Selected Topics in VLSI Design (Module 24513)

Course and contest

Prof. Dirk Timmermann,
Christoph Niemann, Hannes Raddatz

Aim of the course

Developing industry-level skills to design VLSI systems

- Design and simulation using VHDL
- Optimizing hardware designs
- Synthesis for FPGA and ASIC
- Layout of a final chip

Developing soft skills

- Team work
- Presenting your work and ideas

Motivation: **Best results will be awarded**

The Award

Final grades

Grades depend on:

- attending ALL meetings
- regular work on the given tasks
- approach and **understanding** of the design !!!
- qualified presentations of results, conclusions, and optimizations

Best metric != best grade

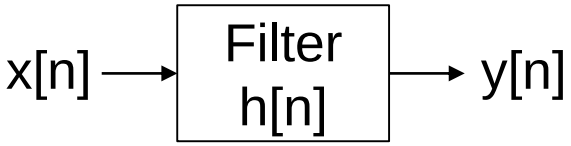
Digital Filters

- Use digital technology to perform numerical calculations on sampled values of the signal.
- The data processing might be processed by
  - a general-purpose processor such as a CPU,
  - a specialized DSP (Digital Signal Processor),
  - or in Hardware

Advantages of digital filters:

- Programmable (in SW) or fast (in HW)
- Extremely stable with respect both to time and temperature
- Easily designed, tested, and implemented on computers
- Can handle all frequency signals accurately
- Are very much versatile

Some Theory

- 

- Output value: $$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k]$$
- h[n]: impulse response of the system to input δ[n]
  (definition: δ[k] = 1 for k = 0, else δ[n] = 0)

Filters

IIR-Filter

- Recursive
- Infinite impulse response (possible)

FIR-Filter

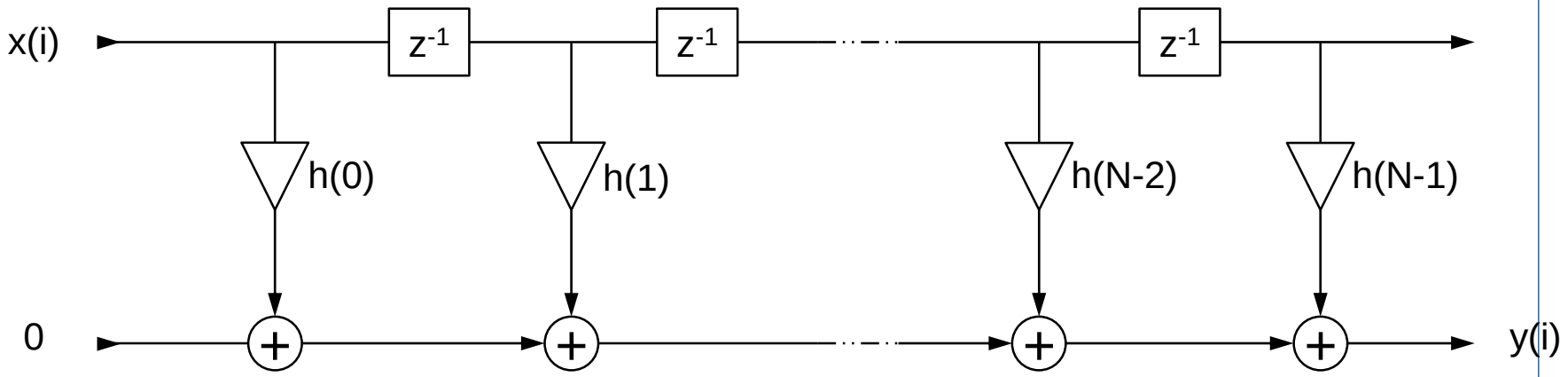
$$y[n] = \sum_{k=0}^{N-1} h_k \, x[n-k]$$

- Non recursive
- Finite impulse response (guaranteed)
- Inherently stable
- y[n] is the filter output at discrete time n
- h_k is the k-th feedforward tap or filter coefficient
- x[n-k] is the filter input delayed by k samples
- N: length of impulse response = number of taps in the FIR filter

