Univerza v Ljubljani
Fakulteta za elektrotehniko

V sistem bomo dodali sličico žoge, ki se bo premikala po zaslonu in zaznala trk z objektom.

3 Prikaz sličice

Za prikaz sličice bomo v projekt iz prejšnje vaje dodali paket sprites.vhd, v katerem je deklarirana sličica žoge velikosti 30 x 30 točk.

```
type slika30x30 is array (0 to 29) of std_logic_unsigned(29 downto 0);
constant zoga: slika30x30 := (
"00000000000000000000000000000000",
"00000000001111111111000000000000",
"00000000111111111111110000000000",
"00000001111111111111111000000000",
...

"00000000001111111111000000000000" );
```

a. Prenesi datoteko sprites.vhd v projektno mapo, odpri projekt in dodaj datoteko (npr. tako da jo v Quartusu odpreš in izbereš **Project > Add current file to project**).
V datoteki **sistem.vhd** dodaj na vrhu opisa paket:

```
library work;
use work.sprites.all;
```

Naredi deklaracije signalov za koordinato žoge, vrstico in barvo:

- 12-bitni nepredznačen vektor **xz** z začetno vrednostjo 100 in 12-bitni **yz** z začetno vrednostjo 5,
- 30-bitni nepredznačen vektor **vrst**
- 1-bitni signal **colorz**

b. Zapiši stavek, ki prebere eno vrstico pomnilnika **zoga** iz naslova, ki ga določa razlika **y** koordinat trenutne točke in koordinate žoge. Vrstico beremo iz zbirke, kadar se nahajamo znotraj koordinat žoge, sicer pa jo postavimo na 0:

```
vrst <= zoga(to_integer(y-yz)) when y>=yz and y<yz+30 else (others=>'0');
```

Vrstica ima 30 bitov in pri določanju izhodne barve moramo izluščiti ustrezen bit s pomočjo razlike **x** koordinat.
Zapiši pogojni stavek, ki nastavi signal **colorz** z vpogledom v vrstico kadar je **x** koordinata znotraj koordinat žoge (x>=xz in x<xz+30), sicer pa postavi **colorz** na 0.

```
colorz <= vrst(to_integer(x-xz));
```

c. Določi še izhodne barve s spremembo izbirnega stavka za signal **rgb.** Spremeni pogoje za prikaz barv znotraj vidnega dela slike: **x**<800 in **y**<600, tako da bo prikazana
barva "001100" pri **colorz='1',** barva "111100" pri **color='1'** in
bela barva na levem, zgornjem in spodnjem robu slike v širini 5 točk (npr. pogoj za zgornji rob je y<5),
v ostalih primerih pa naj bo prikazana barva ozadja "101011".

4 Premikanje sličice

Omeji premikanje pravokotnika na dnu zaslona, tako da se bo gibal le znotraj zaslona in izvedi premikanje žoge v vertikalni smeri.

a. Ugotovi kako je potrebno postaviti pogoje ob spreminjanju koordiat **xp** in **yp,** da bo gibanje pravokotnika omejeno.

b. Dodaj stavke premikanje žogice v smeri y. Deklariraj enobitni signal **smerz,** ki bo določal smer gibanja. Uporabi delilnik ure iz prejšnje vaje in povečuj ali zmanjšuj števec **yz** ob prelivu delilnika in določeni smeri gibanja:
   - navzdol: **yz** naj se povečuje, kadar je **smerz='0'** in je vrednost števca manjša od 600
   - navzgor: **yz** naj se zmanjšuje, kadar je **smerz='1'** in je vrednost števca večja od 5

   Ko pride žogica do spodnje ali zgornje meje, spremeni smer, tako da se bo odbijala gor in dol.

c. Dodaj enobitni signal **trkz** za zaznavanje trka med žogico in pravokotnikom na dnu zaslona. Trk lahko zaznamo tako, da ugotovimo ali sta v nekem trenutku signal **color** in **colorz** oba na '1', kar pomeni da bo na istem mestu slike prikazana točka žoge in točka pravokotnika. V tem primeru postavimo signal **trkz** na '1'.

   V opisu števca **yz** lahko sedaj uporabimo informacijo o trku. Če je ta signal postavljen na '1' in se žoga giblje navzdol, spremenimo smer gibanja in takrat tudi postavimo signal **trkz** nazaj na '0'.