# Assignment 3

**Due October 3, 1996**

*In this assignment you'll design a circuit with VHDL, synthesize it into an FPGA configuration bitmap and test your design by downloading the configuration file to an FPGA on a demonstration board.*

## Question 1

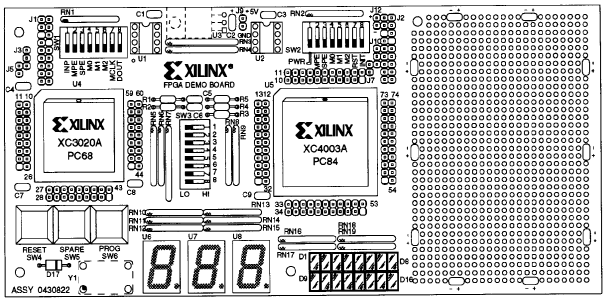
### Circuit Description

The circuit should display a sequence of values on a 7-segment LED display starting with a blank display and followed by the 8 digits of your student number. The display should advance to the next value each time a switch is pressed. The initial (all-zero) state of the circuit should result in a blank display. After all 8 digits have been displayed the circuit should cycle back to the blank state (i.e. there are a total of 9 states).

### Demonstration Board

You will implement the circuit by configuring an FPGA on a demonstration board. This demo board has two Xilinx FPGAs, three 7-segment LED displays and various pushbuttons and switches. You will use the XC4003APC84-6 FPGA, the rightmost LED display (U8), and the pushbutton marked "SPARE" (SW5). The configuration information will be downloaded over a serial port from a workstation.

The following diagram shows the layout of the components on the demo board:




The pushbutton marked "SPARE" is connected to pin 18 of the FPGA and is active-low (when the button is pressed the signal is a logic 0). This means your circuit should change state on the *falling* edge of the clock.

The LED segments 'a' through 'g' located as shown on the diagram on page 2 of Lecture 2 (*Combinational Logic Design*) are connected to pins 49, 48, 47, 46, 45, 50, and 51 respectively. Note that these outputs are also active-low so you need to output a '0' in order to turn on a particular segment.

### Mapping Ports to Pins

Normally the FPGA place/route software assigns pins to ports automatically. However, if a PCB has already been built (as in this case) then it's necessary to map ports to specific pins by supplying the mapping to the place/route software. With the Synopsys/Xilinx tools this can be done by embedding `set_attribute` DC (Design Compiler) commands into the VHDL source using the `dc_script_begin` pragma as shown in the entity statement below.

The DC script contains two other commands: (1) the `set_port_is_pad "*"` command marks all ports as being connected to device pins ("pads"), and (2) the `set_pad_type -no_clock clk` is required because normally DC assumes that clock ports will be connected to one of the special global clock input pins and, unfortunately, pin 18 is not one of these special clock input pins.

```
entity snum is
   port ( clk : in bit ;
      a,b,c,d,e,f,g : out bit ) ;

-- pragma dc_script_begin
-- set_port_is_pad "*"
-- set_attribute clk pad_location -type string "P18"
-- set_pad_type -no_clock clk
-- set_attribute a pad_location -type string "P49"
-- set_attribute b pad_location -type string "P48"
-- set_attribute c pad_location -type string "P47"
```

```
-- set_attribute d pad_location -type string "P46"
-- set_attribute e pad_location -type string "P45"
-- set_attribute f pad_location -type string "P50"
-- set_attribute g pad_location -type string "P51"
-- pragma dc_script_end

end snum ;
```

The entity description above is available in the file `~elec464/asg3ent.vhd`.

## Instructions

Write a VHDL description using the entity statement above that implements the circuit described above.

Follow the instructions given on the Web page for synthesizing your design into a Xilinx FPGA configuration (.bit) file (ignore the warning "i/o pad attribute mismatch on port clk" – this is a result of the `set_pad_type -no_clock clk` command).

Log into a workstation machine that has a demo board attached and use the XChecker program to download the configuration file.

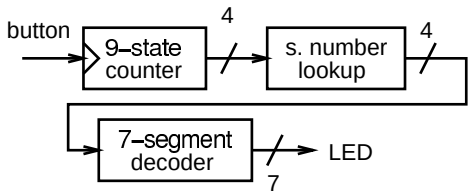
Test your design by pushing the SPARE button and making sure the display sequences through the appropriate states (for *your* student number!).

Submit your VHDL code and the `.bit` file electronically (see instructions on the Web page). The TA will use the .bit file you submit to check your design. Use the key `3vhd` to submit the VHDL code and the key `3bit` to submit the `.bit` file.

Please read and heed the warning on "Run Times" in the instructions and allow plenty of time to synthesize and place/route your design.

## Design Hints

You may want to break up your design into three sections as shown in the diagram below: a counter that sequences through 9 values, a circuit that converts the counter value to the proper digit of your student number, and a circuit that converts digits to the 7 values required to display that digit on the LED.




The following table shows which of the 7 segments should be turned on to generate the different digits.

| digit | a | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 3 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 5 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 6 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 7 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 9 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| blank | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## Simulation

A test bench is available in `~elec464/asg3tb.vhd`. This test bench prints the state of the 7 LED segments and the corresponding digit (if any) after each of 10 clock cycles. Use the same procedure for analyzing and simulating your code and the test bench as in the previous assignment. The pragmas in the comments will be ignored by the simulation software.

## Access to the Demo Boards

The demo boards will be connected to a few of the SPARC workstations in the third-floor room next to the VLSI lab. Note that you can use telnet to log in to the workstations that have the demo boards attached to them even if someone else is using the display.

Please make sure you `exit` the XChecker program as soon as you have tested your design so that other people can use the serial port and demo board.

## Further Information

Although the above information should be sufficient to complete the assignment you can get more information on the demo board, including schematics, from the file `/usr/applic/XSI/online/online/hardware.pdf`.