Marko Markovic, VHDL SPR MAG, Kol I

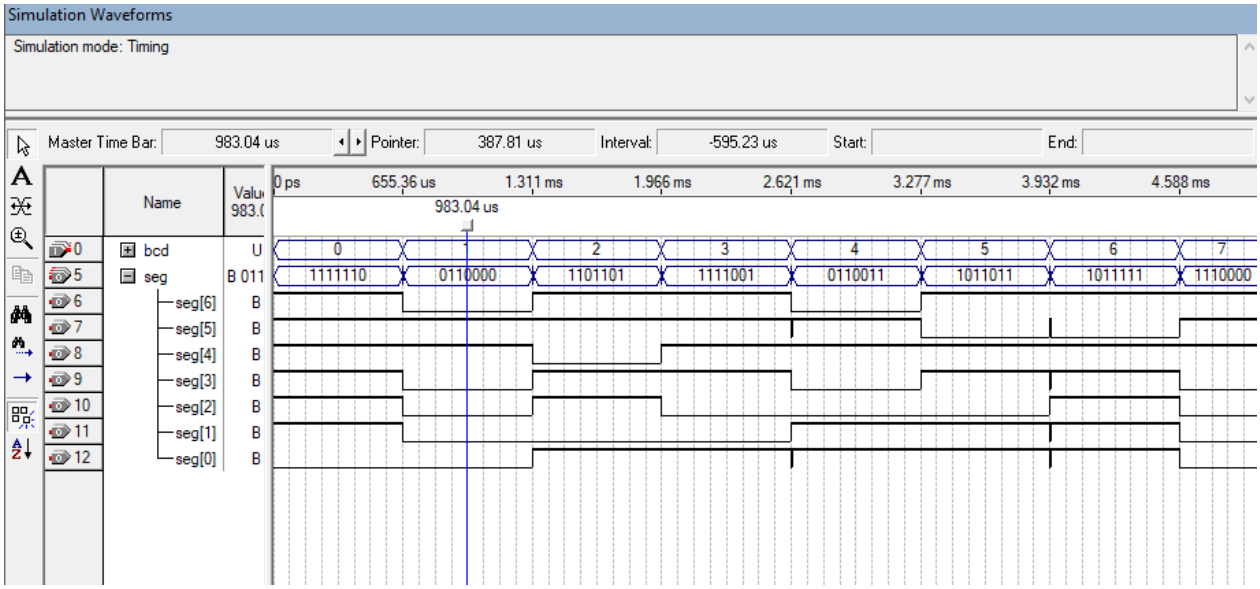
DIO A , ZADATAK 12

KOD:

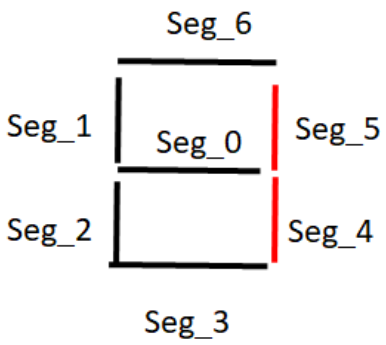
```
-- BCD to 7 segment counter
library ieee;
use ieee.std_logic_1164.all;

entity bcdto7seg is
	port
	(

		bcd: in std_logic_vector(3 downto 0);
		seg: out std_logic_vector(6 downto 0)
	);
end bcdto7seg;

architecture arh_bcdto7seg of bcdto7seg is
begin
	with bcd select
	seg<= "1111110" when "0000",
		"0110000" when "0001",
		"1101101" when "0010",
		"1111001" when "0011",
		"0110011" when "0100",
		"1011011" when "0101",
		"1011111" when "0110",
```

```
                "1110000" when "0111",
                "1111111" when "1000",
                "1110011" when "1001",
                "1110111" when "1010",
                "0011111" when "1011",
                "1001110" when "1100",
                "0111101" when "1101",
                "1001111" when "1110",
                "1000111" when "1111";
end arh_bcdto7seg;
```

SIMULACIJA:



Za bcd=”1” DEC, seg=”0110000”, BINARY, gdje se seg[6]=MSB



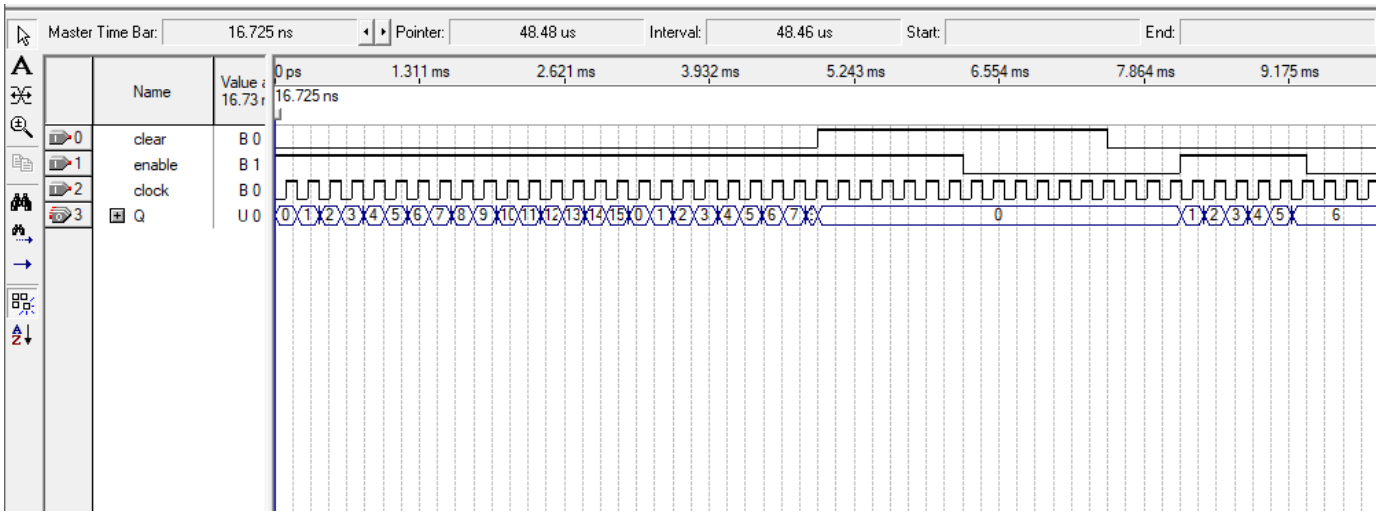
DIO B, ZADATAK 3

KOD:

```
-- 4 bits up counter
library ieee ;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity four_up_counter is
generic (N: natural :=3);
port
(
clock: in std_logic;
clear: in std_logic;
enable: in std_logic;
Q: out std_logic_vector (N downto 0)
);
end four_up_counter;

architecture arh_of_counter of four_up_counter is
```

```
signal temp: std_logic_vector (N downto 0);
begin
process (clock, clear, enable)
begin
if clear = '1' then
temp <= temp - temp;
elsif (falling_edge (clock)) then
if enable = '1' then
temp <= temp + 1;
end if;
end if;
end process;
Q <= temp;
end arh_of_counter;
```

SIMULACIJA



Kada je clear=0, enable=1, tada Q broji od 0-15-0-15. Kada je clear=1, bez obzira sta je enable, tada ke Q=0. Kada clear=0,

enable=1, pa potom 0, tada brojac zadrzava zadnju vrijednost, u nasem slucaju 6.