HOCHSCHULE
RHEIN-WAAL
Rhine-Waal University
of Applied Sciences

LABORATORY 1

Model-based Hardware Design

Lab notes prepared by Mr. Gligor Duchev

Lab notes revised by: Prof. Andy Stamm

Contents

1. Safety instructions

ESD Warning

Caution Although this product has been designed to be as robust as possible, ESD (Electrostatic Discharge) can damage or upset this product. This product must be protected at all times from ESD. Static charges may easily produce potentials of several kilovolts on the human body or equipment, which can discharge without detection. Industry-standard ESD precautions must be employed at all times.

The NI Digital Electronics FPGA Board is designed and intended for use as a development platform for hardware or software in an educational/professional laboratory environment. To facilitate usage, the board is manufactured with its components and connecting traces openly exposed to the operator and the environment. As a result, ESD sensitive (ESDS) components on the board, such as the semiconductor integrated circuits, can be damaged when exposed to an ESD event. To indicate the ESD sensitivity of the NI Digital Electronics FPGA Board, it carries the symbol shown below.

Handling the NI Digital Electronics FPGA Board can damage the board components if ESD prevention measures are not applied. **Before handling or setup, equalize your potential with the board by touching one of the integrated ESD discharge pads.** During all handling and setup, ESD prevention measures must be applied. In addition, the NI Digital Electronics FPGA Board should be handled by the edges. Touching exposed circuits, components or connectors could result in an ESD event.

2. Introduction to NI Elvis

What is NI Elvis?

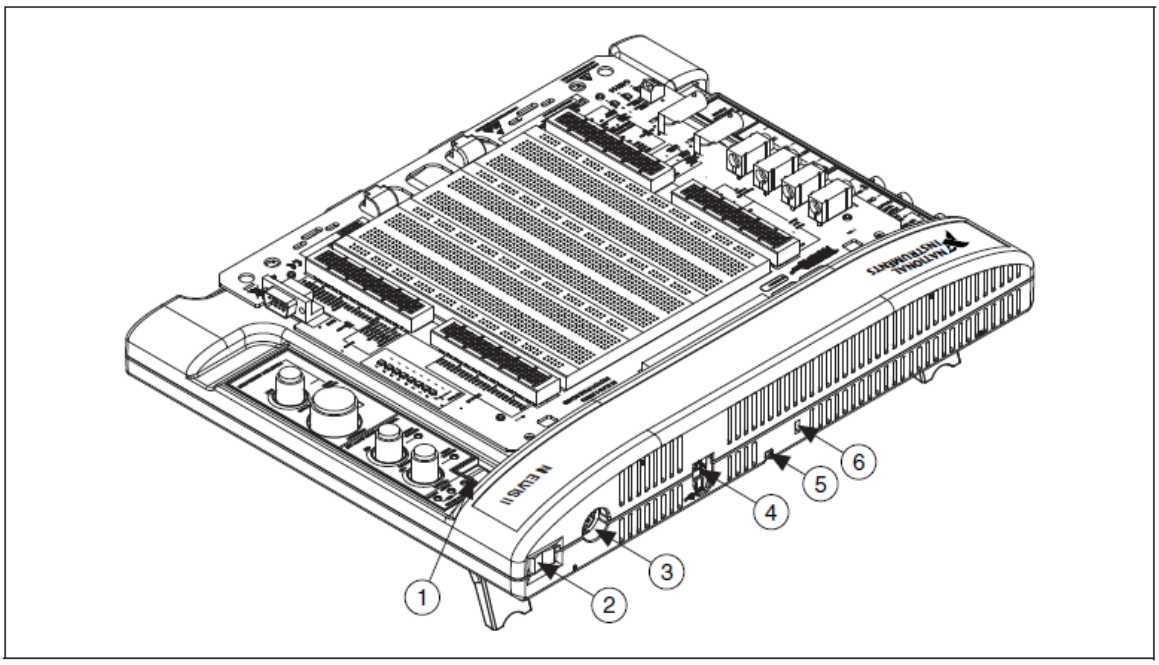
The National Instruments Educational Laboratory Virtual Instrumentation Suite II Series (NI ELVIS II Series) is a LabVIEW-based design and prototyping environment for university science and engineering laboratories. This document explains how to set up and configure NI ELVIS II Series.

