ECEN3002 - Homework 13

Fall, 2020

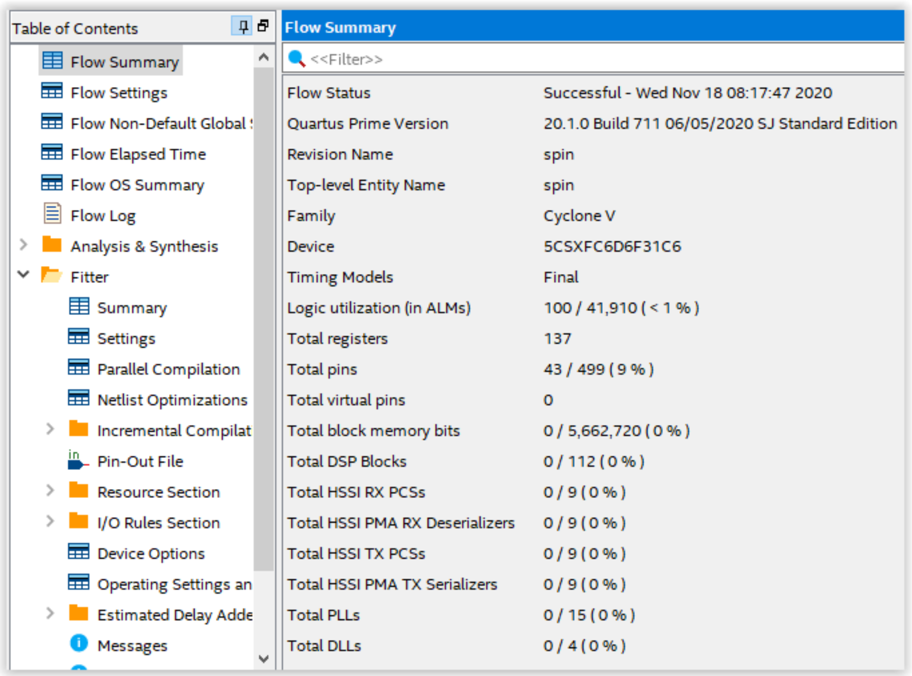
Due Monday November 30, 11:59pm – 30 Points

The purpose of this homework is to become familiar with use of Tcl scripting in Quartus design flows.

After completing this assignment, submit the files highlighted in red in this document.

Part 1: Understanding Quartus Reporting

When you finish running a Quartus project compilation, a Table of Contents in the center of the Quartus GUI lists all the various reports that are available. Some are useful, some are not. Some information may be more useful if it is written in to a custom report file.



Flow Status	Successful - Wed Nov 18 08:17:47 2020
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Standard Edition
Revision Name	spin
Top-level Entity Name	spin
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	100 / 41,910 (< 1 %)
Total registers	137
Total pins	43 / 499 (9 %)
Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)

1) Open any project in Quartus, and compile the design. If you already have a compiled project, you can open the Table of Contents in Processing > Compilation Report. For some reason Quartus does not automatically open the reports for a compiled design when it is reopened in Quartus.

2) Create the following script file

```
load_package report
project_open <myproject>
load_report
set panel_names [get_report_panel_names]
foreach panel_name $panel_names {
post_message "$panel_name"
}
```

Note that <myproject> should be replaced with the name of your open Quartus project. Also, if you place the script in your Quartus project directory, the Tcl window in Quartus will find the file without requiring path information.

3) Open the Tcl command window in Quartus, or launch quartus_sh with the -s switch in your Quartus project directory. Run (or source) your script. All the strings for all the panels and report names available in Quartus will be listed in the Quartus transcript window (or in your Tcl console).

Note: If you use a command window, you must first open the project using the project_open command.

4) Take a screenshot of the list of panels, you will submit this.

Part 2: Creating a .csv file

A comma separated value (or .csv) file is simply a list of information that uses a comma character to delineate fields. You can open a .csv file in a text editor but it doesn't look all that nice, so it is better to open the file in either Excel or Google Sheets. Often having a spreadsheet listing of data is more useful that just a report.

Tcl can be used to easily create a .csv file. For this section, the script will capture the information from the Fitter Summary panel and write that information into a .csv file.

