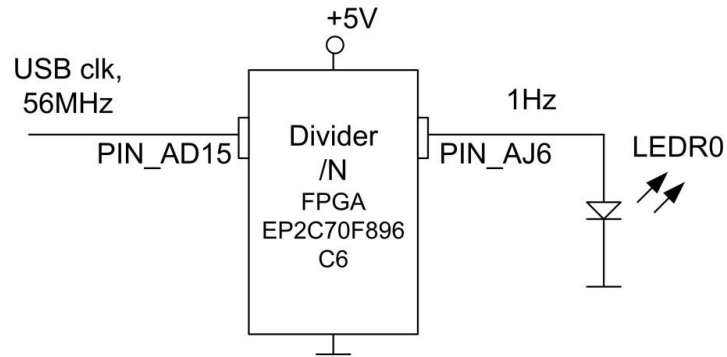


LAB1: Projektovati FPGA kolo, cip, koji na svom izlazu daje blink signal frekvencije 1Hz. Ispravnost i funkcionalnost kola provjeriti na DE2-70 ploci koristeći led diodu LEDR0 koja treba da blinka u ritmu 1sec



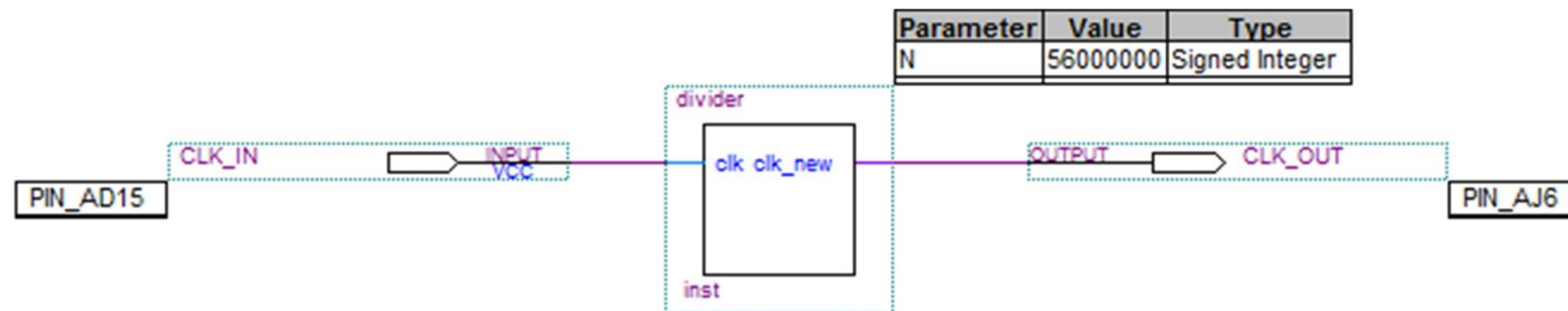
Koraci:

1. Projektovati kod djelitelja frekvencije, divider
2. Dodijeliti cip EP2C70F896C6
3. Dodijeliti pinove na cipu EP2C70F896C6 za ulazni klock 56MHz i izlazni 1Hz
4. Uploadovati pro*.sof file u cip i provjeriti ispravnost rada na DE2-70 ploci