FIR Filter Architectures

$$y[n] = \sum_{k=0}^{N-1} h_k x[n-k]$$

- **Direct form I**

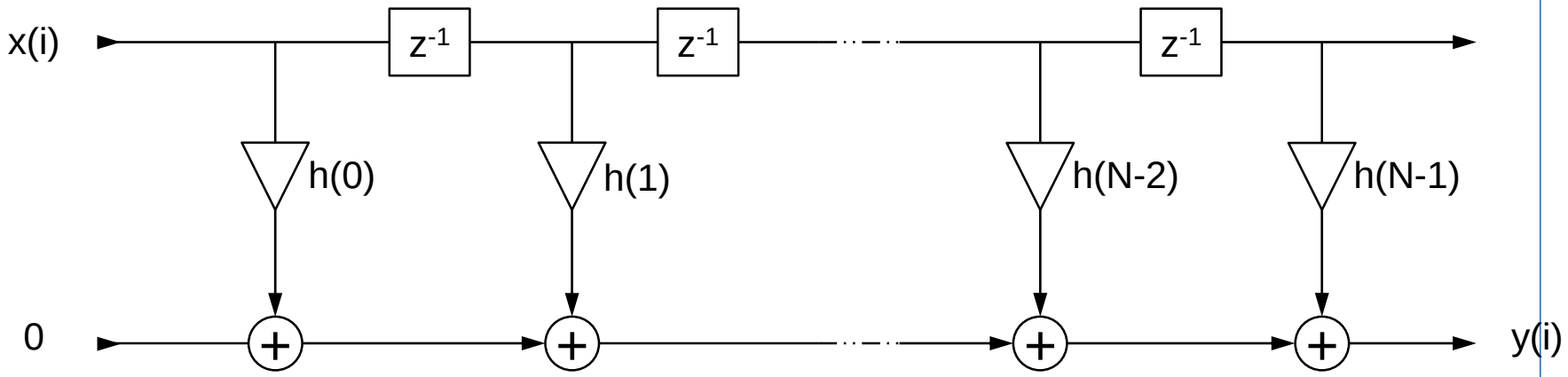


- z-1 : delay by one time sample, implemented by a register
- Drawback: N sequential additions yield tremendous latency
- Approach: use associativity of addition for speedup

