

# EECS 470 Computer Architecture Final Project Presentation

Group 12: Shixin Song, Zesheng Yu, Yuqing Qiu, Chenyan Zhang, Zimeng Zhang

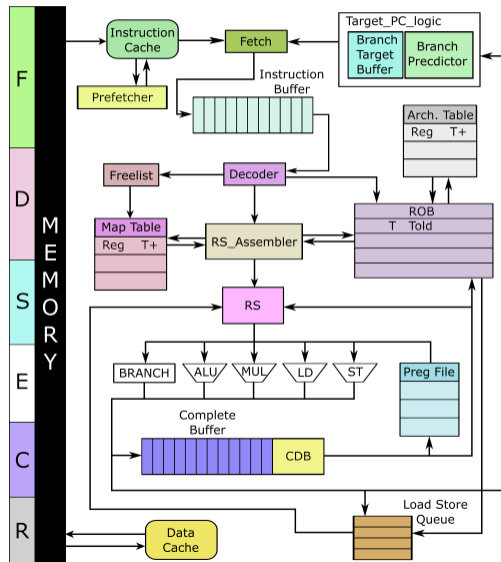
University of Michigan

2022/04/19

# Implemented Features

## Features:

- ▶ MIPS R10K Architecture
- ▶ 3-way superscalar
- ▶ Blocking 4-way set associative DCache
- ▶ GShare Branch Predictor
- ▶ Branch Target Buffer
- ▶ Instruction Prefetcher
- ▶ Instruction buffer



# Fetch Stage

- ▶ Determine fetch instruction PC according to Target PC Logic.
- ▶ Fetch one cache line each time (1-2 instruction).
- ▶ Stall on lcache miss or instruction buffer (32 entry) is full.

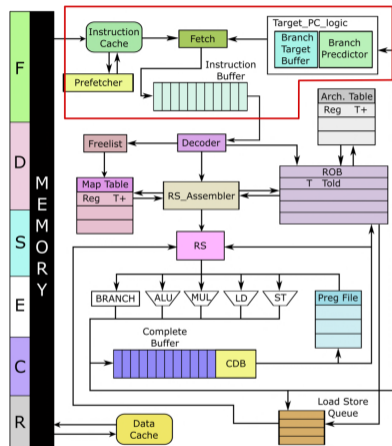


Figure: Target PC Logic Design

# Fetch Stage

## Advantage:

- ▶ Instruction Buffer allows fetch even when dispatch stage stalls.
- ▶ Target PC Logic with a predecoder, GShare branch predictor and BTB allows a more precise instruction fetch.

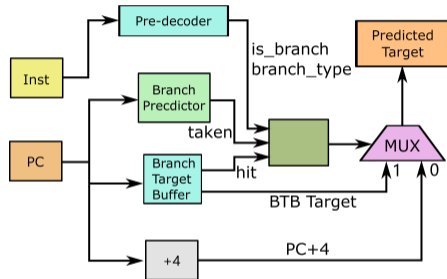
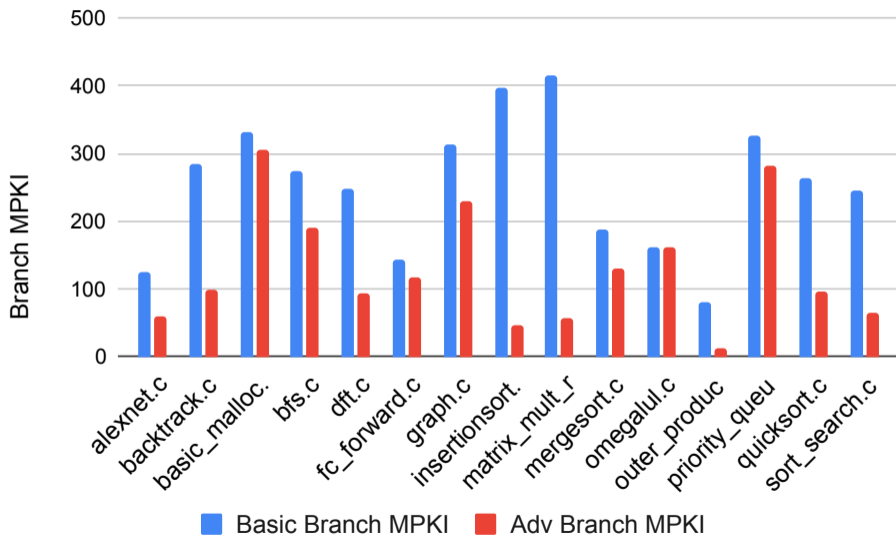


