Publication No. REF-004-004-R00

Digital Signal Processing
Function Library
Reference Manual

Red Rapids

797 North Grove Rd, Suite 101
Richardson, TX 75081
Phone: (972) 671-9570
www.redrapids.com

Table of Contents

List of Figures

List of Tables

1.0 Introduction

Red Rapids offers a variety of products that include embedded digital signal processing (DSP) functions such as tuners and filters. Often the same function is used by multiple product families. This document serves as the single point of reference for DSP functions deployed across the entire Red Rapids product line.

The latest product documentation and software is available for download from the Red Rapids web site (www.redrapids.com).

1.1 Conventions

This manual uses the following conventions:

- Hexadecimal numbers are prefixed by “0x” (e.g. 0x00058C).
- *Italic* font is used for names of registers.
- Blue font is used for names of directories, files and OS commands.
- Green font is used to designate source code.
- Active low signals are followed by ‘#’, For example, TRST#.

☞	Text in this format highlights useful or important information.

!	Text shown in this format is a warning. It describes a situation that could potentially damage your equipment. Please read each warning carefully.

The following are some of the acronyms used in this manual.

- **Fs** Sample Clock Frequency
- **HBF** Halfband Filter
- **LPF** Lowpass Filter
- **FIR** Finite Impulse Response
- **NCO** Numerically Controlled Oscillator

1.2 Revision History

Version	Date	Description
R00	1/10/2017	Initial release.

2.0 Filter Functions

The following subsections provide a detailed description of every filter type embedded in a Red Rapids product. Each subsection begins with a detailed description of the filter architecture and includes the power-on filter characteristics. Other filter configurations may be included for reference.

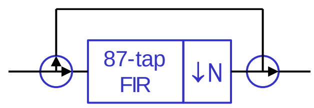
Configuration files for each filter type are supplied in a separate download so that they can be easily imported into analysis tools or application software. Each file contains configuration settings separated by the keywords listed in Table 2-1. The keywords can appear in any order throughout the file, extraneous comments are simply ignored.

Table 2-1 Filter Configuration Keywords

Keyword	Description
Coefficients	This keyword marks the beginning of a list of coefficients that define the characteristics of the filter. There must be one coefficient listed for each available tap, even if the filter is symmetric.
Log2Gain	The value immediately following this keyword conveys the gain through the filter expressed as a binary logarithm (log base2). The filter uses this gain along with the coefficient size to select the most significant bits available from each calculation.
Resample	The value immediately following this keyword defines the ratio of input to output samples processed through the filter. This can be either a downsample (decimation) ratio or upsample (interpolation) ratio depending on the filter.

