

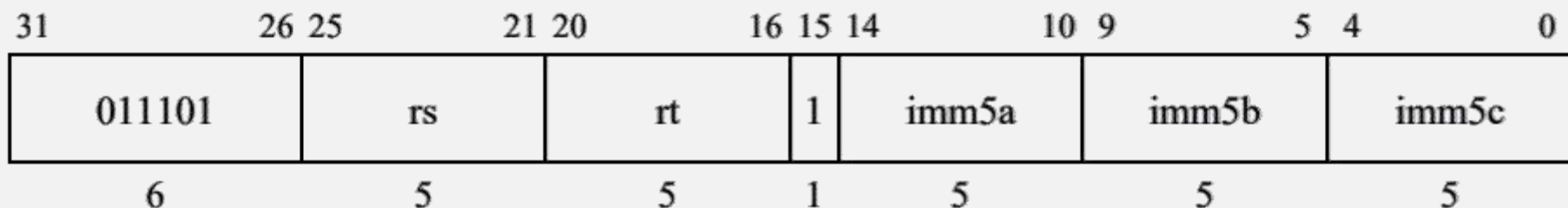


杭州电子科技大学  
HANGZHOU DIANZI UNIVERSITY

# NSCSCC 2023 PUA MIPS

叶剑豪 奚力丰 胡致尧

## RLWINM



汇编格式: rlwinm rt, rs, imm5a, imm5b, imm5c

指令功能: 带掩码的循环左移。

功能描述:

将 rs 寄存器里的 bit[31:0] 循环左移 imm5a 位, 随后根据 imm5b 和 imm5c 的值构建出由连续的 1 组成的掩码并与移位的结果按位与, 将结果写入到 rt 寄存器。imm5b 和 imm5c 表示连续的 1 的起始位置与结束位置, 例如: imm5b 与 imm5c 分别为 4 和 31, 表示掩码从第 5 比特到第 31 比特是连续的 1, 掩码为 0xFFFFFE0; 如果 imm5b 与 imm5c 分别为 30 和 4, 则表示掩码从第 31 比特到第 4 比特是连续的 1, 掩码为 0x8000001F。

## 1. Const.scala 添加进指令表

```
// 比赛增加的指令  
val EXE_RLWINM = 77.U(OP_WID.W)
```

## 2. Instruction.scala 添加指令识别码

```
def RLWINM = BitPat("b011101????????1????????????????")
```

## 3. Decoder.scala 通过ListLookup产生译码信号

```
RLWINM → List(INST_VALID, READ_ENABLE, READ_DISABLE, FU_ALU,  
              EXE_RLWINM, WRITE_ENABLE, WRA_T2, IMM_LSE, SINGLE_ISSUE),
```

## 4. ALU.scala 在运算单元中进行处理

## 不言自证 不证自明

```
val imm5a = src2(14, 10)
val imm5b = src2(9, 5)
val imm5c = src2(4, 0)
val mask = Wire(Vec(32, Bool()))
for (i ← 0 until (32)) {
  when((imm5c ≥ i.U(5.W) && i.U(5.W) > imm5b) ||
        (i.U(5.W) > imm5b && imm5b > imm5c) ||
        (i.U(5.W) ≤ imm5c && imm5b > imm5c)) {
    mask(i) := true.B
  }.otherwise {
    mask(i) := false.B
  }
}
val rlwinmv = (src1 << imm5a) | (src1 >> (32.U - imm5a))
val rlwinmask = mask.asUInt

io.result := MuxLookup(
  op,
  0.U,
  Seq(
    EXE_RLWINM → (rlwinmv & rlwinmask),
```

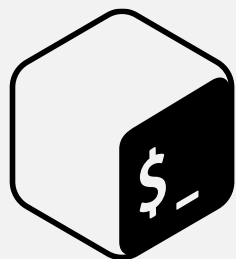


```
for i in [0..31] ←
  if (←
    (imm5c>=i and i>imm5b) or ←
    (i>imm5b and imm5b>imm5c) or ←
    (i<=imm5c and imm5b>imm5c)←
  ): ←
    mask[i]=1←
  else ←
    mask[i]=0←
GPR[rd] ← rotl(GPR[rj], imm5a) & mask←
```

