

龙芯杯个人赛参赛作品设计报告

学校：西南科技大学

姓名：雷宇航

一、设计简介

参赛作品基于龙芯精简指令集^[1]实现了五级流水线+icache（实际六级流水），支持串口通信，可以在 165Mhz 的频率下通过一到三级测试，以及性能测试。

另外基于 nemu 构建了一个 get trace 工具用于 debug。

二、设计方案

（一）总体设计思路

参赛作品由 system verilog 语言构建，并使用了 interface 来连接不同的模块。

处理器使用了流水线技术，共分为 6 个流水级，分别是：IF（实际用于生成 next pc），icache，ID，EXE，MEM，WB。流水线的设计参考了《CPU 设计实战》上的内容。

由于 FPGA 上 ram 的时序限制，icache、MEM 访存时会暂停四个周期。icache，MEM 访存或使用串口时会先经过仲裁器模块，来解决访存的冲突以及选择访问的对象。

IF 流水级实际上并不做取指的操作，只进行对下一条指令地址的选择；实际的取值操作在 icache 流水级进行，icache 采用直接映射的地址映射方式，在该流水级会进行预译码；ID 流水级将从 icache 取出的指令进行后续的译码操作，得到立即数，寄存器文件读写信号及地址等信息；EXE 流水级进行指令的执行以及 ram 访存地址的计算；MEM 流水级进行指令的访存操作；WB 流水级进行对 ram 读入数据的处理，以及进行寄存器写回数据的仲裁。

（二）base ram、ext ram 以及串口仲裁模块设计

该模块的设计参考了 2020 年龙芯杯倪仁涛的参赛作品和高子博的参赛作品。该模块的功能是将内核输出的 ram 访存信号进行处理，判断此时是否应该访问 base ram、ext ram 以及串口，保证访存写入或读出的数据正确。

该模块的输入信号包含访存地址、读写控制信号、数据掩码信号以及写入数据，输出从 ram 或者串口读出的数据。

访存相关控制信号以及写数据到 ram 之间有寄存器以满足时序要求，串口控制采用的是

大赛提供的 UART 模块，并加入了 fifo 缓冲模块对输入输出进行优化。

（三）流水线控制模块设计

流水线的控制参考了《CPU 设计实战》，每一级流水线都包含 valid、validin、allowin、ready_go 信号。通过控制 ready_go 来控制流水线级是否暂停；valid 信号用于判断当前流水级保存的数据是否有效，避免通过插入 nop 指令来清空流水级；validin 信号用于指示上一级流水是否准备好，其保存的数据是否有效；allowin 信号用于指示当前流水级是否接收允许上一级的数据。

三、设计结果

（一）设计交付物说明

```
.
├── asm
├── Makefile # 用于调用调试等工具
├── README.md
├── simpleide # 包含调试，构建内核的工具
|   ├── inst-interpretor # 构建内核工具，用于生成译码相关 sv 代码
|   |   ├── inst.xlsx
|   |   ├── Makefile
|   |   ├── requirements.txt
|   |   └── src # python 源码，读取表格文件内容，生成所需 sv 代码
|   ├── rom # 编译测试汇编代码
|   |   ├── asm
|   |   ├── include
|   |   └── Makefile
|   ├── scripts # 工具运行脚本
|   |   ├── native.mk
|   |   └── tools.mk
|   └── simplecpu # trace 生成工具
```

```

|       |—— include
|       |—— Makefile
|       |—— src # c 语言源码
|—— tb_behav.wcfg # 波形文件
|—— thinpad_top.srsc
|   |—— constrs_1
|   |—— sim_1
|   |—— sources_1
|       |—— ip # 包含
|       |—— loongarch_cpu # cpu 内核
|       |—— new
|—— thinpad_top.xpr

```

(二) 设计演示结果

1 一级测试结果:



图 1 一级测试结果

2 二级测试结果:



图 2 二级测试结果

3 三级测试结果:

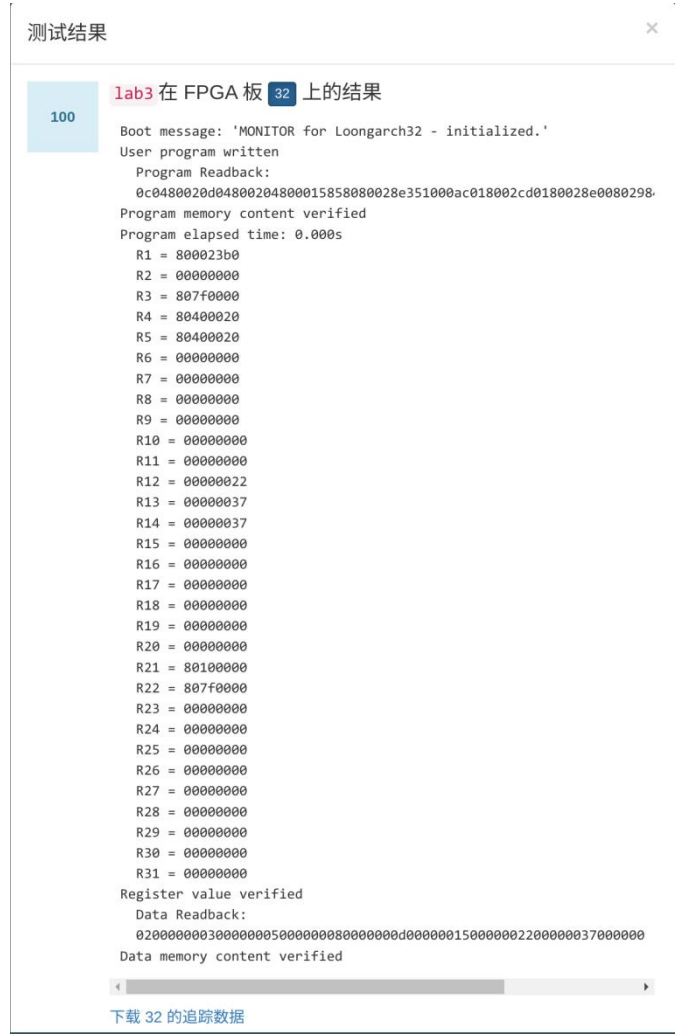


图 3 三级测试结果

4 性能测试结果:



图 4 性能测试 STREAM 测试结果



图 5 性能测试 MATRIX 测试结果



图 6 性能测试 CRYPTONIGHT 测试结果

四、参考设计说明

2020 年“龙芯杯”倪仁涛的参赛作品中的 ram 仲裁模块
(https://github.com/fluctlight001/cpu_for_nscsc2020)

2020 年“龙芯杯”高子博的参赛作品中的 ram 仲裁模块(<https://github.com/cassuto/yamp-32>)
《CPU 设计实战》中的流水线控制相关代码

五、参考文献

- [1] 龙芯中科技术股份有限公司. 龙芯架构 32 位精简版参考手册[EB/OL]. [2024-8-1].
[https://www.loongson.cn/uploads/images/2023041918122813624.龙芯架构 32 位精简版参考手册_r1p03.pdf](https://www.loongson.cn/uploads/images/2023041918122813624.龙芯架构32位精简版参考手册_r1p03.pdf).