

# 项目报告

## 个人介绍

学号	20210544
姓名	石正璞
学校	南京航空航天大学
年级	博士一年级
专业	软件工程（本科时的专业是工业工程）
之前是否设计过处理器	无
电子邮箱	zhengpushi@nuaa.edu.cn

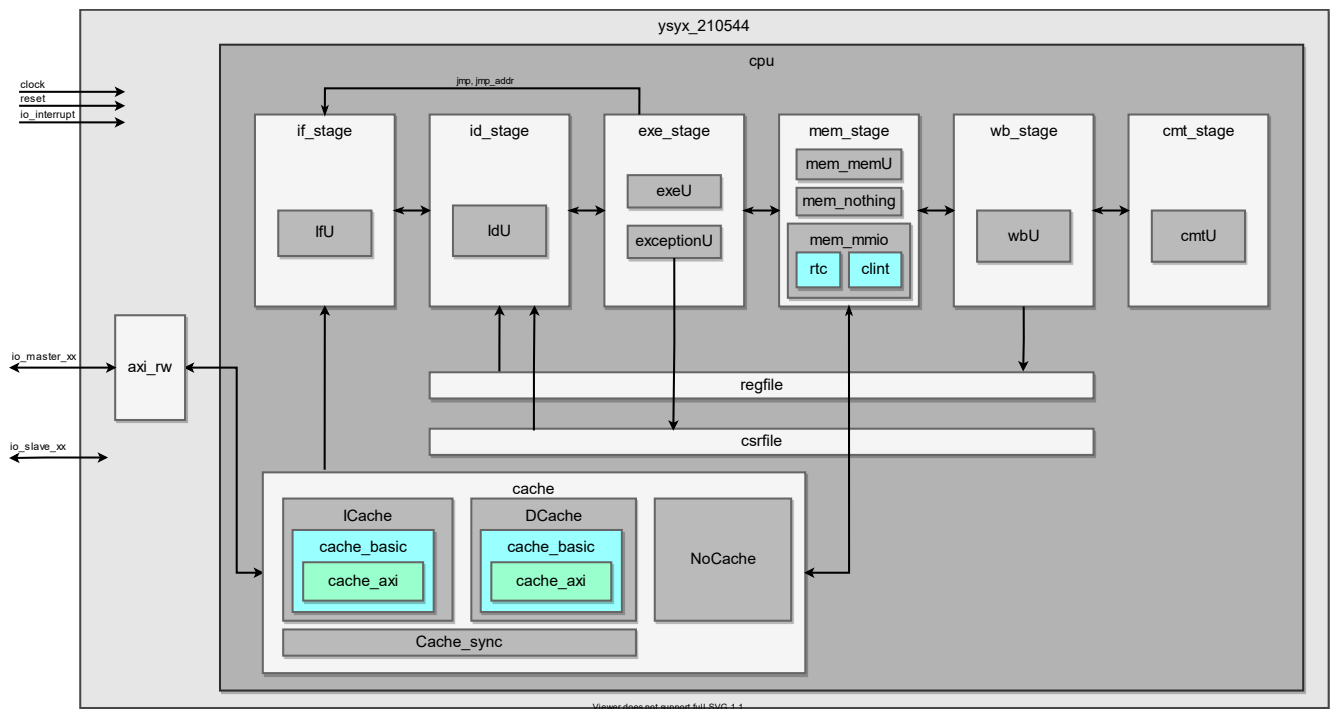
## 项目情况介绍

仓库地址: [https://github.com/zhengpushi/p1\\_2021ysyx/tree/master/code/oscpu](https://github.com/zhengpushi/p1_2021ysyx/tree/master/code/oscpu)

开发语言: Verilog

项目介绍: 在 oscpu-framework 的基础之上, 实现了既定任务。附加的实现了 Cache, 使得运行效率提高很多。遗憾的是: 五级流水线并未能开启, 因为冲突处理机制暂未实现。

## 处理器架构图和模块说明



module	含义
ysyx_210544	顶层模块
cpu	CPU 内核
axi_rw	axi 接口读写，最大 512bit 的读写支持，burst 传输或单次传输。
if_stage	取指接口
IfU	取指功能模块
id_stage	译码接口
IdU	译码功能模块
exe_stage	执行接口
exeU	执行功能模块
exceptionU	异常处理功能模块
mem_stage	访存接口
mem_memU	访存功能模块
mem_nothing	访存功能模块，仅握手
mem_mmio	内存映射功能模块
rtc	Real-Time-Clock 模块
clint	支持包括 clock 中断在内的中断逻辑
wb_stage	写回接口
wbU	写回功能模块
cmt_stage	提交接口
cmtU	提交功能模块
regfile	寄存器文件
csrfile	csr 寄存器文件
cache	cache 主模块。将 ICache/DCache 或 NoCache 需求分别选通交与 axi_rw。
ICache	是 cache_core 的例化，表示指令缓存
DCache	是 cache_core 的例化，表示数据缓存
cache_core	将不对齐的访问处理为 1~2 次 cache_basic 调用
cache_basic	一个基本的 cache 单元，4 路 16 组，每组 512bit，分 4 个 RAM 实现。每个 data_array 占用 4KB；每个 tag_array 占用 208Byte(=1664bit)。
cache_axi	为 cache 而适配的 512bit 数据宽度的 axi 接口
Cache_sync	实现 fencei 逻辑，将 DCache 中的数据同步到 ICache 中
NoCache	不使用 Cache 而直接访问 axi_rw 接口

## 设计特色展示

1. cache\_core 模块，可以将非对齐地址的访问处理为 1 到 2 次对齐地址的访问。
2. Cache 同时支持 Flash 和 Memory 的指令缓存。  
缓存来自 Flash 设备的指令时，AXI 接口是 4 字节（32 位）访问，  
缓存来自 Memory 的指令时，AXI 接口是 64 字节（512 位）访问，  
通过 cache\_axi 接口来自适应。
3. Cache\_sync 模块，收到 fencei 指令后，仅将 DCache 中受影响的数据同步到 ICache，效率较高。

# 心得感想

## 1. 一些感想

- a) `oscpu-framework` 示例代码很好，可帮助理解 CPU 工作流程。
- b) 个人进度刚开始很慢，后来加快了。原因是花了很多精力补充基础。
- c) 尽量不要熬夜，保持精神状态最佳时再工作。
- d) 高级的 `cpu` 特性，需要更多的理论知识。
- e) 对于新的软件模块，先看源码，再使用，避免反复调试浪费时间。
- f) 南京大学计算机系统基础课程实验 `ics2020`，对于理解 AM，NEMU 比较有帮助。
- g) NJU 的几个开源项目，有机会多参与。

## 2. 调试最难的 bug

- a) 后端返回报告显示 `hello` 测试 `fail` 时，花了较多时间调试，同后端助教沟通并反复查看波形，最终解决了问题。期间，`github` 上的 `vcs_report` 已经显示 `pass`，但是助教老师的测试环境下显示没有通过。原因是后端的两个测试环境对 `wready` 信号的设计稍有不同，而我的处理器核刚好只在其中一个环境下触发 `bug`。根本原因是我的处理器核中 `wvalid` 信号在第一次传输数据时多保持了 1 个周期。最终修复了该设计缺陷。

## 3. 参加“一生一芯”得到的收获

- a) 熟悉了 `verilator`、`gtkwave`、AM、NEMU、`difftest` 等项目
- b) 对处理器核设计有了一定的认识
- c) 认识了一些同行朋友。

## 4. 对“一生一芯”的期望或改进

- a) 希望能够继续办下去，让更多的爱好者有机会参与。