中国科学院大学计算机组成原理实验课

实 验 报 告

学号：2018K8009929021　姓名：袁欣怡　专业：计算机科学与技术

实验序号：1　实验名称：基本功能部件设计

一、 逻辑电路结构与仿真波形的截图及说明（比如关键 RTL 代码段{包含注释}及其对应的逻辑电路结构、相应信号的仿真波形和信号变化的说明等）

1.寄存器堆 reg_file

```
always@(posedge clk)
begin
    if (rst)
        r[0]<=0;
    else if (wen) begin
        if (waddr==0) r[0]<=0;
        else r[waddr]<=wdata;
    end
end
```

当 rst 信号为 1 时，重制 0 号寄存器中的值。

否则，在 wen 信号为 1 时：若写入地址不为 0，向相应的寄存器中写数据；若写入地址为 0，0 号寄存器仍为 0。

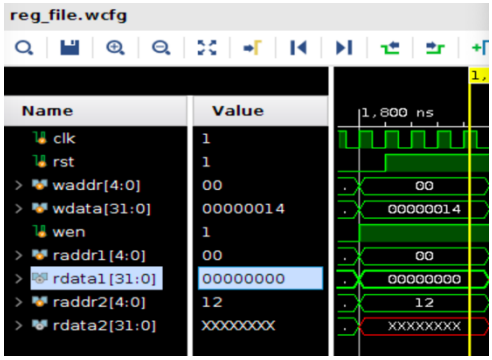

图 1：0 号寄存器中储存的值始终为 0

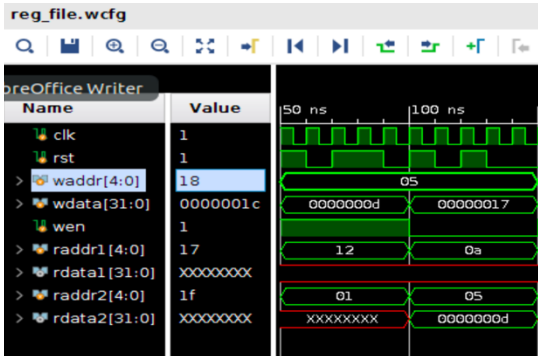

图 2：数据的写入和读取
（图中显示对 5 号寄存器的读写）

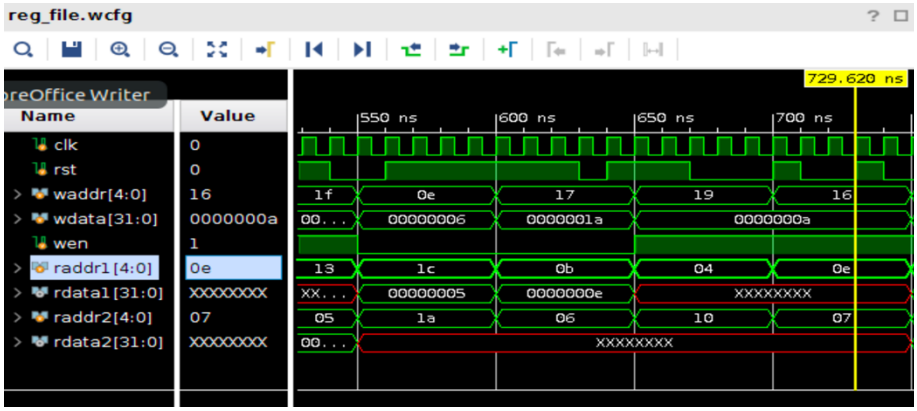

图 3：wen 的作用（550ns 时写 14 号寄存器，但是 wen=0，因此在 700ns 时读取 14 号寄存器仍为空）

2.算数逻辑单元 alu

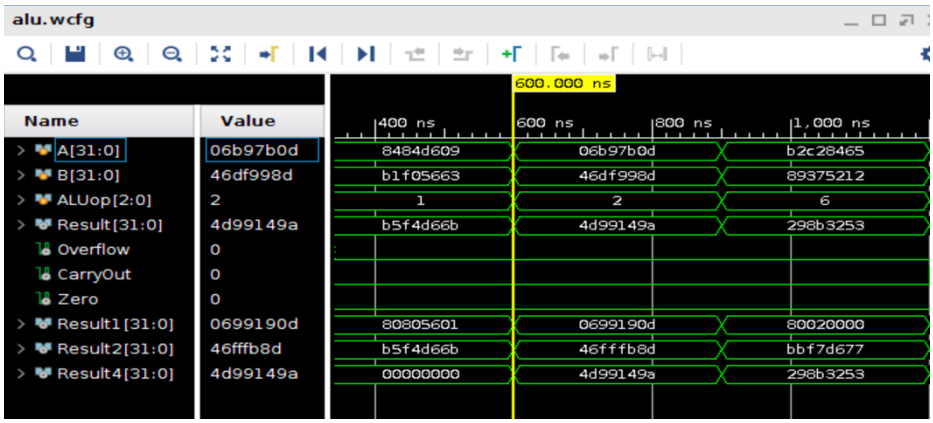

图 4：同时计算出 Result1，Result2 和 Result4，再由 ALUop 决定输出哪一个。

```
    assign carryout1=(B==32'b0)?1:0; //the carryout when turning B into B2
    assign B2=~B+32'b1;
    assign B1=(ALUop==3'b110 || ALUop==3'b111)? B2:B;
    assign {carryin,Result3[30:0]}=A[30:0]+B1[30:0];
    assign {carryout2,Result3[31]}=A[31]+B1[31]+carryin;
    assign carryout3=((ALUop==3'b110 || ALUop==3'b111) && B==32'h80000000)?
~carryout2:carryout2; //when B=32'h80000000, carryout3 is the negation of
carryout2
```

