TSTE87 Laboratory Work – Lab 2

Oscar Gustafsson, Kenny Johansson, and Erik Bertilsson

- Student name: ....
- Student personal id: ....
- Passed: ....

Goals

To see the relationship between multi-rate and single-rate realizations of the same algorithm. Using polyphase decomposition to derive the different realizations.

Preparations

1. Read Section 4.25 in Wanhammar, DSP Integrated Circuits.
2. Read "Additional commands in the DSP toolbox" below.
3. Show that it is possible to move the negation in a first- and second-order allpass section based on symmetric two-port adaptors from within the allpass section to the output if the signs of the adaptor coefficients are negated.
4. Mark which input that is port 1 for all the twoports in the design that is to be implemented in Task 3 (see note in Laboratory work 1).

Additional commands in the DSP toolbox

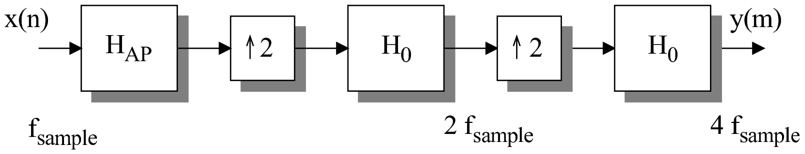
To cascade a two SFGs, the command `cascadesfg` can be used as

```
cascadedsfg = cascadesfg(sfg1, sfg2)
```

where the output(s) of `sfg1` is(are) connected with the input(s) of `sfg2`. Naturally, the number of outputs of `sfg1` must be the same as the number of outputs of `sfg2`. If there is more than one input/output, they will be connected in increasing id-order. Hence, if `sfg1` has two outputs with id 1 and 3 and `sfg2` has two inputs with id 2 and 4, the output with id 1 will be connected to the input with id 2 and the output with id 2 will be connected to the input with id 4. Note that certain nodes may have a different node number in the joined SFG, to avoid multiple nodes with the same number.

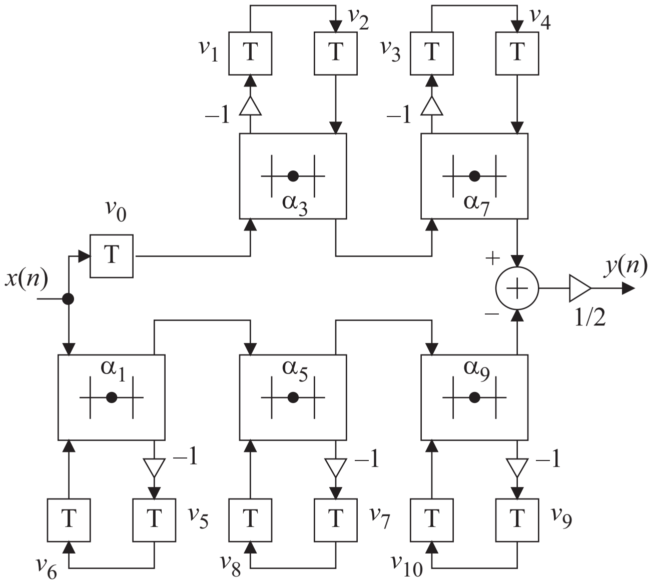
Tasks

The remainder of the laboratory work series will focus on implementing an interpolation filter derived from Case Study 3 in the book DSP Integrated Circuits. The filter consists of a wave digital filter allpass section and two identical bireciprocal lattice wave digital filters (LWDFs). The structure of the filter is shown below.



The filter should interpolate an input signal with 1.6 MHz data rate four times.

1. The first task will be to establish a reference realization for the filter. To keep it simple we realize each filter stage in a straightforward manner. To assist in this there are two files in `/courses/TSTE87/labs/lab2`, namely `interpolatorallpass.m` and `interpolatorbireciprocal.m`, which contains source code for generating the SFGs for the different filters. As the DSP toolbox only can handle one sample rate you will have to simulate each filter separately. Compared with Case Study 3 all negations are removed from the filter to obtain a simpler implementation. The adaptor coefficients in the allpass filter (see task 2 for a figure of the SFG) are: $\alpha_{10} = 0.4573$, $\alpha_{11} = -0.2098$, $\alpha_{12} = 0.5695$, $\alpha_{13} = -0.2123$, $\alpha_{14} = 0.0952$, $\alpha_{15} = -0.2258$, $\alpha_{16} = -0.4490$. For the bireciprocal filter, shown below, the adaptor coefficients are: $\alpha_1 = -0.068129$, $\alpha_3 = -0.242429$, $\alpha_5 = -0.461024$, $\alpha_7 = -0.678715$, $\alpha_9 = -0.888980$.



