bstest user manual

Title	bstest-s3esk (Buttons and switches test)
Author	Nikolaos Kavvadias (C) 2014
Contact	nikos@nkavvadias.com
Website	http://www.nkavvadias.com
Release Date	09 August 2016
Version	1.0.3
Rev. history	
v1.0.3	2016-08-09 Update file names, add GHDL simulation scripts. Add std_logic_textio package as user's package
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

`bstest` is a very simple design for testing push buttons, slide switches and discrete LEDs available on the Spartan-3E Starter Kit board by Xilinx Inc.

2. File listing

The bstest distribution includes the following files:

/bstest-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.
bstest.ucf	User Constraints File for the XC3S500E-FG320-4 device.

bstest.vhd	The top-level RTL VHDL design file.
bstest-syn.sh	Bash shell script for synthesizing the `bstest` design with Xilinx ISE.
clean.sh	A bash script for cleaning simulation artifacts.
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.
std_logic_textio.vhd	Synopsys' std_logic_textio package.
xst.mk	Standard Makefile for command-line usage of ISE.

3. Usage

The bstest 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 `bstest_results.txt` with the values of current time whenever a clock event occurs (as integer) and the signal `ramp`.

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 `bstest-s3esk` directory:

```
$ ./bstest-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 `bstest-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 `bstest.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 bstest.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.
- 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.