2.1 Type 0 (87-tap Programmable FIR)

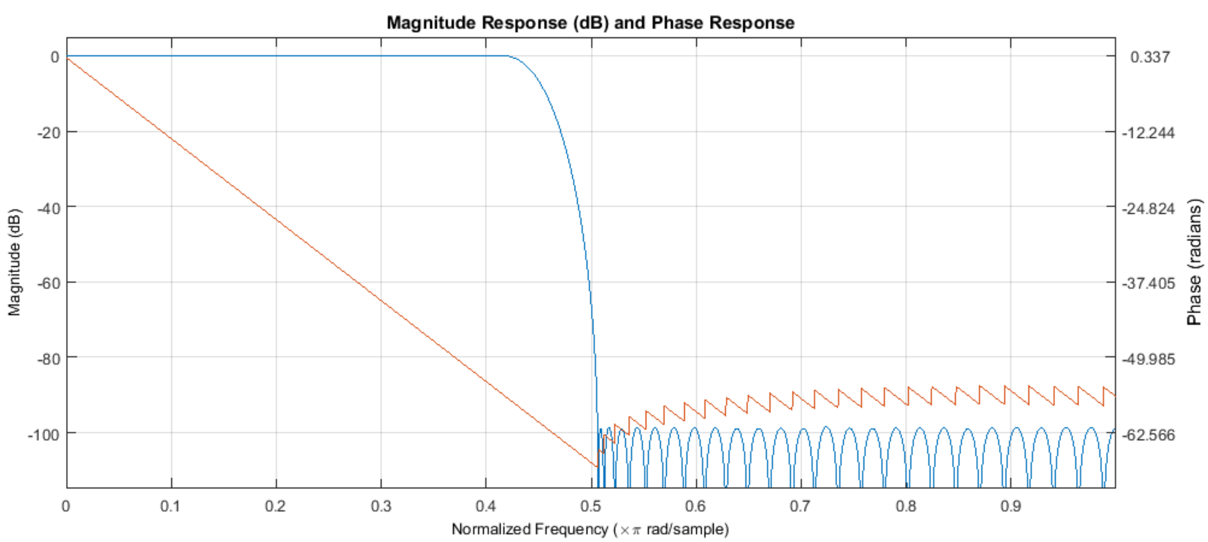
Architecture Features



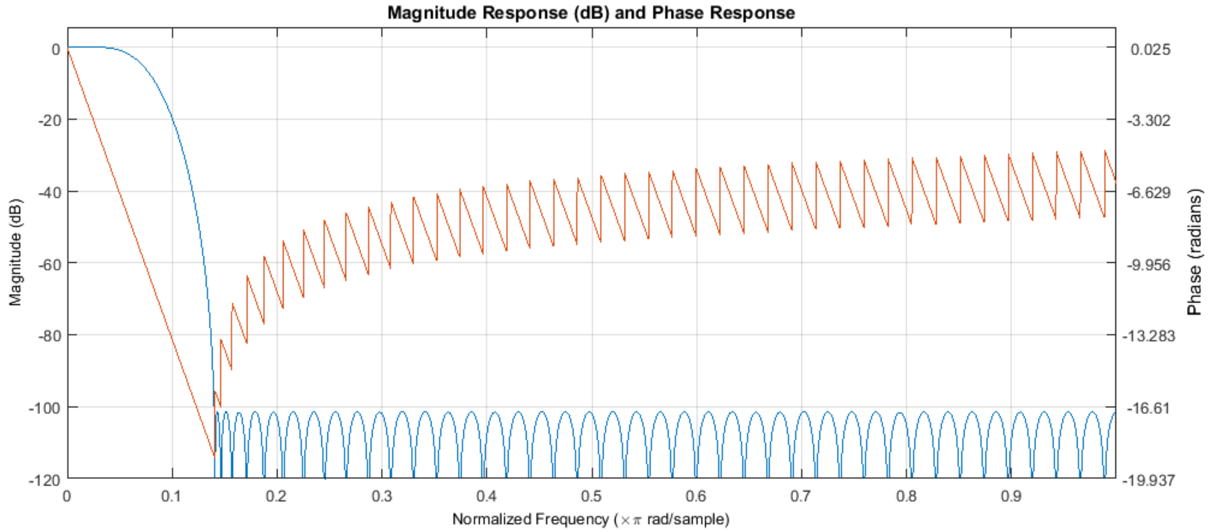
Description	Finite impulse response filter with integer downsampler.		
Number of Taps	87	**Symmetric Taps**	☒ Yes ☐ No
Input Data Size	17 bits	**Output Data Size**	17 bits
Coefficient Size	24 bits	**Reloadable Coefficients**	☒ Yes ☐ No
Max Input Rate	Fs	**Downsample (N) Range**	0 (bypass) to 2^{16}-1

0.42*Fs/2 Lowpass Filter Characteristics (Power-on Default)

Coefficient File	Type0_42LPF2.txt	**Matlab Design Code**
Downsample Ratio	2	NumTaps = 87;
Log2Gain	24	F = [0 0.420 0.506 1];
Passband Edge	0.420	A = [1 1 0 0];
Stopband Edge	0.506	W = [1 950];
Passband Ripple	+/- 0.1 dB	firpm(NumTaps-1,F,A,W);
Stopband Attenuation	-98 dB	

