Marvelous MIPS Machine (MMM)

第六届“龙芯杯”全国大学生计算机系统能力培养大赛答辩

潘子曰，王晶晶，汤尧，叶泽凯

2023 年 3 月 12 日

浙江大学 Capable Chip Creators (CCC)
指导老师 常瑞

Table of contents

简介

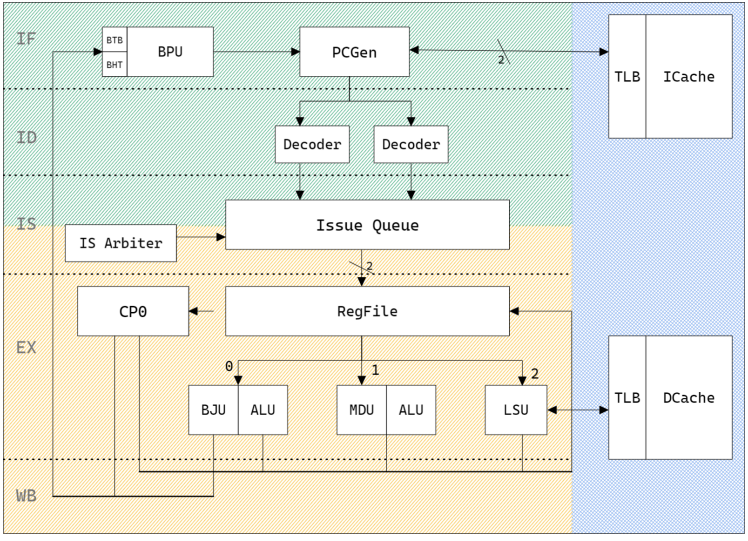
MMM 处理器简介

MMM (Marvelous MIPS Machine) 是一款基于 Chisel 语言开发的 MIPS32 架构处理器。MMM 处理器在设计上采用 KISS (Keep It Simple, Stupid) 原则，架构简单，可配置强。

1. 精简的核内设计
2. 软件辅助的处理器验证
3. 优异的性能体现

MMM 处理器的设计与实现

MMM 整体设计图


IF
ID
IS
EX
WB
BTB
BHT
BPU
PCGen
2
TLB
ICache
Decoder
Decoder
IS Arbiter
Issue Queue
2
CP0
RegFile
0
1
2
BJU
ALU
MDU
ALU
LSU
TLB
DCache

设计亮点

设计亮点

频率 QMC 算法优化的译码器设计

IPC BHT+BTB 分支预测及优化

验证 soc-simulator和基于 QEMU 的差分测试框架

Profiling 基于 soc-simulator 和 chisel 语法特性的性能分析模块

QMC 算法优化的译码器设计

- Quine–McCluskey 译码优化器
- 等效卡诺图，用于最小化布尔函数
- 借鉴 rocket-chip，使用 chisel3 库中的 QMCMinimizer 实现
- 获得约 10% 的时序优化

BHT+BTB 分支预测及优化

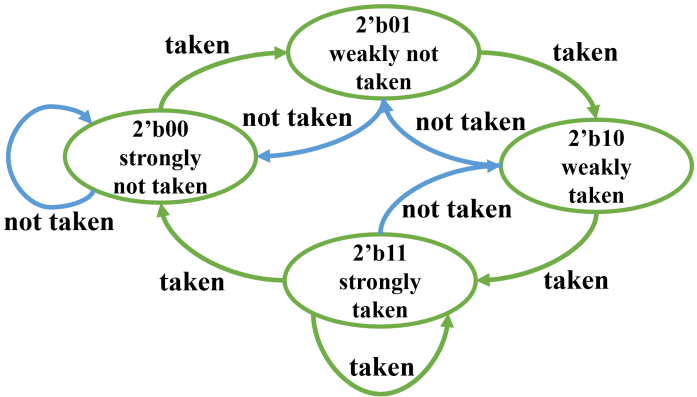


图 1: BHT 预测向量

BHT+BTB 分支预测及优化

BTB 的更新唯一来自于 WB 提交指令阶段。当跳转或分支指令的执行目标地址和预测的目标地址发生冲突时，就会更新 BTB 中存储的分支跳转目标历史。

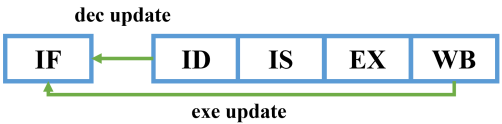


图 2: BHT 更新机制

BHT+BTB 分支预测及优化

失败的分支预测方案

- RAS
- Gshare 分支预测器
- Splitted BlockRAM
- 为双发射的两条指令都提供分支预测

soc-simulator[1]

