

2.按键消抖例程说明

2.1 PGX-Nano 开发板简介

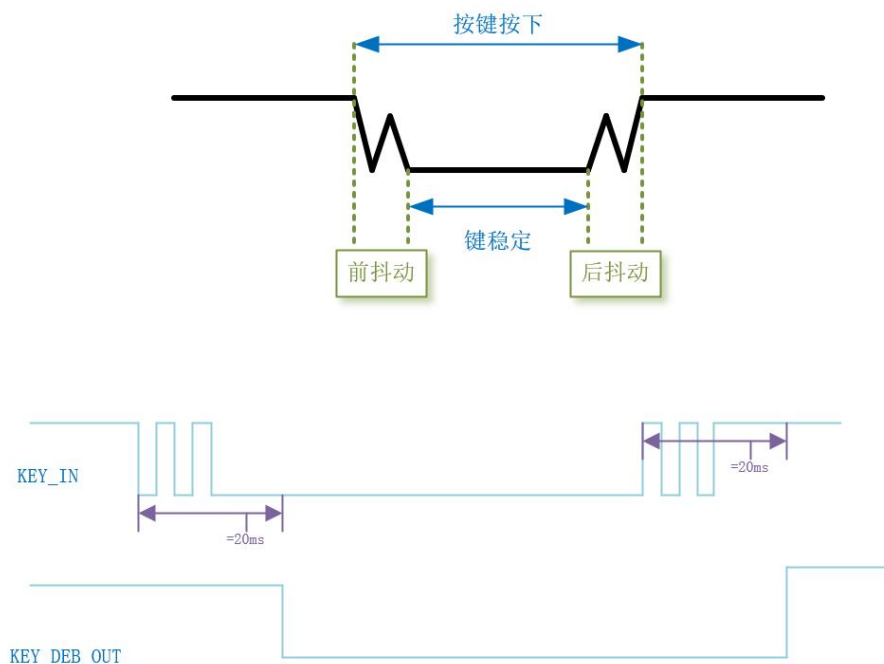
PGX-Nano 开发板提供了 5 个用户按键（PB0~PB4），IO 状态默认高电平，在按键按下时，IO 上电压为低电平（详情请查看“PGX_Nano 开发板硬件使用手册”）。

2.2 实验目的

对按键 PB0（即 S0 按键）进行按键消抖

2.3 实验原理

机械式弹片按键，在按下或松开时会有机械抖动，导致在按下或松开时按键的状态不稳定，在快速的变化，在使用按键输入信号时如果采集了抖动时的状态，会导致工程出现不可控的变化，故而我们需要将这段时间的抖动信号给滤除掉，故此实验称之为按键消抖；



