



# Fractional Resampling Cores for FPGA

## Product Information Sheet

RF Engines have developed a range of fractional resampling architectures for FPGA, that can be used to perform up-sampling or down-sampling of high-speed digital signals. The architectures are highly scalable, and support resampling of just one channel or thousands of channels simultaneously, whilst maintaining efficient use of silicon resources, and providing excellent filtering performance in order to mitigate the effects of aliasing.

The resampling rates can be selected with high precision, and allow selection of output sample rates with a resolution less than one Hertz. When operating on multiple input channels, the architectures treat each channel independently, allowing different input/output sampling rates, and rate changes for each channel. Furthermore, the rate change required for each channel can be updated at runtime, without affecting the operation of other channels.



The FPGA cores are provided as an EDIF netlist for either Xilinx or Altera FPGA devices, and are custom generated for each specific requirement to ensure a minimal FPGA footprint, and lowest power consumption. Like all RF Engines cores, the resamplers are supported by bit-true Matlab models, allowing early validation of the core through simulation, and thereby reducing risk.

The designs are licensed as Intellectual Property (IP) cores to equipment manufacturers and allow a low cost and reduced risk route to product.

## Features

- Fractional architecture allows precise sample rate changes to sub-Hertz resolution
- Architectures available for up and down conversion
- Simultaneous resampling of multiple channels (potentially thousands)
- Resampling rates reconfigurable at run-time
- Aggregate sample rates up to 180 MHz
- High performance filtering reduces aliasing effects
- Silicon usage minimised for each application
- Supported by bit-true Matlab models
- Xilinx and Altera FPGA devices supported

## Applications

- Telecommunications modulators and demodulators
- Software defined radio
- Image and video processing
- Audio processing
- Interfacing with fixed-rate hardware devices

# Resampling Theory

In some resampling systems the desired output sample rate is an integer multiple of the input rate, and the rate conversion function can be implemented as a simple decimating or interpolating filter. In cases where there is not such a simple relationship between the input and output sample rates a fractional resampler is required.

A fractional resampler is a function that changes the sample rate of the signal by an integer ratio of  $L/M$ . The generic block diagram for a fractional resampler is shown in the Figure 1 below. In its generic form, the system consist of an initial interpolator which increases the sample rate by a factor of  $L$  using interpolation, and a decimator which reduces the sample rate by a factor of  $M$ . In between these two blocks is a low-pass filter, which is required to remove the aliases that occur as part of the interpolation process.

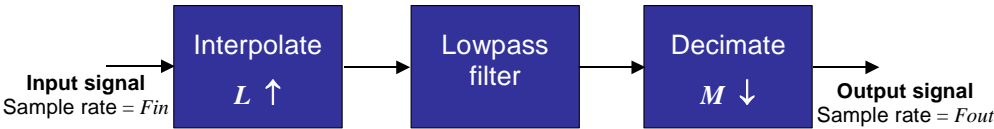


Figure 1 – Generic Fractional Resampler

Time-domain and frequency domain views of two simple examples are shown in Figure 2 and Figure 3. In both cases, the input signal is complex with a sample rate of  $f_{in}$ . The time-domain amplitude plots show the in-phase part of the signal at each stage of the re-sampling process and the frequency-domain plots show complex power spectrum at each stage. A frequency scale of Hz has been used to illustrate the process, for a signal with a bandwidth of  $\pm 1$  Hz.

