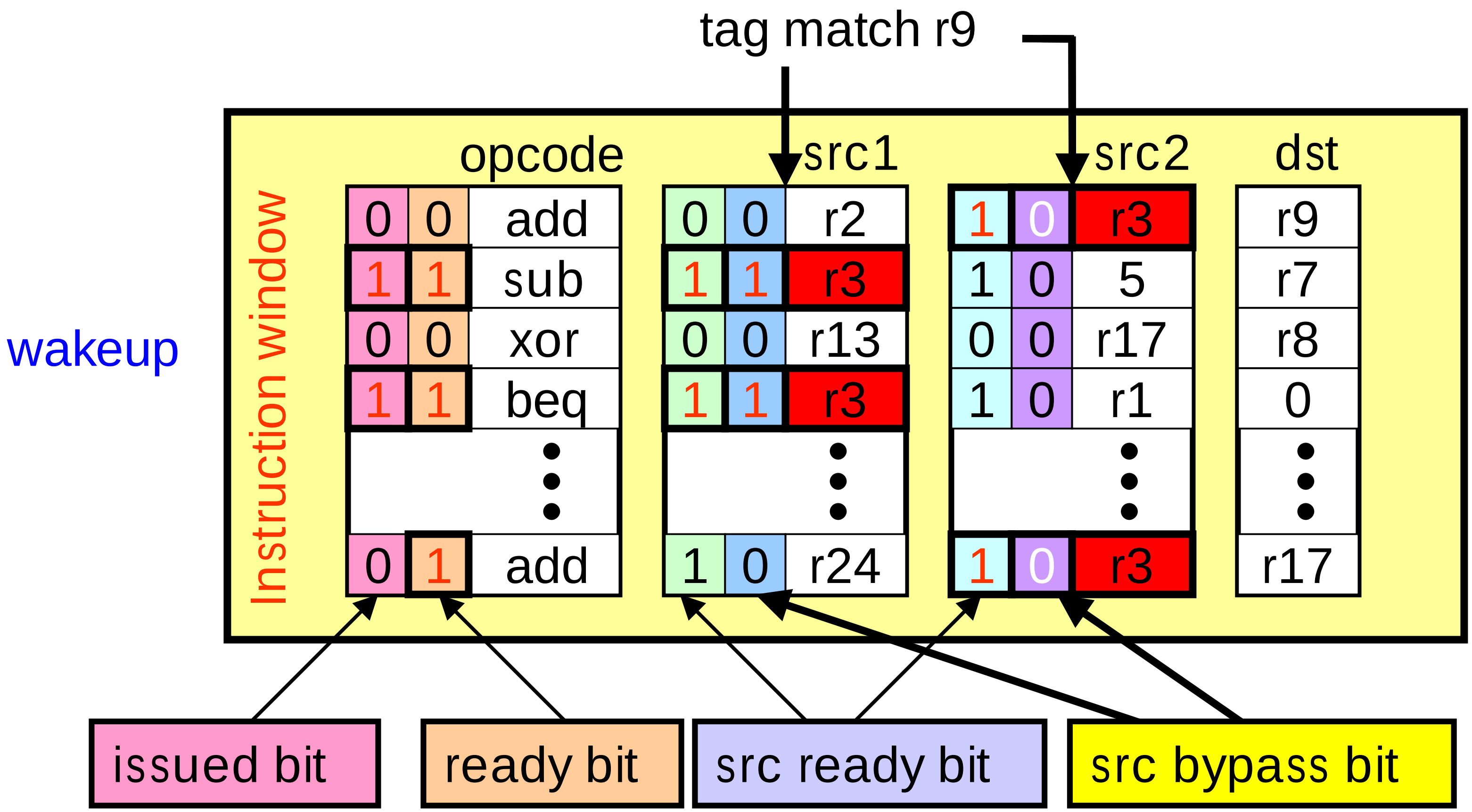
- Optimistic Bypass Hint: [Park et. al. 02'] Reducing register ports for higher speed and lower energy. MICRO-35.
  - Use wakeup tag search to indicate bypassability.
  - Bypassability indicator is not reset when the source instructions have written back to the register file.
  - Problem: Not always correct → could over subscribe the register file read ports.

Conservative Bypass Skip

- Conservative Bypass Skip Scheme
  - Use wakeup tag search to indicate bypassability.
  - Only avoid read port contentions when the value is bypassed from the immediately preceding cycle.



