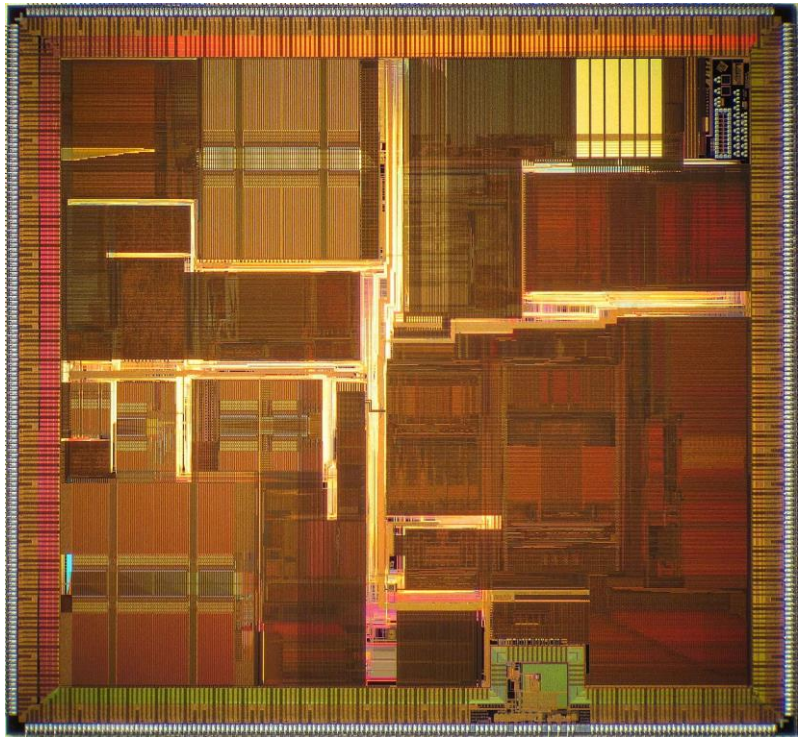
Out-of-Order Processor Design

Group #15 C. Huang, J. Dong, X. Wei & H. Zhang

Design Overview

- Out-of-Order pipeline processor
- Based on R10K
- 3-way scalar



NEC VR10K Die Shot

Basic Features

Fetch Stage

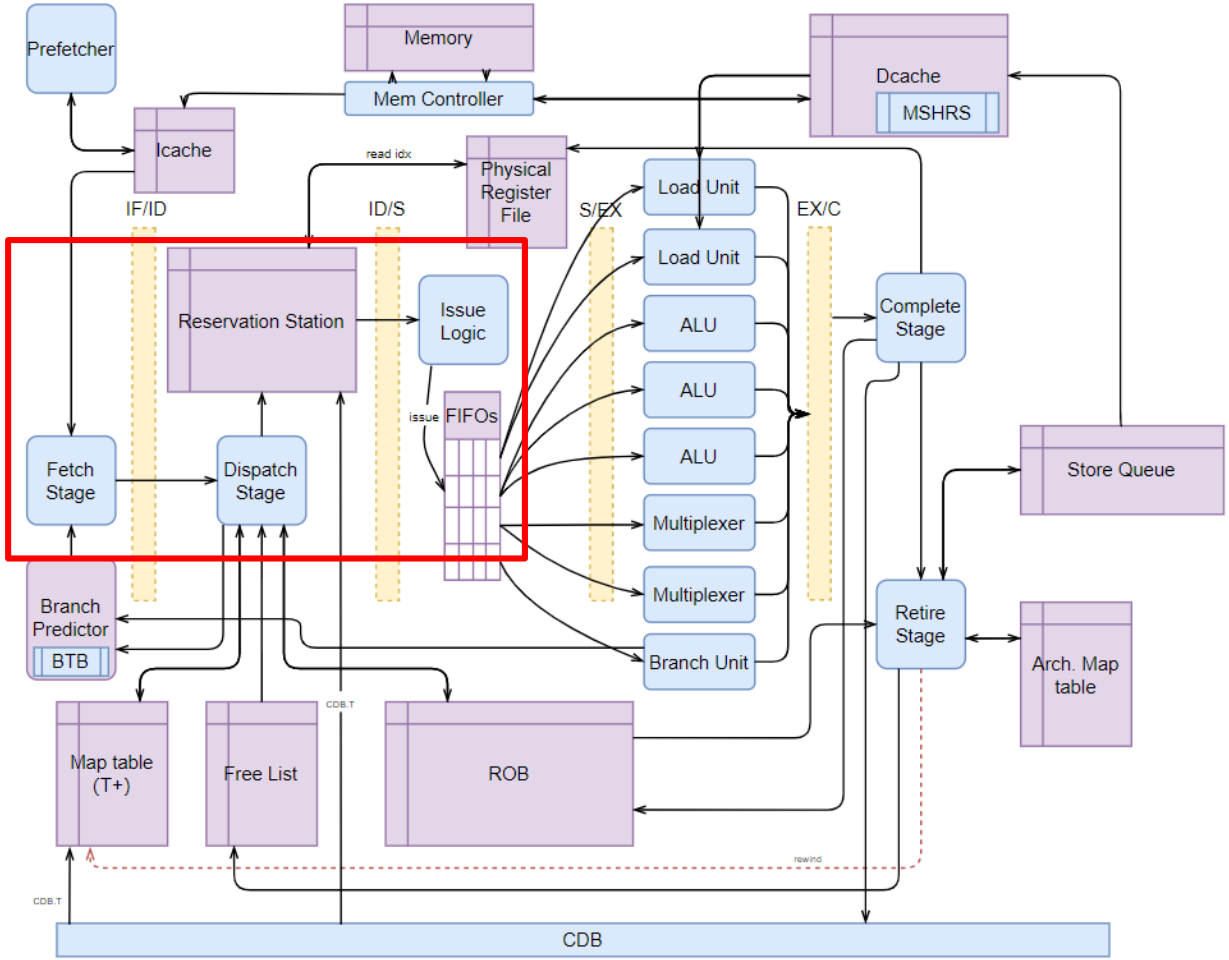
- Three Instructions at a time
- Stalls on cache miss or dispatch stall

