

中国科学技术大学计算机学院  
《数字电路实验》报告



实验题目：综合实验-多功能计算器

学生姓名：陈鸿绪

学生学号：PB21000224

完成日期：12.20

计算机实验教学中心制

2020 年 09 月

## 【实验题目】

综合实验-多功能计算器

## 【实验目的】

综合使用 Vivado 实验软件、FPGA 实验平台等进行一定规模的程序设计和芯片烧写，最后在芯片上实现多功能计算器的功能。

## 【实验环境】

vlab.ustc.edu.cn

Vivado 实验软件

FPGA 实验平台

## 【功能详细介绍】

该多功能计算器可以通过置芯片上不同的 Button，选择不同的计算方式执行加法、减法、乘法、除法、cos、sin 的计算还有素数判断，由于芯片上数字显示位数有限，所以该芯片上只支持两位正整数与两位正整数的计算，对 cos 与 sin 只支持角度制，且角度值只能是四位正整数。所有的计算结果都四位表示，如果涉及正负则在七段数码管上进行显示。

## 【功能具体实现思路】

1. 选择计算方式功能：在代码中根据不同的 sw 按钮状态用不同的 case 进行情况划分，000 表示加法，001 表示减法，010 表示乘法，011 表示除法，100 表示 sin 计算，101 表示 cos 计算，110 表示素数判断，其余状态均为加法。
2. 加减乘除计算实现：在 Verilog 中直接进行相应的计算后显示结果

3. sin 与 cos 的计算：将  $0^{\circ}$  到  $89^{\circ}$  所有整数的 sin 值全部提前记录在 Verilog 代码中，对于计算 sin 值与 cos 值，只需要通过诱导公式进行处理最后落入  $0^{\circ}$  到  $89^{\circ}$  范围内即可。
4. 素数判断：将 1—999 之间所有的素数提前以列表的形式存入代码中，判断时只需要通过遍历查找即可，查找到显示 1，否则显示 0。
5. 运算显示：这里采用分时显示，即让所有数码管进行一定频率的轮换显示从而达到视觉上同时显示的效果。

### 【实验代码展示】

以下为 Verilog 设计代码

```
module test(
input button, clk,
input [7:0] sw,
output reg [3:0] data, //这个时刻需要显示的数据
output reg [2:0] sel, //选择哪一个数码管显示
output reg sign, //显示结果的正负符号值
output reg dot
);
reg button_r1, button_r2;
always@(posedge clk) button_r1 <= button;
always@(posedge clk) button_r2 <= button_r1;
assign button_edge = button_r1 & (~button_r2); //取 button 信号边缘
reg sw1_1, sw1_2;
always@(posedge clk) sw1_1 <= sw[3];
always@(posedge clk) sw1_2 <= sw1_1;
assign sw1 = sw1_1 & (~sw1_2);
reg sw2_1, sw2_2;
always@(posedge clk) sw2_1 <= sw[4];
always@(posedge clk) sw2_2 <= sw2_1;
assign sw2 = sw2_1 & (~sw2_2);
reg sw3_1, sw3_2;
always@(posedge clk) sw3_1 <= sw[6];
always@(posedge clk) sw3_2 <= sw3_1;
assign sw3 = sw3_1 & (~sw3_2);
reg sw4_1, sw4_2;
always@(posedge clk) sw4_1 <= sw[7];
```

```

always@(posedge clk) sw4_2 <= sw4_1;
assign sw4 = sw4_1 & (~sw4_2);
reg [15:0] result=0;
reg [6:0] a=0;
reg [6:0] b=0;
reg [15:0] list[89:0];//存储 sin 值
reg [15:0] list1[168:0];//存储素数值
initial
begin
list[0]=0;
list[1]=175;
list[2]=349;
list[3]=523;
list[4]=698;
list[5]=872;
list[6]=1045;
list[7]=1219;
list[8]=1392;
list[9]=1564;
list[10]=1736;
list[11]=1908;
list[12]=2079;
list[13]=2250;
list[14]=2419;
list[15]=2588;
list[16]=2756;
list[17]=2924;
list[18]=3090;
list[19]=3256;
list[20]=3420;
list[21]=3584;
list[22]=3746;
list[23]=3907;
list[24]=4067;
list[25]=4226;
list[26]=4384;
list[27]=4540;
list[28]=4695;
list[29]=4848;
list[30]=5000;
list[31]=5150;
list[32]=5299;
list[33]=5446;
list[34]=5592;

