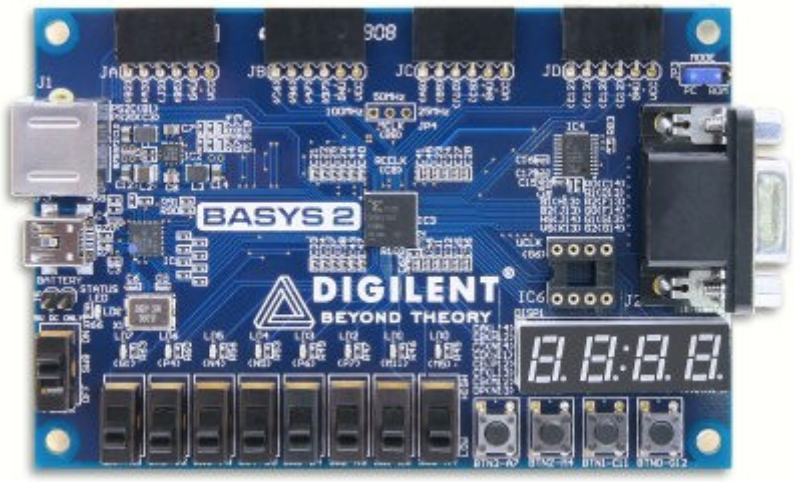
METU
CENG232 Logic Design
FPGA Board User Manual (BASYS2)

Revision: 1.0



1. Introduction

This document aims to familarize you with the FPGA Board hardware, its integrated development environment for programming verilog, and additional tools for aid in programming the device.

2. Handling precautions

When operating the FPGA Board, ESD (Electrostatic Dischange) can cause upset as well as damage to the board components. Therefore, apply ESD prevention measures whenever operating the FPGA Board. Observe the following guidelines:

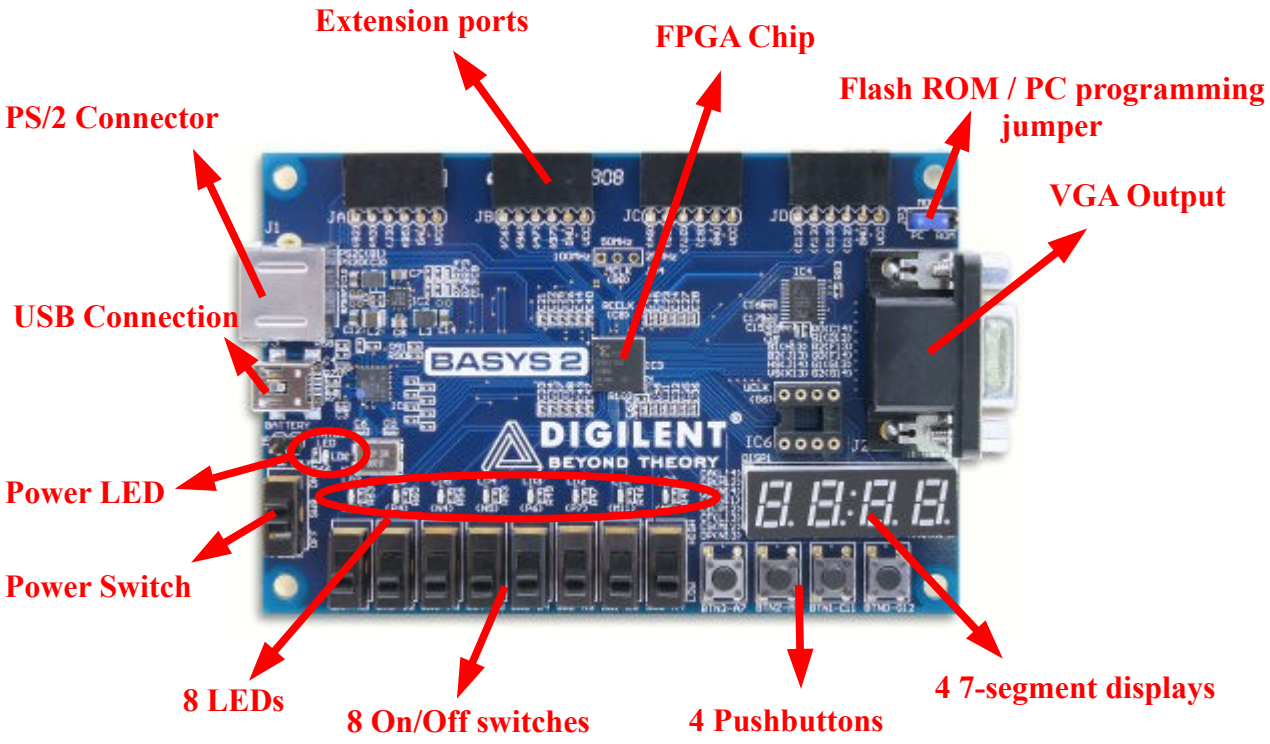
- **Do not touch exposed traces or components on the board, especially while the board is powered on.**
- **Exercise caution when manipulating switches, buttons and other controls while the board is powered on.**
- **Hold the board from its edges.**
- **Use your common sense.**

