
计算机组成原理

实验 2: 运算器 ALU 实验, 熟悉 Vivado 环境

实验报告

李祥泽

2018011331

lee_johnson@qq.com

实验目的

1. 熟悉硬件描述语言;
2. 熟悉开发环境 Vivado, 了解硬件系统开发的基本过程;
3. 掌握 ALU 的基本设计方法和简单运算器的数据传送通路;
4. 验证 ALU 的功能.

实验内容

1. 根据实验原理中的要求, 用 Verilog 语言实现一个简单的 ALU;
2. 在教学计算机 THINPAD-Cloud 上验证实现的 ALU 的功能.

实验步骤

1. 用 Verilog 语言编写 ALU 功能代码, 并用状态机对其进行控制, 使其完成实验要求的操作.
状态码和操作数的输入用微型开关 SW0 ~ SW15, 计算结果的输出用教学机上的 LED 灯来显示.
2. 将代码下载到教学机的 FPGA 芯片中, 并调试完成.
在 THINPAD-Cloud 教学机上运行时, reset 和时钟均用手动开关或按钮, 便于演示. 操作码和操作数在拨码开关 SW0 ~ SW15 上输入; 为便于观察和调试, 每次 ALU 得到操作数, 最好可以在 LED 上显示一下. 最后的运算结果在 LED (L0 ~ L15) 上显示, 溢出标志位 (仅针对加减运算) 在 L0 上显示.
3. 记录实验结果.

实验过程

使用提供的 ALU 示例代码¹中的状态机来控制输入和输出.

具体地说, 补全了状态转移部分 case 代码段的情况 S1, S2, S3; 修改了 S0 和 RST 代码段使其更符合 (我的) Verilog 编程习惯.

状态机的输出生成部分基本保持不动, 其中与 ALU 有关的代码拆分到文件 `ALU.v` 中.

在 `ALU.v` 中完成了操作码的定义和不同操作码下对操作数的计算, 以及在数字运算时标志位的生成; 其中循环左移和算数右移是另外设计模块实现的. 值得说明的有以下两点. 第一, 在补码 (有符号数) 运算中, 溢出标志不能用延长一位来计算, 而是要对操作数和结果的符号做比较. 第二, Verilog 语法中虽然定义了算术右移运算符 `>>>`, 但它综合出的结果仍然是逻辑右移; 想要实现算术右移需要自己另外写模块.

实验结果

本实现通过了 ThinPAD-Cloud 自动评测. 以下另附几组作为示例.

操作码	操作数 A	操作数 B	运算结果	标志位	与预期一致性
0001 ADD	E4F5	1331	F826	0	一致
0010 SUB	7FFF	FFFF	8000	1	一致
0101 XOR	0251	2333	2162	0	一致
1010 ROL	137F	FFF4	37F1	0	一致

思考题

1. ALU 进行算术逻辑运算所使用的电路是组合逻辑电路还是时序逻辑电路？

ALU 进行运算所使用的电路全部是组合逻辑电路, 包括移位运算.

2. 如果给定了 A 和 B 的初值, 且每次运算完后结果都写入到 B 中, 再进行下次运算. 这样一个带暂存功能的 ALU 需要增加一些什么电路来实现？

暂存或记忆功能必须要由时序逻辑来实现. 如果仅实现题目所述功能, 可以增加两个寄存器用来保存 A 和 B, 再加一个寄存器保存 ALU 的计算结果. 其中 A 寄存器的输入连外部赋值输入; B 寄存器的输入由一个 2 选 1 MUX 提供, 一路连结果寄存器, 另一路连外部赋值输入, 选择信号也由外部提供.

首先 MUX 选择外部赋值, 初值给定之后给一个时钟沿, 让数据进入寄存器和 ALU; 计算结果稳定之后给第二个时钟沿, 让计算结果存入寄存器; 最后 MUX 选计算结果, 给第三个时钟沿, 计算结果进入 B 寄存器.

1. 这个“仅供参考”的示例代码, 和测例或实验指导书的差距实在大得一言难尽. 属实是仅供参考. [🔗](#)