



# MBOOM: An Out-of-order Dual-issue MIPS Architecture Processor

第七届“龙芯杯”全国大学生计算机系统能力培养大赛

龚言龙 程家骏 刘思辰

2023 年 8 月



華中科技大學

HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY



# Table of Contents

## 1 Core 架构设计

- ▶ Core 架构设计
- ▶ 操作系统支持
- ▶ 性能优化之路
- ▶ 总结致谢



# MBoom 总体架构

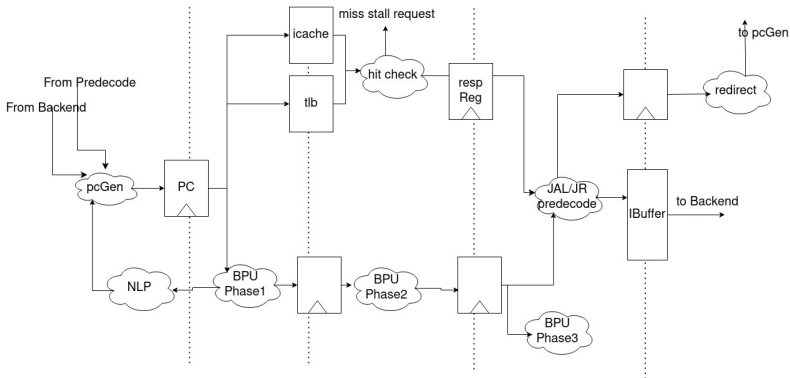
## 1 Core 架构设计

- 乱序双发 CPU。
  - 最多两条指令取指、解码、重命名、分发、提交。
  - 最多四条指令同时执行。
- 16KiB 指令缓存、8KiB 数据缓存。
- 8 项全相连 TLB。



