ENGINE

Teaching online electronics, microcontrollers and programming in Higher Education

Hardware Implementation of Algorithms

8. Video signal generator (VGA interface).

Lead Partner: Warsaw University of Technology

Autor: Lukasz Mik

University of Applied Sciences in Tarnow

Erasmus+

GRANT NUMBER: 2020-1-PL01-KA226-HE-095653

This laboratory instruction has been prepared in the context of the ENGINE project. Where other published and unpublished source materials have been used, these have been acknowledged.

Copyright

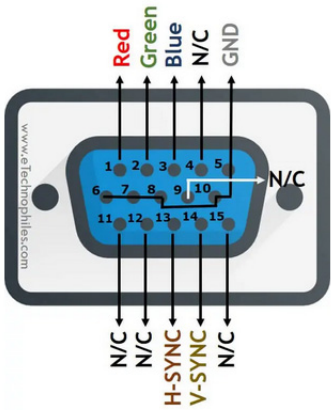
Funding Disclaimer

This project has been funded with support from the European Commission. This report reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

I. VGA interface.

The VGA (Video Graphics Array) standard is a graphics card standard established in 1987 by IBM, which applied it to the IBM PS/2 series of computers. Along with the cards appeared a new DE-15 monitor connector, popularly known as VGA, which is still supported by computer and monitor manufacturers. This is a standard related to analog signals processed by a computer monitor. Screen resolutions and signal parameters are strictly defined and published by the VESA organization (www.vesa.org). The example discussed in this manual generates signals with parameters compliant with the VESA standard for a screen resolution of 640x480 pixels.

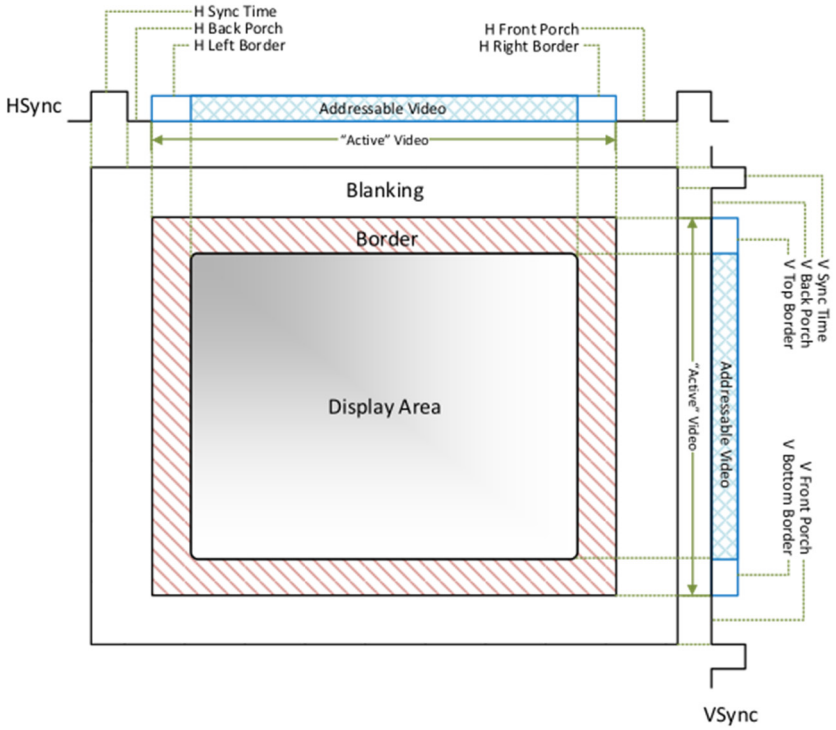
The DE-15 connector is a 3-row D-Sub connector, in which 5 pins are used to generate the VGA signal, responsible for the signals: Red, Green, Blue, H-Sync (Horizontal Synchronization) , and V-Sync (Vertical Synchronization). Red, Green, and Blue are the three analog signals that determine the color of a point on the screen, while H-Sync and V-Sync provide a reference to where the point should appear on the screen. By properly controlling these five signals in accordance with the VESA VGA timing specification, we can display anything we want on any monitor.



