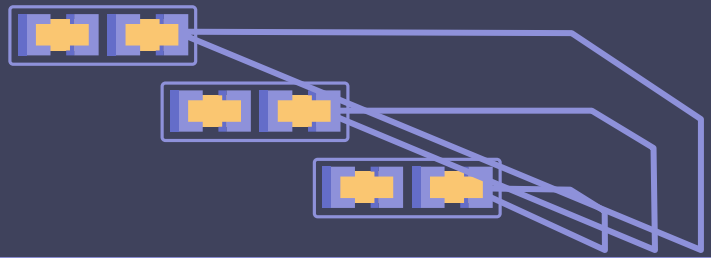
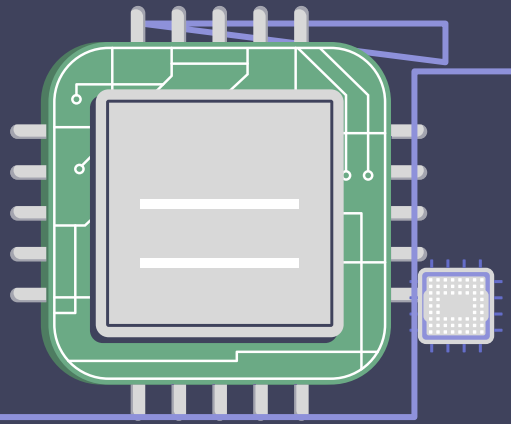
The Speculative 6

A RISC-V, N-Way Superscalar, MIPS R10K-Style Out-of-Order CPU Major Design Experience

[redacted] John Collison, [redacted]