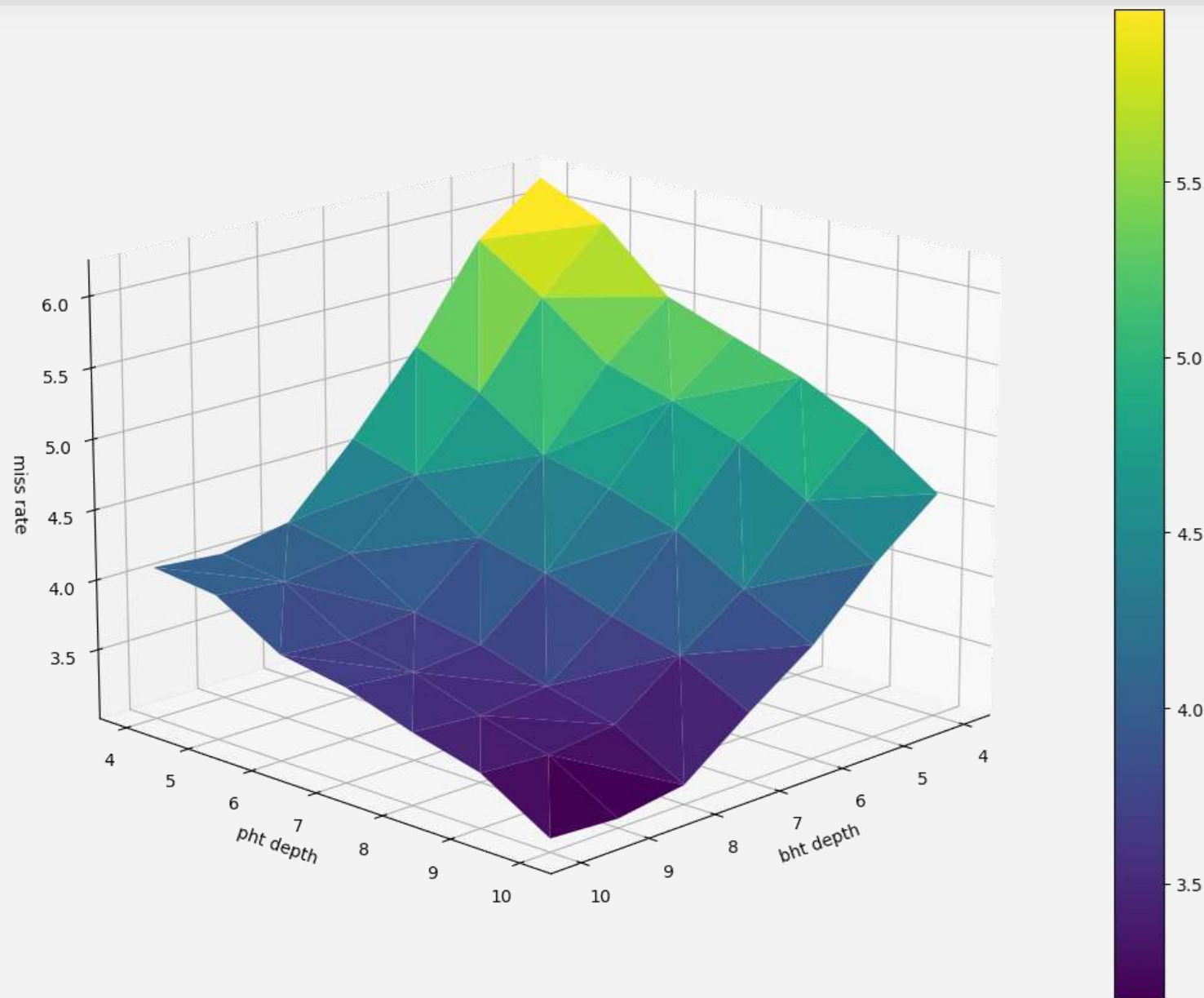
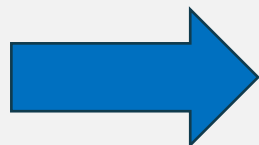
# 基于Verilator仿真器进行设计空间探索



杭州电子科技大学  
HANGZHOU DIANZI UNIVERSITY



简单Bash脚本





VS



- 修改顶层仿真**框架**
- 添加处理器核端口
- 编写**bash**测试脚本
- **自动生成**性能数据

20 min

vs

- 使用Vivado进行仿真
- **手动查看**信号数据

15 hour



## 多种参数任意组合

```
case class CpuConfig(  
  val build: Boolean = true,           // 是否为build模式  
  val hasCommitBuffer: Boolean = false, // 是否有提交缓存  
  val decoderNum: Int = 2,             // 同时访问寄存器的指令数  
  val commitNum: Int = 2,              // 同时提交的指令数  
  val fuNum: Int = 2,                  // 功能单元数  
  val instFetchNum: Int = 2,           // iCache取到的指令数  
  val instFifoDepth: Int = 8,  
  val writeBufferDepth: Int = 16,  
  val mulClockNum: Int = 2,  
  val divClockNum: Int = 8,  
  val branchPredictor: String = "adapt"  
)  
  
case class BranchPredictorConfig(  
  val bhtDepth: Int = 8,  
  val phtDepth: Int = 10,  
)  
  
case class CacheConfig(  
  nway: Int = 2,  
  nbank: Int = 8,  
  nset: Int,  
  bankWidth: Int, // bytes per bank  
) {  
  val indexWidth      = log2Ceil(nset)  
  val bankIndexWidth  = log2Ceil(nbank)  
  val bankOffsetWidth = log2Ceil(bankWidth)  
  val offsetWidth     = bankIndexWidth + bankOffsetWidth  
  val tagWidth        = 32 - indexWidth - offsetWidth  
  val tagvWidth       = tagWidth + 1  
  val bankWidthBits   = bankWidth * 8  
  val burstSize       = 16  
  val ninst            = 4  
  require(isPow2(nset))  
  require(isPow2(nway))  
  require(isPow2(nbank))  
  require(isPow2(bankWidth))  
}
```

```
class InstFifo(implicit val config: CpuConfig) extends Module {  
  val io = IO(new Bundle {  
    val do_flush = Input(Bool())  
  
    val icache = Flipped(Vec(config.instFetchNum, Decoupled(new FifoUnit())))  
    val decoder = Vec(config.decoderNum, Decoupled(new FifoUnit()))  
  })  
}
```

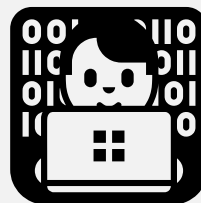
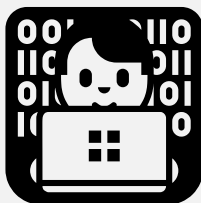
前端取指