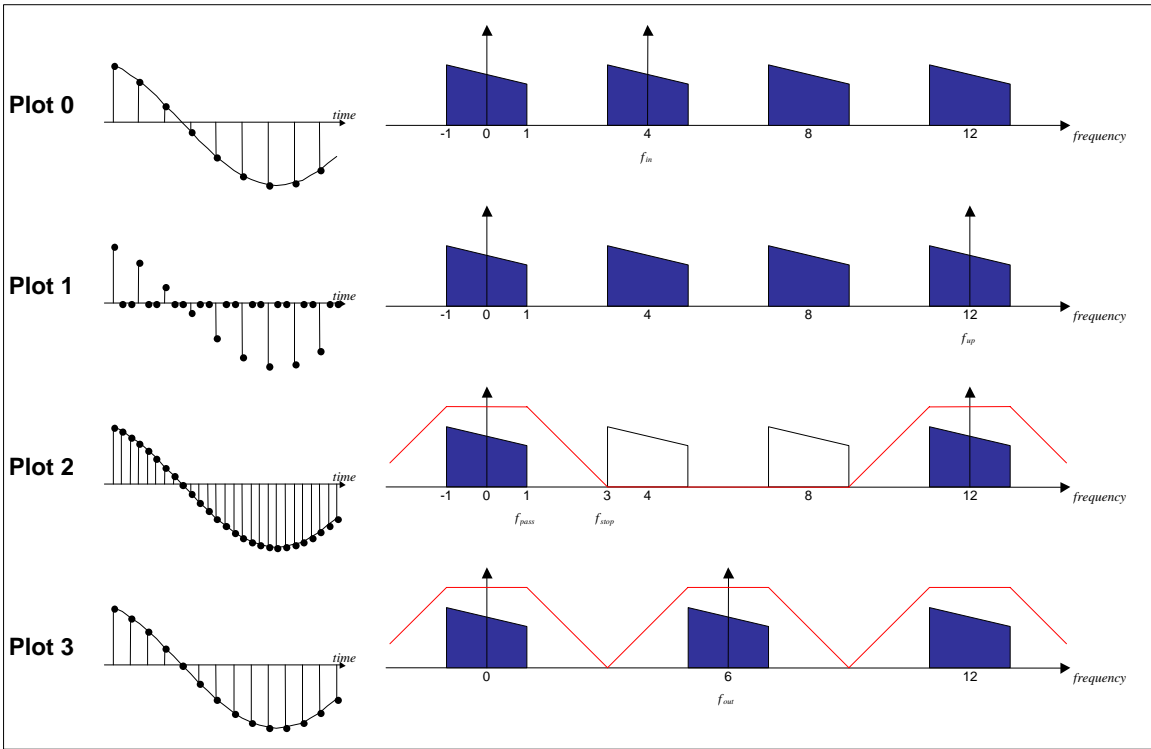


Figure 2: Resample by 3/2

### Resample by 3/2 (up-sample)

Plot 0 shows time and frequency-domain views of the input signal, sampled at a rate of  $f_{in} = 4$ Hz. A section of the infinite sequence of sampling images is shown across the frequency axis at intervals of  $f_{in}$ .

**Plot 1** shows the input signal after up-sampling by a factor of  $L = 3$ . This up-sampling process is achieved by inserting 2 zeros between each input sample as shown in the time-domain plot. The resulting spectrum is identical to that of **Plot 0**, the only difference is that the sampling frequency  $f_{up}$  has been increased by a factor of  $L = 3$ , from 4Hz to 12Hz.

**Plot 2** shows the up-sampled signal after a low-pass filter has been applied. The filter has a frequency response (shown in red) that ensures removal of the images that were present at 4Hz and 8Hz. The time-domain effect of filtering is illustrated by the resulting ‘smoothed’ waveform. The actual ‘smoothness’ depends on the quality of the filter in terms of passband ripple and stopband rejection.

**Plot 3** shows the final re-sampled signal after the up-sampled version has been decimated by  $M = 2$ . The frequency-domain view shows how the sampling image that was present at 12Hz in **Plot 2** has ‘moved’ with the decimated sample rate down to  $f_{out} = 6$ Hz. The low-pass filter that was applied to the up-sampled signal is shown to illustrate the alias-free passband. Note that the final decimation could have been chosen from one of two phases.