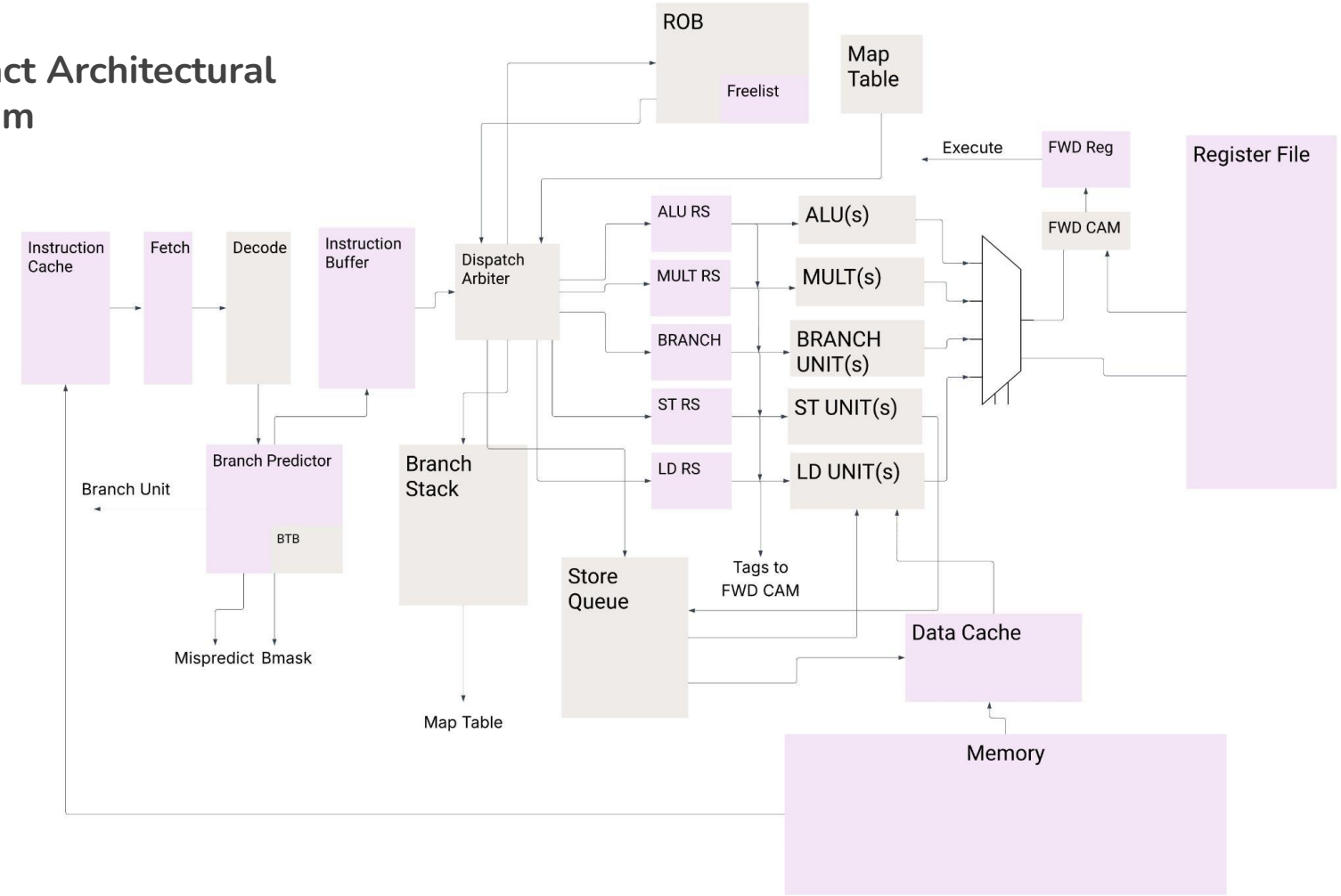


Major Design Decisions

Main Features

- N-Way Superscalar
- Early Branch Resolution
- Early Tag Broadcast
- 2-banked Data Cache
- Store Queue with Byte-Level Forwarding