FIR Filter Architectures

$$y[n] = \sum_{k=0}^{N-1} h_k x[n-k]$$

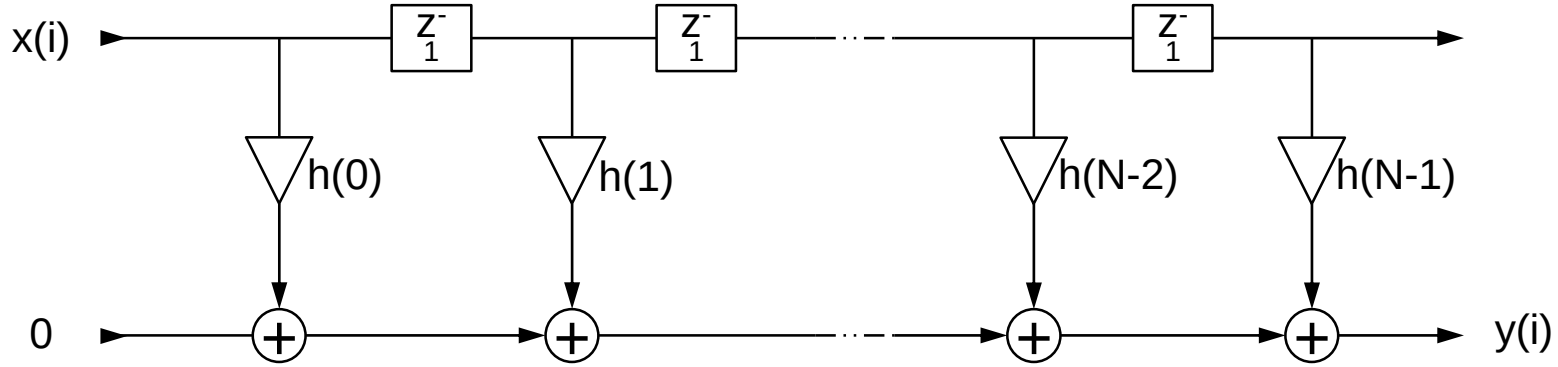
- **Direct form I**



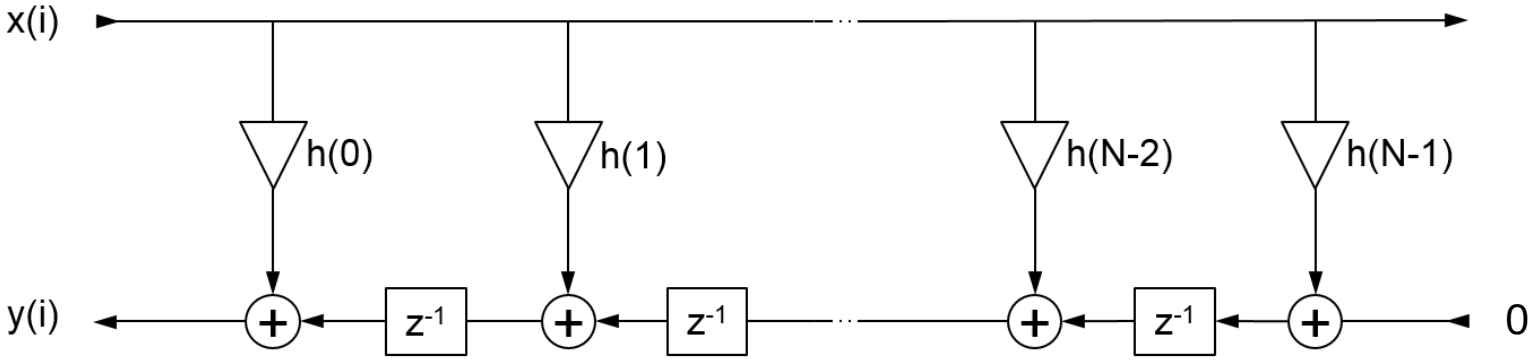
- y(i) = [h(0) + h(1)z-1 + … + h(N-2)z-(N-2) + h(N-1)z-(N-1)] x(i)
  = [(…(h(N-1)z-1 + h(N-2)z-1 + … + h(1))z-1 + h(0)] x(i)
  = [h(0) + z-1 (h(1) + z-1 (…h(N-2) + z-1 h(N-1))…)]] x(i)

FIR Filter Architectures

Direct form I



Direct form II



Project Info

Tasks, tools, sources

Project description

Given:

- Simple description of a 24 Bit FIR filter in VHDL
- Testbench and software to display filter output

Task:

