



第一届大学生计算机系统能力培养大赛

MIPS 处理器设计

西北工业大学计算机学院二队

卢柏岑、付豪、耿毓羚、龚宇航

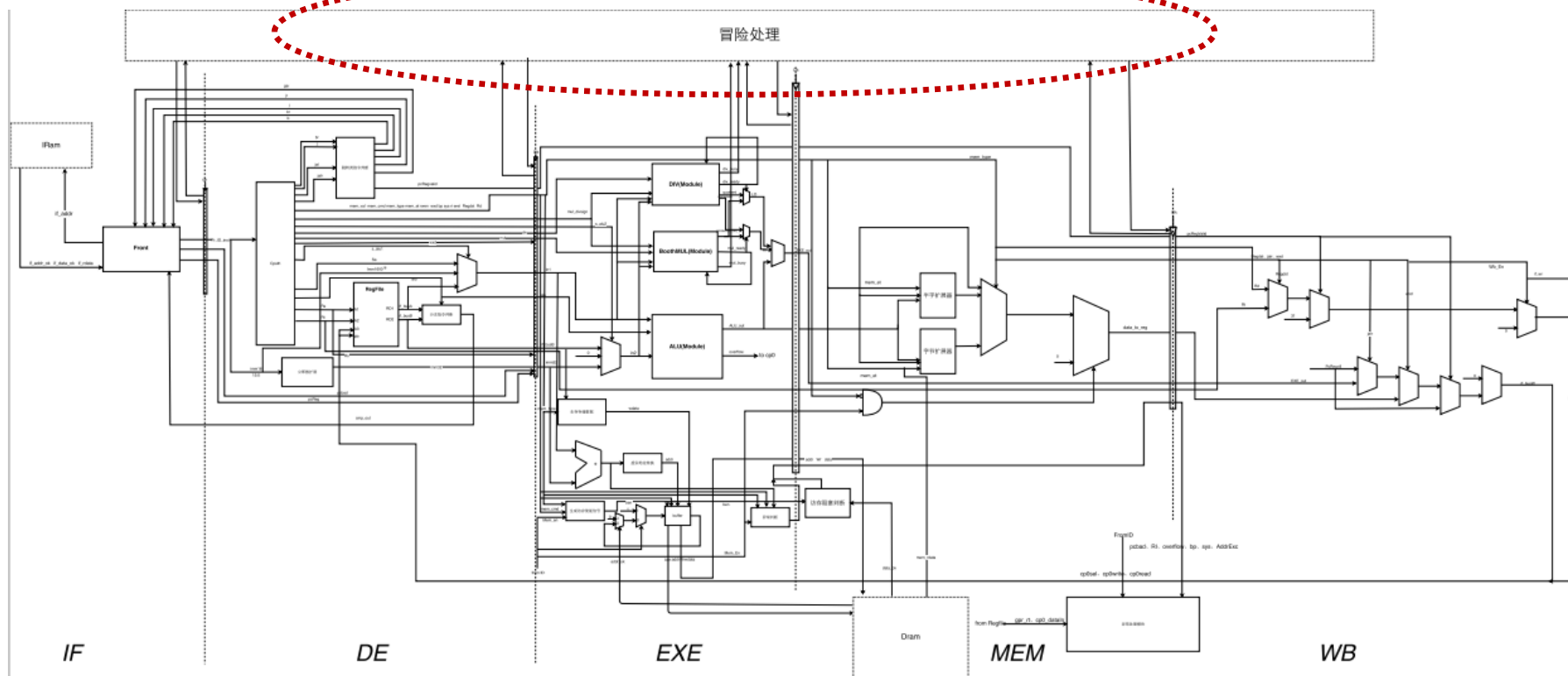
指导教师：安建峰、张萌



汇报提纲

- 处理器结构
- 性能瓶颈
- 性能优化
- 评估与分析
- SoC系统

1.处理器结构



- 单发射五级流水线结构
