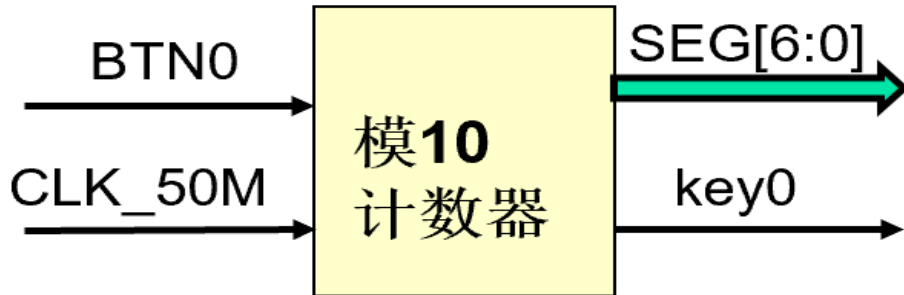


实验三 按键消抖电路设计与应用

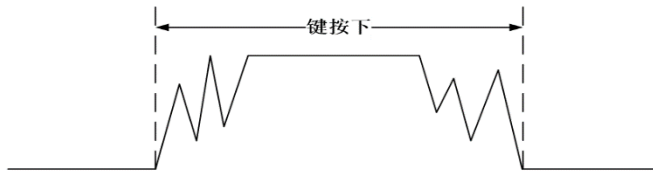
一、实验内容

设计一个对按键 BTN0 进行模 10 的计数器，输出用一个数码管显示，完成该电路的硬件测试。

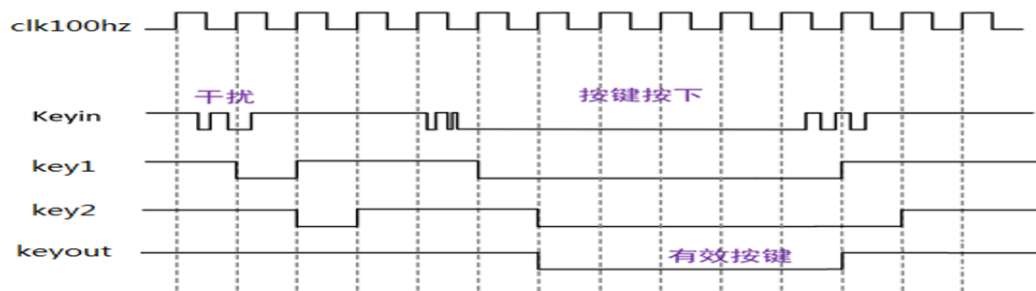


二、实验原理与方案

3.1 按键消抖



当按键按下时，会产生抖动，我们需要消除按键抖动产生的毛刺，产生方波信号，避免按键信号重复触发，使计数值跳变，可以使用两级寄存器消除按键的不稳定状态，key1 为寄存器 1 输出，key2 为寄存器 2 输出。



3.2 数码管显示

使用开发板上的四个共阳极数码管，本次实验只涉及静态显示，所以低电平位选第一个数码管有效。段选信号经过译码送到段选端，即可完成数码管的静态显示。

三、实验过程(源程序)

```
module work_2(
    clk_50mhz,
    btn0,
    seg,
    key0
);
    input clk_50mhz;
    input btn0;
    output reg [6:0] seg;
    output [3:0] key0;
    assign key0=4'b0111;

    integer cnt2=1;
    reg clk100hz;

    reg[3:0] cnt=0;
    //100hz
    always@(posedge clk_50mhz)
    begin
        if(cnt2==250000)//25000000
        begin
            cnt2=1;
            clk100hz=~clk100hz;
        end
        else
            cnt2=cnt2+1;
    end

    wire key_out;
    reg tmp1,tmp2;
    always @(posedge clk100hz) begin
        tmp1 <= btn0;
        tmp2 <= tmp1;
```

```

end
assign key_out = tmp1 | tmp2;

//moshi
always@(posedge key_out)
begin
    if(cnt==9)
        cnt=0;
    else
        cnt=cnt+1;
end

always@(cnt)
begin
    case(cnt)
        0: seg<=7'b0000001;
        1: seg<=7'b1001111;
        2: seg<=7'b0010010;
        3: seg<=7'b0000110;
        4: seg<=7'b1001100;
        5: seg<=7'b0100100;
        6: seg<=7'b1100000;
        7: seg<=7'b0001111;
        8: seg<=7'b0000000;
        9: seg<=7'b0001100;
        default: seg<=7'b1111111;
    endcase
end

endmodule

