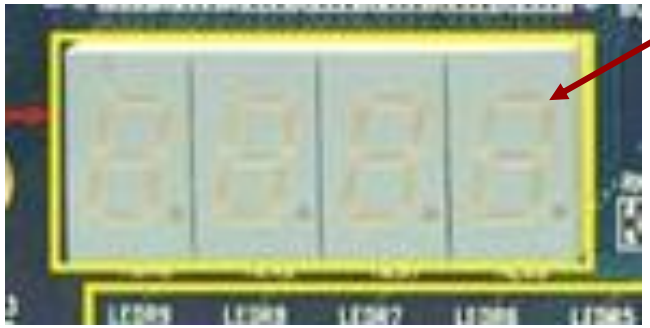
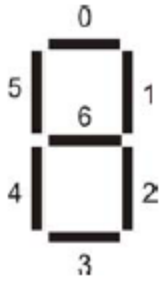
ECSE-323
Digital System Design

Lab #2 – *Combinational Circuit Design with VHDL* **Winter** 2018

Design of the 7-Segment LED decoder/driver .

A 7-segment LED display has 7 individual light-emitting segments, as shown in the picture below. By turning on different segments at any one time we can obtain different characters or numbers. There are four of these attached to the Cyclone FPGA chip on the Altera board (which you will be introduced to in Lab 3), which you will use later in your full implementation of the time-stamp system.





numbering of the LED segments (from Altera DE1 board manual)

In this part of the lab you will design a circuit that will be used to drive the 7-segment LEDs on the Altera board. It takes in a 4-bit code and generates the 7-segment display associated with the input code.

The outputs should be made active-low. This is convenient, as many LED displays, including the ones on the Altera board, turn on when their segment inputs are driven low.

Your circuit should have 2 ***modes***, selected by a 1-bit mode input:

Mode = 0 indicates that the display should interpret the 4-bit input as a hexadecimal number (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)

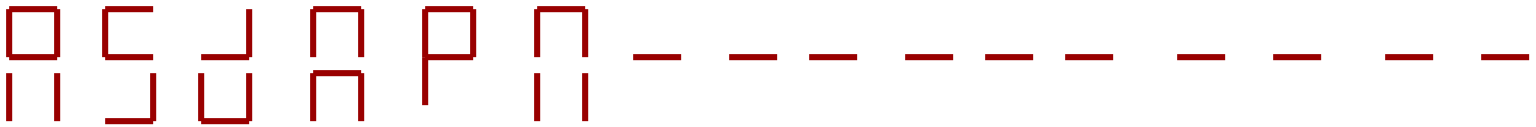
Mode = 1 indicates that the display should interpret the 4-bit input as a symbol value (A,S,d,A,P,M,-,-,-,-,-,-,-,-,-,-,-,-)

Display the symbols on the 7-segment display as shown below:

Code = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Mode = 0

Mode = 1



Use the following entity declaration for your design:

```
entity gNN_7_segment_decoder is
 port ( code         : in std_logic_vector(3 downto 0);
         mode        : in std_logic;
        segments_out         : out std_logic_vector(6 downto 0));
end gNN_7_segment_decoder;
```

To implement the 7-segment LED decoder, use a single selected signal assignment statement (with 32 cases).

To get you started, here are the first few lines of the architecture body. You fill in the rest!

```
architecture behavior of gNN_7_segment_decoder is
        signal xcode: std_logic_vector(4 downto 0);
begin
        xcode(4 downto 1) <= code;  --first 4 bit of xcode is code
        xcode(0) <= mode; --last bit of xcode is mode
        -- select segments_out based on xcode
        with xcode select
                segments_out <=
                "1000000" when "00000", -- code=0, mode=0
                "0001000" when "00001", -- code=0, mode=1
                "1111001" when "00010", -- code=1, mode=0
                -- ...more cases...
                “0111111" when others; --default case "-",  end with semicolon
end architecture ; -- behavior
```

CREATE THE DESIGN FILE AND SYMBOL

Once you have written the VHDL description, analyze it (using the ***Processing/Analyze Current File*** menu item, to check for errors.

When your design is error-free, create a symbol for it, and insert an instance into the top-level project file.

Show your completed VHDL description to your TA.

In preparation for simulation, compile the design for your 7-segment decoder circuit using the ***Processing/Start Compilation*** menu item.

SIMULATE THE DESIGN

Once compilation has been successfully completed do a functional simulation. This simulation should test ***all 32 possible*** input patterns of the input code.

As before, in order to carry out the simulation, you should insert an instance of the *gNN_7_segment_decoder* symbol into your top-level project schematic diagram, and hook up inputs and outputs as needed.

Show the TA the results of your simulation.

. Writeup of the Lab Reports .

Write up a report, describing ***gNN_7_segment_decoder*** circuit.
The reports must include the following items:

- A header listing the group number (and company name if you gave it one), the names and student numbers of each group member.
- A title, giving the name (e.g. ***gNN_7_segment_decoder***) and function of the circuit.
- A description of the circuit's function, listing the inputs and outputs. Provide a pinout or symbol diagram.
- A gate-level schematic diagram of the circuit (can be generated with Quartus Tools ->Netlist Viewers->RTL viewer).
- A complete discussion of how the circuit was tested, showing representative simulation plots, and detailing what test cases were used.
- The VHDL description of the circuit (don’t embed this in the text of the report, instead include it as a separate file in the assignment submission zip file).

The lab report, and all associated design files must be submitted, as an assignment to the myCourses site. Only one submission need be made per group (both students will receive the same grade!).

Combine all of the files that you are submitting into one *zip* file, and name the zip file gNN_LAB_2.zip (where NN is your group number).

The reports are due one week after the last day of the lab period.

Grade Sheet for Lab #2 Winter 2018 __

Group Number:_____.

Group Member Name:_____. Student Number:_____.

Group Member Name:_____. Student Number:_____.

Marks			TA Signatures
	1.	VHDL code for the Input Circuit VHDL	.
	2.	Block Diagram Sketch of the Input circuit	.
	3.	Simulation of the *Input* circuit	.
	4.	VHDL code for the *7_segment_decoder* circuit	.
	5.	Simulation of the *7_segment_decoder* circuit	.
	6.	Demo of working input circuit and display mode "0"	.
	7.	Demo of working input circuit and display mode "1"	.

Each part should be demonstrated to one of the TAs who will then give a grade and sign the grade sheet. Grades for each part will be either 0, 1, or 2. A mark of 2 will be given if everything is done correctly. A grade of 1 will be given if there are significant problems, but an attempt was made. A grade of 0 will be given for parts that were not done at all, or for which there is no TA signature.