指令队列



后端执行





- 大框架上通过整体暂停实现集中控制
- 小框架中通过握手信号实现分布控制

```
class Mul(implicit val config: CpuConfig) extends Module {  
  val io = IO(new Bundle {  
    val src1      = Input(UInt(DATA_WID.W))  
    val src2      = Input(UInt(DATA_WID.W))  
    val signed    = Input(Bool())  
    val valid     = Input(Bool())  
  
    val ready     = Output(Bool())  
    val result    = Output(UInt(HILO_WID.W))  
  })  
}
```

取指

写回

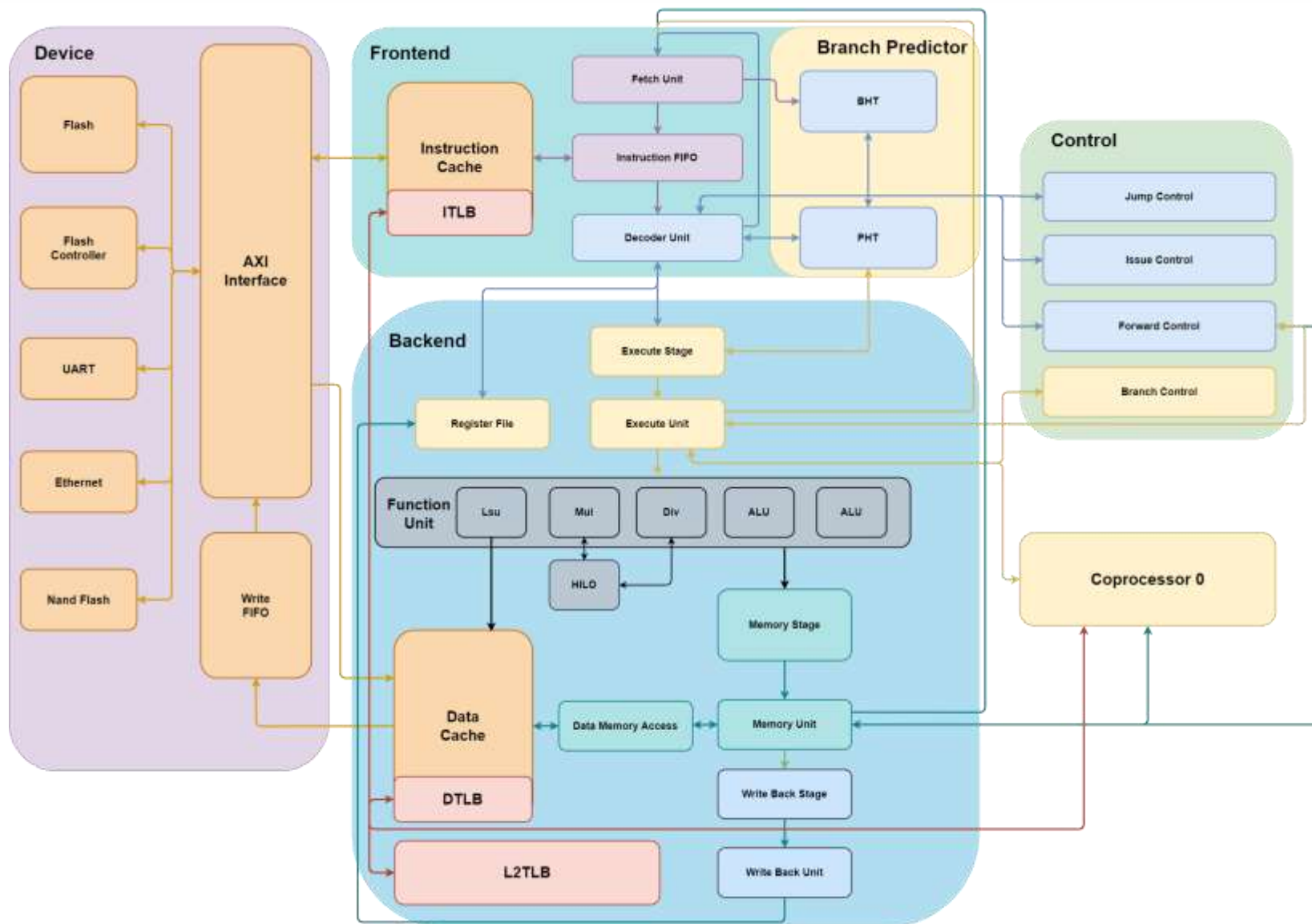
运算

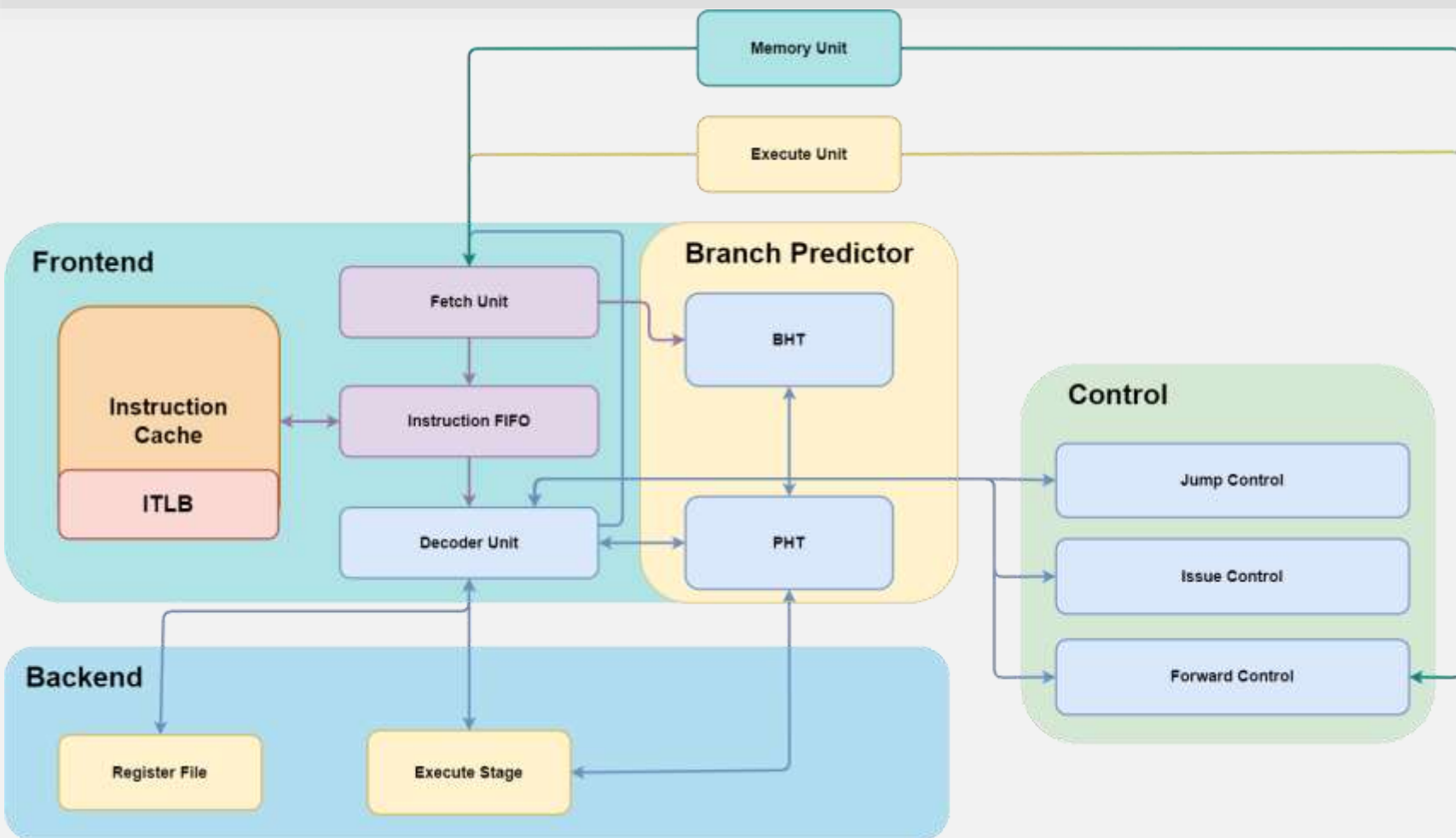