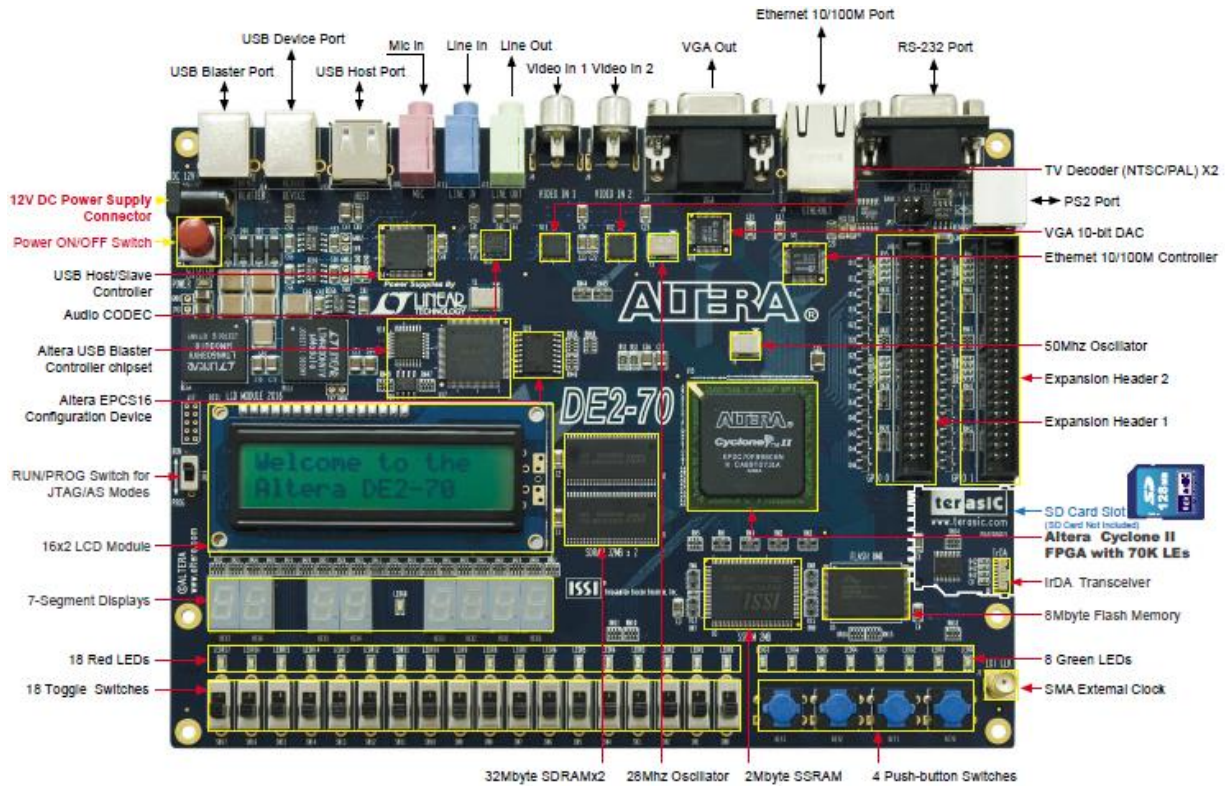
```

--- DE2-70 board
--- clk: PIN_AD15 USB CLK
--- clk_new: PIN_AJ6 LEDR0
--- FPGA EP2C70F896C6
-----clk 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:=56000000);
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;

