Technical Document
ModelSim Starter Edition

Version 0.0.1

Dat Ngo

March 4, 2024

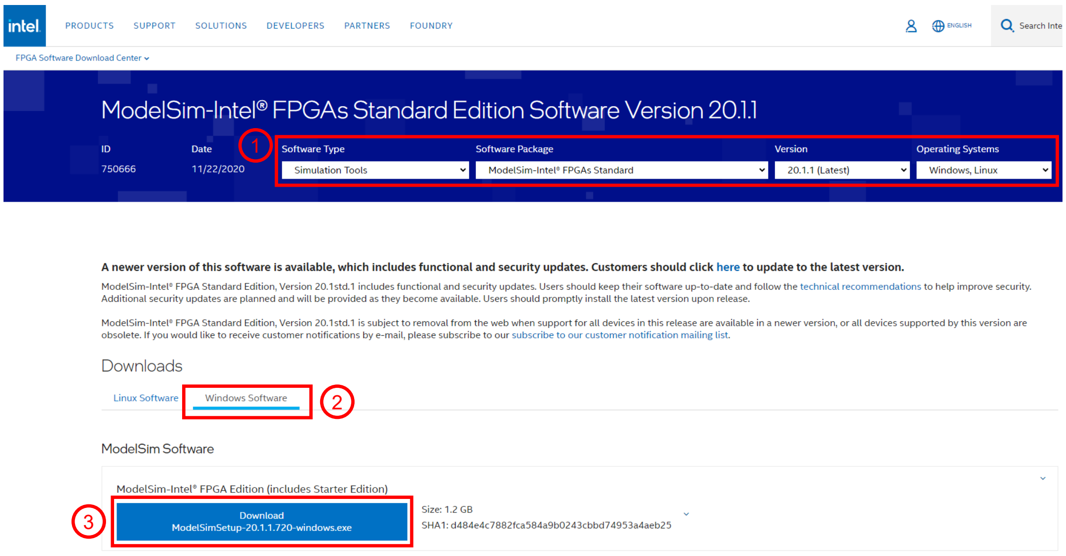
Contents

List of Figures

List of Code Listings

1 Software installation

1. Visit Intel homepage and download the setup file



ModelSim-Intel® FPGAs Standard Edition Software Version 20.1.1

ID: 750666
Date: 11/22/2020
Software Type: Simulation Tools
Software Package: ModelSim-Intel® FPGAs Standard
Version: 20.1.1 (Latest)
Operating Systems: Windows, Linux

A newer version of this software is available, which includes functional and security updates. Customers should click here to update to the latest version.

ModelSim-Intel® FPGA Standard Edition, Version 20.1std.1 includes functional and security updates. Users should keep their software up-to-date and follow the technical recommendations to help improve security. Additional security updates are planned and will be provided as they become available. Users should promptly install the latest version upon release.

ModelSim-Intel® FPGA Standard Edition, Version 20.1std.1 is subject to removal from the web when support for all devices in this release are available in a newer version, or all devices supported by this version are obsolete. If you would like to receive customer notifications by e-mail, please subscribe to our subscribe to our customer notification mailing list.

Downloads

Linux Software | Windows Software

ModelSim Software

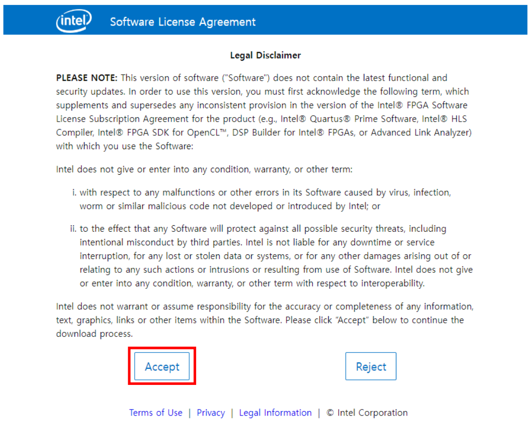
ModelSim-Intel® FPGA Edition (includes Starter Edition)

Download ModelSimSetup-20.1.1.720-windows.exe

Size: 1.2 GB
SHA1: d484e4c7882fca584a9b0243cbbd74953a4aeb25

Figure 1: Visit homepage and download setup file

2. Click “Accept” to start downloading



Software License Agreement

Legal Disclaimer

PLEASE NOTE: This version of software ("Software") does not contain the latest functional and security updates. In order to use this version, you must first acknowledge the following term, which supplements and supersedes any inconsistent provision in the version of the Intel® FPGA Software License Subscription Agreement for the product (e.g., Intel® Quartus® Prime Software, Intel® HLS Compiler, Intel® FPGA SDK for OpenCL™, DSP Builder for Intel® FPGAs, or Advanced Link Analyzer) with which you use the Software:

Intel does not give or enter into any condition, warranty, or other term:

i. with respect to any malfunctions or other errors in its Software caused by virus, infection, worm or similar malicious code not developed or introduced by Intel; or

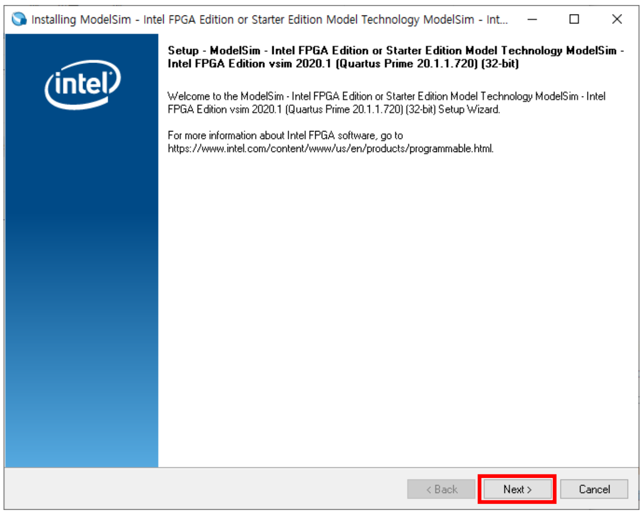
ii. to the effect that any Software will protect against all possible security threats, including intentional misconduct by third parties. Intel is not liable for any downtime or service interruption, for any lost or stolen data or systems, or for any other damages arising out of or relating to any such actions or intrusions or resulting from use of Software. Intel does not give or enter into any condition, warranty, or other term with respect to interoperability.

Intel does not warrant or assume responsibility for the accuracy or completeness of any information, text, graphics, links or other items within the Software. Please click “Accept” below to continue the download process.

Terms of Use | Privacy | Legal Information | © Intel Corporation

Figure 2: Accept conditions and download

3. Click “Next”


Installing ModelSim - Intel FPGA Edition or Starter Edition Model Technology ModelSim - Int...

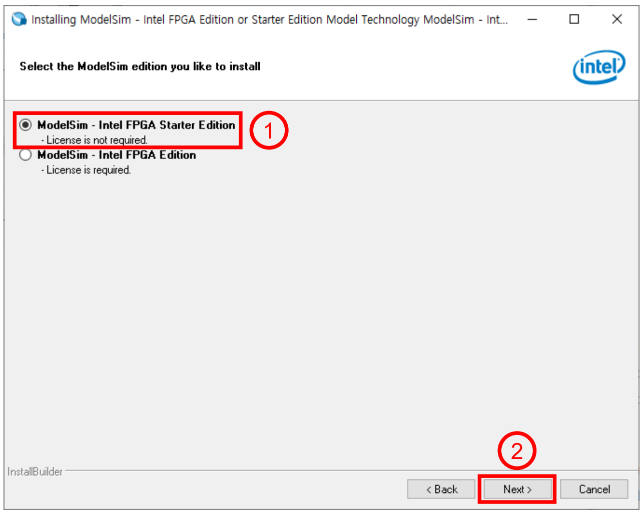
Setup - ModelSim - Intel FPGA Edition or Starter Edition Model Technology ModelSim - Intel FPGA Edition vsim 2020.1 (Quartus Prime 20.1.1.720) (32-bit)

Welcome to the ModelSim - Intel FPGA Edition or Starter Edition Model Technology ModelSim - Intel FPGA Edition vsim 2020.1 (Quartus Prime 20.1.1.720) (32-bit) Setup Wizard.

For more information about Intel FPGA software, go to https://www.intel.com/content/www/us/en/products/programmable.html.