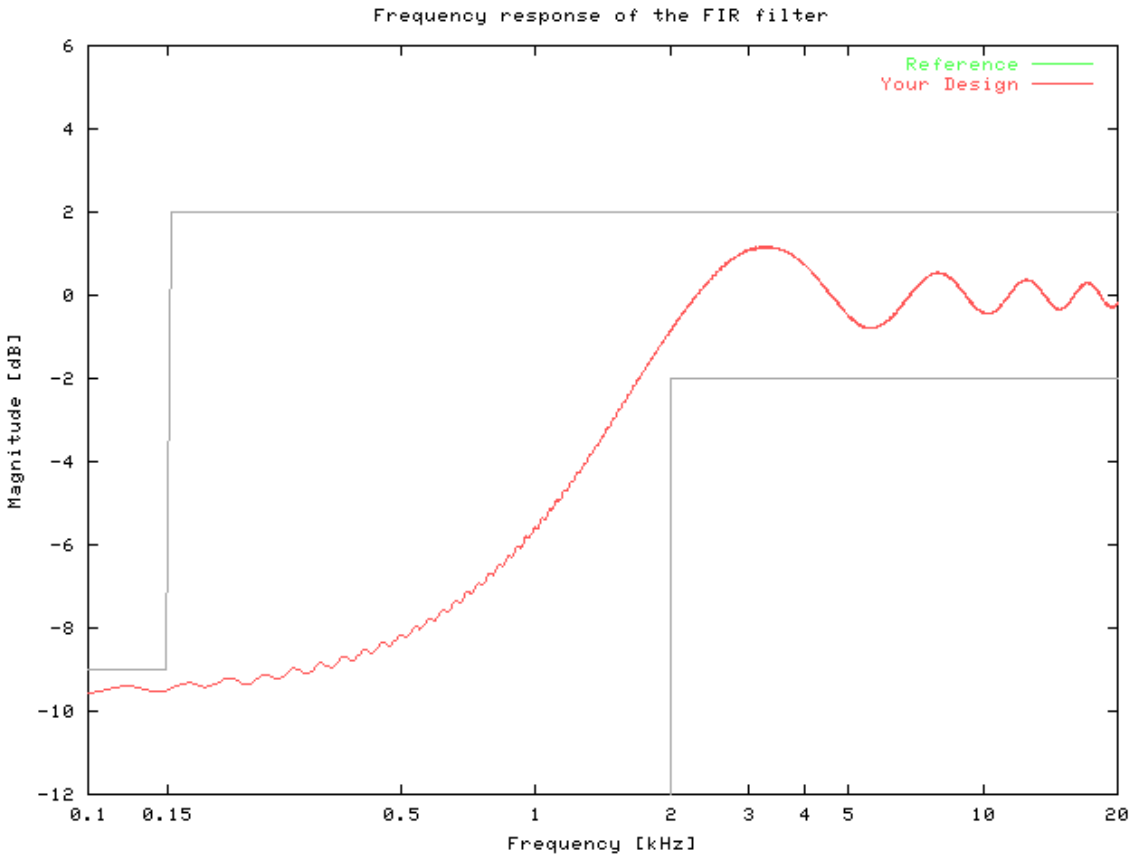
- Modification of VHDL description so that
- Replace ‘+’ and ‘*’ operators
- Synthesis for XILINX FPGA and ST65
- Chip layout in ST65 technology

-  Optimization for best metric

Filter Definition

Stop band:
150 Hz > f_s
Stop band gain < -9dB

Pass band:
1500 Hz < f_p
-2dB < Pass band gain
Pass band gain < 2dB



Five project phases

Phase 1:

- Direct implementation of FIR without using + and * operators
- Result: Your first working FIR filter

Phase 2:

- Achieve 100MHz
- Filter investigations

Phase 3:

- Architectural / Component refinement (CSD, term sharing, …)
- Result: awarded best final FPGA-Design

FPGA

Phase 4:

- Mapping on ST65 technology, exploiting synthesis options
- Result: A working ST65 netlist

Phase 5:

- Layout for ST65 technology, utilizing chip parameters
- Result: ST65 layout, backannotation (awarded best Layout)

