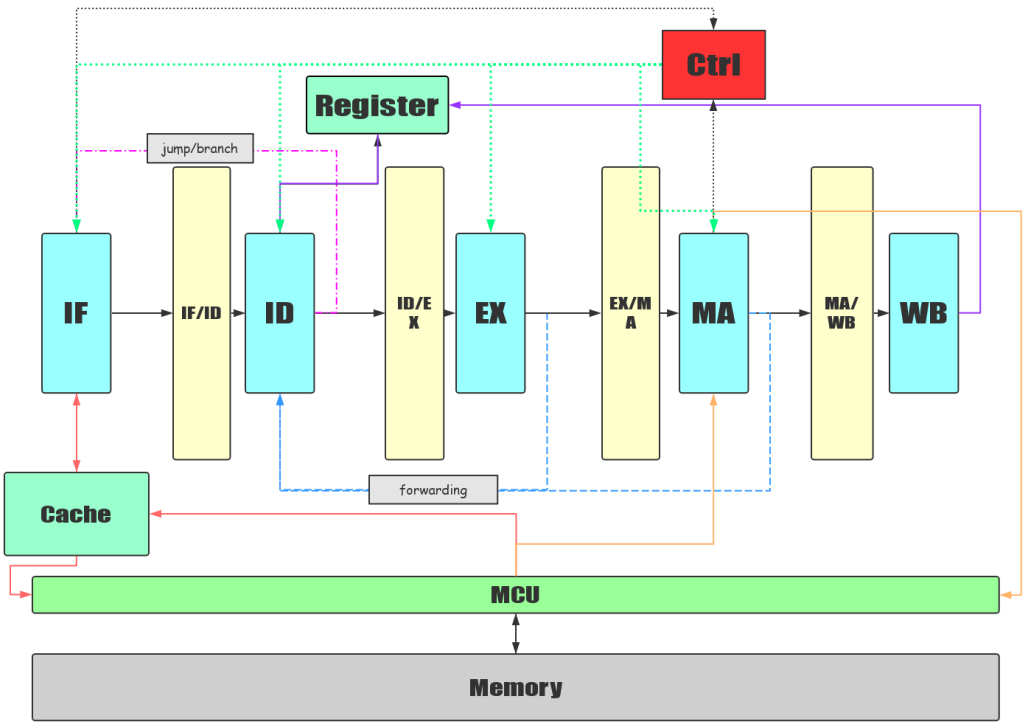
RISCV-32I指令集实现

简介

> 总体上实现了一个能够运行RV32I指令集的cpu架构，采用五级流水的结构，并带有1KB的instruction cache，用verilog HDL实现，可以在Basys3FPGA开发板上运行。

架构



主要特点

> - 带有1024Byte的指令缓存，可以实现一周期内的指令读取
> - 在ID阶段进行branch和jump语句的判断，并向IF回传地址
> - 支持任意长度字节读取的MCU(需要指定常数)
> - 使用forwarding解决非load数据相关，stall解决load相关和控制相关

代码实现相关问题

MCU的构建

> 使用了由时钟控制的状态机，内存交互n Byte需要(n+4)周期(其中两周期用来进行MCU与上层模块的交互和数据传递)。状态机分监听请求，读，写和暂停四个状态，由外部条件和计数器控制状态转移。

Cache的设计

> Cache使用直接映射，每行四条指令，加上8bit的tag和1bit的valid位，共64行，可用空间1024Byte。Cache使用组合逻辑实现，当Cache miss是进入内存获取数据，并暂停流水线。

对跳转指令的处理

> 由于内存读取较慢，为了尽量减少stall带来的时间浪费，面对分支语句时采用了暂停IF的方法，等待ID给出跳转指令的地址和是否跳转。然而这样的设计在时序的收敛上造成了一定的困难。

优化时序的小技巧

- 将一个状态机中只在极少数状态中改变的变量单独使用组合逻辑实现，时状态机更加紧凑。
- 降低信号的扇出，可以使用预编译指令或编译优化指令。
- 提高并行度，将关键路径中延迟最大的提出，使用组合逻辑提前并行运算。
- 提高表达式并行度，使用括号手动将表达式的表达式树划分为平衡二叉树。

参考资料

> - risc-v工具链的安装和使用
> - risc-v文档
> - 自己动手写cpu
> - Xilinx FPGA开发实用教程

- Verilog编程艺术
- 部分汇编测试数据