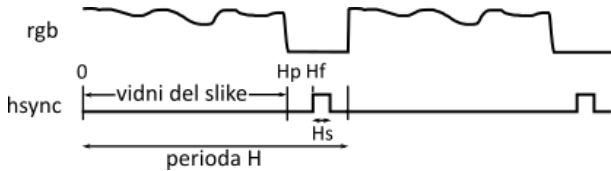


## 5. vaja: Generator slike na razvojni plošči

Slika se na monitorju izrisuje po vrsticah. Na koncu vsake vrstice pošljemo monitorju sinhronizacijski impulz **hsync**. V času sinhronizacije mora biti barvni izhod **rgb** na 0 (zatemnilna perioda). Časovni potek signalov določa standard VGA:



VGA 800x600	število period	število vrstic
perioda	H = 1040	V = 666
vidni del	Hp = 800	Vp = 600
začetek pulza	Hf = 856	Vf = 637
trajanje pulza	Hs = 120	Vs = 6

Ko so prikazane vse vrstice slike, naredimo vertikalno sinhronizacijo (**vsync**).

### VGA komponenta



Naredili bomo komponento za prikaz slike ločljivosti 800 x 600 točk in frekvence osveževanja 72 Hz. Parametri signalov **hsync** in **vsync** so v tabeli podani kot število period ure **clk50** (50 MHz) oz. števila vrstic.

V orodju Vivado naredi nov projekt z imenom VGA in izberi razvojno ploščo Zedboard. Ustvari novo VHDL datoteko ter določi vhodne in izhodne priključke.

V opisu vezja naredi dva števec: **hst** (števec točk, 11-bitni unsigned) in **vst** (števec vrstic, 10-bitni unsigned). Števec **hst** se povečuje s periodo ure in šteje do H-1, kolikor je perioda vrstice. Ko ga resetiramo na 0, povečamo števec vrstic, ki naj šteje od 0 do V-1.

```

if rising_edge(clk50) then
  if hst < H-1 then
    hst <= hst + 1;
  else
    hst <= (others=>'0');
    if vst < V-1 then
      vst <= vst + 1;
    else
      vst <= (others=>'0');
    
```

Števca uporabimo za določanje koordinate trenutne točke, kar bomo potrebovali pri risanju grafike. Koordinate prenesemo v 10-bitna izhodna signala **cx** in **cy**, ki naj bosta tako kot števec podatkovnega tipa **unsigned**.

```

cx <= hst(9 downto 0);
cy <= vst;

```

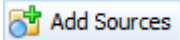
Izhod **rgb** postavi na konstantno vrednost (npr. "010"), ko sta števec **hst** in **vst** znotraj vidnega dela slike (800, 600) in takrat naj bo izhodni signal **en** na '1'. Ko sta števec izven vidnega dela, pa naj bosta **rgb** in **en** na 0.

Sinhronizacijske izhodne impulze dobimo s kombinacijsko primerjavo vrednosti števecov, npr. horizontalni impulz: `hsync <= '1' when hst>=Hf and hst<Hf+Hs else '0';`

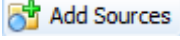
Preveri delovanje komponente na simulatorju, kjer nastavi uro periode 20 ns in izvedi simulacijo za 15 ms.

## Preizkus na razvojni plošči

Za preizkus na razvojni plošči bomo vključili vezje VGA kot komponento v vezje VGAtest, kjer je opisan delilnik ure in povezave na izhodne signale.

Vključi v projekt predlogo vezja VGAtest:  *Next, Add Files...* Dopolni opis vezja z deklaracijo priključkov komponente VGA in dokončaj stavek **port map**. Signali **cx**, **cy** in **en** v testnem vezju niso povezani (**open**).

```
34 entity VGAtest is
35     Port ( clk : in STD_LOGIC;
36           hsync, vsync : out STD_LOGIC;
37           rgb_o : out STD_LOGIC_VECTOR (7 downto 0));
38 end VGAtest;
39
40 architecture Behavioral of VGAtest is
41     component VGA is
42     port( -- dopolni
43         );
44     end component;
45
46     signal clk50: std_logic;
47     signal rgb3: unsigned(2 downto 0);
48 begin
49
50 p: process(clk)
51 begin
52     if rising_edge(clk) then
53         clk50 <= not clk50;
54     end if;
55 end process;
56
57 ul: VGA port map (clk50=>clk50, hsync=>hsync ... -- dopolni
58
59 rgb_o(7 downto 6) <= rgb3(2) & rgb3(2);
60 rgb_o(5 downto 3) <= rgb3(1) & rgb3(1) & rgb3(1);
61 rgb_o(2 downto 0) <= rgb3(0) & rgb3(0) & rgb3(0);
```

Pred prevajanjem vezja dodaj še datoteko z definicijami priključkov:  *Add or Create Constraints, Next, Add Files...* **VGA.xdc**

Izvedi sintezo in implementacijo vezja, po prevajanju pa izdelaj programsko datoteko (*Generate Bitstream*). Za prenos datoteke na razvojno ploščo uporabi *Hardware Manager*.