- 基于 Verilator 的仿真框架，模拟 AXI 协议和龙芯的 confreg
- 支持 golden_trace，易于更换 testbench
- 对多发射结构友好
- 相比 XSim 运行效率高

[1]由重庆大学所以延迟槽会消失对不队设计

基于 QEMU 的差分测试框架

- 自主设计的基于 QEMU 的差分测试框架
- 通过 remote serial protocol 协议与 QEMU 交互
- 修改 QEMU 源码，同步中断、TLB 提升测试效率

```
SSwwiittcchhiinngg  ttoo  ccllooccckkssoouurrccee  MMIIPPSS

Error in $a0, QEMU 80277130, DUT 3c1b8062
------------------QMEU pc trace-------------------
QEMU pc: [0x801d2c00]
QEMU pc: [0x801d2c04]
QEMU pc: [0x801d2c08]
QEMU pc: [0x801d2c0c]
QEMU pc: [0x801d2bd4]   <---- jump
QEMU pc: [0x801d2bd8]
QEMU pc: [0x801d2bdc]
QEMU pc: [0x801d2be0]
QEMU pc: [0x801d2be4]
QEMU pc: [0x801d2be8]
QEMU pc: [0x801d2bf8]   <---- jump
QEMU pc: [0x801d2bfc]
QEMU pc: [0x801d2c00]
```

图 3: 基于 QEMU 的差分测试框架运行结果

性能分析 (Profiling) 部件

在 soc-simulator 的基础上，MMM 处理器使用 chisel 语法特性添加了多个性能分析 (Profiling) 部件，用于快速分析处理器运行时的状态和瓶颈。

```
if (conf.TraceBPUAcc) {
  val brInstCount     = RegInit(0.U(64.W))
  val brMisPredCount = RegInit(0.U(64.W))
  val redirectCount  = RegInit(0.U(64.W))
  val brInst = RegEnable(isExPCBr || isExPCJump, !bubble_w)
  brInstCount := brInstCount + (wbInstsValid(0) && brInst).asUInt
  brMisPredCount := brMisPredCount + (brInst && wbReBranch && !wfds)
  redirectCount        := redirectCount + (wbReBranch && !wfds).asUInt

  when (VecInit(Range.exclusive(0, 3).map(i => RegNext(wbInsts(i).pc))).exists(x => x === conf.EndAddr.U)) {
    val bpuPredAccInt = (brInstCount - brMisPredCount) * 100.U / brInstCount
    val bpuPredArccFrac = ((brInstCount - brMisPredCount) * 10000.U / brInstCount) % 100.U
    printf("bpu predict accuracy: %d.%d%%\tbranch inst: %d\tbranch mispredict: %d\tredirect: %d\n",
      bpuPredAccInt, bpuPredArccFrac, brInstCount, brMisPredCount, redirectCount)
  }
}
```

