University of Montenegro

Faculty of Electrical Engineering

Course : Automated Design of Electrical Circuits and Systems

Theme: Step/Stair Light Timer

Mentor :

Prof. dr Radovan Stojanović

Students:

Marijana Kračunov

Gojko Ratković

Date and place: 11/18/2018

Podgorica

Contents

Summary

This is a lab project, made in MEDEL electronics lab at the University of Montenegro, with mentoring of prof. dr Radovan Stojanovic. In further text, the problem is explained in details and offered an elegant solution. In addition to that, we have enclosed high level design and description of our solution, alongside with hardware and software structure. The accent is on LED control, using a button and its demonstration on FPGA board DE2-70.

Abstract

One of the goals of this assignment is to demonstrate how to control FPGA board using Quartus 9.1 Altera software with control signals i.e. switches and keys available on DE2-70 board, to whom we can send logic 1 or 0 and activate LED for certain amount of time. LED is activated with KEY[0] on board, which must be pressed for at least 0.6 seconds in order to trigger the light.The amount of time that the LED is ON depends on the position of first switch(SW[0]) which can vary from 30 to 60 seconds.

The task has been presented in block diagram which allows us to divide one bigger problem into smaller pieces to avoid distractions and possible errors. This makes it easier to detect possible unwanted actions, make simple changes and redefine our problem. In a similar way, this circuit can be used as a part of few alike assignments, e.g. ALARM of system where user has limited time to enter the password, and when clock counts down to 0 the LED goes off.

Problem description

Our job is to project the STEP automat with debouncing button. The duration of LED on state is controlled with SW[0], which goes from 30 seconds (when toogled off) to 60 seconds (when toogled on). Output LED is assigned to LEDR0 on board and it represents the lightbulb. KEY0 is input button which detects the presence of a person on a step, which is symbolically implemented with a press of the button.

Hardware/software solution and simulation

As a software solution we used Altera Quartus 9.1 which is approved and tested for all FPGA and CPDL based systems. Source code was written in VHDL and the simulations were made and presented in Vector WaveForm file .

