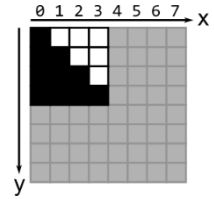


10. vaja: Grafika

Izdelali bomo komponento vezja za prikazovanje sličice iz pomnilnika ROM.

Branje točk iz pomnilnika

Zaradi lažje izvedbe simulacije bomo najprej naredili vezje za prikaz podatkov iz pomnilnika ROM, ki vsebuje 16 eno-bitnih vrednosti. Podatke bomo razdelili na 4 vrstice po 4 bite in bodo predstavljali binarno sliko velikosti 4x4 točke, ki jo bomo postavili na mrežo velikosti 8x8 točk.



Za pomnilnik deklariramo 16-bitni vektor in mu določimo konstantno vrednost. Pomnilnik opisuje stavek, ki izbere eno-bitni izhod (**data**) ob 4-bitnem naslovu (**adr**):

spletno orodje

Deklaracija konstante:

```
rom : u16;
rom = 0b1111011100110001
```

Opis branja ROM:

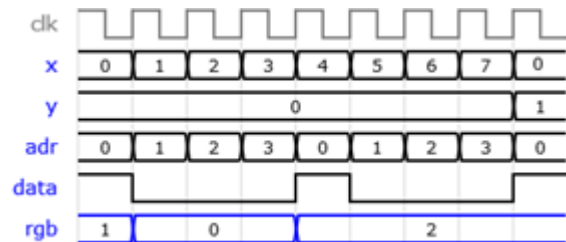
```
data = rom(adr)
```

VHDL

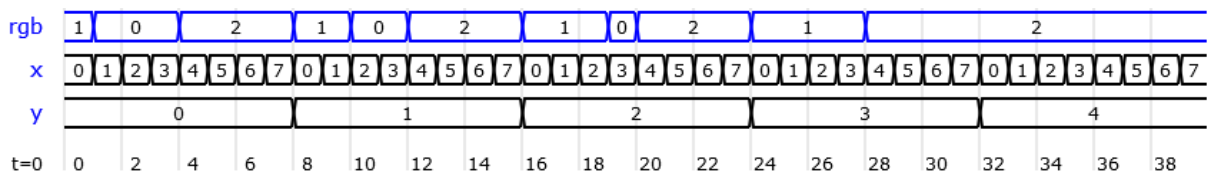
```
constant rom : unsigned(15 downto 0) :=
"1111" &
"0111" &
"0011" &
"0001";
```

```
data <= rom(to_integer(adr));
```

1. Model vezja naj vsebuje 3-bitna števca **x** in **y**. Števec **x** naj se povečuje ob vsakem ciklu ure, števec **y** pa le, kadar je $x = 7$. Opiši vezje s števčema, pomnilnikom in 3-bitnim izhodom **rgb**. Naslov pomnilnika naj določata prva dva bita števca **x**. Kadar je števec manjši od 4, naj **rgb** dobi vrednost data, sicer pa vrednost 2.

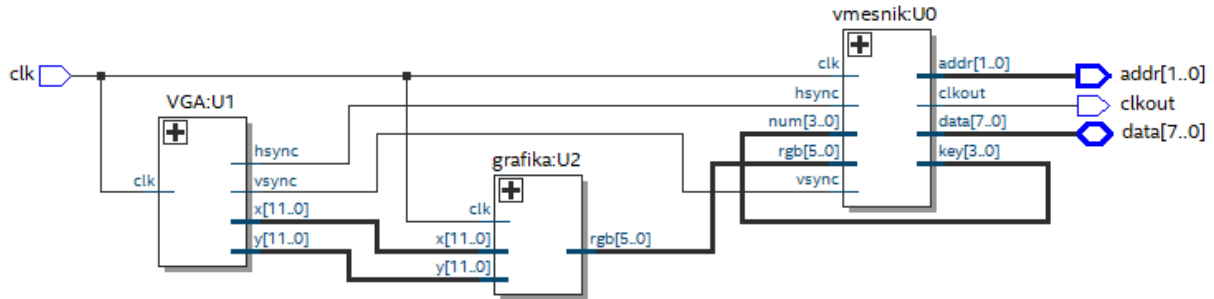


2. Na izhod **rgb** pošiljamo zaporedne točke, najprej iz prve vrstice, nato druge, itd. Ugotovi kako je potrebno naslavljanje pomnilnika, da bomo prebrali celo sliko:



3. Ugotovi, kaj je potrebno spremeniti v kodi, da bi bila sličica iz pomnilnika ROM zamaknjena za dve mesti v desno. (namig: spremeni naslov in pogoje za izhod).

Grafika na razvojni plošči



1. V Quartusu odpri projekt z generatorjem slike in dodaj vezje [grafika.vhd](#) (naloži in prenesi datoteko v projektno mapo, odpri in izberi *Project > Add Current File to Project*). Model vezja vsebuje 1024-bitno konstanto **rom** za prikaz slike kroga iz 32x32 točk:

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

V **grafika.vhd** dodaj stavke, ki iz pomnilnika preberejo vrednost na koordinatah (**x**, **y**) in nastavijo na **rgb** na "111100", ko je prebrana 1 in sta koordinati **x** in **y** med vključno 0 in 31, sicer pa naj bo **rgb** enak 0.

2. Posodobi glavno datoteko **sistem.vhd**, da bo vsebovala komponent grafika in bo povezana tako, kot prikazuje blokovni diagram. Izbriši stavek za prirejanje **rgb**, ker bo sedaj barvo določala komponenta grafika.
3. Dodaj stavke, ki prikažejo premaknjen krog, tako da se začne prikaz na koordinatah (100, 50).
4. *Dodaj 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.