ASIC

Metric

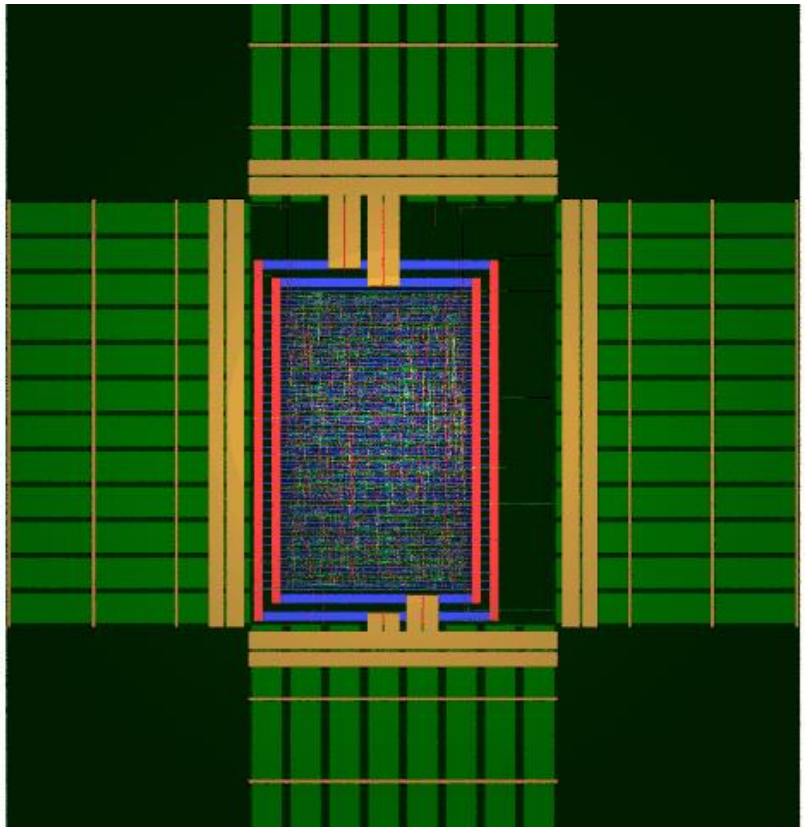
Phase 1-3 – FPGA:

$$Metric_{FPGA} = \frac{\text{Attenuation}}{(\#LUTs + \#FF)^3}$$

Goal – Final Layout

Exemplary chip layout from the project

- FIR-filter
- Low-Pass characteristic
- 16 Bit data width
- 8 taps with 8 Bit data width
- Design results
  - Frequency 595,9 MHz
  - Pipeline Depth 14
  - Core Size 69 500 µm²
  - Chip Size 924 000 µm²
  - Utilization: 71 %
  - Pads: 40
  - IO Pins: 34



Files – VHDL Design

- **fir_filter.vhd** → Reference design of the band-pass filter
- **fir_filter_tb.vhd** → Testbench (TB) for the design of the filter
- **your_filter.vhd** → Implement your design here
- **makros.vhd, syslog.vhd, fir_filter_tb_pack.vhd** → Packages with useful functions for the TB
- **SineVals.txt** → Text file with sample data of a sine oscillation as stimulus for the TB

Files – VHDL Design

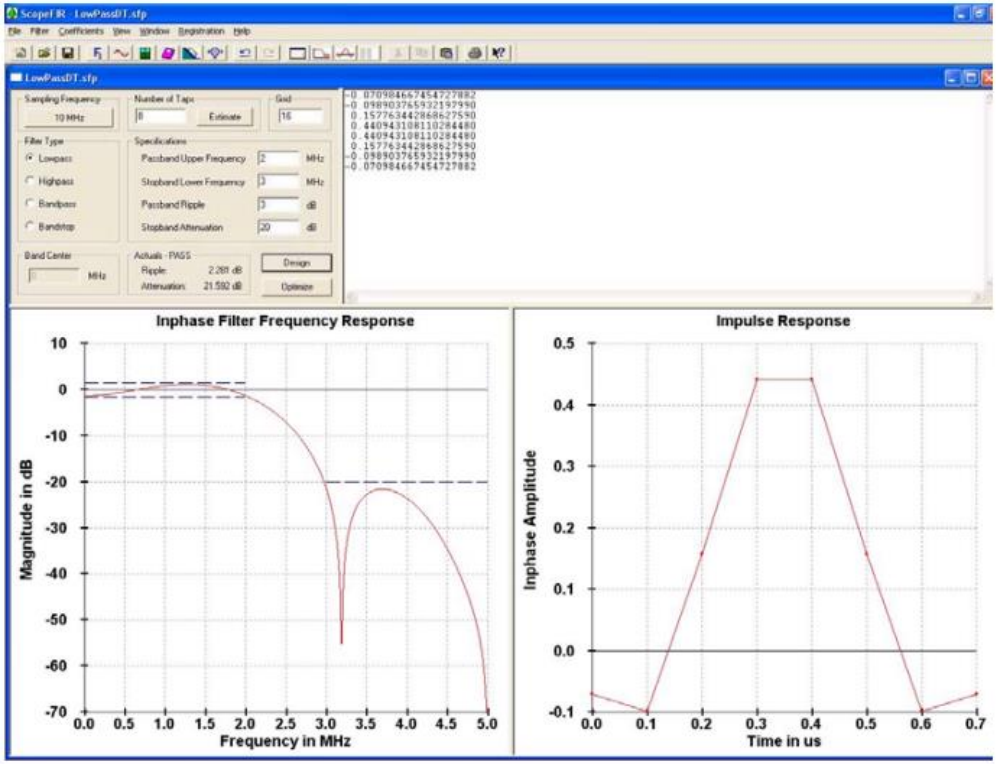
- **FirResults.txt** → Frequency response of the FIR-Filter (generated by the TB)
- **wgnuplot.exe** → Tool to generate graphics from output text files (png or ps)
- **FirResults.plt** → Script to create the graphic of the frequency response out of FirResults.txt (by using wgnuplot.exe)