HIGH-LEVEL DESIGN

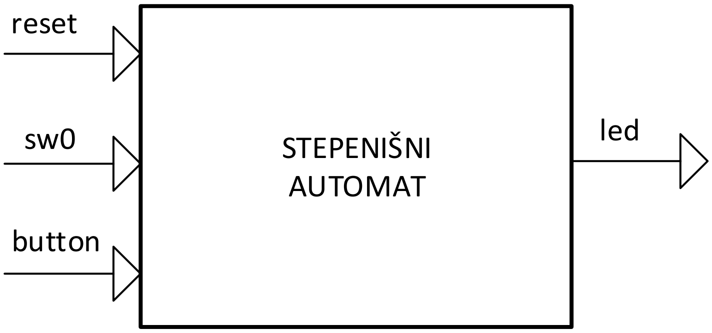


Fig. 1.0

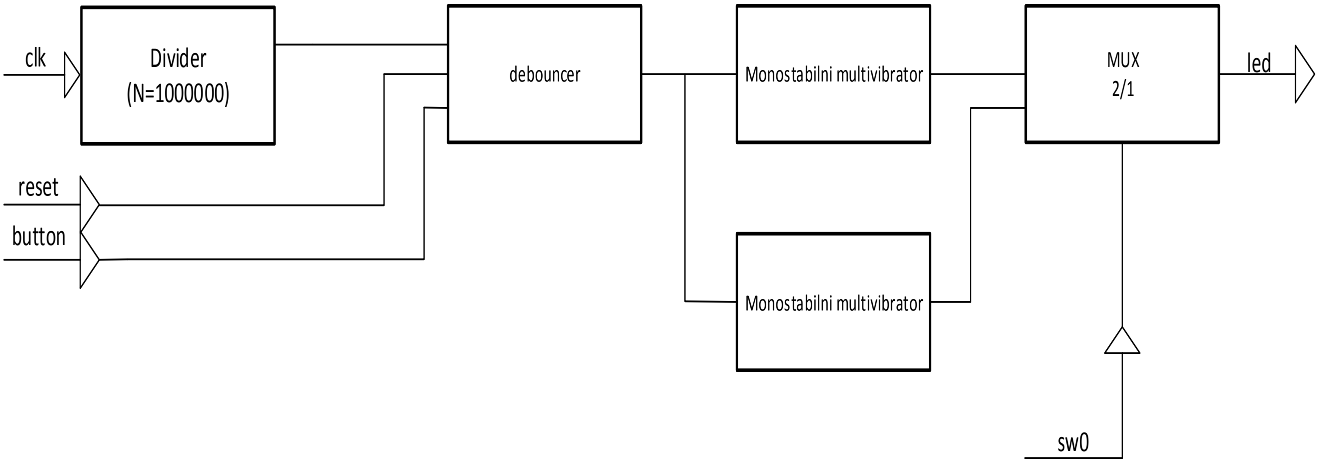


Fig. 1.1

Fig. 1.0 depicts the rough scheme and simplified version of our circuit, while fig. 1.1 illustrates more detailed scheme with separate parts which work as a unity. The drawings were made using Microsoft Visio.

Short explanation in order to better understand the source code:

For the realization of this project we used circuit of frequency divider, MUX 2/1, 2 monostable multivibrators and a debouncer. The input is internal clock, coming from the board, which is divided from 50 MHz to a signal of 5 Hz with above mentioned divider. This signal, along with button and reset are used as inputs for the DEBOUNCER circuit. Logic 1 of this singal needs to last for at least 0.6 seconds to create a short-timed impulse which activates or triggers the monostable multivibrator. One of the multivibrators activate the LED for 30 seconds, while the other one makes its 60 seconds. Which multivibrator is going to be used to power up the LED is dictated with control switch SW[0] of the MUX 2/1 whose working principle is well known and therefore needs no further explanation.

Source code:

```
 --Divider--
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:=10000000);
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;


--Debouncer--
library ieee;
```

```
use ieee.std_logic_1164.all;
entity Debounce is
      port ( clock,reset ,button_in : in std_logic;
                 pulse_out : out std_logic);
end debounce;
architecture behav of debounce is
constant count_max : integer := 3;
constant bin_active : std_logic := '1';
signal count : integer := 0 ;
type state_type is (idle, wait_time);
signal state : state_type :=idle;
begin
process (reset , clock)
           begin
           if( reset ='1') then    state <= idle;
pulse_out <= '0';
                 elsif (rising_edge(clock)) then
                       case (state) is
when idle =>
                                  if( button_in =bin_active) then
                                        state <= wait_time;
                                        count <= count+1;
                                  else
                                        state <= idle;
                                  end if;
                                  pulse_out <= '0';
                             when wait_time =>
                                  if(count =count_max) then
                                        count<= 0;
                                  if(button_in= bin_active) then
                                        pulse_out<='1';
                                  end if;
                                  state <= idle;
                                  elsif (button_in='0') then
                                  count <= 0;
                                  else
                                  count <= count + 1;
                                  end if;
                       end case;
                 end if;
           end process;
end architecture;
```

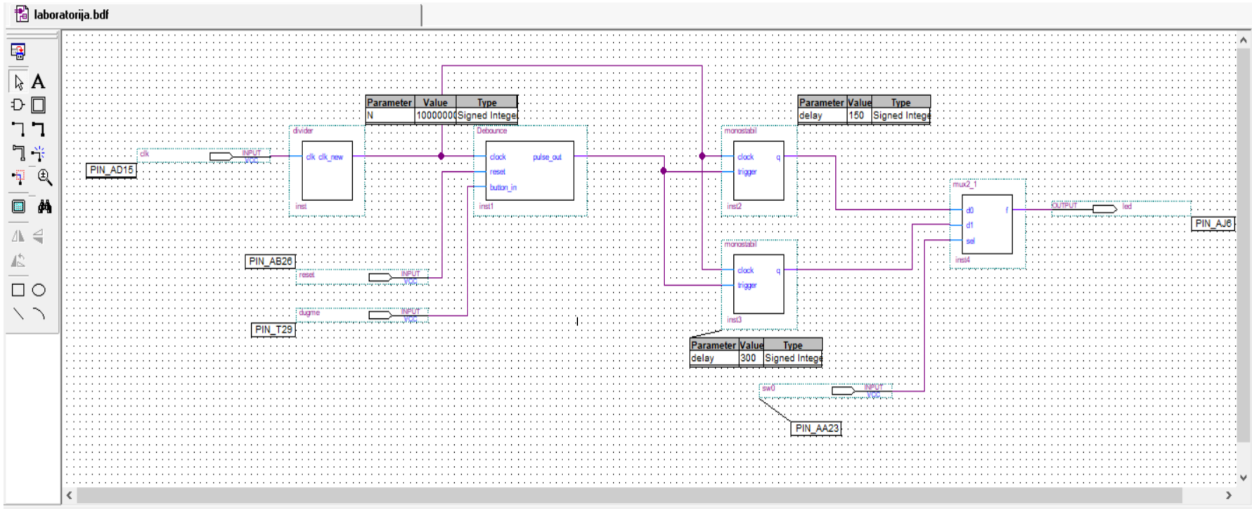
--Monostable multivibrator--

```
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 := 150); //150-za 30s, 300 za 60s
port(clock,trigger :in bit;
      q :out bit);
```

```
end monostabil;
architecture arch of monostabil is
begin
process(clock)
variable count :integer :=0;
variable trig_was :bit ;

begin

      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;
end process;
end arch;
```

```
--MUX2/1--
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY mux2_1 IS
  PORT (d0, d1, sel :IN  STD_LOGIC;
        f          :OUT STD_LOGIC);
END mux2_1;

ARCHITECTURE LogicFunc OF mux2_1 IS
BEGIN
  f <= (d0 AND (NOT sel)) OR (d1 AND sel);
END LogicFunc;
```

