University of Montenegro

Faculty of electrical engineering

Course: Automated construction of electrical circuits and systems

Theme: Switch controlled LED

Students:

Milan Rešetar 8/18

Anđela Kandić 11/18

Date and place:

25/11/2018,Podgorica

Contents:

Abstract:

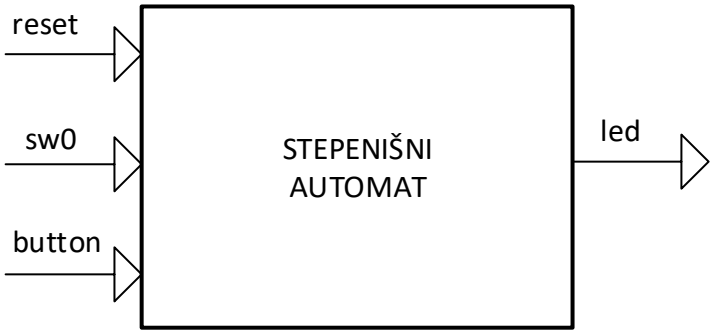
In this project we will show the working principle of a stepping automate with a debounce button. Problem and its solution will be explained in details in the following text,including verification on FPGA board.

Problem description

We are using DE2-70 Altera FPGA board to simulate a stepping automate.The purpose of this circuit is the keep LED on for the certain amount of time.To start the system we have to hold the button(KEY0) for at least 0.6 seconds.We use SW[0] to control the amount of time the LED will be on.If SW[0] is toggled on the LED will be active for 60 seconds and if it is off,then for 30 seconds.We also have a reset switch(SW[1]) used to ’control’ the button,when SW[1] is active the system cannot be started.

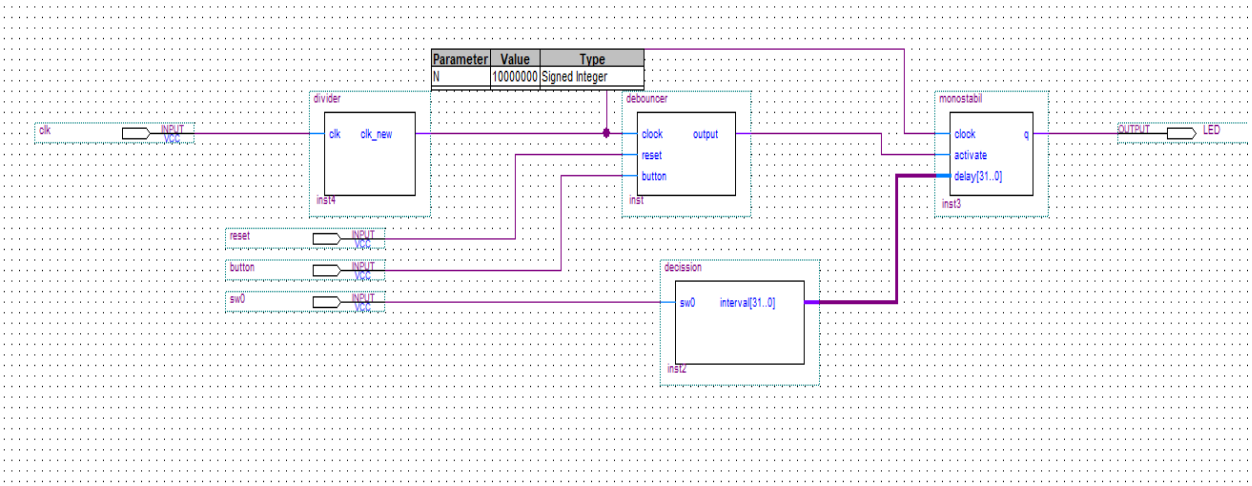
Hardware/software solution

High-level design



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

Block diagram:



The 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 debouncer is
port ( clock, reset ,button : in std_logic;
output : out std_logic );
end debouncer;
architecture arch of debouncer is
constant count_max : integer := 3;
constant button_active : std_logic := '0';
signal count : integer := 0 ;
type state_type is (passive, waiting);
signal state : state_type :=passive;
begin
process (reset,clock)
begin
if( reset ='1') then
state <= passive;
output<= '0';
elsif (rising_edge(clock)) then
case (state) is
when passive =>
if( button = button_active) then
state <= waiting;
count <= count+1;
```

```
else
state <= passive;
end if;
output <= '0';
when waiting =>
if(count =count_max) then
if(button ='1') then
count <= 0;
elsif(button ='0') then
if(button = button_active) then
output<='1';
end if;
state <= passive;
elsif (button ='1') then
count <= 0;
else
count <= count + 1;
end if;
end case;
end if;
end process;
end architecture;
```

Decission

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity decission is
port ( sw0 : in std_logic;
interval : out integer) ;
end entity;
architecture arch of decission is
begin
with sw0 select
interval <= 150 when '0',-- za 30 sekundi
            300 when '1'; -- za 60 sekundi
end architecture;
```

Monostabil

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity monostabil is
port(clock,activate:in bit;
delay: in integer;
```

```
q :out bit);
end monostabil;
architecture arch of monostabil is
begin
process(clock)
variable count :integer :=0;
variable activate_was :bit ;
begin
if(clock 'event and clock='1')then
if activate='1' and activate_was='0' then
count:=delay;
activate_was:='1';
elsif count=0 then
count:=0;
else
count:=count-1;
end if;
if activate='0' then
activate_was:='0';
end if;
end if;
if count =0 then
```

```
q<='0';

else

q<='1';

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
- reset-PIN_AB26
- sw0-PIN_AA23
- button-PIN_T29

Verification on board

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

Link for the video

https://www.youtube.com/watch?v=8GxUDEnXsIY

Literature

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

-DE2-70 User manual version 1.08