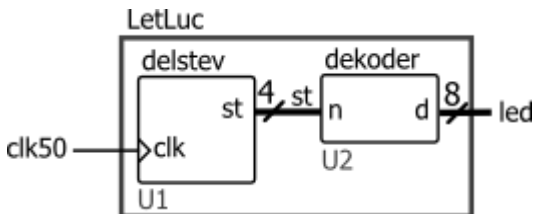




5. Vaja: struktarno načrtovanje

Opis strukture digitalnih vezij: [vhdl_str](#)

Povezovanje komponent



Naredi vezje za zaporedno prižiganje LED s povezavo komponent v jeziku VHDL in preizkusi delovanje na razvojnem sistemu.

- Uporabi projekt **LetLuc** iz prejšnje vaje. V programu Quartus izberi **File>New>VHDL file** in naredi novo datoteko z imenom **delstev.vhd**, v kateri naj bo opisan delilnik ure in 4-bitni števec z izhodnim signalom **st**. V opis vezja prenesi proces, ki opisuje delilnik in števec iz datoteke **letluc.vhd**.
- Naredi še eno novo datoteko z imenom **dekoder.vhd**, v kateri bo kombinacijski dekodirnik, ki pretvori 4-bitno binarno kombinacijo iz vhoda **n** v 8-bitno kombinacijo na izhodu **d** za krmitenje LED. Dekodirnik naj deluje tako, da bo bomo dobili ob zaporednih vrednostih na vhodu vtis potovanja prižgane LED v desno in levo. Npr. kombinacija "0000" se pretvori v "10000000", "0001" v "01000000", ... Opiši dekodirnik s stavkom **with...select** (glej primer: [vhdl_pretok.html#Dekod](#)).

n	d	n	d
0000	10000000	1000	00000001
0001	01000000	1001	00000010
0010	00100000	1010	00000100
0011	00010000	1011	00001000
0100	00001000	1100	00010000
0101	00000100	1101	00100000
0110	00000010	1110	01000000
0111	00000001	1111	10000000

- V datoteki **letluc.vhd** naredi deklaracijo komponent **delstev** in **dekoder**, notranjega signala **st** in stavke **port map** za vključitev ter povezavo komponent. Preizkusi delovanje na simulaciji in na razvojnem sistemu. Primer deklaracije komponente in povezav za komponento **delstev**:

```
architecture Opis of LetLuc is
  component delstev is -- komponenta
    port ( clk : in std_logic;
           st : out unsigned (3 downto 0));
  end component;

  signal st: unsigned(3 downto 0); -- notranji signal
begin

  U1: delstev port map (clk => clk50, st => st);
```

Pomnilnik ROM

Dekodirnik lahko opišemo tudi v obliki tabele, ki predstavlja pomnilnik ROM.

Naloga

Naredi novo VHDL datoteko za pomnilnik ROM, ki ima 4-bitni nepredznačeni vhod **adr** in 8-bitni izhod **d**. Deklariraj podatkovni tip in notranji signal, ki bo predstavljal vsebino pomnilnika:

```
type memory is array (0 to 15) of unsigned(7 downto 0);  
signal ROM: memory := ( "10000000", "01000000",... "10000000");
```

V arhitekturnem delu vezja zapiši stavek, ki da na izhod vrednost pomnilnika kot ga določa naslov **adr**:

```
d <= ROM(to_integer(adr));
```