ALU 设计

本设计的测试的整体方法请见顶层目录的 `Overview.pdf` 。

功能选择

组合逻辑功能可以使用 `assign` 或者 `always` 实现。ALU 的核心在功能选择上，可以使用 `case` 实现。

为了减少存储器件的使用，应该使用 `assign` ，事实上这也是一些现代 CPU 设计工具推崇的做法：

```
assign ans = \
    op == 0? ~a  : \
    op == 1? ~b  : \
    op == 2? a&b : \
    op == 3? a|b;
```

但是这样语法在 `Verilog` 中过于丑陋，并且可读性较差，因此选择引入一个锁存器通过 `case` 完成组合逻辑设计。

```
always @(*) begin
    case (op)
        2'b00: ans = ~a;
        2'b01: ans = ~b;
        2'b10: ans = a & b;
        2'b11: ans = a | b;
        default: ans = 0;
    endcase
end
```

测试

执行 `python` 文件获得随机数据，执行测试用 `verilog` 文件，读取数据以及对应的输出，通过则显示 passed ，未通过则显示错误发生的位置与状态，方便在波形中定位。输出波形在 `test.wlf` 与 `test.vcd` 中，数量过大，因此未添加到本文档中。