Realizovati drajver za dva sedmosegmentna displeja korišćenjem strukturnog opisa. Najpre je potrebno kreirati par entitet- arhitektura za opis drajvera za jedan sedmosegmentni element koji treba da ispisuje cifre u opsegu od 0 – 9. Nakon toga kreirati opis bloka koji ce da ulazni 7-bitni broj (da bi predstavili brojeve u osegu od 0 do 99 potrebno nam je 7 bitova) da rastavi na dva cetvorobitna broja (cifra desetica i jedinica) koji će da se vode na odgovarajuće displeje. Displeji su sa zadedničkom anodom (da bi se upalio segment potrebno je da mu se dovede logička jedinica). Za nedozvoljena stanja ulaza, izlaz staviti u stanje visoke impedanse Z.

Opis glavnog projekta

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity TwoSegDisp is
    Port ( Input : in  STD_LOGIC_VECTOR (6 downto 0);
           OutputD : out  STD_LOGIC_VECTOR (6 downto 0);
                                                OutputJ : out  STD_LOGIC_VECTOR (6 downto 0));

end TwoSegDisp;


architecture Behavioral of TwoSegDisp is
component Hex7segDriver is
    Port ( bcd : in  STD_LOGIC_VECTOR (3 downto 0);
           segY : out  STD_LOGIC_VECTOR (6 downto 0));
end component Hex7segDriver;
component convertor is
    Port ( Ulaz7bit : in  STD_LOGIC_VECTOR (6 downto 0);
           Izlaz10s : out  STD_LOGIC_VECTOR (3 downto 0);
           Izlaz1s : out  STD_LOGIC_VECTOR (3 downto 0));
end component convertor;
signal des: std_logic_vector(3 downto 0);
signal jed: std_logic_vector(3 downto 0);
BEGIN
```

```
        -- Instanciranje komponenti
   CONV: convertor PORT MAP (
         Ulaz7bit => Input,
         Izlaz10s => des,
         Izlaz1s => jed
        );
            BCDD: Hex7segDriver PORT MAP (
         bcd => des,
         segY => OutputD
        );
     BCDJ: Hex7segDriver PORT MAP (
         bcd => jed,
         segY => OutputJ
        );

end Behavioral;
```

Testbench za ispitivanje funkcionalnosti konvertora

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY convTB IS
END convTB;

ARCHITECTURE behavior OF convTB IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT convertor
    PORT(
         Ulaz7bit : IN  std_logic_vector(6 downto 0);
         Izlaz10s : OUT  std_logic_vector(3 downto 0);
         Izlaz1s : OUT  std_logic_vector(3 downto 0)
        );
    END COMPONENT;
```

```
 --Inputs
  signal Ulaz7bit : std_logic_vector(6 downto 0) := (others => '0');

            --Outputs
  signal Izlaz10s : std_logic_vector(3 downto 0);
  signal Izlaz1s : std_logic_vector(3 downto 0);

BEGIN

            -- Instantiate the Unit Under Test (UUT)
  uut: convertor PORT MAP (
        Ulaz7bit => Ulaz7bit,
        Izlaz10s => Izlaz10s,
        Izlaz1s => Izlaz1s
       );

  Ulaz7bit<="0111011", "1001101" after 10ns, "1100010" after 20ns,"0001100" after
30ns,"1111111" after 40ns;

END;
```

Testbench za ispitivanje celog projekta:

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

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--USE ieee.numeric_std.ALL;

ENTITY TwoTB IS
END TwoTB;

ARCHITECTURE behavior OF TwoTB IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT TwoSegDisp
    PORT(
         Input : IN  std_logic_vector(6 downto 0);
         OutputD : OUT  std_logic_vector(6 downto 0);
                                          OutputJ : OUT  std_logic_vector(6 downto 0)
       );
    END COMPONENT;
```

```
    --Inputs
    signal Input : std_logic_vector(6 downto 0) := (others => '0');

            --Outputs
    signal OutputD : std_logic_vector(6 downto 0);
            signal OutputJ : std_logic_vector(6 downto 0);
    -- No clocks detected in port list. Replace <clock> below with
    -- appropriate port name


BEGIN

            -- Instantiate the Unit Under Test (UUT)
    uut: TwoSegDisp PORT MAP (
          Input => Input,
          OutputD => OutputD,
                                          OutputJ => OutputJ
        );

Input<="0111011", "1001101" after 10ns, "1100010" after 20ns,"0001100" after 30ns,"1111111" after 40ns;
END;
```