University of Montenegro

Faculty of electrical engineering

Course: Automated constructon ef electrical circuits and systems

Theme: Real alarm

Students:

Milan Rešetar 8/18

Anđela Kandić 11/18

Date and place:

19/11/2018,Podgorica

Contents:

Abstract

In this assignment we will simulate a real alarm. Problem and its solution will be explained in details in the following text,including verification on FPGA board.

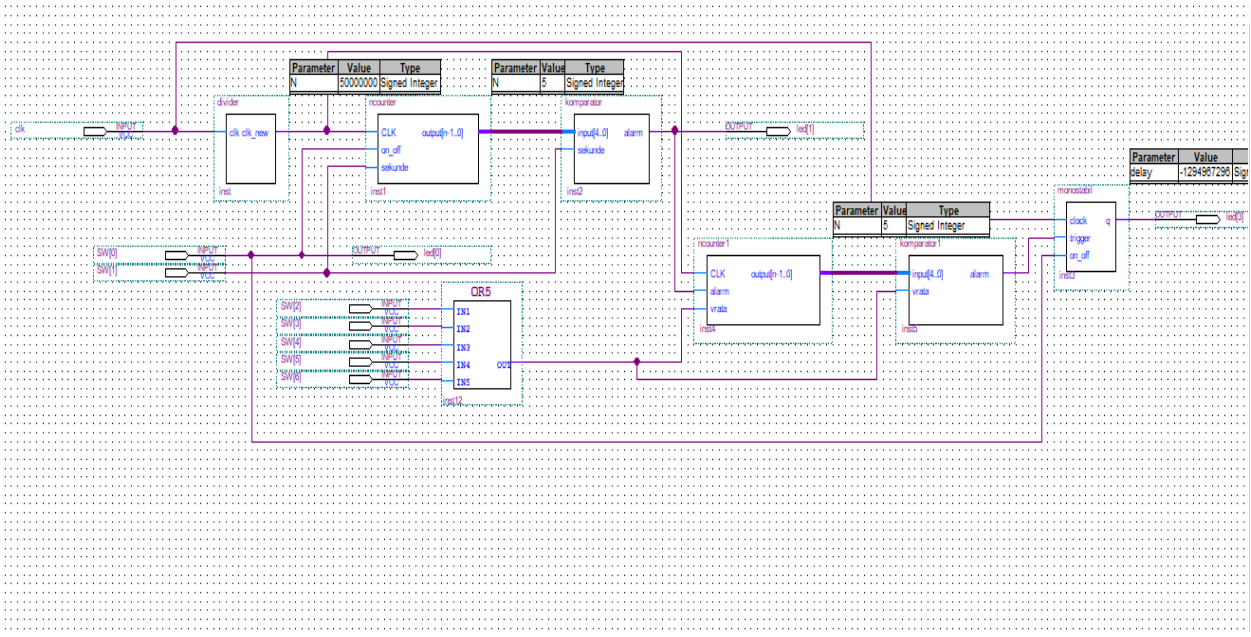
Problem description

We are using DE2-70 Altera FPGA board to simulate a real alarm. With SW[0] we can control the state of alarm(activated if switch is pushed that will turn the LED0 on and deactivated if it's not).With SW[1] is presented the input delay of 30seconds(after that delay LED1 will go on),than we can activate sensors(doors or trunk).With the activation of sensors the siren will turn on(LED3) after 30seconds,which represents the output delay.The siren(LED3) will be on for exactly one minute.We can also reset the system at any time with SW[0].

Hardware/software solution

As software solution Quartus version 9.1 is used.The code and block diagram is presented in VHDL.