- 使用旁路、阻塞和延迟槽等技术处理三种冒险
- 支持异常检测与精确异常
- 使用Chisel描述整个处理器

2.性能瓶颈

➤ 流水线断流

- 分支判断在EX级导致后续指令PC产生滞后
- 2拍的存储器访问

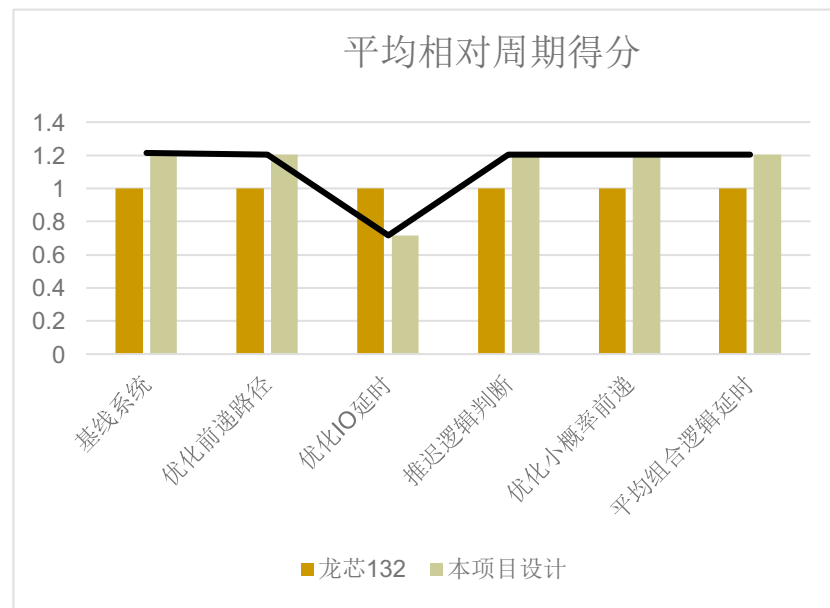
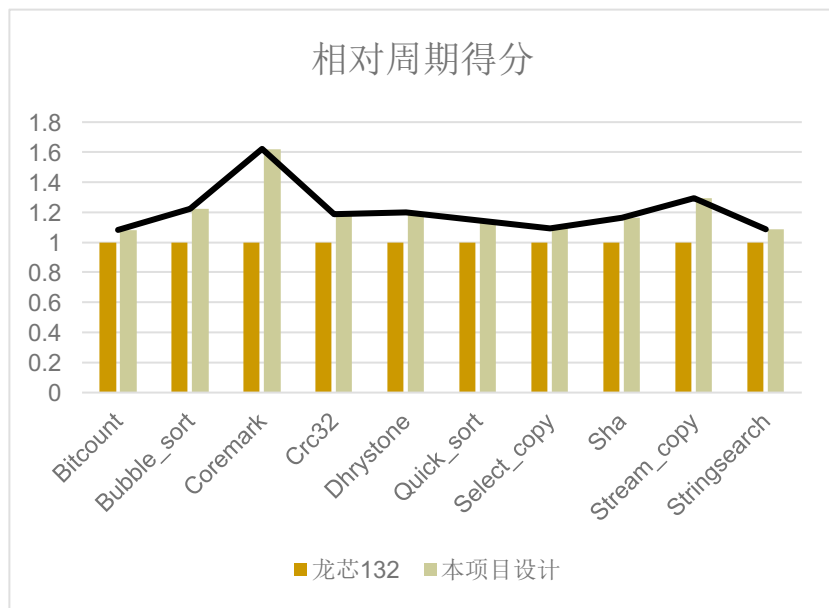
➤ 关键路径延迟

