

# 计算机组成原理 实验报告

姓名：魏钊 学号：PB18111699 实验日期：2020-4-25

## 一、实验题目：

实验一 运算器与排序

## 二、实验目的：

1.掌握算术逻辑单元（ALU）的功能，加/减运算时溢出、进位/借位、零标志的形成及其应用；

2.掌握数据通路和控制器的设计和描述方法。

## 三、实验平台：

Vivado

## 四、实验过程：

### ALU 部分：

module ALU

  #(parameter WIDTH = 32)

  (

    output reg [WIDTH-1:0] y,           //运算结果

    output reg zf,                    //零标志

    output reg cf,                    //进位/借位标志

    output reg of,                    //溢出标志

    output reg sf,                    //最高位是否为 1

    input [WIDTH-1:0] a, b,           //两操作数

    input [2:0] m                    //操作类型

  );

首先声明相关参数与输入输出。

然后实现相关组合逻辑。

加法部分：

  always@(\*)

  case(m)

    3'b000://加

    begin

      {cf,y}=a+b;

      of=(~a[WIDTH-1]&~b[WIDTH-1]&y[WIDTH-1])|(a[WIDTH-1]&b[WIDTH-1]&~y[WIDTH-1]);//溢出标志

      sf=y[WIDTH-1];//符号标志

      zf=~|y;//零标志

    end

减法部分：

```
3'b001://减
begin
    {cf,y}=a-b;
    of=(~a[WIDTH-1]&b[WIDTH-1]&y[WIDTH-1])|(a[WIDTH-1]&~b[WIDTH-1]&~y[WIDTH-1]);//溢出标志
    sf=y[WIDTH-1];//符号标志
    zf=~|y;//零标志
end
```

特别要注意加减法溢出标志的判别。

与、或、异或：

```
3'b010://与
begin
    cf=1'bx;
    of=1'bx;
    y=a&b;
    sf=y[WIDTH-1];//符号标志
    zf=~|y;//零标志
end
```

```
3'b011://或
begin
    cf=1'bx;
    of=1'bx;
    y=a|b;
    sf=y[WIDTH-1];//符号标志
    zf=~|y;//零标志
end
```

```
3'b100://异或
begin
    cf=1'bx;
    of=1'bx;
    y=a^b;
    sf=y[WIDTH-1];//符号标志
    zf=~|y;//零标志
end
```

其他状况：

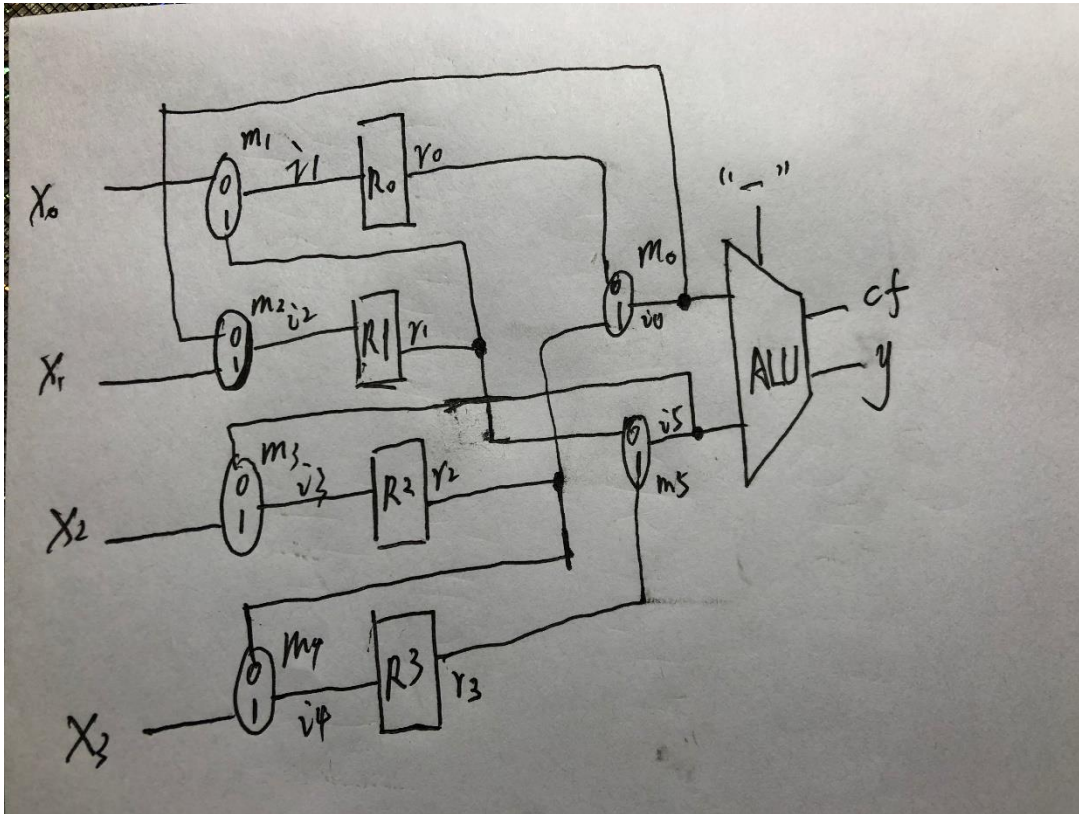
```
default:
begin
    cf=1'bx;
```

```

of=1'bx;
y=32'bx;
sf=1'bx;
zf=1'bx;
end