- Simulate the impulse response for the complete interpolator filter. Use the function `upsample` in MATLAB to expand the signal (insert zeros).
- Plot the frequency response both for the signal after the first interpolator stage and for the complete interpolator filter.

- What is the passband edge for these two cases?

  ....

- Quantize the adaptor coefficients to eleven fractional bits and plot the frequency response. **Use the quantized coefficients for the remainder of the laboratory works.**

  Quantization of a coefficient, α, to W_f fractional bits can be done in one of the two following ways

  ```
  round(alpha * 2^Wf) * 2^-Wf
  quant(alpha, 2^-Wf)
  ```

- How does quantization affect the magnitude response?

  ....

- Simulate the complete interpolator filter using a random signal and save both the input and output data. These signals will be used for reference at later design iterations.

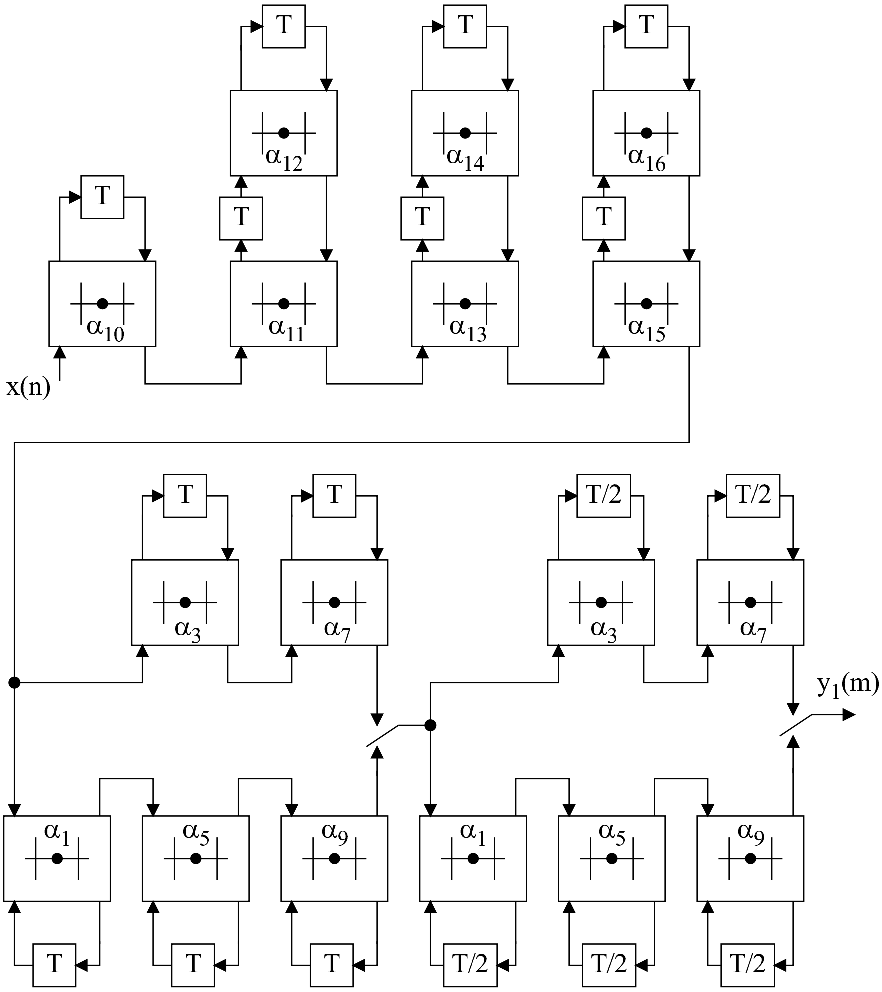
- Plot the discrete delay element values for one of the delays in the last stage bireciprocal filter and for the discrete inputs to the adder in the last stage bireciprocal filter. Comments?