Bypass Bit Scheme



tag match r9

wakeup

Instruction window

		opcode			src1			src2	dst
0	0	add	0	0	r2	1	0	r3	r9
1	1	sub	1	1	r3	1	0	5	r7
0	0	xor	0	0	r13	0	0	r17	r8
1	1	beq	1	1	r3	1	0	r1	0
		⋮			⋮			⋮	⋮
0	1	add	1	0	r24	1	0	r3	r17

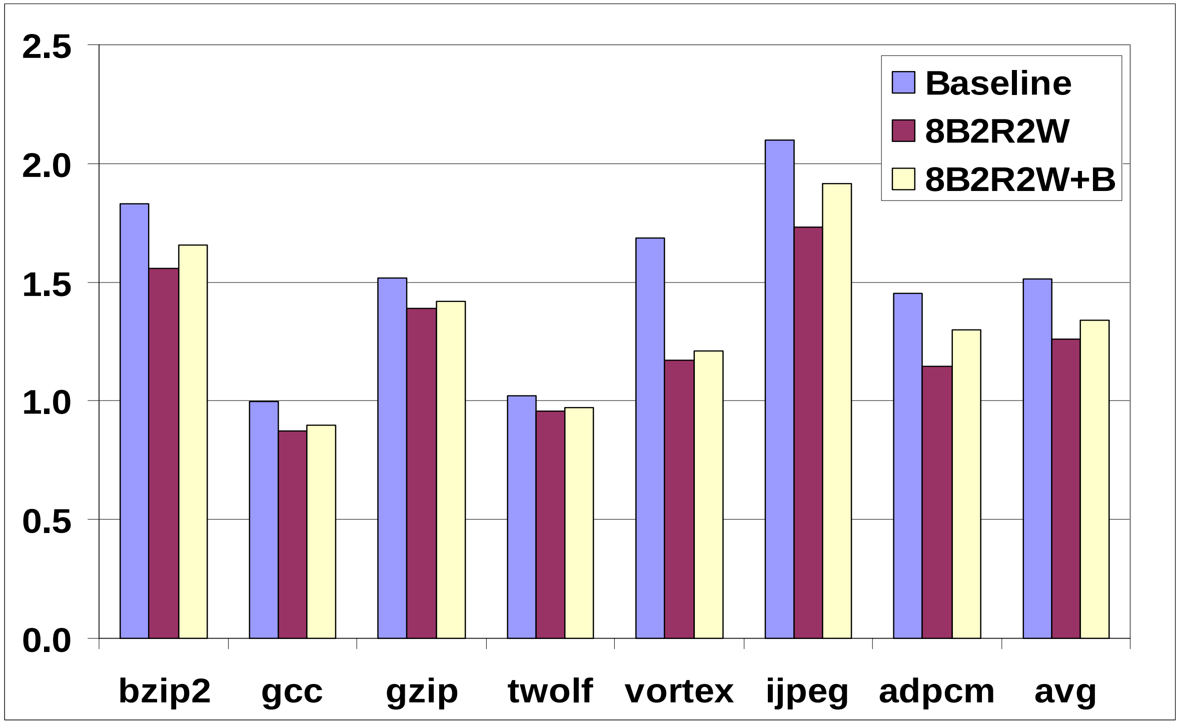
issued bit

ready bit

