Design of a Fractional Sample Rate Converter

Using Simulink, System Generator, and VHDL

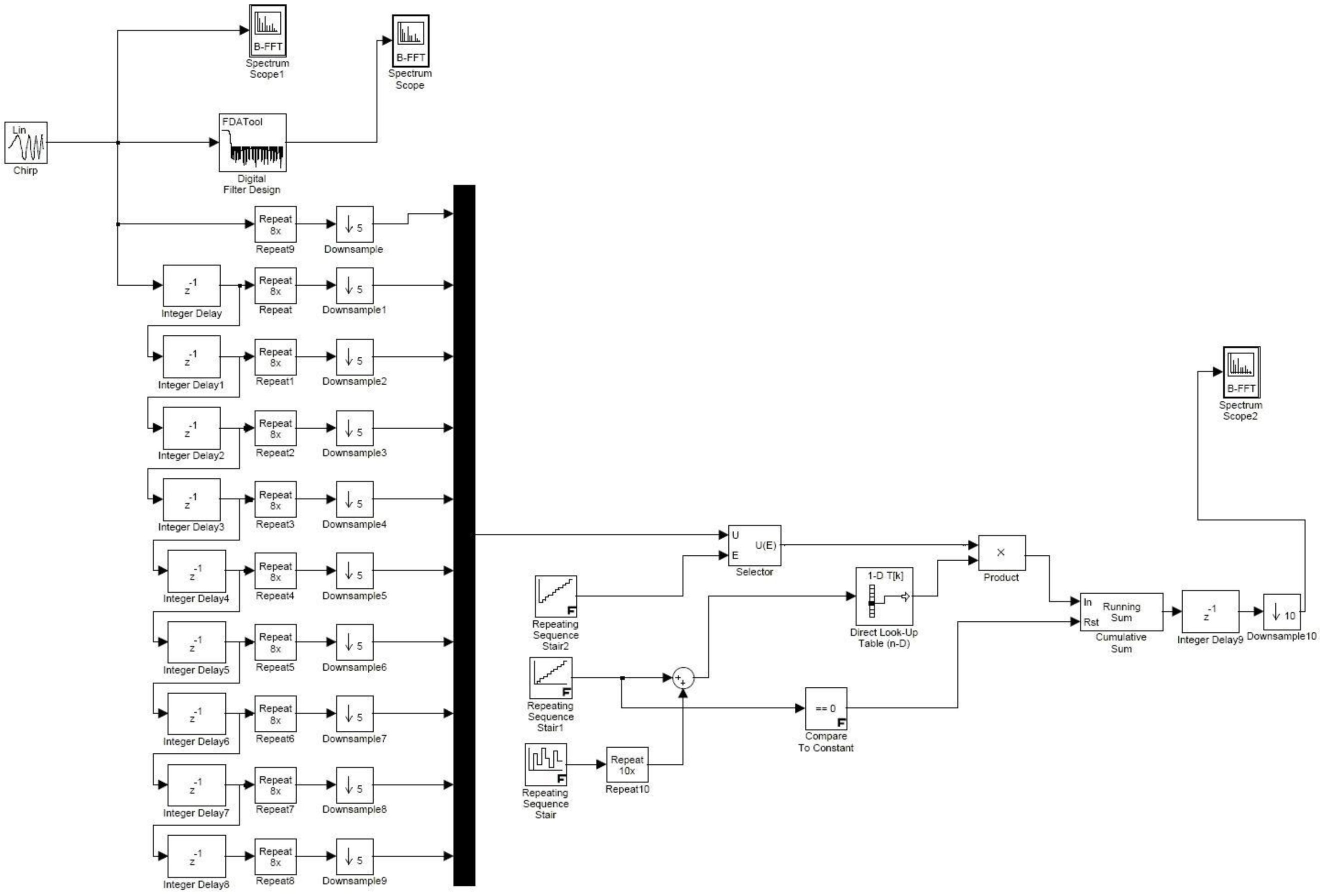


Figure 1: Simulink model of 8/5 fractional up-sampler. All required virtual sampling rates (clocks) are created by Simulink (courtesy of David A. Savory)