Dispatch Stage

- 3 decoders embedded

Reservation Station

- 16 entries
- Use Priority Selectors to choose instructions to issue
- FIFO queues for different kinds of FUs



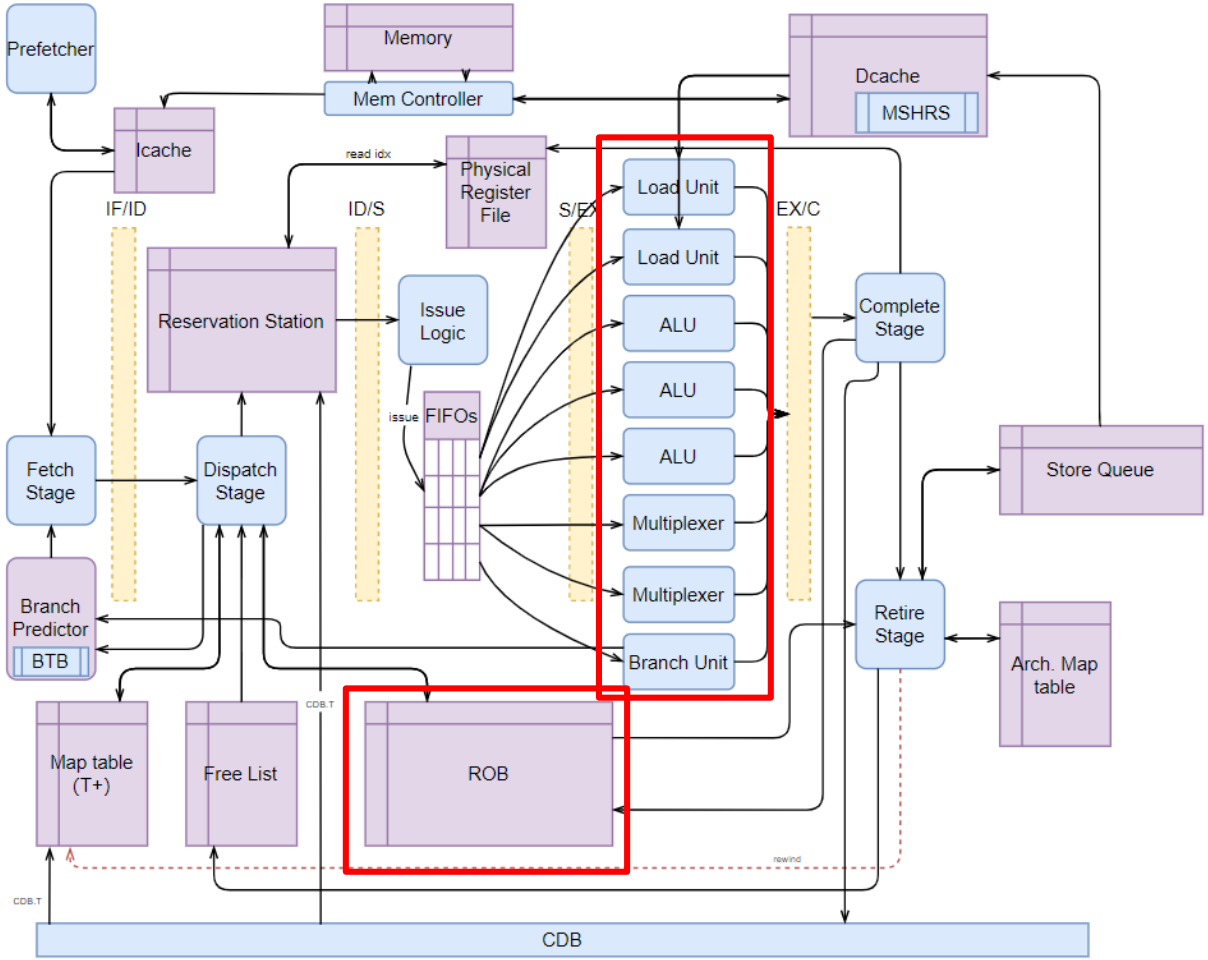
Basic Features

Reorder Buffer

- 32 entries
- Head and tail pointers - non-retire instructions
- Precise state: reset all entries and pointers

