DAT093
Introduction to Electronic System Design

Laboratory assignment 4

Structured design

Sven Knutsson
svenk@chalmers.se
Dept. Of Computer Science and Engineering
Chalmers University of Technology
Gothenburg
Sweden

Goal

Learn how to create a number of blocks of different kind and put them together into one functional design

The blocks we will use are components and functions

We will also synthesize the result into a FPGA and test it on a development board

Test and simulation

We will test your design by running the design that you have downloaded to the development board.

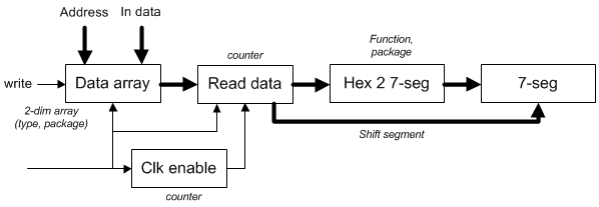
Since you have some freedom in setting up your design there are no testbenches or do-files for the whole design or for the individual blocks.

To do simulation you need to write your own testbenches and do-files

Assignment

Create a system where we can store a number of data words and at the same time display these words on 7-segment displays

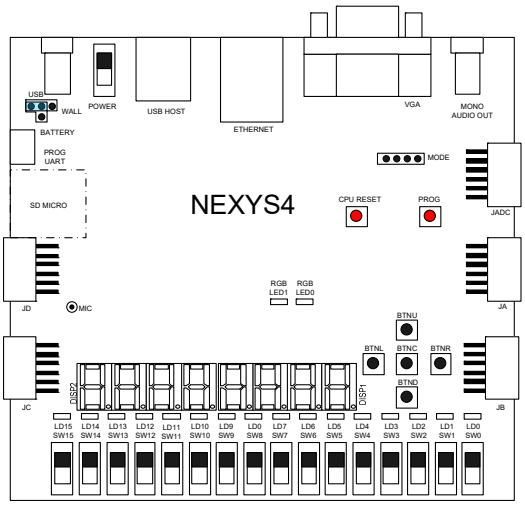
Block diagram



Let´s look at the FPGA board

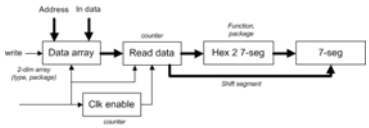
Nexys4 Development Board

The development board is called Nexys4 and contains a Artix-7 FPGA.



Let´s look at what we need from the board

Board resources



The data array should be able to store a number of words.

- We need to define a two dimensional array, a memory, for this.

  The type declaration for the array should be placed in a package.

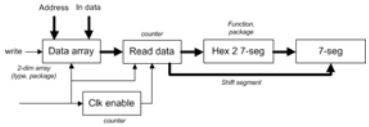
Since we have eight 7-segment displays we need eight positions in the memory.

Since a 7-segment display takes sixteen data values (0x0 – 0xF) we need four bits for the data words, a nibble.

- We need a way to input data and a way to write it to a given place in memory.

  We use the switches and pushbuttons for this.

The data array cont.



We use four switches for the data value and three for the memory address.

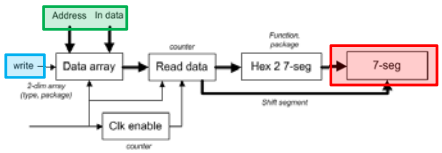
- We need a way to write data to the memory.

  We use one of the push buttons for this.

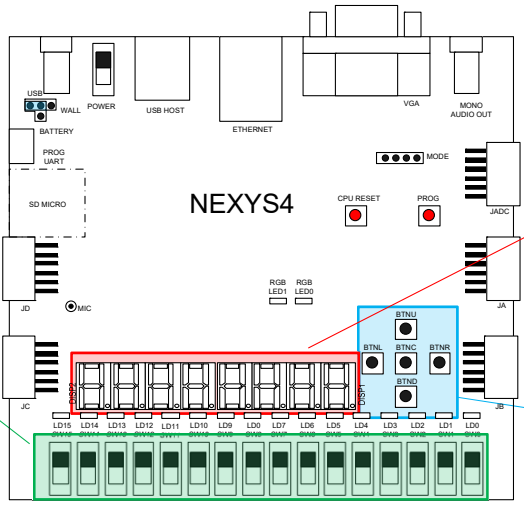
It´s up to you to decide what switches and pushbutton to use.

Please document the allocation so the testing gets simplier

Nexys3 Development Board 2



Let´s look at the blocks

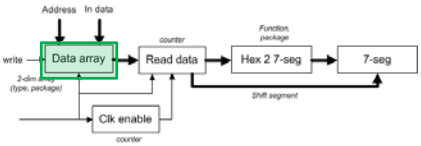


7-segment displays to show the data

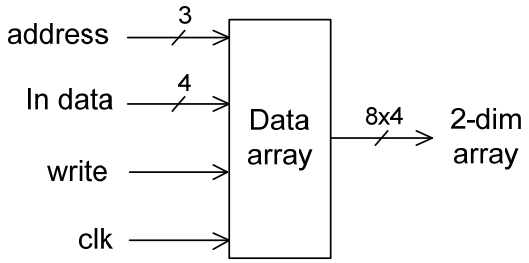
Switches to set the address and data

Push buttons to write the data

The data array



We have the interface for the data array block



The block doesn´t have to be synchronious, controlled by a clock, but synchronious designs usually work better.

