Hochschule
Bonn-Rhein-Sieg

Design Experiments for the FPGA FIR Filter Lecture

Prof. Dr. Marco Winzker, Bonn-Rhein-Sieg University, 2020

Competencies and Learning Objectives

With the lecture you learn about development of an FIR filter from algorithm to FPGA implementation including verification with a self-checking testbench.

Lecture video and this information about the experiments is available as a video lecture:
https://www.youtube.com/playlist?list=PLGzeDuLmmxDr10_4zRujRBVnqij-PZyYb

Feel free to use this material for your needs

Version: 20. January 2020

Set-Up of Design-Flow

Learning goal:

- Understand the design-flow of FPGA design and remote lab usage
- This experiment should be done as a preparation for all other experiments

Level of difficulty: Easy

Experiment:

a) Install the design software on your computer
   - GNU Octave
   - Intel Quartus including ModelSim Starter edition and device files for Cyclone V

b) Download the source files and perform FPGA simulation and synthesis
   - Find a test image and resize it to 1280*720 pixel resolution

c) Upload the bitfile (filetype: sof) to the remote lab and perform the experiment

d) Change the rounding in the filter arithmetic and repeat simulation and synthesis
   - Remove term “+ 16” from submodule sharp_arith.vhd (or change to “+ 15”, “+ 17”)

Switch for Sharpness Improvement

Learning goal:

- Working with the VHDL files
- Understanding the structure of the circuit design

Level of difficulty: Easy

Experiment:

a) Use one of the switches to change the output signal between original image and sharpness improvement
   - Top-level module sharp.vhd has three switches at input “enable_in”

b) Perform FPGA synthesis and execute the experiment on the remote lab

c) Observe two things
   - The vertical position of the output image should not “jump” between the two settings
   - Power consumption should be lower if sharpness improvement is disabled

d) Make sure, that the processing mode (sharpness on/off) does not change in the middle of an image

Switch for Degree of Sharpness Improvement

Learning goal:

- Optimization of the signal processing algorithm
- Design verification with Octave and simulation

Level of difficulty: Moderate

Experiment:

a) Use the switches to change the output signal between different degrees of sharpness improvement
   - Can be implemented by changing the amplification factor A for the highpass filter
   - Recommendation: Four modes low: A=1/4, medium: A=1/2, high: A=1, off: A=0

b) Generate verification images with Octave and perform simulation
   - Recommendation: Use four testbenches for the four modes

c) Perform FPGA synthesis and execute the experiment on the remote lab

d) Compare FPGA resource usage and power consumption with original implementation

Correct Handling of Image Borders

Learning goal:

- Digital design for FPGAs

Level of difficulty: Advanced

Experiment:

The VHDL code has two simplifications for handling image borders

- Pixels at the border are filtered together with pixel of the neighbouring line/frame
- The output is shifted by three lines

a) Implement correct handling of image borders in the VHDL design
   - Check how Octave handles filtering at image borders
   - Recommendation: Octave has several options; consider which is best for hardware implementation

b) Perform the design flow with verification and FPGA implementation

Change Filter Coefficients

Learning goal:

- Optimization of the signal processing algorithm
- Trade-off between image quality and FPGA resources

Level of difficulty: Advanced

Experiment:

a) Use other filter coefficients for the highpass filter
   - Use Octave and modify number of taps “n” and cutoff frequency “w” in Octave function “fir1(n, w, type)”
   - Research literature for other filter functions

b) Perform the design flow with verification and FPGA implementation

c) Compare image quality, FPGA resource usage and power consumption
   - Is a 3-tap filter sufficient? How much resource can be saved?
   - How many taps can be implemented on the FPGA?

Reduce RAM Resources for Line Memory

Learning goal:

- Digital design for FPGAs
- Understanding usage of FPGA resources

Level of difficulty: Moderate

Experiment:

The line memories in sharp_linemem.vhd need a delay of 1280 pixel. FPGA synthesis implements this with BlockRAMs and uses 2048 memory locations.

Note: This is sensible, as long as enough FPGA resources are available.

a) Change the line memory so that only 1280 memory locations are used
b) Perform the design flow with verification and FPGA implementation
c) Compare FPGA resource usage and power consumption
d) How many line memories can be implemented with the original approach and the modification?

Increase Word Width for Intermediate Signal

Learning goal:

- Digital design for FPGAs
- Design verification with Octave and simulation

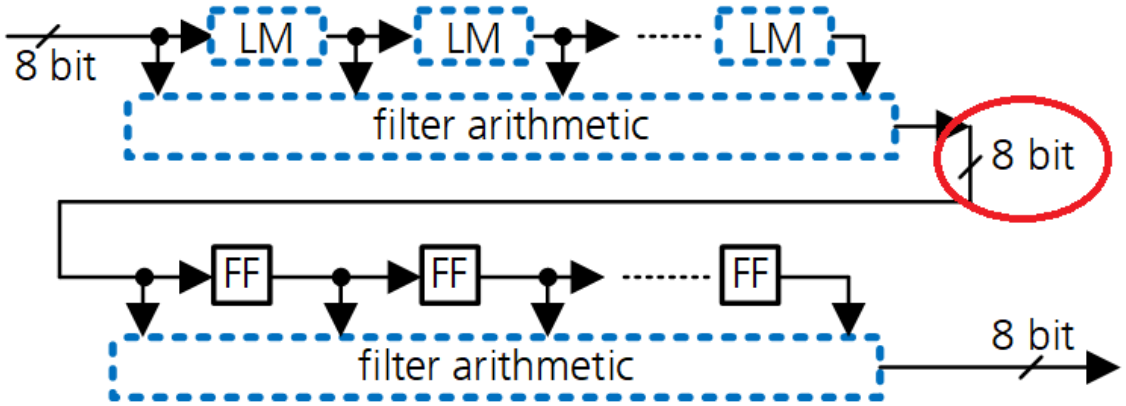
Level of difficulty: Moderate / Advanced

Experiment:

a) Increase the word width of the intermediate signal after the first filter arithmetic

b) Perform simulation and FPGA implementation

c) Compare image quality, FPGA resource usage and power consumption



Advanced:

d) Change Octave code and perform verification with self-checking testbench