**Stupid-MIPS设计报告**

复旦大学

李明辉

一、设计简介

Stupid-MIPS为一款单发射、五级流水线结构的32位MIPS指令集CPU，支持25条指令。本设计支持延迟槽技术解决分支跳转过程中的控制冒险，数据前递技术解决数据相关问题，尽可能减少流水线因冒险导致的停顿。已通过三级评测和性能测试，受限于SRAM访问瓶颈，目前最高频率58MHz。

二、设计方案

（一）总体设计思路

Stupid-MIPS顶层模块thinpad\_top下包含CPU核和SRAM与UART串口控制器两部分。CPU核为典型的五级流水线结构，由取指、译码、执行、访存、写回以及寄存器文件和中间值寄存器组成，除了含有寄存器功能的模块为时序逻辑电路，其余均设计为组合逻辑电路。

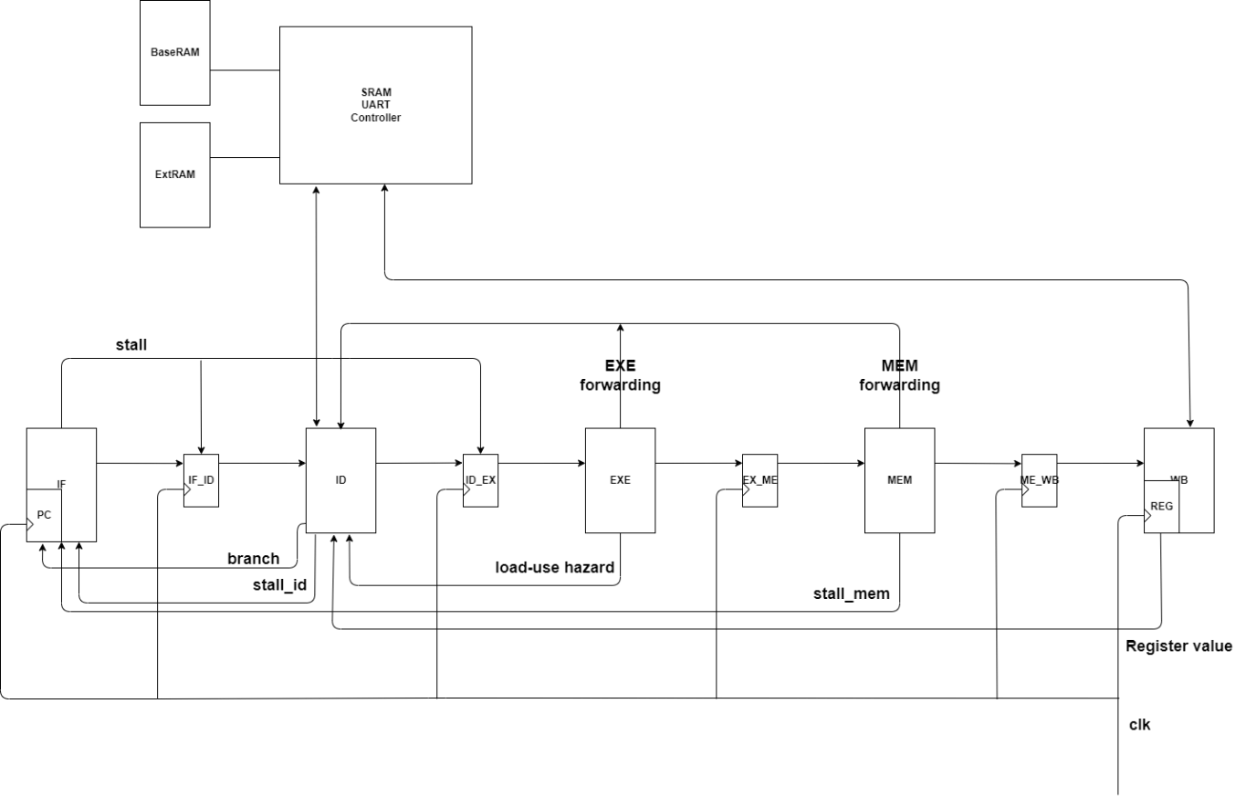


图1：Stupid-MIPS设计

1、取指模块IF

兼负pc寄存器功能，故为时序逻辑电路，控制pc与BASERAM中指令的读取并传入中间值寄存器，后期优化时缩短关键路径将BASERAM读出指令直连到IF\_ID\_REG中间值寄存器。