First work steps

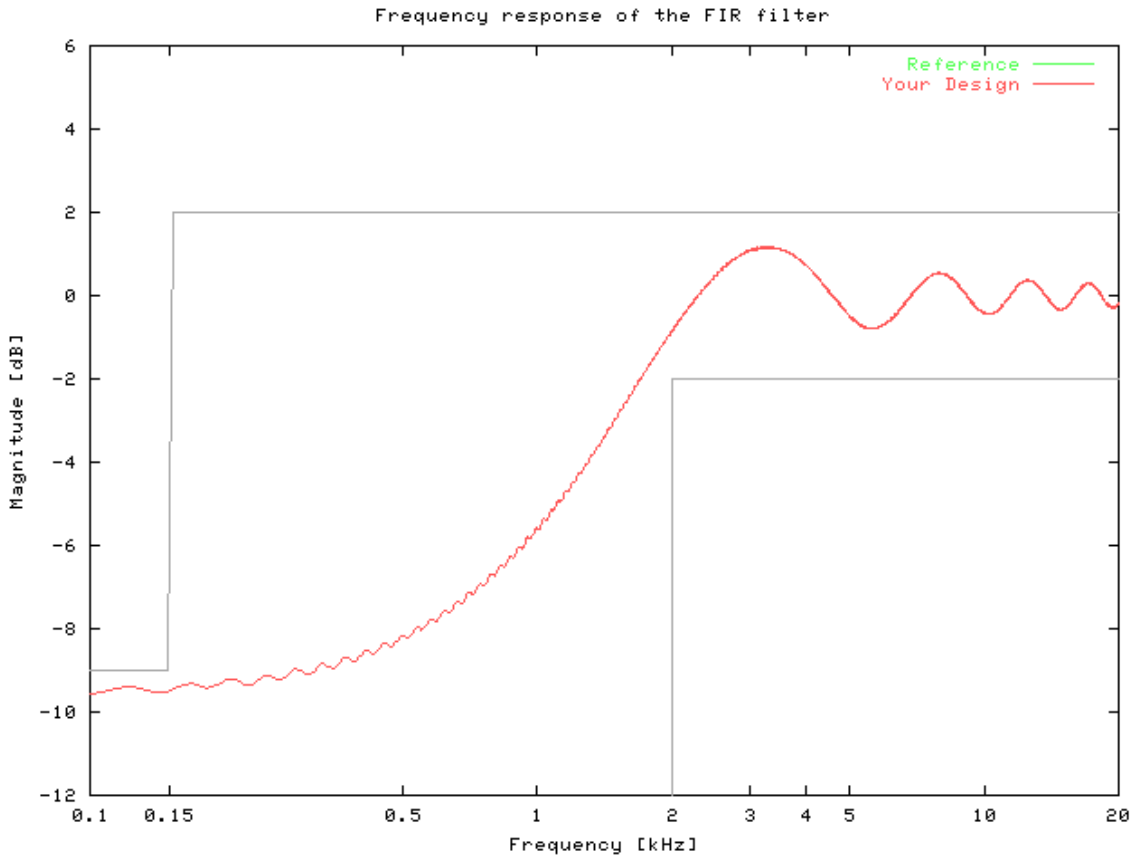
1. Compile your own design without using + and * operators
   → Advice: Start simple (… very simple)
2. Test your own design with the testbench against the reference design
   → Design is functionally checked for consistency (check for error messages)
3. Generating the graphic of the frequency response
   a) Start wgnuplot.exe
   b) Open/execute *.plt (Script)
      (Output can be set to png or ps in *.plt)
   c) [Convert *.ps to pdf]

