



奋斗的小孩之 altera 系列

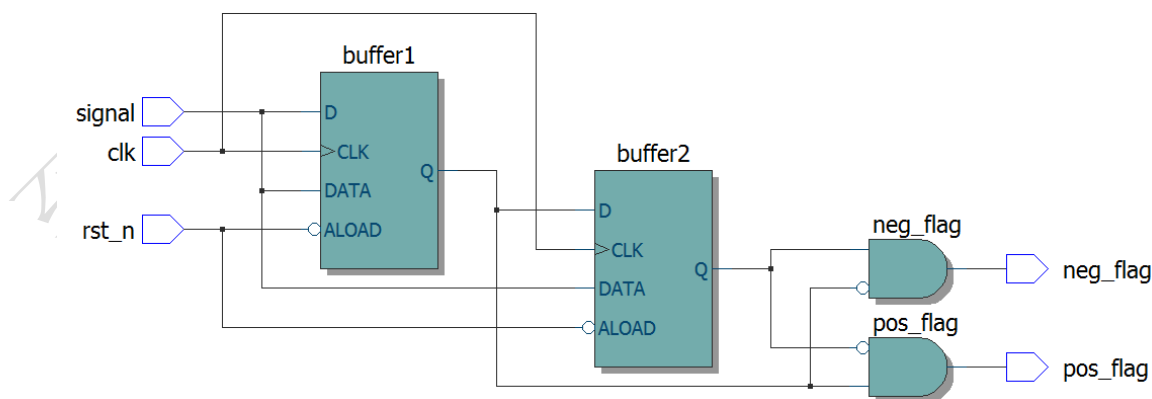
第二十一篇 边沿检测

对于每一个的小实验，我们都可以把它看作是一个小项目，逐步的去分析，设计，调试，最后完成功能。下面我们就开始我们的“小项目”。

项目名称：边沿检测

具体要求：检测输入信号，或者 FPGA 内部逻辑信号的跳变，即上升沿或者下降沿的检测，当检测到边沿后，发出高脉冲。

通过分析上述的“项目名称”和“具体要求”，我们可以设计出如下电路：

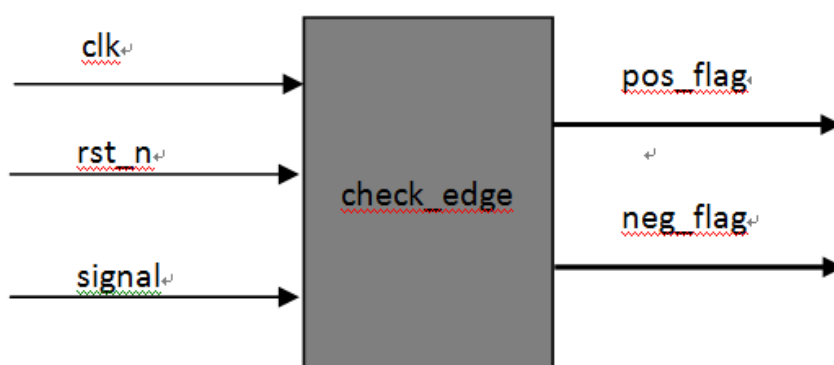


由于寄存器可以使信号延时一个时钟周期，我们可以利用这点，



来完成信号的边沿检测。假设 signal 一直为高电平，则 buffer1 和 buffer2 的输出都是高电平，当 signal 变成低电平时，buffer1 的输出先变成低电平，我们将 buffer1 的输出值取反（高电平）和 buffer2 的输出（高电平）相与得到 neg_flag（高电平），下一个周期时，buffer2 的输出也变成了低电平，buffer1 的输出值取反（高电平）和 buffer2 的输出（低电平）相与得到低电平。故而 neg_flag 只是输出了一个时钟周期的高脉冲。pos_flag 可以同理分析。