# 前端总体架构

## 1 Core 架构设计



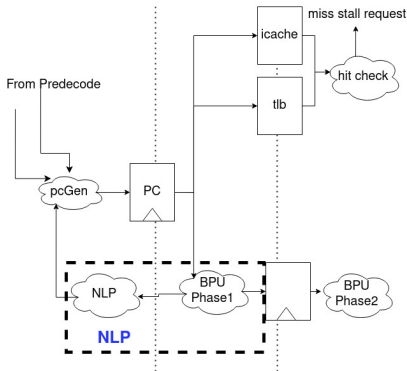


## 分支预测: NLP

### 1 Core 架构设计

单周期分支预测器，减少气泡。

- BTB: 与 BPU 共用，用于预测当前 PC 是否是分支指令，并提供分支跳转地址。
- BIM: 简单二位饱和分支预测器，用于预测分支指令是否跳转。

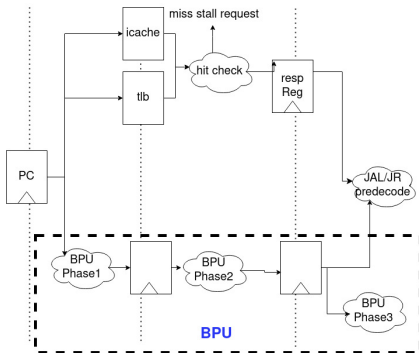




# 分支预测: BPU

## 1 Core 架构设计

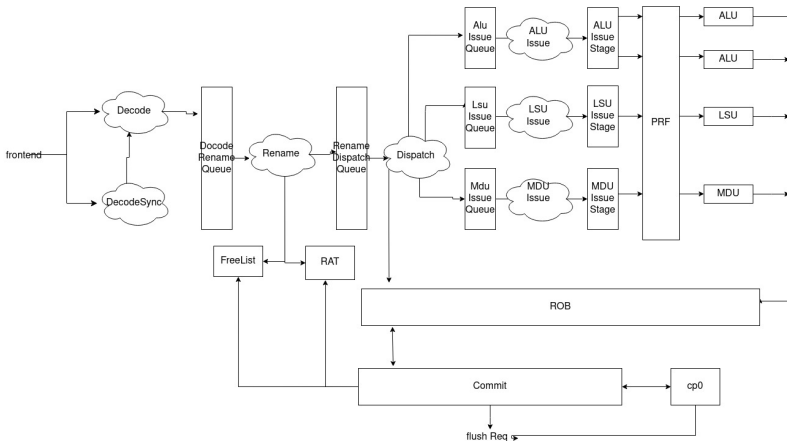
- 由 BTB、BHT、PHT、RAS 四部分组成。
  - BTB: 提供分支类型及静态分支地址。
  - BHT: 提供分支跳转历史。
  - PHT: 位拼接法寻址，二位饱和计数器。
  - RAS: 32 项带计数器地址返回栈。
- 三周期分支预测器
  - 第一周期访问 BTB、BHT
  - 第二周期访问 PHT、RAS，并获取分支预测结果。
  - 第三周期进行 RAS 的 Push / Pop。





# 后端总体架构

## 1 Core 架构设计

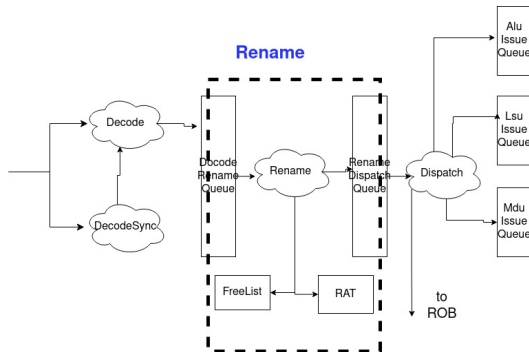




# 重命名阶段

## 1 Core 架构设计

- 基于统一 PRF 方式进行重命名。
- 使用 FreeList 管理空闲寄存器。
  - 采用双端口 FIFO 队列实现 FreeList。
  - commitPtr 保存提交时出队头指针。
- 使用 RMT (重命名映射表) 管理映射关系。
  - aRMT 管理提交时的映射关系。
- 需要 flush 时则分别用 commitPtr、aRMT 恢复 FreeList 和 RMT。



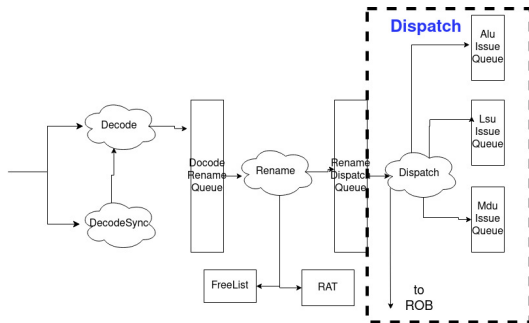




## 分发阶段

### 1 Core 架构设计

- 将指令分发到对应发射队列, 每周期最多分发两条指令。
  - 每周期最多分发两条 ALU 指令、一条 LSU 指令、一条 MDU 指令。
  - 遇到 cpo Write 指令、TLB 指令或 cache 指令会阻塞至 ROB 为空时才分发。
- 分发的同时将对应指令写入 ROB 中, 这是乱序 CPU 从顺序走向乱序的分隔点。



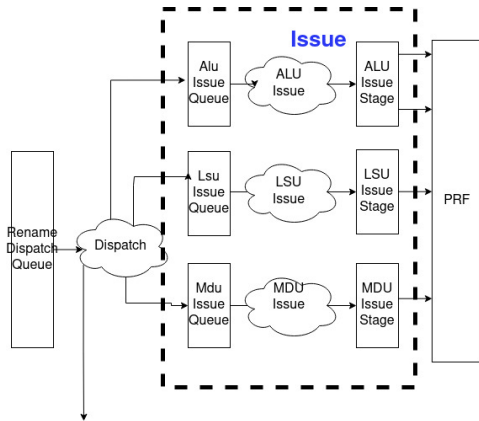


# 发射阶段

## 1 Core 架构设计

两个 ALU 使用集中式发射队列，MDU 和 LSU 采用分布式发射队列。

- 均使用压缩队列实现。ALU 发射队列为 10 项，其余两个为 8 项。
- AluIssue 支持全乱序发射，issueWindow 大小为 8。
- LsuIssue 支持伪乱序发射，issueWindow 大小为 4。
- MduIssue 仅支持顺序发射。

