Practical assignment ECA2: filter architecture design using CλaSH

December 12, 2018

Introduction

Below the set of homework exercises for ECA-2 regarding filters, to be handed in by groups of two students. For some assignments you need a given input signal and/or a large low-pass window with filter coefficients, the files `inputSignal.txt` and `filterCoef.txt` contain this. We assume you successfully completed the tutorial (tutorial.pdf).

deadline for handing in your solutions: 2018-02-02, on canvas.

Remarks on delivery

- We have provided 4 files
  - Filters.hs – Main file for `clashi`, extend this file with your code
  - FilterCoefAndInput.hs – Contains filter coefficients in vector form and input signal in list form, you can import this file in your `Filters.hs` (*import FilterCoefAndInput*)
  - inputSignal.txt – Contains the input signal
  - filterCoef.txt – Contains the filter coefficients
- Add your names and student numbers at the top of the `Filters.hs` file.
- Add your names and student numbers to the front page of your report.
- In some assignments you are asked to draw a diagram, you may use any drawing tool you like. We used the free tool yEd (https://www.yworks.com/)
- For all assignments you are asked to deliver CλaSH code. Please include your CλaSH code for that specific assignment in a text box in your report.
- There is no need to deliver VHDL code.
- Use vector functions, reference documentation of Vectors: `hackage.haskell.org/package/clash-prelude/docs/Clash-Sized-Vector.html`
- In some assignments you are asked to describe/show the combinational path, here you need to draw this path in a figure.

- If you are asked to comment about clock cycles, we are looking for comments describing the amount of clock cycles for latency and throughput.
- You can score 20 points in this assignment. This is 2 points of the final grade.
- Deliver your report (pdf) and the CλaSH.
- Report name: `ECA2_filters_lastname1_lastname2.pdf`.
- The grading also depends on the style and readability of the code.

Preliminary Remarks on Haskell and CλaSH

- A filename of a Haskell/CλaSH program should be of the form `<Filename>.hs`, starting with a capital letter.
- You can transform Haskell to CλaSH by the following steps:
  - add the line

    import `Clash.Prelude`

    as the second line of your file. Among others, this redefines many standard list functions in Haskell towards corresponding vector functions in CλaSH. For example, in Haskell functions such as *take* and *map* work for lists, whereas in CλaSH they work for vectors. If you need such a function for lists, use *Data.List.take*, *Data.List.map*, etc. However, this only works in the `clash` interpreter, and not in a program file.
  - define the hardware types needed, using `Signed`, `Unsigned`, `Vec`, etc.
  - Instructions how to install CλaSH on your own system can be found on `clash-lang.org`.
- Finally, to generate VHDL code using CλaSH, you need to define the variable *topEntity* and make sure that its type is not polymorphic and fully determined. Note that Haskell (and thus CλaSH also) can derive the type of an expression, to be checked in Haskell/CλaSH with:

  ```
  :t <expression>
  ```

  VHDL code is generated by CλaSH using the command[1]

  ```
  :vhdl
  ```

  This will put the resulting VHDL code in a subdirectory `vhdl/<Filename>`. We assume that you have access to *Quartus* for synthesizing the VHDL generated by CλaSH.

[1] Don't bother about possible "`Can't make testbench`" errors, they are not relevant in our context.

