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 |

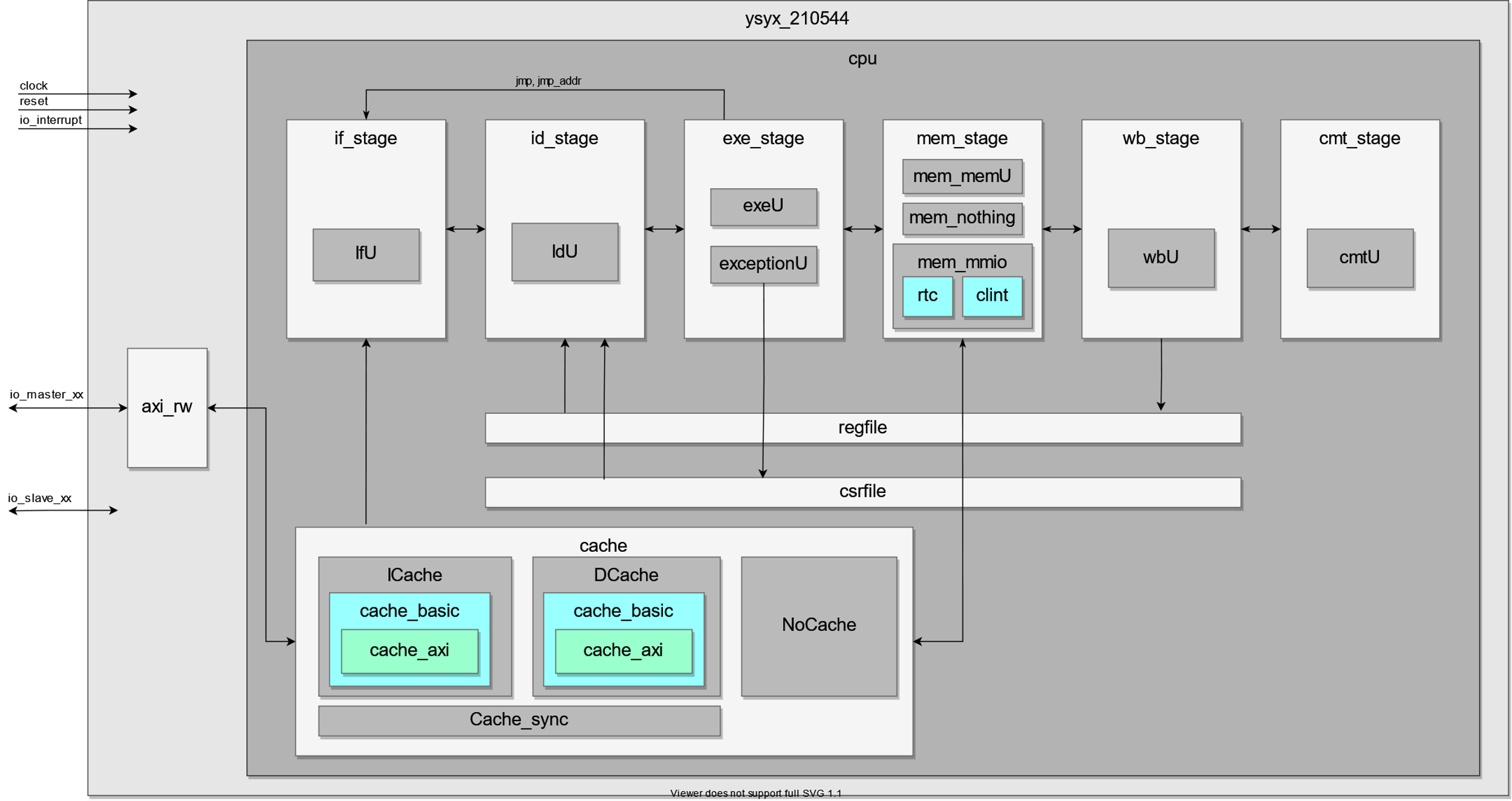
# 项目情况介绍

仓库地址：<https://github.com/zhengpushi/p1_2021ysyx/tree/master/code/oscpu>

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。

1. 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
   1. oscpu-framework示例代码很好，可帮助理解CPU工作流程。
   2. 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。
   3. 尽量不要熬夜，保持精神状态最佳时再工作。
   4. 高级的cpu特性，需要更多的理论知识。
   5. 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。
   6. 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。
   7. NJU的几个开源项目，有机会多参与。
2. 调试最难的bug
   1. 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
   1. 熟悉了verilator、gtkwave、AM、NEMU、difftest等项目
   2. 对处理器核设计有了一定的认识
   3. 认识了一些同行朋友。
4. 对“一生一芯”的期望或改进
   1. 希望能够继续办下去，让更多的爱好者有机会参与。