VGA Pinout

The horizontal resolution is responsible for the number of pixels per line (one line), while the vertical resolution - for the number of lines on the screen. The address of each pixel on the screen is determined by the PCLK (Pixel Clock) frequency. The VGA controller must generate H-Sync and V-Sync timing signals and coordinate the delivery of video data based on PCLK. The V-Sync signal is responsible for the screen refresh rate.

The scheme of operation of the VGA controller on the example of a screen with a resolution of 640 x 480 pixels is shown in the figure below.



The table below presents the time parameters of the signals used to synchronize the image on the monitor screen.

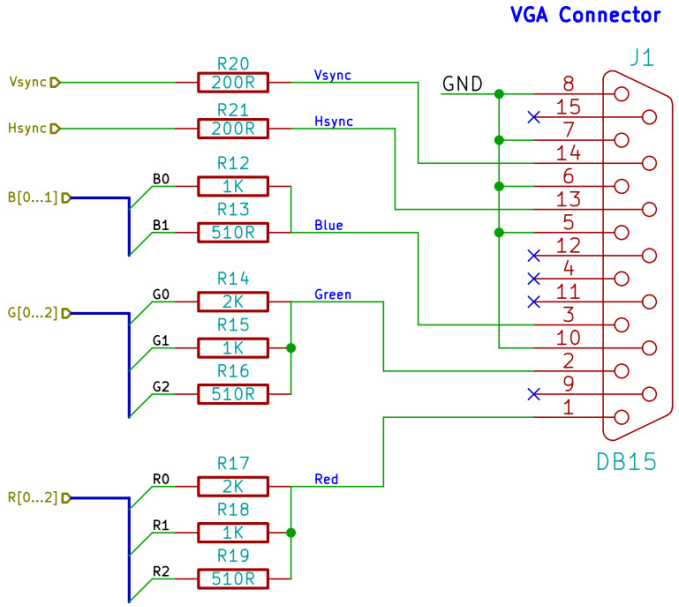
Opis	Oznaczenie	Czas	Szerokość impulsu/częstotliwość
Pixel Clock	tclk	39.7 ns (± 0.5%)	25.175MHz
Hor Sync Time	ths	3.813 µs	96 Pikseli
Hor Back Porch	thbp	1.907 µs	48 Pikseli
Hor Front Porch	thfp	0.636 µs	16 Pikseli
Hor Addr Video Time	thaddr	25.422 µs	640 Pikseli
Hor L/R Border	thbd	0 µs	0 Pikseli
V Sync Time	tvs	0.064 ms	2 Linie
V Back Porch	tvbp	1.048 ms	33 Linie
V Front Porch	tvfp	0.318 ms	10 Linii
V Addr Video Time	tvaddr	15.253 ms	480 Linii
V T/B Border	tvbd	0 ms	0 Linii

Additional delay times, called porches, for horizontal and vertical synchronization signals are a remnant of the control method of analog CRT monitors, where a spot on the screen was drawn by a stream of electrons on a phosphor. Since the VGA interface is of the analog type, it is necessary to keep all the parameters in the table, which were determined based on the VESA requirements.

II. Video signal generator in VHDL language.

The VGA controller must generate several signals according to the specifications in the table above. According to the specifications, the pixel clock should be 25.175 MHz. Much simpler in an FPGA is to generate a 25 MHz clock using a digital clock signal manager (DCM). Variation at the level of ±0.5% in clock frequency is acceptable for most monitors. Additionally, it is necessary to implement 2 counters responsible for counting positions on the screen. One counter, responsible for counting pixels in a line, must be reset when it reaches 799. When the pixel counter is reset, the line counter is incremented by 1 (going to a new line). Similarly, the vertical line counter must reset to zero when it reaches the end of the frame (the last line of the image). The next figure shows how to generate H-Sync and V-Sync signals based on counter values.

