EENG 5560 Final Project

Assigned: March 19, 2024
Demos : April 30, 2024 and May 2, 2024
Source Files and Report Due: May 2, 2024

Contents

Question

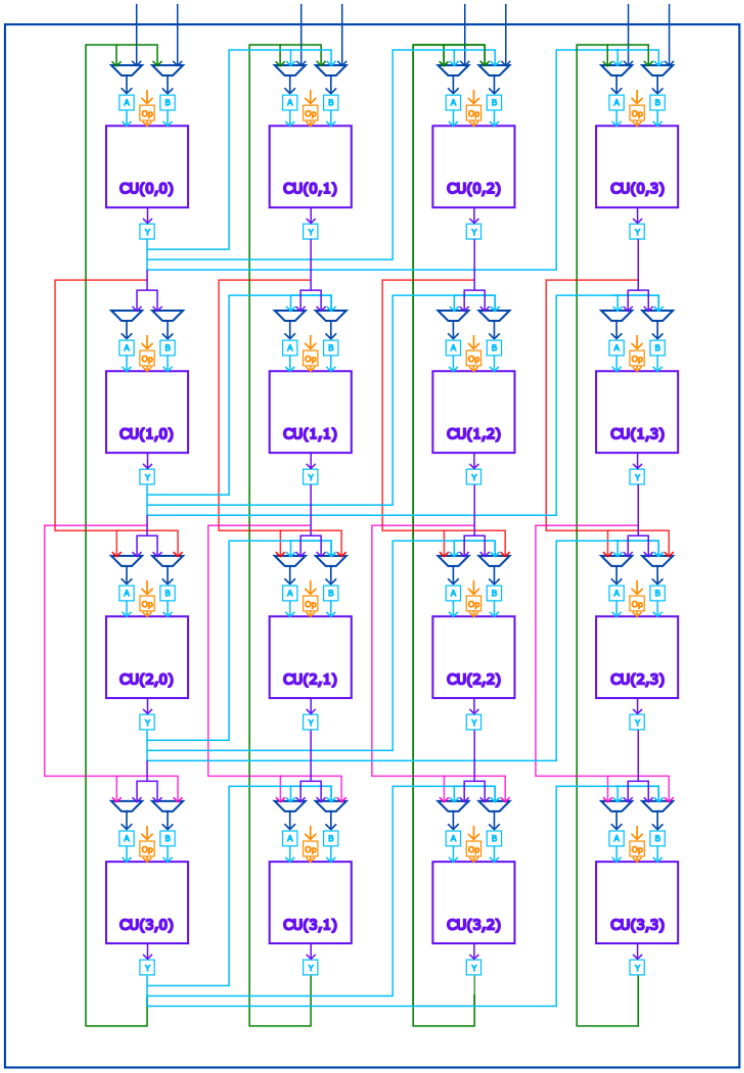

Figure 1: Overall design.

Using the same parameterized CU from your assignments with a data width parameter set to 4, implement the design shown in Fig. 1. All data and

operation values are first sent into storage units, not directly accessed from or to the CUs. It is also required to use arrays for all of the ports (but generates are not required).

Overall, you will implement an 8x4 design but using virtualization with a 4x4 fabric. The first row has the possible data input source of both external data (navy blue) or the output of the CU in its same column position but from the last row (green), after a single run. In all rows, the CUs in columns 1 through 3 are capable of receiving data from horizontal connections to the CU in the first column of their respective rows. From row 1 onwards, the data inputs for the CUs could also be from the CU in the previous row from the same column position. For rows 2 and 3, the data inputs also can be from multi-level connections from rows 0 (red) and 1 (pink), respectively.

Test cases

You need to make testbenches for all components used, but you only have to show the calculations and waveforms for the overall design in this assignment. The outputs of all the CUs need to be shown, not just the top level final outputs. The test cases for the overall design are as follows.

