

实验 2：ALU 模块实现

MIPS (Microprocessor without Interlocked Pipeline Stages), 是一种采取精简指令集 (RISC) 的指令集架构 (ISA), 由美国 MIPS 计算机系统公司开发。MIPS 广泛被使用在许多电子产品、网络设备、个人娱乐设备与商业设备上。最早的 MIPS 架构是 32 位, 最新的版本已经变成 64 位, 其商业市场主要竞争对手为 ARM 与 RISC-V。在国内外一些著名大学中计算机架构的课程上, 学生们通常会首先学习 MIPS 架构。这个架构极大地影响了后来的精简指令集架构, 如 Alpha。我国的龙芯 CPU 在起步的时候, 也采用 MIPS 架构, 2020 年 8 月龙芯推出了自主设计的 CPU 指令集——LoongArch, 其中依然可以看到 MIPS 一些设计理念。32 位 MIPS 1 代的指令集共有 31 条, 本次实验需要同学们实现 MIPS 指令中的 ALU 指令所对应的 ALU 计算单元。

3.4.1 实验目的

1. 熟悉 MIPS 指令集里的运算指令, 学会对这些指令进行归纳分类。
2. 了解 MIPS 指令结构, 熟知指令类型, 了解指令功能。
3. 熟悉并掌握 ALU 的原理、功能和设计。
4. 通过 Verilog HDL 方式实现 ALU 程序设计, 通过 Vivado 软件进行开发调试。
5. 为后续 CPU 设计的实验打下基础。

3.4.2 实验设备

1. 装有 Xilinx Vivado 的计算机一台。

3.4.3 实验要求

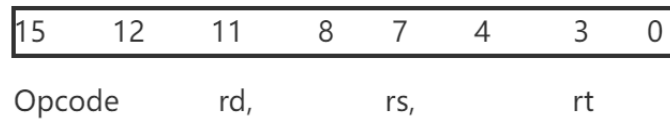
1. 学生要实现以下这些指令

表 1. Opcode 表

Function	Opcode
AND	0000
OR	0001
XOR	0010
NOT	0011
ADD	0100
SUB	0101
SRA	0110
SLL	0111

表格中一共有八个算术和逻辑指令。它们是 ADD、SUB、AND、OR、XOR、NOT、SRA 和 SLL。其中 ADD、SUB、AND、OR、XOR 和 NOT 指令具有三地址格式。这些指令的汇编级语法是

Opcode rd, rs, rt



两个操作数是 (rs) 和 (rt)，目标是寄存器 rd（注意：对于 NOT，rt 是不关心的可以设置为任意值 X）。ADD 和 SUB 指令分别以二进制补码的方式将两个操作数相加和相减，并将结果保存在寄存器 rd 中。AND、OR 和 XOR 指令分别对两个操作数进行按位与、按位或和按位异或运算，并将结果保存在寄存器 rd 中。NOT 指令反转 rs 并将其存储在 rd 中。

ADD、SUB、AND、OR、XOR 和 NOT 指令还设置或清除**状态字寄存器 PSW** 中的零 (Z)、溢出 (V) 和符号 (N) 位。

- 当且仅当操作的输出为零时设置 Z 标志。
- 当且仅当操作导致溢出时，V 标志由 ADD 和 SUB 指令设置。如果遇到 AND、OR、XOR 和 NOT 指令在执行的时候需要清除 V 标志。
- 当且仅当 ADD 和 SUB 指令的结果为负时，才会设置 N 标志。AND、OR、XOR 和 NOT 指令清除 N 标志。

除了上一段中描述的那些之外，没有其他指令设置或清除 **PSW** 标志。

2. SRA 是算数右移指令， SLL 是逻辑左移指令

SRA 和 SLL 指令具有以下汇编级语法。

Opcode rd, rs, imm

其中 imm 是 SRA 和 SLL 指令的无符号表示的 4 位立即数 (Immediate)。SRA 和 SLL 按 imm 字段中指定的位数移位 (rs)，并将移位后的结果保存在寄存器 rd 中。SRA 是右移算术，SLL 是左移逻辑。SRA 和 SLL 指令保持标志不变。