- 逐级前递路径策略
 - 小概率事件的操作数前递
 - 流水级延迟不平衡
- ### ➤ CPU内部组合逻辑与IO直接连接使得IO访问成为关键路径

3. 性能优化策略

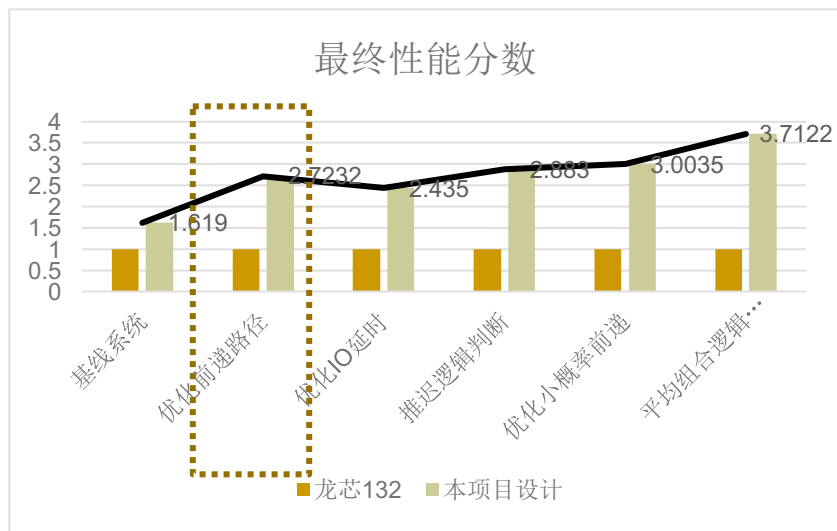
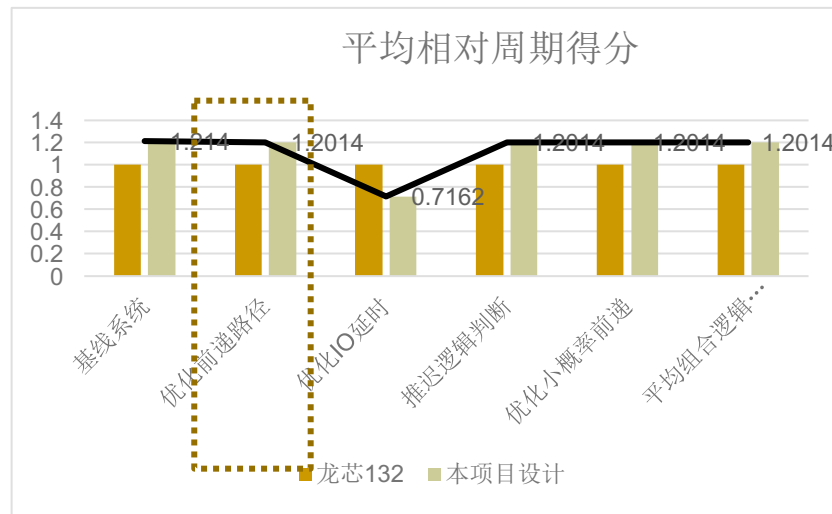
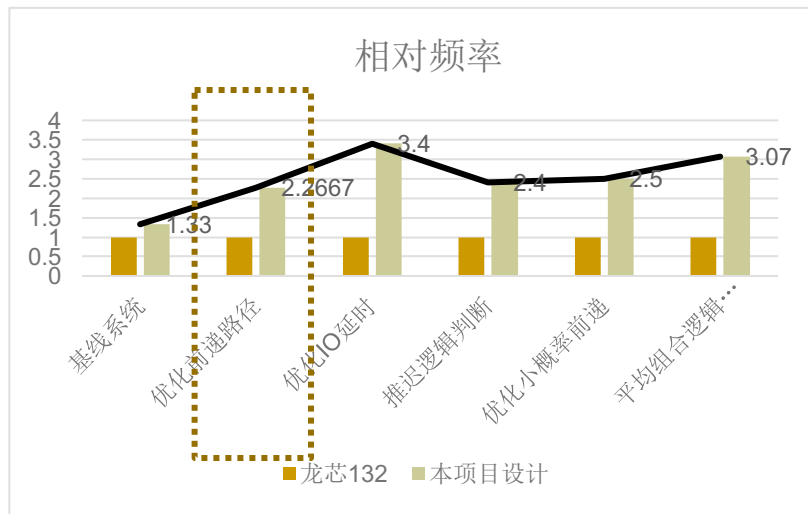
- 将分支跳转判断提前到译码 (ID) 级
- 将发出访存请求提前到执行 (EX) 级
- 优化前递路径
- 优化逻辑判断
- 优化小概率事件
- 均衡组合逻辑延时
- 优化IO延时