性能分析 (Profiling) 部件

```
bitcount:
            329974 insts,              240730 cycles,                2690 d$ stalls,
bpu predict accuracy:                   89. 22%       branch inst:               48689       brar
dcache access:               34997     read miss:                 22 write miss:
icache access:              236201     miss:                 122      miss rate:
score: 1c9ef, total_ticks = 482459, stall_clk = 58225, has_commit = 183005, dual_commit = 93377, dua

bubble_sort:
           1709078 insts,              1614259 cycles,               23676 d$ stalls,
bpu predict accuracy:                   68.  5%       branch inst:              241077       brar
dcache access:               613557     read miss:                517 write miss:
icache access:              1610499     miss:                 101      miss rate:
score: c3c73, total_ticks = 3229518, stall_clk = 788659, has_commit = 826100, dual_commit = 81401, d

coremark:
           4087263 insts,              3621890 cycles,                9821 d$ stalls,
bpu predict accuracy:                   87.  1%       branch inst:              775780       brar
dcache access:               827305     read miss:                 97 write miss:
icache access:              3403285     miss:                6216      miss rate:
score: 1b983b, total_ticks = 7244780, stall_clk = 1568466, has_commit = 2053924, dual_commit = 5174

crc32:
           2562669 insts,              2245797 cycles,                4352 d$ stalls,
bpu predict accuracy:                   93.  9%       branch inst:              525163       brar
dcache access:               531725     read miss:                 74 write miss:
icache access:              2241998     miss:                  98      miss rate:
score: 1118c9, total_ticks = 4492594, stall_clk = 751590, has_commit = 1494707, dual_commit = 57894,
```

图 4: 性能分析结果展示

系统展示

操作系统启动

- 裁剪指令（BL, soft float 等）
- TLB 相关指令（TLBP、TLBR、TLBWI、TLBWR 等）
- 原子指令（LL、SC）
- 其他功能指令（MOVN、MADD、LWL、TGE、CLZ、CLO 等）
- 协处理器 0（EntryHi, EntryLo, Context, Index, Ebase 等）
- 通过 PMON 加载 uCore、Linux 操作系统并启动

操作系统启动

```
dmfe0 at localbus0: address 00:98:76:64:32:19
in if attach
loopdev0 at mainbus0out configure
====before init ps/2 kbd
devconfig done.
ifinit done.
domaininit done.
init_proc....
MSTI
SYMI
SBDE

 * PMON2000 Professional *
Configuration [FCR,EL,NET]
Version: PMON2000 2.1 (ls1b) #21: Wed Aug 10 15:00:57 CST 2022 commit 
Supported loaders [srec, elf, bin]
Supported filesystems [mtd, net, fs/yaffs2, fat, fs, disk, socket, tty,
This software may be redistributed under the BSD copyright.
Copyright 2000-2002, Opsycon AB, Sweden.
Copyright 2005, ICT CAS.
CPU unidentified @ 99.99 MHz / Bus @ 99.99 MHz
Memory size 128 MB (128 MB Low memory,   0 MB High memory) .
Primary Instruction cache size 2kb (16 line, 2 way)
Primary Data cache size 2kb (16 line, 2 way)

BEV1
BEV2
BEV3
BEV0
BEV in SR set to zero.

NAND DETE
NAND device: Manufacturer ID: 0xec, Chip ID: 0xf1 (Samsung NAND 128MiB
NAND_ECC_NONE selected by board driver. This is not recommended !!
Scanning device for bad blocks
Bad eraseblock 895 at 0x06fe0000
NANDFlash info:
erasesize       131072 B
writesize       2048 B
oobsize         64 B
PMON> 
```

(a) PMON

```
Entry address is 80000000
PMON> g
++setup timer interrupts
Initrd: 0x8002c000 - 0x801267ff, size: 0x000fa000, magic: 0x2f8dbe2a
(THU.CST) os is loading ...

Special kernel symbols:
  entry  0x800000f8 (phys)
  etext 0x8002A420 (phys)
  edata 0x80126000 (phys)
  end   0x80129B10 (phys)
Kernel executable memory footprint: 1022KB
memory management: buddy_pmm_manager
memory map:
  [80000000, 84000000]

freemem start at: 801AA000
free pages: 00003E56
## 00000020
checking pmm, errors can be ignored.
Page allocation failed, possibly due to OOM or uninitialized page area.
Page allocation failed, possibly due to OOM or uninitialized page area.
check_alloc_page() succeeded!
pmm check passed!
check_pgdir() succeeded!
check_boot_pgdir() succeeded!
-------------------- BEGIN --------------------
-------------------- END --------------------
check_slab() succeeded!
kmalloc_init() succeeded!
check_vma_struct() succeeded!
check_pgfault() succeeded!
check_vmm() succeeded.
sched class: RR_scheduler
ramdisk_init(): initrd found, magic: 0x2f8dbe2a, 0x000007d0 secs
sfs: mount: 'simple file system' (225/25/250)
vfs: mount disk0.
kernel_execve: pid = 2, name = "sh".
user sh is running!!!
$ 
```