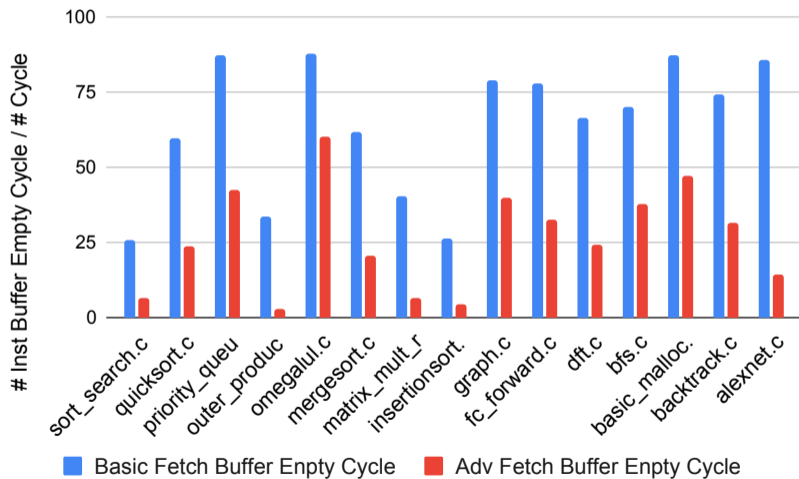
Figure: Target PC Logic Design

# Branch MPKI



## Inst Buffer Empty Cycle

To take full use of the instruction buffer, we then implement a next-line prefetcher.



# Dispatch Stage and Issue Stage

## Dispatch Stage:

- ▶ Decode raw instructions into unified packets
- ▶ Dispatch instructions in FIFO order based on the free count of ROB and FUs
- ▶ Ensure freelist never meets structural hazards

## Issue Stage:

- ▶ Number of RS entry same as Number of FUs
- ▶ Once ready, RS issue instructions to the corresponding FU
- ▶ Physical register file supports at most 6 read requests

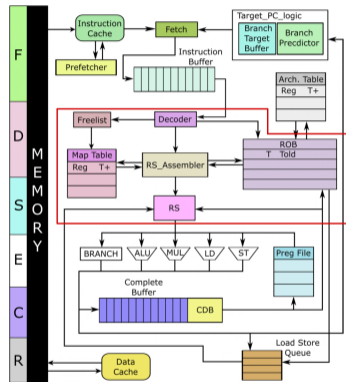


Figure: Dispatch and Issue Stage

# Execute Stage and Complete Stage

## Execute Stage:

- ▶ Instruction from RS, value for Preg file
- ▶ 3 ALUs, 3 MULTs, 3 LDs, 3 STs, 1 Branch
- ▶ MULT take 2 cycles, others 1
- ▶ More than 3 inst completed per cycle

## Complete Stage:

- ▶ Complete buffer with max size same as ROB size
- ▶ Broadcast at most 3 inst results per cycle

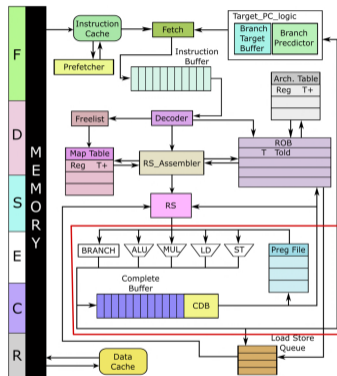


Figure: Execute and Complete Stage



# Retire Stage

## Retire Stage

- ▶ 3 completed instructions at the head of ROB can be retired.
- ▶ On branch misprediction, start squashing ROB and RS, copy map table and calculate freelist
- ▶ When stores retire, LSQ will send storing requests to DCache.