4.1 提前发出访存请求、提前分支判断逻辑



体系结构优化

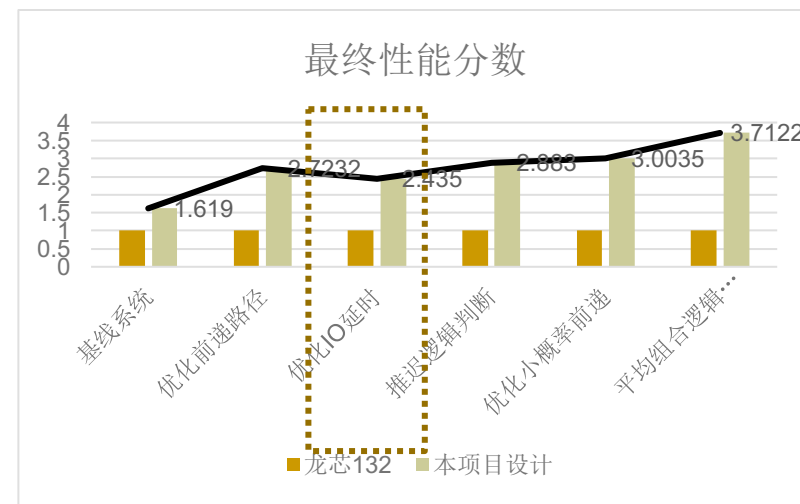
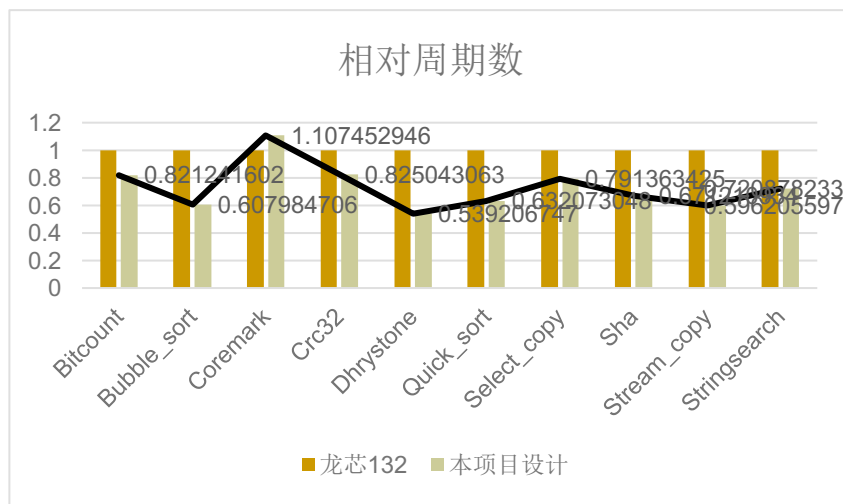
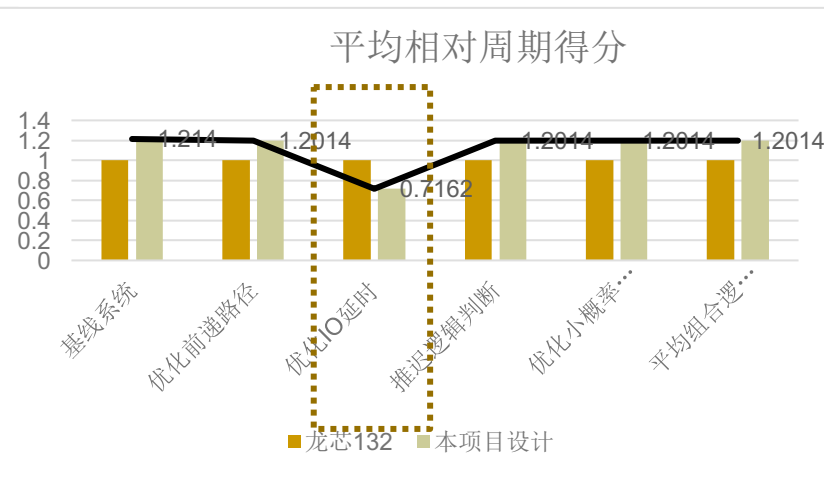
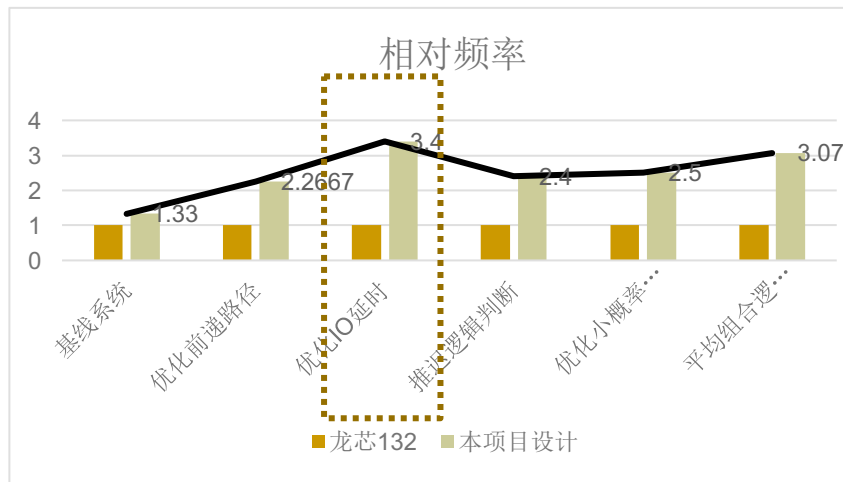
4.2 优化前递路径



