**LoongArch微系统设计报告**

学校 北京邮电大学

姓名 张星昱

一、设计简介

本次提交的设计采用Chisel语言，设计了支持LoongArch-C3指令集的32位CPU，完成了初赛所需的22条指令。本设计采用了五级流水线顺序单发射模型，实现了简单动态分支预测，添加了直接映射icache和全相联写数据队列，使用SRAM协议作为总线协议，能够访问BASE\_RAM、EXT\_RAM以及带FIFO的UART控制器。最终可以在105MHz频率下通过三级功能测试以及性能测试。

二、设计方案

（一）总体设计思路

整个设计主要分为CPU与SoC两个部分，如图1所示。

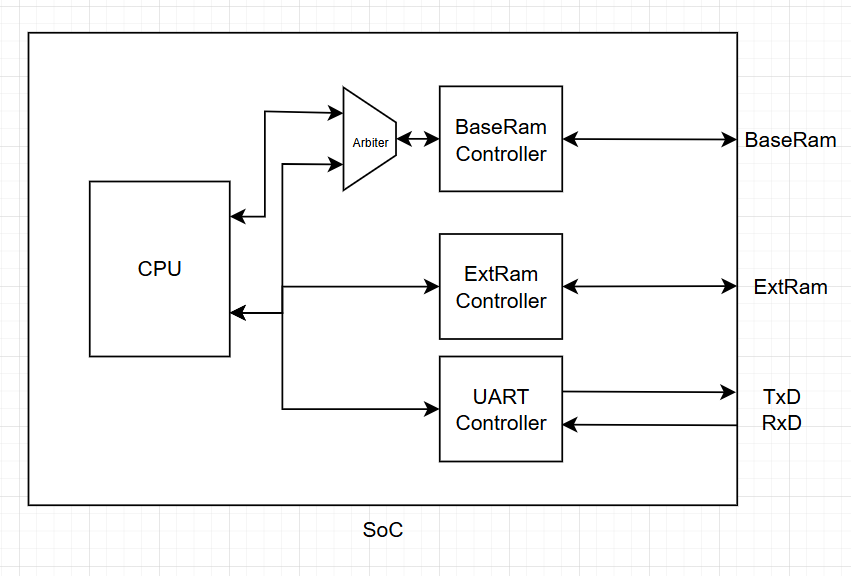


图1 SoC架构图

CPU部分拥有两个SRAM协议接口给取指和访存使用，SOC部分对这两个接口进行仲裁与地址分配。取指接口只能访问BASE\_RAM控制器；访存接口可以访问所有控制器，同时在访问BASE\_RAM控制器时，阻塞取指。

CPU采用顺序单发射五级流水线模型。各部分简述如下：

**取指（IF）部分**：访问icache并向SoC发出访存请求；接收分支单元的跳转结果，更新预测器；通过预测器决定下一个PC值；将icache或访存得到的指令、当前pc以及下一个pc发送到ID，供后续模块使用。

**译码（ID）部分**：将得到的指令译码；判断前递条件同时取数；将控制信号等发送到EX。

**执行（EX）部分**：根据译码进行计算，同时将操作数送往三级流水线乘法器；计算三种分支目的；将访存相关信号发送到LS，将分支相关信号发送到BR。

**访存/分支（LS/BR）部分**：此部分两模块并行处理。为了保证EX执行两周期后得到乘法器结果，访存模块统一设计为两个周期：若写队列命中，则多流水一周期后送往WB，否则停顿一周期后把从SoC得到的访存结果送往WB；分支模块根据EX的结果生成最终分支目的送往IF；

**写回（WB）部分**：得到ALU、访存以及乘法器结果，根据控制信号选择最终数据写回寄存器。

CPU架构图如图2所示。

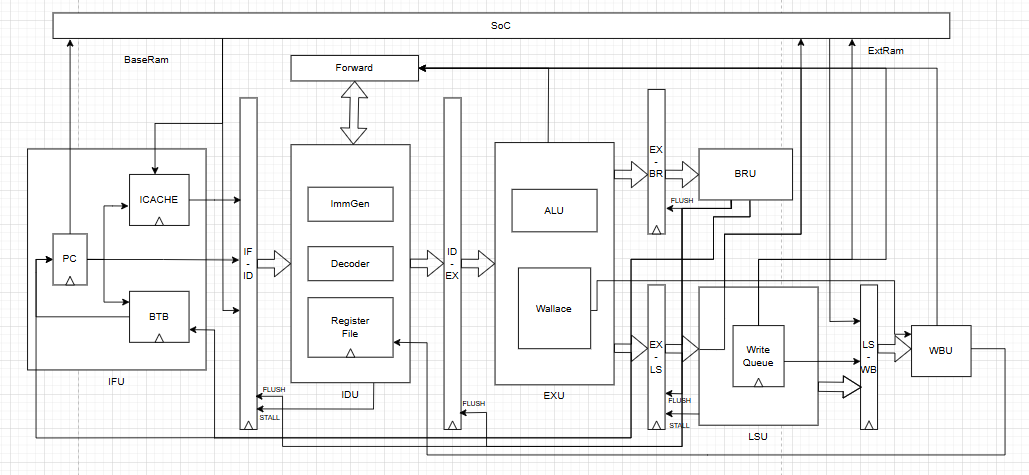


图2 CPU架构图

（二）指令Cache模块设计

由于SRAM读写最少需要两周期且不可流水化，因此取指令Cache的块大小为32位，共保存64块。为了方便实现，映射方式为直接映射。

本模块调用了Vivado提供的Distribute Memory Generator IP核。该IP核生成了64×48bit的LUTRAM，可以视作寄存器，能够在一个周期内读出数据，节省了取指周期。本模块使用的表结构如表1所示：