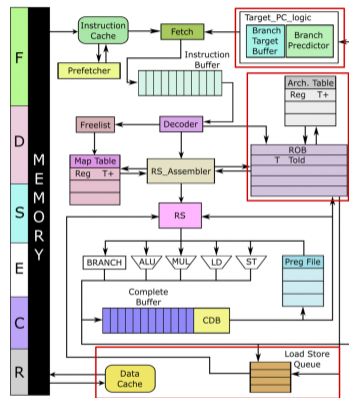


Figure: Retire Stage

## Final performance we have achieved

- ▶ Reduce the clock period from one memory access latency to 17.5ns
- ▶ Performance evaluated by Time/Instruction is largely improved

# Performance Analysis

- ▶ Basic design vs. in-order pipeline
- ▶ Time/Instruction reduced significantly

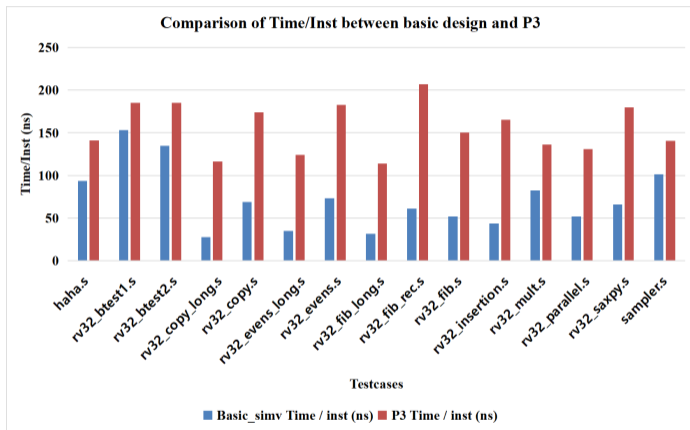


Figure: Time/Inst of Basic design vs P3

# Performance Analysis

- ▶ Advanced design vs. basic design
- ▶ Time/Instruction reduced significantly
- ▶ especially on the cases with less load and store instructions

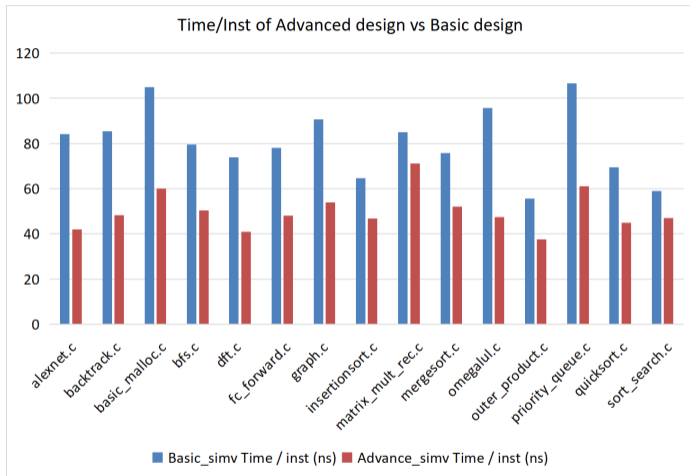


Figure: Time/Inst of Advanced design vs Basic design

## Challenges we have overcome

- ▶ Redesign ROB, RS and LSQ to decrease the synthesis clock period from 20ns+ to 8ns
- ▶ Revise the given C program to make smaller program to debug

## Future improvements

- ▶ Non-Blocking Data Cache to support multiple load and store requests
- ▶ Separated LSQ to support multiple load in execute stages and multiple store retirement
- ▶ Early branch resolution to squash incorrect branches in advance

Thanks for listening!