频率提升了 70.42%
同时IPC未明显降低

预赛提交

4.3 优化IO延时

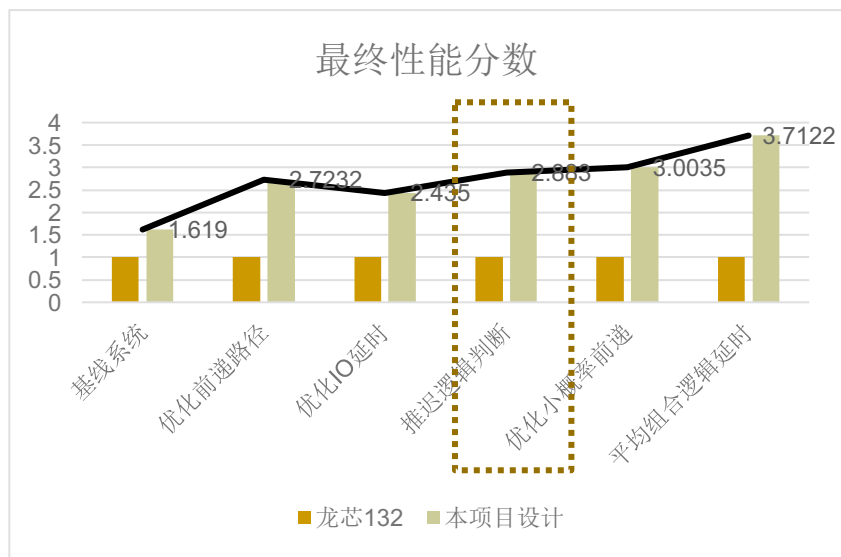
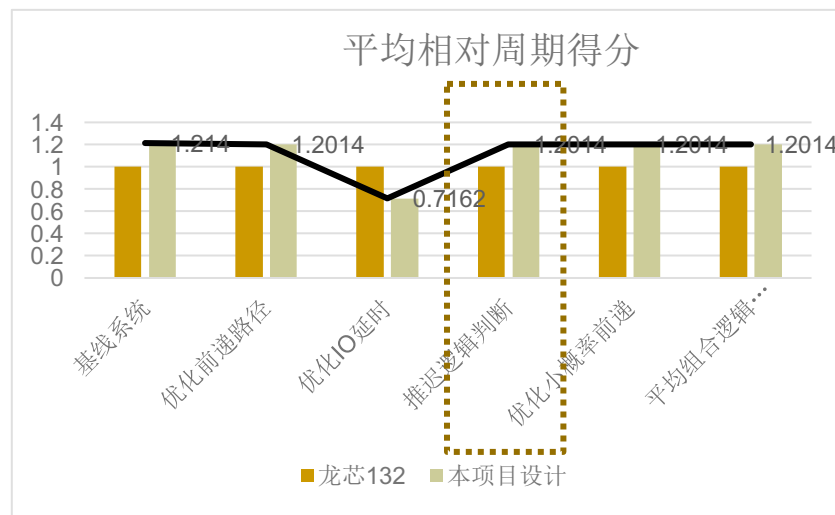
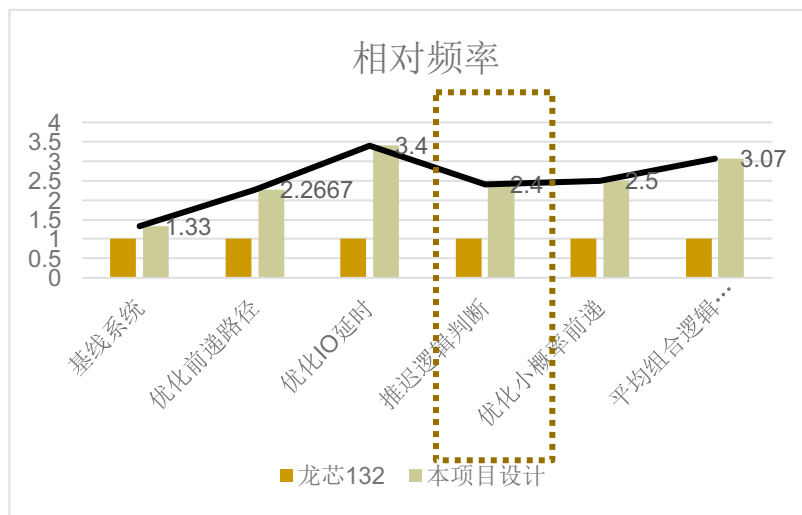