对于大部分情况来说，~B+1 和 B 的第一位不同，且“取反加一”的计算过程不会产生进位。但是有两个例外，即 B=0 和 B=32`h80000000。用 carryout1 和 carryout3 对这两种情况单独处理。

```
    assign Result=(ALUop==3'b000)?Result1:((ALUop==3'b001)?Result2:((ALUop==3'b010
|| ALUop==3'b110 || ALUop==3'b111)?Result4:32'b0));
    assign Zero=(Result==0)?1:0;
    assign Overflow=(ALUop==3'b010 || ALUop==3'b110 || ALUop==3'b111)?overflow:
1'b0;
    assign CarryOut=(ALUop==3'b010)?carryout2:((ALUop==3'b110 || ALUop==3'b111)?
~carryout:1'b0);
```

根据 ALUop 的不同值给 Result，Overflow 和 Carryout 赋值时，需要注意考虑非法输入。（我本来将非法输入时的值处理成 Z，后来老师指出不能在代码中含有 X、Z 等符号，要保证输出都是有效值。）

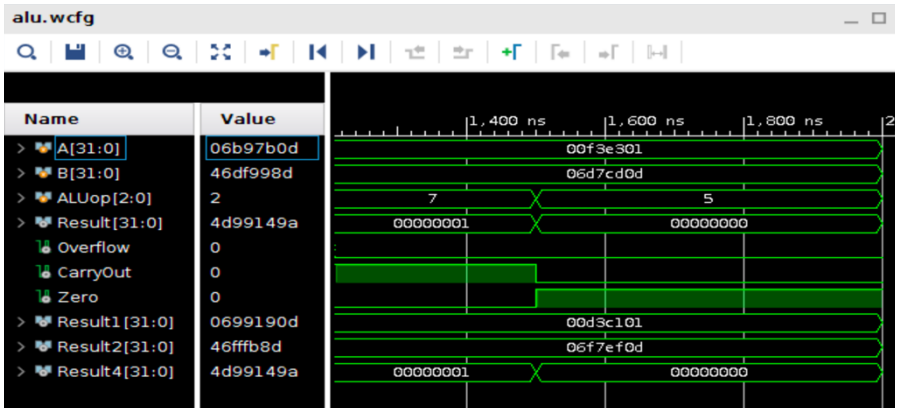

图 5：最后 ALUop=5 时为非法输入，此时输出为 0。

二、 实验过程中遇到的问题、对问题的思考过程及解决方法（比如 RTL 代码中出现的逻辑 bug，仿真、本地上板及云平台调试过程中的难点等）

1. 在写寄存器堆时，一开始不能理解 0 号寄存器的作用，后来才发现我自己把问题想复杂了；
2. 在寄存器堆中给 r[0][31:0]赋值时可以只写成 r[0]，不必写后面的[31:0]。
3. 在写算式逻辑单元时，需要同时产生三个结果，再由操作码选择其中一个输出。我将三个结果分别命名为 Result1，Result2 和 Result4，且一开始没有写注释说明它们的作用。经过助教老师的提醒，本次加上了一些注释，以后更要避免这样的命名方式。
4. 在考虑 carryout 和 overflow 的问题时，比较容易忽略 A-0x00000000 和 A-0x80000000 两种特殊情况，造成错误。
5. 有时候在云平台上产生的结果会和自己仿真时的波形不一致，这时候可以检查是否真的连接上了云平台，或者会尝试重新生成比特流。

三、 对讲义中思考题（如有）的理解和回答

（思考题：前面讲过，我们需要实现的 ALU 部件应该支持五种基本操作，但是 ALUop 信号位宽为 3 位，因此最多可以支持 2 的 3 次方种操作，即八种操作。那么，请问同学们，多出的三种情况我们该怎么处理呢?在写 Verilog 代码时又应该注意些什么问题呢?）

在输出结果时可以选择对多的三种情况输出 Result=0（或者其他值）。我在实验中选择的是输出 0。

在进行 assign 赋值时，需要将所有合法输入都列入条件，不能将一部分合法输入和非法输入混合处理。

四、 对于此次实验的心得、感受和建议（比如实验是否过于简单或复杂，是否缺少了某些你认为重要的信息或参考资料，对实验项目的建议，对提供帮助的同学的感谢，以及其他想与任课老师交流的内容等）

1. 希望老师对于进位借位和溢出可以放一些例子在 PPT 上面，方便后面写程序的时候参考。

2. 另外感谢同学们在群里指出的脚本里的错误和连接云平台时可能出现的报错。