

**Digitalna Elektronika (PRK301) -- Konstrukcijske vježbe 2021./2022.**

Ime i prezime:	Domin Radić
Broj zadatka:	Z32
Način rješavanja:	VHDL
Razvojna ploča:	Nexys3

Izraditi sklop koji će na 7-segmetnim pokaznicima prikazivati dekadsku znamenku (0 – 9) koja rotira od lijevog ka desnom pokazniku svakih 0.5 sekundi, uz mogućnost promjene smjera.  
Dekadsku znamenku odabrati koristeći sklopke.

***Rješenje zadatka*****Glavni modul**

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;
```

```
entity z32 is  
  Port ( s : in  STD_LOGIC;  
        clk : in  STD_LOGIC;  
        br : in  STD_LOGIC_VECTOR (3 downto 0);  
        an : out STD_LOGIC_VECTOR (3 downto 0);  
        seg : out STD_LOGIC_VECTOR (6 downto 0));  
end z32;
```

```
architecture Behavioral of z32 is  
  type stanje is(s0,s1,s2,s3);  
  signal sad,bud:stanje;  
  signal clk_o:std_logic;  
  signal seg0:std_logic_vector(6 downto 0);
```

```
begin
```

```
fdg:entity work.freqDivGen generic map(50000000) port map(clk,clk_o); -- djelitelj frekvencije  
ss0:entity work.deg_seg port map(br,seg0);    -- dekoder
```

```
process(clk_o) -- sekvencijalni dio koda  
begin  
  if(rising_edge(clk_o))then --rastući brid signala takta  
    sad<=bud;  
  end if;  
end process;
```

process(sad) -- **kombinacijski dio koda**

begin

case sad is

when s0 =>

seg<=seg0; --**prikaz na 7-segmentnom pokazniku**

an<="0111"; --**aktivne anode**

if(s='0') then

bud<=s1;

else

bud<=s3;

end if;

when s1 =>

seg<=seg0;

an<="1011";

if(s='0') then

bud<=s2;

else

bud<=s0;

end if;

when s2 =>

seg<=seg0;

an<="1101";

if(s='0') then

bud<=s3;

else

bud<=s1;

end if;

when others =>

seg<=seg0;

an<="1110";

if(s='0') then

bud<=s0;

else

bud<=s2;

end if;

end case;

end process;

end Behavioral;

---

### **Djeljitelj frekvencije**

```
entity freqDivGen is
  generic(hz:integer:= 100000000);
  Port ( clk : in  STD_LOGIC;
        clk_o : out STD_LOGIC);
end freqDivGen;
```

```
architecture Behavioral of freqDivGen is
  signal temp:std_logic;
```

```
begin
  process(clk)
    variable br:integer range 0 to hz/2:=0;
    begin
      if(clk'event and clk='1')then
        br:=br+1;
        if(br>=hz/2)then
          temp<=not temp;
          br:=0;
        end if;
      end if;
    end process;
    clk_o<=temp;
```

```
end Behavioral;
```

---

### **Dekoder**

```
entity deg_seg is
  Port ( br : in  STD_LOGIC_VECTOR (3 downto 0);
        seg : out STD_LOGIC_VECTOR (6 downto 0));
end deg_seg;
```

```
architecture Behavioral of deg_seg is
```

```
begin
  with br select
    seg<=
      "0000001"when"0000", --0
      "1001111"when"0001", --1
      "0010010"when"0010", --2
      "0000110"when"0011", --3
      "1001100"when"0100", --4
      "0100100"when"0101", --5
      "0100000"when"0110", --6
      "0001111"when"0111", --7
      "0000000"when"1000", --8
      "0000100"when others; --9
```