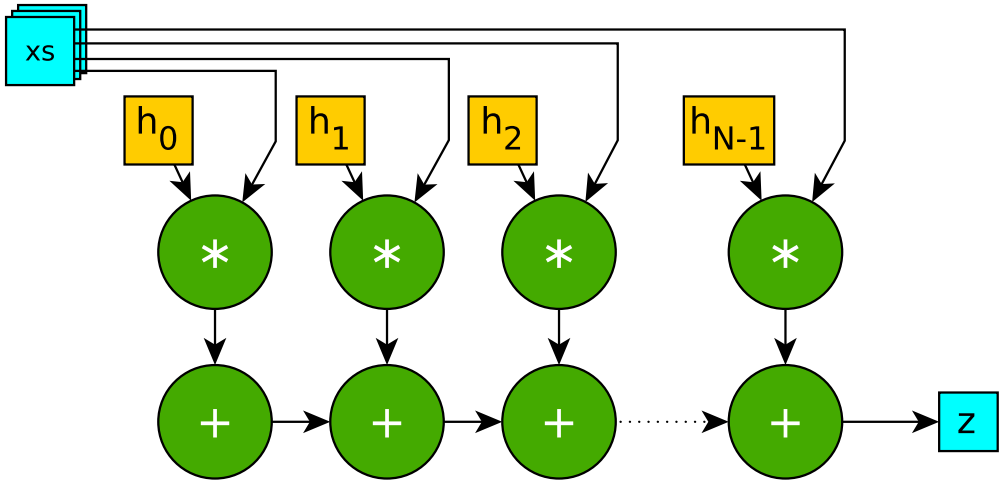

Figure 1: A layout of a FIR filter

1 Parallel FIR filter with N=6

Figure 1 depicts a structure of a FIR filter where all the inputs arrive simultaneously. The goal of the following assignment is to implement the same structured FIR filter in CλaSH with $N = 6$. Use `Signed 8` as number type for the values. Note that `xs` is an input vector with length `N`, and the `hs` variables are part of the state of the filter. `z` is an output (we ignore saturation). You can use any arbitrary number as filter coefficients (as long as the synthesis tooling does not optimize the multipliers away).

Assignment 1 (2 pts):

Create CλaSH code, generate VHDL, synthesise, show the Flow Summary, and the RTL schematic. Describe how the CλaSH code, resource consumption, and RTL schematic correspond to each other. Describe/show the longest combinational path.

2 A larger parallel FIR filter

To filter out a low frequency signal we use a low-pass filter, we make use of the following filter coefficients: (this is not a de-facto on how to construct filter coefficients):

$$l(n) = \begin{cases} \frac{\sin\left(2\pi f_t\left(n-\frac{M}{2}\right)\right)}{\pi\left(n-\frac{M}{2}\right)}, & \text{for } \left(n-\frac{M}{2}\right) \neq 0, \\ 2f_t, & \text{for } \left(n-\frac{M}{2}\right) = 0, \end{cases} \tag{1}$$

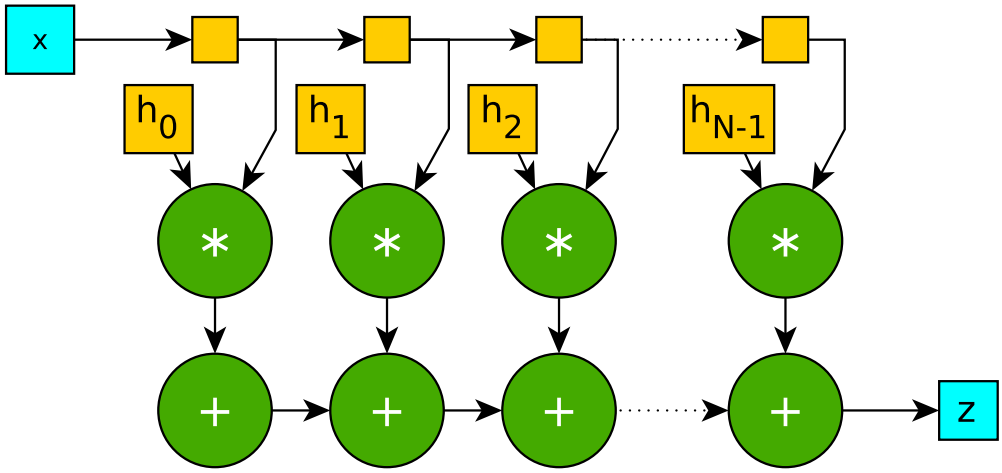

Figure 2: A transformed FIR filter

with the following constants:

$$\begin{aligned} \text{samplingFrequency} &= 2000, \\ \text{cutOffFrequency} &= 50, \\ \text{filterOrder} &= 100, \\ f_t &= \frac{\text{cutOffFrequency}}{\text{samplingFrequency}}, \\ M &= \text{filterOrder} - 1 \end{aligned} \tag{2}$$

The list with filter coefficients are in the file `filterCoef.txt`. The goal of this assignment is to implement a FIR filter with the same structure as in Assignment 1, but with a different filter window. Use `SFixed 5 13` as numerical type for all the values.

Assignment 2 (0.5 pts):

Create CλaSH code, generate VHDL, perform Analysis & Synthesis, show the Flow Summary. Comment on how the CλaSH code, resource consumption, and RTL schematic correspond to each other. Describe the longest combinational path. You do not have to place the entire RTL schematic in your report, just a fraction of it to explain the structure. (Note: The Fitter (Place & Route) will fail because there are not enough IO ports available on the device, only Analysis & Synthesis is required).

3 FIR filter with N=6

Figure 2 shows a FIR filter with registers on the input, which means that not all inputs arrive simultaneously. For the following assignment use the `Signed 8` number type for the values.