You decide!

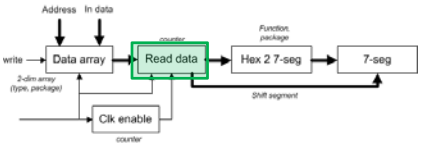
Read data

We need a block that can continously read the words from the array, one by one, and display them on consecutive 7-segment digits.

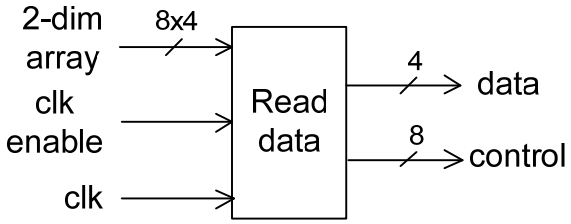
The 7-segment display works by using a common data input and then we activate the digit(s) in which we want to display the current data.

This means that the Read data block should read data one by one from the two dimensional array, output the data to the displays and at the same time activate the digits in a sequence and then repeat this in a circular fashion

Read data cont.



We have the block diagram

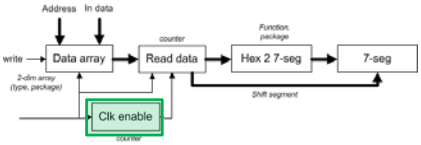


To handle the data one by one we need a clock.

We also need a clock enable signal to slow down the sequence.

There is a paper called Hints on clocking on the homepage where you can read about clock enable.

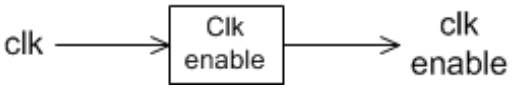
Clk enable



The clock enable block decides the rate of the update of the display.

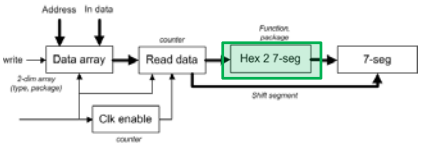
The update rate should be fast enough to make the 7-segment digits appear to be lit all of the time and not flicker but at the same time slow enough to lit the LEDs properly

Experiment with different rates!



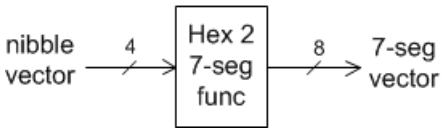
We need a counter to create the clock enable signal

Hex to 7-segment



The 7-segment displays can´t display 4 bit data directly. The data must be converted from binary code into 7-segment code. There´s also a dot so we have eight bits in total

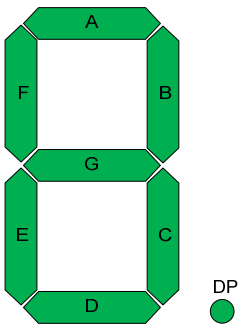
You should create a function, not a component, that does this conversion. The function should be placed in the same package as the earlier type declaration.



The 7-segment display can be of two different types and this dictates the implementation.

7-segment displays

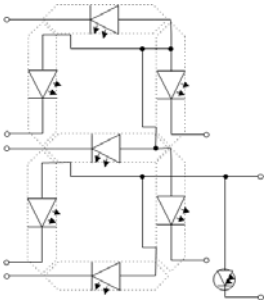
The seven segment digit really has eight bits, the last bit is a dot



Each segment is formed by a LED

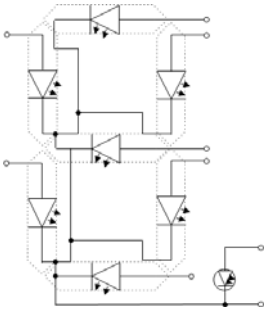
7-segment displays

The LEDs can be connected with a common anode or a common cathode



The segments are addressed by low (inverted) data

The digit is activated by a high signal



The segments are addressed by high data

The digit is activated by a low (inverted) signal

The development board

The development board is called The Nexys4 Trainer board.

You can find the Nexys4 Board Reference Manual on the course home page and this is where you find out about the slide switches, push buttons and 7-segment displays.

To implement the design on the board you will have to assign the signals to and from your VHDL design to the correct pins on the development board.

The development board cont.

This is done by using a constraints (XDC) file. You can find a frame work for this file on the course home page.

There are two versions of the Nexys4 board that look almost the same but differ in the type of memory used. The two will need different XDC files.

One of the boards is called Nexys4, the other one is called Nexys4DDR.

The constraints file for the Nexys4 Board is called Nexys4_Master.xdc while the constraints file for the Nexys4DDR Board is called Nexys4DDR_Master.xdc.

There is a document called Introduction to Xilinx Vivado on the homepage where you can read about how to use the XDC file

The UCF file

Let´s end this presentation by looking at the XDC files

Demonstration