3. Board

The FPGA Board we will use is a Digilent BASYS2. It contains a Xilinx Spartan3E-250 FPGA chip at its center, with various inputs and outputs. The board is powered and programmed through USB.

Every time the board is powered, the FPGA chip needs to be programmed&initialized from scratch. The board features a Flash ROM, which can store a programming image file while the board is powered off. This image is used to program the FPGA chip at power-on.

The FPGA chip, and the Flash ROM can be programmed from provided software separately. The Flash ROM contains a demonstration image file, which can be used to test the features of the board.



3.1. Main usage

3.1.1. USB Connection

Used for powering the board, and programming the FPGA chip. *Only connect/disconnect the USB cable when the board is powered off from its power switch.*

3.1.2. Power switch & Power LED

Only connect/disconnect the USB cable when the board is powered off from its power switch.

3.1.3. Flash ROM / PC Programming jumper

Only change it while the board is powered off. If set to ROM, the board will be initialized from the demonstration image stored in the flash chip. In order to program the chip from USB cable, it has to be set to PC *(Although programming the chip while the jumper is on ROM is possible, it may lead to incorrect behavior, so avoid doing it)*

3.2. Inputs

The board provides inputs to the FPGA through 8 switches, 4 pushbuttons, and a PS/2 keyboard/mouse connector.

3.2.1. On/Off switches

3.2.2. Pushbuttons

3.2.3. PS/2 connection

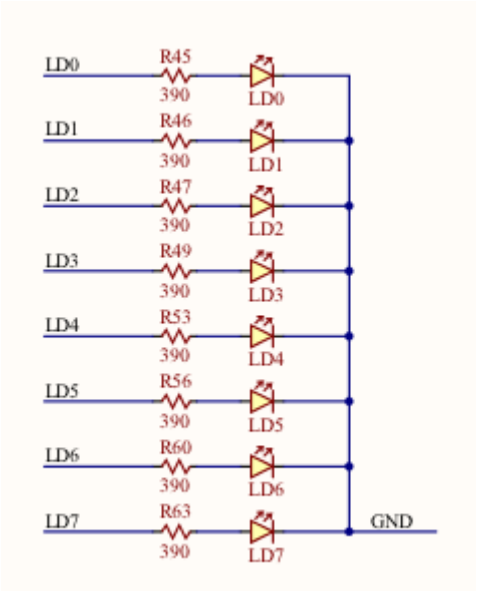
PS/2 connector is directly connected to the FPGA chip, and it is the programmer's responsibility to handle its signals properly. Do not connect anything unless the chip is properly programmed.

3.3. Outputs

The board provides outputs using 8 leds, 4 7-segment displays, and a VGA output.