src ready bit

src bypass bit

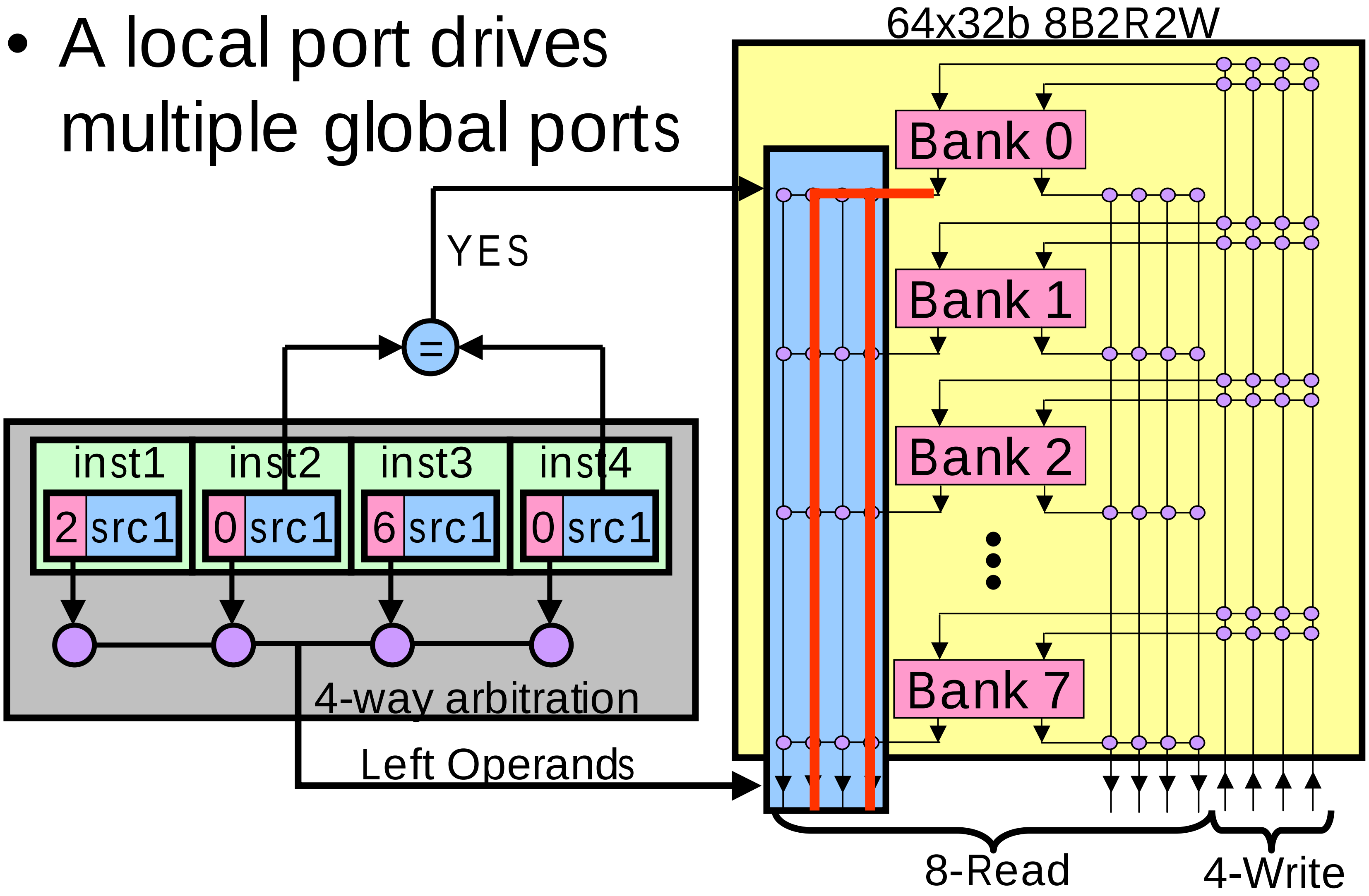
IPC Comparison (2)



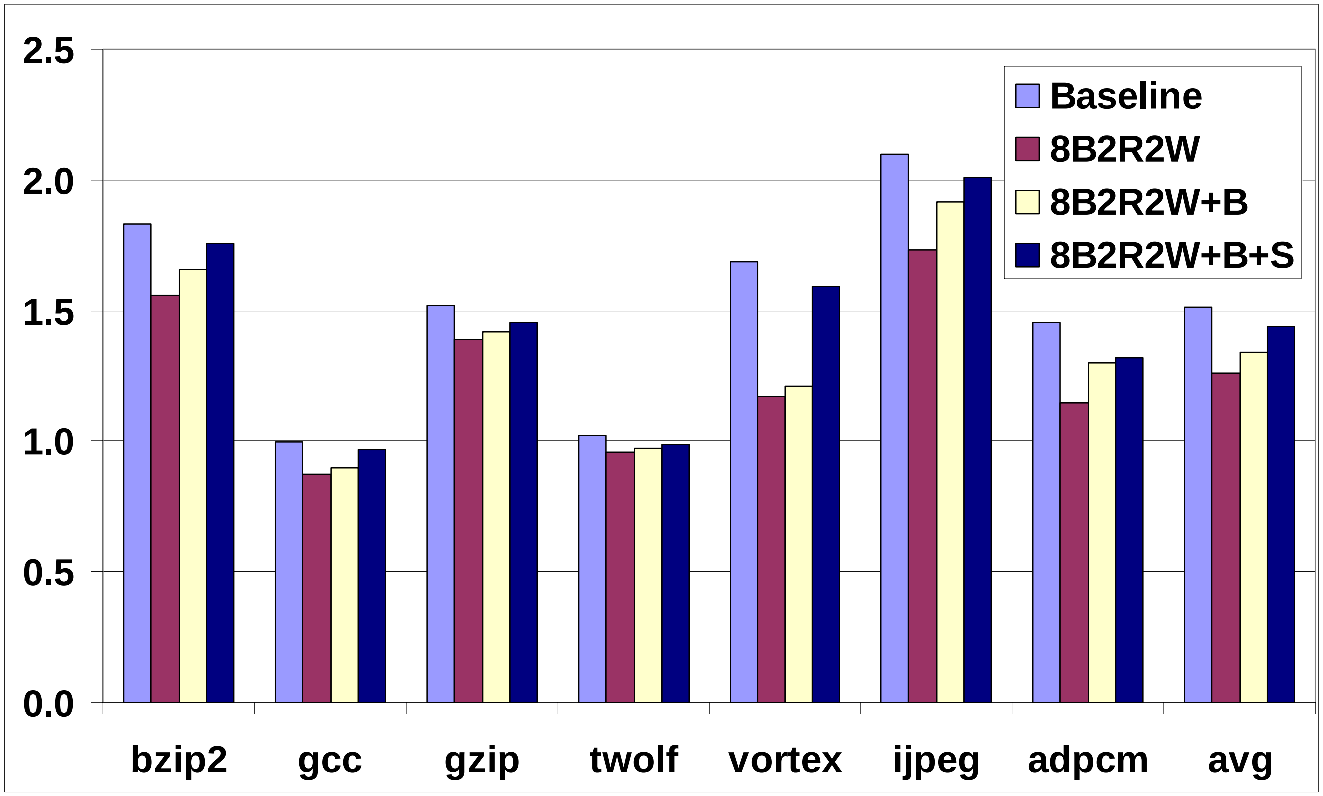
- Our conservative bypass skip scheme improves IPC by 5% on average.
- IPC degradation ranges from <0.1 (9%) to 0.5 (28%) with an average of 0.2 (12%).