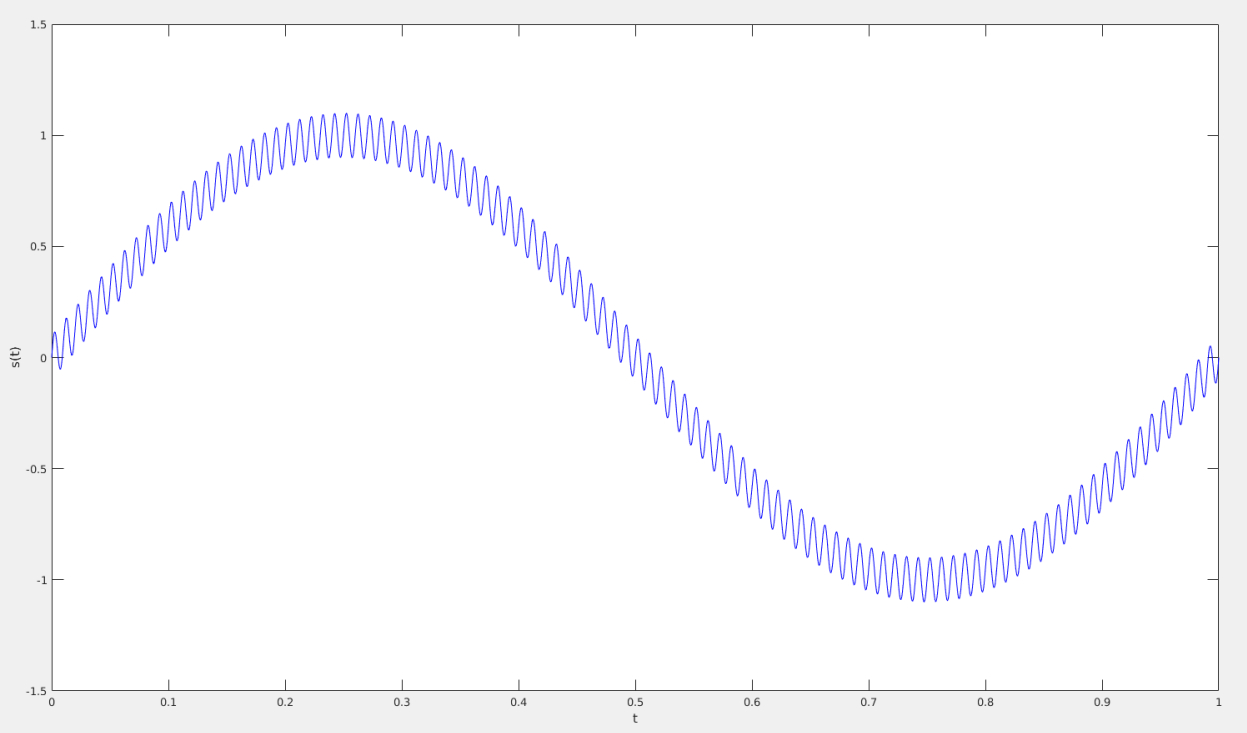

Figure 3: The input signal for Assignment 4 and 6.

Assignment 3 (2 pts):

Create the FIR in CλaSH with $N = 6$, generate VHDL, synthesise, show the Flow Summary, and the RTL schematic. Describe how the CλaSH code, resource consumption, and RTL schematic correspond to each other.

4 A larger FIR filter

Figure 3 shows the input signal:

$$s(t) = a_l \sin(\omega_l t + \rho_l) + a_h \sin(\omega_h t + \rho_h) \tag{3}$$

where:

$$\begin{aligned} a_l &= 1, & a_h &= 1, \\ \omega_l &= 2\pi f_l, & \omega_h &= 2\pi f_h, \\ f_l &= 1, & f_h &= 100, \\ \rho_l &= 0, & \rho_h &= 0. \end{aligned} \tag{4}$$

The following assignment is to create a FIR filter in CλaSH with the same structure as described in Assignment 3. Use `SFixed 5 13` as number type for values. For this assignment use the same filter coefficients as in Assignment 2. The `simulate` function can simulate designs in CλaSH. (simulate your `mealy` machine, not your `topEntity`) As simulation input use the input from Figure 3 (file

`inputSignal.txt` contains the input signal). You may use any plotting program (for example: MATLAB or GNUPlot) to plot the output signal.

Assignment 4 (0.5 pts):

Create the transformed FIR in CλaSH, generate VHDL, synthesise, show the Flow Summary, and show parts of the RTL schematic to clarify the structure. Describe the relation between the CλaSH code, resource usage, RTL schematic, clock cycles, and combinational path. Show how you used the `simulate` function, and give a plot of the result.