The Numato Elbert V2 evaluation board with the Spartan-3A chip has a VGA interface with a limited number of colors that can be generated. If we look at the interface diagram, we can see simple DACs for each color component, composed of resistor ladders.



For green and red it is a 3-bit converter, while for blue it is a 2-bit one. Resistors on the Hsync and Vsync lines are designed to limit the current flowing in the circuits generating the synchronization pulses on the monitor side.

Taking into account the number of bits for each color component, we can generate a maximum of $2^3 * 2^3 * 2^2 = 256$ colors for each pixel displayed on the monitor screen. If we wanted to display pictures, we would have to reduce the number of colors to 256 bits, taking into account the number of bits for its individual components. This will be the topic of the next lessons.

During this class, we will generate frames of a monochrome image, i.e. 2 colors will be displayed - white and black. White color can be obtained by giving all lines B[0..1], G[0..2] and R[0..2] high states, i.e. 3.3V.

Before starting the exercises, download the source files and save them in folders with appropriate names on the local disk of your computer.

1) Drawing a circle on the monitor screen.

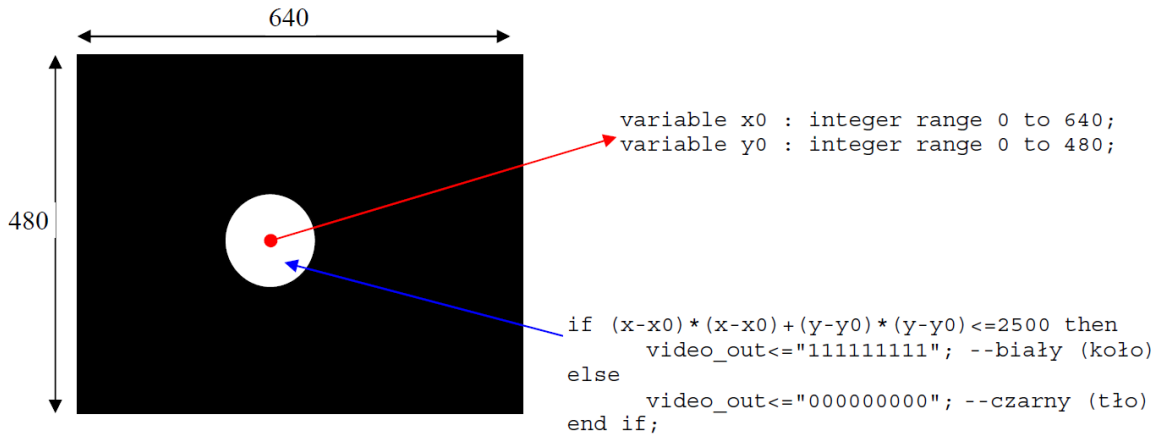
Drawing a circle on the screen is done using the formula describing a circle in the Cartesian coordinate system:

$$(x - x_0)^2 + (y - y_0)^2 \leq r^2$$

where:

$r > 0$ – radius of circle

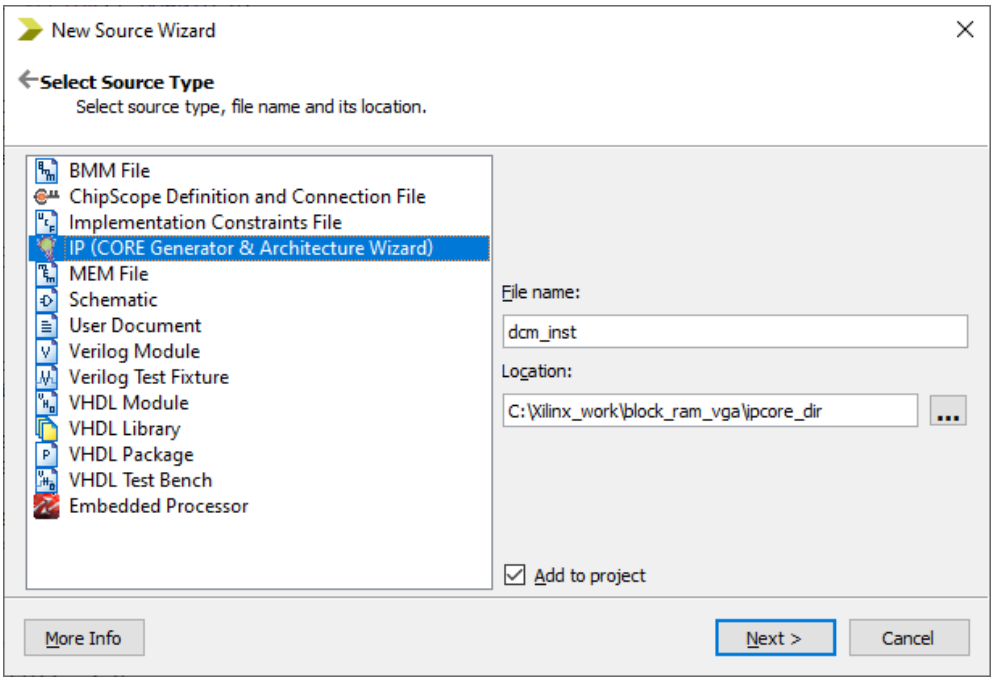
(x_0, y_0) – coordinates of circle center

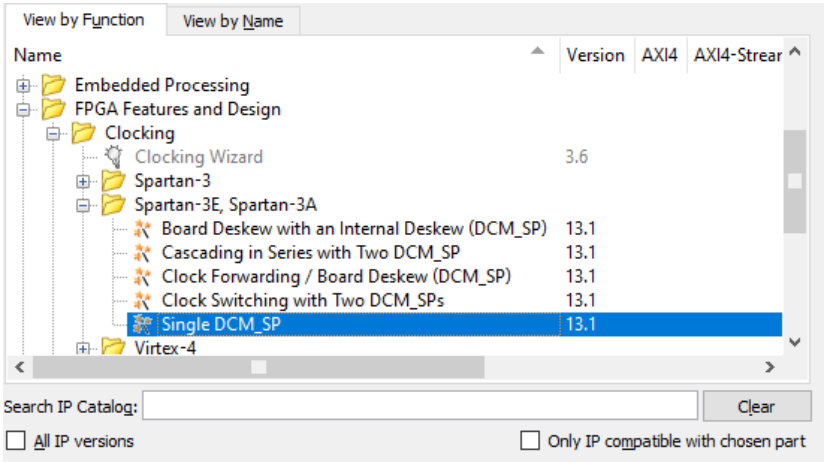


Launch ISE Design Suite 14.7, from the File menu select New Project. Enter the appropriate name of the project, e.g. *circle_vga*, and in the next window set the following target layout for the created project:

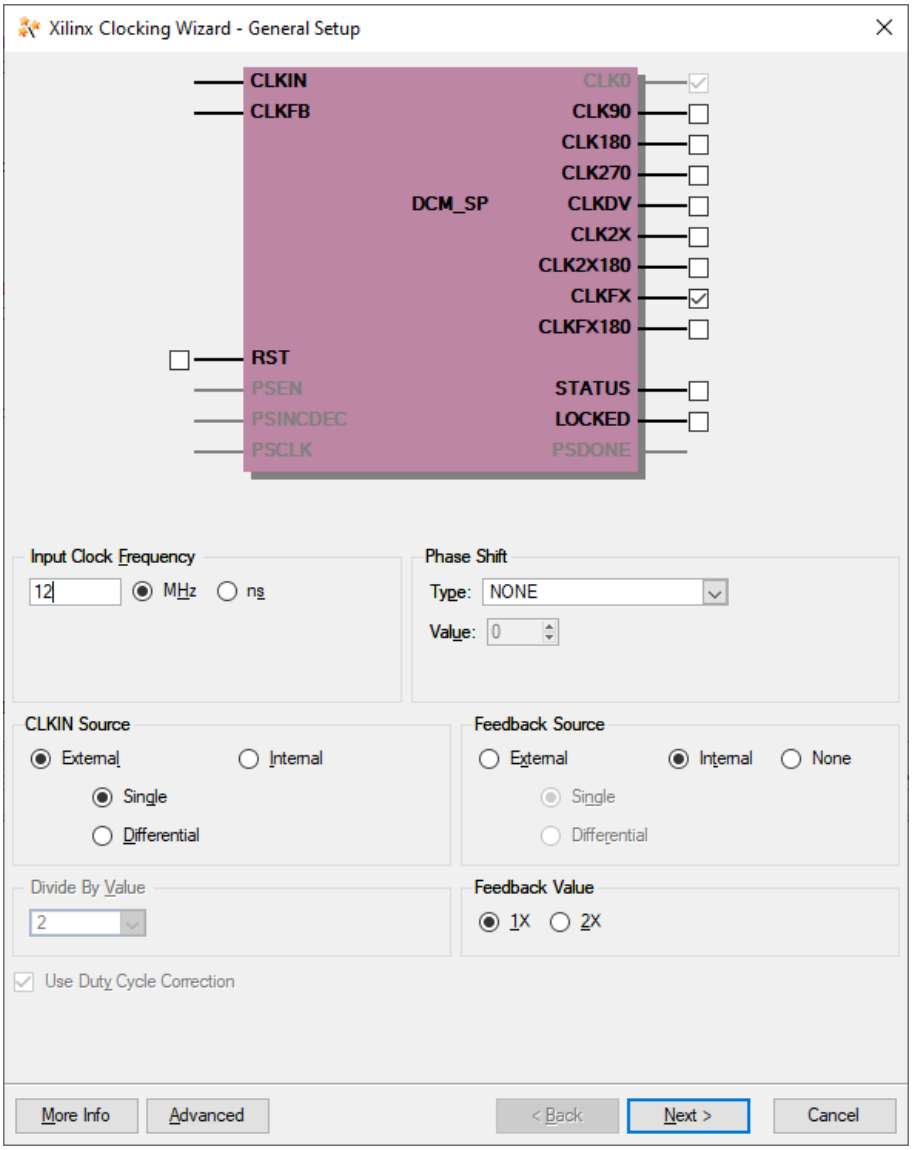
Property Name	Value
Evaluation Development Board	None Specified
Product Category	All
Family	Spartan3A and Spartan3AN
Device	XC3S50A
Package	TQ144
Speed	-4
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	VHDL
Property Specification in Project File	Store all values
Manual Compile Order	☐
VHDL Source Analysis Standard	VHDL-200X
Enable Message Filtering	☐

Add source files named *circle_vga.vhd* and *circle_vhd.ucf* to the project. Then we add a new source to the project and using the IP Core generator we add a DCM block and type *dcm_inst* as its name.