  ....

- Determine the number of adaptor operations required per second.

  ....

2. As an intermediate design iteration we will apply polyphase decomposition to the bireciprocal filters. This results in the filter structure below. Now there are only two different sample rates, as the allpass filter and the first bireciprocal filter runs at the same sample rate. Remember to use the quantized coefficients.



- Create the SFG. This can, e.g., be done by modifying the SFG:s used in Task 1.
- Simulate the impulse response and using your reference signal to validate that the filter has the same function as the first design iteration. To interleave signals you can use `reshape`. Note that it is important to take the first value from the correct branch.

  A commutator with two input branches, x_1, and x_2, and an output, y, can be implemented as

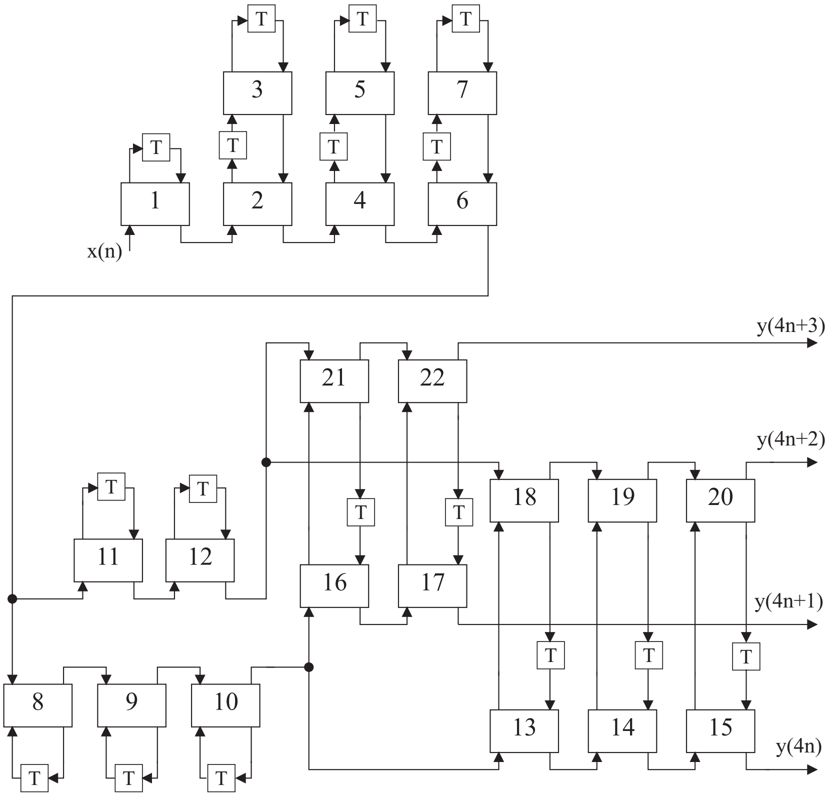
  ```
  y = reshape([x1;x2], 1, 2*length(x1))
  ```

  or alternatively

  ```
  y = [upsample(x1, 2), 0] + [0, upsample(x2, 2)]
  ```

- Determine the number of adaptor operations required per second.

3. Finally, we will realize a single rate version of the interpolator. This is shown in the figure below. This is also the SFG to be used for subsequent laboratory works.



- Create the SFG. Identify adaptor coefficients by comparing with Task 2.
- Simulate the impulse response and using your reference signal to validate that the filter has the same function as the first design iteration. Save the output from the impulse response simulation, to be used in the final question.
- Determine the number of adaptor operations required per second.

....

- Plot the precedence graph of the filter. How many two-port operations are there in the critical path?

....

- Introduce pipelining such that the critical path is at most three adaptors. Mark in the figure where delays are introduced. Check by plotting the precedence graph.
- Simulate the impulse response for the pipelined filter. Compare the phase response with the non-pipelined design by plotting both in the same figure. Comments?

....