Digital Systems Design Coursework

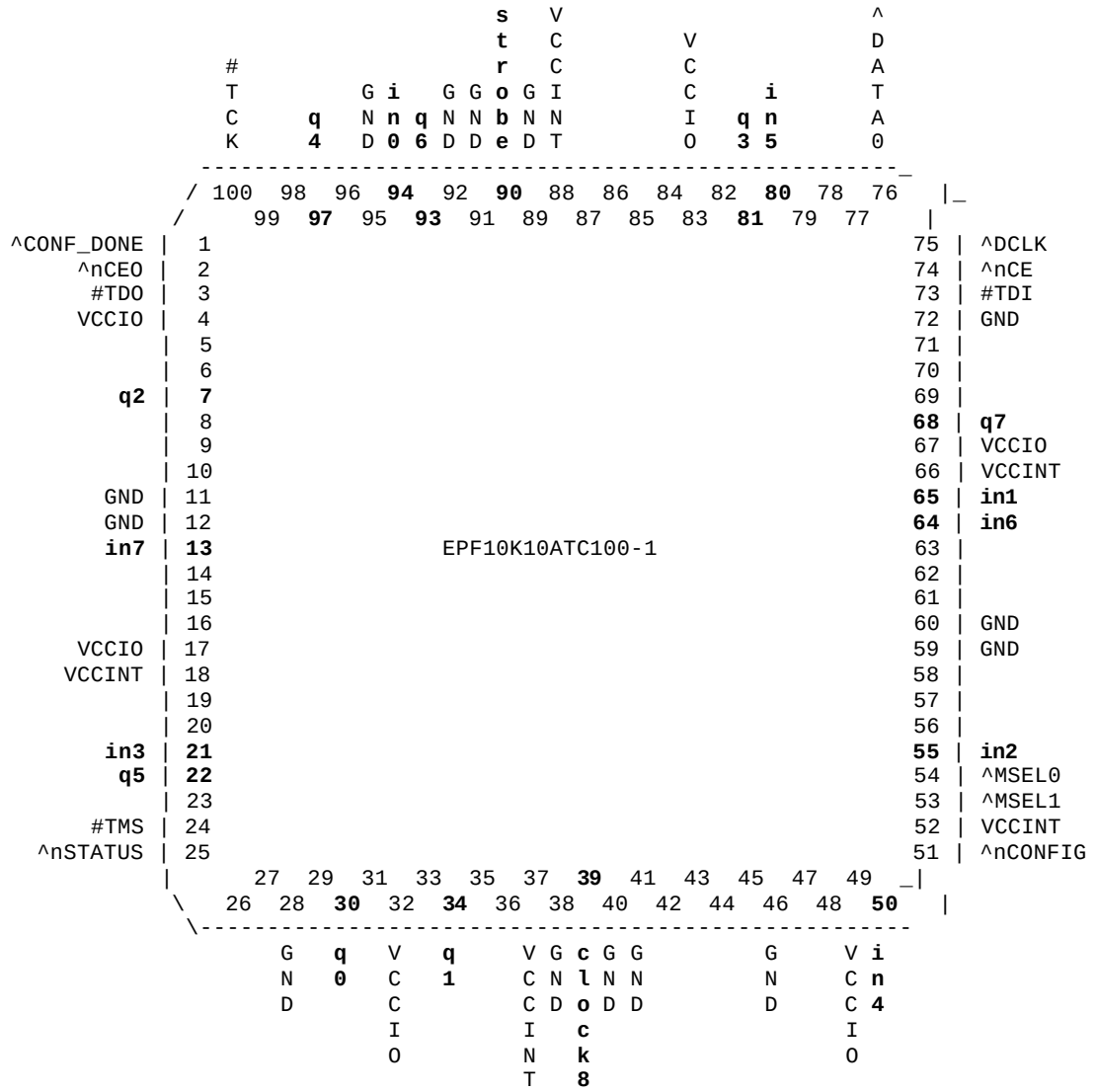
Data Sheet

Description

This chip implements a biquadratic filter on an Altera FLEX10KA. The finished version of the filter has been programmed with the filter coefficients that describe a notch filter at a frequency of p/4. The filter accepts *8-bit, 2's complement, fixed point* binary inputs and provides the same at the output. Inputs / outputs are therefore, within the range -4 = x = 3.9687500 in steps of 0.0312500.