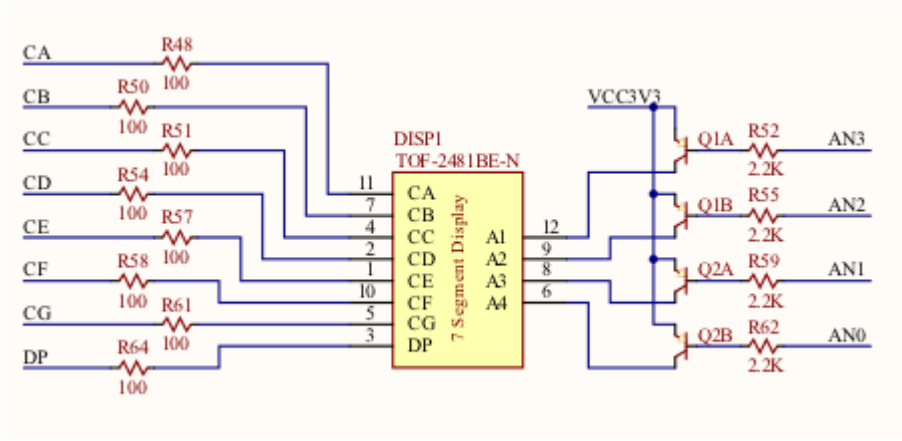
3.2.1. LEDs

The LED outputs are accessed individually. Providing Logic high to a LDx output sets a LED on, Logic low sets a LED off.



3.2.2. 7-segment Displays

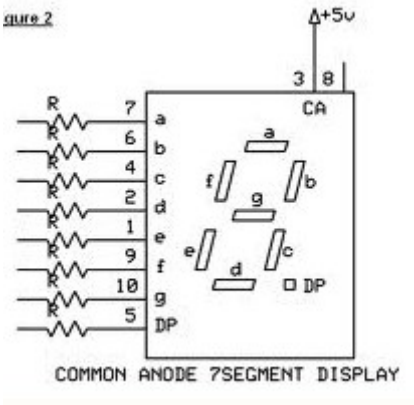
The 7-segment displays are of common anode type. There is a single bus to drive the individual LEDs, and 4 displays' anodes are controllable.

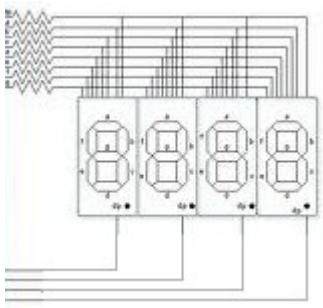


AN0..AN3 control the anodes of individual 7-segment displays. Providing logic-low will enable the display, and logic-high will disable the display.

CA..CG and DP controls the individual segments within a 7-segment display. DP controls the dot. Providing logic-low will enable the segment, logic-high will disable the segment.

e.g. To turn on only the E segment of only the 2nd display, AN1 must be logic low (and AN0 AN2 and AN3 must be high), and CE must be logic low (and other Cx are logic high).





If you set all of the AN0...AN3 values to logic low at the same time, then all 7-segments will be enabled. But, they will display the same value as they share their CA...CG and DP inputs.

In order to display different values in different 7-segments, you should do time multiplexing. (i.e. Only turn on the 1st display for ¼ seconds, then only turn on the 2nd display for ¼ seconds, ... if you do it fast enough, human eye won't notice the flickering effect of the displays)

3.2.3. VGA output

The board provides 8 bit color output (3 bits red, 3 bits green, and 2 bits blue), and gives direct access to horizontal sync and vertical sync outputs. It is the programmer's responsibility to provide correct timing on the HSYNC and VSYNC outputs for use with a monitor.

