GEORGIA INSTITUTE OF TECHNOLOGY

SCHOOL of ELECTRICAL AND COMPUTER ENGINEERING

Lab 5: Filter Design

Due Date : Thursday October 5 (11:55 pm)

---

1. Introduction

A Digital FIR Filter is basically a set of MAC operations specified in an order depending upon on the nature of the filter. We will be designing a simple 8-tap filter in this lab. Thus the filter equation is **output = (a*x(0)) + (b*x(1)) + (c*x(2)) + (d*x(3)) + (e*x(4)) + (f*x(5)) + (g*x(6)) + (h*x(7)).** Here the coefficients a through h will be passed by the testbench. Also, note that the multiplication and addition are **signed operations**. The template for the fir filter is provided in "filter_8.vhd" in /src directory.

http://surf-vhdl.com/how-to-implement-fir-filter-in-vhdl/

The design is heavily based on the link above (with some changes of course). You can use it for reference.

The explanations of the input and output ports are given below.

i. clk => Clock signal

ii. rst_n => Reset signal. Note that the sequential elements that are used in your design should have **asynchronous active low reset**. i.e. if the reset is pulled low, the flops should reset regardless of the state of the clock

iii. coeff_en_in => Coefficient enable signal. When coeff_en_in is'1', all the complex filter coefficients are to be captured at once(in the same cycle) into the design. **The testbench is designed in such a way that at the beginning of the test, all the coefficients are passed at once. After this data is provided serially.**

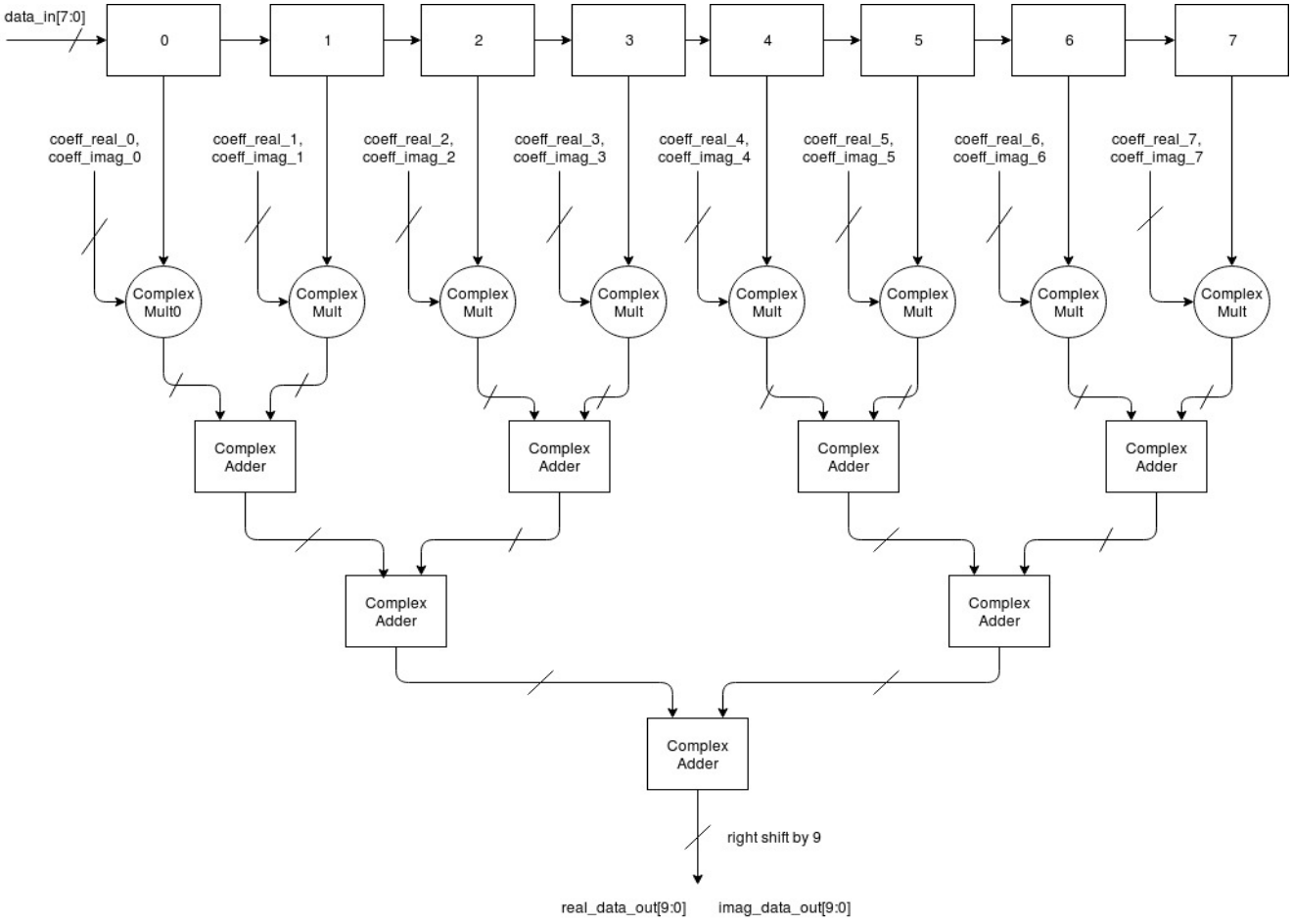
iv. data_en_in => data_en_in is the qualifier for the data. When data_en_in ='1', data which is passed from testbench is valid and is to be captured in the design.