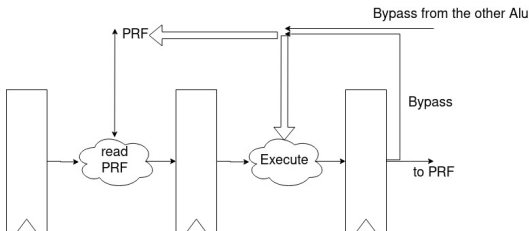




# ALU 运算单元

## 1 Core 架构设计

- 三级流水线运算。
  1. PRF: 接收 bypass 数据, 获取操作数。
  2. Execute: 接收 bypass 数据, 并执行整数计算。
  3. WriteBack: 写回数据, 激活 bypass 通路, 并通知 ROB。
- 支持 Alu 之间的完全 bypass, 整数指令之间的 RAW 相关性没有气泡。



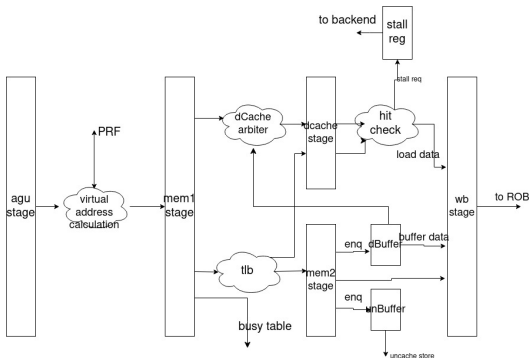


# LSU 访存单元

## 1 Core 架构设计

- 四级流水线访存

1. AGU: 获取访存虚拟地址。
2. Mem1: 发起 TLB 访问请求, load 指令参与访 DataCache 仲裁。
3. Mem2(Cache 缺失或 UnCache Load 会阻塞该阶段): Load 指令获取到 Load 数据, Store 指令按照访存属性入队。
4. WriteBack: 数据写回 PRF, 并通知 ROB。

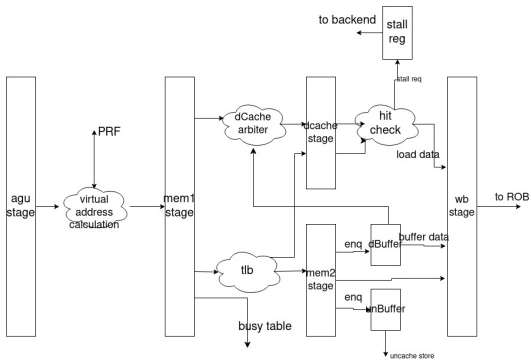




# LSU 访存单元

## 1 Core 架构设计

- 支持推测唤醒，将 load-use 减少至两个周期。
  - mem1 阶段发起唤醒信号，若 mem2 阶段检测到 cache 缺失或者 unCache Load 则阻塞所有指令的执行流水线。
- dBuffer 大小为 8 项，unBuffer 大小为 4 项，尽可能减少由于 store 指令造成的阻塞。