Figure 3: Click “Next”

4. Select starter version and click “Next”

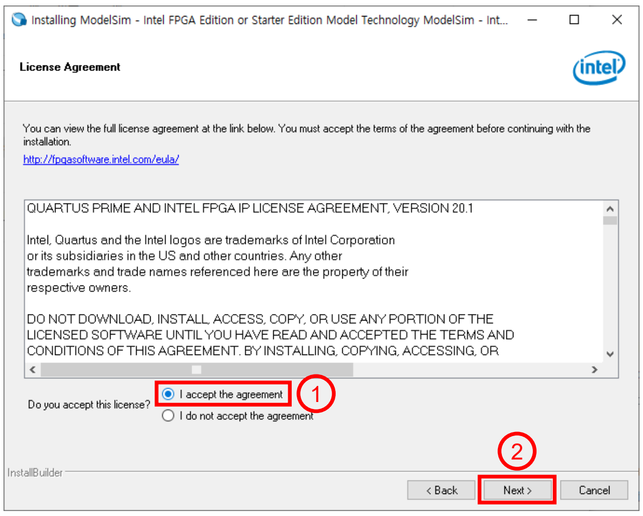

Installing ModelSim - Intel FPGA Edition or Starter Edition Model Technology ModelSim - Int...

Select the ModelSim edition you like to install

- [✓ **ModelSim - Intel FPGA Starter Edition**]
  - License is not required.
- **ModelSim - Intel FPGA Edition**
  - License is required.

Figure 4: Select starter version

5. Accept the agreement and click “Next”


Installing ModelSim - Intel FPGA Edition or Starter Edition Model Technology ModelSim - Int...

License Agreement

You can view the full license agreement at the link below. You must accept the terms of the agreement before continuing with the installation.
http://fpgasoftware.intel.com/eula/

QUARTUS PRIME AND INTEL FPGA IP LICENSE AGREEMENT, VERSION 20.1

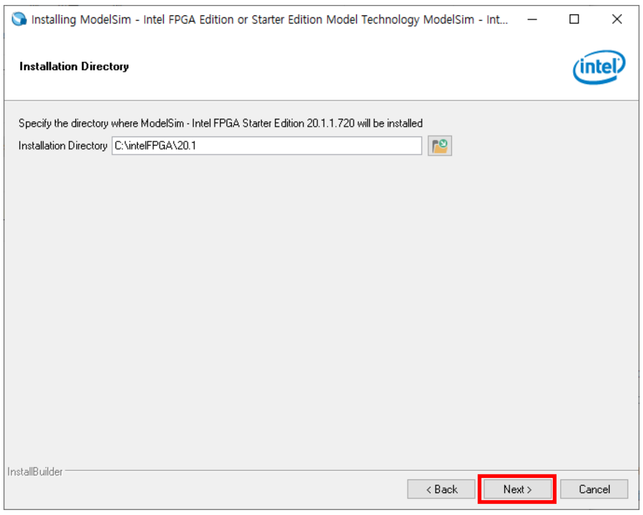
Intel, Quartus and the Intel logos are trademarks of Intel Corporation
or its subsidiaries in the US and other countries. Any other
trademarks and trade names referenced here are the property of their
respective owners.

DO NOT DOWNLOAD, INSTALL, ACCESS, COPY, OR USE ANY PORTION OF THE
LICENSED SOFTWARE UNTIL YOU HAVE READ AND ACCEPTED THE TERMS AND
CONDITIONS OF THIS AGREEMENT. BY INSTALLING, COPYING, ACCESSING, OR

Do you accept this license? [✓ I accept the agreement] I do not accept the agreement

Figure 5: Accept the agreement

6. Click “Next”


Installing ModelSim - Intel FPGA Edition or Starter Edition Model Technology ModelSim - Int...

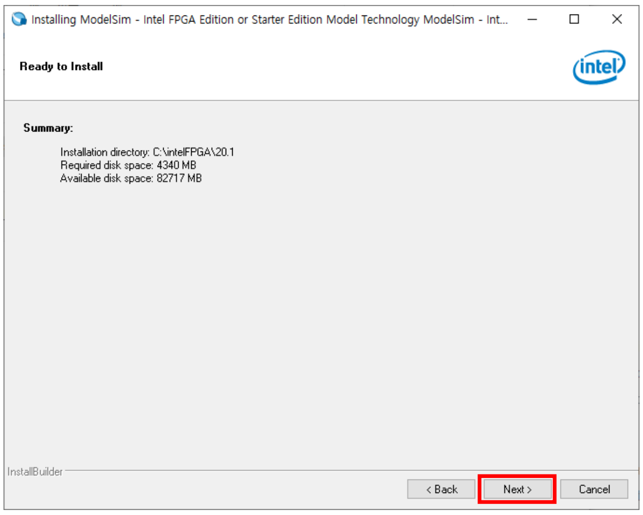
Installation Directory

Specify the directory where ModelSim - Intel FPGA Starter Edition 20.1.1.720 will be installed

Installation Directory C:\intelFPGA\20.1

Figure 6: Click “Next”

7. Click “Next”


Installing ModelSim - Intel FPGA Edition or Starter Edition Model Technology ModelSim - Int...