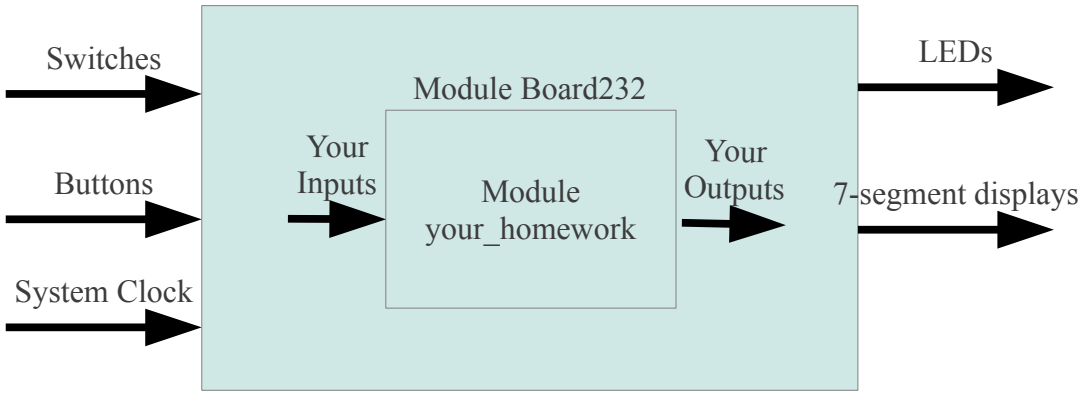
Please refer to the official BASYS2 board documentation for VGA usage.

4. Xilinx

You will use Xilinx WebPACK software as a verilog environment. Please follow the course material for which version to use. Xilinx WebPACK works both in Windows and Linux environment.

4.1. Board File

Writing a verilog program for the BASYS2 board requires some project configuration and a master verilog file to route data between board i/o and your homework files. You will be provided with this project directory; ceng232_board.zip



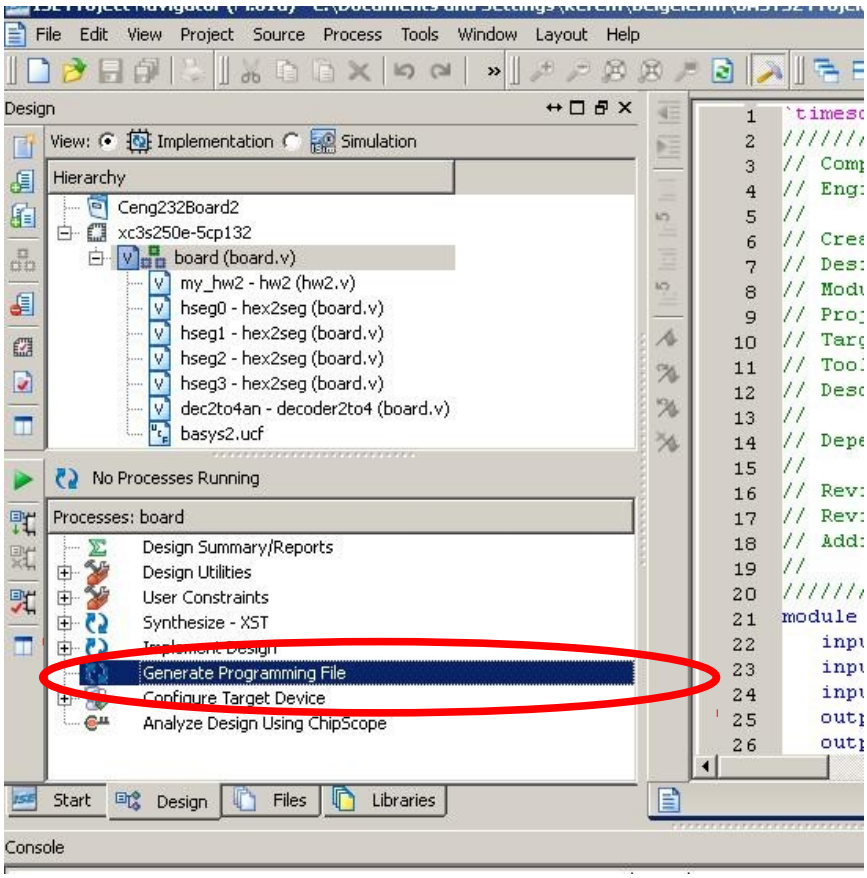
Additionally, a slightly modified board file board232.v will be provided for each laboratory assignment according to the homework specifications. This will ensure correct switches and buttons get routed to correct inputs of your modules.

