Instructions

Objective

In this lab, you will develop a 7-segment decoder which can convert a 4-bit signal into a corresponding hexadecimal digit for display using the DE2-115's 7 segment displays. You will integrate your code with the comet code provided in the lecture slides in the same design, using the 7 segment displays to indicate which of the 26 LED's is illuminated (for example, when the rightmost LED is illuminated, the 7 segment displays should show `01`, and when the leftmost LED is illuminated, the 7 segment displays should show `1a`). To make sure all 8 displays are working, the number will be repeated 4 times (once on each pair of 7 segment modules).

TIP: your 7 segment decoder will be used for future labs. You should build it as a standalone module in a separate Verilog file from your top-level module.

TIP: the terms "7 segment display" and "hex display" may be used interchangeably in this and future labs

TIP: the 7 segment displays are active low.

Specific Tasks

1. Setup the FPGA tools for your Linux account
    - (add `source /usr/local/3rdparty/cad_setup_files/altera.bash` to your .bashrc file and open a new terminal)
2. Download and extract the project skeleton.
3. Run the shell command `./csce611.sh compile && ./csce611.sh program` to make sure your DE2-115 board is working properly. You should see a bouncing "comet" on the 26 LEDs. If not, or if the script fails, ask a TA for help.
    - You can use this same command to re-compile your project and program the board again at any time.
    - If you see a message like `./csce611.sh: line 47: QUARTUS_ROOTDIR: unbound variable`, then you have not set up your shell session to access the Quartus tools properly. Try running `source /usr/local/3rdparty/cad_setup_files/altera.bash`
4. Develop your 7 segment decoder. This will be a standalone Verilog module which takes one 4 bit input, and has one 7 bit output. You can find information about the 7 segment display in the DE2-115 user guide.

    - **TIP**: a skeleton for a decoder is provided in the file `hexdriver.v`. You are not required to use it, but may find it a helpful starting point.

5. Instantiate 8 instances of your decoder, one for each 7 segment display.
6. Devise a method to determine at any given time what number should be displayed on the 7 segment displays and implement it either in the top-level module or as a separate module according to your preference. There are several ways to accomplish this.
    - We group the 7 segment displays into groups of 2, since the numbers 0 to 26 require two hexadecimal digits to display.
7. Your board should still show the bouncing comet while also simultaneously updating the hex displays.

Deliverables

- You will turn in **one** file, it should be named `CSCE611_Fall2019_comet_yourusername.7z`. You should replace `yourusername` with your actual username that you use to log into the Linux lab computers with. The script will automatically generate a correct file name for you – you usually will not need to modify it.
    - You must turn in your code for this project. Your submission **must** be packed by using the command `./csce611.sh pack`.
    - See also: Appendix 1.
    - **TIP**: `./csce611.sh pack` must be executed in a graphical X session, if you are accessing the Linux labs over ssh, be sure to use `-X` or `-Y`. If you don't know what this means, ignore this bullet point.
- Each group only needs to submit once via either partner.

Rubric

- (A) 30 points – board shows bouncing comet on the LEDs
- (B) 30 points – hex displays working
    - (B.1) 16 points – hex displays correctly show the digits 0 to F (one point per digit).
    - (B.2) 14 points – hex displays show the expected values as the comet bounces.
- (C) 30 points – comet and hex displays operate simultaneously.

Maximum score: 90 points.

Project structure

The structure of the provided skeleton is as follows:

- `comet_system_builder_settings.cfg`: Terasic System Builder settings used to generate the project file.
- `CSCE611_lab_comet.htm`: log of pin assignments used for the DE2-115 board
- `CSCE611_lab_comet.qpf`: Quartus Project File for this project
- `CSCE611_lab_comet.qsf`: Quartus Script File for this project – this is where project settings such as device, top-level entity, and pin mappings are configured.
- `CSCE611_lab_comet.sdc`: Synopsis Design Constraint file defining information about the system clock.
- `CSCE611_lab_comet.sv`: Top-level System Verilog module for the project.
- `hexdriver.sv`: empty module to write your hex decoder in should you so choose
- `csce611.sh`: front-end wrapper for course-specific scripts.
- `scripts/`: course-specific scripts.

You should not directly modify any of the above files except for `CSCE611_lab_comet.sv` and `hexdriver.sv`. The QPF and QSF files may be updated using the Quartus GUI or TCL commands. You may add additional Verilog files if you would like.

Appendix 1 – Example Output of `csce611.sh pack`

This shows an example of a terminal session where an assignment was successfully packed.

```
$ ./csce611.sh pack
Gtk-Message: GtkDialog mapped without a transient parent. This is
    discouraged.
Gtk-Message: GtkDialog mapped without a transient parent. This is
    discouraged.
cleaned simulation files
attempt 1/10 to clean project... OK
cleaned hardware build files

7-Zip [64] 9.20  Copyright (c) 1999-2010 Igor Pavlov  2010-11-18
p7zip Version 9.20 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,8 CPUs)
Scanning

Creating archive /acct/cad3/CSCE611_Fall2019_comet_cad3.7z

Compressing  lab_comet/README.md
```



```
Compressing  lab_comet/CSCE611_lab_comet.htm
Compressing  lab_comet/csce611.sh
Compressing  lab_comet/scripts/jtag.sh
Compressing  lab_comet/scripts/listfiles.tcl
Compressing  lab_comet/scripts/help.txt
Compressing  lab_comet/README.pdf
Compressing  lab_comet/comet_system_builder_settings.cfg
Compressing  lab_comet/CSCE611_lab_comet_assignment_defaults.qdf
Compressing  lab_comet/CSCE611_lab_comet.qpf
Compressing  lab_comet/CSCE611_lab_comet.qsf
Compressing  lab_comet/CSCE611_lab_comet.qws
Compressing  lab_comet/CSCE611_lab_comet.sdc
Compressing  lab_comet/CSCE611_lab_comet.sv
Compressing  lab_comet/hexdriver.sv

Everything is Ok
```