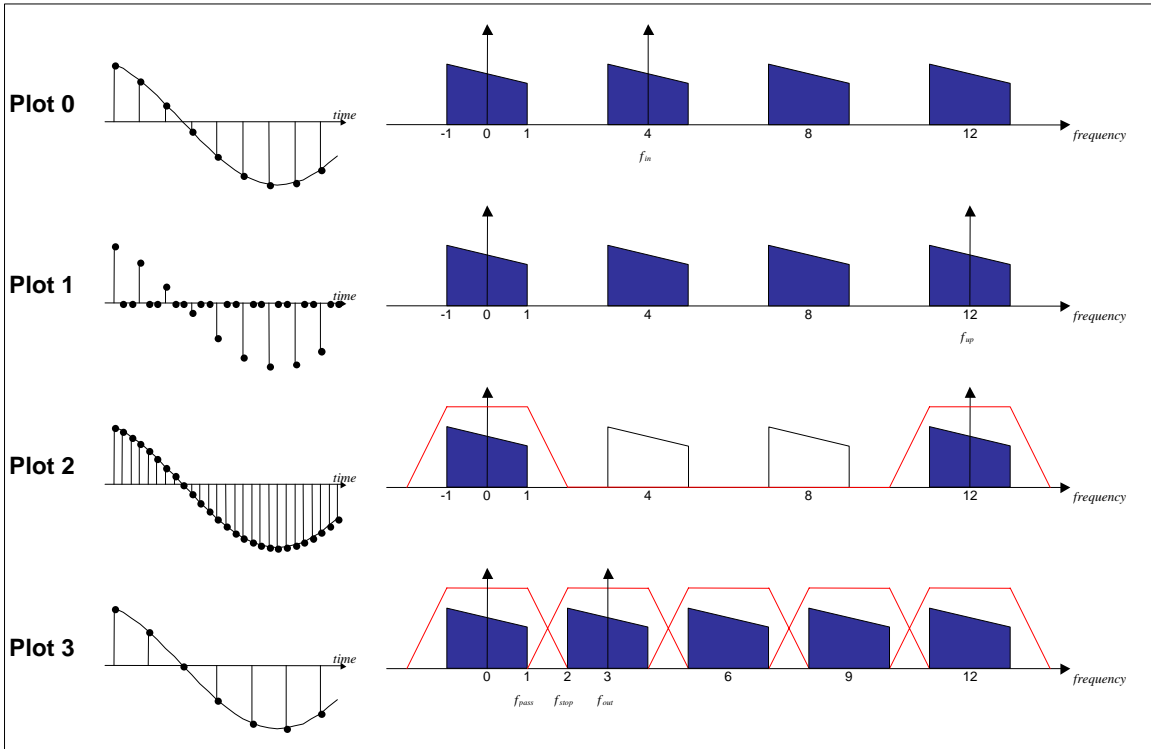


Figure 3: Resample by 3/4

**Resample by 3/4 (down-sample)**

**Plot 0** and **Plot 1** are identical to those shown in the ‘Resample by 3/4’ example above since the up-sample factor is the same.

**Plot 2** shows the up-sampled signal after a low-pass filter has been applied. The filter has a frequency response (shown in red) that ensures removal of the images that were present at 4Hz and 8Hz. The actual frequency response shown has a wider stopband region than is required for image removal alone. The reason for this is explained later. The time-domain effect of filtering is illustrated by the resulting ‘smoothed’ waveform. The actual ‘smoothness’ depends on the quality of the filter in terms of passband ripple and stopband rejection.

**Plot 3** shows the final re-sampled signal after the up-sampled version has been decimated by 4. The frequency-domain view shows how the sampling image that was present at 12Hz in **Plot 2** has ‘moved’ with the decimated sample rate down to  $f_{out} = 3$ Hz. The low-pass filter that was applied to the up-sampled signal is shown to illustrate why the stopband region was required to be larger than for image removal alone. It can be seen that the low-pass filter must roll-off before the passband edge of the first sampling image to ensure an alias-free passband.

## Down-Sampler Cores

The block diagram of the down-sampler architecture is shown in Figure 4. The system includes an optional channel interleaver, the down-sampler itself, and an optional channel deinterleaver. The core is driven by the incoming samples, such that data is accepted into the core and processed whenever samples arrive.

