ledramp user manual

Title	ledramp-s3esk (LED ramp effect)
Author	Nikolaos Kavvadias (C) 2014, 2015, 2016 (Modifications/ additions)
Source	User "buserror" (C) 2007
Contact	nikos@nkavvadias.com
Website	http://www.nkavvadias.com
Release Date	18 June 2014
Version	1.0.2
Rev. history	
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

`ledramp` is a "Knight Rider" style LED ramp effect, which uses two complementary PWM phases for consecutive states. Essentially the first state (PULSE) is a counting state lasting for 0.1 sec. The second state, SHIFT, is where the actual shift (left or right) is applied and the boundary conditions (for the leftmost and rightmost positions) are taken into account.

This version of the design directly uses the 50 MHz clock source available on the Xilinx Spartan-3E starter kit board.

The design has been adapted from this known original source: http://www.avrfreaks.net/index.php?name=PNphpBB2&file=printview&t=54866&start=40

2. File listing

The ledramp distribution includes the following files:

/ledramp-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.
ledramp_s3esk.ucf	User Constraints File for the XC3S500E-FG320-4 device.
ledramp_s3esk.vhd	The top-level RTL VHDL design file.
ledramp-syn.sh	Bash shell script for synthesizing the `ledramp` design with Xilinx ISE.
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.
xst.mk	Standard Makefile for command-line usage of ISE.

3. Usage

The `ledramp` 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 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 `ledramp-s3esk` directory:

```
$ ./ledramp-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 `ledramp-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 `ledramp_s3esk.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 ledramp_s3esk.bit
```

5. Program the selected device.

```
program -p 1
```

6. Exit IMPACT.

```
exit
```

4. Prerequisites

- [suggested] MinGW environment on Windows 7 (64-bit).
- 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.