In 232 laboratories, you will not deal directly with board inputs and outputs – although configuring and using them is quite simple.

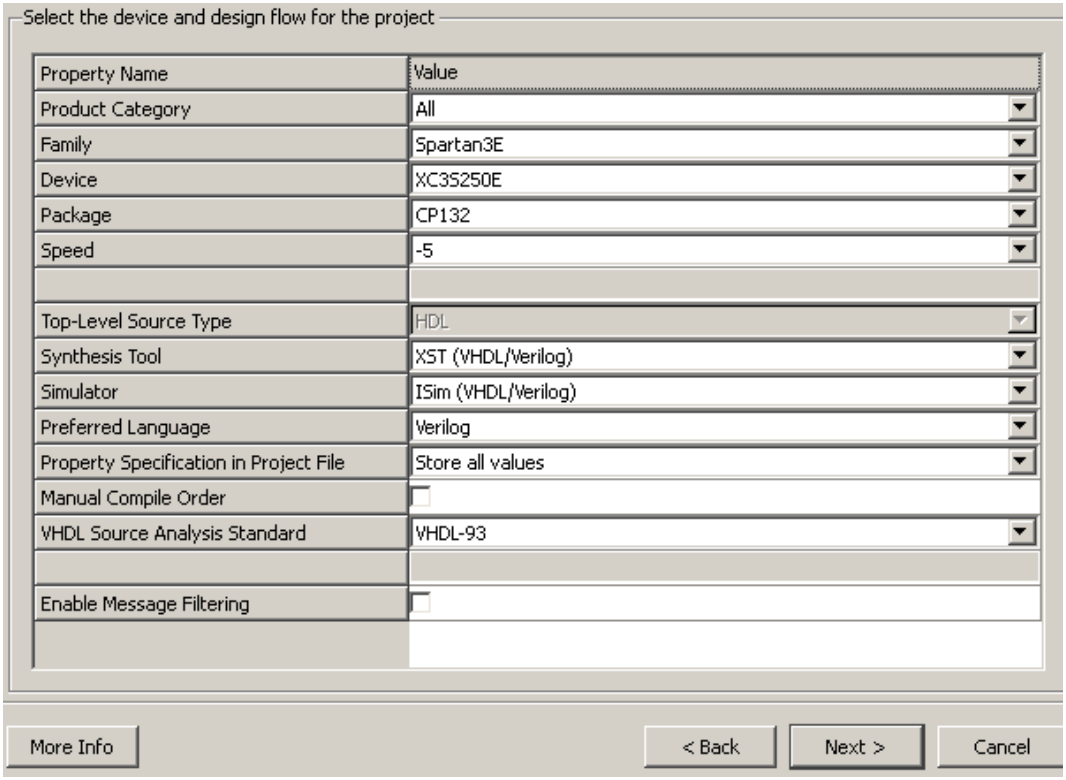
4.2. Compiling BIT file

You have to compile your verilog project into a BIT file. This binary BIT file will be uploaded to the FPGA chip, which will then get executed on the board.

To compile, make sure you are in the “Implementation” view. Select the master module (Board232 module), and execute “Generate Programming File” action. This will generate a *.bit* file in the project directory.



4.3. Creating a new project



If you need to create a new project for the board, make sure you select the above options.

After creating the project, you should enter the project settings and do a Clock adjustment. From the implementation panel, select the top level module. Right click on “Generate Programming File” action, and enter “Process Properties”. A new window will appear. Select “Startup Options” category from the left. On the right side, you should set the **“FPGA Startup Clock”** parameter to **“JTAG CLOCK”.**

Also, to access board's inputs and outputs, you will need to use contents of the definition file (*basys2.ucf*) from the provided sample project. This file contains information on how the FPGA chips pins are interconnected with input/output devices.