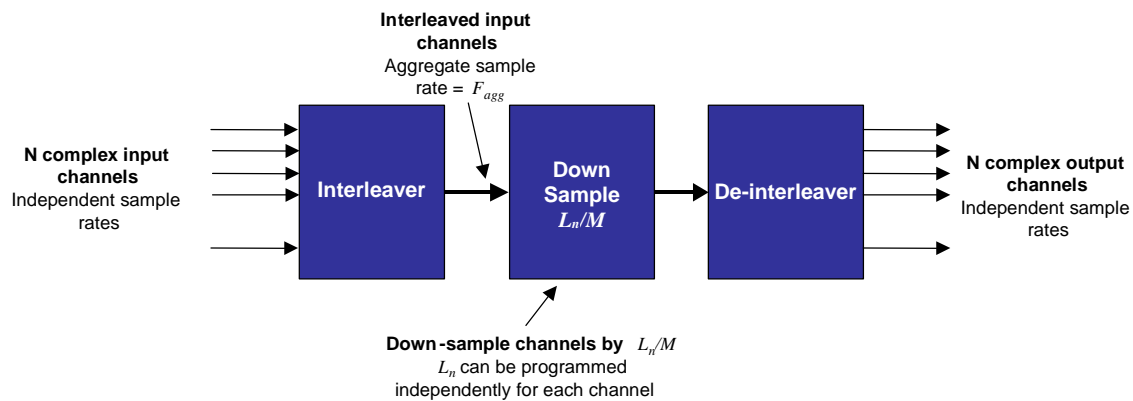


Figure 4 - Down-sampler block diagram

### Down-sampler Input Format

The down-sampler core accepts a single stream of complex samples, containing a number of time-interleaved channels, with each channel identified by a channel address that must be presented to the core in parallel with the sample itself. Samples are in two's complement integer format. The input sequence and sample rates of the individual channels can be unrelated.

The maximum number of channels accepted by the core,  $N$ , is fixed at the time the core is generated. The core is also tailored to the maximum sample rate of each channel and the maximum aggregate sample rate (the combined sample rate of the interleaved channels).

### Rate Change Control

The down sampling rate is determined independently for each channel according to a fraction,  $L_n/M$ , where  $n$  is the channel number.

$M$  is the control modulus of the core, and is also fixed at the time the core is generated. Typically this value is set to  $2^{36}$ , which results in a rate change frequency resolution of  $<1\text{mHz}$ .

$L_n$  is an integer value programmed at run time for each of  $N$  channels, and determines the actual rate change. The minimum allowable value for  $L_n$  is set at the time the core is generated, and this determines the maximum amount of down-sampling that the core can perform. Whilst a minimum value of  $L_n = 1$  can be supported, higher values result in better silicon efficiency.

### Resampling Phase Control

The down-sampler also supports optional *resampling phase control*. This allows the resampling phase to be programmed in the same way as the rate change for each channel. When  $L = M$ , the rate change amount is unity, and the down-sampling core can be used as a simple phase changer.

**Core Parameters**

The table below summaries the parameters that, are required in order to generate a down-sampler core. RF Engines are pleased to suggest suitable parameters, and bit-true models can be provided in order to confirm that selected parameters are suitable for a particular application.

Parameter	Description	Notes
$N$	Maximum number of channels that can be simultaneously down-sampled by the core.	Can be several thousand depending on FPGA resources available.
$M$	Control modulus. Denominator of resampling fraction.	Typically $2^{36}$
$L_{min}$	Minimum allowed numerator value of the resampling fraction.	Can be 1, but higher values result in more efficient silicon usage
$F_{inmax}$	Maximum input channel sample rate	
$F_{aggmax}$	Maximum aggregate sample rate. Sum of all channel input rates.	Typically limited to the FPGA clock rate.
Input bit width	Bit width of I and Q integer samples	
Resampling phase control	Optional function to allow control of the resampling phase	
Filter performance	Resampling filter characteristics: Alias-free passband Stopband attenuation (max-alias level) Passband ripple of resampling filter	Typical performance: Passband: 80% Nyquist Stop band attn: 75 to 90 dB Pass band ripple: 0.1dB pk-pk