NI ELVIS II Series has the following features:

- USB-based workstation with a removable prototyping board for use in circuit development and Experimentation

- Integrated instruments for computer-based measurement and control including the following:

– Multi-channel data acquisition capabilities including both analog and digital I/O
– Digital Multimeter (DMM)
– 2-channel oscilloscope with a sampling rate of 1.25 MS/s
– Function generator
– Fixed and variable power supplies
– 2- and 3-wire impedance analyzer



1	Prototyping Board Power Switch	4	USB Connector
2	Workstation Power Switch	5	Cable Tie Slot
3	AC/DC Power Supply Connector	6	Kensington Security Slot

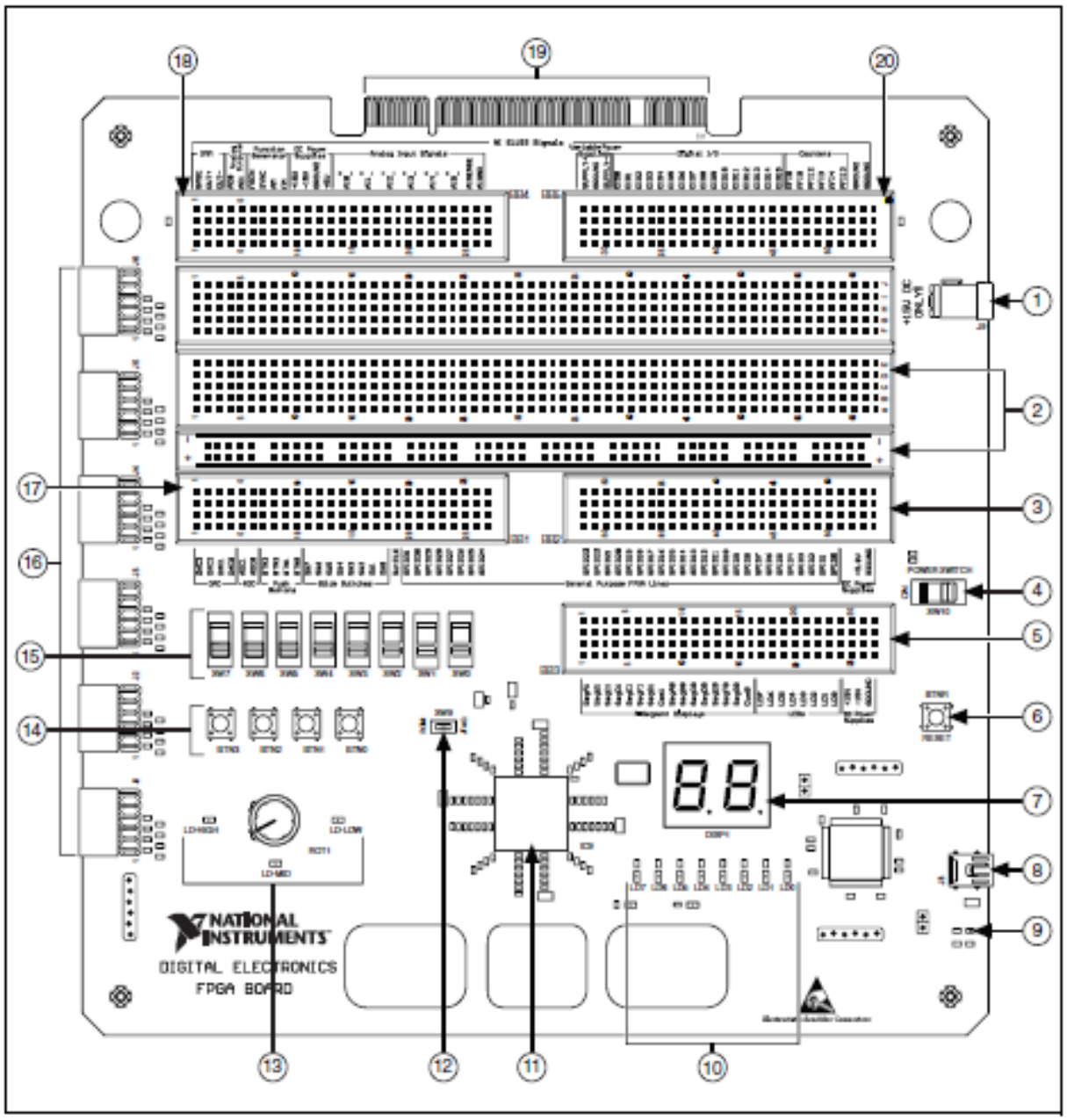
Figure 1: Rear View of NI ELVIS II Workstation (NI ELVIS shown)

3. Introduction to NI FPGA

The NI FPGA board is an extension to the NI ELVIS which allows to program and test FPGA based prototyping.