频率提升了155.64%，但IPC下降了41.00%。

与前递路径优化相比总体性能略有下降



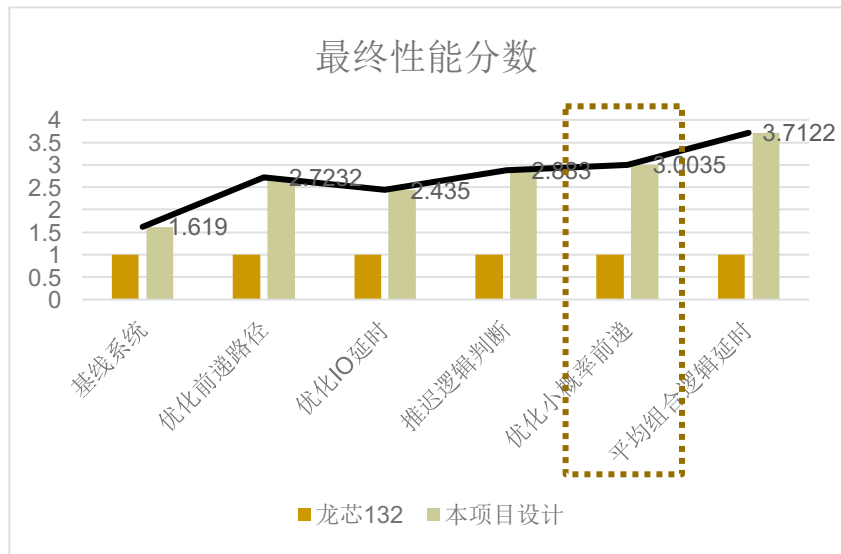
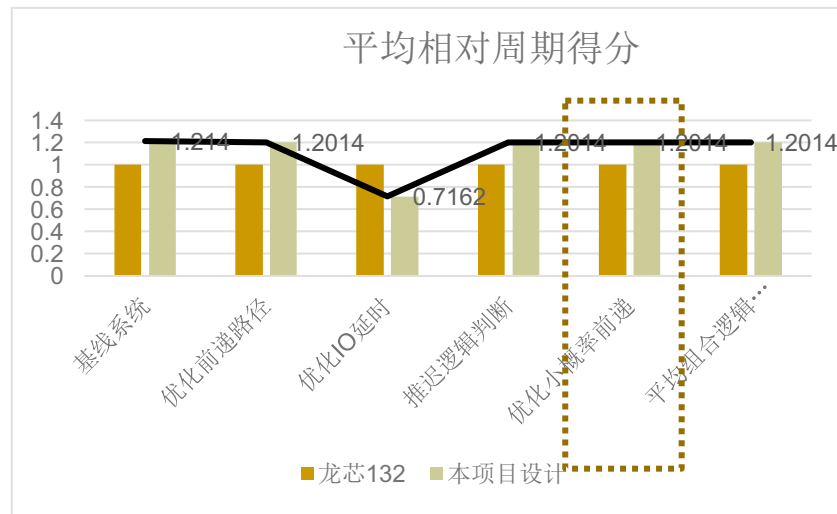
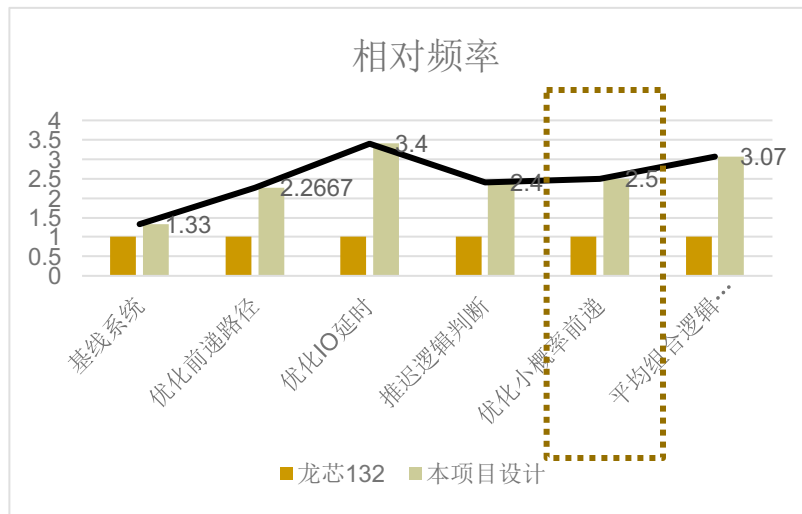
4.4 优化逻辑判断