前后抖动时间约为 5~10ms，取按键抖动区间开始标识，持续 10-20ms 后标识归零，在抖动区间内输出保持，非消抖区间，按键状态输出。

2.4 实验源码设计

```
`timescale 1ns / 1ps
`define UD #1
module btn_deb_fix#(
    parameter          BTN_WIDTH = 4'd1,
    parameter          BTN_DELAY = 20'hF423F
)
(
    input               clk,
    input               [BTN_WIDTH-1:0] btn_in,

    output reg [BTN_WIDTH-1:0] btn_deb_fix
);
    reg [19:0]          cnt[BTN_WIDTH-1:0];
    reg [BTN_WIDTH-1:0] flag;

    reg [BTN_WIDTH-1:0] btn_in_reg;
    always @(posedge clk)
    begin
        btn_in_reg <= `UD btn_in;
    end

    genvar i;
    generate
    begin
        for(i=0;i<BTN_WIDTH;i=i+1)
        begin
            always @(posedge clk)
            begin
                if (btn_in_reg[i] ^ btn_in[i]) //取按键边沿开始抖动区间标识
                    flag[i] <= `UD 1'b1;
                else if (cnt[i]== BTN_DELAY) //持续 20ms 后归零
                    flag[i] <= `UD 1'b0;
                else
                    flag[i] <= `UD flag[i];
            end

            always @(posedge clk)
            begin
                if(cnt[i]== BTN_DELAY) //计数 20ms 时归零
                    cnt[i] <= `UD 20'd0;
                else if(flag[i]) //抖动区间有效时计数
                    cnt[i] <= `UD cnt[i] + 1'b1;
                else //非抖动区间保持 0
                    cnt[i] <= `UD 20'd0;
            end
        end
    end
endgenerate
endmodule
```

```

end

always @(posedge clk)
begin
    if(flag[i])                //抖动区间，消抖输出保持
        btn_deb_fix[i] <= `UD btn_deb_fix[i];
    else                        //非抖动区间，按键状态传递到消抖输出
        btn_deb_fix[i] <= `UD btn_in[i];
    end
end
end
endgenerate
endmodule

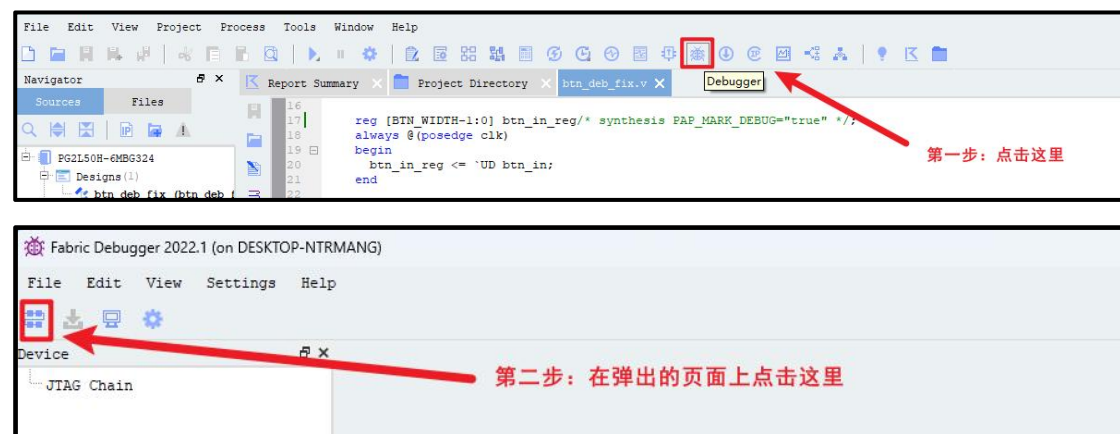
```

