

CrocoMIPS 设计报告

学校：哈尔滨工业大学（深圳）

姓名：许辰涛

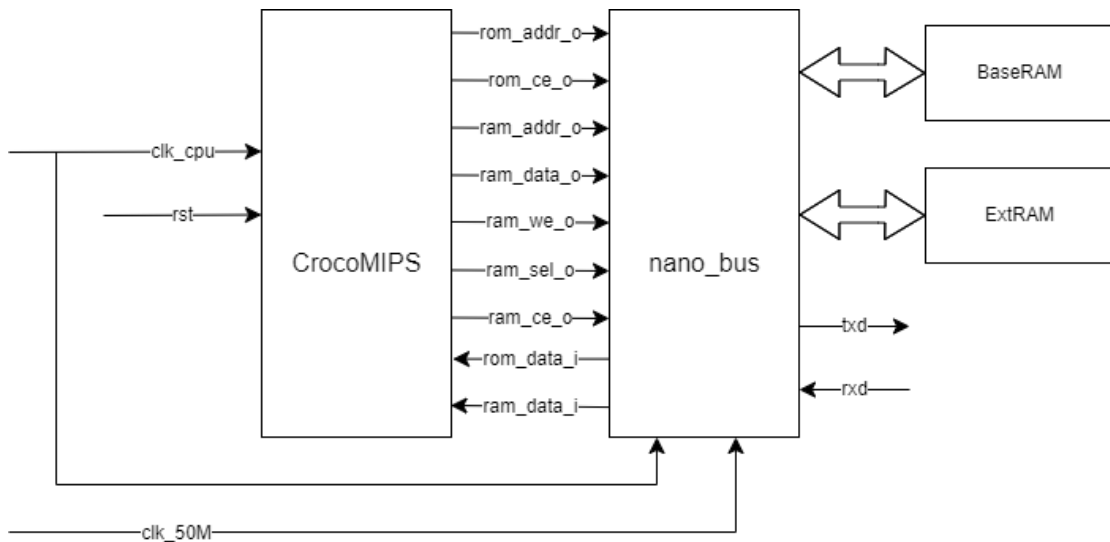
一、设计简介

CrocoMIPS 是一款符合第八届“龙芯杯”全国大学生计算机系统能力培养大赛个人赛技术方案的基于 MIPS-C3 指令集的 CPU，采用简洁而又经典的五级流水线单发射结构，采用数据前递来解决数据冒险，采用流水线暂停来解决取值和访存阶段同时访问 BaseRAM 的结构冒险和 Load 相关问题。该处理器使用带 FIFO 的串口进行通信，可以正常运行监控程序的各项指令，已通过三级测试，可以稳定在 58MHz 的频率下通过性能测试。

二、设计方案

（一）总体设计思路

本项目主要分为 CrocoMIPS 和 nano_bus 两个模块（图一），其中 CrocoMIPS 模块为处理器核心，nano_bus 模块接收处理器的访存请求并将其分配给 BaseRAM、ExtRAM 或是串口，串口频率固定为 50MHz 避免访问不稳定，同时带有 FIFO 来提高串口性能。



图表 1

（二）CrocoMIPS 模块设计

CrocoMIPS 为经典的五级流水处理器核心，分为 IF、ID、EX、MEM、WB 五个阶段（图二）。采用数据前递来解决数据相关，采用流水线暂停来解决访存结构冒险以及 Load 相关问题，采用数据前递而不是全部采用流水线暂停提高了 IPC，但是造成了较长的组合逻辑，在权衡利弊之后采用了当前方案。

pc_reg 即 IF 阶段，根据 ID 传递来的分支跳转信息更新 PC，访问指令存储器获取指令。

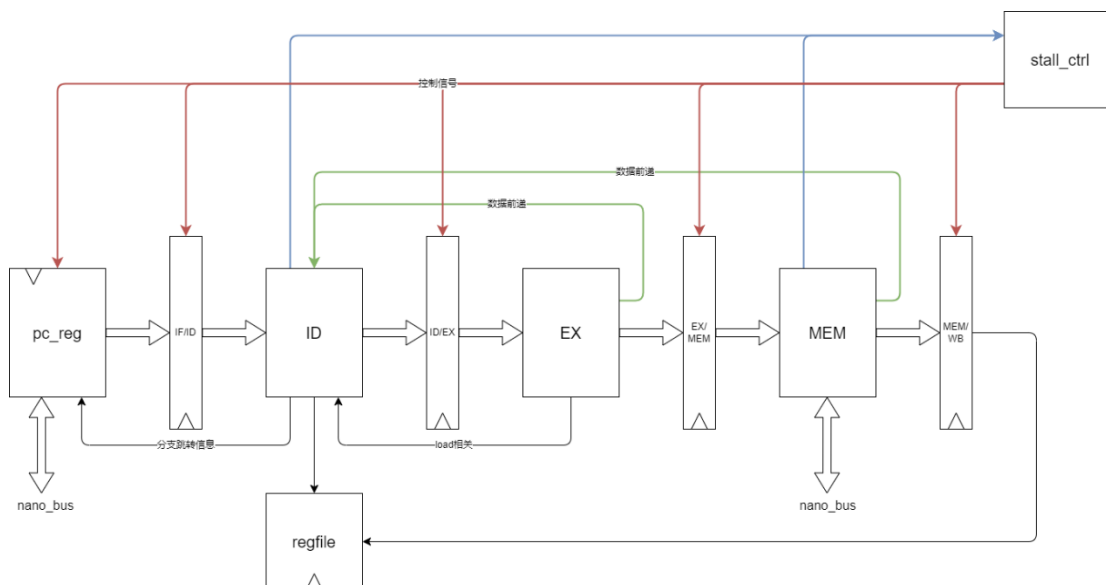
ID 阶段，对指令译码，生成操作码，源操作数等，同时需要进行分支判断和 load 冒险判断。