Abstract Architectural Diagram

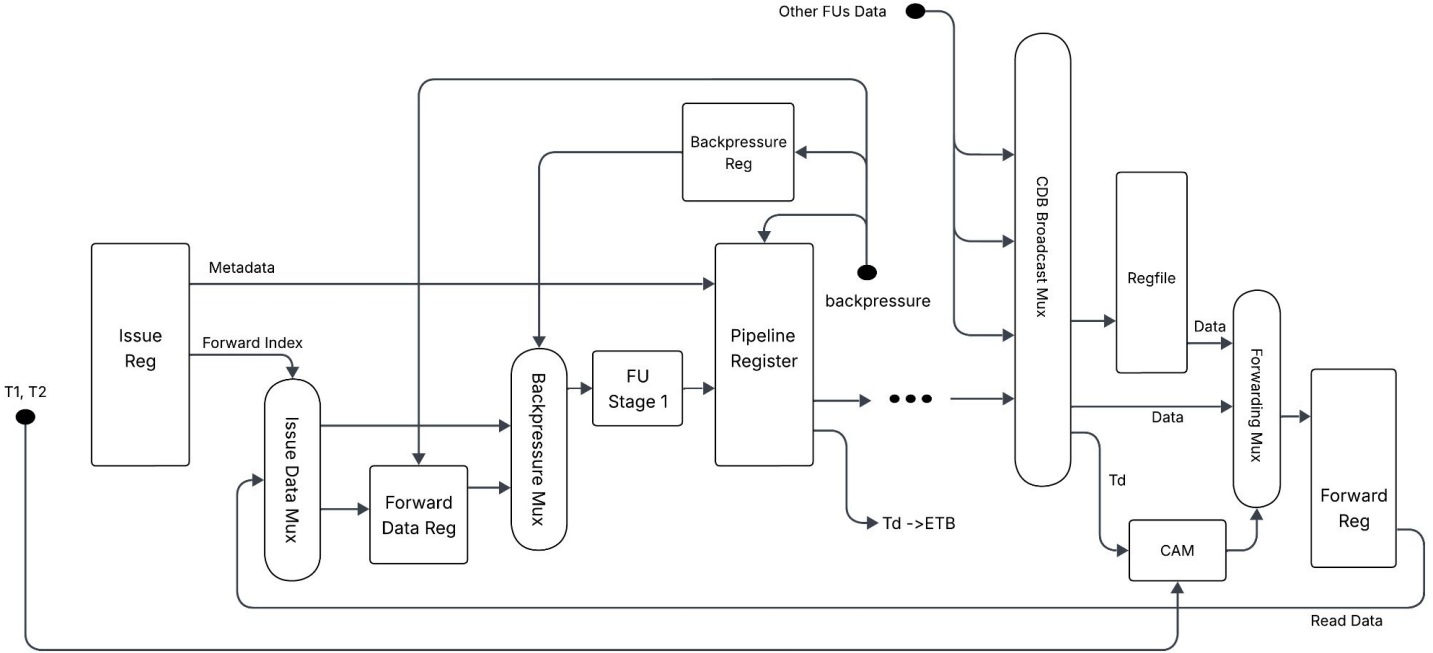


Early Branch Resolution

- Branch Stack checkpoints map table, ROB tail, and SQ tail
- Does what the map table would’ve done one cycle later
- ROB head + tail used to control free list, so free list did not have to be checkpointed

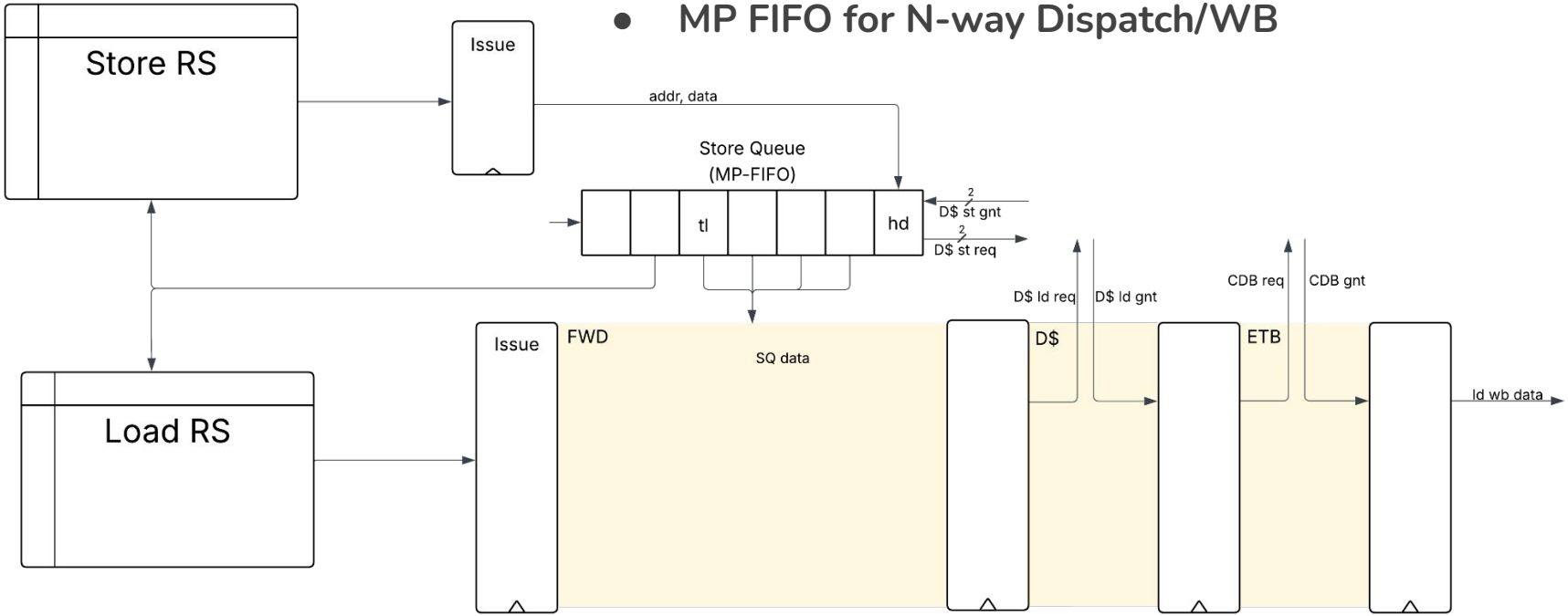
Early Tag Broadcast

- Split clock forwarding design
- Dependent instructions can run with no cycle gap