Block diagram:



The code:

Divider-it allows us to use 50MHz clock signal from the FPGA board.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity divider is
generic(N:integer:=50000000);
port
( clk: in std_logic;
clk_new : out std_logic);
end divider;
architecture clk_div_behav of divider is
signal clk_temp : std_logic;
signal temp : integer range 0 to N-1;
begin
process(clk, clk_temp)
begin
if(clk'event and clk='0') then
if(temp=N/2-1)then
temp<=temp+1;
clk_temp<='1';
```

```
elsif (temp=N-1) then
temp <= 0;
clk_temp<='0';
else
temp<=temp+1;
end if;
clk_new<=clk_temp;
end if;
end process;
end clk_div_behav;
```

Counter-counts 30seconds input delay.

```
library ieee;
use ieee.std_Logic_1164.all;
use ieee.std_Logic_unsigned.all;
entity ncounter is
generic(N:integer:=5);
port(CLK:in std_logic;
output:out std_logic_vector(N-1 downto 0);
on_off:in std_logic;
sekunde:in std_logic);
end ncounter;
```

```
architecture arhitektura of ncounter is
signal brojac:std_logic_vector(N-1 downto 0);
begin
process(CLK,on_off)
begin
if(CLK' event and CLK ='1') then
if(on_off='1') then
if(sekunde='0')then
brojac<="00000";
elsif (brojac="11110" and sekunde='1') then
brojac<="11110";
else
brojac<=brojac+1;
end if;
elsif(on_off='0') then
brojac<="00000";
end if;
end if;
end process;
output<=brojac;
end arhitektura;
```

Comparator-that will stop the counter after 30 seconds and give signal for turning the LED1 on.

```
library ieee;

use ieee.std_logic_1164.all;

entity komparator is

port(input:in std_logic_vector(4 downto 0);

alarm:out std_logic;

sekunde: in std_logic);

end komparator;

architecture arh of komparator is

begin

process(input,sekunde)

begin

if(sekunde='1') then

case  input is

when "11110" => alarm<='1';

when others => alarm<='0';

end case;

elsif(sekunde='0') then

case input is

when "00000" => alarm<='0';

when others => alarm<='0';
```

```
end case;

end if;

end process;

end arh;
```

The codes for counter1 and komparator1 are exactly the same as for the counter and komparator with slightly modified inputs as shown in the block diagram above.

Monostabil-last component used to keep the siren on for precisely one minute.

```
LIBRARY IEEE;

USE IEEE.STD_LOGIC_1164.all;

USE IEEE.STD_LOGIC_ARITH.all;

USE IEEE.STD_LOGIC_UNSIGNED.all;

entity monostabil is

generic (delay : integer := 3000000000);

port(clock,trigger :in bit;

on_off:in std_logic;

q :out bit);

end monostabil;

architecture arch of monostabil is

begin
```

```
process(clock,on_off)
variable count :integer :=0;
variable trig_was :bit ;

begin
        if(on_off='1')then
        if(clock 'event and clock='1')then
                if trigger='1' and trig_was='0' then
                        count:=delay;
                        trig_was:='1';
                elsif count=0 then count:=0;
                else count:=count-1;
                end if;
                if trigger='0' then trig_was:='0';
                end if;

        end if;

        if count =0 then q<='0';
        else q<='1';
        end if;
        else
```

```
        q<='0';
        count:=0;
        end if;
end process;
end arch;
```

As hardware solution we have used FPGA board DE2-70 Altera(Cyclone II).

Signals and their pins:

- clk-PIN_AD15
- enable-SW[0]-PIN_AA23
- seconds-SW[1]-PIN_AB26
- door 1-SW[2]-PIN_AB25
- door 2-SW[3]-PIN_AC27
- door 3-SW[4]-PIN_AC26
- door 4-SW[5]-PIN_AC24
- trunk-SW[6]-PIN_AC23

Verification on board

Verification has been sucesfully executed as shown in the link below.

Link for the video

https://www.youtube.com/watch?v=IRln2stLMMM&t=7s

Literature

-Radovan D.Stojanović-AUTOMATIZOVANO PROJEKTOVANJE DIGITALNIH SISTEMA (VHDL i FPGA)

-DE2-70 User manual version 1.08