Functional Units

- ALUs handle stores



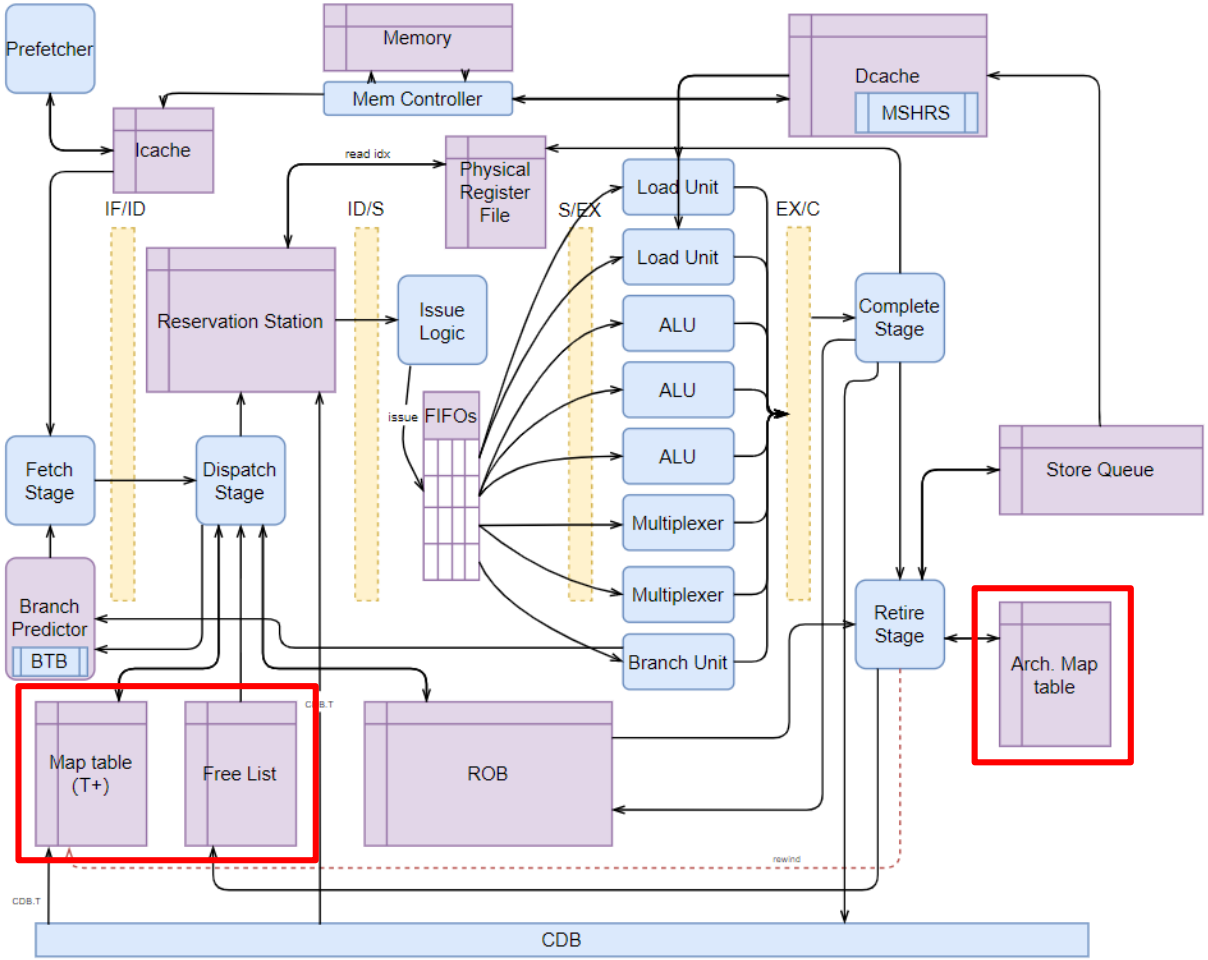
Basic Features

Freelist

- 32 entries
- Head and tail pointers - free reg in between
- Precise state: set head pointer right after tail pointer

Map Tables

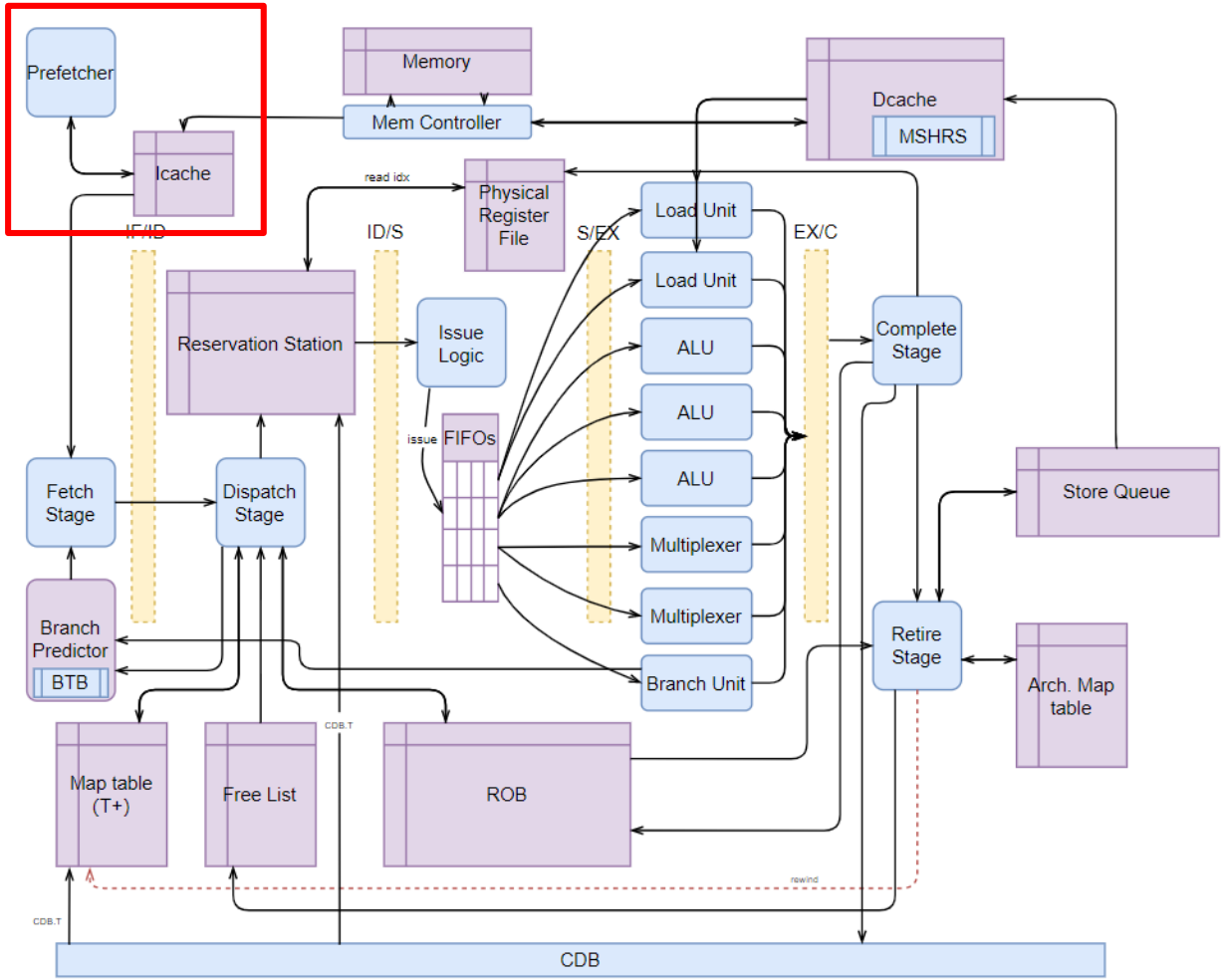
- 32 Arch regs
- 64 Physical regs



Advanced Features

Prefetcher

- Triggered on every reference
- Prefetches 12 lines* of data from memory and write into ICache
- Tells fetch if some data has been requested from memory



