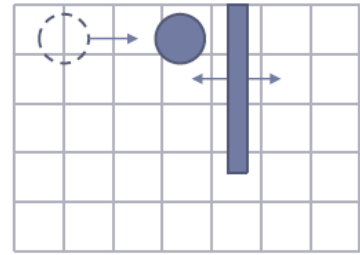


13. vaja: Interaktivni sistem

Grafičnemu sistemu s procesorjem bomo dodali premikajoči zid (pravokotnik) in zaznavanje trka med grafičnimi objekti.



Grafika

1. V komponento grafika dodaj pravokotnik velikosti 10 x 100 točk, z izhodiščem na (x,y)=(200,10). Deklariraj dva signala za transformirane koordinate (npr. **xt2**, **yt2**) in enobitni signal **obj**, ki naj bo '1', kadar smo znotraj objekta (when $xt2 < 10$ and $yt2 < 100$).
2. V proces za določanje izhodne barve dodaj pogoj, da je barva bela, kadar smo znotraj objekta. Ugotovi, na katero mesto v pogojnih stavkih bi zapisal dodaten pogoj in preveri delovanje na razvojni plošči.

```

if x<800 and y<600 then -- vidni del slike
  if xt<32 and yt<32 and data='1' then
    rgb <= "111100";
  else
    if x(4 downto 0)="11111" or
       y(4 downto 0)="11111" then
      rgb <= "001111";
    else
      rgb <= "010101";
    end if;
  end if;
else -- zatemnjen del slike
  rgb <= "000000";
end if;

```

```

if obj='1' then
  rgb <= "111111";

```



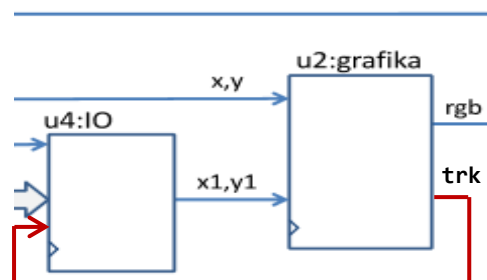
3. Dodaj deklaracijo dveh nepredznačenih signalov: 24 bitni delilnik ure (**d**) in 6-bitni števec (**cnt**). Iz glavne datoteke kopiraj proces za opis delilnika, ki naj namesto določanja signala **ce** povečuje števec **cnt**. Delilnik naj šteje po modulu 1 000 000. Vrednost števca prištej k transformirani koordinati **xt2** in preizkusi delovanje.

```

p: process(clk)
begin
  if rising_edge(clk) then
    if d=1000000 then
      d <= (others=>'0');
      cnt <= cnt+1;
    else
      d <= d + 1;
    end if;
  end if;
end process;

```

4. Dodaj enobitni izhodni signal **trk**, ki ga nastavljamo v procesu za določanje barv. Trk naj se postavi na '1', kadar sta hkrati izpolnjena pogoja za prikaz sličice kroga in zidu (**obj**='1'), sicer pa naj ima privzeto vrednost '0'. Ta signal poveži v glavni datoteki (**sistem.vhd**) s komponento **io**, kjer naj bo deklariran kot vhod.



Vhodno-izhodni vmesnik

V vhodno izhodni vmesnik bomo dodali logiko zaznavanja trka. Signal **trk** je odvisen od števecv koordinat (x,y), ki se spreminjajo s frekvenco 50 MHz in je na '1' kratek čas med osveževanjem slike. Naredili bomo prekinitveni register (**int**), ki se bo postavil na '1', ko zazna trk, na '0' pa šele takrat, ko procesor prebere njegovo vrednost.

5. V komponenti **io.vhd** deklariraj enobitni vhodni signal **trk** in enobitni notranji signal **int**, ki naj se postavi na '1' ob uri in pogoju **trk='1'**.
6. Signal **int** pošljemo na procesor (vodilo **datout**) ob branju iz naslova 5, na enak način, kot je narejeno branje tipk. Ko se vrednost prenese na **datout**, naj se tudi resetira.
7. Dodaj še izhodni priključek **status**, na katerega naj bo povezan signal **int**. V glavni datoteki deklariraj med priključki 8-bitni izhodni vektor **led** in poveži en bit (npr. **led(0)**) na **status**. Prevedi sistem in preveri, ali ob trku zasveti LED.

Program

1. Dopolni program učnega procesorja tako, da bo v glavni zanki preverjal ali je prišlo do trka. Ko zazna prekinitvev (na naslovu 5 prebere vrednost različno od 0), naj skoči na prvi ukaz, kar bo povzročilo premik sličice na začetno lokacijo.
2. Razmisli, kako bi naredil program za štetje zaznanih trkov. Vrednost števca bi lahko v programu poslal na nov izhodni register (dopolni vmesnik **io.vhd**) in na LED.