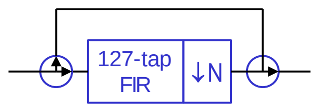


0.02*Fs/2 Lowpass Filter Characteristics		
Coefficient File	Type0_02LPF10.txt	**Matlab Design Code**
Downsample Ratio	10	NumTaps = 87; F = [0 0.020 0.140 1]; A = [1 1 0 0]; W = [1 4]; firpm(NumTaps-1,F,A,W);
Log2Gain	26	
Passband Edge	0.020	
Stopband Edge	0.140	
Passband Ripple	+/- 0.03 dB	
Stopband Attenuation	-101 dB	



2.2 Type 1 (127-tap Programmable FIR)

Architecture Features



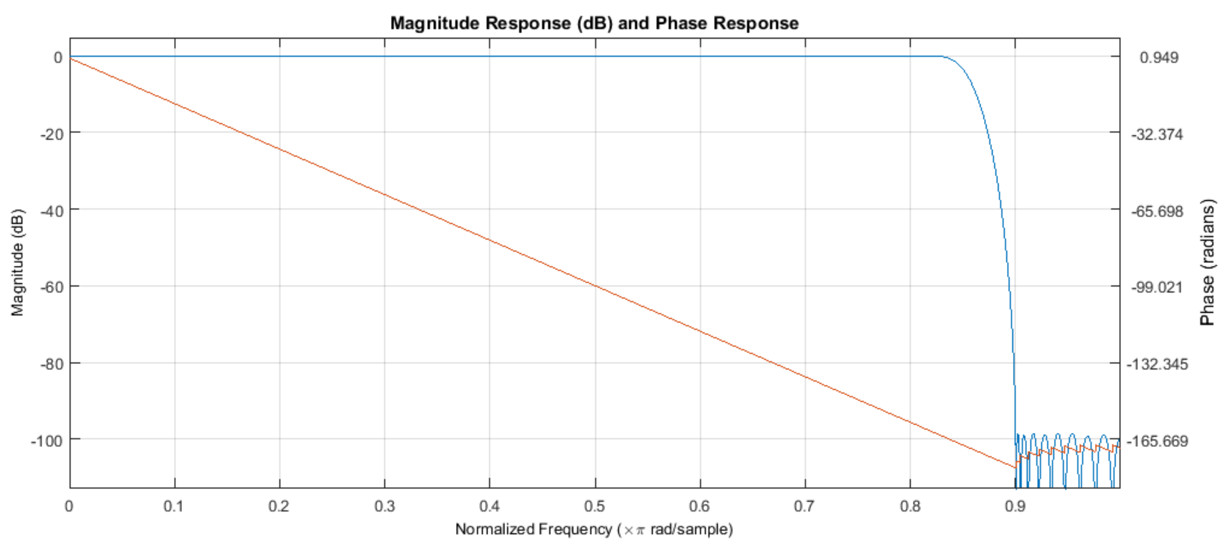
Description	Finite impulse response filter with integer downsampler.		
Number of Taps	127	**Symmetric Taps**	☒ Yes ☐ No
Input Data Size	17 bits	**Output Data Size**	17 bits
Coefficient Size	24 bits	**Reloadable Coefficients**	☒ Yes ☐ No
Max Input Rate	Fs/2	**Downsample (N) Range**	0 (bypass) to 2^{16}-1

0.82*Fs/2 Lowpass Filter Characteristics (Power-on Default)

Coefficient File	Type1_82LPF1.txt
Downsample Ratio	1
Log2Gain	23
Passband Edge	0.820
Stopband Edge	0.900
Passband Ripple	+/- 0.015 dB
Stopband Attenuation	-98 dB