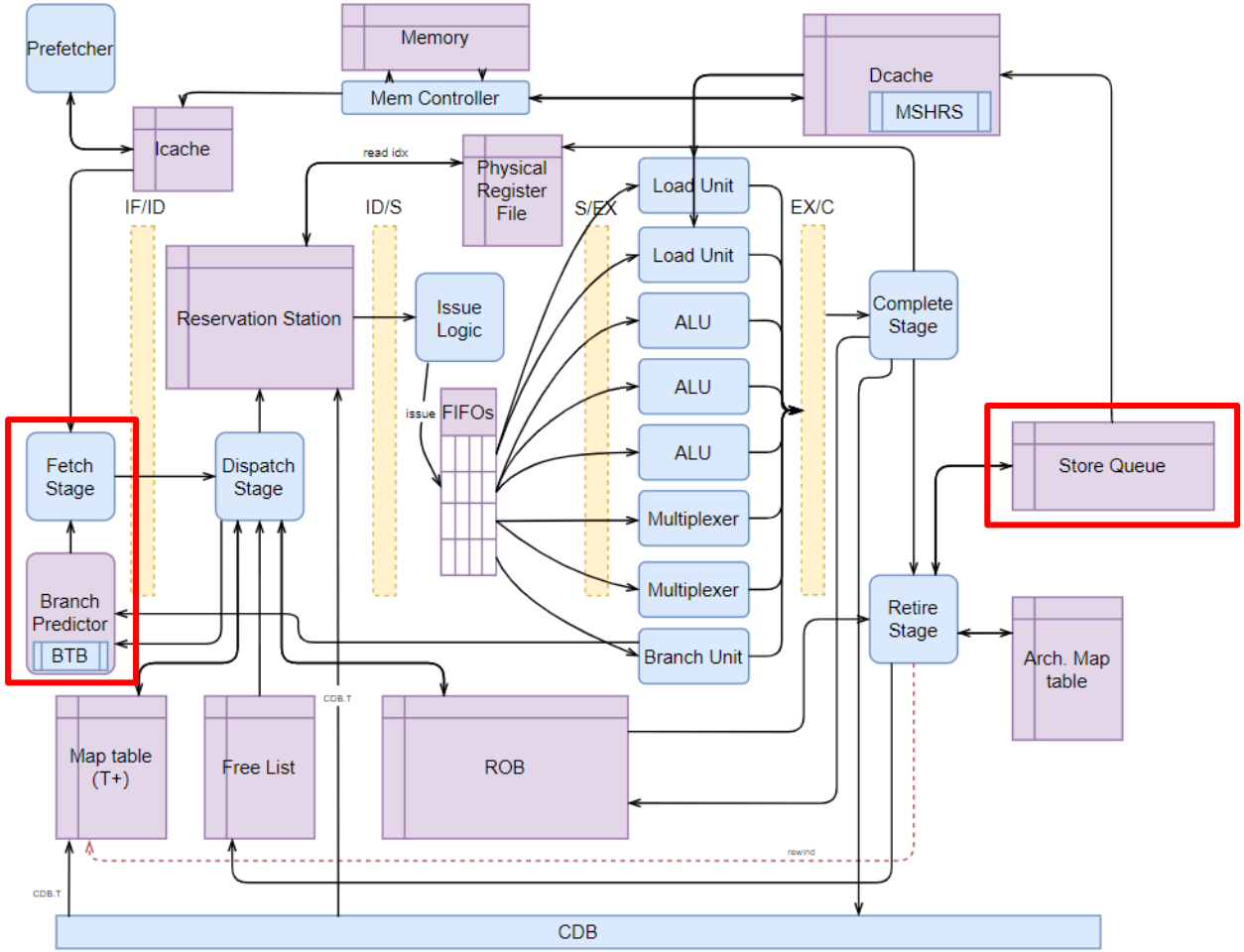
Advanced Features

Branch Predictor

- 32 entries
- Directly mapped
- Based on a 2-bit saturating counter
- Initialized as not-taken

Store Queue

- 8 entries
- Every load instruction waits until all the store instructions in the queue are executed