**Table 1 - Down-sampler parameters required to generate core**

**Example Down-sampler Cores**

Parameter	Description	Value
$N$	Maximum number of channels	512
$M$	Control modulus	$2^{36}$
$L_{min}$	Minimum allowed numerator value of the resampling fraction	$2^{36}/100$
$F_{inmax}$	Maximum input channel sample rate	25 MS/s (complex)
$F_{aggmax}$	Maximum aggregate input sample rate	180 MS/s (complex)
Input bit width	Bit width of I and Q integer samples	18-bit (2's complement)
Resampling phase control	Optional function to allow control of the resampling phase	Not supported
Filter performance	Alias-free passband	80% Nyquist
	Stopband attenuation (max-alias level)	> 75 dB
	Passband ripple of resampling filter	< 0.1dB pk-pk
Rate change frequency resolution	Frequency resolution with which output rates may be controlled	$F_{in} / 2^{36}$
Maximum decimation	Largest down-sampling rate supported	100
Output bit width	Bit width of I and Q integer samples	24-bit (2's complement)
FPGA target		Xilinx Virtex II Pro 50 XC2VP50-5FF1152
FPGA resources	Placed Virtex II Pro Logic Slices	2925 (12%)
	Placed Virtex II Pro Block Multipliers	18 (7%)
	Placed Virtex II Pro Block RAMs	38 (16%)

**Example 1 – 512-channel down-sampler, maximum decimation rate = 100**

Parameter	Description	Value
$N$	Maximum number of channels	512
$M$	Control modulus	$2^{36}$
$L_{min}$	Minimum allowed numerator value of the resampling fraction	1
$F_{inmax}$	Maximum input channel sample rate	25 MS/s (complex)
$F_{aggmax}$	Maximum aggregate input sample rate	180 MS/s (complex)
Input bit width	Bit width of I and Q integer samples	18-bit (2's complement)
Resampling phase control	Optional function to allow control of the resampling phase	Not supported
Filter performance	Alias-free passband Stopband attenuation (max-alias level) Passband ripple of resampling filter	80% Nyquist > 75 dB < 0.1dB pk-pk
Rate change frequency resolution	Frequency resolution with which output rates may be controlled	$F_{in} / 2^{36}$
Maximum decimation	Largest down-sampling rate supported	$2^{36}$
Output bit width	Bit width of I and Q integer samples	24-bit (2's complement)
FPGA target		Xilinx Virtex II Pro 50 XC2VP50-5FF1152
FPGA resources	Placed Virtex II Pro Logic Slices Placed Virtex II Pro Block Multipliers Placed Virtex II Pro Block RAMs	3552 (15%) 18 (7%) 56 (24%)

**Example 2 – 512-channel down-sampler, maximum decimation rate =  $2^{36}$**

## Up-Sampler Cores

The block diagram of the up-sampler architecture is shown in Figure 5. The system includes an optional channel interleaver, the up-sampler itself, and an optional channel deinterleaver. The up-sampler is a 'master' core that defines when a new input sample can be written into the core.

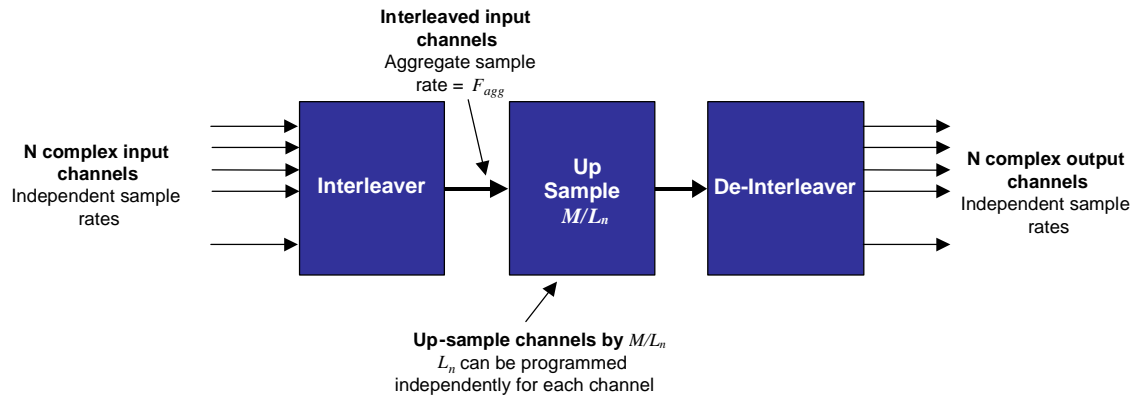


Figure 5 - Up-sampler block diagram

### Up-sampler Input Format

The up-sampler core accepts a single stream of complex samples, containing a number of time-interleaved channels, with each channel identified by a channel address that must be presented to the core in parallel with the sample itself. Samples are in two's complement integer format. The input sequence and sample rates of the individual channels can be unrelated.