3.1. The NI FPGA board features (shown in Figure 2.):

- Eight slide switches, SW0 through SW7 (15)
- Four momentary-contact push-buttons, BTN0 through BTN3 (14)
- Eight individual surface-mount LEDs, LD0 through LD7. Each LED is connected on one side through a 390Ω current-limiting resistor to the power line (10)
- Two digit seven-segment display, DISP1, in a common cathode configuration. (7)
- Rotary push-button knob, ROT1, that is used to set the frequency range and value inside the range for an external clock generated by a microcontroller. (13)
-50 MHz onboard clock oscillator as the clock input. The 50 MHz clock output line, GCLK0.



1	Power Connector	7	Seven-Segment Displays	14	Push Buttons
2	General-Purpose Breadboard Banks	8	USB Connector	15	Slide Switches
		9	LD-G LED	16	Digilent Pmod Connectors
3	Signal Breadboard Bank BB2	10	LEDs	17	Signal Breadboard Bank BB1
4	Power Switch	11	FPGA	18	Signal Breadboard Bank BB4
5	Signal Breadboard Bank BB3	12	Switch SW9	19	NI ELVIS Connector
6	Reset Button	13	Rotary Push-Button Knob/LEDs	20	Signal Breadboard Bank BB5

Figure 2: The NI Digital Electronics FPGA Board

3.2. UCF File Constraints

3.2.1. Slide Switches

The UCF file constraints for the eight slide switches, SW0 to SW7, are listed as follows. SWx refers to the slide switch line, LOC indicates the FPGA line location, and IOSTANDARD is the I/O standard used.

```
Net "SW0" LOC="J11" | IOSTANDARD = LVCMOS33;
Net "SW1" LOC="J12" | IOSTANDARD = LVCMOS33;
Net "SW2" LOC="H16" | IOSTANDARD = LVCMOS33;
Net "SW3" LOC="H13" | IOSTANDARD = LVCMOS33;
Net "SW4" LOC="G12" | IOSTANDARD = LVCMOS33;
Net "SW5" LOC="E14" | IOSTANDARD = LVCMOS33;
Net "SW6" LOC="D16" | IOSTANDARD = LVCMOS33;
```

3.2.2. Push Buttons

The UCF file constraints for the four push buttons, BTN0 to BTN3, are listed as follows. BTNx refers to the push button line, LOC indicates the FPGA line location, and IOSTANDARD is the I/O standard used.

```
Net "BTN0" LOC="C13" | IOSTANDARD = LVCMOS33;
Net "BTN1" LOC="D12" | IOSTANDARD = LVCMOS33;
Net "BTN2" LOC="C12" | IOSTANDARD = LVCMOS33;
Net "BTN3" LOC="C10" | IOSTANDARD = LVCMOS33;
```

3.2.3. LEDs

The UCF file constraints for the eight LEDs, LED0 to LED7, are listed as follows. LEDx refers to the LED line, LOC indicates the FPGA line location, IOSTANDARD is the I/O standard used, SLEW refers to the slew rate, the maximum rate of change of a signal, and DRIVE indicates the current drive strength on the FPGA in milliamps.

```
Net "LED0" LOC="C11" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
Net "LED1" LOC="D11" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
Net "LED2" LOC="B11" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
Net "LED3" LOC="A12" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
Net "LED4" LOC="A13" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
Net "LED5" LOC="B13" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
Net "LED6" LOC="A14" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
Net "LED7" LOC="B14" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
```

4. Introduction to ISE Design Suite 14.7

This tutorial will introduce the reader to the Xilinx ISE software. Step-by-step instructions will be given to guide the reader through generating a project, creating a design file, compiling the project, and downloading the design to an FPGA board.

The Xilinx ISE tools allow the design to be entered several ways including graphical schematics, state machine diagrams, VHDL, and Verilog. This tutorial will focus on VHDL entry, but the other methods are similar and can be easily explored once the reader is comfortable with the ISE software.

4.1. Xilinx software

The Xilinx ISE 14.7 software will be used in this lab. All menus structures and screen shots are taken from this software version. This tutorial will NOT deal with the Foundation family of software.