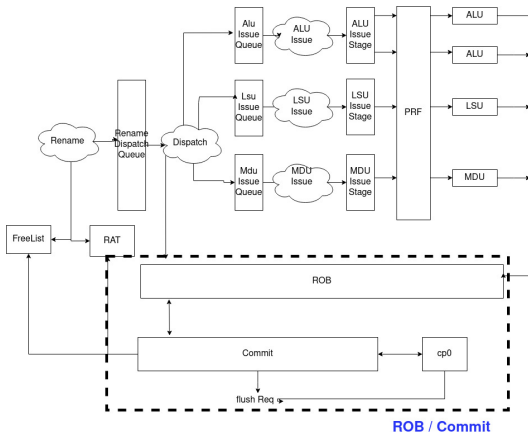


# ROB 及 Commit

## 1 Core 架构设计

### 用于顺序提交指令

- ROB 最多存放 32 项指令，每周期支持两条指令提交。
- 跳转指令和延迟槽同时提交，
  - 分支预测训练正确性。
  - 方便分支预测失败恢复状态。
- 中断在最早未提交指令可以提交时再处理。





# 缓存设计

## 1 Core 架构设计

### 指令缓存:

- 四路组相连, 每行 32 字节, 16KiB
- VIPT
- 采用 PLRU 替换算法。
- 实现为二阶段 Cache。缓存命中有一个周期延迟。

### 数据缓存:

- 二路组相连, 每行 32 字节, 8KiB
- VIPT
- 采用 PLRU 替换算法。
- 实现为二阶段 Cache。缓存命中有一个周期延迟。



# Table of Contents

## 2 操作系统支持

- ▶ Core 架构设计
- ▶ 操作系统支持
- ▶ 性能优化之路
- ▶ 总结致谢





# 引导程序

## 2 操作系统支持

- uboot 2023.8
- 移植处理
  - 根据时钟与外设情况编写设备树。
  - 关闭指令支持 (branch-likely、LL/SC)

```
U-Boot 2023.10-rc2+ (Aug 16 2023 - 07:21:45 +0000)

DRAM: 128 MiB
emalite ethernet@bfff0000: eth_phy_binds_nodes: mdio subnode found!
emalite ethernet@bfff0000: * Found child node: 'phy@1'
emalite ethernet@bfff0000: - bound phy device: 'phy@1'
Core: 5 devices, 5 uclasses, devicetree: separate
Loading Environment from <NULL>... OK
In: serial@bfe40000
Out: serial@bfe40000
Err: serial@bfe40000
Net: EMACLite: bff00000, phyaddr 1, 1/1
eth0: ethernet@bfff0000
Hit any key to stop autoboot: 0
ethernet@bfff0000 Waiting for PHY auto negotiation to complete.. done
BOOTP broadcast 1
DHCP client bound to address 10.0.0.177 (10 ms)
** Warning: no boot file name; using '0A0000B1.img'
Using ethernet@bfff0000 device
TFTP from server 10.0.0.1; our IP address is 10.0.0.177
Filename '0A0000B1.img'.
Load address: 0x84000000
Loading: #####
```



# Linux 移植

## 2 操作系统支持

- Linux v6.3.9
- 移植处理：
  - 根据时钟与外设情况编写设备树，同步更改编译选项。
  - 向 Kconfig 构建系统中添加所需源文件。
  - 关闭指令支持 (浮点、branch-likely、LL/SC)。
- 指令集处理：
  - PREF、SYNC、WAIT 等实现为 NOP。
  - BusyBox 中 LL/SC，内核会识别模拟。

```
# MIPS32 ISA - Search (p5 of 19)
Coprocessor (an essential part of the processor that is
implementation-defined in MIPS I-V), CP1 is an optional
floating-point unit (FPU) ... See more
Application-specific extensions
Registers See more
Calling conventions
MIPS has had several calling conventions, especially on the 32-bit
platform.
The O32 ABI is the most commonly-used ABI, owing to its status as
the original System V ABI for MIPS. It is strictly stack-based,
with only four registers $a0-$a3 available to pass ... See more
Simulators
Open Virtual Platforms (OVP) includes the freely available for
non-commercial use simulator OVPsim, a library of models of
processors, peripherals and platforms, and APIs which enable users
to develop their own models. The models in the library are open
source, ... See more
Design
MIPS is a modular architecture supporting up to four coprocessors
(CP0/1/2/3). In MIPS terminology, CP0 is the System Control
-- press space for next page --
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /search [delete]=history list
```