3. 八个算术指令的机器级编码是

0aaa dddd ssss tttt

其中 aaa 代表操作码（见表 1），dddd 和 ssss 分别代表 rd 和 rs 寄存器。tttt 字段代表 rt 寄存器或 imm 字段。

4. 请实现这 8 种 ALU 运算，列出操作码的编码（），其中包括加减运算，其中减法在内部要转换为加法，与加法运算共同调用实验一里的加法模块。
5. 8 种的 ALU 运算均需要两个时钟周期才能完成
6. 要求设计一种指令译码器，可以根据 Opcode 来产生该指令所对应的控制信号。
7. 使用 Verilog 编写或优化相应代码 alu.v。
8. 编写仿真代码，得到正确的测试波形图，来测试每一条指令
9. 将以上的所有指令的设计在一个单独的 ALU 模块，设计一个外围模块 (TestBench) 去调用该模块，完成设计并编写代码 alu_testbench.v，在 TestBench 中需要对每一条

指令进行 8 个测试用例，每一次测试要求均输入的是随机数；TestBench 也需要提供操作数。

10. 请描述测试原理，并给出波形图，检查运算结果的正确性。

3.4.4 实验步骤

参考实验一的实验步骤：

1. 建立完整的工程，工程中包括 alu 模块、testbench.v（仿真文件）
2. 完成仿真，并将仿真波形图像截图，示例图为加法运算波形仿真图。

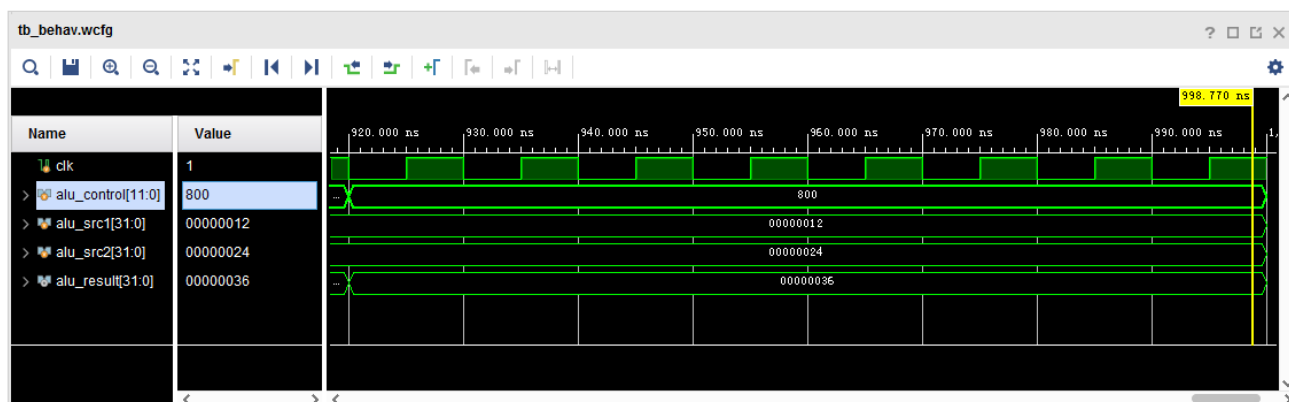


图 3-4-5 加法仿真波形图

3.4.7 实验报告需包含以下内容：

1. 实验目的
2. 实验设置
3. 实验内容
 - A. 列出你设计的运算种类
 - B. 尝试优化 alu.v 模块代码的时序（其他部分也可以改写或优化，包括必要的注释）
 - C. 编写仿真模块 testbench.v 代码
 - D. 功能仿真测试时序图截图
4. 实验小结