Filter Definition

- Filter investigation and changes allowed
- Software tools:
  - MatLab
  - ScopeFIR
  - WinFilter



FIR filter - Frequency response



Example for the representation of the reference design and your own design in the frequency domain

Organisationally Informations

Form, schedule, remarks

Terms and conditions

- Independent work in the laboratories
- Attend all meetings (scheduled in 2-week interval)
- Execute all tasks within the given time
- Document your activities and results
- Present your conclusions and achievements (time limit: 5 min!)
- The **designs and corresponding data** have to be sent to Christoph Niemann and Hannes Raddatz **two days before** the presentation (christoph.niemann@uni-rostock.de, hannes.raddatz@uni-rostock.de)
  (corresponding data: metric, frequency, and pipeline depth)
- The **presentations** have to be sent to Christoph Niemann and Hannes Raddatz **one day before** the presentation

Remarks – Phase 1

Considerations for design handover

- Final synthesis has to be done with Vivado Version 14.4 (as it is installed in the laboratories)
- Target platform is Digilent ZedBoard
- Submit the following files: *.vhd and your_filter.syr
- Simulation with ModelSim
- Synthesis setting → -max_dsp= 0

Remarks - General

- Meetings: Thursday 11:00, R 1218
  - Check the dates and possible changes
- Work collaboratively
  - Share the work and help each other
- Think first, act afterwards
  - Trying menu options might lead to better results, but no better understanding
- Consider
  - Public holidays, weekends
  - Limited number of licenses, workstations
- Laboratory 1218 is reserved for the project
  - Mo. - Fr. 15:00 – 19:00
  - also before 15:00 possible if free
  - the building will be locked after 20:00, thus leave the building before that time
  - Do not lock computers when not used

Presentations

Each presenter has 5 minutes, last minute will be indicated.

Recommended flow for reasoning:

1. Observations: What do/did you observe in the design?
   - Problems, drawbacks, potential enhancements …
2. Do your observations leave room for enhancements?
   - What do you expect to change?
   - What does theory promise?
3. Present and discuss results of the implemented design
   - Conclude based on your results (e.g. approach was reasonable, unsuccessful)
4. Outlook on next steps/improvements
   - Is there room for further improvements? Why? What do you plan to do?

Schedule

Date	Milestone	Description
13.10.	Kick-off	Introduction and start with an exemplary design
17.10.	VHDL-Recap	Recapitulation of VHDL, tools and flow
27.10.	Meeting 1	Initial results for Xilinx, FPGA, VHDL
10.11.	Meeting 2	Results with backannotated optimizations for Xilinx FPGA
24.11.	Meeting 3	Final FPGA-Design
08.12.	Meeting 4	Netlist for ST65; Results for speed, power and area
11.01.	Submission	**!!! Submission of final design to Christoph Niemann!!!**
12.01.	Final meeting	Results from Cadence Tools (First Encounter) for final layout 18:00 Winner is invited to a dish of his choice.

Questions?

VHDL-Recap: 17.10.16
Next meeting: October, 27th

Aim:
Presentation of initial results for the transformation, design without + or * operator. (5 minutes each)