```

**SORT 部分:**



**MUX 模块:**

```

module MUX
#(parameter WIDTH = 32)
(
    input m,
    input [WIDTH-1:0] a,b,
    output reg [WIDTH-1:0] y
);
always @(*)
    if(m)
        y=b;
    else
        y=a;
endmodule

```

根据控制信号 m,输出 a 或者 b。

**Register 模块:**

```

module register
#(parameter WIDTH = 32)

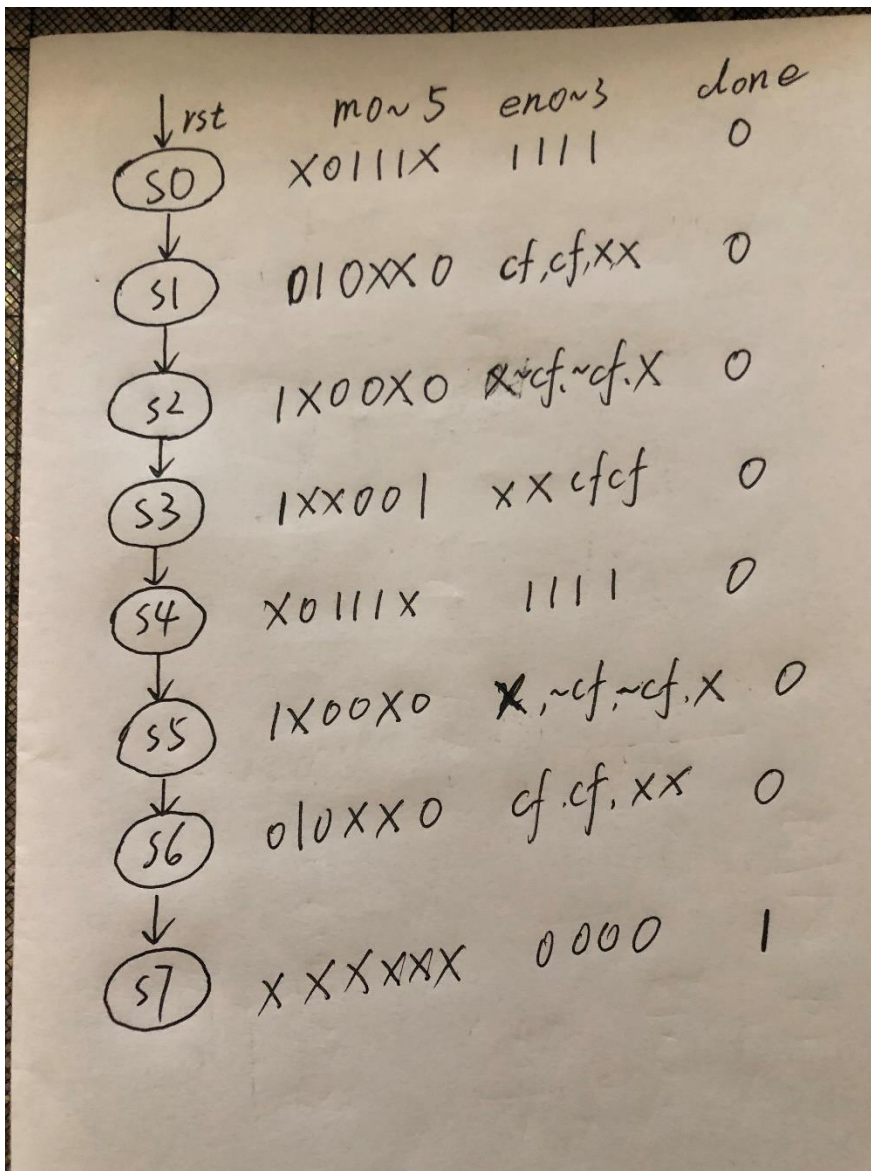
```

```

(
  input [WIDTH-1:0] a,
  output reg [WIDTH-1:0] y,
  input clk,
  input en,
  input rst
);
always @(posedge clk or posedge rst)
  if(rst)
    y=0;
  else
    if(en)
      y=a;
endmodule