Memory Interface

Non-blocking Instruction Cache

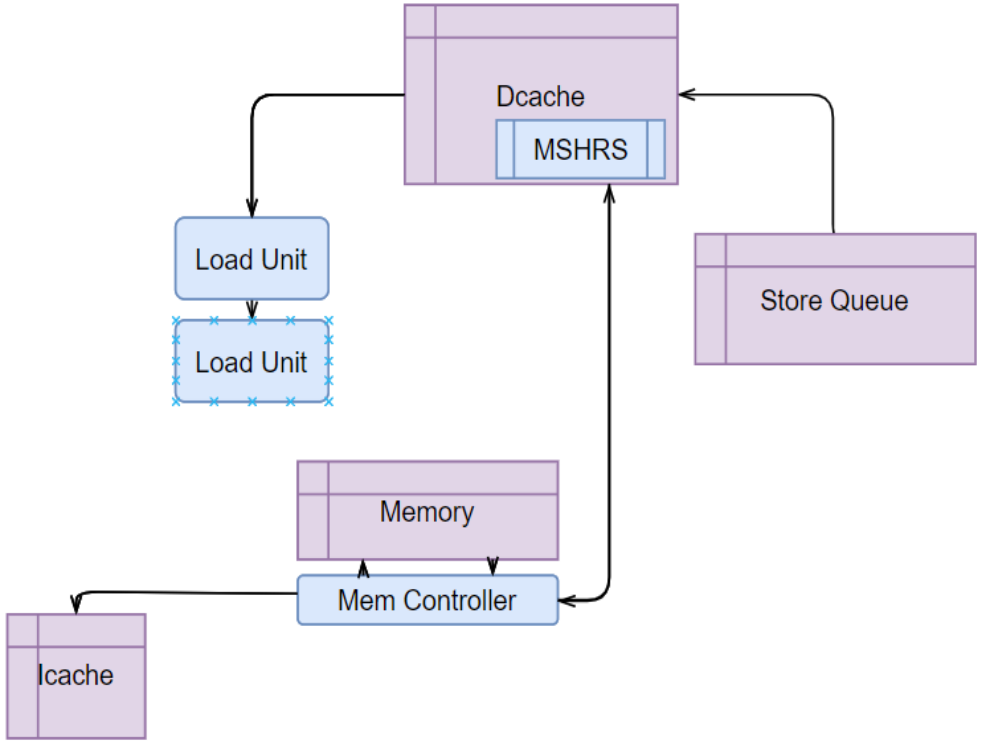
- 32 lines, directed mapped

Non-blocking Data Cache

- 32 lines, directed mapped
- Write-back
- Write-allocate
- Uses a Dcache controller with a 16-entry Miss Status Handling Registers (MHSRs) to sequentialize memory accesses.

Memory Controller

- Give Dcache higher priority



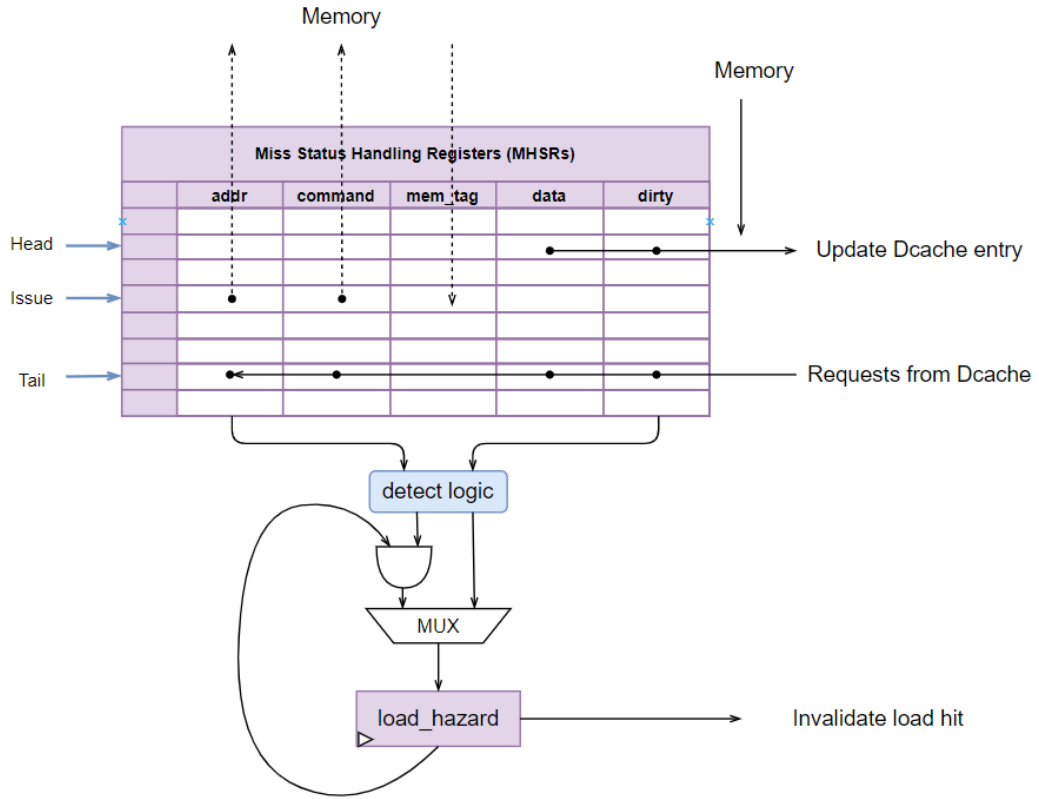
Memory Interface

Miss Status Handling Registers

- 16 entries, FIFO
- Designed to track memory tags, and record information of multiple accesses.

Dcache Controller

- Special cache updating logic
- A “load hazard” detector to avoid loading wrong data.

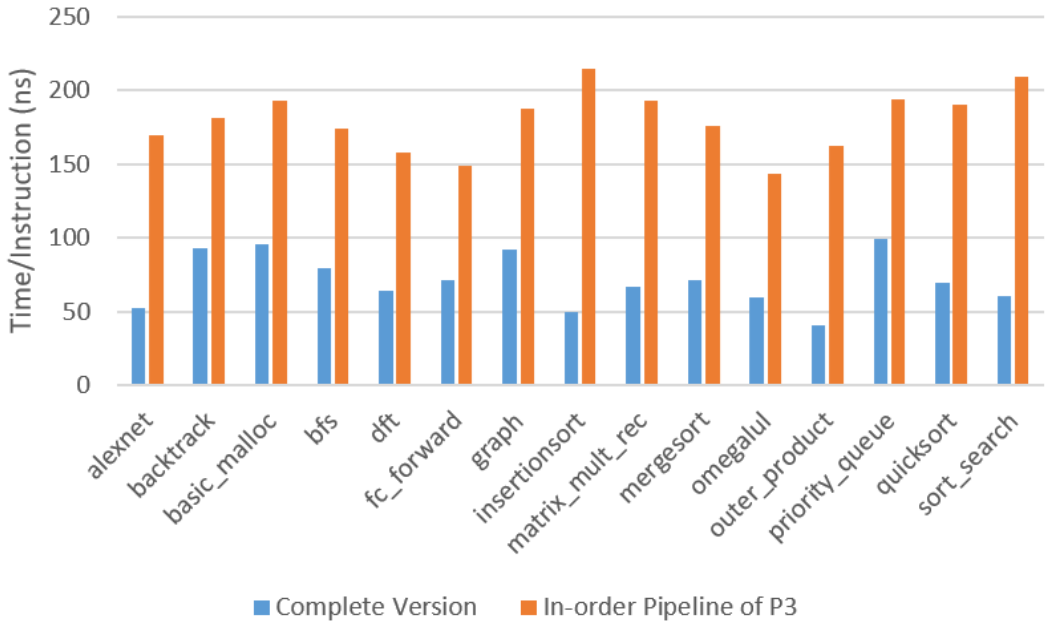


Performance Analysis

Overall Performance:

- Versus in-order pipeline
- Time/Instruction reduced significantly

Time/Instruction = CPI × Period



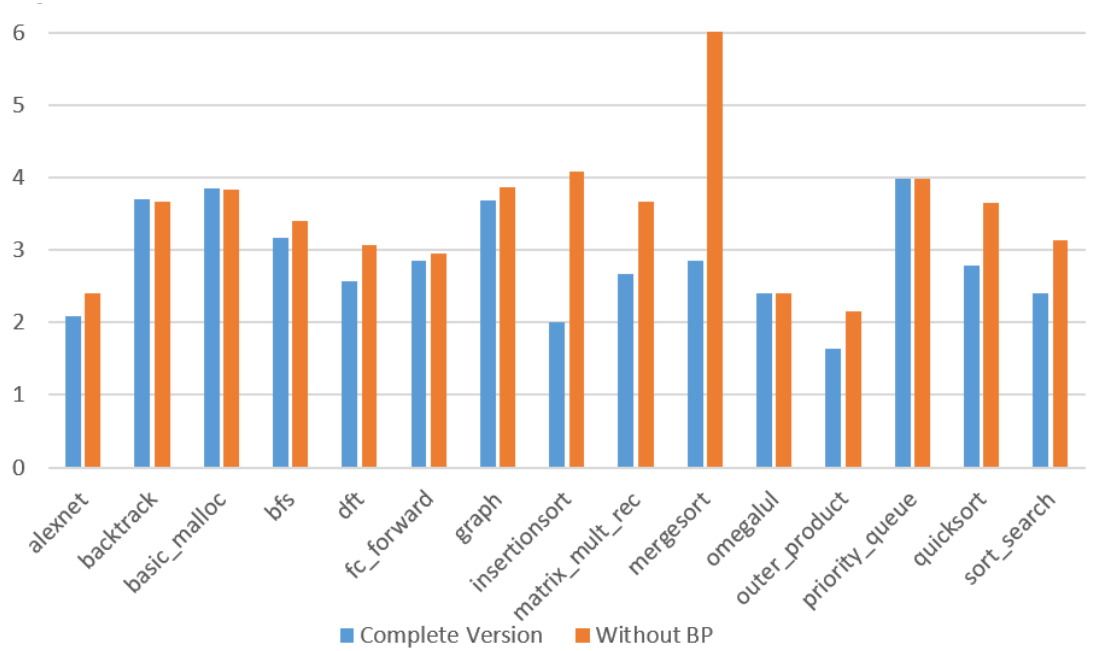
Performance Analysis

Prefetch:

- Overall efficient
- Poor in branch-taken

Branch Predictor:

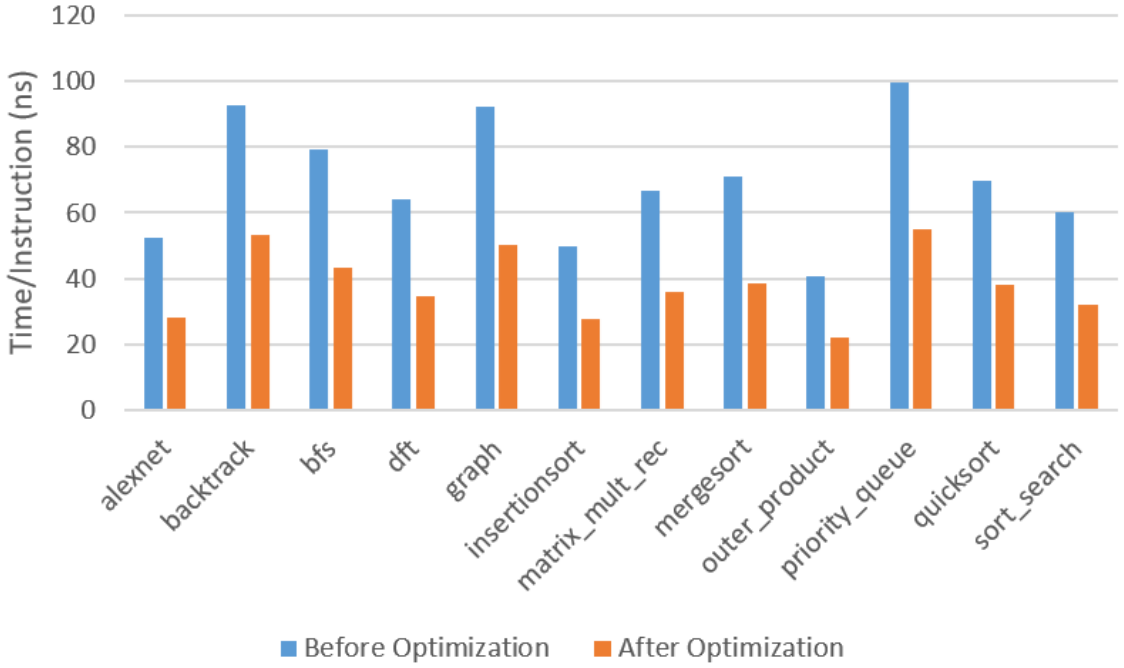
- Overall efficient
- Poor in changeable pattern