架构图如下：



signal: 被检测信号

pos_flag: 检测出上升沿后发出的高脉冲

neg_flag: 检测出下降沿后发出的高脉冲

系统设计：

1. 工程的名称：ckeck_edge。

设计代码如下：



/*

模块名称: ckeck_edge

模块功能: 检测上升沿或者下降沿, 检测到边沿后, 发出高脉冲。

编写时间: 2016-08-16

作者: 至芯科技---奋斗的小孩

邮箱: zxopenhxs@126.com

*/

```
module check_edge (clk, rst_n, signal, neg_flag, pos_flag);
```

```
    input clk;
```

```
    input rst_n;
```

```
    input signal;
```

```
    output neg_flag;
```

```
    output pos_flag;
```

```
    reg buffer1;
```

```
    reg buffer2;
```

```
    always @ (posedge clk or negedge rst_n)
```

```
    begin
```



```
        if (!rst_n)

            begin

                buffer1 <= signal;

                buffer2 <= signal;

            end

        else

            begin

                buffer1 <= signal;

                buffer2 <= buffer1;

            end

        end

    end

    assign neg_flag = buffer2&&(~buffer1);

    assign pos_flag = (~buffer2)&&buffer1;

endmodule
```

解析:

在模块复位时，笔者将 buffer1 和 buffer2 的输出设置成和输入信号相同的电平值，原因如下：

如果设置成高电平，而输入信号在复位器件为低电平，那么在复位结束后就会产生一个我们不想要的 neg_flag（分析方法同设计方



法)。另一种情况读者可以自己分析，方法是相同的。

激励代码如下：

```
/*
```

模块名称：check_edge_tb

模块功能：为 check_edge 模块提供激励信号

编写时间：2016-08-16

作者：至芯科技——奋斗的小孩

邮箱：zxopenhxs@126.com

```
*/
```

```
`timescale 1ns/1ps
```

```
module check_edge_tb;
```

```
    reg clk;
```

```
    reg rst_n;
```

```
    reg signal;
```

```
    wire neg_flag;
```

```
    wire pos_flag;
```

```
    initial begin
```



```
    clk = 1'b1;

    rst_n = 1'b0;

    signal = 1'b1;

    # 200.1

    rst_n = 1'b1;

    # 200

    signal = 1'b0;//下降沿

    # 200

    signal = 1'b1;//上升沿

    # 200

    $stop;

end

always # 10 clk = ~clk;

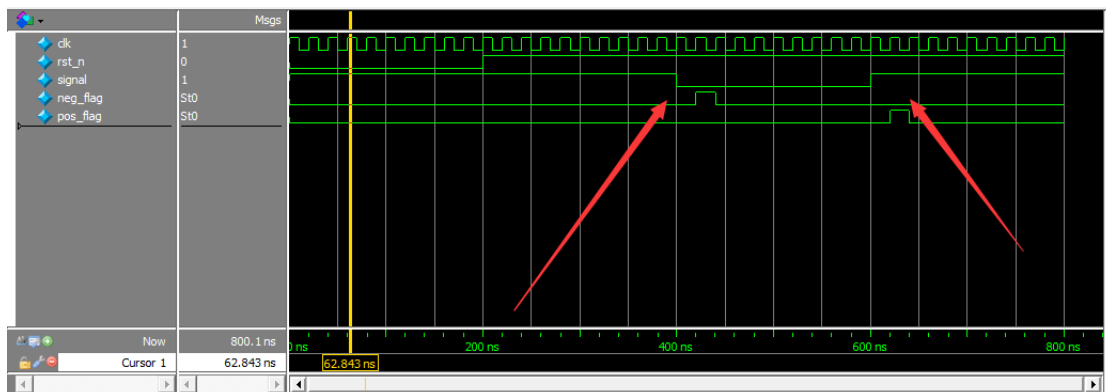
check_edge check_edge_dut(
    .clk(clk),
    .rst_n(rst_n),
    .signal(signal),
    .neg_flag(neg_flag),
    .pos_flag(pos_flag)
```



);

endmodule

仿真波形如下:



当输入信号有下降沿时，neg_flag输出了高脉冲，输入信号有上升沿时，pos_flag输出了高脉冲。

检测边沿是用寄存器来完成的，所以脉冲的输出会晚一个时钟输出，在高频率时钟对低频率信号检测中，没有影响。设计正确。如果还是有不明白的读者可以发邮件到我邮箱或者加群询问。

制作人: 奋斗的小孩

fpga 交流群: 282124839