Run 1

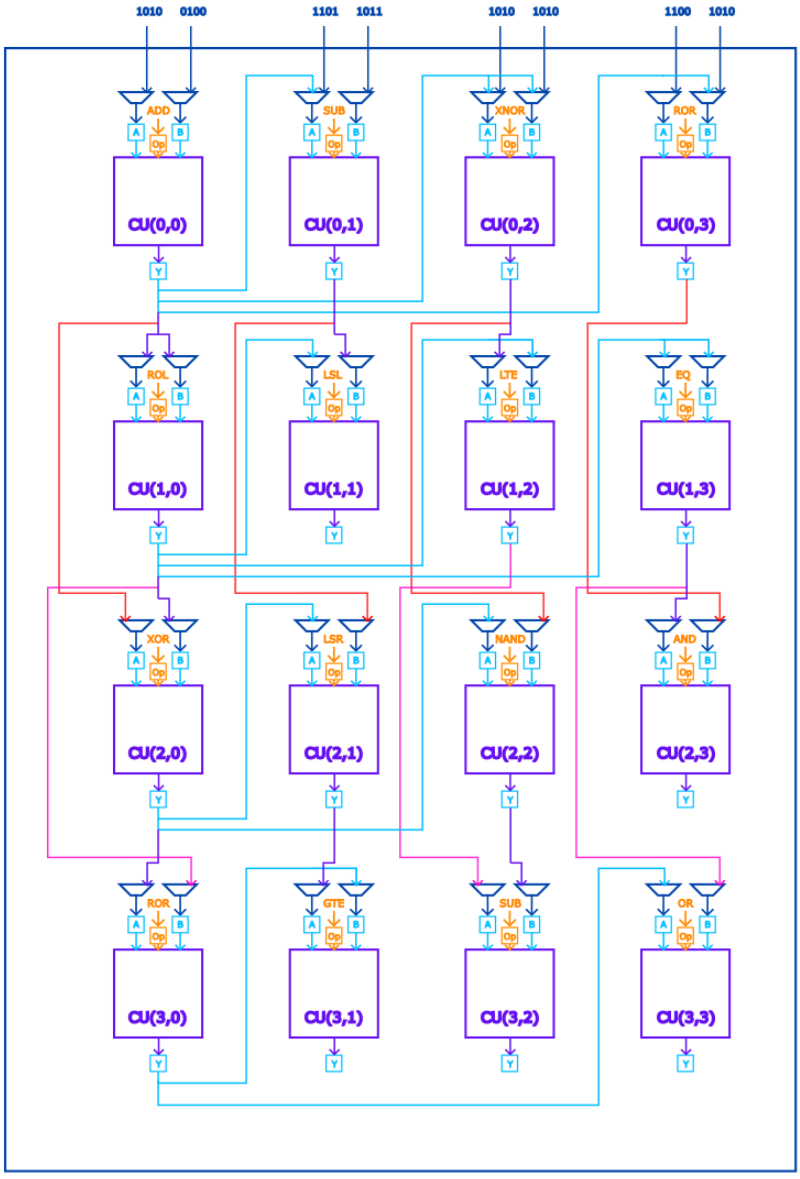

Figure 2: Run 1.

In the first run, the data to the first row of CUs will be provided externally or from the output of the CU in the first column (column 0). The individual connections to choose for each CU are shown in Fig. 2 as well as detailed in the table below. Please note that for the first row, although all of the external data inputs are shown, not all of them are used. If there is another arrow for either of the CU's mux, use that data instead of the external data.