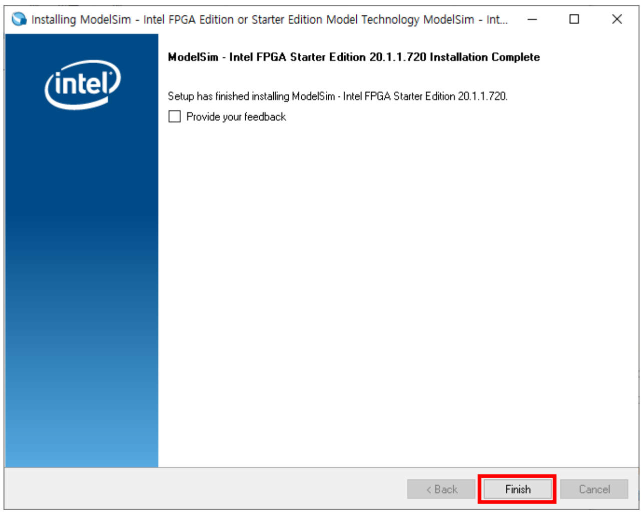
Ready to Install

Summary:

Installation directory: C:\intelFPGA\20.1
Required disk space: 4340 MB
Available disk space: 82717 MB

Figure 7: Click “Next”

8. Wait until finish


Installing ModelSim - Intel FPGA Edition or Starter Edition Model Technology ModelSim - Int...

ModelSim - Intel FPGA Starter Edition 20.1.1.720 Installation Complete

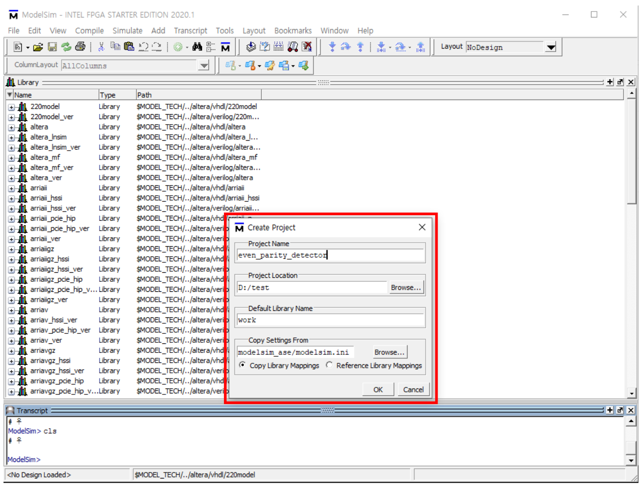
Setup has finished installing ModelSim - Intel FPGA Starter Edition 20.1.1.720.

☐ Provide your feedback

Figure 8: Wait until finish

2 Examples

1. Open ModelSim and select “File → New → Project”
2. Enter the project name and select the project location


ModelSim - INTEL FPGA STARTER EDITION 2020.1
File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks Window Help
Layout NoDesign
ColumnLayout AllColumns
Library
Create Project
Browse...
OK
Cancel
Transcript

Name	Type	Path
220model	Library	$MODEL_TECH/../altera/vhdl/220model
220model_ver	Library	$MODEL_TECH/../altera/verilog/220m...
altera	Library	$MODEL_TECH/../altera/vhdl/altera
altera_lnsim	Library	$MODEL_TECH/../altera/vhdl/altera_l...
altera_lnsim_ver	Library	$MODEL_TECH/../altera/verilog/altera...
altera_mf	Library	$MODEL_TECH/../altera/vhdl/altera_mf
altera_mf_ver	Library	$MODEL_TECH/../altera/verilog/altera...
altera_ver	Library	$MODEL_TECH/../altera/verilog/altera
arriaii	Library	$MODEL_TECH/../altera/vhdl/arriaii
arriaii_hssi	Library	$MODEL_TECH/../altera/vhdl/arriaii_hssi
arriaii_hssi_ver	Library	$MODEL_TECH/../altera/verilog/arriaii...
arriaii_pcie_hip	Library	$MODEL_TECH/../altera/vhd
arriaii_pcie_hip_ver	Library	$MODEL_TECH/../altera/veri
arriaii_ver	Library	$MODEL_TECH/../altera/veri
arriaiigz	Library	$MODEL_TECH/../altera/vhd
arriaiigz_hssi	Library	$MODEL_TECH/../altera/vhd
arriaiigz_hssi_ver	Library	$MODEL_TECH/../altera/veri
arriaiigz_pcie_hip	Library	$MODEL_TECH/../altera/vhd
arriaiigz_pcie_hip_v...	Library	$MODEL_TECH/../altera/veri
arriaiigz_ver	Library	$MODEL_TECH/../altera/veri
arriav	Library	$MODEL_TECH/../altera/vhd
arriav_hssi_ver	Library	$MODEL_TECH/../altera/veri
arriav_pcie_hip_ver	Library	$MODEL_TECH/../altera/veri
arriav_ver	Library	$MODEL_TECH/../altera/veri
arriavgz	Library	$MODEL_TECH/../altera/vhd
arriavgz_hssi	Library	$MODEL_TECH/../altera/vhd
arriavgz_hssi_ver	Library	$MODEL_TECH/../altera/veri
arriavgz_pcie_hip	Library	$MODEL_TECH/../altera/vhd
arriavgz_pcie_hip_v...	Library	$MODEL_TECH/../altera/veri

