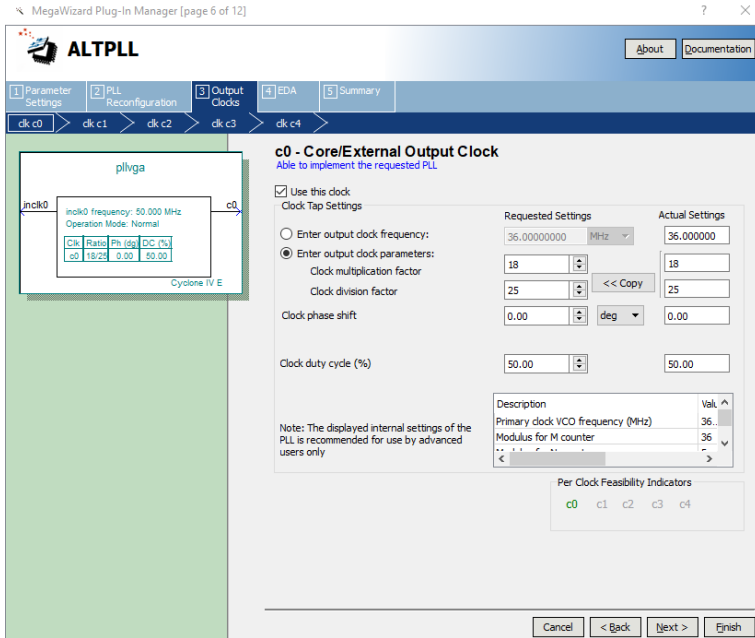


— 12 Generování VGA videosignálu



Další hodinové výstupy nebudeme potřebovat, nepotřebujeme ani specifický fázový posun nebo duty cycle, takže můžeme klidně použít Finish. Wizard vygeneruje několik souborů a vloží je do projektu (popř. se zeptá, jestli to má udělat).

12.5 Kalkulačka!

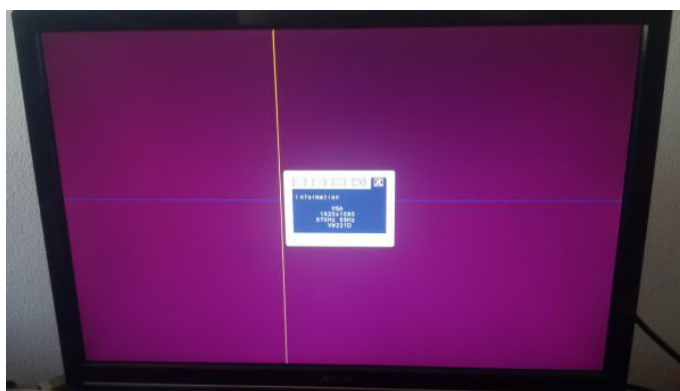
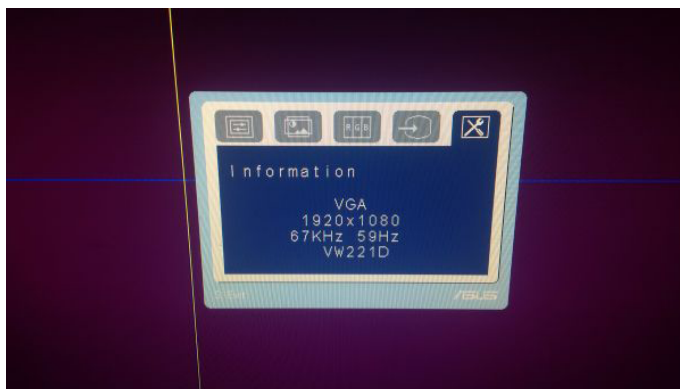
Jestli se vám motá hlava z nadměry konstant a frekvencí, tak nezufojte! Připravil jsem pro vás kalkulátor frekvencí pro VGA, PLL a vůbec všechno to, co jsme si teď říkali. Stačí zadat frekvenci krystalu, kterou máte k dispozici, a kliknout na Go!

V tabulce se objeví přehledně všechna VGA rozlišení, která lze z této hodinové frekvence získat. K nim všechny potřebné konstanty, a dokonce i nastavení PLL (násobitel a dělitel). Kalkulačka vždy počítá frekvence tak, aby násobitel a dělitel byla celá čísla. Jejich velikost si můžete omezit v poli *PLL max div* (např. pro jednodušší PLL).

Když si vyberete vhodné rozlišení a kliknete na něj, tak vám kalkulačka vygeneruje vylepšenou komponentu sync.vhd (viz výše) a ukáže aktuální hodnoty průběhu signálů.

<https://datacipy.cz/vga>

Při svých testech jsem se odvázně pouštěl dál a dál, a nakonec jsem použil rozlišení 1920×1080@60 Hz, tedy Full HD. Pixelová frekvence je 148,5 MHz, PLL koeficient 99/32 (a opravdu, $48 * 99 / 32 = 148,5$). S trochou rozechvění jsem nahrával kód do FPGA, a po několika sekundách se na monitoru objevilo:



Jak se říká: *Bylo to tam!*

12.6 Jednoduchý obrazec

Synchronizace je nezbytný základ, je to kostra, na který se navěsí vlastní zobrazování. Vylepšená komponenta sync posílá kromě synchronizačních pulsů a signálu blank i dvě čísla, totiž posy a posy, neboli pozici horizontálně a pozici vertikálně. Funkce je následující: Když je zatmění (blank), tyto čísla ignorujte a na výstupy R,G,B posílejte 0. Pokud není zatmění, posílejte na výstupy barvu pixelu na dané souřadnici.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

ENTITY VGA IS

port (
  clock50: in std_logic;
  vga_hs,vga_vs: out std_logic;
  vga_r: out std_logic;
  vga_g : out std_logic;
  vga_b: out std_logic
);

END VGA;

architecture main of vga is

signal vgaclk:std_logic:='0';
signal x:integer range 0 to 1919;
signal y:integer range 0 to 1439;
signal blank: std_logic;

-----
  component PLL1 is
    port (
      inclk0      : IN STD_LOGIC := '0';
      c0         : OUT STD_LOGIC
    );
  end component PLL1;
-----
  component sync is
  port (
  clk: in std_logic;
  posx:out integer range 0 to 1919;
  posy:out integer range 0 to 1439;
  hsync,vsync: out std_logic;
  blank: out std_logic
  );
end component sync;
```