(b) uCore

```
nsgmni has been set to 240
alg: No test for stdrng (krng)
Block layer SCSI generic (bsg) driver version 0.4 loaded (major 253)
io scheduler noop registered
io scheduler anticipatory registered
io scheduler deadline registered
io scheduler cfq registered (default)

ls1x_fb init
------------[ cut here ]------------
WARNING: at mm/page_alloc.c:1806 __alloc_pages_nodemask+0x1c4/0x648()
Modules linked in:
Call Trace:
[<8020aba8>] dump_stack+0x8/0x34
[<80233b04>] warn_slowpath_common+0x70/0xb0
[<80292278>] __alloc_pages_nodemask+0x1c4/0x648
[<80292714>] __get_free_pages+0x18/0x74
[<802186f4>] dma_alloc_coherent+0x54/0xd8
[<806c2198>] ls1x_fb_probe+0x234/0x484
[<80451b84>] driver_probe_device+0x104/0x2b0
[<80451dec>] __driver_attach+0xbc/0xc4
[<80451 1dc>] bus_for_each_dev+0x68/0xd0
[<80450960>] bus_add_driver+0x240/0x2e4
[<80452150>] driver_register+0x88/0x184
[<806c249c>] ls1x_fb_init+0xb4/0x1ac
[<8020e25c>] do_one_initcall+0x3c/0x210
[<806b4374>] kernel_init+0x70/0x160
[<802100d4>] kernel_thread_helper+0x10/0x18

---[ end trace 4eaa2a86a8e2da22 ]---
ls1x-fb: probe of ls1x-fb failed with error -12
Serial: 8250/16550 driver, 4 ports, IRQ sharing disabled
serial8250.0: ttyS0 at MMIO 0x1fe40000 (irq = 2) is a ST16650
console [ttyS0] enabled, bootconsole disabled
console [ttyS0] enabled, bootconsole disabled
```

(c) Linux

总结与展望

总结与展望

- “龙芯杯”比赛中大量的实践经验
- chisel 开发和软件验证的良好配合
- 设计痛点：延迟槽带来的设计挑战
- 展望：LoongArch 挑战赛 + 更好的验证框架 + 更简洁的设计

谢谢专家老师，敬请指正！

成员介绍

潘子曰　大四｜**验证框架**、CP0、Backend、uCore、Linux

王晶晶　大二｜BPU、Frontend、PMON

汤尧　大二｜Decoder、Cache、Linux

叶泽凯　大二｜TLB、CP0、Backend、uCore

QMCMinimizer

```
def apply(addr: UInt, default: BitPat, mapping: Iterable[(BitPat, BitPat)]): UInt =
  chisel3.util.experimental.decode.decoder(QMCMinimizer, addr, TruthTable(mapping, default))
def apply(addr: UInt, default: Seq[BitPat], mappi
  val nElts = default.size
  require(mappingIn.forall(_._2.size == nElts),
    s"All Seq[BitPat] must be of the same length,
  )

  val elementsGrouped = mappingIn.map(_._2).trans
  val elementWidths = elementsGrouped.zip(default
    (default :: elts.toList).map(_.getWidth).max
  }
  val resultWidth = elementWidths.sum
```

```
chisel3.util.experimental.decode
object QMCMinimizer
extends Minimizer
```

A `Minimizer` implementation to use Quine-Mccluskey algorithm to minimize the `TruthTable`.
This algorithm can always find the best solution, but is a NP-Complete algorithm, which means, for large-scale `TruthTable` minimization task, it will be really slow, and might run out of memory of JVM stack.
In this situation, users should consider switch to `EspressoMinimizer`, which uses heuristic algorithm providing a sub-optimized result.

