DALHOUSIE UNIVERSITY

Final Project: Design a Simple Beamformer

ECED 4260 IC DESIGN AND FABRICATION
Department of Electrical and Computer Engineering
Fall 2023

CONTENTS

OBJECTIVES

designing a simple beamformer for focused Ultrasound imaging on the DE1-SoC boards. The project will consist of three primary phases: the initial design phase, the implementation phase, and the validation phase. You will gain experience in doing preliminary design in MATLAB, building UART interfaces, and in how to approach system-level integration and optimization.

DESIGN OF A BEAMFORMER

1 Introduction

The primary goal of this project is to familiarize ECED4260 students with the process and tools required to design and fabricate a system prototype. In groups of two to four students, you will design and implement a simple beamformer, capable of taking in 8 channels of raw data and creating a single A-line of an ultrasound image. You will use HDL to design this system, targeting the DE1-SoC platform. The project will give you experience designing with RTL descriptions, building input/output interfaces, and teach you how to approach system-level integration and optimization. **Functional implementation will be your primary goal**. To better expose you to real design decisions and tradeoffs, we hope you can optimize your design for cost (FPGA resource utilization) and performance (speed).

In this design project, your first step is to use the specifications given to implement the project in MATLAB. This will give you a good understanding of the design flow, and required internal signals needed to implement this on the DE1-SoC board, as well as create the delay profiles needed for beamforming. Next you will map the high- level specifications to a design which can be translated into a hardware implementation (Top entity diagrams, Architecture schematics, State transition diagrams). After that, you will design each submodule in VHDL/Verilog and debug these implementations. This can end up taking a lot of time if you have not properly thought out your design, however, this should not be a great concern if you were able to implement the beamformer in MATLAB prior. After you have built a working implementation, the next step will be optimizing it for speed and resource use on the target FPGA platform. Finally you will compare the A-Line generated in MATLAB with that generated in the FPGA.

Finally, good time management and good design organization is critical to your success. Good luck!

2 Schedule

Week 10/11, Design planning and MATLAB implementation

Week 12, HDL coding and ModelSim testing; Nov. 23rd Rough Design Due;

Week 13 , Integrate beamformer, optimization of your implementation. Week 14,

Dec. 5 & 7, Demonstration and Project presentation.

Dec. 9th, Final Report Due.

3 Project Specification

Focused ultrasound consists of generating what is known as a B-Mode image by gathering multiple scans know as A-Lines and stitching them together. For each A-line, the beamforming process can be split up into multiple parts:

1. Data is collected from the 8 elements in the ultrasound transducer and sent to ADC's for processing. Since we do not have access to ultrasound transducers, this data will be fed into the FPGA through USB communications (Will most likely provide this part), and the raw data will be provided in a file for reading.

2. The data is then sent to the up sampler through 8 channels, the method of choice for sending the data will be up to you. Two potential options provided in the schematic are to store the data into Block RAM first and select the relevant data or to stream all the data directly into the up sampler. Be aware that altho streaming is easierto code , it requires much stricter timing to implement on the FPGA.

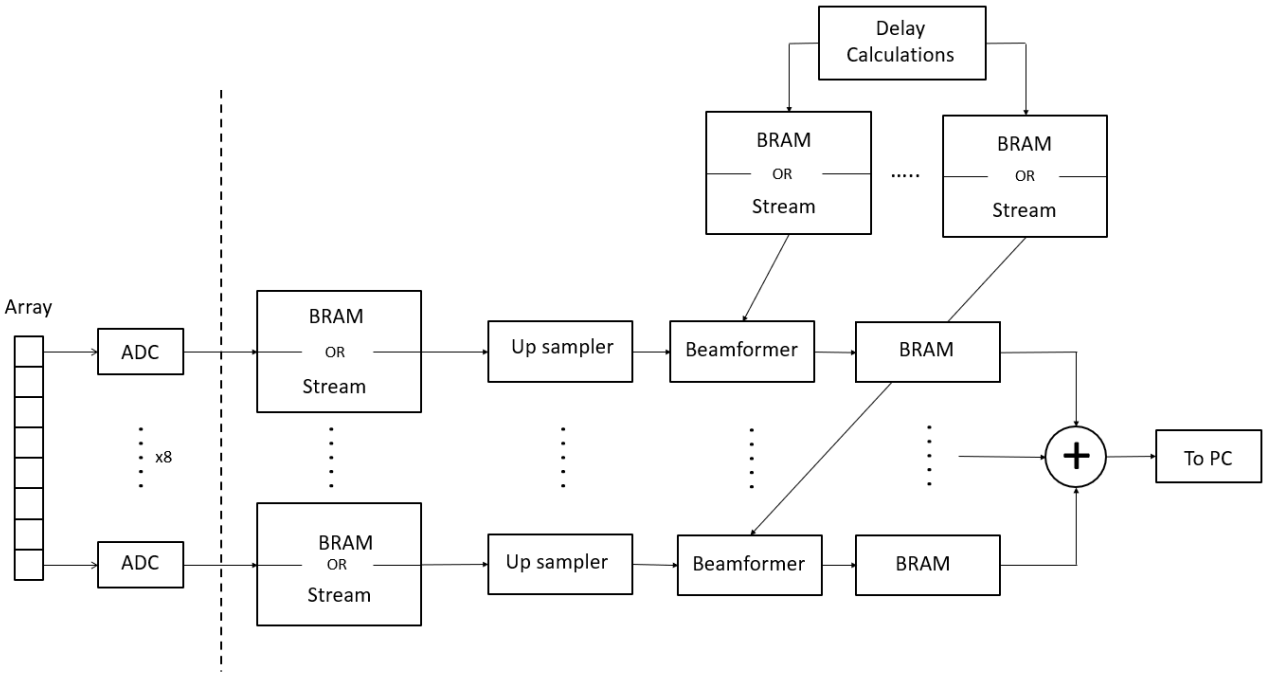
3. Before applying the delay profiles, the data must be up sampled. This is again your choice, two potential options are through zero insertion or by using an FIR filter.