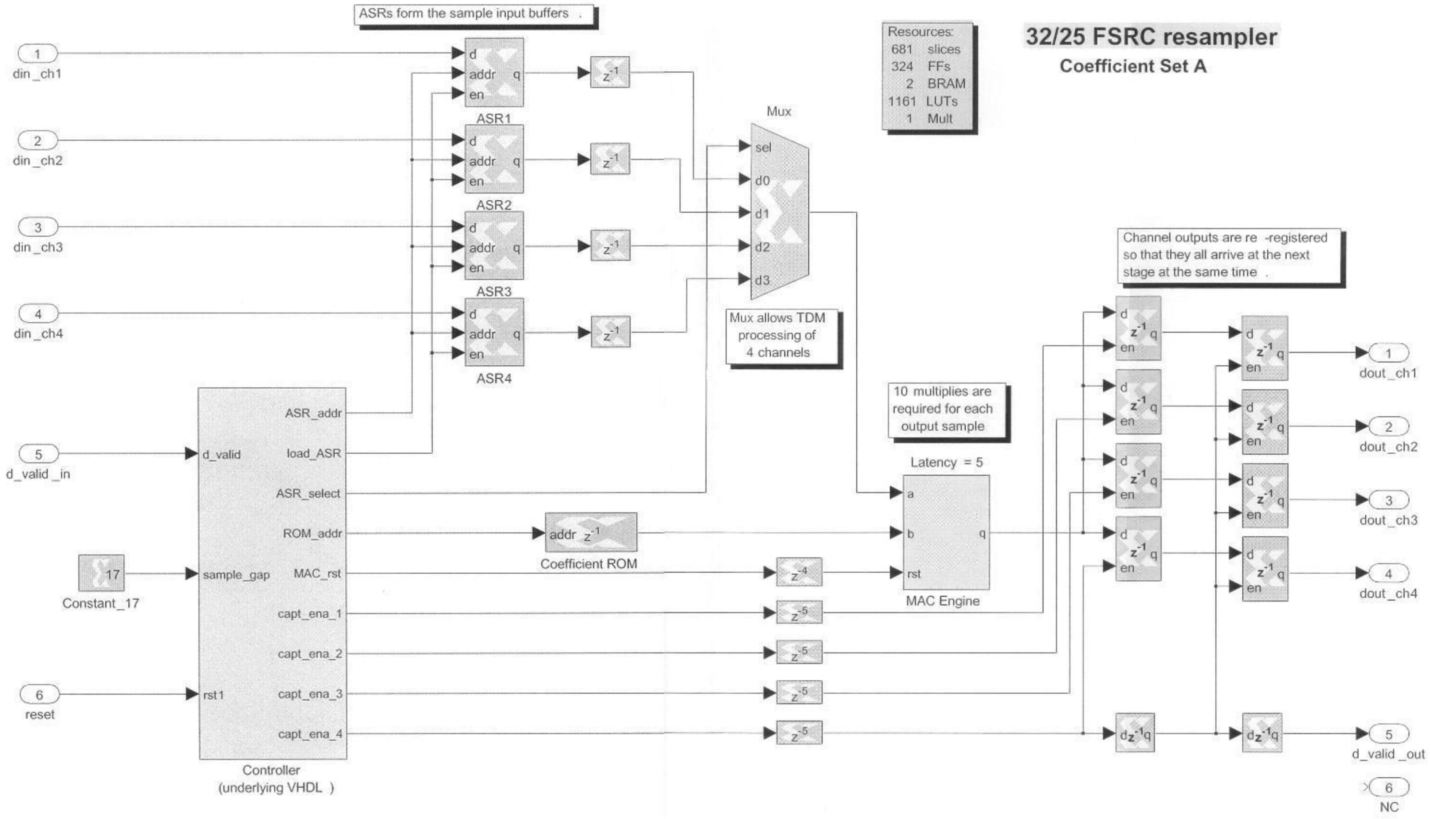
First Inversion was given the task of designing a Digital Down Converter (DDC) into an FPGA on an existing circuit board assembly. The input and output sampling rates needed to be the same as in the previous design, but the baseband frequency was different. Of course, this meant that all the intermediate filter frequencies were different as well. Ultimately, the only way to provide output samples at the correct rate was to design a fractional up-sampler. The fraction specified was 1024/625.

While working with *Xilinx's System Generator* it became apparent there was no provision for fractional re-sampling. Attempting to design the required frequency shift using standard System Generator up-sampling and down-sampling constructs, there simply weren't enough DCMs to handle all the intermediate clocks required. Besides this, combining filters that resampled at integral rates used far too many multiplier and block RAM resources to be practical.

A new approach was required. Part of the solution involved modeling the problem in *Mathwork's Simulink* using an approach documented in common texts on DSP techniques. A simplified model producing a re-sampling rate of 8/5 is shown in Figure 1. This approach has been called a time-variant coefficient filter, because, while there are a large number of coefficient taps distributed over the many phase filters, only a small subset of taps are brought into play for each output sample computation. The selected subset changes with each output sample. This keeps the number of MACs per second at a manageable level.

Still, the problem of dealing with a large number of intermediate frequencies and the lack of sufficient clock resources for the task remained, as might be inferred by a close inspection of Figure 1. The apparent problem doubles when it's kept in mind that two 32/25, or two 8/5 plus two 4/5, re-samplers needed cascading to achieve the required 1024/625 rate change.

The solution was to essentially ignore the idea of re-clocking the data at various pipeline stage outputs at a new clock rate. Instead, all computation was done at the highest FPGA-internal clock rate, and a Data Valid signal accompanied each new output sample as it flowed through the pipeline. Essentially, the computations were "batch processed," dispatching output samples as quickly as possible after each new input sample was received. By TDM multiplexing the MAC and coefficient look-up resources, multiple channels could be processed on a given filter. This used the precious multiply and memory resources at near maximum bandwidth. Using the Data Valid signal as a write enable, the output samples could be placed in small FIFO and then clocked out at the final, desired output sample rate.



32/25 FSRC resampler

Coefficient Set A

Resources:	
681	slices
324	FFs
2	BRAM
1161	LUTs
1	Mult

ASRs form the sample input buffers .

Mux allows TDM processing of 4 channels

10 multiplies are required for each output sample

Channel outputs are re -registered so that they all arrive at the next stage at the same time .

Figure 2: Synthesizable System Generator model of Fractional Sample Rate Converter

Most of this was accomplished with standard System Generator constructs, as shown in Figure 2, which shows a single stage of two cascaded 32/25 re-samplers. However, that tool best lends itself to changing the sampling rate by integral amounts as samples pass through its various processing stages. It is not as well suited to "batch process" samples at a fixed clock rate while sample rates change. Therefore, a controller block with underlying VHDL was created to control both the "time variant" coefficient selection and the sporadic output sample creation (sporadic relative to high speed master clock of the FPGA).

The cascaded re-sampling stages were identical except for parameterized details, like coefficient sets. The resulting design was successfully simulated and verified under both laboratory and field test conditions.