

简单的类 MIPS 单周期处理器部件实现——控制器， ALU

目的

设计模块实现 CPU 中控制器、算术-逻辑单元(ALU)及其控制单元的逻辑。

设计思路

根据指令表编写对应的控制信号及运算结果即可。

模块描述

控制器接收输入为指令最高 6 位，根据这 6 位的指令类型输出各类控制信号。在代码中由 case 语句实现。

算术-逻辑单元控制器接收一个来自控制器的 2 位操作码、一个来自指令最高 6 位的指令码和一个来自指令最低 6 位的额外操作码，以决定向算术-逻辑单元输出的具体操作码。在代码中用 if-else 语句实现控制。

算术-逻辑单元接收输入为 2 个 32 位运算数和一个来自算术-逻辑单元控制器的 4 位操作码，根据操作码对两个运算数进行一定的运算后将结果输出，并额外输出结果是否为 0 以供 beq 指令参考。在代码中用 if-else 语句实现控制。

仿真描述

控制器仿真采用 5 种实现的指令的控制信号模拟：nop 或 R-type，lw，sw，beq，j。

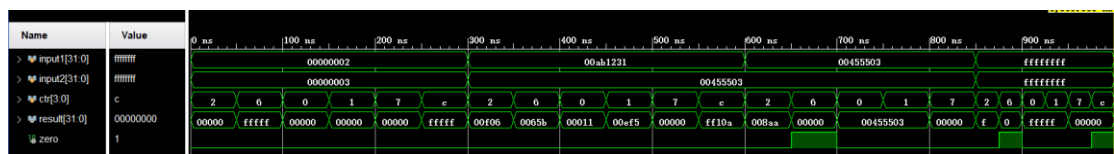
算术-逻辑单元控制器仿真结合来自控制器的 2 位 aluOp 和指令最低 6 位 func，有多种情况，由于其中有大量 don't care 情况，故在仿真中采用了 7 种组合。

算术-逻辑单元仿真考虑 3 种情况，对于输入 A、B，可能有 $A < B$ ， $A > B$ 和 $A = B$ 的情况，而进行的操作包含加、减、与、或、 $A < B$ 时输出 1、或非等情况。仿真中包含了以上所有 18 种情况。

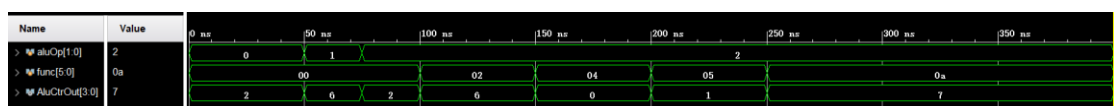
实验结果

仿真波形如下所示（由于软件显示问题，显示的值可能并非完整值）。

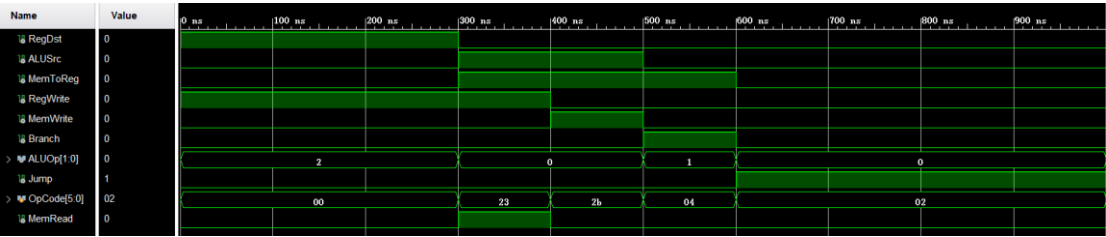
算术-逻辑单元波形：



算术-逻辑单元控制器波形：



控制器波形：



通过分析相应仿真波形，结果正确。

总结

本次实验较为简单。通过 3 个核心单元的模拟，加深了对于 CPU 内部元件运行逻辑的理解，加强了对于 vivado 软件使用的掌握。在后面的实验中，由于功能需要进一步扩充以适应整体 CPU 的运行需要，我对这些部件进行了不同程度的改进和封装，但其核心部分都是以本次实验为基础进行的。因此，本次实验对后面的实验有非常重要的作用。