```

```
list[35]=5736;  
list[36]=5878;  
list[37]=6018;  
list[38]=6157;  
list[39]=6293;  
list[40]=6428;  
list[41]=6561;  
list[42]=6691;  
list[43]=6820;  
list[44]=6947;  
list[45]=7071;  
list[46]=7193;  
list[47]=7314;  
list[48]=7431;  
list[49]=7547;  
list[50]=7660;  
list[51]=7771;  
list[52]=7880;  
list[53]=7986;  
list[54]=8090;  
list[55]=8192;  
list[56]=8290;  
list[57]=8387;  
list[58]=8480;  
list[59]=8572;  
list[60]=8660;  
list[61]=8746;  
list[62]=8829;  
list[63]=8910;  
list[64]=8988;  
list[65]=9063;  
list[66]=9135;  
list[67]=9205;  
list[68]=9272;  
list[69]=9336;  
list[70]=9397;  
list[71]=9455;  
list[72]=9511;  
list[73]=9563;  
list[74]=9613;  
list[75]=9659;  
list[76]=9703;  
list[77]=9744;  
list[78]=9781;
```

```
list[79]=9816;
list[80]=9848;
list[81]=9877;
list[82]=9903;
list[83]=9925;
list[84]=9945;
list[85]=9962;
list[86]=9976;
list[87]=9986;
list[88]=9994;
list[89]=9998;
list[90]=0;
list1[1]=2;
list1[2]=3;
list1[3]=5;
list1[4]=7;
list1[5]=11;
list1[6]=13;
list1[7]=17;
list1[8]=19;
list1[9]=23;
list1[10]=29;
list1[11]=31;
list1[12]=37;
list1[13]=41;
list1[14]=43;
list1[15]=47;
list1[16]=53;
list1[17]=59;
list1[18]=61;
list1[19]=67;
list1[20]=71;
list1[21]=73;
list1[22]=79;
list1[23]=83;
list1[24]=89;
list1[25]=97;
list1[26]=101;
list1[27]=103;
list1[28]=107;
list1[29]=109;
list1[30]=113;
list1[31]=127;
list1[32]=131;
```

```
list1[33]=137;  
list1[34]=139;  
list1[35]=149;  
list1[36]=151;  
list1[37]=157;  
list1[38]=163;  
list1[39]=167;  
list1[40]=173;  
list1[41]=179;  
list1[42]=181;  
list1[43]=191;  
list1[44]=193;  
list1[45]=197;  
list1[46]=199;  
list1[47]=211;  
list1[48]=223;  
list1[49]=227;  
list1[50]=229;  
list1[51]=233;  
list1[52]=239;  
list1[53]=241;  
list1[54]=251;  
list1[55]=257;  
list1[56]=263;  
list1[57]=269;  
list1[58]=271;  
list1[59]=277;  
list1[60]=281;  
list1[61]=283;  
list1[62]=293;  
list1[63]=307;  
list1[64]=311;  
list1[65]=313;  
list1[66]=317;  
list1[67]=331;  
list1[68]=337;  
list1[69]=347;  
list1[70]=349;  
list1[71]=353;  
list1[72]=359;  
list1[73]=367;  
list1[74]=373;  
list1[75]=379;  
list1[76]=383;
```

```
list1[77]=389;  
list1[78]=397;  
list1[79]=401;  
list1[80]=409;  
list1[81]=419;  
list1[82]=421;  
list1[83]=431;  
list1[84]=433;  
list1[85]=439;  
list1[86]=443;  
list1[87]=449;  
list1[88]=457;  
list1[89]=461;  
list1[90]=463;  
list1[91]=467;  
list1[92]=479;  
list1[93]=487;  
list1[94]=491;  