```

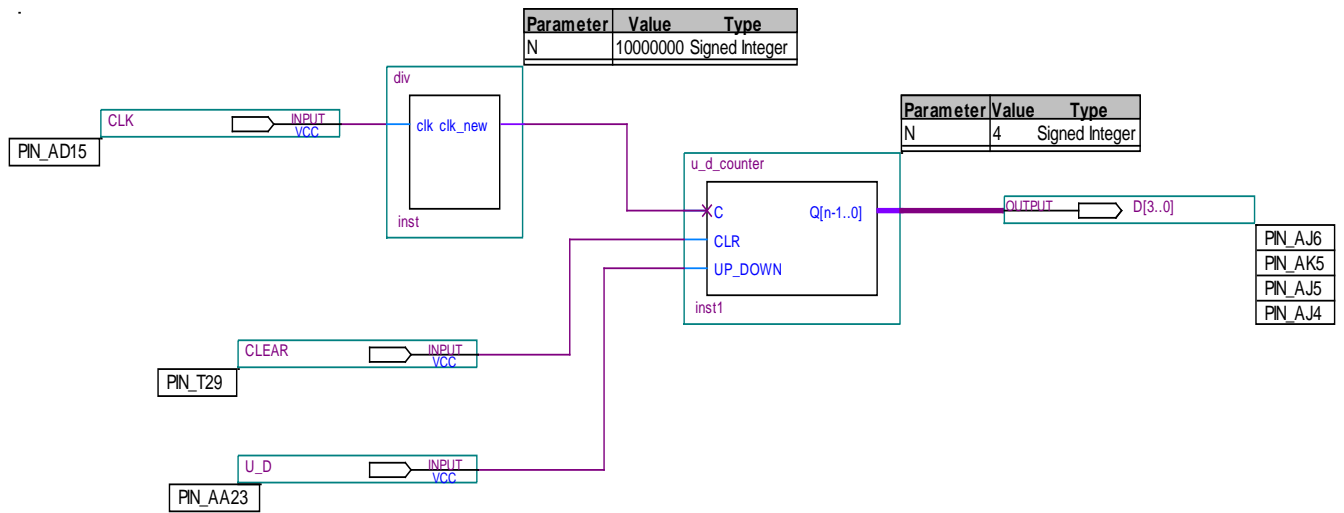


Lab vježba, DE2-70 eksperimentalna ploča



1. Pročitati fajl "Getting_Started_with_DE2-70_board.pdf"
2. Pročitati fajl "DE2_70_User_manual_v109.pdf"
3. Presnimiti odgovarajući CD

Lab2: Projektovati kolo brojača koji se pogoni klokom frekvencije 1Hz, a može da broji na gore ili na dolje, što zavisi od stanja ulaza U_D. Brojac ima i asinhroni clear ulaz, CLEAR. Izlazi brojaca, D[3..0] su vezani na LED diode na DE2-70 ploči.



U_D: SW0

CLEAR: KEY0

CLK: 50MHz clock

D[0]: LEDR0, D[1]: LEDR1, D[2]: LEDR2, D[3]: LEDR3,

Pomocni kodovi:

-----clk divider-----

Library ieee; Use ieee.std_logic_1164.all;

Use ieee.std_logic_arith.all; Use ieee.std_logic_unsigned.all;

entity div is

generic(N:integer:=24);

port (clk: in std_logic; clk_new : out std_logic);

end div;

architecture clk_div_behav of div 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;
```

--Up down counter

```
-- CLR 0

library ieee;

use ieee.std_logic_1164.all;

use ieee.std_logic_unsigned.all;

entity u_d_counter is
generic(N:integer:=4);

port(C, CLR, UP_DOWN : in std_logic;

      Q : out std_logic_vector(N-1 downto 0));
end u_d_counter;

architecture archi of u_d_counter is

signal tmp: std_logic_vector(3 downto 0);

begin

process (C, CLR)

begin

if (CLR='0') then

tmp <= "0000";

elsif (C'event and C='1') then

if (UP_DOWN='1') then

tmp <= tmp + 1;

else

tmp <= tmp - 1;
```

```

end if;

end if;