接收来自译码阶段检测load-use 冒险和访存阶段结构冒险的结果，生成并向中间值寄存器发出流水线暂停的信号。

2、译码模块ID

组合逻辑电路，根据输入的指令和pc进行译码，将准备好的操作数等信息写入中间值寄存器ID\_EX\_RG。

支持数据回传，读寄存器时会检查此时执行和访存阶段的指令是否有写寄存器的请求，根据检查结果准备操作数解决数据冒险。

根据条件确定跳转指令，并将地址传回IF。

通过查看上一条指令是否为load指令以及写入地址判断load-use冒险，并向IF模块传递检测结果。

3、执行模块EXE

组合逻辑电路，根据传入操作数和ALU操作类型完成计算，与访存相关的信号在该阶段计算生成，将计算结果和访存信息传入中间值寄存器EX\_MEM\_REG。同时计算结果和写寄存器地址前递到译码阶段。

4、访存模块MEM

组合逻辑电路，根据访存类型处理内存映射问题，并生成访存地址、芯片使能信号和片选信号传递给SRAM控制器，将返回结果写入中间值寄存器MEM\_WB\_REG。

通过检测访存地址是否为BASERAM范围判断是否发生结构冒险，并向IF模块传递检测结果。

5、写回模块WB

根据写寄存器地址将数据写入寄存器。

将寄存器文件合并至WB模块中，为时序逻辑电路。若在同周期读写相同地址，读出内容连接所写数据，支持在同一周期完成读写操作。

6、中间值寄存器xx\_xx\_REG

时序逻辑电路，时钟上升沿接收前一模块数据，并将原先数据向后传递。接收暂停信号时IF\_ID\_REG保持不变，ID\_EX\_REG向EXE传空值，其余不受影响，以此通过流水线暂停解决load-use冒险和结构冒险。