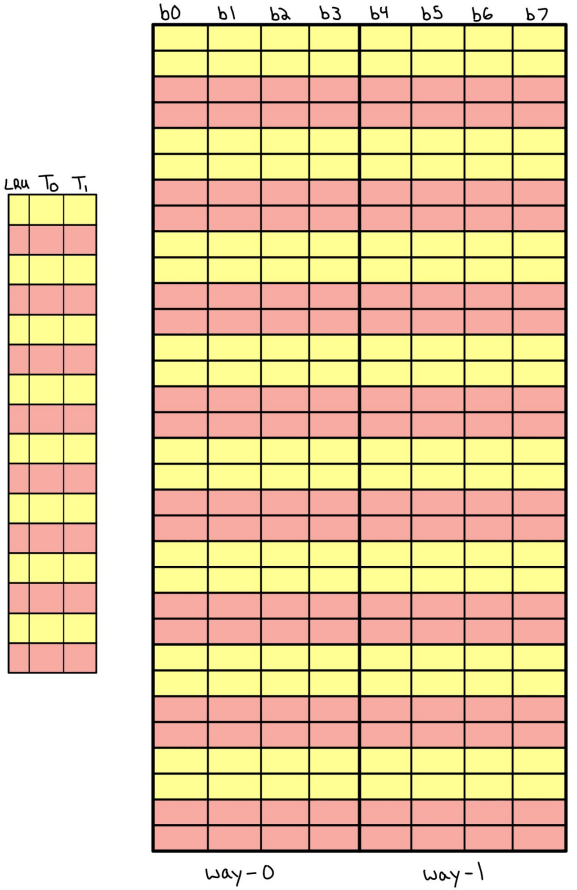
LSQ Design

- Store Queue with Byte Level Forwarding
- All SQ Data to Load FU for Forwarding
- SQ Entries Marked for WB Upon ROB Retirement
- MP FIFO for N-way Dispatch/WB



D$ Design (16-memDP)

- 2 Banks
- 2-way Set Associative
- Byte-Level wr_en (8 memDP Per Bank)
  - Store Directly to D$
  - Removes Read/Modify/Write



Performance

Achieved Performance

Achieved:

5-way

13 ns 4-way & Branch Stack of 8*

12ns 4-way & Branch Stack of 4*

Settled:

1.631 CPI*

16 ns clock

2-way

*Couldn’t test Icache synthesis in time, set safety net to 16ns

*Weighted Average

Performance on Simple Icache vs Maximum Fetch

Theoretical (CPI)

Ways	alexnet	outer_product	insertionsort	evens_long	sort_search	fib_rec	basic_malloc
1	1.61	1.52	1.64	1.21	1.71	1.89	2.31
2	1.04	1.10	1.17	0.73	1.31	1.58	1.86
4	0.89	0.97	1.11	0.50	1.23	1.44	1.67

Simple (CPI)

Ways	alexnet	outer_product	insertionsort	evens_long	sort_search	fib_rec	basic_malloc
1	4.95	1.68	1.67	1.79	1.74	1.9	6.01
2	4.49	1.12	1.2	1.32	1.37	1.64	5.59
4	4.486	1.099	1.15	1.318	1.285	1.54	5.59

Branch Predictor Characteristics (Ideal Fetching)