Block-diagram file:



laboratorija.bdf

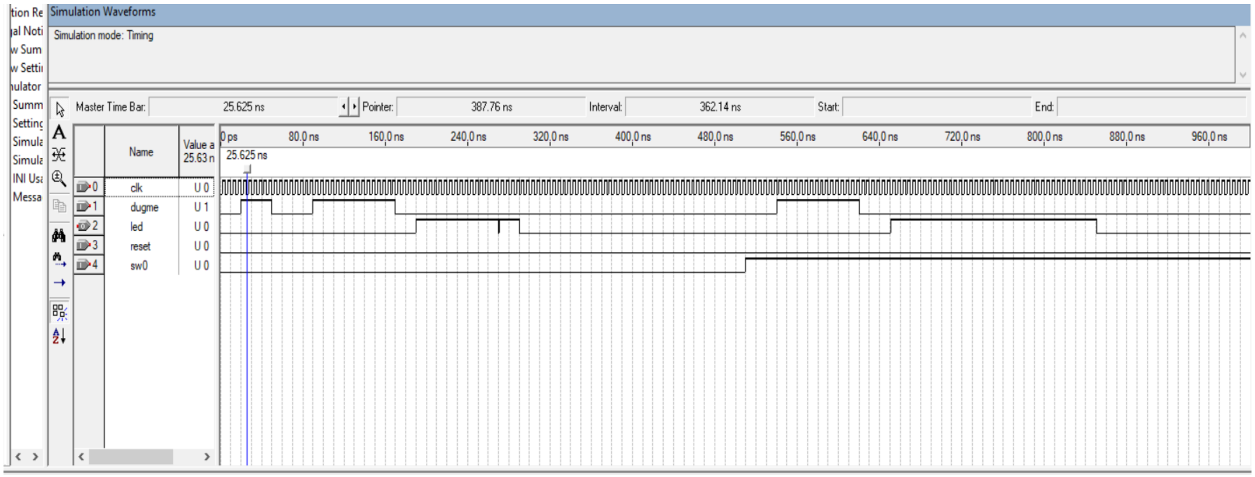
Parameter	Value	Type
N	10000000	Signed Intege

Parameter	Value	Type
delay	150	Signed Intege

Parameter	Value	Type
delay	300	Signed Intege

Fig. 1.2

Simulation in wvf (Vector Waveform File) :



Simulation Waveforms

Simulation mode: Timing

Master Time Bar: 25.625 ns Pointer: 387.76 ns Interval: 362.14 ns Start: End:

	Name	Value a 25.63 n
0	clk	U 0
1	dugme	U 1
2	led	U 0
3	reset	U 0
4	sw0	U 0

Fig. 1.3

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

Signals and assigned pins :

- clk- PIN_AD15 (internal clock of 50MHz frequency)
- reset- PIN_AB26 (SW[1])
- button- PIN_T29(KEY[0])
- sw0-PIN_AA23(SW[0])

Verification on board

Verification was successfully done in a way described in previous paragraph. As a proof we enclose to this statement a video, which was made in MEDEL electronics lab.

Link for the video

https://www.youtube.com/watch?v=qJvKbKp39GI

Literature

[1] Radovan D. Stojanovic AUTOMATIZOVANO PROJEKTOVANJE DIGITALNIH SISTEMA (VHDL i FPGA)

[2] DE2-70 User manual version 1.08