v. data_in => The input data port from which data is passed to the design. **Note that real data is passed from the design**

vi. coeff_real_(0 to 7) and coeff_imag(0 to 7) => These are the complex coefficients used in fir filter for multiplication.

vii. real_data_out and imag_data_out => Outputs from the design. After the complex operations, the real data which is passed into the input is transformed into real and imaginary components.

viii. out_valid => control signal for the output. When ctrl = '1', the outputs real_data_out and imag_data_out are valid. This information is useful to the testbench which captures the data only when output is valid.
The reason why valid is important is that different architectures will lead to different latencies. Thus, the outside world should know when the output data is to be captured. So, let's say Engineer A has designed his system so that latency is 8 cycles. Engineer B's design has a latency of 10 cycles. To the outside world (in this case the testbench), this does not matter as the testbench is agnostic to the internal details of the design. It must check only the valid signal.



This system can be implemented in numerous ways with different latencies, area and power. You can use any architecture as long as you can clearly draw the block diagram of your architecture in the report.
There is a sample input file from which the data is read into the testbench . The testbench dumps out an output file “sample_output_filter_reference.txt”. Compare this output with the reference output file provided with the assignment. The output will be the same for any architecture. The grading will take into consideration the functional correctness, utilization (area) and timing of the design. Note that you must create a clock for the Vivado Process Flow. This can be done by adding “lab5_constraints.xdc”to the constraints while creating the project. The constraints file contains the name and period of the clock which is created. 3 re-submissions are allowed for this lab. The resources information is obtained from “filter_8_utilization_placed.rpt”. The timing information is obtained from “filter_8_timing_summary_routed.rpt”. The power information is obtained from “filter_8_power_routed.rpt”.
Note that our design is extremely bad for area but is very fast.
The testbench gives a data sample every cycle. In your design, you can choose to use a memory/buffer which stores all these samples and will parse it to the main logic at a speed which is decided by the latency of your architecture. This additional memory overhead will not be considered for grading.

2. Instructions

i. Draw the block diagram of the architecture you will be using if it is different from the reference block diagram. (If you are using the same architecture as provided in this document, you do not have to submit block diagram)

ii. Write the design in vhdl(filter_8.vhd) per the block diagram. The template for the design has been provided filter_8.vhd in /src folder. Verify the design by comparing it with the reference output. (70% of grade for first two deliverables. (**Note that if the design is not verified, the block diagram and implementation will not be considered**)

iii. Perform Synthesis and Implementation for the design. List the area (#Slice LUTs, #Bonded IO Buffers, Primitives) and Timing (Worst Negative slack) (10 % of grade).
iv. Answer the following question – " Why do you think we are doing the right shift operation at the end?" (20% of grade)
Hint: Think of how you do fractional multiplication for fixed point numbers

3. Deliverables

i. Create a PDF which lists the #resources (Slice Logic Distribution, IO Information, #Primitives) and Worst Negative Slack of "filter_8.vhd". The title of the document should be of the form "lab5_firstname. lastname.pdf"
ii. The PDF should also have the answer to the question
iii. The design top file "filter_8.vhd"
iv. Other modules (if applicable)
v. Output files "sample_output_filter.txt" and "sample_output_filter_cycle.txt".

I have been facing grading issues with my scripts because students are not following the naming convention in the files and folders. In some submissions, I noticed a 2-level hierarchy in the structure which was not what we had asked for. Some students had submitted tarred folder and not the zipped folders. Taking all of this into consideration, we have decided to make some changes. Please **do not move these files into a folder. Upload all the attachments directly on t-square. Also, note that not following the naming conventions or any of the instructions might lead to a penalty.**

Note: Late submissions are not accepted. In case of extraordinary circumstances, written permission must be obtained from Dr.Madisetti.