Prac 5 – Implementation of a Digital FIR Filter

NOTE: Things can take time. So be prepared with the lab activity, do all the preliminary designs and as much implementation as you can beforehand.

Design Task

You are required to implement the following digital filter (a finite impulse response (FIR) filter) on an FPGA and test its performance.

$$y(n) = \sum_{i=0}^{16} a_i x(n-i)$$

Digital implementation of this filter is done using the Xilinx System Generator tool flow, which integrates Matlab-Simulink and Xilinx Vivado tools to allow rapid prototyping of digital signal processing (DSP) systems on FPGAs using a model-based structural approach. More details on this tool flow are provided in supporting resources on Blackboard and also will be provided during the lectures and labs. Unless you have an in-person practical session at 47-401 lab, you should use the RDP connections to lab computers to complete this lab. Also, you must use only the basic processing elements such as adders, multipliers, delays, multiplexers etc, to implement your designs and you are not allowed to use system level Xilinx IP-blocks for DPS such as FIR filters, FFTs etc.

1. Verify the operation of the filter using the Matlab code provided, which also produces the filter coefficients. The filter is a low-pass filter and the provided Matlab script uses a two-tone signal to verify the filter operation in software implementation.
2. Implement the filter using the Xilinx System Generator tools in Matlab-Simulink using the standard direct-form structure **with no-pipelining and without any optimisation**. You will have to select appropriate fixed-point word lengths for both the input (i.e. gateway in blocks) and the filter coefficients. Verify the correct operation of your implementation by playing the output sound and also by comparing the output to the output obtained from the software implementation. Obtain the FPGA resources, critical path delay, maximum frequency, and throughput figures for this implementation using HDL netlist generation and using Artix-7 series FPGA family as the target.
3. Now, optimise the above design using pipelining/register-retiming/data broadcast structure (use one or a combination of these techniques covered in the lectures) as much as you can to improve the critical path delay (i.e. time complexity) at the cost of area complexity. Also, optimise the fixed-point word lengths as much as you can (i.e., by comparing the hardware output to the software output and try to minimise the error between the two). Verify the correct operation of the filter and obtain the FPGA resources, critical path delay, maximum frequency, and throughput figures for this optimised implementation using HDL netlist generation and using Artix-7 series FPGA family as the target. Tabulate your findings and compare and contrast the two designs.

Report Content and Format

A typeset PDF report must be submitted by the due date with the following content:
Introduction stating the problem description, any assumptions, and design objectives
Design description including a block diagram explaining the complete design. Block diagrams should be drawn using standard symbols for inputs/outputs, signals and busses as shown below.

- Block diagram (either drawn or a screenshot from the System Generator model with clear labelling of input, output, and basic functional blocks being used), functional verification (i.e. a plot showing the software and hardware outputs or an error figure between the two), and FPGA resources, speed, throughput performance of the **un-optimised** design
- Block diagram (either drawn or a screenshot from the System Generator model with clear labelling of input, output, and basic functional blocks being used), functional verification (i.e. a plot showing the software and hardware outputs or an error figure between the two), and FPGA resources, speed, throughput performance of the **optimised** design
- Selected system word lengths and how you obtained them (i.e. justification)
- Comparison of the two designs
- Conclusion – a reflection of what you've achieved in this exercise, problems (if you had any) and potential improvements to your design (if any)
- References (if any)
- You must have the marking sheet on the next page as the last page of your report and your marks will be indicated on this sheet (this has also been uploaded on BB as a single pdf file so that you can append this to your report as the last page)
- If you don't complete all the tasks by the due date, you can still submit the report explaining what you have achieved, and this will be treated as an attempt to the practical. Note that you need to attempt all the practicals to pass the course.

Marking Criteria

The pracs in this course are marked to a specific criteria. This means you must demonstrate sufficient understanding and functionality and the marking will be done according to the rubric provided below. You must attempt each prac (i.e., a report must be received by the due date) to pass the course. All designs VHDL code used **must be your own work.** You are NOT permitted to use other VHDL code sources, unless directed to. Plagiarism is unacceptable and please read and understand the School Statement on Misconduct, available on the ITEE website at: http://www.itee.uq.edu.au/itee-student-misconduct-including-plagiarism . To avoid problems make sure that your VHDL code is the product of your work and do not let anybody else 'reuse' your code.

Marks	**Criteria**
Simulation	
0	Not sufficient results are provided
1	Results shown only for the un-optimised design – working correctly
2	Results shown only for the optimised design – working correctly
3	Results shown for both designs - working correctly
4	Both designs are working correctly, and a comparison is provided
Report	
0	No evidence of content or work
1	Some content, insufficient explanation of circuit
2	Reasonable content, some explanation of circuit (both approaches are explained)
3	Good content, reasonable explanation of circuit (both approaches are explained)
4	Excellent content, good explanation of circuit (both approaches are explained)
Oral assessment	
0	No knowledge of the design
1	Very little knowledge of the design
2	Reasonable knowledge of the design
3	Good knowledge of the design.
4	Excellent knowledge of the design.
Total (12):	**Marker Initials:** **Date:**