

8. vaja: Prikaz besedila

Oblike znakov so zapisane v pomnilniku kot sličice velikosti 8 x 8 točk. V komponenti CROM je pomnilnik s 64 sličicami črk, ločil in števil, ki predstavljajo znake v tabeli ASCII s kodami med 0x20 in 0x5F. Komponenta ima 7-bitni vhod **ascii** za kodo znaka in trenutni koordinati (**cx**, **cy**). Izhod je enobitni signal **pix**, ki ga uporabimo za prikazovanje točk.



8.1 Prikaz vrstice besedila

Dodaj v vezje VGAtest komponento CROM in jo poveži z notranjimi signali. Deklarirati bo potrebno dva nova signala: 7-bitni standardni vektor **ascii** in enobitni signal **pix**. Deklariraj tudi podatkovni tip in signal za zbirko 8-bitnih vrednosti, v kateri bo shranjena vrstica besedila. Npr. "DES 2017" zapišemo s ASCII kodami v šestnajstičkem zapisu:

D	E	S		2	0	1	7
0x44	0x45	0x53	0x20	0x32	0x30	0x31	0x37

```
type vrstica is array(0 to 7) of std_logic_vector(7 downto 0);
signal reg: vrstica := (X"44", X"45", X"53", X"20", X"32", X"30", X"31", X"37");
```

Dodaj v vezje logiko za prikaz vrstice besedila na vrhu slike. Celotno besedilo zasede 64 x 8 točk in naj bo izpisano med koordinatami [0,0] in (64, 8). Najprej je potrebno iz zbirke **reg** prebrati kodo znaka, ki se trenutno prikazuje in ga prenesti v 7-bitni signal **ascii**. Naslov trenutnega znaka določimo glede na koordinato **cx**: med 0 in 7 beremo znak **reg(0)**, med 8 in 15 beremo znak **reg(1)**, med 16 in 23 **reg(2)**...

Kako bi to napisal na čim krajši način?

Iz pomnilnika CROM dobimo **pix**, ki določa izhodno barvo:

- če je en='1':
 - če smo znotraj besedila (cx<64 in cy<8)
 - če je pix='1': rgb3="111"
 - sicer: rgb3="000"
 - sicer naj bo izhod enak barvi ozadja
- sicer: rgb3="000"

Preizkusi delovanje prikaza besedila na simulaciji s testno strukturo iz prejšnje vaje.