Matlab Design Code

```
NumTaps = 127;
F = [ 0  0.820  0.900  1 ];
A = [ 1  1  0  0 ];
W = [ 1  15 ];
firpm(NumTaps-1,F,A,W);
```

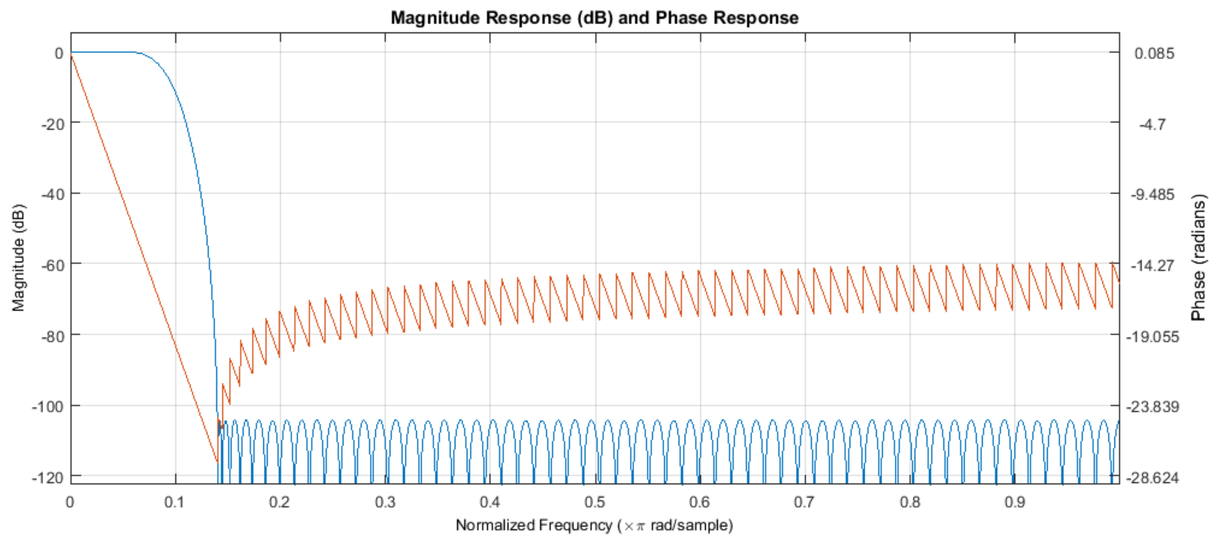


0.04*Fs/2 Lowpass Filter Characteristics

Coefficient File	Type1_04LPF10.txt
Downsample Ratio	10
Log2Gain	26
Passband Edge	0.040
Stopband Edge	0.140
Passband Ripple	+/- 0.00005 dB
Stopband Attenuation	-104 dB

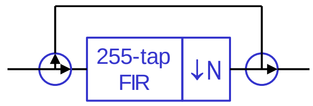
Matlab Design Code

```
NumTaps = 127;
F = [ 0  0.040  0.140  1 ];
A = [ 1  1  0  0 ];
W = [ 1  1 ];
firpm(NumTaps-1,F,A,W);
```



