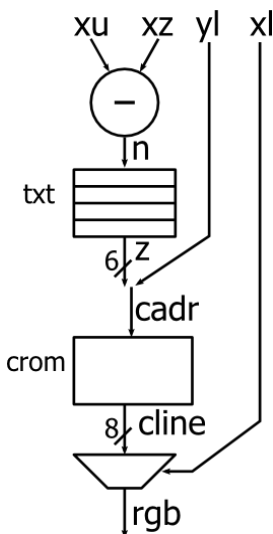
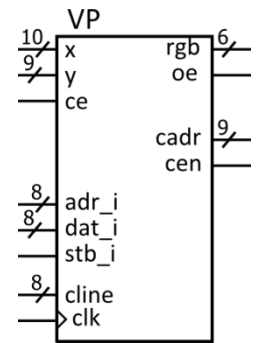


7. Vaja: Video procesor: besedilo

Besedilo je sestavljeno iz 6-bitnih kod znakov, ki so zapisani v zaporednih lokacijah pomnilnika txt(). Pomnilnik deklariramo kot zbirko 32 besed in nastavimo nekaj začetnih vrednosti:

```
type regs is array (0 to 31) of unsigned(5 downto 0);
```

```
signal txt: regs := ("100100", "100101", "110011", "010010", "010000", "010001",  
"010100", others=>"000000");
```



Notranji signali:

alias xu: **unsigned**(6 downto 0) **is** x(9 downto 3);

alias xl: **unsigned**(2 downto 0) **is** x(2 downto 0);

alias yu: **unsigned**(5 downto 0) **is** y(8 downto 3);

alias yl: **unsigned**(2 downto 0) **is** y(2 downto 0);

z: **unsigned** (5 downto 0); -- koda in

n: **unsigned** (6 downto 0); -- indeks znaka

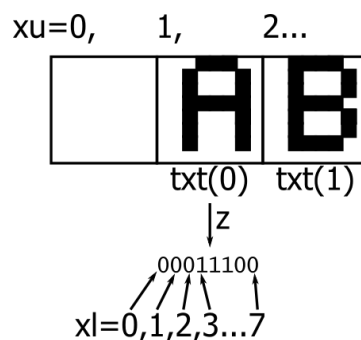
in registri:

xz: **unsigned** (6 downto 0) := "0000001"; -- začetna koordinata x

yz: **unsigned** (5 downto 0) := "000000"; -- začetna koordinata y

num: **unsigned** (4 downto 0) := "00111"; -- število znakov za izpis

Položaj znakov na zaslonu določa začetna koordinata (xz, yz), ki je koordinata kvadratov velikosti 8x8 točk. Koordinate točk na zaslonu (x,y) razdelimo na zgornji podvektor (xu, yu) in spodnji podvektor (xl, yl). Zgornji podvektor določa položaj znaka na zaslonu, spodnji pa točke znotraj znaka.



Izpisovanje poteka tako, da določimo indeks znaka v pomnilniku txt(), $n = xu - xz$ in iz pomnilnika preberemo kodo znaka z. Kodo znaka uporabimo za branje vrstic (cline) iz znakovnega pomnilnika crom(). V znakovnem pomnilniku je vsak znak zapisan v osmih vrsticah, zato je naslov pomnilnika sestavljen iz kode znaka in spodnjih 3 bitov koordinate y (podvektor yl). Prebrano vrstico cline razdelimo glede na spodnje 3 bite koordinate x (podvektor xl) na posamezne bite, ki naj določajo barvo izhodne točke (rgb).

Napiši še logiko za izhodni signal *oe*, ki je postavljen na 1, kadar so koordinate (*xu,yu*) znotraj območja risanja besedila. Območje je določeno z začetno koordinato (*xz,yz*) in številom znakov (*num*). Izhod *oe* pa naj bo aktiven takrat, ko komponenta bere iz pomnilnika *crom()*. Pri kombinacijski izvedbi vezja sta signala *oe* in *oe* lahko enaka.

Naredi opis komponente video procesor in preizkusi delovanje s pripravljeno testno strukturo.

Dodaj v opis vezja sekvenčno logiko, ki omogoča spreminjanje vsebine pomnilnika in nastavljanje registrov preko vhodnega vodila Wishbone (*adr_i*, *dat_i*, *stb_i*). Vpis podatka poteka iz vodila *dat_i* ob uri in pogoju *stb_i=1*. Pomnilnik in registri so na naslovih:

<u>adr_i</u>	<u>register</u>
0000 0001	xz
0000 0010	yz
0000 0011	num
001n nnnn	txt(n)