表1 icache表项结构

|  |  |  |  |
| --- | --- | --- | --- |
| Preserve(1bit) | Valid(1bit) | Tag(14bit) | Instruction(32bit) |

在取指时，将PC[7:2]作为地址，组合地读出Instruction并判断Tag与Valid。若是符合，则IF不停顿；若不符合，则停顿一周期，等待SRAM数据，并在数据到来时更新表项。

（三）分支预测器设计

分支预测器采用了分支缓存表（BTB）加两位饱和计数器（PHT）的预测方式。BTB共64项，采用直接映射方式。出于综合面积与延迟等考虑，PHT只采用64项，直接映射方式，且没有全局/局部历史记录。

分支缓存表调用了Vivado提供的Distribute Memory Generator IP核。IP核例化属性同icache，定义的表结构如表2所示：

表2 BTB表项结构

|  |  |  |  |
| --- | --- | --- | --- |
| Valid(1bit) | Type(1bit) | Tag(14bit) | Address(32bit) |

在取指时，将PC[7:2]作为地址，组合地读出Address并判断Tag与Valid。若是符合，则根据PHT得到跳转目的更新PC。

当BR部分valid时，BTB得到相应的PC、跳转类型以及跳转目标，更新表项。同时PHT作相应的状态变化。

（四）乘法器设计

乘法器采用2位Booth编码Wallace树的三级流水线设计。其中第一级保存16个Booth编码结果，第二级保存Wallace树结果，最后一级保存加法器结果。乘法器在EX阶段取数，统一在WB阶段得到数据。

（五）写队列设计

为了让访存阶段阶段尽可能减少停顿，这里只为ST类指令添加了长为4的带全相联比较写队列。后续可以根据情况为LD指令添加dcache提高IPC。

由于写操作是可以后台进行的，当写队列未满时，可不用停顿，将数据直接存入写队列，让写队列接管LS的访存接口。当写队列满时，停顿直到可以入队。

当读操作valid时，停止写队列，让LS模块接管访存接口向SoC发送读请求，同时将读地址和写队列中的地址比较。若命中，则读操作不停顿，否则停顿直到SoC返回数据。

（六）UART模块设计

UART模块在样例工程给出的基础上，为传输模块和接收模块分别添加长度为4的FIFO，一定程度上优化了性能。

三、设计结果

（一）设计交付物说明

本设计项目文件结构如下所示：

2024576/

├─ .ci-scripts/ // CI脚本文件夹

├─ asm/ // 汇编程序文件夹

├─ core/ // Chisel项目/Verilator仿真文件夹

├─ core/src/ // Chisel项目源文件夹

├─ include/ // Verilator仿真头文件夹

├─ src/ // Verilator仿真源文件夹

├─ modules/ // Chisel生成文件以及其他verilog文件夹

├─ Makefile // Chisel生成、Verilator仿真脚本

└─ build.sc // Chisel项目构建文件

├─ thinpad\_top.srcs/ // Vivado项目源文件夹

├─ thinpad\_top.xpr // Vivado项目文件

└─ design.docx // 设计报告

Chisel项目编译：在linux上安装Mill与JDK17，在./core下运行make verilog，生成的verilog文件放在./core/modules/下。

Verilator仿真：在linux上安装Verilator，在./core下运行make verilator，即可在./core/build下生成对应二进制文件。使用方法：./core/build/top `PATH\_TO\_BINARY`

Vivado综合：在./core下运行make copy，即可将生成的verilog文件放到Vivado项目源码文件夹中，在Vivado下综合即可。

（二）设计演示结果

性能测试的运行结果如下图所示：



图3 STREAM程序

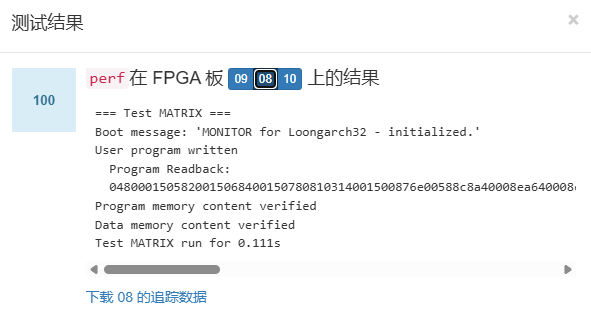


图4 MATRIX程序



图5 CRYPTONIGHT程序

四、参考设计说明

五级流水线的划分参考了计算机组成与设计：硬件/软件接口（第 5 版）[1]。

乘法器参考了中国科学技术大学《计算系统综合实验》Lab8中《乘法器与除法器的设计[2]》。

分支预测器参考了中国科学技术大学《计算系统综合实验》Lab8中《分支预测器的设计[3]》

UART控制器实例化了样例项目的async.v

五、参考文献

[1] David A. Patterson, John L. Hennessy. 计算机组成与设计：硬件/软件接口（第 5 版）. 王党辉等译. 机械工业出版社.

[2] 马子睿. 乘法器与除法器[EB/OL]. 2023[2024-8-4]. https://soc.ustc.edu.cn/CECS/lab8/board.assets/乘法器与除法器的设计.pdf

[3] 马子睿. 分支预测器的设计[EB/OL]. 2023[2024-8-4]. https://soc.ustc.edu.cn/CECS/lab8/board.assets/分支预测器的设计.pdf