7、SRAM与UART串口控制器RAM\_Serial\_Controller

接收CPU核访存和串口信息，进行地址映射后从SRAM中读取数据返回。在示例基础上添加FIFO缓冲模块提高稳定性。

三、设计结果

（一）设计交付物说明

提交设计目录层次如下图：

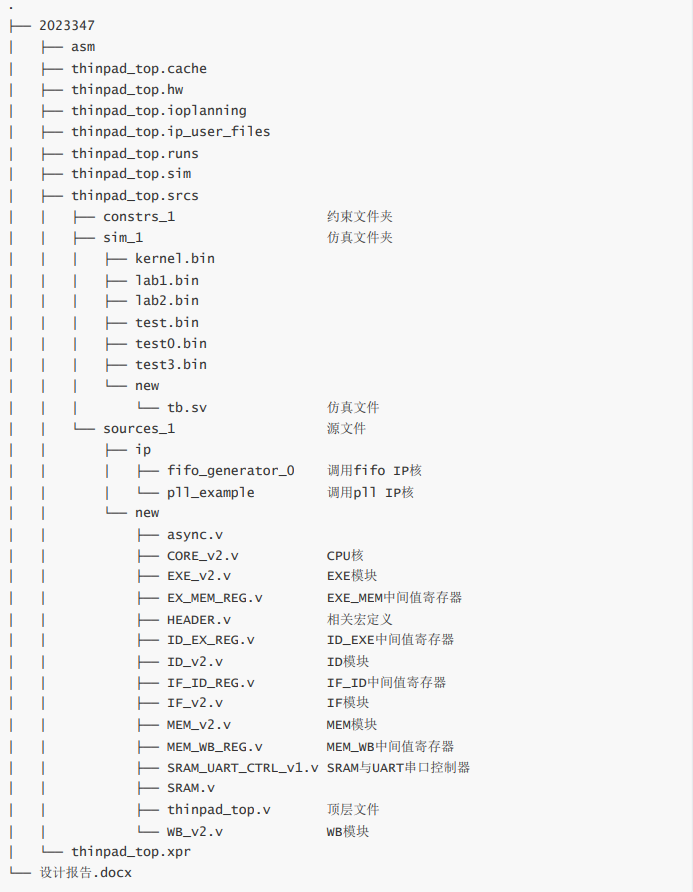


图2：文件目录

有关源码部分仅需关注thinpad\_top.srcs目录中的内容；

仿真时可以修改tb.sv文件中.bin文件所在目录对不同测试程序进行仿真，仿真文件夹下test.bin，test0.bin，test3.bin为debug过程中从测试程序中拆分的部分片段；

Vivado2019.2打开thinpad\_top.xpr生成bit流文件上传远程实验平台即可开始实验；

（二）设计演示结果

功能测试lab3结果：

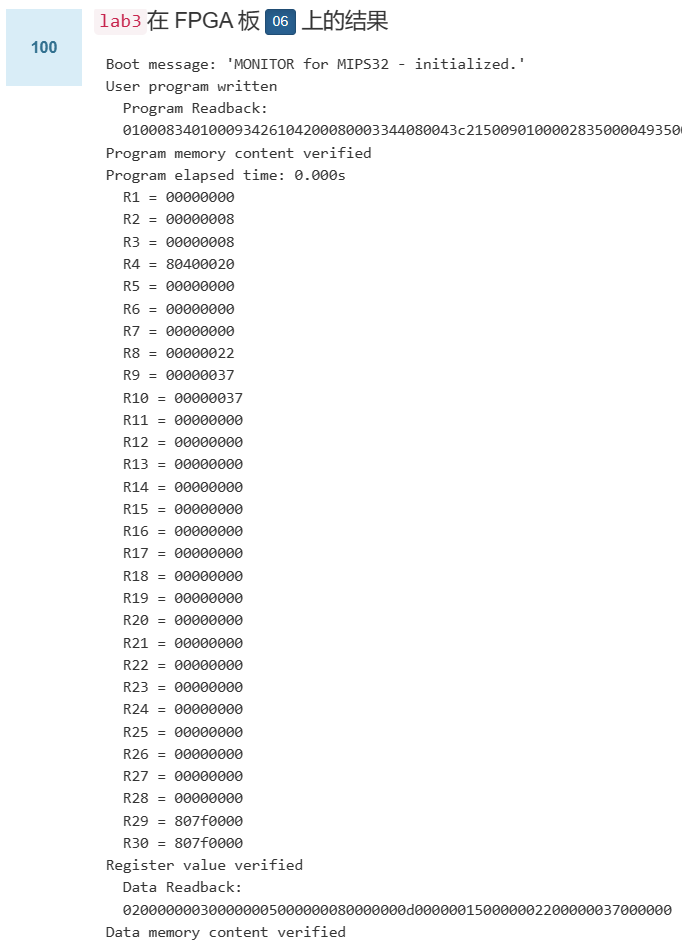


图3：功能测试lab3

性能测试数据见下表：

表1：性能测试数据

|  |  |  |  |
| --- | --- | --- | --- |
| 测试程序 | STREAM | MATRIX | CRYPTONIGHT |
| 时间 | 0.108s | 0.154s | 0.398s |







图4：性能测试结果

四、参考设计说明

Stupid-MIPS在比赛提供的串口模块基础上增加Vivado FIFO IP核；

SRAM和UART串口控制器参考2020年开源代码思路，将两个控制器整合，同时处理BASERAM、EXTRAM和串口数据，根据CPU核传入的访存类型请求信号处理输出结果。参考链接：https://github.com/xiazhuo/nscc2022\_personal

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.

[2]胡振波. 手把手教你设计CPU：RISC-V处理器. 异步图书，2018.

[3] Randal E.Bryant / David O'Hallaron. 深入理解计算机系统. 机械工业出版社，2014.