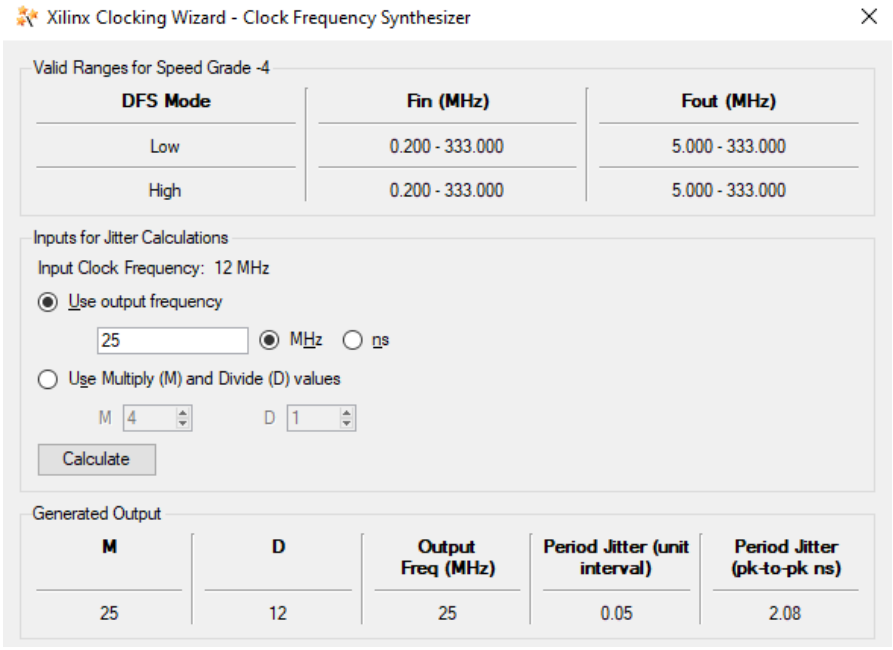


In the next two steps, we confirm the settings and select the VHDL language as the source language for the DCM block instance. Then we set the parameters of the clock signal at the input - 12 MHz and select the CLKFX output, uncheck the rest.

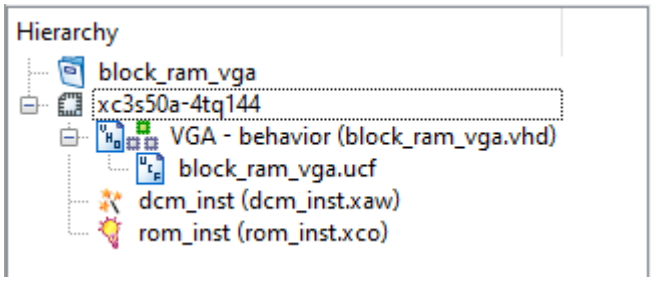


After pressing the *Next* button in this window and in the next one, the DCM block configurator will take you to the clock frequency synthesizer page. In the synthesizer

window, we specify the output clock frequency of 25 MHz. It is necessary for proper operation of the video signal generator for a monitor with a VGA port, working with a resolution of 640 x 480 pixels. These performance parameters of a VGA monitor are defined by the VESA standard.



The dcm_inst.xaw file should appear in the project.



NOTE: Lines related to the DCM block have been commented out in the source files. After adding this block, uncomment the relevant lines shown in the listing below.

```
component dcm_inst
port(
     CLKIN_IN : IN std_logic;
     CLKFX_OUT : OUT std_logic
    );
end component;
...

DCM1: dcm_inst PORT MAP(CLKIN_IN => clk , CLKFX_OUT => clk_25);
```

After compiling the project and generating the circle_vga.bit configuration file, program the target system and check the effect of the system on the VGA monitor screen.

2) A simple animation of a square moving across the screen.

Create a new project called animation_vga and add the **.vhd* and **.ucf* source files with the same name to it. In the next step, add a DCM block and uncomment the code snippet for this component. After configuring the Spartan-3A system with the *.bit file generated from this project, a white square on a black background should appear on the screen, which will move with a frequency of 60 Hz, right to a certain position and back again.

Tasks:

- the first project with a circle should be modified so that only the border of the circle (circle) is displayed on the screen.
- the second project should be converted into an animation of a square in the vertical and horizontal planes at the same time.
- add an animation of a smooth color change with a frequency of 60 Hz (screen refresh) to the circle

References

- J. Majewski, P. Zbysiński – *Układy FPGA w przykładach.* Wydawnictwo BTC, Legionowo 2007.
- Digilent Reference Page: Programmable Logic Tutorials - *VGA Display Controller.* https://digilent.com/reference/learn/programmable-logic/tutorials/vga-display-congroller/start