External Data Inputs

CU#	A	B
CU(0,0)	1010	0100
CU(0,1)	1101	1011
CU(0,2)	1010	1010
CU(0,3)	1100	1010

Connectivity and Operations

CU#	SourceA	SourceB	Oper
CU(0,0)	External	External	ADD
CU(0,1)	CU(0,0)	External	SUB
CU(0,2)	CU(0,0)	CU(0,0)	XNOR
CU(0,3)	External	CU(0,0)	ROR
CU(1,0)	CU(0,0)	CU(0,0)	ROL
CU(1,1)	CU(1,0)	CU(0,1)	LSL
CU(1,2)	CU(1,0)	CU(0,2)	LTE
CU(1,3)	CU(1,0)	CU(1,0)	EQ
CU(2,0)	CU(0,0)	CU(1,0)	XOR
CU(2,1)	CU(2,0)	CU(0,1)	LSR
CU(2,2)	CU(2,0)	CU(1,2)	NAND
CU(2,3)	CU(1,3)	CU(06,3)	AND
CU(3,0)	CU(2,0)	CU(1,0)	ROR
CU(3,1)	CU(2,1)	CU(3,0)	GTE
CU(3,2)	CU(1,2)	CU(2,2)	SUB
CU(3,3)	CU(3,0)	CU(1,3)	OR

Run 2

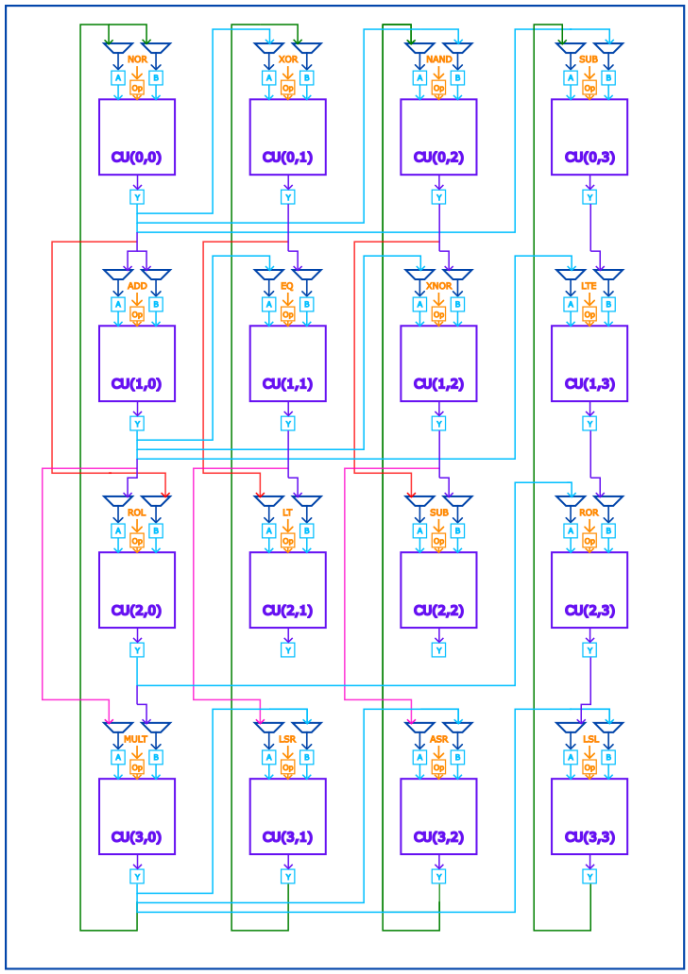

Figure 3: Run 2.

Connectivity and Operations

CU#	SourceA	SourceB	Oper
CU(0,0)	CU(3,0)	CU(3,0)	NOR
CU(0,1)	CU(0,0)	CU(3,1)	XOR
CU(0,2)	CU(3,2)	CU(0,0)	NAND
CU(0,3)	CU(3,3)	CU(0,0)	SUB
CU(1,0)	CU(0,0)	CU(0,0)	ADD
CU(1,1)	CU(1,0)	CU(0,1)	EQ
CU(1,2)	CU(1,0)	CU(0,2)	XNOR
CU(1,3)	CU(1,0)	CU(0,3)	LTE
CU(2,0)	CU(1,0)	CU(0,0)	ROL
CU(2,1)	CU(2,0)	CU(0,1)	LT
CU(2,2)	CU(0,2)	CU(1,2)	SUB
CU(2,3)	CU(1,3)	CU(0,3)	AND
CU(3,0)	CU(1,0)	CU(2,0)	MULT
CU(3,1)	CU(1,1)	CU(3,0)	LSR
CU(3,2)	CU(1,2)	CU(3,0)	ASR
CU(3,3)	CU(2,3)	CU(3,0)	LSL

