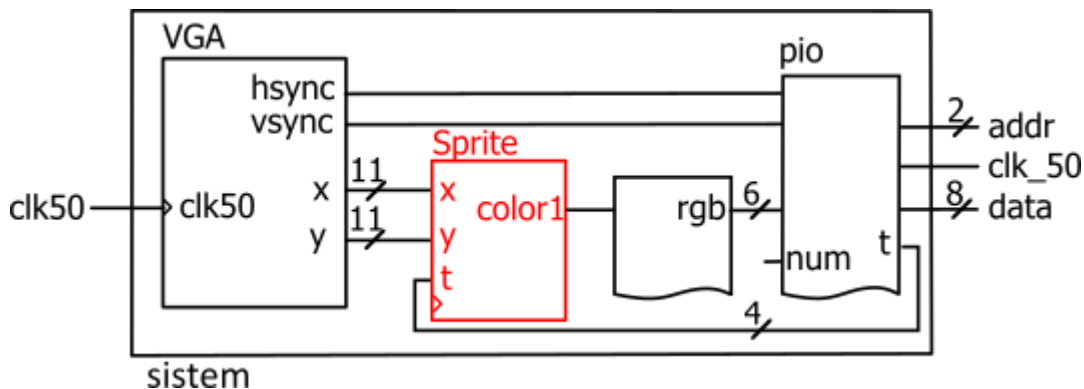




2. vaja: Grafika s sličicami

Naredili bomo grafični krmilnik za prikaz slike, ki je sestavljen iz gibljivih sličic (angl. *sprite*). Za demonstracijo bomo naredili komponento za prikaz ene sličice, ki jo lahko premikamo po zaslonu.

Gibljiva sličica



Naloga1

Naredili bomo novo vezje `sprite.vhd`, ki bo kot komponenta vključeno v VGA krmilnik. Zunanji priključki:

```
entity sprite is
port (clk50 : in std_logic;           -- ura
      x,y : in unsigned(10 downto 0); -- koordinate trenutne točke
      t: in unsigned(3 downto 0);     -- tipke
      color1: out std_logic);        -- barva trenutne točke
end sprite;
```

-- V komponenti `Sprite.vhd` definiraj ROM z bitno sliko velikosti 16 x 16 točk, ki predstavlja npr. sličico žoge.

-- Deklariraj tudi dva 11-bitna notranja signala `x1` in `y1`.

```
type slika is array (0 to 15) of std_logic_vector(0 to 15);
```

```
constant zoga: slika :=
( "0000001111000000",
  "0000111111110000",
  "0001111111111000",
  "0011111111111100",
  "0111111111111110",
  "0111111111111110",
  "1111111111111111",
  "1111111111111111",
  "1111111111111111",
  "1111111111111111",
  "1111111111111111",
  "0111111111111110",
  "0111111111111110",
  "0011111111111100",
  "0001111111111000",
  "0000111111110000",
  "0000001111000000" );
```

- a. Napiši proces, ki primerja vrednost koordinat **x** in **y** s koordinatami **x1** in **y1** in če je znotraj okvirja sličice žoge določi izhod **color1**, sicer ga pa postavi na '0':

```
if x>=x1 and x<x1+16 and y>=y1 and y<y1+16 then
  color1 <= zoga(to_integer(y-y1))(to_integer(x-x1));
else
  color1 <= '0';
end if;
```

- b. Napiši kodo za spreminjanje koordinat **x1** in **y1** ob pritisnjenih vhodnih tipkah. Dve tipki (npr. t(0) in t(1)) uporabi za povečanje koordinate **x1** oz. **y1**, drugi dve pa za zmanjšanje koordinate. V vezju dodaj ustrezen delilnik ure, da se bodo koordinat spreminjale s primerno frekvenco.
- c. V datoteki **sistem.vhd** deklariraj vezje Sprite kot komponento, deklariraj notgranji signal **color1** in poveži signale.

```
cspr: sprite port map (clk50, x, y, t, color1);
```

- d. Dopolni pogoj za izhodni signal **rgb** in preizkusi delovanje vezja na razvojnem sistemu.
- o če smo znotraj vidnega dela slike (hst<800, vst<600) naj bo
 - **rgb** <= "001100" če je color1='1', sicer
 - **rgb** <= "111100" če je color='1', sicer
 - **rgb** <= "000011"
 - o izven vidnega dela slike naj bo **rgb** <= "000000"

Naloga2

- a. V datoteki **sprite.vhd** zapiši pogoje, ki omejijo premikanje žoge, tako da pride izven zaslona.