$$2^{2}x_0 + 2^{1}x_1 + 2^{0}x_2 + 2^{-1}x_3 + 2^{-2}x_4 + 2^{-3}x_5 + 2^{-4}x_6 + 2^{-5}x_7$$

Pin out

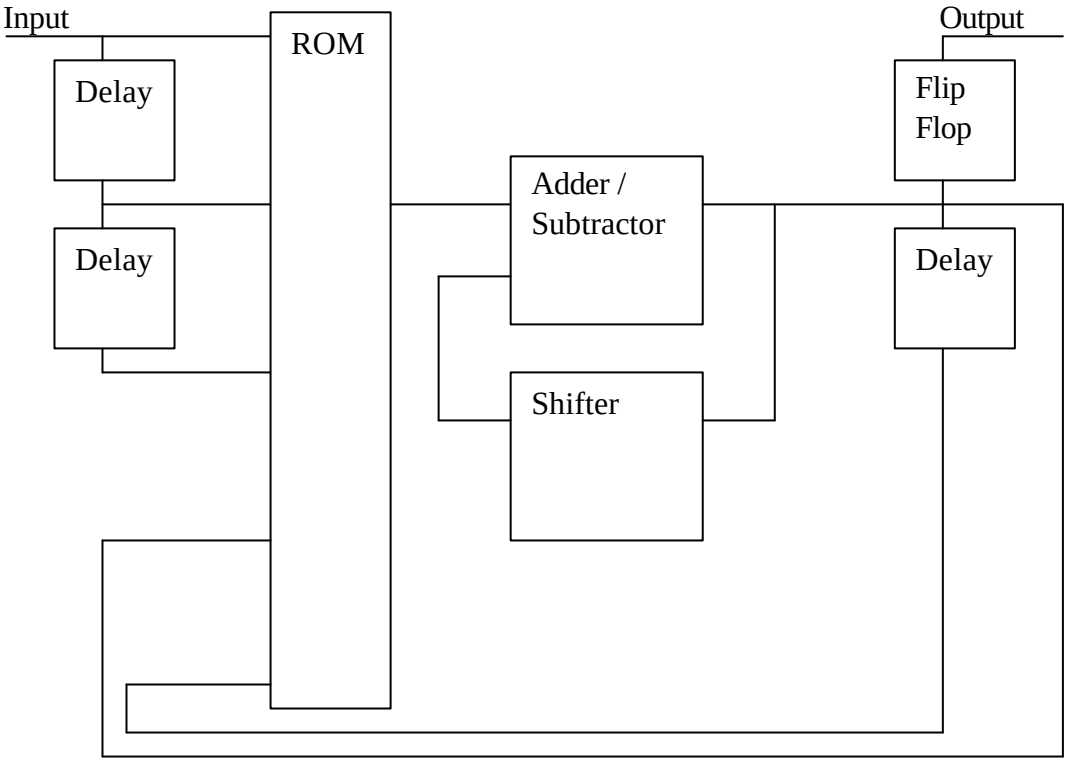


Unmarked pins are reserved and should not be used.
Pins in bold are data I/O pins, clocks, strobes, etc.
Other pins are for power and diagnostics.

Principle of operation

A clock waveform should be supplied on the *clock8* pin. Inputs should be offered up, a sample at a time, to the *in{0..7}* pins. When the input pins have been set they can be strobed into the chip by holding the *strobe* pin high for one clock cycle. The user must ensure that a rising edge of the clock occurs in the middle of the strobe. The input pins only need to be held constant for the duration of the strobe. After the first sample has been strobed into the filter, subsequent samples must be strobed in every 10 clock cycles. The filter induces a latency of one sample into the sample stream. (i.e. 10 clock cycles.) Outputs appear on the *q{0..7}* pins on the rising edge of the strobe and remain there until the rising edge of the next strobe. The filter can be purged completely by waiting at least 30 clock cycles before continuing to strobe data in and out of the filter.

