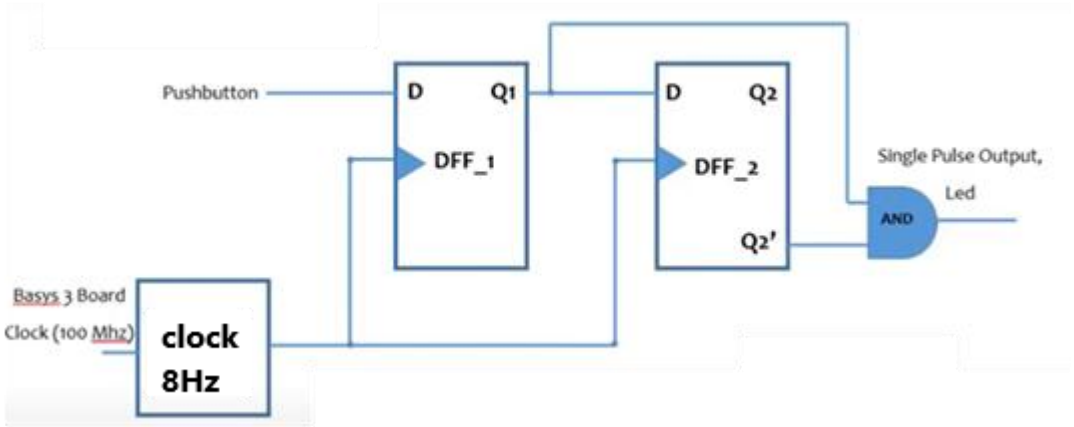
# button debounce

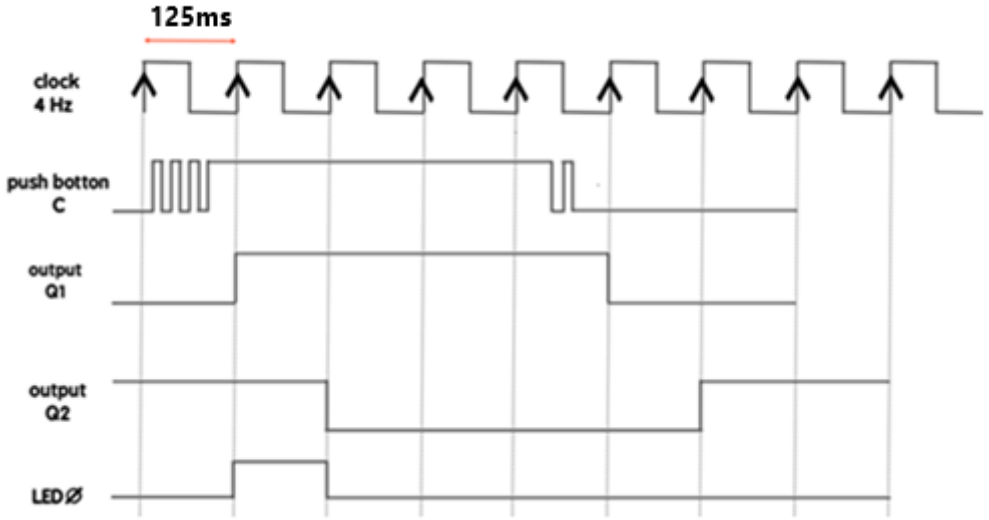
Dec 17,2024
by SIKWON

FPGA의 버튼을 누를 때 단일 펄스만 생성하도록 Verilog에
간단한 디바운싱 회로를 구현.

FPGA 의 버튼을 눌렀다 놓으면 푸시 버튼 신호에서 예기치 않은 위아래 바운스가
많이 발생. 디바운싱 회로는 예상대로 바운싱 없이 clock 기간을 가진
단일 pulse만 생성




