Programmable Logic Design – I

Introduction

In labs 11 and 12 you built simple logic circuits on breadboards using TTL logic circuits on 7400 series chips. This process is simple and easy for small circuits. With increasing complexity of the logic circuitry the possibility of wiring errors grows and it becomes increasingly difficult to debug the circuit. Another problem is the difficulty in finding all the needed logic circuitry on available chips. To address these problems the electronics industry has developed the concepts of Programmable Logic Devices (PLD's) or Field Programmable Gate Arrays (FPGA's). The basic idea behind these devices is the notion that logic circuitry of arbitrary complexity can be constructed from simple gates connected with appropriate links and the technical advance that has made this possible is the development of large gate arrays with computer programmable links. The design process then consists of specifying the logic design by means of a logic design language such as VHDL or by entering it on a schematic layout. A computer program then turns this design into a series of instructions that are downloaded into the chip to establish the desired logic circuitry. Facilities are provided to specify the pin out of the logic, to control placement of logic circuits on the chip and to impose timing constraints.

Design Tools

For our designs we will be using the Xilinx Corporation (www.xilinx.com) ISE Foundation 9.2i. We will start by entering our design in the form of a circuit schematic but, in later stages of the labs we will use a high level language, VHDL, to benefit from its power and flexibility. In our designs we will sample only a few of the features and capabilities of this software package which is widely used in the electronics industry today.

Hardware

We will download our designs into a Digilab D2XL board connected to a Digilab Digital I/O board (DI01) shown in Figure 1 below. The FPGA chip on the D2XL board is a member of the Spartan II family, the XC2S30, with package type tq144, embodying 972 logic cells with a total of 30,000 gates. While this size of device was state-of-the-art a few years ago, rapid advances in technology have pushed the largest device sizes to many millions of gates.

The D2XL board's I/O resources are limited to a single pushbutton and one LED for use with a test program to verify proper operation. A large variety of I/O devices, however, are available on the DI01 board attached to theD2XL by means of two 40 pin connectors. Our two experiments will exploit the features of the D2XL/DI01 combination to design a number of circuits that will demonstrate the usefulness of this procedure.

Getting Started

Hooking up the Hardware

The circuitry is extremely delicate and can easily be destroyed if handled improperly. Static electricity which is easily generated is particularly dangerous and care must be taken to wear a grounded wrist strap when handling the circuitry. Your instructor will show you how to use it properly. Your two boards should be connected to one another, with power cord installed and a programming cable from the parallel port of the PC to the JTAG connector of the D2XL attached. Ask your instructor for help if this is not the case.

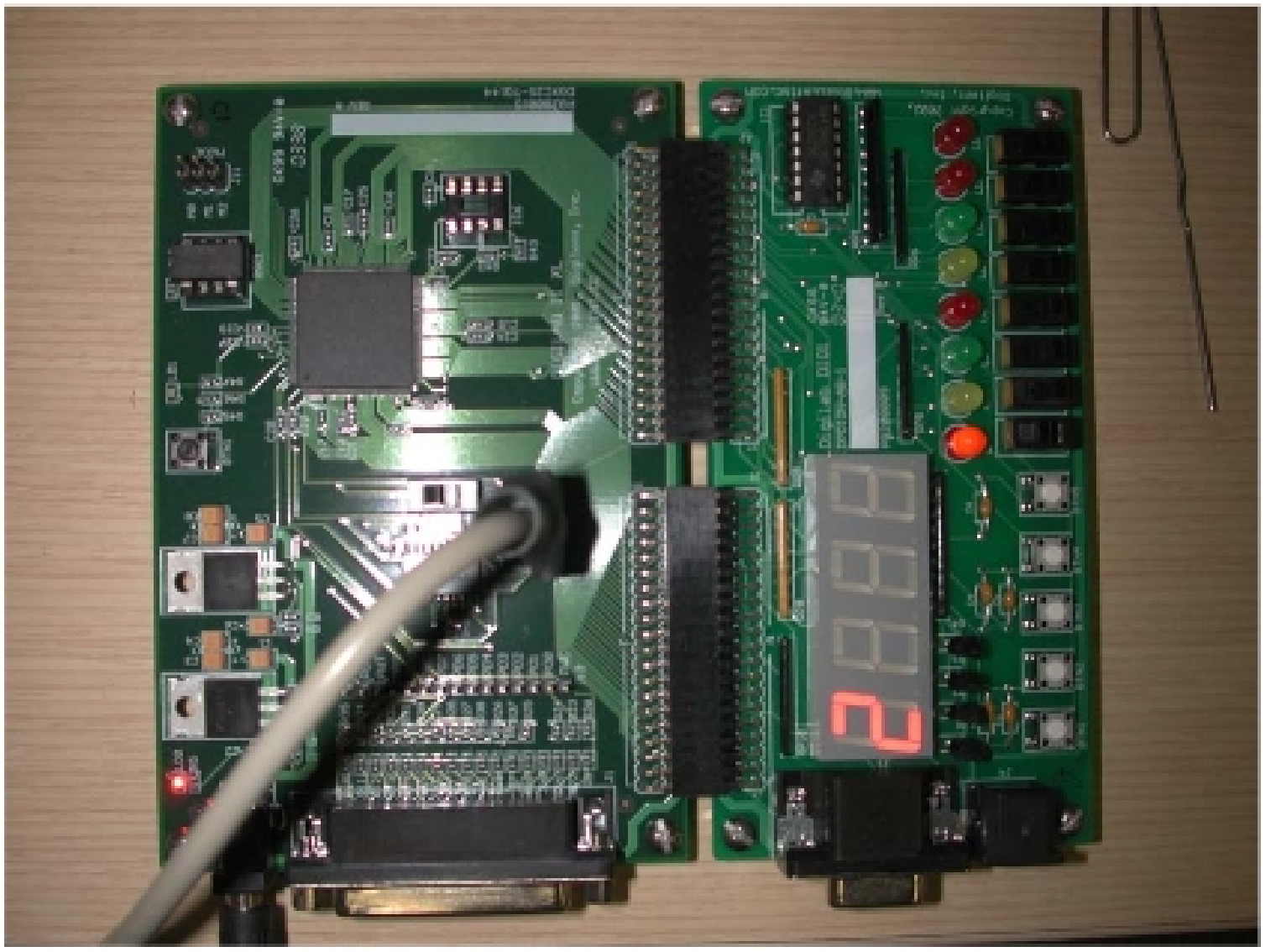

Figure 1: The D2XL and DI01 boards with programming cable attached.