```

当复位信号 `rst` 有效时寄存器内部置为 0，否则根据使能信号 `en` 判断是否改变寄存器内部数值。



SORT 模块，先定义输入输出及参数：

```

module SORT
#(
    parameter N = 32,
    parameter S0=3'b000,//载入
    parameter S1=3'b001,//x0 与 x1 比较
    parameter S2=3'b010,//x1 与 x2 比较
    parameter S3=3'b011,//x2 与 x3 比较
    parameter S4=3'b100,//x0 与 x1 比较
    parameter S5=3'b101,//x1 与 x2 比较
    parameter S6=3'b110,//x0 与 x1 比较
    parameter S7=3'b111//排序完成
)
(
    output reg [N-1:0] s0, s1, s2, s3, //排序后的四个数据（递增）
    output reg done, //排序结束标志
    input [N-1:0] x0, x1, x2, x3,//原始输入数据
    input clk, rst //时钟（上升沿有效）、复位（高电平有效）
);

```

然后定义内部线路:

```

reg [2:0] state,next_state;
reg m0,m1,m2,m3,m4,m5;
wire [N-1:0] i0,i1,i2,i3,i4,i5,r0,r1,r2,r3;
reg en0,en1,en2,en3;
wire cf;

```

然后声明相关 MUX,Register 及 ALU:

MUX

```

M0(.m(m0),.y(i0),.a(r0),.b(r2)),
M1(.m(m1),.y(i1),.a(x0),.b(r1)),
M2(.m(m2),.y(i2),.a(i0),.b(x1)),
M3(.m(m3),.y(i3),.a(i5),.b(x2)),
M4(.m(m4),.y(i4),.a(r2),.b(x3)),
M5(.m(m5),.y(i5),.a(r1),.b(r3));//选择器部分

```

register

```

R0(.clk(clk),.rst(rst),.en(en0),.a(i1),.y(r0)),
R1(.clk(clk),.rst(rst),.en(en1),.a(i2),.y(r1)),
R2(.clk(clk),.rst(rst),.en(en2),.a(i3),.y(r2)),
R3(.clk(clk),.rst(rst),.en(en3),.a(i4),.y(r3));//寄存器部分

```

ALU alu(.m(3'b001),.cf(cf),.a(i0),.b(i5));//ALU 部分

两段式及控制单元:

```

//两段式
always @(posedge clk or posedge rst)
    if(rst)
        state<=S0;
    else
        state<=next_state;

always @(*)
begin
    case (state)
        S0: next_state=S1;
        S1: next_state=S2;
        S2: next_state=S3;
        S3: next_state=S4;
        S4: next_state=S5;
        S5: next_state=S6;
        S6: next_state=S7;
        default:
            next_state=S7;
    endcase
end
//控制单元
always @(*)
begin
    {m0,m1,m2,m3,m4,m5,en0,en1,en2,en3,done,s0,s1,s2,s3}=15'b0;
    case (state)
        S0: {m1,m2,m3,m4,en0,en1,en2,en3}=8'b0111_1111;//载入数据
            S1,S4,S6://寄存器 0 和寄存器 1 中的数据比较，及判断是否交换
            begin
                m0=0;
                m1=1;
                m2=0;
                m5=0;
                en0=cf;
                en1=cf;
            end
        S2,S5://寄存器 1 和寄存器 2 中的数据比较，及判断是否交换
            begin
                m0=1;
                m2=0;
                m3=0;
                m5=0;
                en1=~cf;
                en2=~cf;
            end
    endcase
end