Read Sharing

- A local port drives multiple global ports



IPC Comparison (3)



- Adding read sharing improves IPC by another 7% on average.
- IPC degradation ~0.1 across all the benchmarks with an average of <0.1 (5%).

Read Sharing Findings

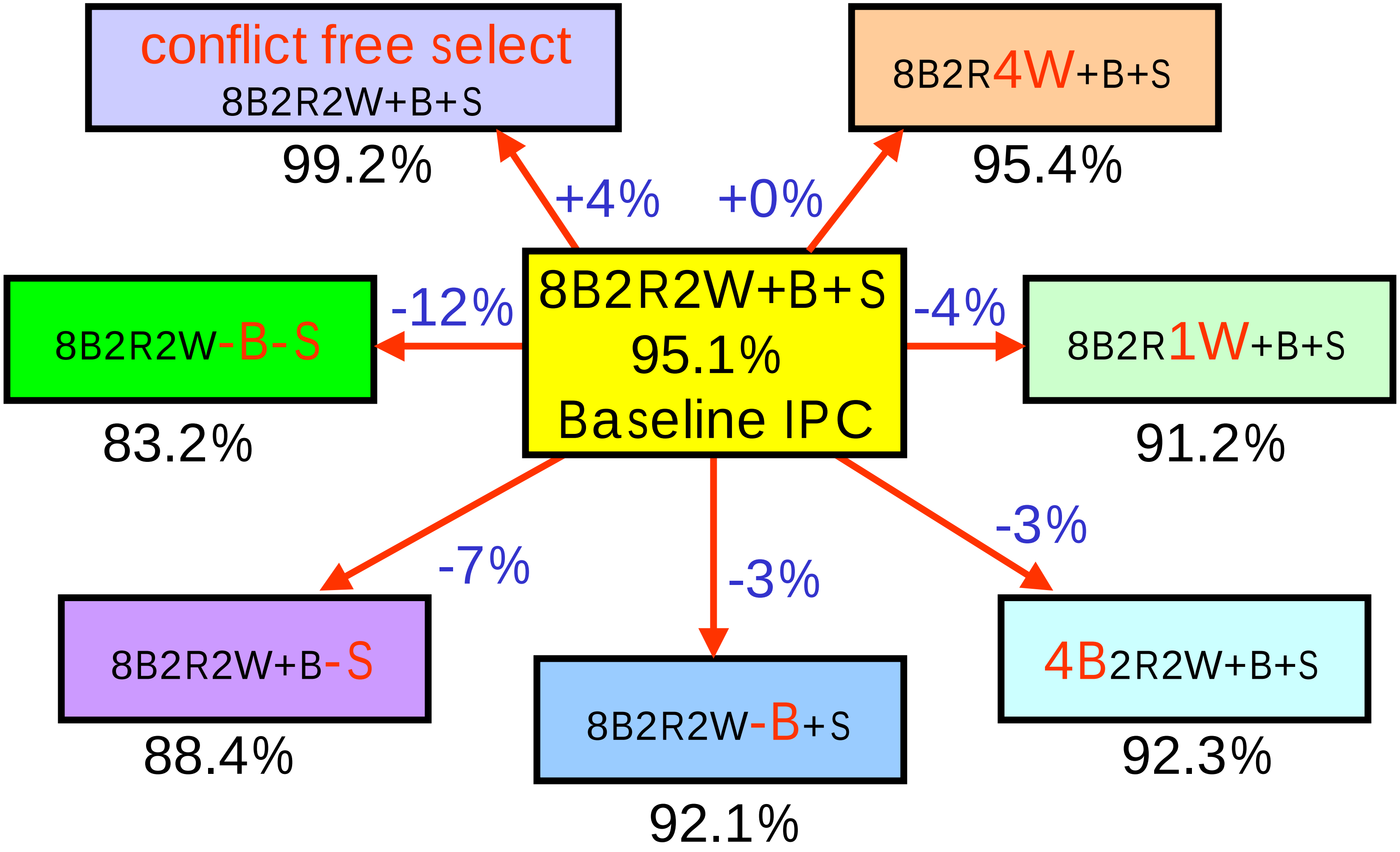
- Why are so many instructions reading the same register?