regfile 实现了寄存器堆，有两个读端口（组合逻辑）和一个写端口（时序逻辑），承担了 WB 写回阶段的功能。

EX 阶段，根据译码阶段的结果经行运算，给出访存信息或写寄存器信息。

MEM 阶段，根据 EX 阶段的结果访问数据存储器，生成 SRAM 的控制信号。

stall_strl 模块根据暂停请求类型生成控制码，控制流水线的特定阶段暂停，不暂停的阶段可以继续运行。在 pc_reg 取址阶段和 MEM 访存阶段通过类 SRAM 接口访问 nano_bus 来访问存储器和串口。



图表 2

（三）nano_bus 模块设计

Nano_bus 接收 CPU 通过类 SRAM 接口传输的访存请求，通过内存映射来选择访问 BaseRAM、ExtRAM 还是串口，同时在取址和访存阶段同时访问 BaseRAM 时考虑结构冒险。

处理访存结果的格式并在当周期返回给 CPU 的相应模块。

在发布包体提供的串口接收器和发送器基础上增加 FIFO 提高串口的性能，同时串口频率固定为 50MHz 以防止频率过高导致的串口访问不稳定。

三、设计结果

（一）设计交付物说明

```
├──constrs_1
|   └──new
|       thinpad_top.xdc           // 约束文件，添加延迟保持异步 SRAM 稳定
├──sim_1
|   ├──imports
|   └──new
|       |   28F640P30.v
|       |   clock.v
|       |   cpld_model.v
|       |   flag_sync_cpld.v
|       |   sram_model.v
|       |   tb.sv                 // 仿真文件
|       └──include
└──sources_1
    ├──CrocoMIPS
    |   croco_mips.v             // 处理器核模块
    |   defines.v               // 宏定义
    |   ex.v                    // EX 阶段
    |   ex_mem.v                // EX/MEM 级间寄存器
    |   id.v                    // ID 阶段
    |   id_ex.v                 //ID/EX 级间寄存器
    |   if_id.v                 //IF/ID 级间寄存器
    |   mem.v                   // MEM 阶段
```

```

|      mem_wb.v          //MEM/WB 级间寄存器
|      pc_reg.v         // IF 阶段
|      regfile.v        // 寄存器堆
|      stall_ctrl.v     // 流水线暂停控制
├──ip
|   ├──fifo_generator_0 // 串口所用 FIFO 缓存
|   └──pll_example      // 时钟分频
└──new
    async.v
    nano_bus.v          // 内存映射简单总线
    SEG7_LUT.v
    thinpad_top.v       // 顶层文件
    vga.v

```

注意：仿真时需要将 tb.sv 文件中的二进制机器码文件路径修改为自己电脑上欲运行文件的路径，同时将 defines.v 文件设置为 global include

（二）设计演示结果

一级测试：

33949	2024-05-31 19:01:25	↔ 46d108b4	Finished	100
-------	---------------------	------------	----------	-----

二级测试：

33183	2024-05-26 11:32:21	↔ d1fe2b3e	Finished	100
-------	---------------------	------------	----------	-----

三级测试：

33181	2024-05-26 11:30:46	↔ d1fe2b3e	Finished	100
-------	---------------------	------------	----------	-----

性能测试：

43108	2024-07-22 21:38:39	↔ b99f7f99	Finished	100
-------	---------------------	------------	----------	-----

（提交版本以平台上为准）

100

perf 在 FPGA 板 07 09 06 上的结果

```
=== Test CRYPTONIGHT ===
Boot message: 'MONITOR for MIPS32 - initial
User program written
Program Readback:
4080043cadde053cefbea534cefa063c0cb0c634
Program memory content verified
Data memory content verified
Test CRYPTONIGHT run for 0.398s
```

100

perf 在 FPGA 板 07 09 06 上的结果

```
=== Test MATRIX ===
Boot message: 'MONITOR for MIPS32 - initial
User program written
Program Readback:
4080043c4180053c4280063c60000724251800001
Program memory content verified
Data memory content verified
Test MATRIX run for 0.154s
```

100

perf 在 FPGA 板 07 09 06 上的结果

```
=== Test STREAM ===
Boot message: 'MONITOR for MIPS32 - initial
User program written
Program Readback:
1080043c4080053c3000063c2130860005008610
Program memory content verified
Data memory content verified
Test STREAM run for 0.108s
```

四、参考设计说明

- (1) 雷思磊. 《自己动手写 CPU》参考五级流水线结构和控制逻辑
- (2) https://github.com/xiazhuo/nscc2022_personal 参考串口同通信
- (3) 使用 fifo_generator IP 核

五、参考文献

- [1] 雷思磊. 自己动手写 CPU. 北京: 电子工业出版社,2014.
- [3] 汪文祥,邢金璋.CPU 设计实战. 北京: 机械工业出版社,2021.