The filter is implemented with distributed arithmetic. As the sample bits come into the filter they enter a delay line and are tapped off as and when they are needed. Corresponding bits from each of the samples currently residing in the filter are then applied to the inputs of a ROM. The resulting data values for each of the sample bits are summed across a whole sample, with respect being paid to the most significant bit of each sample, which is the "sign bit". The final output values are then held on the output lines and also fed into another delay line and tapped of as future inputs to the filter.



Design decisions

This filter processes each bit from the sample in serial. This means that it takes 8 clock cycles to calculate the output for each sample. It then takes a further two clock cycles for the output to propagate through the output flip flops and get injected into the output delay line. The output delay line saves historic output values for later reuse as inputs. Total time taken to process each sample is therefore 10 clock cycles. By

working with several bits in parallel it would be possible to reduce the processing time. For example, if two bits were processed simultaneously, the total time taken would be 6 clock cycles. If four bits were considered this would reduce to 4 clock cycles, etc. However, this increase in speed would come at the expense of circuit complexity and chip area. Each doubling in processing power would double the amount of circuitry needed. i.e. 2 identical ROMs would be required, etc, etc. It is also possible to reduce the size of the ROM by half as outlined at http://access.ee.ntu.edu.tw/course/VSP/SLIDE/6%20DISTRIBUTED%20ARITHMETIC.PPT This technique would be useful for reducing the memory resource usage if the parallel design above was to be implemented. However, the basic design as it stands is not very big and there is more than ample ROM available on a FLEX10K.

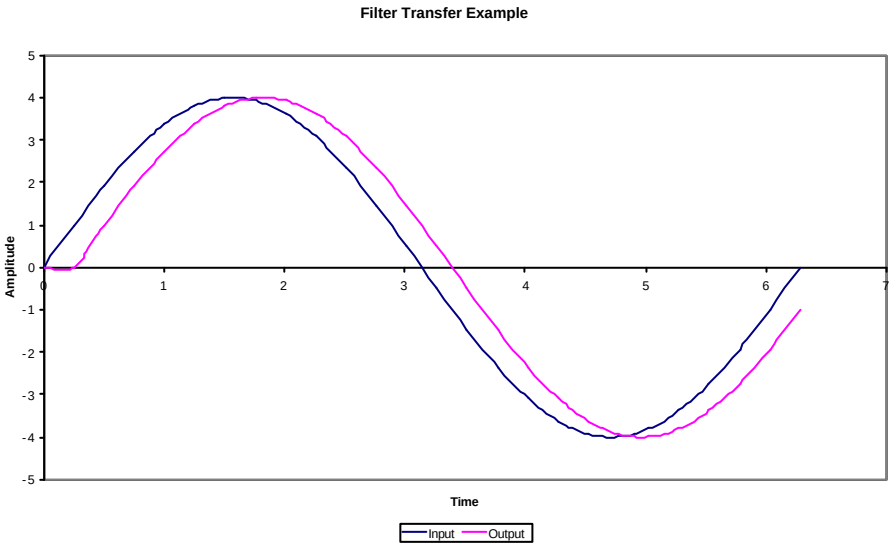
It was decided to require the user to supply a strobe input so that it was not necessary to hold the sample on the input pins all the time and also so that the user could specify when they wanted to use the filter. The filter can be used at an arbitrary time (given 30 clock cycles to flush out any extraneous values) without fear of the circuit being a fraction of a sample out of sync with the rest of the circuit in which it is used.

The circuit was initially built using 74-series macro functions (A.K.A Old-Style Macrofunctions). However, it was discovered that the LPM series of megafunctions were more easily manipulated and often implemented in a less archaic and more portable manner than the 74-series. Although portability is not required for this design, because it was specified for synthesis on a FLEX10K, it is good practice.