频率提升了 80.45%
同时IPC未明显降低

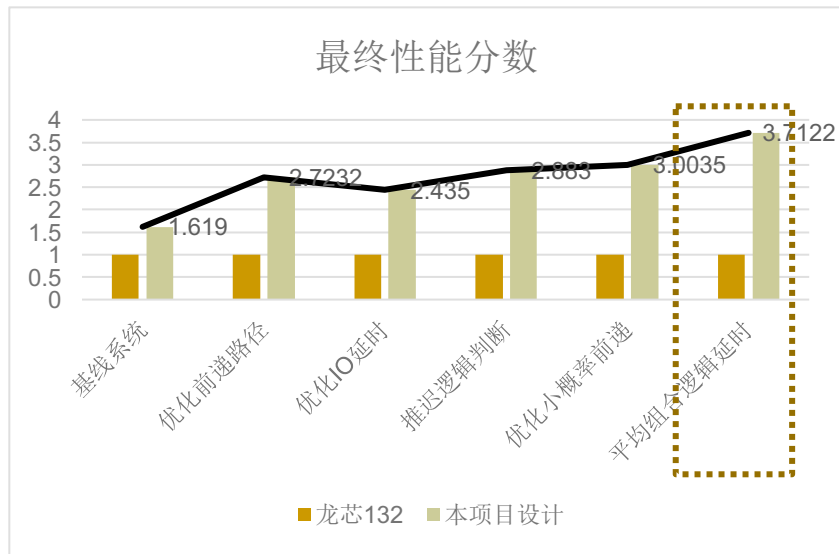
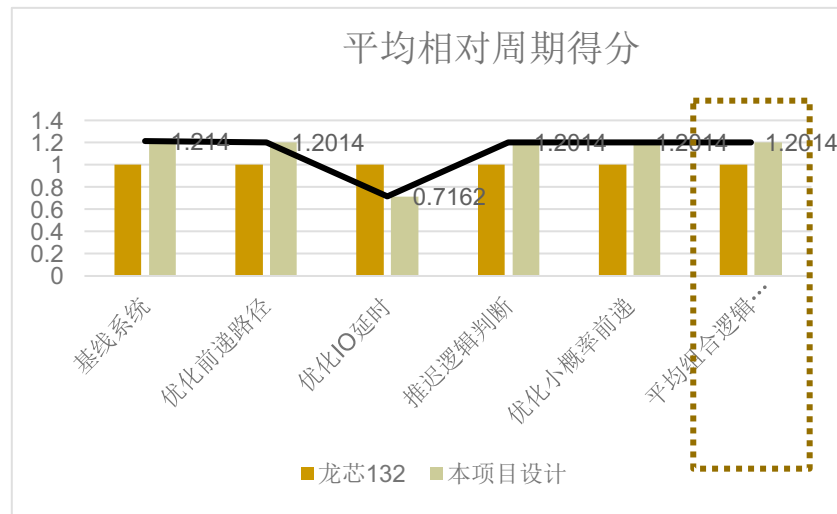
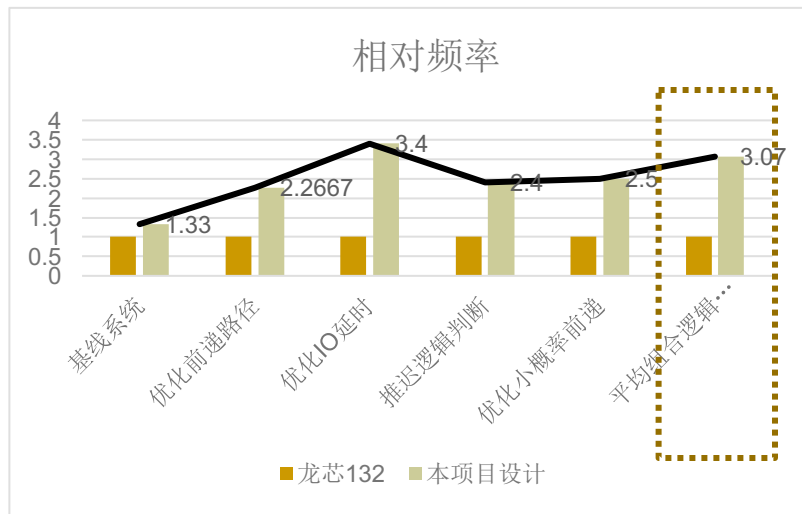


4.5 优化小概率事件



频率提升了 87.97%
同时IPC未明显降低

4.6 平均组合逻辑延时



频率提升了 130.83%
同时IPC未明显降低

4.7 最终性能

	预赛	决赛	优化比率
综合频率	68.842MHz	92.696MHz	34.65%
相对频率	2.2947	3.0899	34.65%
相对周期得分	1.2014	1.2014	0%
性能得分	2.7569	3.7122	34.65%