Formatting

Steps: Printing code to pdf

For each source file (both the design and the simulation VHDL or other HDL language files), print the code to pdf either in Vivado or by opening the code in any other text editor and printing it from there. It is preferable to combine all of the pdfs of the code into a single pdf, you can do so using this website or any other websites/software with that capability. The steps to print the code to a pdf from Vivado are as follows:

1. With the VHDL\source file currently open and being viewed in the text editor, click the file tab at the top and near the bottom select the print option (or press Ctrl + P).

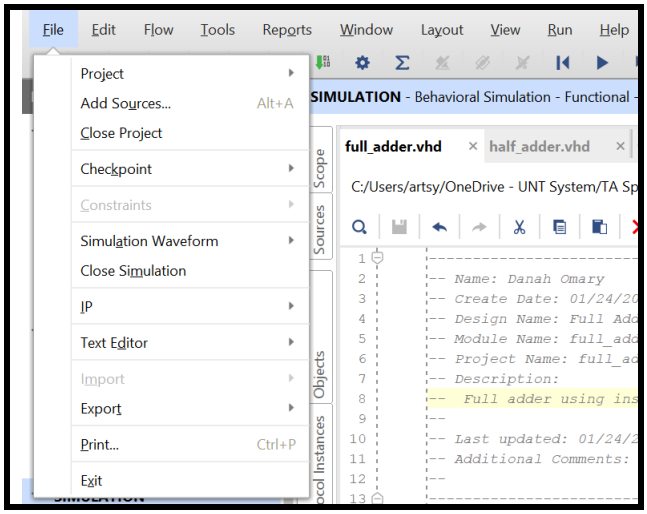

Figure 4: Print button.

2. In the popup, in the dropdown next to the “Name: ” text, make sure to select Microsoft Print to PDF. Then click ok.

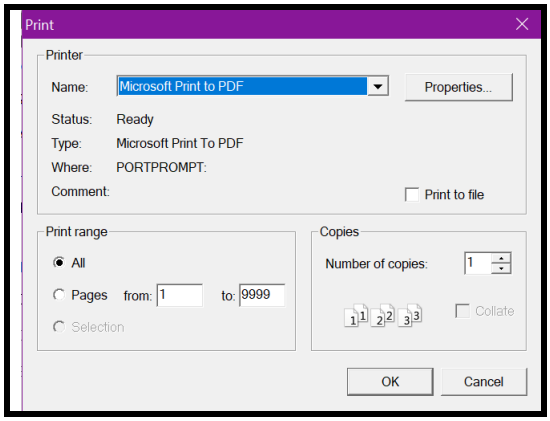

Figure 5: Printing popup with print to pdf chosen.

3. Name the pdf file something clearly indicating what it is and then browse to where you'd like to save the pdf to.

Source files zipping

In a **separate** **zip file, include the following files (and only these files):**

- **A README text file that includes a list of all the files that should be included and any special instructions needed to run the top level module (in Windows, to create a text file, in the file explorer right click empty space and choose "New - Text Document"). This will help in the case that any files that were supposed to be included ended up missing.**
- **VHDL file for top level component**
- **VHDL files for all subcomponents and any of their testbenches**
- **VHDL file for top level testbench**

Do not include other files from the project or the submission (i.e. the report and pdf of the vhdl code), only the VHDL files are needed.

