中国科学院大学计算机组成原理实验课

实验报告

2016K8009929035

王华强

计算机科学与技术

prj-1 处理器中的基本功能部件

1. 电路实现

1.1 关键代码

这里使用两种写法, 第一种是使用reg, 并用vivado的优化功能将其综合为组合逻辑.

另一种写法是直接使用assign语句. 相比之下这种写法更能反映ALU的本质, 但在修改上会很困难.

第一种写法需要将部分output信号设为output reg, 其核心代码如下:

```
module alu(
        input [`DATA_WIDTH - 1:0] A,
        input [`DATA_WIDTH - 1:0] B,
        input [2:0] ALUop,
        output Overflow,
        output CarryOut,
        output Zero,
        output [`DATA_WIDTH - 1:0] Result
);


        reg Overflow;
        reg CarryOut;
        // reg Zero;
        reg [`DATA_WIDTH - 1:0] Result;
        reg [`DATA_WIDTH-1:0] Temp;

        assign Zero=({Result}==0);

always@*//将会被综合成组合逻辑
begin
        case(ALUop[2:0])
        `AND:
                begin
                        {CarryOut,Result}=A&B;
                        Overflow=`UNDEFINED;
                        Temp=`UNDEFINED;
                end
        `OR:
                begin
                        {CarryOut,Result}=A|B;
                        Overflow=`UNDEFINED;
                        Temp=`UNDEFINED;
                end
        `ADD:
                begin
                        {CarryOut,Result}=A+B;
                        Overflow=(A[`DATA_WIDTH - 1]==B[`DATA_WIDTH - 1])
```

```
                        &&(Result[`DATA_WIDTH - 1]!=A[`DATA_WIDTH - 1]);
                        Temp=`UNDEFINED;

                end
        `SUB:
                begin
                        {CarryOut,Result}=A+~B+1;//补码减法, A-B=A+(-B), -B(补码)=(全部取反+1)
                        Overflow=(A[`DATA_WIDTH - 1]!=B[`DATA_WIDTH - 1])
                        &&(Result[`DATA_WIDTH - 1]!=A[`DATA_WIDTH - 1]);
                        Temp=`UNDEFINED;

                        //todo Overflow
                end
        `SLT:
                begin
                        {CarryOut,Temp}=A+~B+1;//补码减法, A-B=A+(-B), -B(补码)=B(部取反+1)
                        Overflow=(A[`DATA_WIDTH - 1]!=B[`DATA_WIDTH - 1])
                        &&(Temp[`DATA_WIDTH - 1]!=A[`DATA_WIDTH - 1]);
                        Result[`DATA_WIDTH-1:1]=0;
                        Result[0]=Temp[`DATA_WIDTH-1]^Overflow;
                end
        default:
                begin
                    {CarryOut,Temp}=`UNDEFINED;
                        Overflow=`UNDEFINED;
                        Temp=`UNDEFINED;
                end
        endcase
end
```

1.2 逻辑电路结构

两种逻辑电路图分别如下.

1.3 仿真波形的截图及说明

testbench的核心代码如下:

```
task test;
input [`DATA_WIDTH-1:0] a;
input [`DATA_WIDTH-1:0] b;
input [2:0] op;
begin
        A=a;
        B=b;
        ALUop=op;
        $display("A:%d B:%d ALUop:%d Overflow:%d
                 CarryOut:%d Zero:%d Result:%d",
                 A[`DATA_WIDTH-1:0],B[`DATA_WIDTH-1:0],
                 ALUop, Overflow, CarryOut, Zero,
                 Result[`DATA_WIDTH-1:0]);
        #1;
end
endtask
```

这是测试任务, 通过调用类似于`test(a,b,aluop);`的语句

```
        initial
```

```
        begin
                test(1,1,`ADD);
                test(88,5,`ADD);
                test(1,3,`ADD);
                test(1555,111,`SUB);
                test(111,111,`SUB);
                test(1555,11111,`SUB);
                test(1555,11111,`AND);
                test(1555,11111,`OR);
                test(1555,11111,`SLT);
                test(11111,33,`SLT);
                test(32'hffffffff,1,`SLT);
                test(32'hffffffff,2,`SLT);
                test(1,2,`SUB);
        end
```

testbench输出结果如下:

```
G:\workpath\Notes\ComputerComposition\project\prj1>vvp ".\alu_test.v.vvp"
A:          1 B:          1 ALUop:2 Overflow:x CarryOut:x Zero:x Result:          x
A:         88 B:          5 ALUop:2 Overflow:0 CarryOut:0 Zero:0 Result:          2
A:          1 B:          3 ALUop:2 Overflow:0 CarryOut:0 Zero:0 Result:         93
A:       1555 B:        111 ALUop:6 Overflow:0 CarryOut:0 Zero:0 Result:          4
A:        111 B:        111 ALUop:6 Overflow:0 CarryOut:0 Zero:0 Result:       1444
A:       1555 B:      11111 ALUop:6 Overflow:0 CarryOut:0 Zero:1 Result:          0
A:       1555 B:      11111 ALUop:0 Overflow:0 CarryOut:1 Zero:0 Result:4294957740
A:       1555 B:      11111 ALUop:1 Overflow:0 CarryOut:0 Zero:0 Result:        515
A:       1555 B:      11111 ALUop:7 Overflow:0 CarryOut:0 Zero:0 Result:      12151
A:      11111 B:         33 ALUop:7 Overflow:0 CarryOut:1 Zero:0 Result:          1
A:4294967295 B:          1 ALUop:7 Overflow:0 CarryOut:0 Zero:1 Result:          0
A:4294967295 B:          2 ALUop:7 Overflow:0 CarryOut:0 Zero:0 Result:          1
A:          1 B:          2 ALUop:6 Overflow:0 CarryOut:0 Zero:0 Result:          1
```

仿真波形如下:

2. 问题与思考

3. 思考题

ALUop是三位的, 对于这8种可能, 没有定义的三种可能应该作出何种处理?

逻辑上:

对于没有必要的指令给出一组任意的值, 这里设置为0;

代码上:

设置default分支, 防止生成意料之外的硬件结构(如锁存器);

在第一种写法中, 很有可能

4. 实验心得

5. 致谢

500块的新内存条真吉尔好用(笑).

陈欲晓同学提供了ALU写法的第二种思路, 在此致谢.