5. Uploading BIT file to FPGA

You will use Digilent Adept 2 software for programming the FPGA. Digilent Adept works only in Microsoft Windows. You can download the software from this link: http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,66,828&Prod=ADEPT2 (Download “Adept 2.9.4 System”, or newer version if available)

Installing Digilent Adept 2 will install necessary Windows drivers for the FPGA board.

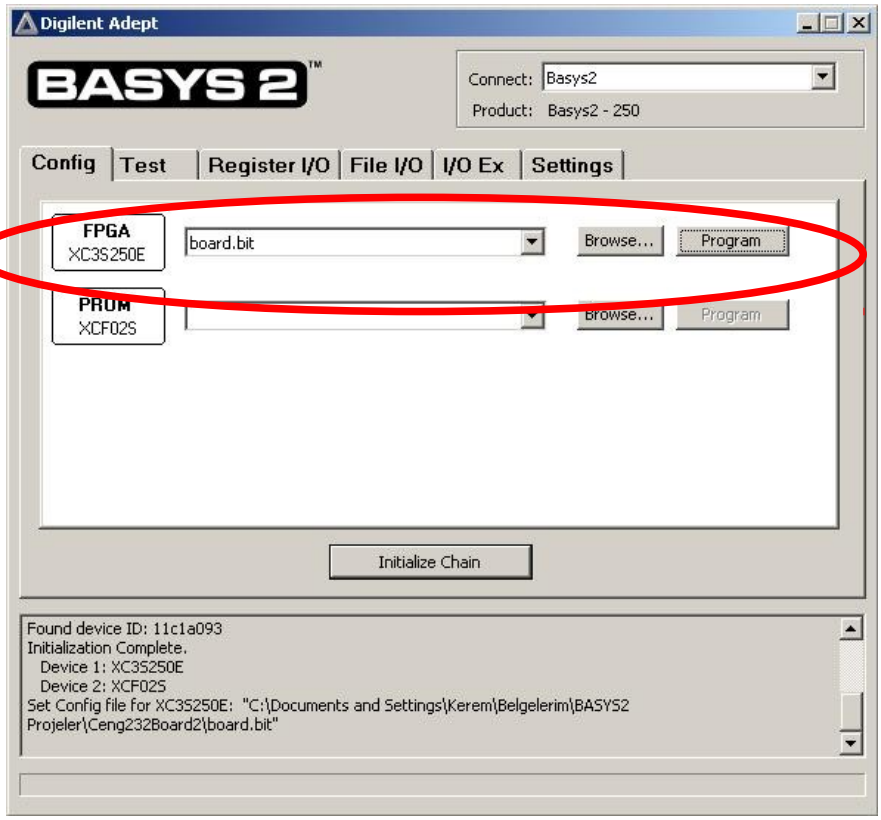
Digilent Adept can be used to program both the FPGA and the on board flash ROM. For the course purposes, only program the FPGA and leave the ROM image intact.

1. Make sure **PC/ROM** jumper is set to **PC**
2. Connect the board to PC
3. Switch the board ON
4. Open Digilent Adept 2, it should recognize “BASYS2”
5. In the “Config” tab, choose the bit file for the **FPGA** part

   *Do not load a bit file to **PROM** part, this configures onboard flash ROM*
6. Click on Program button

   - After programming is complete, your software will start running on the board

   - You can go to step-5 at any point to reload another bit file


Digilent Adept
BASYS 2
Connect: Basys2
Product: Basys2 - 250
Config
Test
Register I/O
File I/O
I/O Ex
Settings
FPGA
XC3S250E
board.bit
Browse...
Program
PROM
XCF02S
Browse...
Program
Initialize Chain
Found device ID: 11c1a093
Initialization Complete.
Device 1: XC3S250E
Device 2: XCF02S
Set Config file for XC3S250E: "C:\Documents and Settings\Kerem\Belgelerim\BASYS2 Projeler\Ceng232Board2\board.bit"