5 Symmetric FIR filter with $N = 6$

The low-pass filter used in the following assignments is symmetric (see equation (5), where x_n is the input signal, and h_k are the filter coefficients). In these assignments use the symmetry of the filter coefficients to reduce the hardware usage by a factor of 2.

$$h_0x_0 + h_1x_1 + h_2x_2 + h_2x_3 + h_1x_4 + h_0x_5 \tag{5}$$

Assignment 5 (3 pts):

Design an FIR filter with the following hardware constraints:

- 1 input (Signed 8)
- 3 multipliers (DSPs)
- Low-pass window length 6 (Signed 8 values)
- 6x8 registers

Describe the optimization algebraically. Create CλaSH code, Generate VHDL, synthesise, show the Flow Summary, and the RTL schematic. Describe the relation between the CλaSH code, resource usage, RTL schematic, clock cycles, and combinational path. Describe the ratio between the amount of adders and the amount of multipliers needed. Compare the longest combinational path with the one from Assignment 1 and comment on the difference.

6 Symmetric FIR filter

In this assignment use the structure from Assignment 5, the filter coefficients from Assignment 2, and the input from Assignment 4. You can use `SFixed 5 13` as number type.

Assignment 6 (1 pts):

Create the FIR in CλaSH, generate VHDL, synthesise, show the Flow Summary, and show parts of the RTL schematic to clarify the structure. Describe the relation between the CλaSH code, resource usage, RTL schematic, clock cycles, and combinational path. Show how you used the `simulate` function, and give a plot of the result.

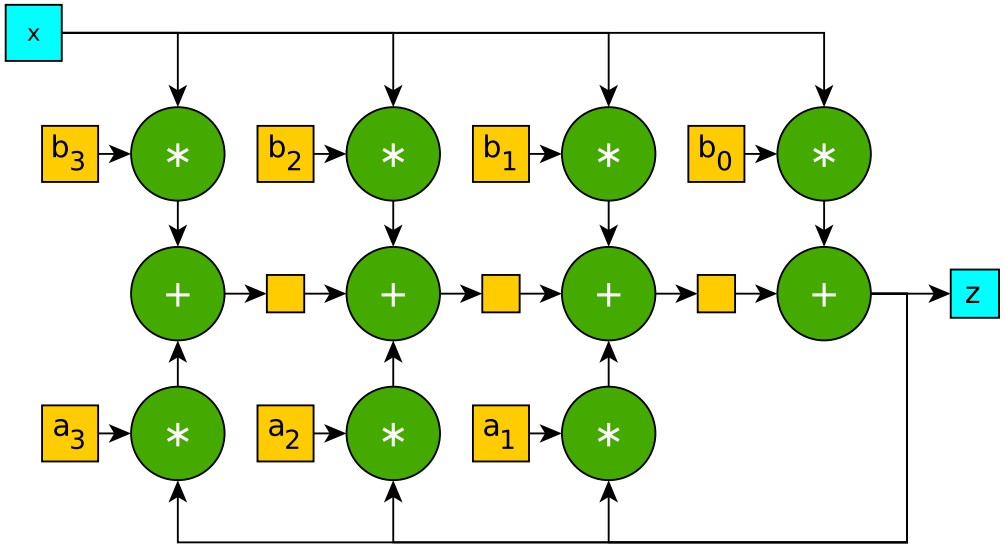

Figure 4: IIR filter structure

7 Transformed symmetric FIR filter with $N = 6$

For this assignment the filter from Assignment 5 is used. In practice the following transformation steps keep the functionality of the filter the same.

- Reverse the direction of all lines
- Interchange input and output
- Replace junction points by adders and adders by junction points

Assignment 7 (3 pts):

Perform the transformation as described above and show the diagram. Create the transformed FIR in CλaSH, generate VHDL, synthesise, show the Flow Summary and RTL schematic. Comment on the clock cycles, and combinational path with respect to Assignment 5

8 IIR filter

Figure 4 shows a structure of an FIR filter. In the following assignment implement such a filter with the following coefficients (`SFixed 5 13`):

$$bs = [b_0 = 0.0623348, b_1 = 0.1870044, b_2 = 0.1870044, b_3 = 0.0623348] \quad (6)$$

$$as = [a_1 = 0.9853304, a_2 = -0.5929545, a_3 = 0.1089457] \quad (7)$$

Assignment 8 (4 pts):

Create the IIR in CλaSH, generate VHDL, synthesise, show the Flow Summary, and the RTL schematic. Describe the relation between the CλaSH code, resource usage, RTL schematic, clock cycles, and longest combinational path. Use as input signal [1,0,0,0,...] and give a plot of the result.

9 Transformed IIR filter

For this assignment the same transformation technique is used as in 7.

Assignment 9 (4 pts):

Perform the transformation as described in 7 and show the diagram. Create the transformed IIR in CλaSH, generate VHDL, synthesise, show the Flow Summary and RTL schematic. Comment on the resources, clock cycles, and combinational path with respect to Assignment 8. Show the simulation with the input [1,0,0,0,....] and give a plot of the result.