Project Name: even_parity_detector
Project Location: D:/test
Default Library Name: work
Copy Settings From: modelsim_ase/modelsim.ini
[✓ Copy Library Mappings] Reference Library Mappings

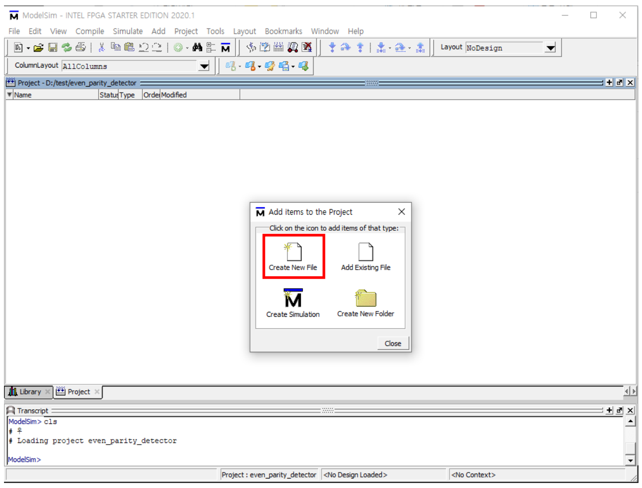
```
# ♀
ModelSim> cls
# ♀

ModelSim>
```

<No Design Loaded> $MODEL_TECH/../altera/vhdl/220model

Figure 9: New project

3. Select “Add new file” and add two files
   - Source file : `even_parity_detector.vhd`
   - Test bench : `test_bench.vhd`

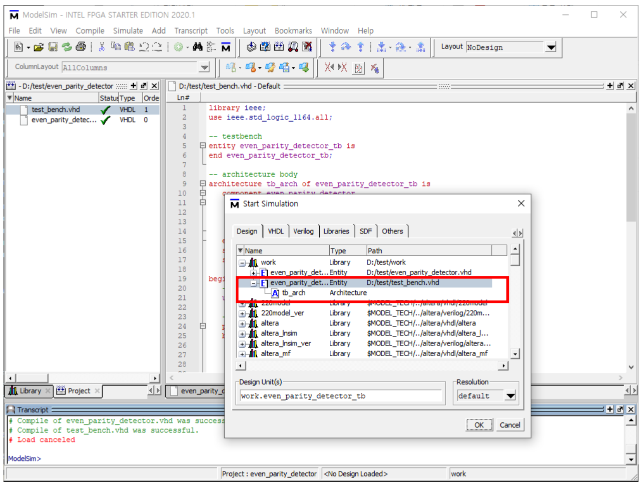

ModelSim - INTEL FPGA STARTER EDITION 2020.1
File Edit View Compile Simulate Add Project Tools Layout Bookmarks Window Help
Layout NoDesign
ColumnLayout AllColumns
Project - D:/test/even_parity_detector
Name
Status
Type
Order
Modified
Add items to the Project
Click on the icon to add items of that type:
Create New File
Add Existing File
Create Simulation
Create New Folder
Close
Library
Project
Transcript

```
ModelSim> cls
# ♀
# Loading project even_parity_detector
ModelSim>
```

Project : even_parity_detector <No Design Loaded> <No Context>

Figure 10: Add new file

4. Select “Compile → Compile all”
5. Select “Simulate → Start simulation”
6. Select the test bench and click “OK”



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

-- testbench
entity even_parity_detector_tb is
end even_parity_detector_tb;