Performance Analysis

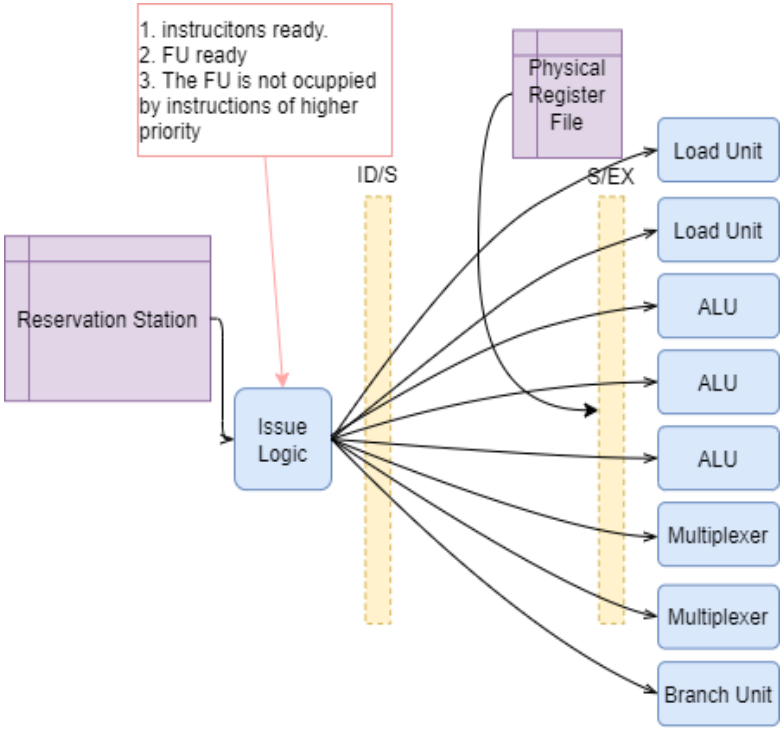
Up-to-date Version:

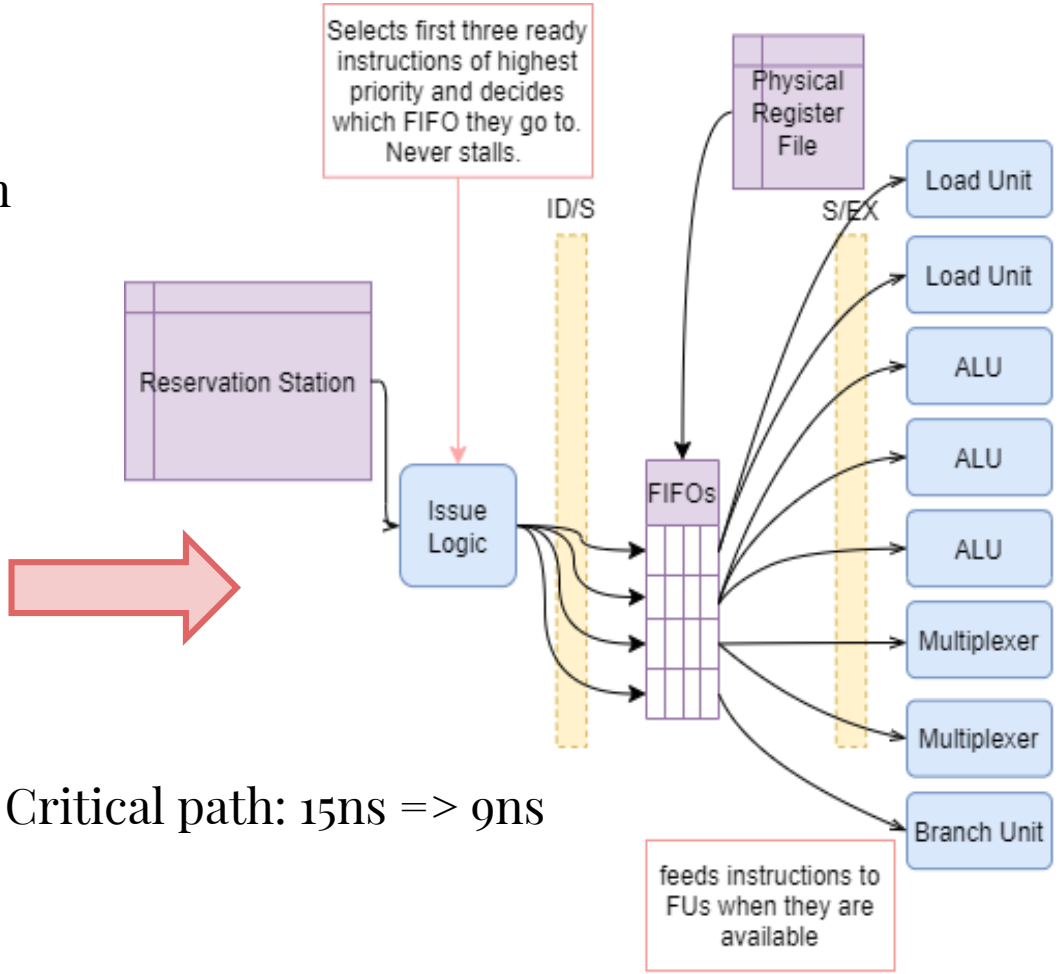
- Period greatly reduced
- CPI slightly increased
- Half time/instruction