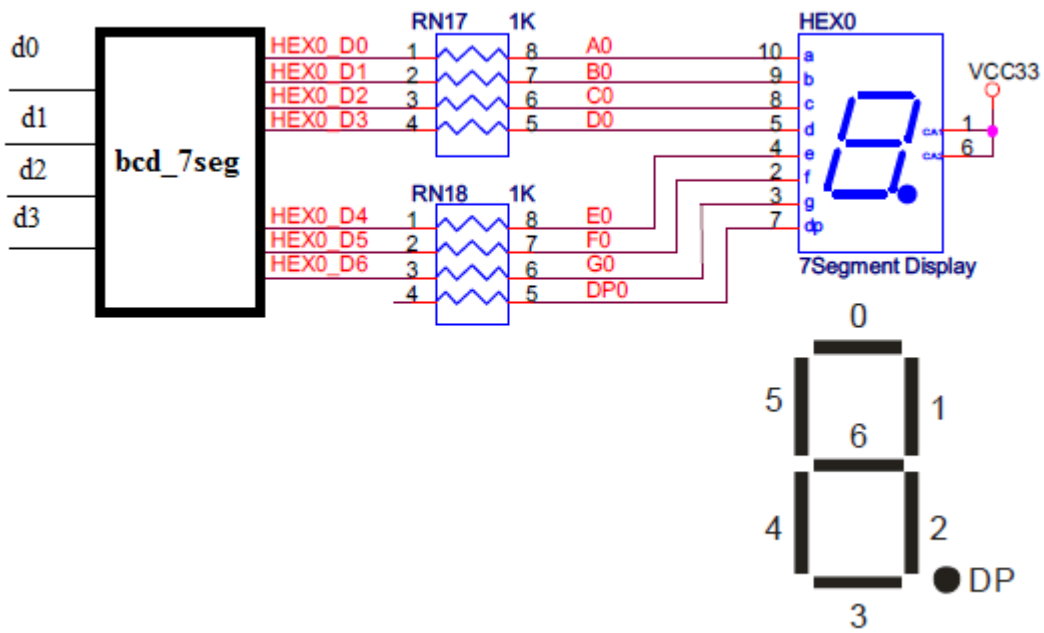
end process;












Q <= tmp;

end archi;

```

Lab 3: Projektovati kolo bcd_7seg dekodera. d0, d1, d2, d3 ulazi su priključeni na SW0, SW1, SW2, SW3 prekidače koji simuliraju dovodjenje 0 ili 1. a, b,c,d, e,f,g izlazi su priključeni na odgovarajuće segmente displeja. Raspored pinova čipa na koje su vezani odgovarajući ulazi i izlazi



	a	Output	PIN_AE8
	b	Output	PIN_AF9
	c	Output	PIN_AH9
	d	Output	PIN_AD10
	d0	Input	PIN_AA23
	d1	Input	PIN_AB26
	d2	Input	PIN_AB25
	d3	Input	PIN_AC27
	e	Output	PIN_AF10
	f	Output	PIN_AD11
	g	Output	PIN_AD12

```
--bcd_7seg.vhd
```

```
--Common Anode BCD-to-seven-segment decoder
```

```
LIBRARY ieee;
```

```
USE ieee.std_logic_1164.ALL;
```

```
ENTITY bcd_7seg IS
```

```
PORT(
```

```
  d3, d2, d1, d0 : IN STD_LOGIC;
```

```
  a, b, c, d, e, f, g : OUT STD_LOGIC);
```

```
END bcd_7seg;
```

```
ARCHITECTURE seven_segment OF bcd_7seg IS
```

```
  SIGNAL input : STD_LOGIC_VECTOR (3 downto 0);
```

```
  SIGNAL output : STD_LOGIC_VECTOR (6 downto 0);
```

```
BEGIN
```

```
  input <= d3 & d2 & d1 & d0;
```

```
  WITH input SELECT
```

```
  output <= "0000001" WHEN "0000",--display 0
```

```
  "1001111" WHEN "0001",--display 1
```

```
  "0010010" WHEN "0010",--display 2
```

```
"0000110" WHEN "0011",--display 3
```

```
"1001100" WHEN "0100",--display 4
```

```
"0100100" WHEN "0101",--display 5
```

```
"1100000" WHEN "0110",--display 6
```

```
"0001111" WHEN "0111",--display 7
```

```
"0000000" WHEN "1000",--display 8
```

```
"0001100" WHEN "1001",--display 9
```

```
"1111111" WHEN others;
```

```
-- Separate the output vector to make individual pin outputs.
```

```
a <= output(6);
```

```
b <= output(5);
```

```
c <= output(4);
```

```
d <= output(3);
```

```
e <= output(2);
```

```
f <= output(1);
```

```
g <= output(0);
```

```
End seven_segment;
```