list1[95]=499;  
list1[96]=503;  
list1[97]=509;  
list1[98]=521;  
list1[99]=523;  
list1[100]=541;  
list1[101]=547;  
list1[102]=557;  
list1[103]=563;  
list1[104]=569;  
list1[105]=571;  
list1[106]=577;  
list1[107]=587;  
list1[108]=593;  
list1[109]=599;  
list1[110]=601;  
list1[111]=607;  
list1[112]=613;  
list1[113]=617;  
list1[114]=619;  
list1[115]=631;  
list1[116]=641;  
list1[117]=643;  
list1[118]=647;  
list1[119]=653;  
list1[120]=659;
```



```
list1[121]=661;
list1[122]=673;
list1[123]=677;
list1[124]=683;
list1[125]=691;
list1[126]=701;
list1[127]=709;
list1[128]=719;
list1[129]=727;
list1[130]=733;
list1[131]=739;
list1[132]=743;
list1[133]=751;
list1[134]=757;
list1[135]=761;
list1[136]=769;
list1[137]=773;
list1[138]=787;
list1[139]=797;
list1[140]=809;
list1[141]=811;
list1[142]=821;
list1[143]=823;
list1[144]=827;
list1[145]=829;
list1[146]=839;
list1[147]=853;
list1[148]=857;
list1[149]=859;
list1[150]=863;
list1[151]=877;
list1[152]=881;
list1[153]=883;
list1[154]=887;
list1[155]=907;
list1[156]=911;
list1[157]=919;
list1[158]=929;
list1[159]=937;
list1[160]=941;
list1[161]=947;
list1[162]=953;
list1[163]=967;
list1[164]=971;
```

```

list1[165]=977;
list1[166]=983;
list1[167]=991;
list1[168]=997;
end
wire [3:0] result_1,result_2,result_3,result_4;
wire [3:0] a_1,a_2;
wire [3:0] b_1,b_2;
assign result_1=((sw[1:0]==2'b11)&&(a==0))?15:(result%10); //需要显示结果的个位
assign result_2=((sw[1:0]==2'b11)&&(a==0))?15:(result/10)%10; //结果的十位
assign result_3=((sw[1:0]==2'b11)&&(a==0))?15:(result/100)%10; //结果的百位
assign result_4=((sw[1:0]==2'b11)&&(a==0))?15:(result/1000); //结果的千位
assign a_1=a%10;
assign a_2=(a/10);
assign b_1=b%10;
assign b_2=(b/10);
reg [15:0] theta;
reg [15:0] p;
reg flag=0;
reg [15:0] t;
integer i=0;
always@(*)
begin
    case(sw[2:0])
        3'b000: result=a+b; //加法计算
        3'b0001: begin //减法计算
            if(a>b) result=a-b;
            else result=b-a;
        end
        3'b010: result=a*b; //乘法计算
        3'b011: result=b/a; //除法计算
        3'b100: begin //sin 计算
            theta=b*100+a;
            theta=theta-360*(theta/360);
            if(theta>180) flag=1;
            else flag=0;
            theta=theta-(theta/180)*180;
            if(theta>90) result=list[180-theta];
            else result=list[theta];
        end
        3'b101: begin //cos 计算
            theta=b*100+a;
            theta=theta-360*(theta/360);
            if(theta>90 && theta<270) flag=1;

