龙芯杯团体赛参赛指北

ysyx五组王植鑫

本指北仅为个人见解，具体以官方赛制规则为准

龙芯杯团体赛简介

需要做什么

与ysyx的区别

需要做什么

- 实现一款基于la32r架构的高性能核
- 在自己la32r核上启动Linux
- 移植丰富的外设与软件用于整活

与一生一芯的区别

一生一芯	龙芯杯
模拟器、Soc、Core	侧重Core
自己独立完成	1-4人团队协作
侧重系统能力培养	侧重比拼高性能核和丰富的展示
丰富的引导方向	弱引导，需要逐步探索
锻炼学习能力	锻炼团队协作
周期长	周期短
从零开始逐步搭建	较为开箱即用

赛制规则

以第八届龙芯杯为例

- 40%的核性能分
- 20%启动系统分
- 10%自定义指令答题
- 30%答辩分
- 你要面对的现实

40%性能分

- 决赛中会拆分性能分为20% **IPC分数**，20% **频率分数**
- IPC分数和频率分数按照队伍排名映射分数为0~20分
- 即使你的性能分遥遥领先别人太多，映射完后与其他队伍的差距会进一步缩小
- 剩下60%是关键

20%启动系统分

- 启动Linux并成功运行ls指令即可拿满该项
- 事实是在这一届比赛中启动的队伍听说并不是很多
- 头部队均能启动并卷花活在本项拉不开差距
- 二等和三等奖之间仅相差了启动系统

更事实的是，对于非头部队，启动系统收益甚至大于做多发核

10%自定义指令答题

- 请务必在答题前模拟一遍增加指令编译的过程并确保环境没有问题
- 完成题目即可拿满本项分
- 在这上面丢分真的很蠢

30%答辩分

- 包括但不限于，利用开发板资源进行的外设展示，Linux-la32r的软件移植展示
- 对于争取特等一等奖的队伍来说十分甚至有九分重要
- 没活就只能咬打火机 😭

你要面对的现实

- 龙芯杯的初赛要当作决赛对待（10天起系统也太赶了吧啊喂）
- 想拿三等So Easy但是想更往上进一步就要狠狠地卷
- 团队协作效果差
- 可能不像一生一芯那么有趣
- 后期会被漫长的综合布线过程充斥，有趣感减弱，等待时序报告结果十分煎熬（综合布线提时序阶段平均每次综合达30多分钟，很磨人耐心）

致胜法宝

以第八届龙芯杯为例

- ChipLab敏捷开发平台与Openla500
- 往届开源核
- ChipLab可以让你获得好用的difftest工具
- Openla500可以让你快速入门la32r指令集
- 往届开源核可以让你对性能部分知根知底

difftest&diffcore

- difftest可以快速的帮你定位错误
- diffcore?
  - difftest信息快速但是不全面，无法细微的观察核发生了什么
  - diffcore虽然慢，但是信息全面，可以帮助你快速理解学习Chiplab soc的正确行为以及更深入理解la32r指令集

造轮子|用轮子

- 对于比赛来说，Chiplab足够好用，可以满足大部分开发需求
- 分析Chiplab平台使用痛点
- 针对痛点造轮子，同时也要考虑备赛时间够不够用

性能知根知底

- 比赛会诞生很多优秀设计
  - 站在巨人的肩上
  - 闭门造车不可取
- 一届更比一届强
  - 记录性能参数
  - 分析微架构设计，找到高频秘诀
  - 看懂代码、加入性能计数器（Btb、Cache、多发率），分析优点与不足
  - 对比自己核与开源核差距，针对弱点逐个击破

备赛规划

以第八届龙芯杯为例

- 时间规划
- 队内分配
- 团队协作
- 经典环节

时间规划

适用于有一定基础的（一生一芯多周期阶段）

- 不建议过早准备，前期激情满满后期会变得疲软
- 如果自身安排较为紧张只想拿个三等，7-8月单发射核足以，用用心启动系统也够了
- 如果想要再往上卷一卷(多发顺序、乱序)
  - 启动系统是必须项！！！
  - 多发：5-8月，基础薄弱的适当延长
  - 乱序：孩子没做过，等待大佬们的分享

队伍分配

- 流水线整体架构设计
- Cache，Axi，Tlb，Btb
- 外设、软展示
- 对于头部队伍竞争，外设和软展示(30%的答辩分)即为重要，70%的分对于头部队伍来说并不能拉开什么差距

团队协作

- 流水线架构设计与Cache等单元设计进度不一
  - 尝试先用开源项目代替并验证功能
  - 等队友开发完后换回去
- 代码尽量划分出不同的模块（否则你会领会到天天都在解决冲突）
- 外设、软展示与core没有关系可以基于openla500开发再迁移到自己核上
- 建议Cache，Axi，Tlb，Btb开发进度领先核架构（惨痛教训，开发进度不一致导致花大量时间反复提频）

经典环节

😭 队友呢，队友救一下啊（找不到队友单挑环节）

- 现在还早，考虑培养一个？
- 如果实在找不到，可以在一生一芯里提前学习Cache、分支预测等易用易迁移模块，可以让你在写la32r的核中少很多重复步骤

实际参加比赛发现的问题

仅为自己参赛经验感受

开发周期过长

- 决定开发的时候是3月份，一生一芯的进度是刚做完总线，并提前完成了流水线
- 3-5月，单发射六级流水处理器，主要是为了走一下起系统流程和la架构特点（一部分模块复用了开源的，例如乘除法器，Cache，加快了写核进度减少了在这些地方浪费的时间）(本来也想搞一个diff工具，在这顶上花费了很多时间，但是最后还是鸽了）
- 6-8月，双发射七级流水线，6-7月完成微架构设计，7-8月提频，起系统，提ipc。
- 不建议准备久的原因？建议参加龙芯杯的可以先略soc，把Cache和后面性能调优的做了。我这样的流程走下来其实单发射这个步骤有点多余。

初赛即决赛

惨痛的教训

- 初赛作品交完后决赛那段时间相当紧张
- 决赛展示的内容初赛提交后才开始准备，队伍里没有懂这方面，决赛草草交了个简单的外设演示
- cacop和tlb等指令到决赛再准备不但会严重影响提频，还会影响起系统等。
- 决赛为了提ipc改了激进的方案，最后发现激进的方案又启动不了系统。一路回退，导致决赛性能没怎么提。

开发规划混乱

- 没有指定开发规范，后期因为个人习惯不同，不同码风在项目上乱飞
- 各分工开发进度不一，没有整体完成后再开始提频，导致后期把大量时间花在了反复提频上，也浪费了大量时间
- 未深入研究手册用错误的方法使用IP核，造成仿真过上板寄的现象，浪费了大量时间
- 没有明确的时间规划

总结

- 针对赛制规则找优势
- 掌握致胜法宝
- 团队协作决定成败
- 最后，祝愿大家明年都能获得好成绩(有团队赛的问题欢迎随时来单杀我)