```

四、实验结果与分析(仿真程序与仿真图，硬件测试图)

4.1 仿真程序

```

module work2_testbench;

// Inputs
reg clk_50mhz;
reg btn0;

```

```

// Outputs
wire [6:0] seg;
wire [3:0] key0;

// Instantiate the Unit Under Test (UUT)
work_2 uut (
    .clk_50mhz(clk_50mhz),
    .btn0(btn0),
    .seg(seg),
    .key0(key0)
);

initial begin
    // Initialize Inputs
    clk_50mhz = 0;
    btn0 = 0;
    // Wait 100 ns for global reset to finish

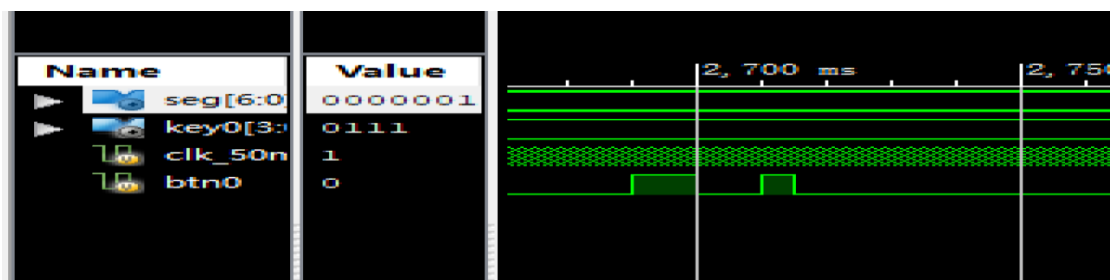
    // Add stimulus here

end
always#10 clk_50mhz=~clk_50mhz;

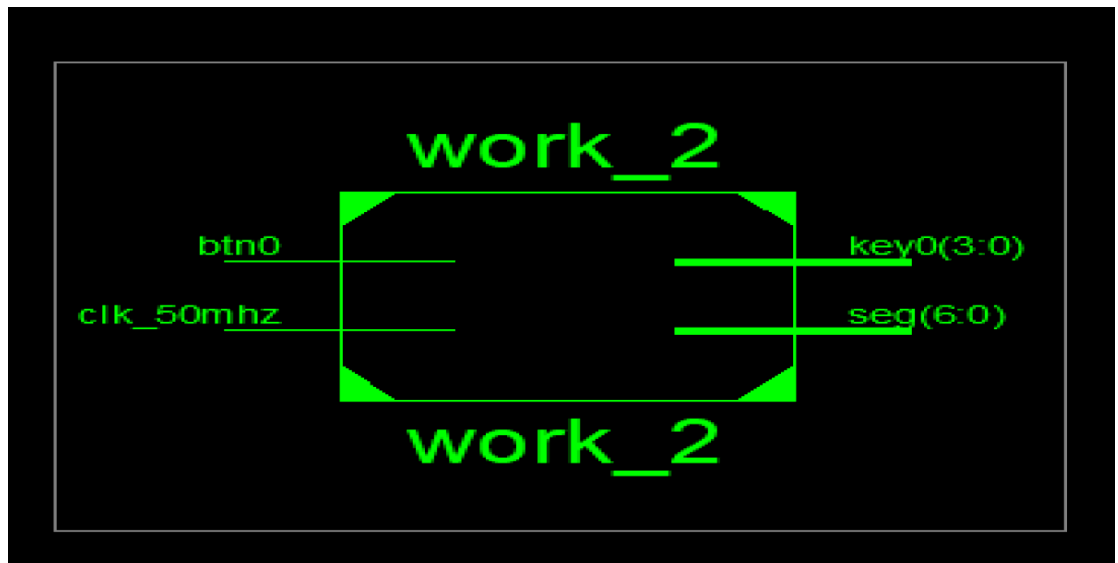
always
begin
    #10000000
    btn0=1;
    #5000000
    btn0=0;
    #50000000;
    btn0=1;
    #10000000
    btn0=0;
end
endmodule

```

4.2 仿真图



4. 3RTL



4.4 硬件测试图