5. SoC 系统

➤ SoC硬件环境

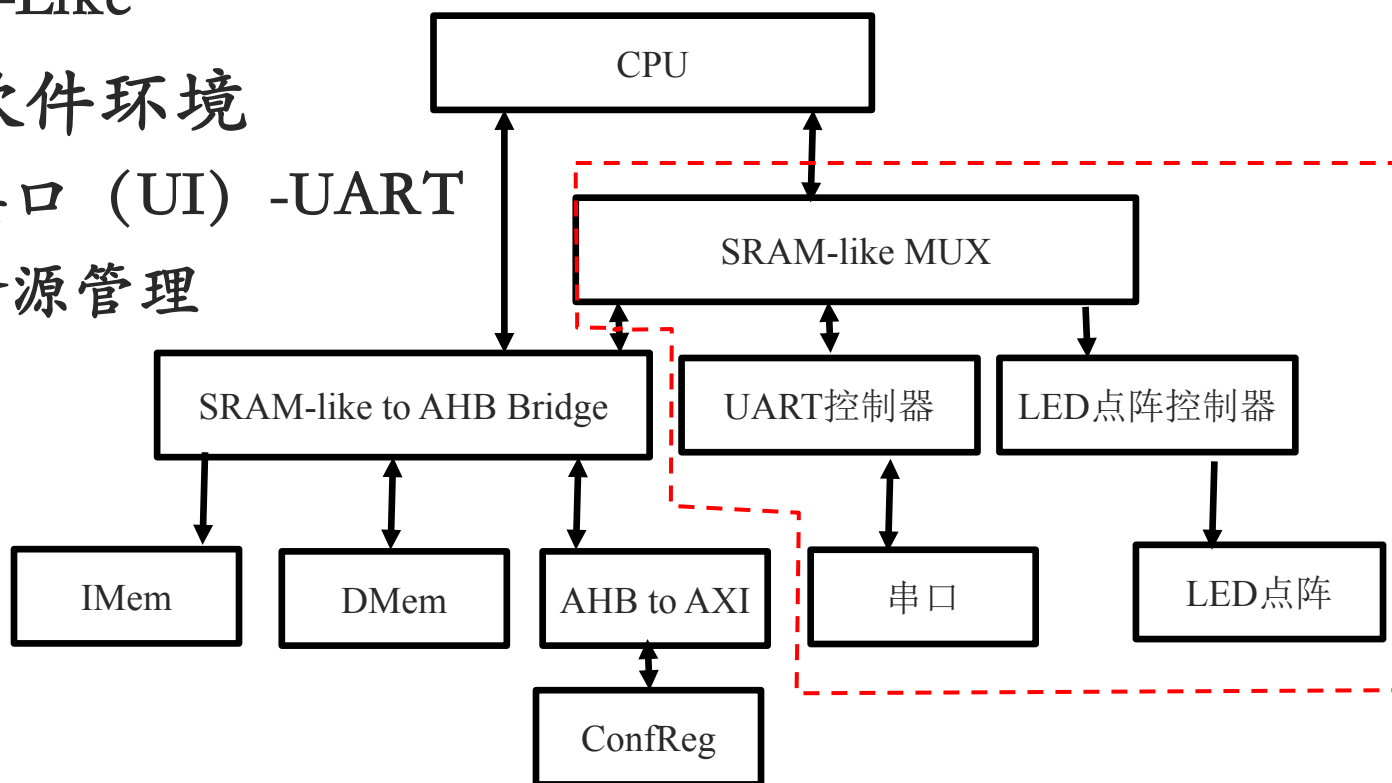
➤ 外设资源

➤ SRAM-Like

➤ 运行时软件环境

➤ 用户接口 (UI) -UART

➤ 外设资源管理



请各位专家批评指正
谢谢



Chisel与RISC-V

```
import chisel3._
class GCD extends Module {
  val io = IO(new Bundle {
    val a = Input(UInt(32.W))
    val b = Input(UInt(32.W))
    val e = Input(Bool())
    val z = Output(UInt(32.W))
    val v = Output(Bool())
  })
  val x = Reg(UInt(32.W))
  val y = Reg(UInt(32.W))
  when (x > y) { x := x -% y }
  .otherwise { y := y -% x }
  when (io.e) { x := io.a; y := io.b }
  io.z := x
  io.v := y === 0.U
}
```

- 具有结构化的描述优势
- 通过选择不同后端产生C仿真和Verilog仿真
 - Scalar+C
 - Verilator
- Berkeley与SiFive使用Chisel描述RISC-V结构