# Powerful Ultra Architecture MIPS



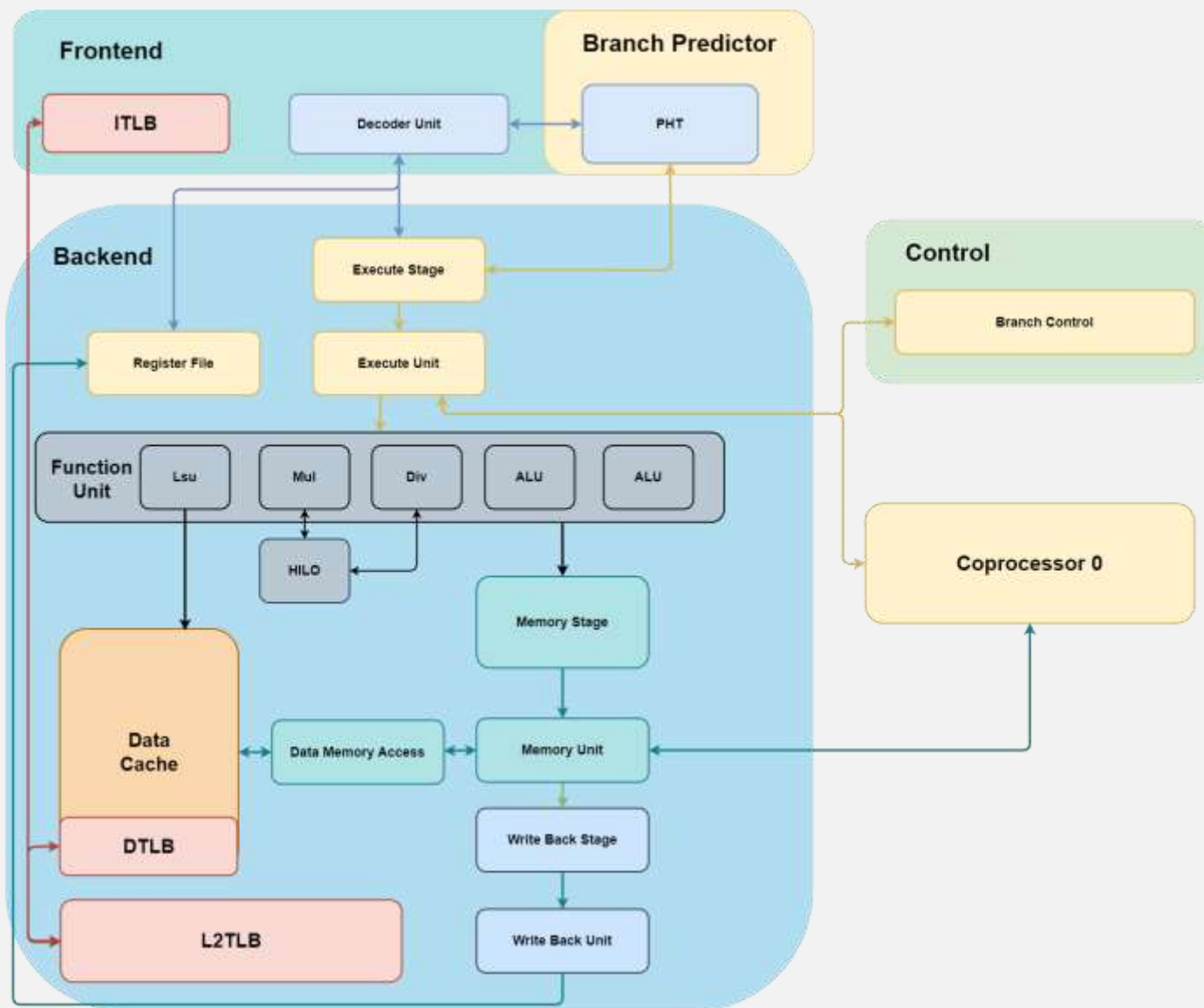
杭州电子科技大学  
HANGZHOU DIANZI UNIVERSITY

## 处理器总体架构



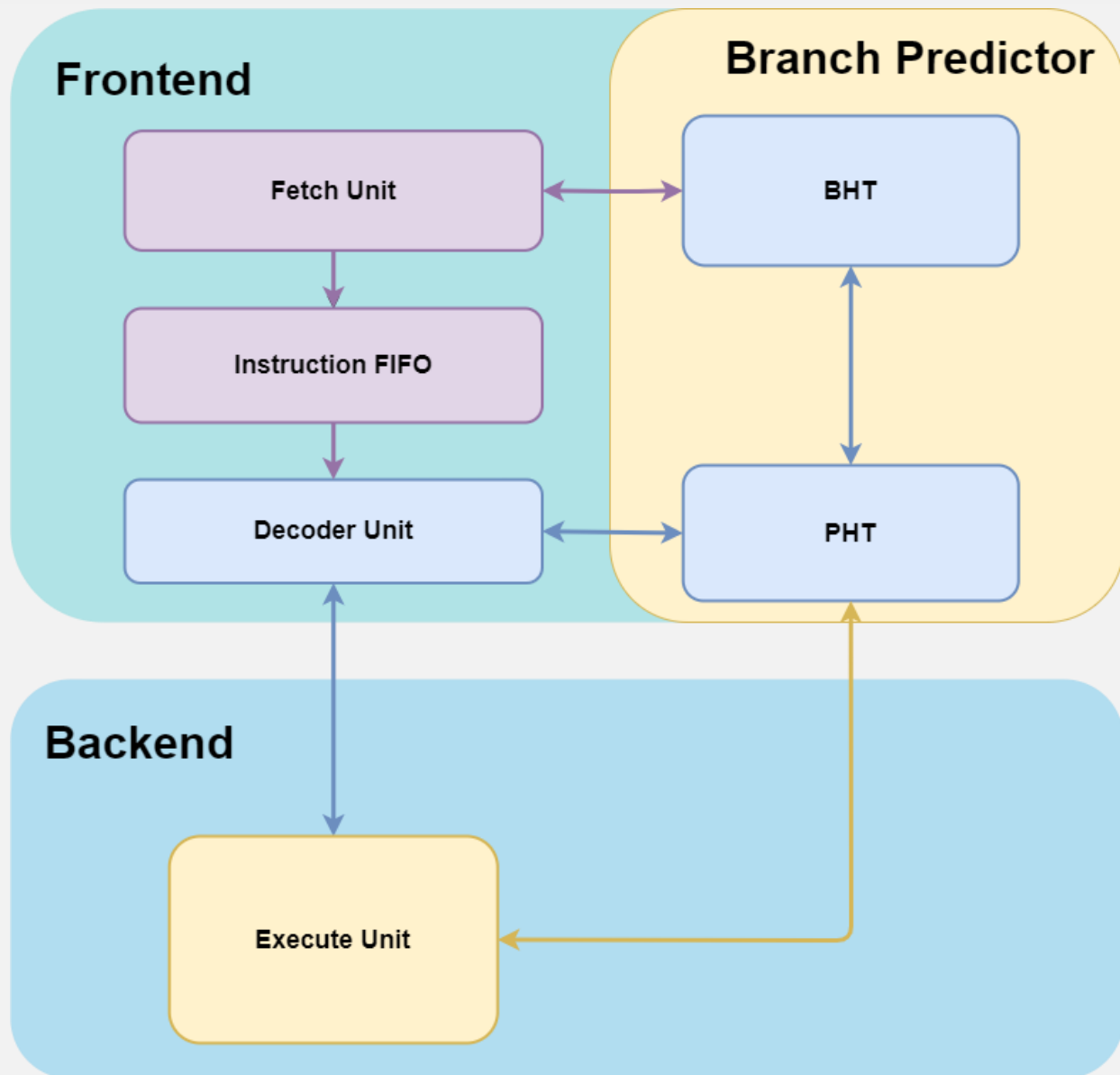


- 指令Fifo解耦取指
- 双发策略
- 例外
- RAW冲突
- 访存冲突
- 延迟槽处理



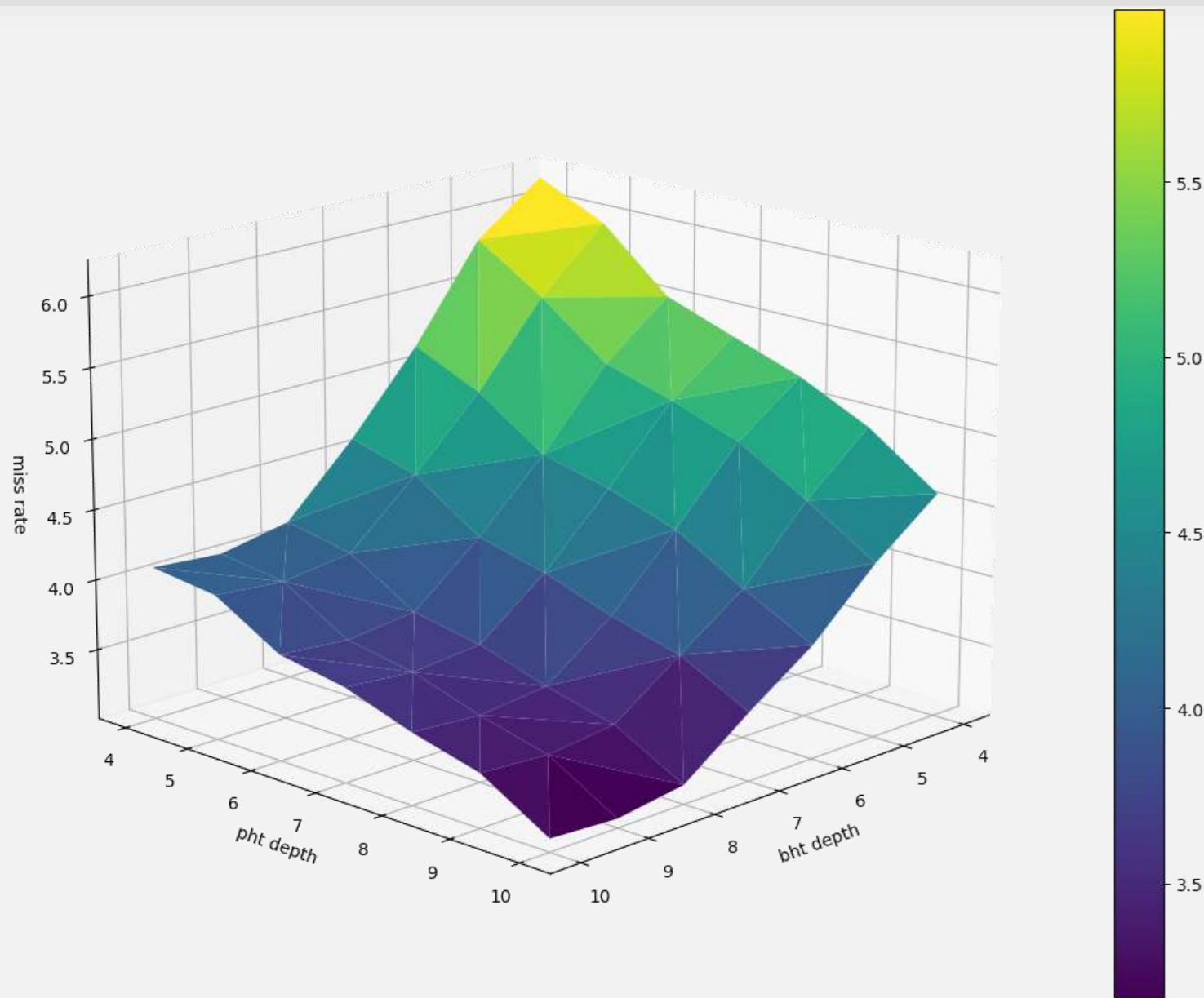
- **两级访存**
- **执行级发出访存请求**
- **访存级收到访存数据**
- **两级TLB**
- **访问缓存时访问TLB1**
- **TLB1发生缺失找TLB2**
- **TLB2处理缺失与例外**
- **频率提高13%**

- **局部历史分支预测**
- **2bit饱和更新**
- **两级预测不降频**
  - **取指阶段获得分支历史**
  - **译码阶段进行分支预测**
  - **执行阶段进行分支更新**
- **命中率 94.48%**
- **设计空间探索**