To start the ISE Design Suite, double-click the Project Navigator icon on your desktop, or select Start > All Programs > Xilinx ISE Design Suite > Xilinx Design Suite 14 > ISE Design Tools > Project Navigator

4.2. Creating a New Project

To create a new project using the New Project Wizard, do the following:

1. From Project Navigator, select File > **New Project**.
   The New Project Wizard appears.
2. Choose the location of the project and write its name.
3. Verify that HDL is selected as the Top-Level Source Type, and click **Next**.

4. Select the following values in the New Project Wizard—Device Properties page:
   • Product Category: All
   • Family: Spartan3E
   • Device: XC3S500E
   • Package: FT256
   • Speed: -4
   • Synthesis Tool: XST (VHDL/Verilog)
   • Simulator: ISim (VHDL/Verilog)
   • Preferred Language: VHDL
5. Click **Next**, then Finish to complete the project creation.
6. After this, right click on the Xilinx FPGA in the left upper corner, in the Design window.
7. Click on **New Source** (the New Source Wizard pops up).

8. Choose VHDL Module, provide a name for the file, and skip the Specify ports window by pressing **Next** and **Finish** the VHDL Module creation.

4.3. Task 1: Toggle a LED with a button

With this task we are going to write a code that is going to turn on a LED on the FPGA board whenever a defined switch is toggled.

Steps:

1. In this course make it a practice to always include the

   use IEEE.STD_LOGIC_ARITH.ALL;

   use IEEE.STD_LOGIC_UNSIGNED.ALL;

   Libraries whenever creating code.

2. Define the entity with its ports:

   SW0: in STD_LOGIC:='1'

   LED0: out STD_LOGIC:='0'

3. For these entities we will define the behavior as following:

   LED0 <= SW0;