2.3 Type 2 (255-tap Programmable FIR)

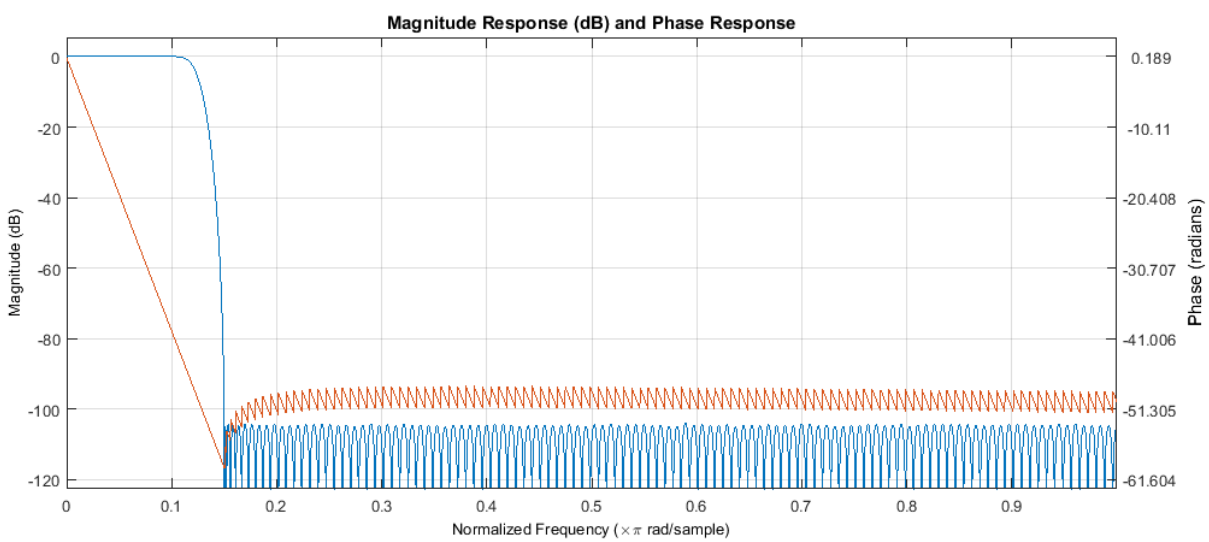
Architecture Features



Description	Finite impulse response filter with integer downsampler.		
Number of Taps	255	**Symmetric Taps**	☒ Yes ☐ No
Input Data Size	17 bits	**Output Data Size**	17 bits
Coefficient Size	23 bits	**Reloadable Coefficients**	☒ Yes ☐ No
Max Input Rate	Fs/4	**Downsample (N) Range**	0 (bypass) to 2^{16}-1

0.10*Fs/2 Lowpass Filter Characteristics (Power-on Default)

		Matlab Design Code
Coefficient File	Type2_10LPF8.txt	
Downsample Ratio	8	NumTaps = 255;
Log2Gain	25	F = [0 0.100 0.150 1];
Passband Edge	0.100	A = [1 1 0 0];
Stopband Edge	0.150	W = [1 1];
Passband Ripple	+/- 0.00005 dB	firpm(NumTaps-1,F,A,W);
Stopband Attenuation	-104 dB	



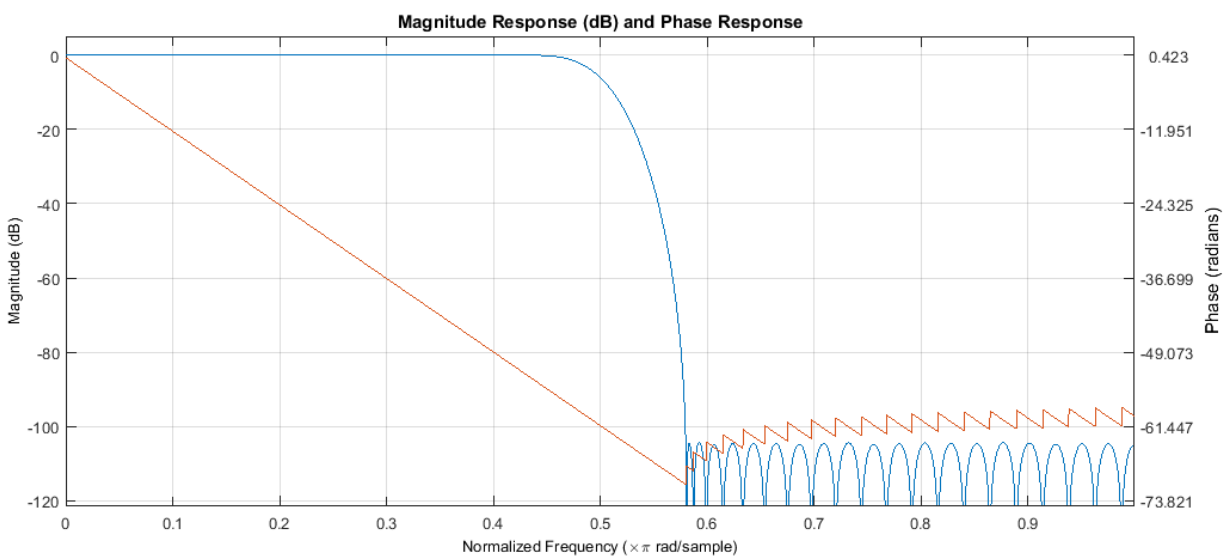
2.4 Type 3 (79-tap Halfband FIR)