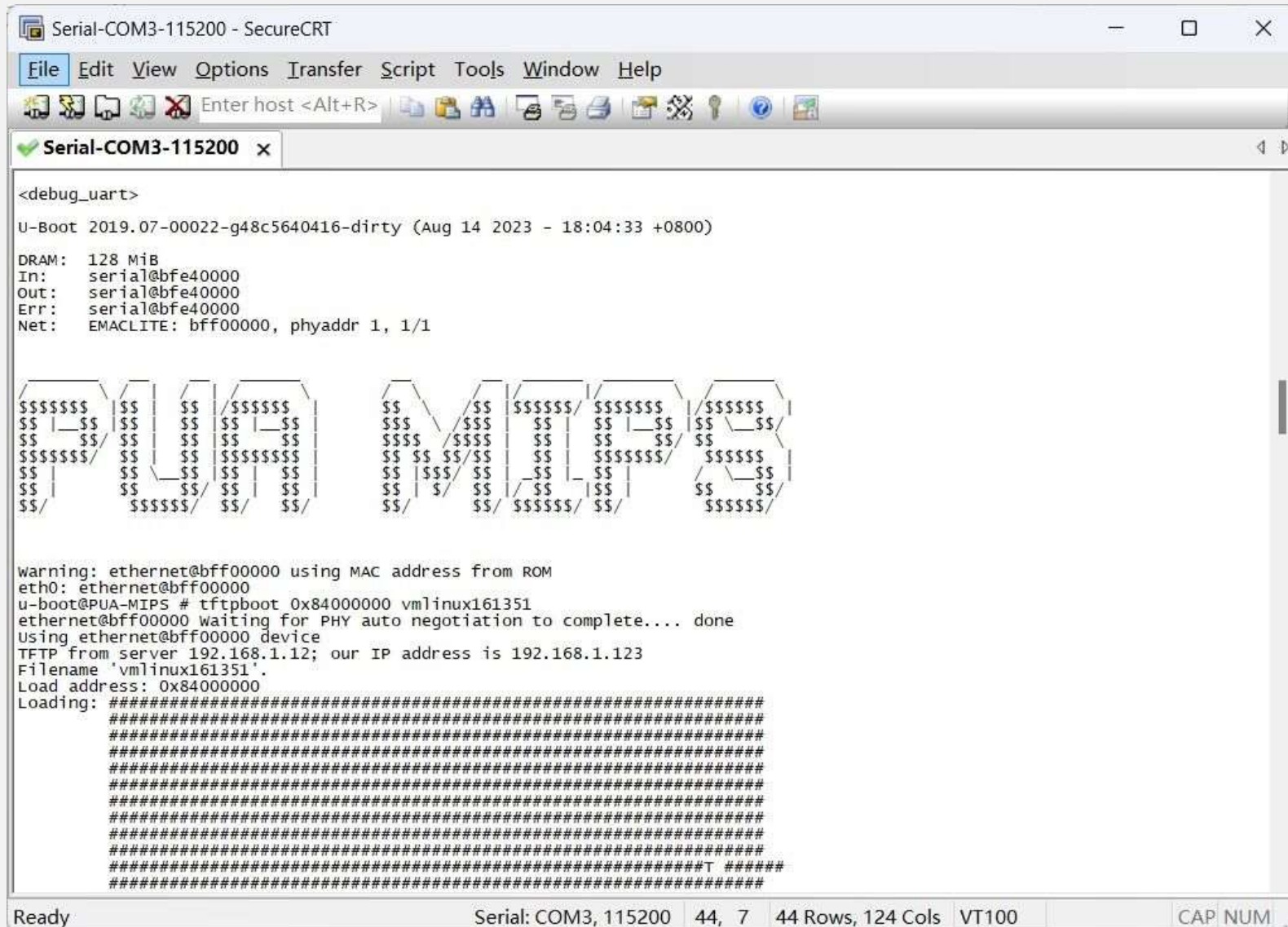


- **局部历史分支预测**
- **2bit饱和更新**
- **两级预测不降频**
  - **取指阶段获得分支历史**
  - **译码阶段进行分支预测**
  - **执行阶段进行分支更新**
- **命中率 94.48%**
- **设计空间探索**



测试集	ICache hit	DCache hit
bitcount	99.83%	96.91%
bubble sort	99.97%	99.79%
coremark	99.36%	95.59%
crc32	99.98%	97.85%
dhrystone	99.89%	81.88%
quick sort	99.96%	97.85%
select sort	99.98%	99.33%
sha	99.94%	98.89%
stream copy	99.65%	93.37%
stringsearch	99.96%	90.52%
几何平均值	99.72%	95.00%

- **指令缓存**
  - **命中率 99.72%**
- **数据缓存**
  - **命中率 95.00%**
  - **Victim Cache**
    - **缺失率 ↓14.32%**
  - **Write Fifo**
    - **阻塞率 ↓21.71%**



- 移植处理
  - 去除Branch-likely指令
  - 编写设备树



```
Serial-COM3-115200 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
Serial-COM3-115200 x
1.067736 fuse: init (API version 7.38)
1.141344 Block layer SCSI generic (bsg) driver version 0.4 loaded (major 252)
1.149744 io scheduler mq-deadline registered
1.155173 io scheduler kyber registered
4.597239 Serial: 8250/16550 driver, 4 ports, IRQ sharing disabled
4.718985 printk: console [ttyS0] disabled
4.733406 ife40000.serial: ttyS0 at MMIO 0x1fe40000 (irq = 3, base_baud = 6250000) is a 16550A
4.744015 printk: console [ttyS0] enabled
4.744015 printk: console [ttyS0] enabled
4.753765 printk: bootconsole [early0] disabled
4.753765 printk: bootconsole [early0] disabled
4.816822 printk: bootconsole [ns16550a0] disabled
4.816822 printk: bootconsole [ns16550a0] disabled
4.852614 nt35510: Initializing the nt35510
4.858253 nt35510: registered correctly with major number 251
4.865761 nt35510_of_probe
5.114123 nlcd nt35510: nt35510: device created correctly, reg=(ptrval)
5.154685 xilinx_emaclite 1ff00000.ethernet: Device Tree Probing
5.162564 xilinx_emaclite 1ff00000.ethernet: No MAC address found, using random
15.742667 xilinx_emaclite 1ff00000.ethernet: MAC address is now 1e:de:44:c0:3f:96
15.764497 xilinx_emaclite 1ff00000.ethernet: Xilinx EmacLite at 0x1FF00000 mapped to 0x(ptrval), irq=8
15.798700 NET: Registered PF_INET6 protocol family
15.830473 Segment Routing with IPv6
15.836901 In-situ OAM (IOAM) with IPv6
15.842789 sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
15.865874 NET: Registered PF_PACKET protocol family
15.874836 Key type dns_resolver registered
16.282123 Key type .fscrypt registered
16.286819 Key type fscrypt-provisioning registered
16.306511 clk: Disabling unused clocks
16.480250 Freeing unused kernel image (initmem) memory: 6540K
16.486968 This architecture does not have kernel memory protection.
16.494184 Run /sbin/init as init process

Please press Enter to activate this console.
~ #
~ # ls
bin      etc      linuxrc  sbin     test
dev      lib      proc     sys      usr
~ # cd dev
/dev # ls
console  ptmx    tty      tty3     urandom
null     random  tty2     tty4     zero
/dev #
```