Optimization

- Issue selection logic Separation





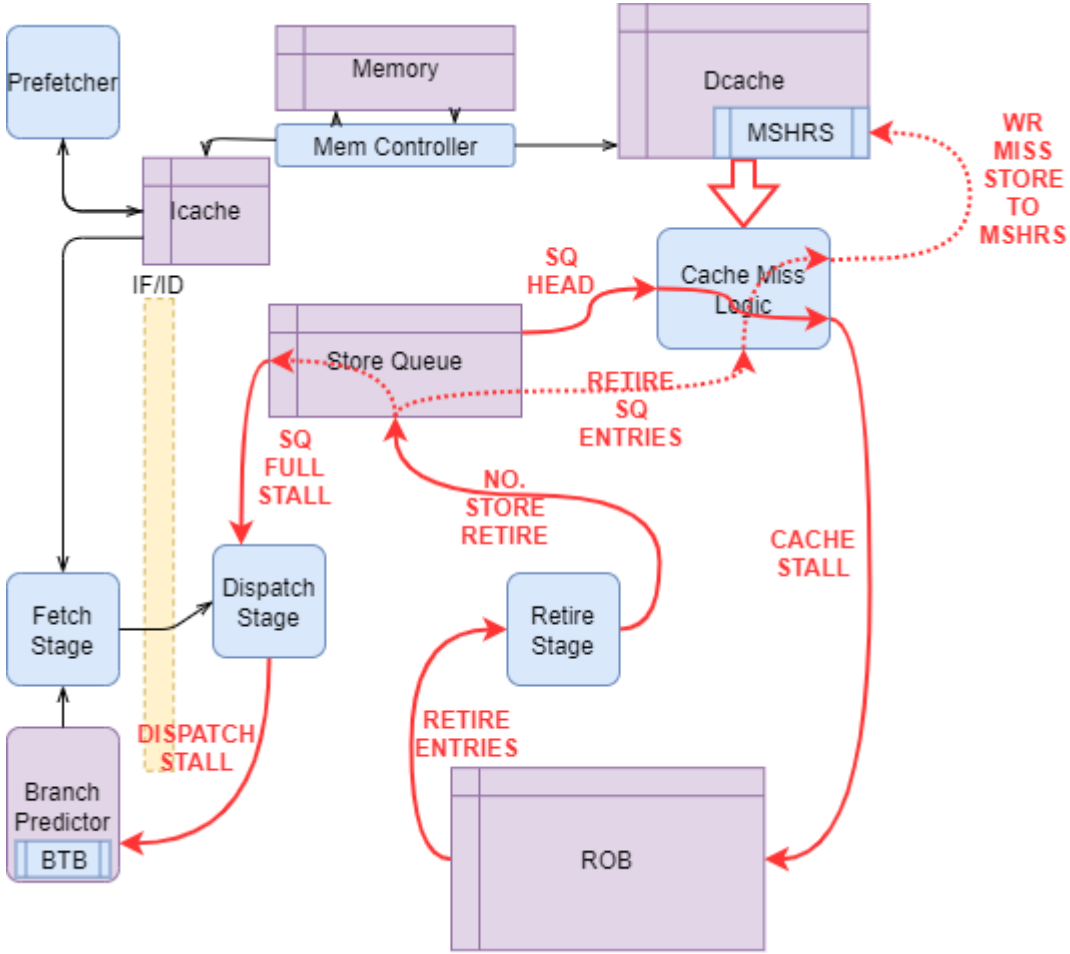
Critical path: 15ns => 9ns

Optimization

- DCache Stall Logic Critical Path Elimination

  Target: Allow only one missed store at a time

1. Cancel SQ internal forwarding
2. Eliminate redundant cache miss logic

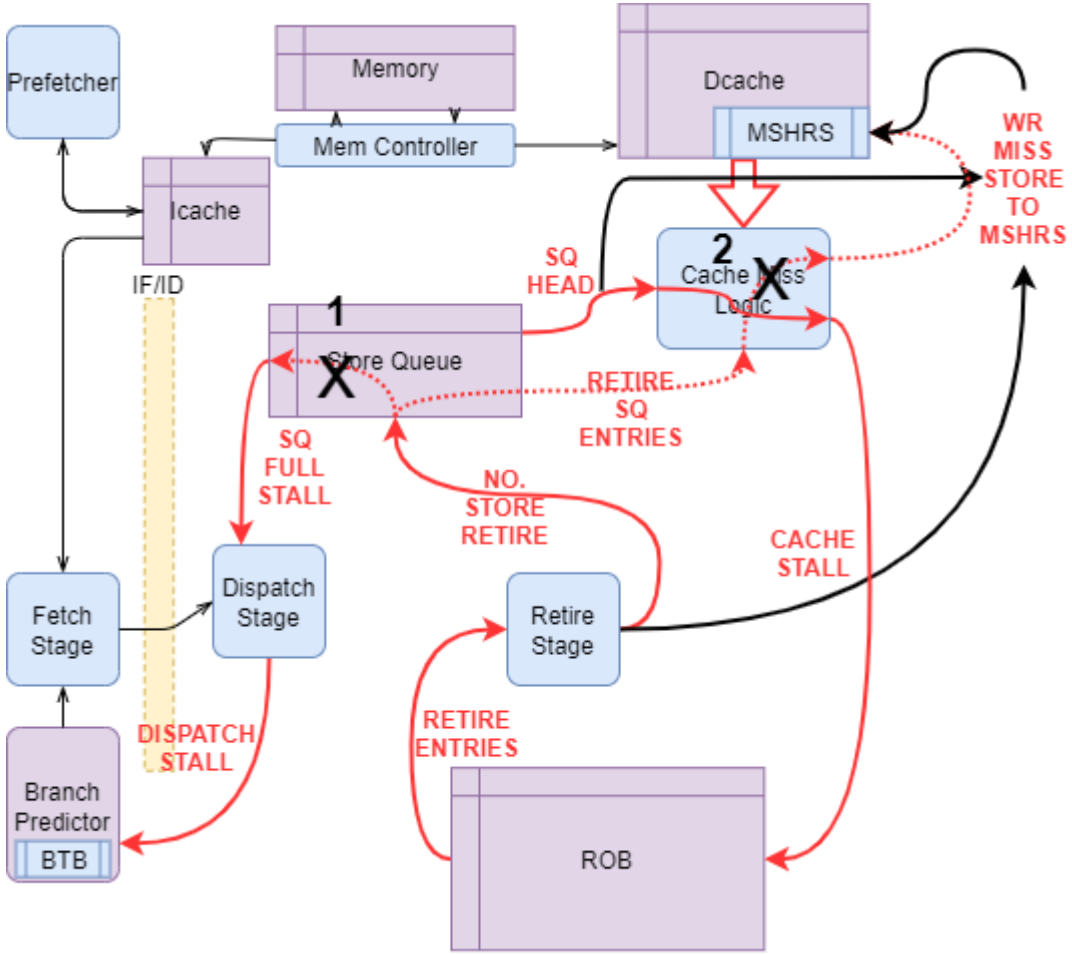


Optimization

- DCache Stall Logic Critical Path Elimination

  Target: Allow only one missed store at a time

1. Cancel SQ internal forwarding
2. Eliminate redundant cache miss logic



Cycle period: 25ns => 13.1ns

1.85x

Time/inst: 71.07ns => 38.35ns

Speedup After Optimization

Questions?