```
end Behavioral;
```

## Prostor za objašnjenje VHDL koda

### Definiranje ulaza i izlaza sklopa

Prije deklaracije ulaza i izlaza potrebno je uključiti potrebne biblioteke koje obuhvaćaju operacije ključne za pisanje a kasnije i za samu izvedbu koda, nakon toga unutar entity-a dijela koda definiramo signal takta, ulaze i izlaze sklopa. Ulazi koje koristimo su: signal takta **clk** koji predstavlja signal ulazne frekvencije sklopa, **br** – dekadski broj koji unosimo putem sklopki na razvojnom sustavu i ulaz **s** o kojem ovisi smjer kretanja dekadskih brojeva na pokaznicima. Izlazi koje definiramo su: **an** – anode pokaznika koje se sastoje od 4 bita i **seg** – koji predstavlja 7-bitni kod za prikaz na 7-segmentnom pokazniku.

U architecture dijelu koda deklariramo novi tip podatka **stanje** sa 4 stanja (**s0,s1,s2,s3**). Nakon toga deklariramo signale za sadašnje i buduće stanje (**sad** i **bud**) koji su tipa **stanje**, signal takta **clk\_o** koji će predstavljati frekvenciju promjene između svake anode pokaznika i signal **seg\_o** kao pomoćni signal za prikaz dekadskih brojeva na pokazniku.

Nakon toga se redom pozivaju sklopovi za dijeljenje frekvencije **freqDivGen** i dekodier **deg\_seg**.

### Djelitelj frekvencije freqDivGen

Sklop **freqDivGen** koristimo kako bi podijelili frekvenciju koja će nam omogućiti da vidimo brojeve na pokaznicima. Ulazna frekvencija razvojnog sustava Nexys3 iznosi 100MHz, unutar **generic map** unosimo broj koji će podijeliti frekvenciju razvojnog sustava od 100Mhz kako bi dobili željenih 0.5Hz. Ovo smo napravili iz razloga što je je ulazna frekvencija jako velika, te ne bismo vidjeli nikakvu promjenu na pokaznicima ukoliko bi ostala ista.

### Dekoder deg\_seg

Sklop **deg\_seg** predstavlja dekodier, kao ulaz mu navodimo dekadski broj koji smo unijeli na sklopkama razvojnog sustava, a kao izlaz nam se vraća kod za prikaz na 7-segmetnom pokazivaču. Sve se izvršava unutar WITH/SELECT/WHEN izraza koja će u odnosu na naš uneseni 4-bitni binarni kod dati 7-bitni kod potreban za prikaz na pokazivaču. Segmenti pokazivača (a-g) su aktivni na logičku '0'.

## Sekvencijalni i kombinacijski dio koda

Ostatak koda je pisan kao FSM VHDL kod.

Prvi proces je dio sekvencijalnog VHDL koda te u njemu određujemo trenutno stanje sklopa.

Ukoliko se na našem signalu takta **cp\_o** pojavi rastući brid, naš sklop će okinuti te sadašnje stanje prelazi u buduće stanje, samim time se pokreće i proces koji je dio kombinacijskog dijela koda.

U kombinacijskom dijelu koda određujemo izlaz i buduće stanje sklopa, u ovom zadatku početno stanje je stanje **s0** koje ujedno predstavlja i prvu anodu, broj koji će biti prikazan je uvijek isti i on se prikazuje na svim ostalim stanjima (**s1,s2,s3**) tj. anodama pokaznika opisan kao **seg<=seg0**.

Svako buduće stanje aktivira određenu anodu (sve anode su aktivne na logičku '0'). Signal koji određuje buduće stanje sklopa je tipkalo **s**, koje funkcionira na način da pomoću IF petlje postavimo uvjet da li je tipkalo aktivno ili ne i na osnovu toga ćemo odrediti buduće stanje sklopa.

Ukoliko je na tipkalu stanje '0' anode se aktiviraju s lijeva na desno te broj rotira od lijevog ka desnom pokazniku, ako je u stanju '1' naše anode se aktiviraju u suprotnom smjeru te na pokaznicima vidimo obrnuti smjer kretanja dekadskih brojeva (od desnog ka lijevom pokazniku).

### *User Constrains File*

```
NET "br[0]" LOC = T10;  
NET "br[1]" LOC = T9;  
NET "br[2]" LOC = V9;  
NET "br[3]" LOC = M8;  
NET "an[0]" LOC = N16;  
NET "an[1]" LOC = N15;  
NET "an[2]" LOC = P18;  
NET "an[3]" LOC = P17;
```

```
NET "clk" LOC = V10;  
NET "s" LOC = A8;
```

```
NET "seg[6]" LOC = T17;  
NET "seg[5]" LOC = T18;  
NET "seg[4]" LOC = U17;  
NET "seg[3]" LOC = U18;  
NET "seg[2]" LOC = M14;  
NET "seg[1]" LOC = N14;  
NET "seg[0]" LOC = L14;
```