1) Create this script:

```
load_package report
project_open [get_current_project]
load_report
set panel_name "Fitter||Fitter Summary"
set csv_file "fitter_summary.csv"
set fh [open $csv_file w]
set num_rows [get_number_of_rows -name $panel_name]

for { set i 0 } {$i < $num_rows} {incr i } {
     set row_data1 [get_report_panel_row -name $panel_name \
          -row $i ]

     puts $fh [join $row_data "," ]

}
close $fh
unload_report}
```

A few notes about the script. Notice that in the second line, instead of hardcoding the project name, the command get_current_project was used. Second, notice that the panel Fitter||Fitter Summary was listed as part of the output seen in Part 1 above. Next, doing file IO requires the use of a file handle. Using the w switch in the `set fh [open $csv_file w]` line means that the file will be created if it doesn't exist, and that the contents will be overwritten each time the script is run. It is very important that once a file handle is opened, it must also be closed before the script completes.

Finally, the backslash character \ is used for line continuation in Tcl. If the command is all in a single line, that character is not necessary.

2) Run the script, and open the resulting .csv file in your spreadsheet program of choice. Compare the spreadsheet with the Fitter Summary as displayed in Quartus. See a problem?

What is the problem? Anytime there is a comma in the data, it uses a new column.

3) Since Quartus used commas in some of the larger numbers displayed in the Fitter Summary pane, the .csv in interpreting these numbers as spanning multiple columns. This is an unfortunate side effect, but easily remedied. To get a spreadsheet program to treat a large number containing commas as a single number, the number can be enclosed in double quotes.

4) Execute the Tcl command help -pkg report to view the available Tcl commands in the report package. The command get_report_panel_data looks interesting and might help in this situation.

By using get_report_panel_data instead of get_report_panel_row, the script can be easily modified to read the two columns of data individually. Once we have the column 1 data, it will be easy to enclose the data values in double quotes.

5) Replace the looping section of the script with this:

```
for { set i 0 } {$i < $num_rows} {incr i } {
     set data1 [get_report_panel_data -name $panel_name \
          -row $i -col 0 ]
     set data2 [get_report_panel_data -name $panel_name \
          -row $i -col 1 ]
     set data3 \"$data2\"
     puts $fh [join [list $data1 $data3] "," ]
```

We read each of the two columns separately. For the second column value, we enclose the value with double quotes. Note that in order to use a double quote, which is a special formatting character in Tcl, an escape character (\) is required. In the puts statement, we have to create a list before inserting the , as the separating character.

6) Rerun the script. Better? Yes

Submit your now corrected .csv file.

Part 3: Automating script execution

For this part, you will add the automation required to check that all I/O pins have location assignments. You will use the pins.tcl script that is included with the homework documents.

1) Close your Quartus project. In your project .qsf file, add this line:

set_global_assignment -name POST_MODULE_SCRIPT_FILE "quartus_sh:pins.tcl"

This line assumes that the file pins.tcl is in your quartus project folder.

2) Reopen your Quartus project. Go to Processing > Start > Start Analysis & Synthesis. When this step completes, scroll to the bottom of the transcript window, and find the line where the pins.tcl file was executed. Notice there is nothing written to the console from the script. Since the script only generates output when the quartus_fit step executes, nothing extra was written to the transcript window.

3) Repeat step 3 only this time run the Start Fitter step. Again, scroll down to the bottom of the transcript window. **Take a screenshot of the last 7 or so lines that display the script**

output.

4) Close your Quartus project. In the .qsf file, comment out one or more pin location assignments.

5) Do a full compile on this Quartus project. The project should halt after the quartus_fit step and display messages indicating the missing pin location assignments.

Take a screenshot of the script messages.

Notice that the error indication(s) are printed in red. You have the option of setting the severity level of a message in the post_message Tcl command.

6) Add back the pin locations you commented out in your .qsf file.

Part 4: Creating a compile script with auto download to board

1) Generate a project Tcl file – Project > Generate Tcl File for your project. Close Quartus.

2) Edit the generated file to load the flow library and compile execution step.

At the top, add `package require ::quartus::flow`
At the end, immediately after `export_assignments` `execute_flow -compile`

3) Open a command window and cd to your project directory. Launch quartus_sh -s. Source your Tcl script, and make sure the compile completes successfully.

4) In your script, add a command to run a script post flow.

```
set_global_assignment -name POST_FLOW_SCRIPT_FILE "quartus_stp:prog.tcl"
```

5) prog.tcl will be a one line script

```
qexec "quartus_pgm.exe <project_name>.cdf"
```

Make sure you put in the correct project name. If your project does not have a .cdf file, you must create one by opening the Quartus Programmer GUI and setting up the programming flow. A .cdf will be created when you close the programmer.

6) Rerun the complete script and confirm that your board is programmed successfully.

Submit your completed script from Part 4.