

10. vaja: Grafika

Izdelali bomo komponento vezja za prikazovanje grafike na računalniškem monitorju. Grafiko prikazujemo v obliki zaporedja točk, ki jih digitalni sistem pošilja na monitor. Naloga vezja bo določitev 3-bitne vrednosti (barve) posamezne točke na podlagi trenutne koordinate (x, y).

Slika iz vrstice pomnilnika

Naredi vezje, ki prikaže eno vrstico s sliko iz štirih zaporednih točk. Vrednosti točk (0 ali 1) naj bodo zapisane v pomnilniku vrste ROM:

spletno orodje

Pomnilnik opišemo z zaporedjem pogojev:

```
if (adr=0) data=1
else if (adr=1) data=0
else if (adr=2) data=0
else data=1
```

VHDL

Definiramo 4-bitno vektorsko konstanto:

```
constant rom: unsigned(3 downto 0) := "1001";
data <= rom(to_integer(adr));
```

1. Model vezja naj vsebuje 4-bitni števec (**x**), 4-bitni vhod (**xs**) in pomnilnik. Števec določa naslov pomnilnika. Podatek iz pomnilnika naj gre na 3-bitni izhod (**rgb**), kadar je števec med 0 in 3, sicer pa naj bo na izhodu vrednost 2.

x	0	1	2	3	4	5	6	7	8
rgb	1	0	1			2			
adr	0	1	2	3	0	1	2	3	0

2. Razmisli, kako bi dopolnil model vezja, da bo vrstica iz pomnilnika (1001) prikazana na mestu, ki ga določa **xs**, npr. pri **xs=1** je zamaknjena za 1 mesto, pri 2 za dve mesti... Namig: preoblikuj vezje tako, da bo vsebovalo transformacijo koordinate **x** za vrednost **xs**.

x	0	1	2	3	4	5	6	7	8
xt	14	15	0	1	2	3	4	5	6
rgb	2	1	0	1		2			
adr	2	3	0	1	2	3	0	1	2

Slika v več vrsticah

1. Dodaj v opis vezja še koordinato **y**, premik **ys** in ustrezno transformirano koordinato **yt**. Pomnilnik razširi, tako da bo vseboval 4 x 4 enobitnih vrednosti. Napisan naj bo še vedno v obliki vektorja, iz katerega beremo posamezne bite. Prvi štirje biti predstavljajo prvo vrstico, naslednji drugo, itn. Opis v jeziku VHDL:

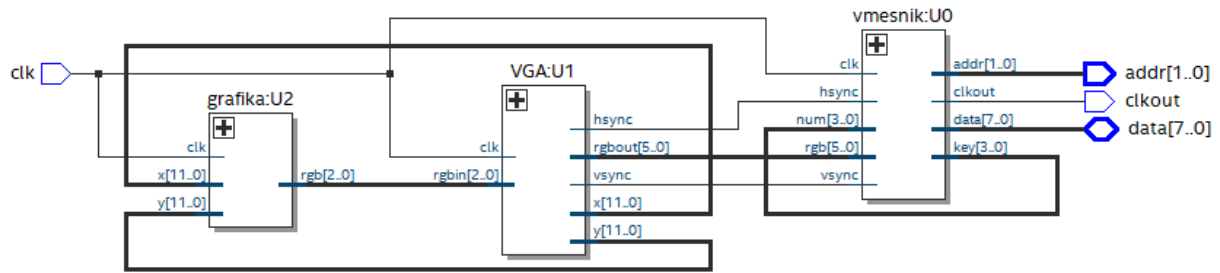
```
constant rom : unsigned(15 downto 0) :=
"1001" &
"0110" &
"0110" &
"1001";
```

Naslov pomnilnika sestavi iz spodnjih 2-bitov transformiranih koordinat **xt** in **yt**:

```
adr <= yt(1 downto 0) & xt(1 downto 0);
```

Preizkusi delovanje na simulaciji, kjer nastavi vrednost vrstice (**y**) in opazuj izhodne točke.

Projekt za razvojno ploščo



1. Odpri projekt [B18vga.qar](#) v programu Quartus. V projektu je digitalni sistem iz treh komponent: vmesnik za razvojno ploščo, generator signalov za monitor VGA in grafika. Generator VGA skrbi za časovni potek prikazovanja slike ločljivosti **800 x 600** točk s 6-bitno barvno globino. Na komponento grafika sta povezani 12-bitni koordinati trenutne točke (**x**, **y**) iz komponente pa pride 3-bitni podatek o barvi točke **rgb** (rdeča, zelena in modra).
2. Uredi komponento grafika, ki vsebuje 1024-bitno konstanto rom za prikaz slike kroga iz 32x32 točk:


```
constant rom: unsigned((32*32)-1 downto 0) := (
  "00000000000000000000000000000000" &
  "00000000001111111111100000000000" &
  "00000000111111111111111000000000" &
```

 Dodaj VHDL proces in stavke, ki iz pomnilnika preberejo vrednost in jo pošljejo na **rgb**, ko sta koordinati **x** in **y** med vključno 0 in 31, sicer pa naj gre na **rgb** vrednost 2. Prevedi projekt in preizkusi delovanje na razvojni plošči.
3. Spremeni logiko za določanje barv, tako da bo celotno ozadje slike modro, prikazan krog pa rumene barve.
4. Dodaj stavke, ki prikažejo premaknjen krog, tako da se začne prikaz na koordinatah (100, 50).
5. Dodaj še stavke, ki prikažejo še en krog v drugačni barvi in na drugem mestu na zaslonu. Razmisli, katere signale je potrebno definirati in kateri ostanejo enaki.