```

```

        else flag=0;
        theta=theta-(theta/180)*180;
        if(theta>90)
            begin
                theta=theta-90;
                result=list[theta];
            end
        else result=list[90-theta];
    end
    3'b110: begin//遍历判断素数
        p=100*b+a;
        t=0;
        for(i=1;i<169;i=i+1)
            begin
                if(p==list1[i]) t=1;
            end
        result=t;
    end
    default: result=a+b;
endcase
end
always@(posedge clk)
begin
    if(sw1)
        begin
            a<=(a/10)*10+((a+1)%10);
        end
    if(sw3)
        begin
            b<=(b/10)*10+((b+1)%10);
        end
    if(sw2)
        begin
            a<=((a/10+1)%10)*10+a%10;
        end
    if(sw4)
        begin
            b<=((b/10+1)%10)*10+b%10;
        end
    if(button_edge)
        begin
            a=0;
            b=0;
        end
end

```

```

        end
reg [4:0] temp;//构造的计时器
always@(posedge clk)
    begin
        temp=temp+1;
    end
always@(posedge clk)//分时显示
    begin
        sel=temp[4:2];
        case(sel)
            3'b000: data<=result_1;
            3'b001: data<=result_2;
            3'b010: data<=result_3;
            3'b011: data<=result_4;
            3'b100: data<=a_1;
            3'b101: data<=a_2;
            3'b110: data<=b_1;
            3'b111: data<=b_2;
        endcase
        if(sw[2:0]==3'b100 || sw[2:0]==3'b101)
            begin
                sign=flag;
                dot=1;
            end
    end
end
endmodule

```

以下为 XDC 文件:

```

1 set_property -dict { PACKAGE_PIN E3 IOSTANDARD LVCMOS33 } [get_ports { clk }];
2 create_clock -add -name sys_clk_pin -period 10.00 -waveform {0,5} [get_ports{clk}];
3 set_property -dict { PACKAGE_PIN B18 IOSTANDARD LVCMOS33 } [get_ports { button }];
4 set_property -dict { PACKAGE_PIN G18 IOSTANDARD LVCMOS33 } [get_ports { dot }];
5 set_property -dict { PACKAGE_PIN F18 IOSTANDARD LVCMOS33 } [get_ports { sign }];
6
7 set_property -dict { PACKAGE_PIN D14 IOSTANDARD LVCMOS33 } [get_ports { sw[0] }];
8 set_property -dict { PACKAGE_PIN F16 IOSTANDARD LVCMOS33 } [get_ports { sw[1] }];
9 set_property -dict { PACKAGE_PIN G16 IOSTANDARD LVCMOS33 } [get_ports { sw[2] }];
10 set_property -dict { PACKAGE_PIN H14 IOSTANDARD LVCMOS33 } [get_ports { sw[3] }];
11 set_property -dict { PACKAGE_PIN E16 IOSTANDARD LVCMOS33 } [get_ports { sw[4] }];
12 set_property -dict { PACKAGE_PIN F13 IOSTANDARD LVCMOS33 } [get_ports { sw[5] }];
13 set_property -dict { PACKAGE_PIN G13 IOSTANDARD LVCMOS33 } [get_ports { sw[6] }];
14 set_property -dict { PACKAGE_PIN H16 IOSTANDARD LVCMOS33 } [get_ports { sw[7] }];
15
16 set_property -dict { PACKAGE_PIN A14 IOSTANDARD LVCMOS33 } [get_ports { data[0] }];
17 set_property -dict { PACKAGE_PIN A13 IOSTANDARD LVCMOS33 } [get_ports { data[1] }];
18 set_property -dict { PACKAGE_PIN A16 IOSTANDARD LVCMOS33 } [get_ports { data[2] }];
19 set_property -dict { PACKAGE_PIN A15 IOSTANDARD LVCMOS33 } [get_ports { data[3] }];
20
21 set_property -dict { PACKAGE_PIN A18 IOSTANDARD LVCMOS33 } [get_ports { sel[2] }];
22 set_property -dict { PACKAGE_PIN B16 IOSTANDARD LVCMOS33 } [get_ports { sel[1] }];
23 set_property -dict { PACKAGE_PIN B17 IOSTANDARD LVCMOS33 } [get_ports { sel[0] }];

```

## 【总结与思考】

我学会了综合利用各种工具进行芯片功能设计。