23T2 COMP4601 Week 2 Lab Hand-in exercises 22/05/2023
(Notes and questions on PP4FPGAs – Chapter 2 exercises)

This hand-in is worth 10% of your mark in the course.

Your answers to indicated exercises should be submitted in a single PDF file less than 5MB large by 5pm Monday 12 June using the Week 2 lab submission link on the course website.

Prior to commencing the lab in Week 2, we assume that you have configured your lab environment and to have completed Ch. 2 of the 2020.1 Vivado HLS Tutorial.

During Week 2, you are expected to complete the indicated sections of Chapters 3 & 4 of the 2020.1 Vivado HLS Tutorial and to carry out selected exercises from Ch. 2 of the text.

Completing the exercises below should take 2 – 3 hours after you have completed Chs 2 – 4 of the Vivado HLS Tutorial.

You will be assessed on your ability to respond concisely to questions. As a rough guide, we are expecting you to submit no more than about 5 pages. Please do not include printouts of raw synthesis reports or schedules unless explicitly asked to do so. Rather, tabulate comparative results and discuss the trends across multiple solutions.

Please note that the virtual machines in the CSE labs are NON-PERSISTENT. This means that any work you have stored locally – on the C:\ drive, say – will not be saved when you close the machine, or if your session exits abnormally. You therefore need to save any files you have created locally to a removable device or to your CSE home before you power down the machine.

Exercises

1. Complete the exercise at the end of Sec 2.5 on page 38 of the text.

   The code from Figure 2.1 is contained in the design file `fir11_initial.c`[1]. A recommended method for completing the exercise follows:
   a. Copy `fir11_initial.c` to `fir11_sec2_5.cpp` in a project directory on your C: drive, `C:\comp4601\lab2`, say. Note that the file type needs to be changed to C++ in order to be able to include and use the `ap_int.h` header file.
   b. Create a new Vivado HLS project named `sec2_5` in `C:\comp4601\lab2`, add the `fir11_sec2_5.cpp` file and top function `fir`; set the target clock period to 1 ns and select the part `xqzu5ev-ffrb900-1-i`, which is identical to the device you have access to in the course.
   c. Synthesize the design (Run C synthesis) to obtain the synthesis results.

[1] The design files from the text are included in a zip file that can be accessed from the Overview section of the Lectures page of the course website.

d. Create a new solution for each target clock frequency, copying all other design criteria over from solution1 (the previous solution) and incrementing the target clock period.
e. Re-synthesize the design.
f. Repeat steps d. and e. until all 10 solutions have been created.
g. Compare the solutions and examine the schedule for any designs you find interesting by setting the active solution in the Project menu and selecting the analysis perspective.
h. Note that there are options to export individual or compared synthesis results for printing at the bottom of each report.
i. Hand in:
    i. Tabulate the timing results and utilization for all solutions. (You may extract the table from a comparison report.)
    ii. Discuss trends and notable features of the results you have obtained in your hand-in for this week's lab.
    iii. In terms of latency and utilization, which solutions would you regard as best?
    iv. Considering the number of resources of each type available on the FPGA you used, what would you consider to be a "significant" change in the utilization of those resources that is worth factoring into choosing the best solution?
    v. Describe the factors influencing the estimated clock period as the target clock period is varied.
    vi. Why do you think the iteration latency is 4 – 13 cycles when the target period is 1 ns?

2. Complete the exercise at the end of Sec 2.6 on page 38 of the text.

   Create a copy of `fir11_initial.c` to `fir11_sec2_6.cpp` in `C:\comp4601\lab2.`

   Note that the creation of a project for this exercise can be streamlined once you have completed Exercise 1. above. You will find a file named `script.tcl` in `C:\comp4601\lab2\sec2_5\solution1` – this file can be edited to create a .tcl file like you have already used in the Vivado Tutorial. Copy `script.tcl` to `C:\comp4601\lab2` and edit it to change the name of the project to `sec2_6,` the name of the file to be added to `fir11_sec2_6.cpp,` and the clock period to 10 ns. Comment out the synthesis and export commands and add an exit command to the end of the file.

```
############################################################
## This file is generated automatically by Vivado HLS.
## Please DO NOT edit it.
## Copyright (C) 1986-2020 Xilinx, Inc. All Rights Reserved.
############################################################
open_project sec2_6
set_top fir
add_files fir11_sec2_6.cpp
open_solution "solution1"
set_part {xqzu5ev-ffrb900-1-i}
create_clock -period 10 -name default
#source "./sec2_5/solution1/directives.tcl"
#csim_design
#csynth_design
#cosim_design
#export_design -format ip_catalog
exit
```

This .tcl file can now be run in the Vivado HLS command prompt using the command `vivado_hls -f script.tcl` from the `C:\comp4601\lab2` directory. This command can even be run while the Vivado HLS Tool is open. Once you have run the script in the command window, you can File→Close Project and Open the project `sec2_6` in the Vivado HLS Tool in a streamlined manner.
To reiterate the steps for carrying out the exercise:

a. Open the source code when you enter the project
b. Synthesize the design to obtain baseline results
c. Create a new solution
d. Replace the `Shift_Accum_Loop` code with the code from `fir11_ifelse.c`
e. Resynthesize the design
f. Compare the synthesis results
g. Check the schedules for both designs
h. <u>Hand in:</u>
    i. Briefly describe and explain the results you have obtained in your hand-in for this week's lab.
    ii. Compare the estimated latency of the designs.
    iii. Explain how the multiplication was computed after loop hoisting.

3. Complete the exercise in the middle of Sec 2.7 on page 39 of the text.

Follow a similar method as described for Ex. 2 steps a – h above. Since you will be editing the code, create a fresh copy of `fir11_initial.c` named `fir11_sec2_7.cpp`. Edit `script.tcl` to set up the new project for this exercise.

Instead of the instructions in step 2.d, replace the `Shift_Accum_Loop` code with the code from `fir11_partition.c`.

<u>In your hand-in for this week's lab,</u> briefly compare the implementations before and after loop fission. Explain the difference in performance. How does the resource utilization change, and why?

4. Complete the exercise on unrolling the TDL loop in the middle of page 41 of the text.

   This exercise makes use of the modified copy of `fir11_sec2_7.cpp` that results after the `Shift_Accum_Loop` code has been replaced with the code from `fir11_partition.c`. You may wish to name this copy `fir11_sec2_8.cpp`

   Synthesize the baseline solution, and then create three new solutions, using unroll factors of 4, 8 and complete on the TDL loop. Add the **unroll** directive to the directive file, not the source file.

   Create another solution that unrolls the TDL loop completely and adds an **array_partition** directive with complete partitioning of variable `shift_reg`.

   Hand in: Check the schedules for each solution and include a brief description and explanation of your 5 solutions in your hand-in for this week's lab.

5. In the spirit of the exercise at the end of Sec 2.8 on page 43, create a new solution for the code from Ex. 4 above. Unroll the MAC loop completely and observe the synthesis results and the schedule.

   Hand in: What is the impact on the resource utilization vis-à-vis the baseline case from Ex 4? How does the performance change? Briefly describe and explain your observations in the hand-in for this week's lab.

6. Complete the exercise on loop pipelining described on page 45. You may wish to copy `fir11_sec2_8.cpp` to create `fir11_sec2_9.cpp`. Obtain a baseline without directives before pipelining successive solutions with increasing iteration intervals.

   Hand in: Check the estimated loop latency for each solution in the synthesis report. Tabulate your latency and utilization results. Report your answers to the exercise on p. 45 in the hand-in for this week's lab.