The maximum number of channels accepted by the core,  $N$ , is fixed at the time the core is generated. The core is also tailored to the maximum sample rate of each channel and the maximum aggregate sample rate (the combined sample rate of the interleaved channels).

### Rate Change Control

The up sampling rate is determined independently for each channel according to a fraction,  $M/L_n$ , where  $n$  is the channel number.

$M$  is the control modulus of the core, and is also fixed at the time the core is generated. Typically this value is set to  $2^{36}$ , which results in a rate change frequency resolution of <1mHz signals.

$L_n$  is an integer value programmed at run time for each of  $N$  channels, and determines the actual rate change.  $L_n$  may take any value from 1 to  $M$ .

### Resampling Phase Control

The up-sampler also supports optional *resampling phase control*. This allows the resampling phase to be programmed in the same way as the rate change for each channel. When  $L = M$ , the rate change amount is unity, and the up-sampling core can be used as a simple phase changer.

**Core Parameters**

The table below summaries the parameters that, are required in order to generate a up-sampler core. RF Engines are pleased to suggest suitable parameters, and bit-true models can be provided in order to confirm that selected parameters are suitable for a particular application.

Parameter	Description	Notes
$N$	Maximum number of channels that can be simultaneously up-sampled by the core.	Can be several thousand depending on FPGA resources available.
$M$	Control modulus. Numerator of resampling fraction.	Typically $2^{36}$
$F_{outmax}$	Maximum output channel sample rate	
$F_{aggmax}$	Maximum aggregate output sample rate. Sum of all channel input rates.	Typically limited to the FPGA clock rate.
Input bit width	Bit width of I and Q integer samples	
Resampling phase control	Optional function to allow control of the resampling phase	
Filter performance	Resampling filter characteristics: Alias-free passband Stopband attenuation (max-alias level) Passband ripple of resampling filter	Typical performance: Passband: 80% Nyquist Stop band attn: 75 to 90 dB Pass band ripple: 0.1dB pk-pk

Table 2 – Up-sampler parameters required to generate core

Parameter	Description	Value
$N$	Maximum number of channels	512
$M$	Control modulus	$2^{36}$
$F_{outmax}$	Maximum output channel sample rate	25 MS/s (complex)
$F_{aggmax}$	Maximum aggregate output sample rate	180 MS/s (complex)
Input bit width	Bit width of I and Q integer samples	18-bit (2's complement)
Resampling phase control	Optional function to allow control of the resampling phase	Not supported
Filter performance	Alias-free passband Stopband attenuation (max-alias level) Passband ripple of resampling filter	80% Nyquist > 75 dB < 0.1dB pk-pk
Rate change frequency resolution	Frequency resolution with which output rates may be controlled	$F_{out} / 2^{36}$
Maximum decimation	Largest up-sampling rate supported	$2^{36}$
Output bit width	Bit width of I and Q integer samples	24-bit (2's complement)
FPGA target		Xilinx Virtex II Pro 50 XC2VP50-5FF1152
FPGA resources	Placed Virtex II Pro Logic Slices Placed Virtex II Pro Block Multipliers Placed Virtex II Pro Block RAMs	2844 (12%) 18 (7%) 29 (13%)

**Example 3 – 512-channel up-sampler, maximum interpolation rate =  $2^{36}$**



rfengines limited

Innovation Centre  
St Cross Business Park  
Newport, Isle of Wight  
PO30 5WB  
United Kingdom

Tel: +44 1983 550330  
Fax: +44 1983 550340  
Email: [info@rfel.com](mailto:info@rfel.com)  
Website: [www.rfel.com](http://www.rfel.com)