1. Groups of load and store instructions that depend on the stack pointer tend to be issued together. (procedure call/return points)

2. Branch instructions that depend on the same register also tend to be issued together.

- Confirms findings in previous work.
  - [Balasubramonian et. al. 01'] Reducing the complexity of the register file in dynamic superscalar processors. MICRO-34
  - [Wallace et. al. '96] A scalable register file architecture for dynamically scheduled processors. *Proc. PACT.*

IPC Sensitivity to Configuration



conflict free select
8B2R2W+B+S
99.2%

+4%

+0%

8B2R4W+B+S
95.4%

8B2R2W-B-S
83.2%

-12%

8B2R2W+B+S
95.1%
Baseline IPC

-4%

8B2R1W+B+S
91.2%

-7%

-3%

-3%

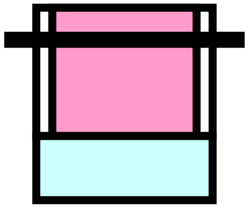
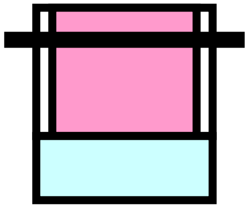
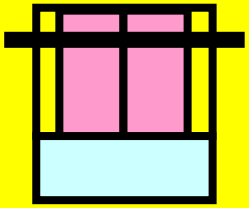
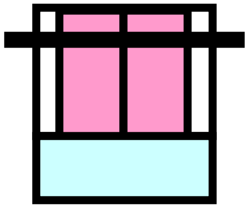
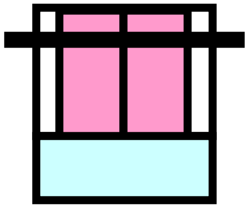
8B2R2W+B-S
88.4%

8B2R2W-B+S
92.1%

4B2R2W+B+S
92.3%

Register File Characteristics

- Area: Magic, 0.25μm TSMC CMOS process
- Delay & Energy: HSPICE, 2.5V supply voltage

64x32b, 8 Read Ports & 4 Write Ports					
Type	Baseline	8B8R4W	8B2R2W	8B2R1W	8B1R1W
Area	100%	123%	37%	32%	30%
Delay	100%	83%	75%	75%	77%
Energy	100%	61%	59%	58%	41%
Packing Bitline					

Errata

- Corrected Table 2

Delay	8r4w	2r2w	2r1w	1r1w
1 bank	100.00%	--	--	--
4 bank	92.38%	79.05%	79.05%	81.90%
8 bank	83.88%	74.76%	74.76%	77.14%

http://www.cag.lcs.mit.edu/scale/

Discussion

- Why Design with Multi-Banked Register File?
  - Reduce Area Dramatically
  - Reduce Access Time → Higher Clock Rate
  - Reduce Energy Consumption
  - Cause Only Slight IPC Degradation
  - Scale With Technology
    - Wire Delay
    - Leakage Power
- Future Work:
  - SMT Architecture

Conclusion

- For register file with a small number of local ports per bank, the overall register file area is dominated by bank interconnect.
- Using more ports per bank to reduce the IPC impact of a simpler and faster pipelined control scheme that allows higher frequency operation.
- For four-issue processors, we reduce register file area by over a factor of three, access time by 25% and access energy by 40%, while reducing IPC by less than 5%.

Thank You

- http://www.cag.lcs.mit.edu/scale/