```

```

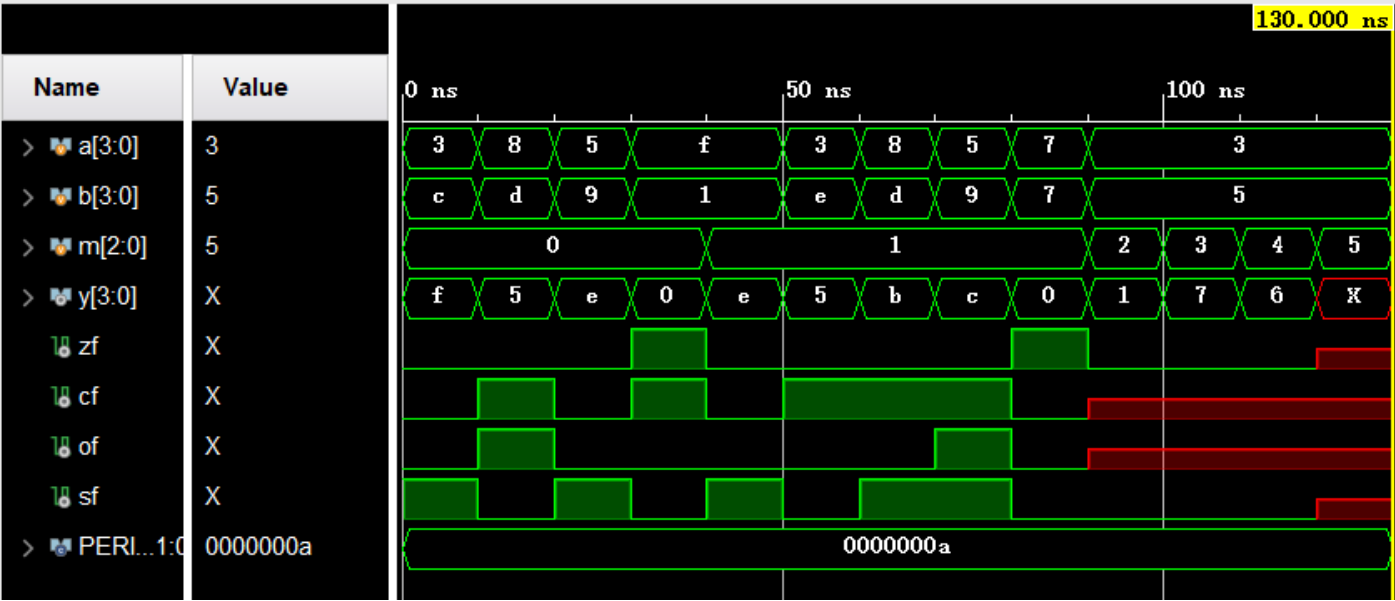
end
S3://寄存器 2 和寄存器 3 中的数据比较，及判断是否交换
begin
    m0=1;
    m3=0;
    m4=0;
    m5=1;
    en2=cf;
    en3=cf;
end
S7: //排序完成输出
begin
    done=1;
    en0=0;
    en1=0;
    en2=0;
    en3=0;
    s0=r3;
    s1=r2;
    s2=r1;
    s3=r0;
end
endcase
end

```

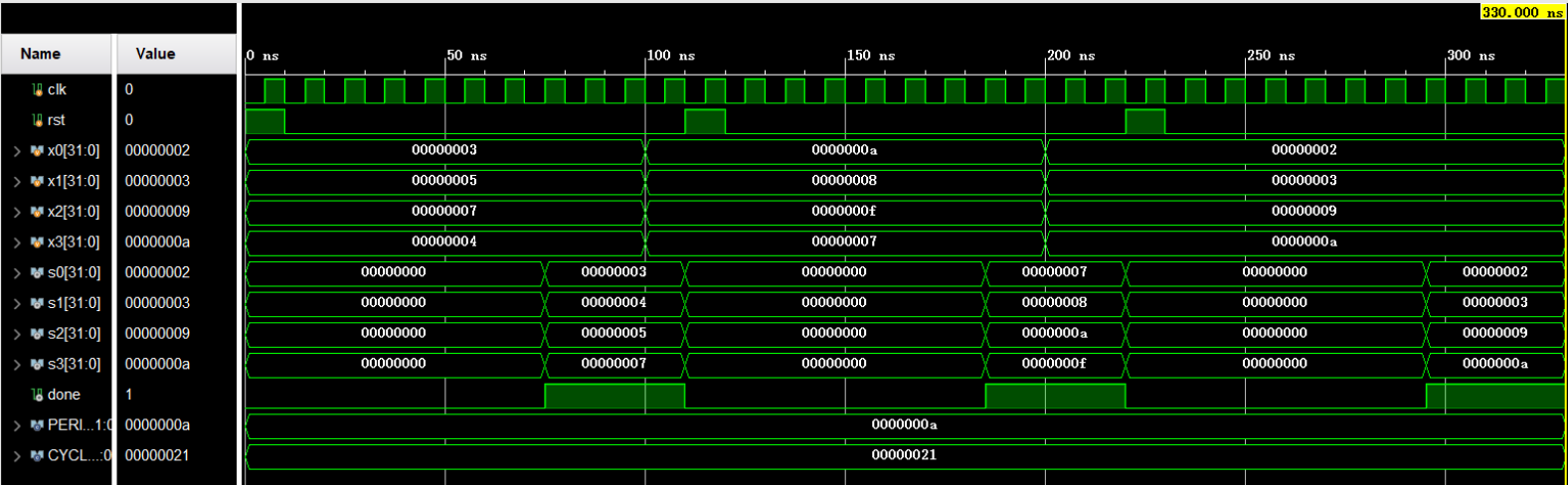
相关控制信号根据之前数据通路，状态图进行设置。

五、实验结果：

ALU 仿真：



SORT 仿真：



六、心得体会：

通过本次实验了解了 ALU 内部通路，掌握了其功能，了解了相关标志位的形成和应用。

复习掌握了结构化方式描述数据通路，控制器及有限状态机的设计。