Simulation results

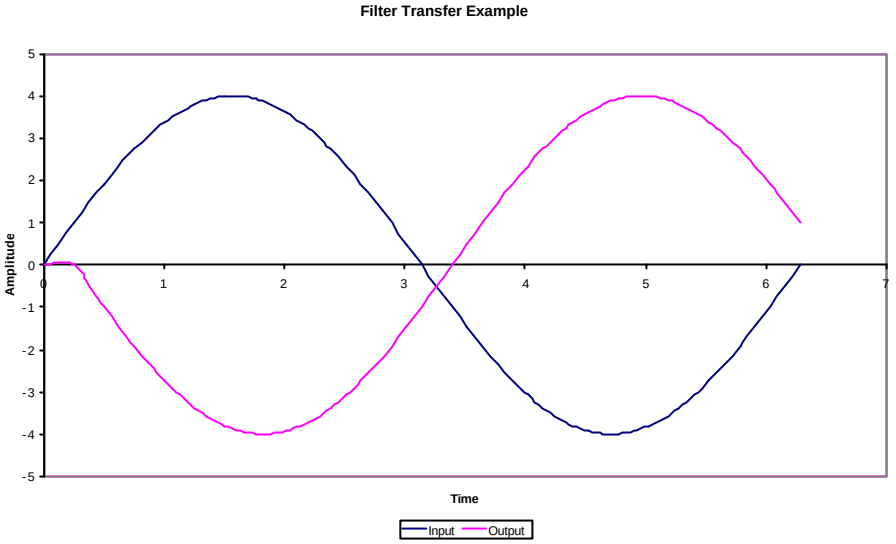
In order to ease testing, the filter was constructed in parts and then assembled. Initially the ROM was initialised such that the filter was an amplifier with a gain of one:

A_0: 1
A_1: 0
A_2: 0
B_1: 0
B_2: 0



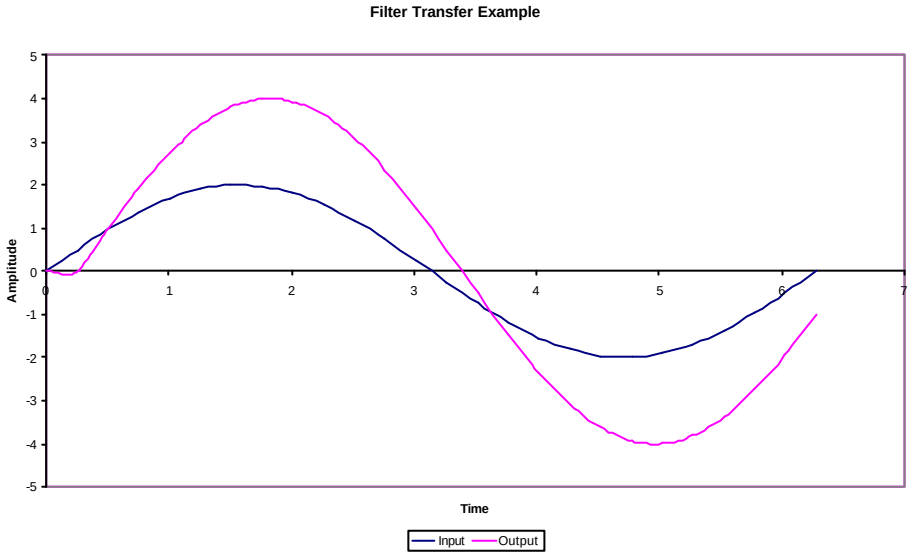
The filter was then set up with a gain of negative one:

A_0: -1
A_1: 0
A_2: 0
B_1: 0
B_2: 0



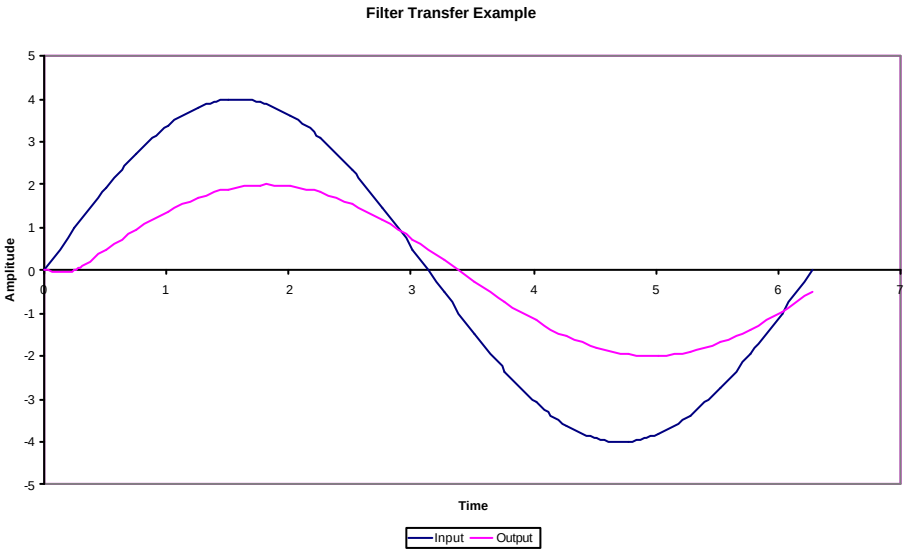
With a gain of two:

A_0: 2
A_1: 0
A_2: 0
B_1: 0
B_2: 0



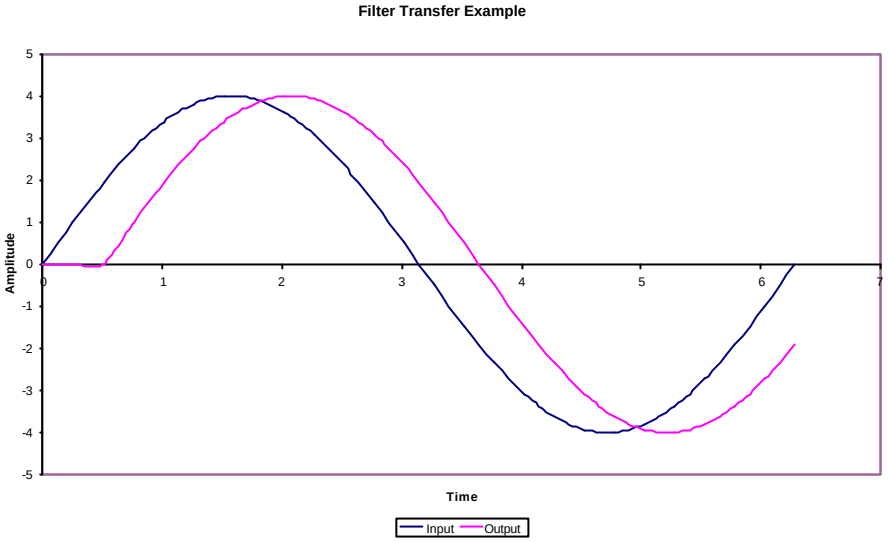
With a gain of one half:

A_0:½
A_1: 0
A_2: 0
B_1: 0
B_2: 0



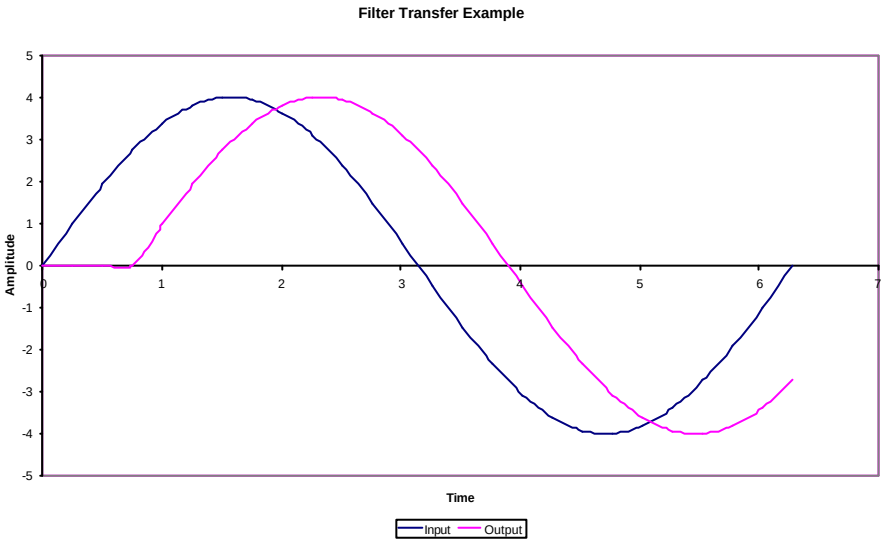
As a delay by one sample:

A_0: 0
A_1: 1
A_2: 0
B_1: 0
B_2: 0



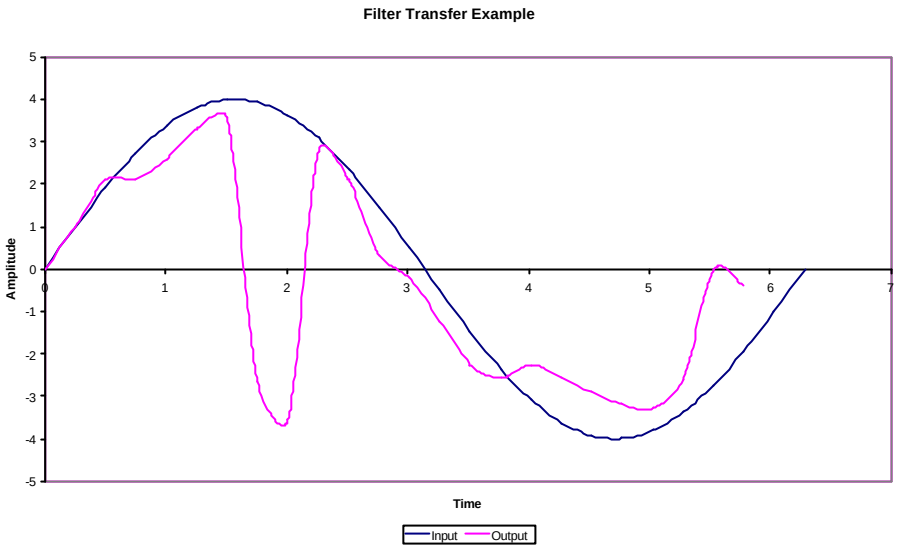
Finally as a delay by two samples:

A_0: 0
A_1: 0
A_2: 1
B_1: 0
B_2: 0



These tests showed that the different parts of the filter were behaving as expected. ROM values for coefficients that described a p/4 notch filter were then calculated using a spreadsheet:

A_0: 1
A_1: -0.1414
A_2: 10.01
B_1: 0.1273
B_2: 0.1539



Many hours were spent trying to trace the problem through the circuit, but no insight could be found. Eventually there was no time remaining to continue the analysis.

Timing analysis

This circuit can be clocked with a maximum frequency of 49.26MHz. This results in a clock period of 20.3ns and allows it to process 4.926Msamples / second. The setup/hold times for the inputs (*in{0..7}*) is about 3.0ns. However, the largest is *in2* with a setup/hold time of 4.6ns. Therefore, if the inputs are considered as a whole, the setup/hold time is 4.6ns.

As many of the speed optimisations as could be found were switched on. Speed was always prioritised over chip area because the circuit is very simple and could easily fit many times over onto a FLEX10K.

Personal Contribution

I was originally conducting this project as part of a pair. However, after the first lab session my partner decided that they did not want to pursue the course further. This results in almost the entire project being my own work, apart from the filter coefficients which were calculated by my partner.