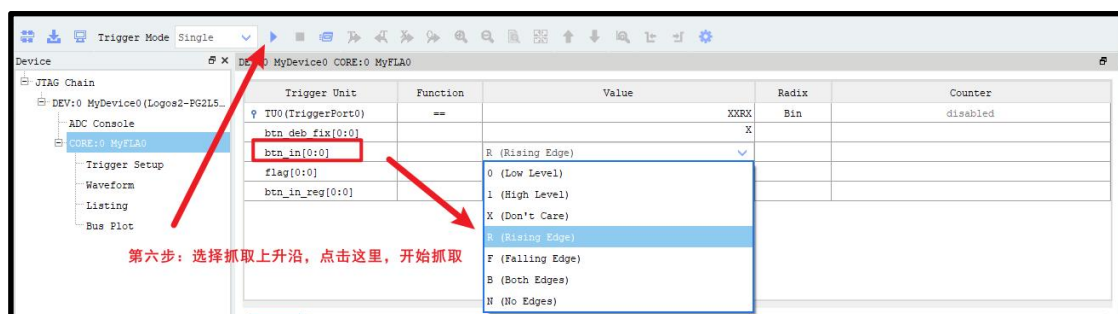
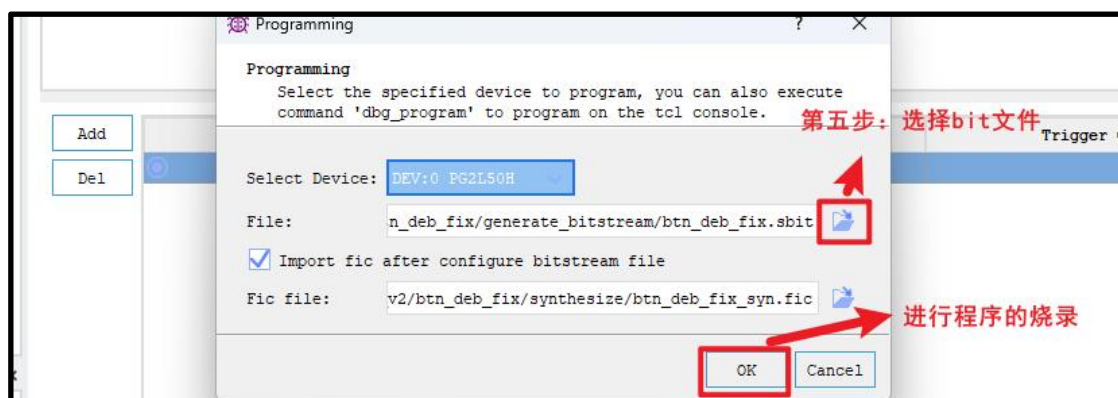
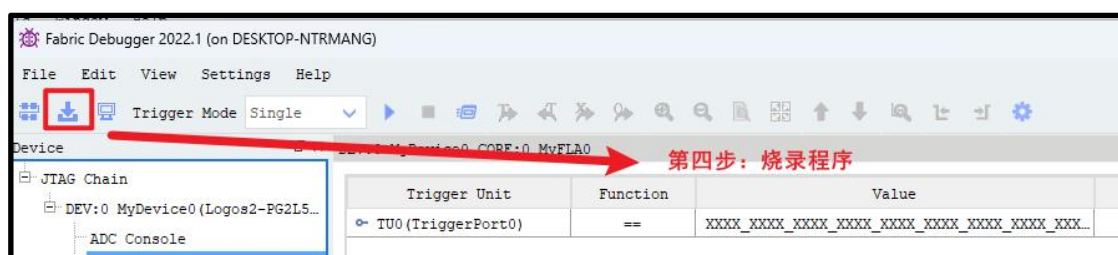
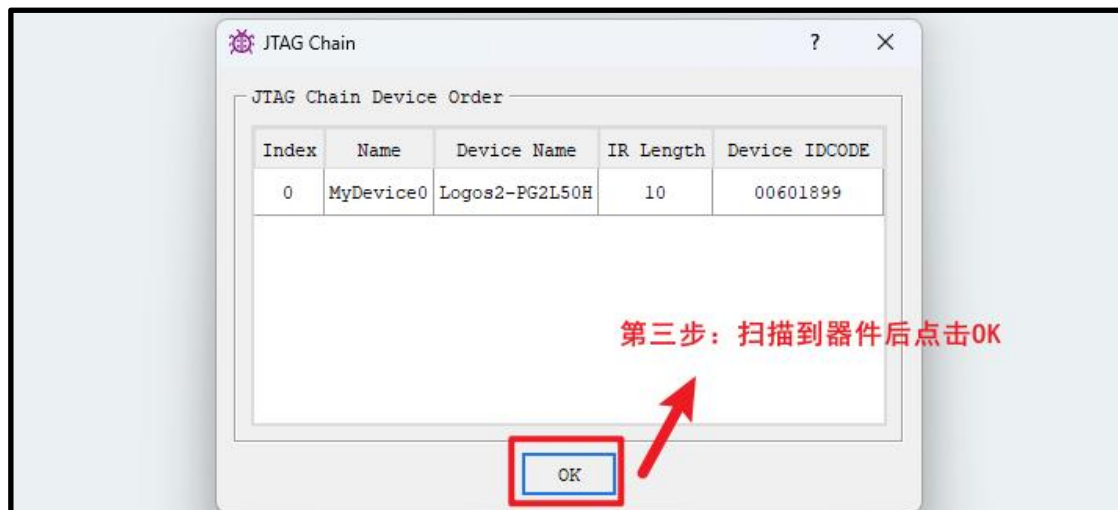
这个 module 的设计中新增加一种语法说明：**parameter**；在 verilog 中 parameter 是对常量进行定义，将 parameter 定义放在 module 的接口中是可进行模块传递，传递方式请看后面模块例化；

2.4 实验现象

1、首先按照下图步骤使用 debug 核抓取数据，需要保持板卡在上电状态。

（debug 核的快速使用可参考 PDS 快速使用手册，文档位置：1_Demo_document\工具使用篇\02_PDS 快速使用手册）





2、此时点击 PB0（即 S0 按键），观察到消抖前按键信号有明显的抖动，消抖后按键信号保持稳定。