4. Now we can do the beamforming! This is surprisingly easy, you will need to generate a set of delays in MATLAB using the characteristics and equations given and save them in your firmware. From there you simply extract the relevant samples from the raw data based on those delay values. I.E: if your dalays for a channel are 5, 25, and 50, you simply extract the 5^{th}, 25^{th} and 50^{th} sample from the raw data and discard the rest.

5. Then all the beamformed data in each of the eigth channels are stored onto the FPGA with the method of your choicing before being summed together.

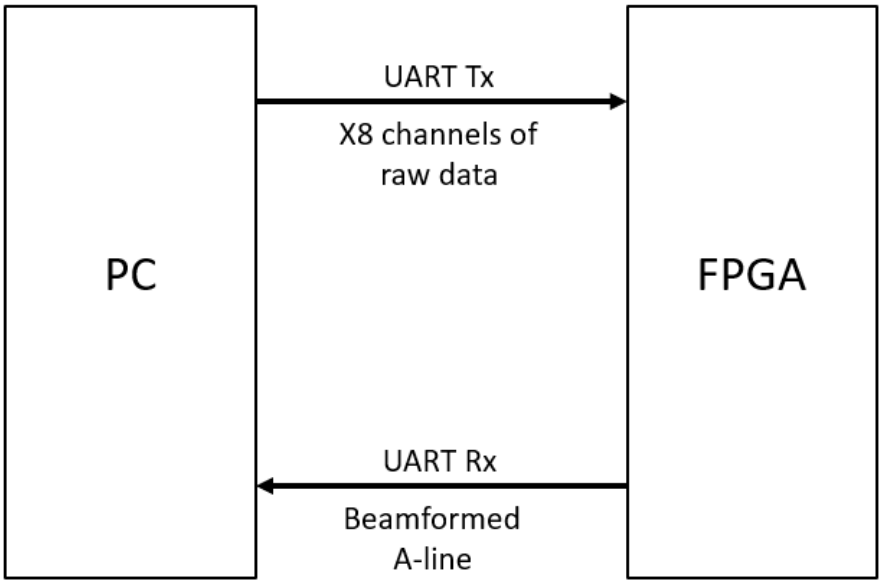
6. Finally, the data is sent back to the PC for viewing through USB communication.

Each component needs to be tested separately. All the annotated simulation waveforms should be included in the report. Then you can put all these components together to make a complete data path. Debug and test the complete data path circuit. Make sure to include the corresponding annotated simulation waveform in the report. In addition to simulating the individual components, you will be expected to load the beamformed result from MATLAB into the testbench and show that the outputs from the FPGA and MATLAB are the same.


Delay
Calculations
BRAM
OR
Stream
.....
BRAM
OR
Stream
Array
ADC
BRAM
OR
Stream
Up sampler
Beamformer
BRAM
x8
To PC
ADC
BRAM
OR
Stream
Up sampler
Beamformer
BRAM

Next, a finite state machine can be designed as the controller of the system. Debug and simulate the FSM until it works properly. Make sure to include the FSM simulation waveform in the report.

Be aware that although the structure and/or IP core for the USB communication will be provided, it will be up to you to configure it such that it sends and receives the data from the PC properly.



The inputs the the beamformer are simply a start and reset which may come from a switch or push button, as well as the data coming in through USB. The Output will be the generated A-line which will also be transmitted by USB. Start is simply used to tell the Beamformer to collect the data and begin the beamforming process whereas Reset will be used to put the beamformer in a reset state, where it waits for the start button to be pushed again.

Other signals will be determined by you through designing the beamformer in MATLAB. Verilog and MATLAB use very similar logic so the variables you create in MATLAB should directly reflect th esignals needed in your FPGA.

DEMONSTRATION

You will need to demonstrate your designed system on a FPGA board to the TA/instructor in the lab.

LAB REPORT INSTRUCTIONS

A formal report is required for this project. A guideline on how to write your report will be posted on the course website.

GRADING[1,2]

Report (50%)		Presentation (50%)
Demonstration	(30%)	
Abstraction & Introduction	(10%)	
Design process, simulation and testing method	(35%)	---
Discussion & Conclusion	(15%)	
Presentation (clarity, neatness, format etc.)	(10%)	

- Lab report and project presentation each account for 50% of the total score of the final project.

Good luck!