sbt: edu.berkeley.cs:chisel3_2.13:3.5.3:jar (chisel3_2.13-3.5.3.jar)

图 6: MMM 在 Decoder 中调用 QMCMinimizer

Split BlockRAM

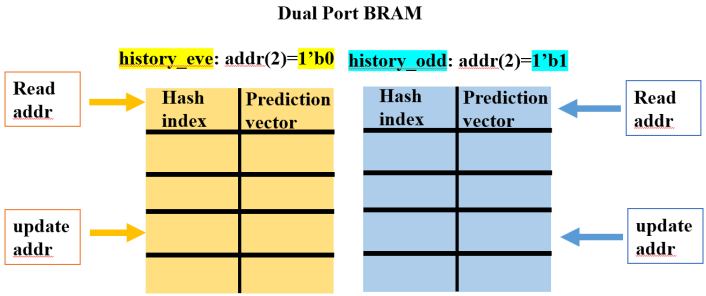


图 7: split BRAM **的示意图**

- **顺序双发隐藏特性**：**发射的两条指令地址必然连续** pc & pc + 4
- addr(2) **存在差异，简化逻辑**
- **保证了** split bram **的逻辑简洁性与可行性**

Split BlockRAM

```
// addr(2) = 0
history_eve.io.clk   := clock
history_eve.io.rst   := reset
history_eve.io.addra := Mux(io.req.next_line(2),getHashSplitIndex(io.req.next_line+4.U),getHashSplitIndex(io.req.next_line))
history_eve.io.addrb := getHashSplitIndex(waddr)
history_eve.io.dina  := DontCare
history_eve.io.dinb  := wdata
history_eve.io.wea   := false.B
history_eve.io.web   := update && (!waddr(2))

// addr(2) = 1
history_odd.io.clk := clock
history_odd.io.rst := reset
history_odd.io.addra := Mux(io.req.next_line(2), getHashSplitIndex(io.req.next_line), getHashSplitIndex(io.req.next_line + 4.U))
history_odd.io.addrb := getHashSplitIndex(waddr)
history_odd.io.dina := DontCare
history_odd.io.dinb := wdata
history_odd.io.wea := false.B
history_odd.io.web := update && waddr(2)
```

图 8: split BRAM **的部分代码**

Split BlockRAM

BITCOUNT

```
bitcount started!
          329587 insts,          240751 cycles,          2694 d$ stalls,          4309 i$ stalls,          105 mdu stalls,          838 common stalls
bpu predict accuracy:                90. 91%  branch inst:          48599  branch mispredict:          4414 redirect:          10512
dcache access:          34960  read miss:          22 write miss:          8  mmio:          44  miss rate:          0.  8%
icache access:          236442  miss:          116  miss rate:          0.  4%
bitcount finished!
```

```
bitcount started!
          329974 insts,          240730 cycles,          2690 d$ stalls,          4529 i$ stalls,          259 mdu sta
lls,          857 common stalls
bpu predict accuracy:                89. 22%  branch inst:          48689  branch mispredict:          5246 redirect:
          10600
dcache access:          34997  read miss:          22 write miss:          8  mmio:          45  miss
rate:          0.  8%
icache access:          236201  miss:          122  miss rate:          0.  5%
bitcount finished!
```

BUBBLESORT

```
bubble_sort started!
          1708712 insts,          1585078 cycles,          4248 d$ stalls,          3643 i$ stalls,          0 mdu stalls,          834 common stalls
bpu predict accuracy:                69. 66%  branch inst:          240976  branch mispredict:          73104 redirect:          73173
dcache access:          613516  read miss:          59 write miss:          34  mmio:          24  miss rate:          0.  1%
icache access:          1581435  miss:          95  miss rate:          0.  0%
bubble_sort finished!
```

```
bubble_sort started!
          1709078 insts,          1614259 cycles,          23676 d$ stalls,          3760 i$ stalls,          179 mdu sta
lls,          1042 common stalls
bpu predict accuracy:                68.  5%  branch inst:          241077  branch mispredict:          77020 redirect:
          77124
dcache access:          613557  read miss:          517 write miss:          77  mmio:          25  miss
rate:          0.  9%
icache access:          1610499  miss:          101  miss rate:          0.  0%
```