Testing the D2XL board

You should have a Xilinx ISE 9.2i icon on your screen. Double click on it to open the program. As a first program to download we want to use "Di01Demo"(C:/Digilent/Di01Demo.ise) to test the integrity of the D2XL board and the attached Digital I/O 1 board. If another project comes up, close it from the "File" menu and use "Open Project"

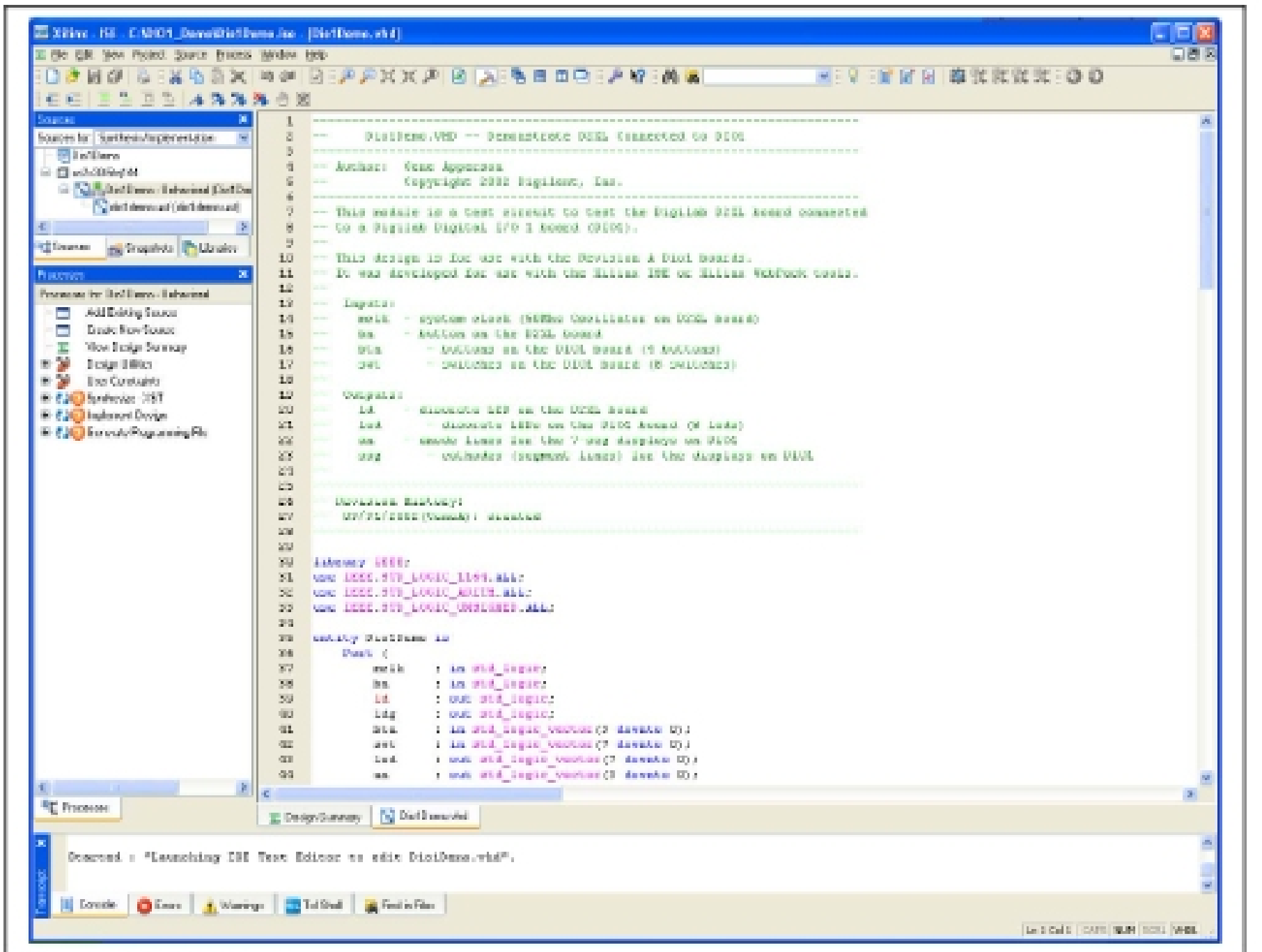

Figure 2: Project Navigator

from the same menu to open "Di01Demo". If you are successful you should get a screen display like Figure 2.

You need to double-click on the 3rd line in the top left window to get the display in the top right window. These four windows represent the design environment for the project with "Sources" in the top left window, the "Processes" for a given Source below it, the contents of selected files in the right hand window and text files below. On top of the "Sources" window is a tab to select the type of sources to be displayed. In this instance the sources are those associated with "Synthesis/Implementation" but we will also be interested with sources for "Behavioral Simulation". Before proceeding we need to check that we have selected the proper chip and simulation software. If you right-click the