-- architecture body
architecture tb_arch of even_parity_detector_tb is
```

Name	Type	Path
work	Library	D:/test/work
even_parity_det...	Entity	D:/test/even_parity_detector.vhd
even_parity_det...	Entity	D:/test/test_bench.vhd
tb_arch	Architecture	
220model_ver	Library	$MODEL_TECH/../altera/verilog/220m...
altera	Library	$MODEL_TECH/../altera/vhdl/altera
altera_lnsim	Library	$MODEL_TECH/../altera/vhdl/altera_l...
altera_lnsim_ver	Library	$MODEL_TECH/../altera/verilog/altera...
altera_mf	Library	$MODEL_TECH/../altera/vhdl/altera_mf

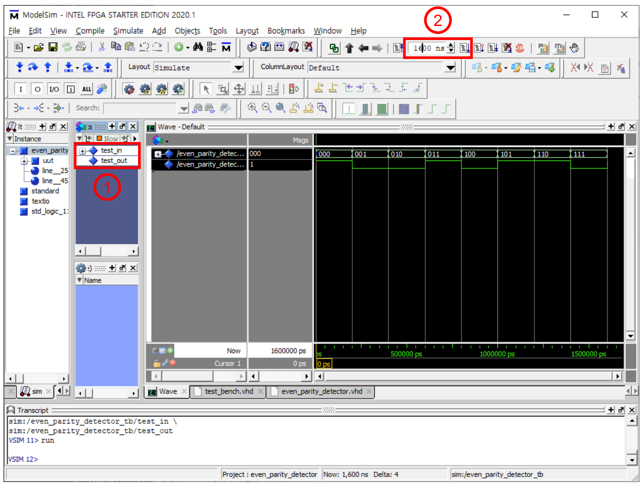
Design Unit(s): work.even_parity_detector_tb
Resolution: default

```
# Compile of even_parity_detector.vhd was success
# Compile of test_bench.vhd was successful.
# Load canceled

ModelSim>
```

Figure 11: Start simulation

7. “Select signals → Right click → Add wave”
8. Set step length and click “Run”



```
sim:/even_parity_detector_tb/test_in \
sim:/even_parity_detector_tb/test_out
VSIM 11> run

VSIM 12>
```

Project : even_parity_detector   Now: 1,600 ns   Delta: 4   sim:/even_parity_detector_tb

Figure 12: Run simulation

9

3 Appendix

3.1 Source file

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

-- entity declaration
entity even_parity_detector is
   port (
      a: in std_logic_vector(2 downto 0);
      even: out std_logic
   );
end even_parity_detector;

-- architecture body
architecture xor_arch of even_parity_detector is
   signal odd: std_logic;
begin
   even <= not odd;
   odd <= a(2) xor a(1) xor a(0);
end xor_arch;
```

Listing 1: Source file

3.2 Test bench

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

-- testbench
entity even_parity_detector_tb is
end even_parity_detector_tb;

-- architecture body
architecture tb_arch of even_parity_detector_tb is
   component even_parity_detector
      port(
         a: in std_logic_vector(2 downto 0);
         even: out std_logic
      );
   end component;
   signal test_in: std_logic_vector(2 downto 0);
   signal test_out: std_logic;

begin
   -- instantiate the unit under test
   uut: even_parity_detector
      port map(a=>test_in, even=>test_out);
   -- test pattern generator
   process
   begin
      test_in <= "000";
      wait for 200 ns;
      test_in <= "001";
      wait for 200 ns;
      test_in <= "010";
      wait for 200 ns;
```

Listing 2: Test bench

```
      test_in <= "011";
      wait for 200 ns;
      test_in <= "100";
      wait for 200 ns;
      test_in <= "101";
      wait for 200 ns;
      test_in <= "110";
      wait for 200 ns;
      test_in <= "111";
      wait for 200 ns;
   end process;

   -- verifier
   process
      variable error_status: boolean;
   begin
      wait on test_in;
      wait for 100 ns;
      if ((test_in="000" and test_out='1') or
          (test_in="001" and test_out='0') or
          (test_in="010" and test_out='0') or
          (test_in="011" and test_out='1') or
          (test_in="100" and test_out='0') or
          (test_in="101" and test_out='1') or
          (test_in="110" and test_out='1') or
          (test_in="111" and test_out='0'))
      then
         error_status := false;
      else
         error_status := true;
      end if;
      -- error reporting
      assert not error_status
         report "test failed."
         severity note;
   end process;
end tb_arch;
```

Listing 3: Test bench (cont.)