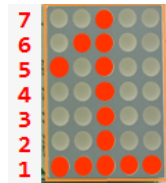


9. vaja: Matrični prikazovalnik



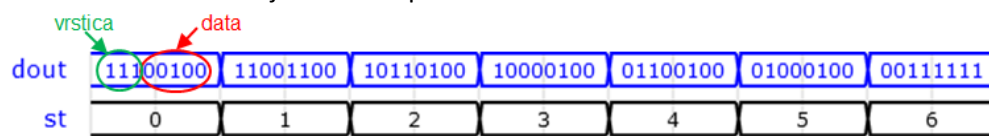
Naredili bomo vmesnik za prikaz števil na matričnem prikazovalniku.

Krmiljenje prikazovalnika

Matrični prikazovalnik za en znak ali števkovo vsebuje 35 LED v matriki sedem krat pet. Toliko LED ne priklopimo neposredno na krmilno vezje, ker bi zasedli preveč priključkov. Prikazovalniki zato izvajajo časovno multipleksiranje: v nekem trenutku krmilimo le eno vrstico. Če vrstice dovolj hitro preklapljam, ne opazimo utripanja ampak vidimo celoten prikaz.

V našem primeru je prikazovalnik povezan na vmesnik razširitvene plošče, ki mu vezje FPGA periodično pošilja 8-bitni podatek (**dout**) v obliki: **vrstica & data**. Vrstica je 3-bitno število med 1 in 7, podatek **data** pa predstavlja 5-bitno vrednost stanja LED v vrstici.

- Naredi model vezja s pomnilnikom ROM velikosti 7 x 5 bitov. V pomnilniku naj bo z enkami in ničlami predstavljena številka 1. V vezju naj bo 3-bitni števec **st**, ki šteje od 0 do 6 in določa naslov za branje pomnilnika. Sestavi 8-bitni izhodni signal **dout** iz številke vrstice in stanja vrstice iz pomnilnika.

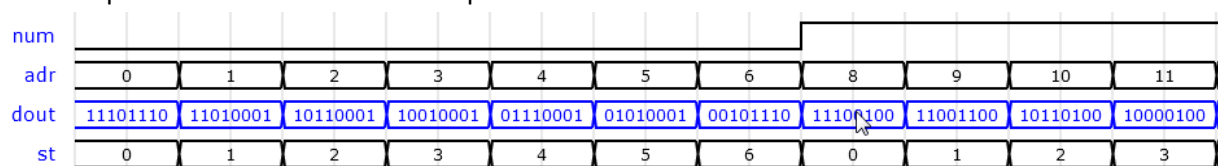


adr	data
0	"00100"
1	"01100"
2	"10100"
3	"00100"
4	"00100"
5	"00100"
6	"11111"

Slika prikazuje števec **st** in izhodni signal. Izhod je lahko določen kombinacijsko ali pa z zakasnitvijo ene periode ure.

S katerim izrazom najboljše preslikamo vrednost števca v signal vrstica?

- Razširi pomnilnik na 16 x 5 bitov in dodaj **na začetek** pomnilnika kombinacije, ki predstavljajo sliko števila 0. Med sliko številke 0 in 1 (pri **adr**=7) in na koncu (**adr**=15) naj bo v pomnilniku prazna vrstica (same ničle). Dodaj v vezje enobitni vhodni signal **num** s katerim izbiramo med prikazom 0 in 1 na matričnem prikazovalniku.



Kako sedaj najbolj enostavno določimo ustrezen naslov pomnilnika?

Zakaj je dobro, da je pomnilniku rezerviranih 8 celic za posamezno sliko številke?

- Razširi pomnilnik s slikami števil od 2 do 9 in naredi vezje, ki bo prikazovalo števila od 0 do 9. Ustrezno popravi deklaracijo vhodnega signala **num** in poišči rešitev za preostale kombinacije signala **num** (npr. prazen prikaz, prikaz šestnajstih števk, prikaz napake...).

Preizkus na razvojni plošči

Za preizkus na razvojni plošči je potrebno v vezje dodati še delilnik ure, ker imamo hitro sistemsko uro frekvence 50 MHz, vmesnik na razširitveni plošči pa zahteva počasnejše osveževanje vrstic.

1. Dodaj v vezje 17-bitni števec, ki bo v vlogi delilnika. Števec vrstic naj se spreminja le, kadar je 17-bitni števec enak 0.

Ugotovi s kakšno frekvenco se bo osveževal prikaz na matričnem prikazovalniku?

2. Vključi vezje za krmiljenje prikazovalnika (prikaz.vhd) v Quartusov projekt iz prejšnje vaje. Dodaj med zunanje priključke v glavni datoteki **sistem.vhd** signale **data**, **addr**, **clkout** in jih nastavi, kot je prikazano.

Prevedi vezje in preizkusi delovanje na razvojni plošči. Prikaz spreminjaj s kombinacijo pritisnjenih tipk, ki jih nastavljaš v orodju Tools > In-System Sources and Probes.

```
entity sistem is
  port (
    clk: in std_logic;
    data : inout unsigned(7 downto 0);
    addr : out unsigned(1 downto 0);
    clkout: out std_logic;
    led : out std_logic_vector(7 downto 0)
  );
end sistem;

architecture RTL of sistem is
  ...
begin

  addr <= "10"; -- izbiralnik razširitvene plošče
  clkout <= clk; -- ura razširitvene plošče

  uPrikaz: entity work.prikaz port map(
    clk => clk,
    num => unsigned(source),
    dout=> data
  );
```