# Table of Contents

## 3 性能优化之路

- ▶ Core 架构设计
- ▶ 操作系统支持
- ▶ 性能优化之路
- ▶ 总结致谢



## 性能优化进程

### 3 性能优化之路

优化方式	IPC 比值	提升
朴素乱序多发流水线	29.597	\
分支预测改用三周期	30.634	3.50%
LSU 推测唤醒	34.264	11.85%
消除乘法阻塞	34.812	1.60%
压缩访存流水线	36.671	5.34%
nlp 减少分支预测气泡	37.419	1.38%
alu 完全 bypass	37.873	1.21%

性能优化主要关注点:

1. 分支预测

2. 访存优化

这也是 CPU 设计两大痛点。



## 最终 IPC 比值性能

### 3 性能优化之路

测试名	IPC 比值	分支预测正确率
bitcount	41.862	93.4%
bubble_sort	<b><u>35.249</u></b>	<b><u>83.9%</u></b>
coremark	<b><u>32.074</u></b>	<b><u>89.2%</u></b>
crc32	46.554	98.9%
dhrystone	37.342	97.3%
quick_sort	<b><u>27.056</u></b>	<b><u>81.4%</u></b>
select_sort	41.781	96.5%
sha	45.277	98.0%
stream_copy	37.631	98.9%
stringsearch	38.312	95.7%

- 综合 IPC 比值 37.873。
- 总体 IPC 不高。
  - 结果前传未做到位。
- IPC 比值方差很大。
  - 流水线很长，分支预测失败惩罚很大。

还有什么优化空间吗？



## checkpoint

### 3 性能优化之路

测试名	优化前	优化后	提升率
bitcount	41.862	42.295	1.03%
bubble_sort	35.249	38.257	8.53%
coremark	32.074	33.983	5.95%
crc32	46.554	46.762	0.45%
dhystone	37.342	38.018	1.81%
quick_sort	27.056	32.643	20.65%
select_sort	41.781	42.170	0.93%
sha	45.277	45.523	0.54%
stream_copy	37.631	37.720	0.24%
stringsearch	38.312	38.683	0.97%

- IPC 比值从 37.873 升至 39.366，总提升率 3.9%。
- 采用 4 项 checkpoint 缓存状态。
- 分离 BRU 用于提前唤醒。
- 上板功能、性能测试通过，跑系统有问题，且资源占用极大影响频率，未进入最终提交版本。



## 流水线前传 (通过 PRF 实现完全的 Alu 结果的前传)

### 3 性能优化之路

测试名	优化前	优化后	提升率
bitcount	41.862	42.870	2.41%
bubble_sort	35.249	36.757	4.28%
coremark	32.074	33.190	3.48%
crc32	46.554	46.952	0.85%
dhystone	37.342	37.923	1.56%
quick_sort	27.056	28.034	3.61%
select_sort	41.781	42.570	1.89%
sha	45.277	46.486	2.67%
stream_copy	37.631	37.996	0.97%
stringsearch	38.312	39.192	2.30%

- IPC 比值从 37.873 升至 38.780, 总提升率 2.4%。
- 主要提升的是 ALU 到 LSU 的唤醒效率。
- 决赛提交之后实现。



# Table of Contents

## 4 总结致谢

- ▶ Core 架构设计
- ▶ 操作系统支持
- ▶ 性能优化之路
- ▶ 总结致谢





# Good Luck!

## 4 总结致谢

- 经过近半年的努力我们终于实现了我们最初的梦想: 在自己的 CPU 上跑 Linux。
- 仍留有不少遗憾, 这与我们后期开发效率低下很有关系。此次经历让我们认识到用好 verilator、GEM5 等工具的重要性。

感谢周健老师长期以来的指导以及龙芯杯群友的帮助!



# MBOOM: An Out-of-order Dual-issue MIPS Architecture Processor

*Thank you for listening!*  
*Any questions?*