Introduction to Electronic System Design (DAT093) Lab 6: A multirate system

Sven Knutsson, Lars Svensson

Version 2.1, October 12, 2018

1 Introduction

This final lab session of DAT093 combines the sampled system of Lab 5 with the FIR filters of Lab 2. In Lab 5, the stream of samples from the ADC was simply sent back out to the DAC unmodified. Here, the ADC and DAC will use *different sample rates,* which necessitates interpolation, decimation, and low-pass filtering to avoid frequency aliasing.

You may find this lab PM quite short and lacking in detail. This is intentional. The knowledge and skills gathered during the preceding labs, together with access to the Chalmers electronic library and the rest of the Internet, should be sufficient for completing the lab. If you get stuck anyway, the teaching staff is here to help.

The intentional lack of detail means that you will need to make some decisions in order to arrive at your final design. During this process, *make notes of your considerations,* not only the results of your deliberations! After completing the lab, you may write a short report which can gain you some bonus grading points; your design considerations are a valuable part of that report. Refer to the course website for details.

2 Preparation

With your lab partner, revisit the FIR filter designs you produced during Lab 2. Compare your designs and decide which one to use here[1]. Especially consider the support for generic filter length. Make notes of your considerations!

Two simple cases of sample rate changes are integer-factor *upsampling*, such that the new sample rate $f_{S.new} = f_{S.old} \times I$, and *downsampling*, such that the new sample rate is $f_{S.new} = f_{S.old}/D$, where I and D are the integer-valued interpolation and decimation factors, respectively. Read up on these operations, for example starting with the uploaded PowerPoint presentation and/or with the Wikipedia articles. Make sure you understand the use of lowpass interpolation and decimation filters, respectively, and how to select their cutoff frequencies.

It is obviously possible to cascade an upsampling and a downsampling operation to change the sample rate of a signal by a rational number M/N. When no other processing will be carried out at the higher frequency, the interpolation and decimation filters will be directly connected, and may therefore be replaced with a single lowpass filter to save hardware. How will you determine the necessary cutoff frequency for the combined filter?

3 System design

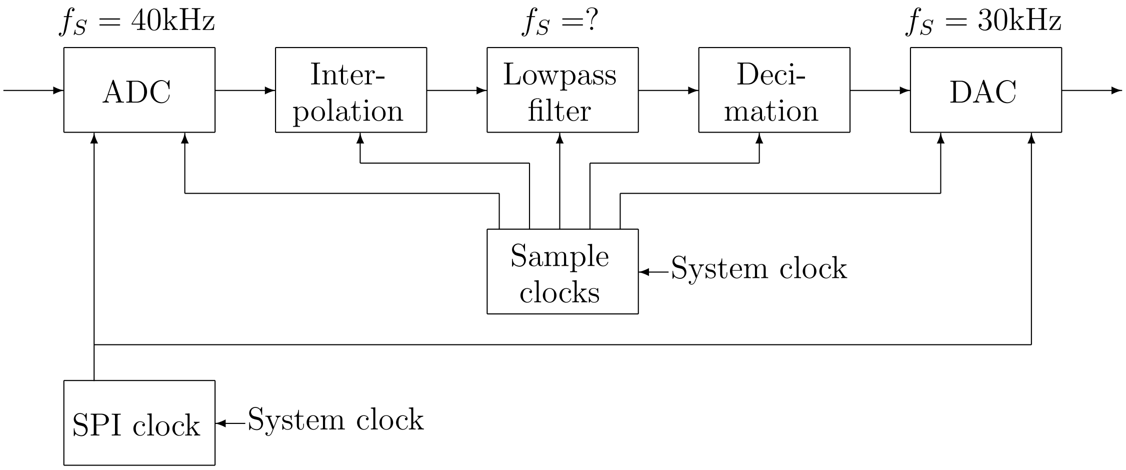

Figure 1: A mixed-rate system

Figure 1 shows a block diagram of the mixed-rate system to be designed and

[1] You may feel that your skills have improved significantly since Lab 2, and that neither of your Lab-2 designs should be used here. Feel free to re-implement the filter in that case.

implemented. Most of the blocks have been designed in previous labs and should be reusable with little or no modification.

- Identify the blocks needed to implement the system in Figure 1, and make note of any necessary modifications and extensions. For example, to generate the different sample clocks, you will need to extend your clock-enable module to several outputs. (An alternative could be to use several modules with one output each.)

Preparation: Starting with the input and output sample rates as given in Figure 1 and taking into account the integer-factor requirement, determine a usable intermediate sample rate. *Note:* the clock-generation method used in the previous labs can only generate sample rates with an integer relation with the system clock, which will not be sufficient to achieve the exact desired clock rates. Select rates as close as possible to the desired ones.

The frequency response of a sampled filter is specified in relation to the sample rate.

Preparation: From the intermediate sample rate and your considerations of the specifications of the combined lowpass filter, determine the filter cutoff frequency as a fraction of the intermediate sample rate.

Given the cutoff frequency, the next task is to determine a suitable filter impulse response. In general, a longer filter (with more filter taps) will be able to do a better job of suppressing the undesired frequencies. There is therefore a cost-performance tradeoff which is fruitful to explore.

Many software tools can be used to synthesize a filter impulse response to approximate a desired frequency response. For example, the MATLAB[2] DSP System Toolbox is extremely capable and is a good choice if you are already familiar with it. Simpler alternatives may be found on the Internet; for example, TFilter, found at `http://t-filter.engineerjs.com/`, is easy to get started with. Feel free to use some other tool if you so prefer. Document your choice of tools, your choice of filter parameters, and your experiences during filter design.

Preparation: Use your chosen tool to derive tap values for the FIR filter, for some filter lengths from 15 to 30 taps. Document your results. Convert the filter coefficient sets to fractional-numbers format.

[2] `http://www.mathworks.com`

To reduce potentially wasted time, discuss your solutions with the lab TA before major implementation effort.

4 System implementation

The work in the lab follows the familiar routines. *Note:* the ADC and DAC work with unsigned binary numbers, where 000...0 is the lower end of the value range. The filters use the two's-complement form, where 000...0 is in the middle of the range; it is therefore necessary to convert the numbers before and after the filter. Fortunately, conversion from one representation to the other is as simple as inverting the most-significant bit.

- Implement and test the system of Figure 1, following the practices from the previous sessions. It is recommended that you work incrementally, starting from the Lab-5 solution and adding blocks one by one in a suitable order. Document your progress.
- As in Lab 5, apply a sinewave signal to the ADC input and use an oscilloscope to observe the input signal together with the DAC output signal. Verify that the output signal follows the input, and that the sample frequencies are according to specifications.

Show your working system to the lab TA to be marked as passed on the lab.

5 Wrap-up

After completing this lab session, you are expected to be able to carry out the following tasks:

- Design and implement hardware to change the sample rate of a signal.
- Design hardware which generates the miscellaneous clocks necessary for a multirate signal processing system.
- Use available software tools to derive FIR filter tap values starting from filter specifications.
- Re-use previously-designed hardware blocks in a new situation and modify them to suit.

The overall sample rate change specified here is well supported with the relatively simple hardware that you have designed. In your post-lab report, consider also the following questions:

- What problems might arise for a much larger sample rate change, such as a factor of ten? One hundred?
- A common multirate system involves the conversion of audio signals from the CD standard sample rate of 44.1kHz to the Digital Audio Tape (DAT) sample rate of 48kHz, used by much studio equipment. If you were to use your solution for such a rate conversion, the internal sample rate would be quite high (how high?). Can you think of a way to ameliorate this situation?