Accuracy

alexnet	outer_product	fib_rec	btest2	dft
80.9%	85.9%	74.7%	33.3%	77.7%

Maximum Branches Allowed and GShare Index Bits (CPI Weighted Average)

#branches \ PHT bits	4 bits	6 bits
4 branches	1.076	1.074
8 branches	1.072	1.068

Further Improvements

Optimized Instruction Cache

- 4-Way Set-Associative Cache
- 2 Cache Banks
- Stream Buffer Prefetching

Optimized Instruction Cache

Test Program	Simple I-Cache CPI	Fancy I-Cache CPI
`btest1`	8.229	----
`btest2`	6.982	----
`copy_long`	0.992	0.686
`copy`	1.538	1.288
`evens_long`	1.318	0.804
`evens`	1.687	1.333
`fib_long`	1.235	0.690
`fib_rec`	1.503	----

Test Program	Simple I-Cache CPI	Fancy I-Cache CPI
`fib`	1.533	1.093
`haha`	5.278	1.889
`insertion`	1.147	1.070
`mult_no_lsq`	1.845	1.300
`no_hazard`	5.500	2.000
`parallel`	1.1650	0.995
`sampler`	5.845	----
`saxpy`	1.642	1.337

Optimized Reservation Station

- Add history logic for issuing older instructions
- Use priority selectors instead of prefix sums
- Merge stages with faster issue logic

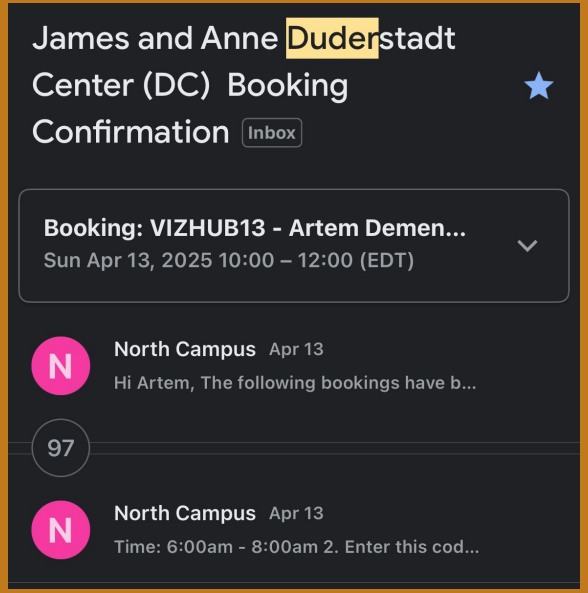
Victim Cache

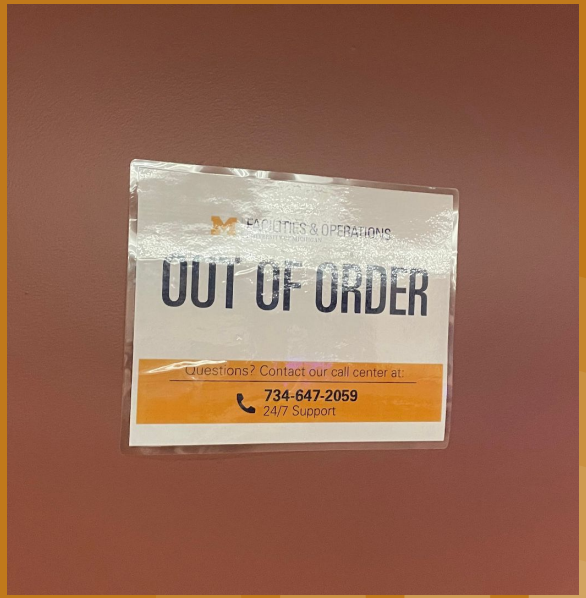
- Size 4
- Fully Associative
- For data cache only

Overall Takeaways from this Project

- Communication is key
- Discuss heavily detailed design before implementation
- Everything took longer than expected
- Leave ample time for integration and debugging
- Do not focus on premature optimizations

Questions?





Haha, get it...?