The code should look like this

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_ARITH.ALL;
4  use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6  entity Switch_LED is Port (SW0: in  STD_LOGIC:='1';
7                             LED0 : out  STD_LOGIC:='0');
8  end Switch_LED;
9
10 architecture Behavioral of Switch_LED is
11 begin
12
13 LED0 <= SW0;
14
15 end Behavioral;
16
```

Figure 3: First code for toggling a LED

4. After you are done with writing the code, you need to create an Implementation Constraints File, otherwise random pins will be assigned to the ports.

5. Right click on the FPGA in the Design window.

6. Click on **New Source**, and create Implementation Constraints File.

7. Provide a name for it and click **Finish**.

8. In this file we need to assign the pins that are provided in the FPGA documentation, they are also provided on page 5 and 6 of this lab guide in the section UCF File Constraints.

9. For this specific program we are going to need:

Net "SW0" LOC="J11" | IOSTANDARD = LVCMOS33;

Net "LED0" LOC="C11" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;

10. Now you can finally synthesize the Design. Do this by clicking on **Synthesize-XTS** in the Design window under running processes.

11. After the design is checked, you can go ahead and click on **Generate Programming File.**

5. Implementing the design

5.1. Programming the PROM

To program the formatted file into the platform flash PROM through the onboard USB circuitry, connect the USB cable to the NI Digital Electronics FPGA Board, apply power to the board by plugging in the Power Supply to the NI Elvis Board, move the power switch to the ON position, move the Prototyping Board Power Switch to the ON position, turn on the board by moving SW10 to the ON position, and complete the following steps:

1. Launch Xilinx ISE (if it is not yet open).

2. Select File»Open Project, and navigate to the location of PROM.ise.
   Click OK (if you need to load the project).

3. In the Processes pane of the project window, expand Configure Target Device.

4. If the iMPACT - Welcome to iMPACT window does not open, launch iMPACT by selecting Start»Programs»Xilinx ISE Design Suite»ISE Design Tools»64-bit Tools»iMPACT.

5. Select Configure devices using Boundary-Scan (JTAG) and automatically connect to a cable and identify Boundary-Scan chain (these should be selected by default). Click **Finish**. The Assign New Configuration File window opens.

6. Select the main.bit configuration file in the "Assign New Configuration File" window. Click **Open**.

7. You have the option of selecting additional configuration files (for the ROM).

8. Because you do not need a new configuration, click **Cancel**. The Device Programming Properties window opens.

9. In the Device Programming Properties window, select Device 1 (FPGA, xc3s500e) to program Device 1. The NI Digital Electronics FPGA Board FPGA is a Xilinx XC3S500E Spartan-3E FPGA.

10. Click **OK**. The Boundary Scan pane opens in the project window.

    Steps 11 to 13 are optional if you want to assign a bitfile to the ROM which allows the FPGA to reload the SW after a power loss.

11. Right-click the xcf04s file icon, and select Assign New Configuration File to assign the PROM file (.mcs), to the XCF04S platform flash PROM on the JTAG chain. Click **Open**.

12. Right-click the xcf04s myplatformflash.mcs icon and select Program.

13. In the Device Programming Properties window, select Device 2 (PROM, xcf04s), the PROM type to be programmed.

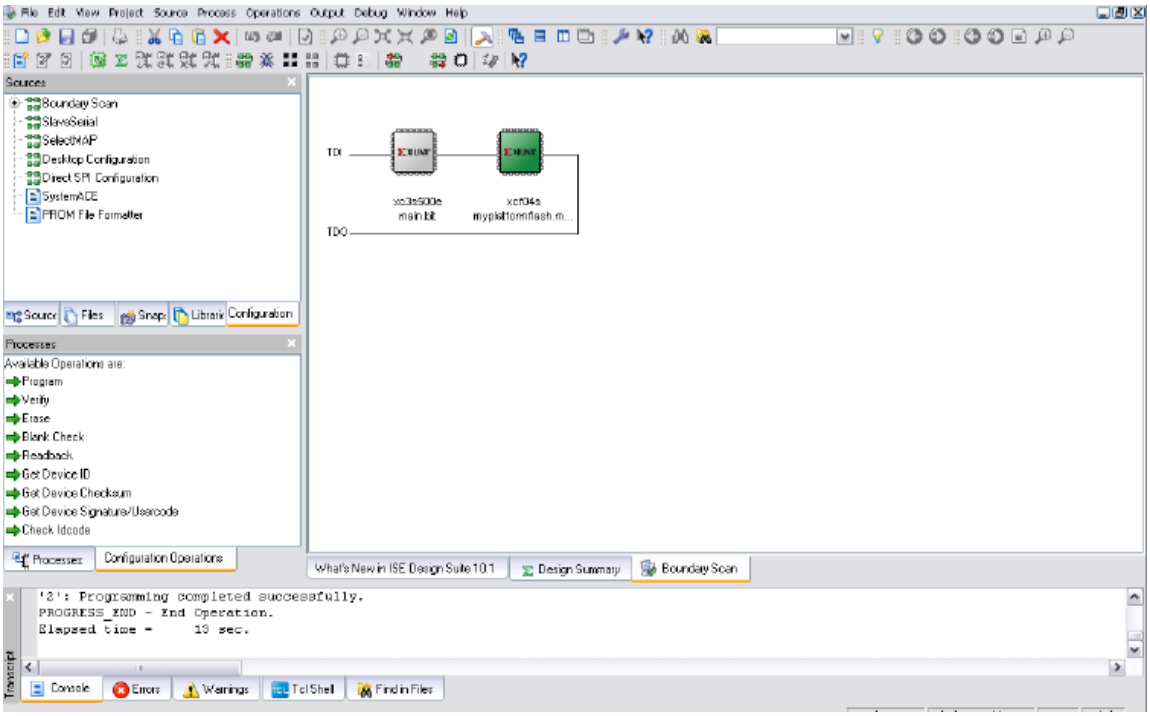


Figure 4: PROM.ise Project Window (Boundary Scan)

14. Put a check mark in the following Device Programming Properties options:
    - **Verify**—Verifies the PROM is correctly programmed and matches the downloaded configuration bit stream. This option is recommended though it increases overall programming time.
    - **Erase Before Programming**—Erases the platform flash PROM completely before programming, ensuring that no previous data lingers. This option is recommended though it increases overall programming time.
    - **Load FPGA**—Forces the FPGA to reconfigure after programming the platform flash PROM.

15. Click **OK**. A Progress Dialog window opens and displays the execution progress. After the PROM is successfully programmed, the Boundary Scan pane displays Program Succeeded.

16. Select File»Close Project and save all changes.

5.2. Testing the download

To test that the download was successful, complete the following steps.

1. Reboot the NI Digital Electronics FPGA Board by pressing the reset button.

2. Verify that the FPGA is running the (PROM) downloaded application.