color_maker user manual

Title	color_maker-s3esk (simple VGA tester)
Author	Nikolaos Kavvadias (C) 2014, 2015, 2016
Contact	nikos@nkavvadias.com
Source	Mike Field (for the VGA controller; see AUTHORS)
Website	http://www.nkavvadias.com
Release Date	08 August 2016
Version	1.1.1
Rev. history	
v1.1.1	2016-08-08 Added clean-up script.
v1.1.0	2016-07-10 GHDL simulation scripts and generation of log file for VGA simulation.
v1.0.2	2016-07-10 Update for 2016.
v1.0.1	2014-06-18 Changed README to README.rst; COPYING to LICENSE.
v1.0.0	2014-06-09 Initial release for the Spartan-3E Starter kit board.

1. Introduction

`color_maker` is a simple design for testing VGA output. This version of the `color_maker` produces 3-bit RGB color (R1G1B1) as supported by the Xilinx Spartan-3E starter kit board. 3 out of the 4 available slide switches (specifically switches SW2, SW1, SW0) are used for setting a specific color out of the eight unique colors that are available.

The following table summarizes the available colors.

R	G	B	Description
0	0	0	Black (Noir)
0	0	1	Blue
0	1	0	Green

... continued on next page

R	G	B	Description
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

For the standard VGA resolution (640x480@60Hz), a 25MHz clock is used, as produced by the `clockdiv` clock divider (divide by 2). The VGA controller is implemented by `vgactrl.vhd`. The color selection logic is very simple and directly assigns SW2 to the R, SW1 to G and SW0 to the B component. For 800x600@72Hz, a 50 MHz clock should be used.

The `vga_controller` design has been adapted from the work by Mike Field: http://hamsterworks.co.nz/mediawiki/index.php/Hidef_snow

2. File listing

The `color_maker` distribution includes the following files:

/color_maker-s3esk	Top-level directory
AUTHORS	List of authors.
LICENSE	3-clause modified BSD license.
README.rst	This file.
README.html	HTML version of README.rst.
README.pdf	PDF version of README.rst.
clean.sh	A bash script for cleaning simulation artifacts.
clockdiv.vhd	Configurable, portable, clock divider.
color_maker.vhd	Color assignment logic.
color_maker_top.ucf	User Constraints File for the XC3S500E-FG320-4 device.
color_maker_top.vhd	The top-level RTL VHDL design file.
color_maker_top_tb.vhd	Testbench for the top-level RTL VHDL design file.
color_maker_top-syn.sh	Bash shell script for synthesizing the `color_maker` design with Xilinx ISE.
ghdl.mk	Makefile for VHDL simulation with GHDL.
ghdl.sh	Bash shell script for running the simulation with GHDL.
impact_s3esk.bat	Windows Batch file for automatically invoking Xilinx IMPACT in order to download the generated bitstream to the target hardware.
rst2docs.sh	Bash script for generating the HTML and PDF versions.
vgactrl.vhd	RTL VHDL code for the VGA controller.
xst.mk	Standard Makefile for command-line usage of ISE.

3. Usage

The color_maker distribution includes scripts for logic synthesis automation supporting Xilinx ISE. The corresponding synthesis script can be edited in order to specify the following for adapting to the user's setup:

- XDIR: the path to the /bin subdirectory of the Xilinx ISE/XST installation where the xst.exe executable is placed
- arch: specific FPGA architecture (device family) to be used for synthesis
- part: specific FPGA part (device) to be used for synthesis

3.1. Running the simulation script

This step assumes that the GHDL executable is in the user's $PATH, e.g., by using:

```
$ export PATH=/path/to/ghld/bin:$PATH
```

Then the simulation shell script can be run from a UNIX/Linux/Cygwin command line:

```
$ ./ghdl.sh
```

This will produce a text file named color_maker_top_results.txt with the values of current time whenever a clock event occurs (as integer) and the signals hs, vs, red, green and blue (as binary). This can be used by an external tool, the VGA simulator (http://ericeastwood.com/lab/vga-simulator/) for visualizing the outcome if a VGA/CRT monitor would be driven. A downloadable version of the VGA simulator also exists: https://github.com/MadLittleMods/vga-simulator

In order to work with the VGA simulator without further changes, the red, green and blue signals are extended to 3, 3, and 2 bits, respectively in the testbench.

To clean up simulation artifacts, including the generated diagnostics file, use the clean.sh script:

```
$ ./clean.sh
```

3.2. Running the synthesis script

For running the Xilinx ISE synthesis tool, generating FPGA configuration bistream and downloading to the target device, execute the corresponding script from within the color_maker-s3esk directory:

```
$ ./color_maker_top-syn.sh
```

In order to successfully run the entire process, you should have the target board connected to the host and it should be powered on.

The synthesis procedure invokes several Xilinx ISE command-line tools for logic synthesis as described in the corresponding Makefile, found in the the color_maker-s3esk directory.

Typically, this process includes the following:

- Generation of the `*.xst` synthesis script file.
- Generation of the `*.ngc` gate-level netlist file in NGC format.
- Building the corresponding `*.ngd` file.
- Performing mapping using `map` which generates the corresponding `*.ncd` file.
- Place-and-routing using `par` which updates the corresponding `*.ncd` file.
- Tracing critical paths using `trce` for reoptimizing the `*.ncd` file.
- Bitstream generation (`*.bit`) using `bitgen`, however with unused pins.

As a result of this process, the `color_maker_top.bit` bitstream file is produced.

Then, the shell script invokes the Xilinx IMPACT tool by a Windows batch file, automatically passing a series of commands that are necessary for configuring the target FPGA device:

1. Set mode to binary scan.

```
setMode -bs
```

2. Set cable port detection to auto (tests various ports).

```
setCable -p auto
```

3. Identify parts and their order in the scan chain.

```
identify
```

4. Assign the bitstream to the first part in the scan chain.

```
assignFile -p 1 -file color_maker_top.bit
```

5. Program the selected device.

```
program -p 1
```

6. Exit IMPACT.

```
exit
```

4. Prerequisites

- [suggested] Linux (e.g., Ubuntu 16.04 LTS) or MinGW environment on Windows 7 (64-bit).
- [suggested] GHDL simulator: http://ghdl.free.fr The 0.33 version on Linux Ubuntu 16.04 LTS was used.
- [optional] The VGA simulator: http://ericeastwood.com/lab/vga-simulator/

- Xilinx ISE (free ISE webpack is available from the Xilinx website): http://www.xilinx.com. The 14.6 version on Windows 7/64-bit is known to work.