Split BlockRAM

COREMARK

```
coremark started!
4087046 insts,  3909014 cycles  265964 d$ stalls,  268369 i$ stalls,  103650 mdu stalls,  6743 common stalls
bpu predict accuracy:  86. 85%  branch inst:  775676  branch mispredict:  101928 redirect:  123799
dcache access:  827312  5611 write miss:  881  mmio:  44  miss rate:  0. 78%
icache access:  3640644  miss:  7592  miss rate:  0. 20%
coremark finished!
```

```
coremark started!
4087263 insts,  3621890 cycles  9821 d$ stalls,  218605 i$ stalls,  103050 mdu sta
lls,  2326 common stalls
bpu predict accuracy:  87. 1%  branch inst:  775780  branch mispredict:  100726 redirect:
126155
dcache access:  827305  read miss:  97 write miss:  88  mmio:  45  miss
rate:  0. 2%
icache access:  3403285  miss:  6216  miss rate:  0. 18%
coremark finished!
```

CRC32

```
crc32 started!
2563339 insts,  2213942 cycles  4357 d$ stalls,  3466 i$ stalls,  48520 mdu stalls,  846 common stalls
bpu predict accuracy:  95. 13%  branch inst:  525410  branch mispredict:  25536 redirect:  40385
dcache access:  531770  75 write miss:  10  mmio:  23  miss rate:  0. 1%
icache access:  2210476  miss:  89  miss rate:  0. 0%
crc32 finished!
```

```
crc32 started!
2562669 insts,  2245797 cycles  4352 d$ stalls,  3799 i$ stalls,  48724 mdu sta
lls,  929 common stalls
bpu predict accuracy:  93. 9%  branch inst:  525163  branch mispredict:  36263 redirect:
45367
dcache access:  531725  read miss:  74 write miss:  10  mmio:  24  miss
rate:  0. 1%
icache access:  2241998  miss:  98  miss rate:  0. 0%
crc32 finished!
```

Split BlockRAM

QUICKSORT

```
quick_sort started!
          1713518 insts,          1640021 cycles,          15018 d$ stalls,          8312 i$ stalls,          0 mdu stalls,          931 common stalls
bpu predict accuracy:          79. 47%          branch inst:          314320          branch mispredict:          64514 redirect:          71772
dcache access:          382814          261 write miss:          134          mmio:          24          miss rate:          0. 10%
icache access:          1631709          miss:          232          miss rate:          0.  1%
quick_sort finished!
```

```
quick_sort started!
          1713894 insts,          1800569 cycles,          163926 d$ stalls,          8650 i$ stalls,          179 mdu sta
lls,          1212 common stalls
bpu predict accuracy:          78. 44%          branch inst:          314425          branch mispredict:          67783 redirect:
          78065
dcache access:          382855          read miss:          1840 write miss:          1314          mmio:          25          miss
rate:          0. 82%
icache access:          1791919          miss:          239          miss rate:          0.  1%
quick_sort finished!
```

STRINGSEARCH

```
stringsearch started!
          1189252 insts,          990396 cycles,          39641 d$ stalls,          3905 i$ stalls,          0 mdu stalls,          952 common stalls
bpu predict accuracy:          93. 29%          branch inst:          278938          branch mispredict:          18715 redirect:          24684
dcache access:          328204          418 write miss:          365          mmio:          24          miss rate:          0. 23%
icache access:          986491          miss:          107          miss rate:          0.  1%
stringsearch finished!
```

```
stringsearch started!
          1189643 insts,          1120877 cycles,          143923 d$ stalls,          4521 i$ stalls,          179 mdu st
lls,          1138 common stalls
bpu predict accuracy:          90. 29%          branch inst:          279047          branch mispredict:          27068 redirect:
          30686
dcache access:          328245          read miss:          1577 write miss:          1275          mmio:          25          miss
rate:          0. 86%
icache access:          1116356          miss:          119          miss rate:          0.  1%
stringsearch finished!
```