## 移植处理

- 去除Branch-likely

指令

- 编译时关闭浮点支持

- 编写设备树

- 20个历史分支
- 513次提交
- 130次问题修复
- 80次代码重构
- 20次性能优化

## 性能分



stash@[0] WIP on fuck: bc08579 feat: 目前应该ok	17 Aug 2023 19:14	Clo91eaf	1ede496f
feat: 目前应该ok	17 Aug 2023 18:30	Liphen	bc08579d
perftest origin feat: 停止改动, 作为最终版, 稳定88MHz	17 Aug 2023 11:58	Liphen	ad8fbc7f
origin/test fix: 稳定88MHz	17 Aug 2023 11:57	Liphen	1377af09
stash@[1] WIP on perftest: bec10b5 fix(mul): 修复cnt问题	17 Aug 2023 18:32	Clo91eaf	8173b2f7
fix(mul): 修复cnt问题	16 Aug 2023 18:12	Liphen	bec10b5a
fix(hilo): hilo搬到mem中, 解决exbug	16 Aug 2023 16:26	Liphen	b53f6456
stash@[3] WIP on SeqDualIssueCPU: ab51152 feat<instfifo>: add instfifo prefetch	16 Aug 2023 16:08	Clo91eaf	922dc1af
SeqDualIssueCPU origin feat<instfifo>: add instfifo prefetch	16 Aug 2023 14:06	Clo91eaf	ab51152f
cutmem origin feat: mem拆两级	16 Aug 2023 13:21	Liphen	43ebf3aa
fix(jump ctrl): 修复一处数据前送错误	16 Aug 2023 11:42	Liphen	303c5fb7
feat(hilo): 搬到mem中	16 Aug 2023 11:28	Liphen	dfcff472
feat(mem2): 准备增加mem2级	16 Aug 2023 10:57	Liphen	a61b8ef9
stash@[2] WIP on cutpipe: e32e268 90M.	16 Aug 2023 19:05	Clo91eaf	08c7018b
cutpipe origin 90M.	15 Aug 2023 19:37	Clo91eaf	e32e268e
freeze bpu to 4, 6	15 Aug 2023 16:27	Clo91eaf	a1208aac
Buffer origin Merge remote-tracking branch 'origin/cutpipe' into Buffer	15 Aug 2023 14:13	Clo91eaf	84f953c2
fix<inst buffer>: fix flush bug.	15 Aug 2023 12:44	Clo91eaf	9480efc2
fix<instBuffer>: fix flush bug.	15 Aug 2023 12:26	Clo91eaf	305226a7
fix<instBuffer>: fix block bug.	15 Aug 2023 12:14	Clo91eaf	15670cfb
fix<decoder>: fix interrupt bug.	15 Aug 2023 12:06	Clo91eaf	6e4e7df3
fix(fu): 修复inst0传入的ex错误问题	14 Aug 2023 20:26	Liphen	5d9f8917
fix<fu>: fix hilo write bug.	14 Aug 2023 19:54	Clo91eaf	1758f362
fix(hilo): 修复写信号的逻辑错误	14 Aug 2023 17:09	Liphen	4f3cb866
fix<instBuffer>: fix inst buffer flush bug.	14 Aug 2023 16:11	Clo91eaf	5f8942ff
fix<instFifo>: 修复了无法判断延迟槽的bug.	14 Aug 2023 15:22	Clo91eaf	3a5eb161
fix(cache): 修改部分逻辑	14 Aug 2023 14:52	Liphen	5af62fd4
fix<ctrl>: fix lw problem.	14 Aug 2023 14:22	Clo91eaf	fcc76b35
fix<instFifo>: fix value replace bug.	14 Aug 2023 12:59	Clo91eaf	ba21747d
fix<instBuffer>: fix instBuffer flush bug.	14 Aug 2023 12:54	Clo91eaf	24083022
fix(exe stage): 修复import	14 Aug 2023 12:16	Liphen	696a3828
Merge branch 'cutpipe' into SeqDualIssueCPU	14 Aug 2023 12:05	Liphen	9c4248bd
fix<instBuffer>: fix bug.	14 Aug 2023 11:17	Clo91eaf	a79355f9
refactor<instBuffer>: rewrite the fun	14 Aug 2023 11:07	Clo91eaf	80072223





杭州电子科技大学  
HANGZHOU DIANZI UNIVERSITY

谢谢!