Example submission files

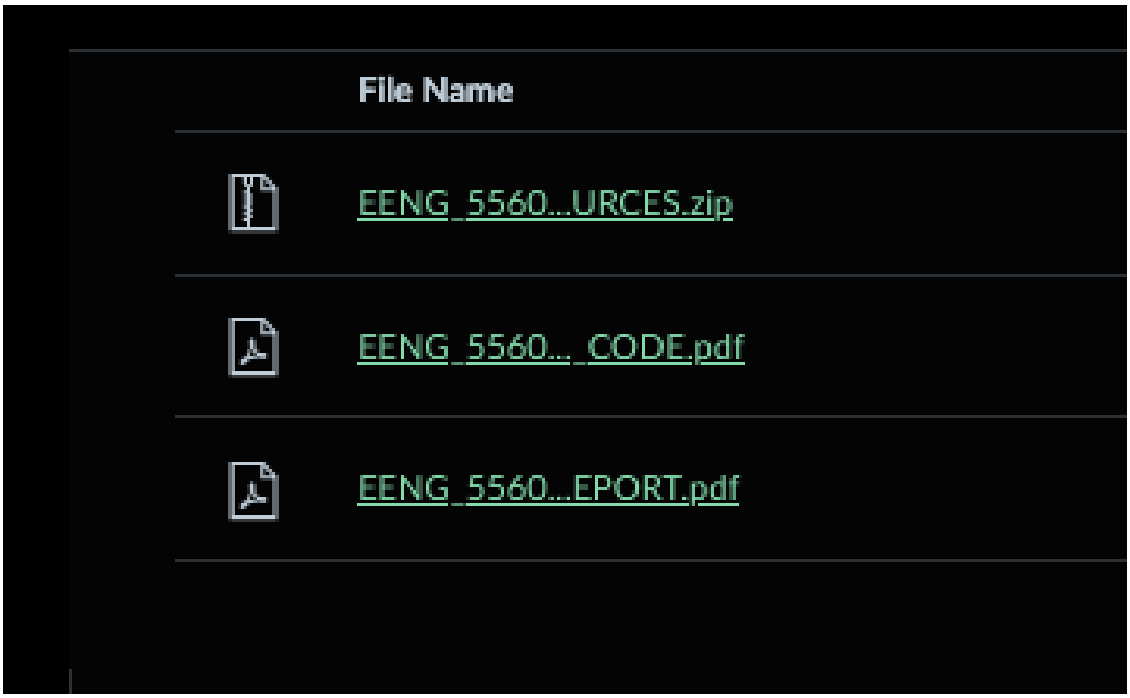

Figure 6: Example showing the file types and good naming conventions of a submission.

In total you will have 3 major parts to your submission:

1. **Report(s): pdf(s)**
2. **Code: pdf(s)**
3. **Source files: zip folder**

As can be seen in the figure, the pdfs for the code and report *NEED TO BE OUTSIDE OF A ZIP FILE*.

Other formatting notes

If you would like to make a report yourself rather than follow the provided template, please make sure to include all of the things listed out in the checklist below as well as a Table of Contents and page numbers on each page. Do not include all of the VHDL code in the report, please keep it separate from the report.

Checklist

Your submission should include (using helpful/distinguishable file names):

- ☐ **Report (either in pdf or .docx file type) including:**
  - ☐ **Design: Block diagrams, design explanation**
  - ☐ **Generated RTL Schematic and Block design (top module only)**
  - ☐ **Simulation waveforms for the test cases, including the overall outputs and relevant intermediate signals (top module only)**
  - ☐ **Table with Calculated outputs vs Simulation outputs for all test cases (top module only)**
- ☐ **PDF(s): VHDL (or verilog or systemverilog) code for the top level component, all subcomponents, and the testbenches (can combine all code pdfs into one pdf overall or include individual pdfs)**
- ☐ **Zip file with all the VHDL source code files needed to run the top level component**