FPGA의 버튼에 대한 디바운싱 회로




디바운싱 회로의 예상 파형

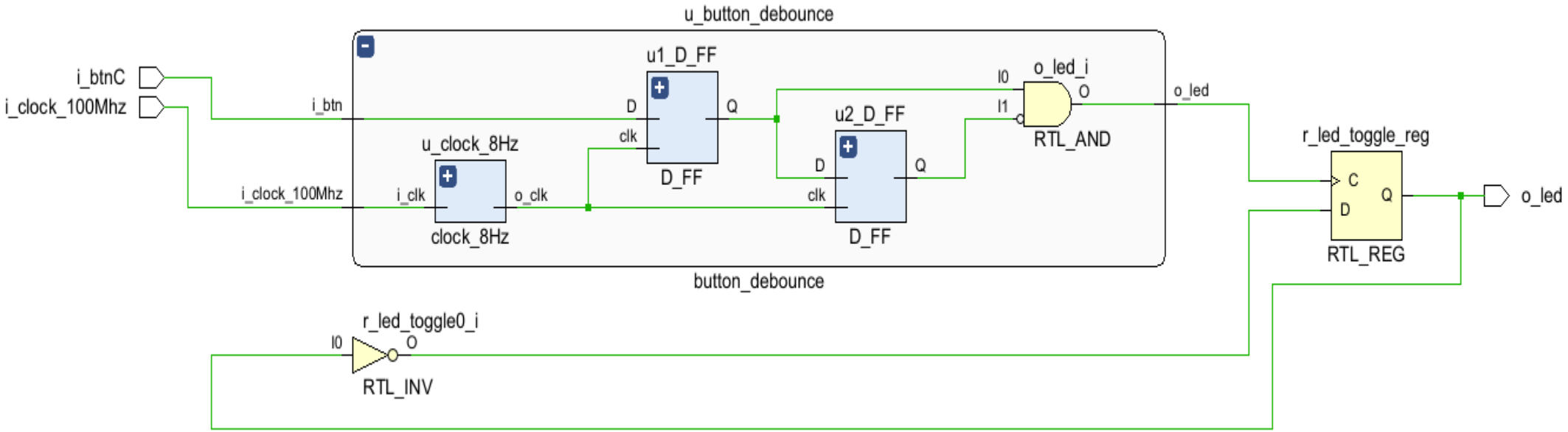
100Mhz → 8Hz

$$\frac{100M}{8} = 12{,}500{,}000 \text{ cycles}$$

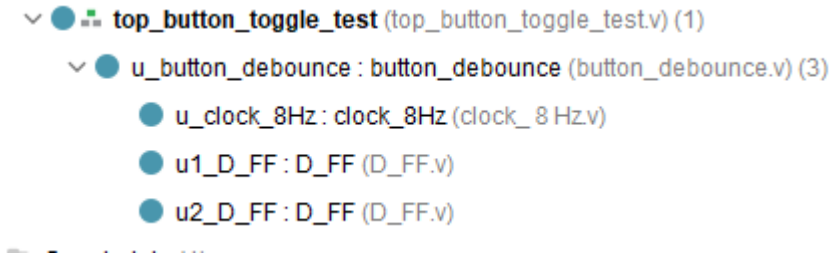
$$\frac{12{,}500{,}000}{2} = 6{,}250{,}000 \ (10us * 62500개)$$

# ELABORATED DESIGN Schematic

아래는 btnC를 1번 눌렀다 떼면 led가 on <--> off  toggle



# verilog code




```
`timescale 1ns / 1ps

module top_button_toggle_test (
                input i_btnC,
                input i_clock_100Mhz,
    output o_led
);

wire w_clock_8Hz;
wire w_btn_debounce;
reg  r_led_toggle = 1'b0;

button_debounce u_button_debounce (
   .i_btn(i_btnC),
   .i_clock_100Mhz(i_clock_100Mhz),
   .o_led(w_btn_debounce)
);

    always @(posedge w_btn_debounce)
    begin
        r_led_toggle <= ~r_led_toggle;
    end

    assign o_led = (r_led_toggle == 1) ? 1'b1 : 1'b0;
endmodule
```

top_button_toggle_test.v

```
`timescale 1ns / 1ps

module button_debounce(
                input i_btn,
                input i_clock_100Mhz,
                output o_led
);

                wire w_out_clk;
                wire w_Q1, w_Q2, w_Q2_bar;

                clock_8Hz u_clock_8Hz(
                                .i_clk(i_clock_100Mhz),
                                .o_clk(w_out_clk)
                );

                D_FF u1_D_FF(
                                .clk(w_out_clk),
                                .D(i_btn),
                                .Q(w_Q1)
                );

                D_FF u2_D_FF(
                                .clk(w_out_clk),
                                .D(w_Q1),
                                .Q(w_Q2)
                );

                assign w_Q2_bar = ~w_Q2;
                assign o_led = w_Q1 & w_Q2_bar;
endmodule
```

button_debounce.v

```
`timescale 1ns / 1ps

module clock_8Hz(
    input i_clk,   // input clock of the basys3
    output reg o_clk   // 8Hz clock
);

   reg [25:0] i_count=0;

    always @(posedge i_clk)
    begin
        i_count <= i_count + 1;
        if (i_count == 62_500)    // 8Hz 125000/2 ->
62500
        begin
                i_count <= 0;
                o_clk = ~o_clk;
        end
     end
endmodule
```

< clock_8Hz.v >

# verilog code

```
`timescale 1ns / 1ps

module D_FF(
      input clk,
      input D,
      output reg Q,
      output reg Qbar
 );

  always @ (posedge clk)
  begin
        Q <= D;
        Qbar <= !Q;
   end
endmodule
```

D_FF.V

# Basys3-Master_orginal.xdc

```
## This file is a general .xdc for the Basys3 rev B board
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project

## Clock signal
set_property -dict { PACKAGE_PIN W5   IOSTANDARD LVCMOS33 } [get_ports i_clock_100Mhz]
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports i_clock_100Mhz]



## LEDs
set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports {o_led}]
##Buttons
set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports i_btnC]
set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
set_property CONFIG_MODE SPIx4 [current_design]
```