Architecture Features

Description	Finite impulse response filter with integer downsampler.		
Number of Taps	79	**Symmetric Taps**	☒ Yes ☐ No
Input Data Size	17 bits	**Output Data Size**	17 bits
Coefficient Size	24 bits	**Reloadable Coefficients**	☐ Yes ☒ No
Max Input Rate	Fs	**Downsample Ratio**	2

0.42*Fs/2 Lowpass Filter Characteristics (Power-on Default)

Coefficient File	Type3_42HBF2.txt	**Matlab Design Code**	
Log2Gain	24	firhalfband('minorder',0.42,0.000009);	
Passband Edge	0.420		
Stopband Edge	0.580		
Passband Ripple	+/- 0.00005 dB		
Stopband Attenuation	-104 dB		



3.0 Tuner Functions

The following subsections provide a detailed description of every tuner type embedded in a Red Rapids product. The basic function of the tuner is frequency translation, but some architectures also perform the conversion between real and complex data representations.

Virtually every digital tuner relies on a numerically controlled oscillator (NCO) to generate sine and cosine waveforms set to the desired translation frequency. These waveforms are digitally mixed (multiplied) with the signal to be translated. A simple binary counter, or phase accumulator, forms the foundation of the NCO. Each count value can be equated to a point on the unit circle as shown in Figure 3-1.

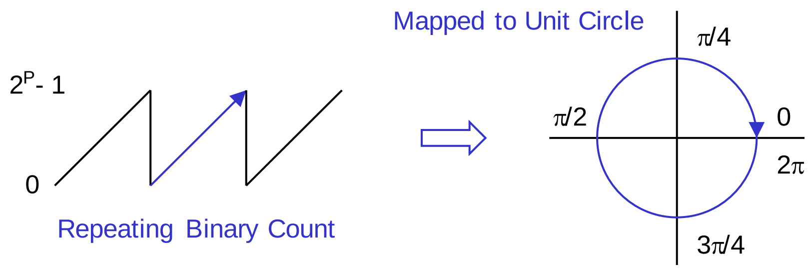


Figure 3-1 Phase Accumulator Theory of Operation

The frequency of the NCO is determined by how fast the hardware cycles through a full count, or one complete revolution of the unit circle. In practice, the phase accumulator is counting at a rate equivalent to the sample frequency (Fs) of the incoming signal. Counting by one produces a waveform of frequency Fs / 2^P, which becomes the tuning resolution since it is the lowest frequency achievable. Counting by a phase increment (Φ) greater than one can produce higher frequencies. This is easily accomplished in hardware using an accumulator with the input set to the phase increment as shown in Figure 3-2. The resulting output frequency is Φ * Fs / 2^P.

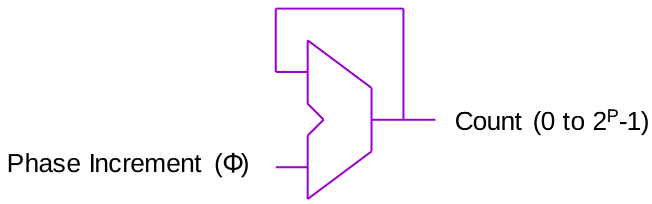
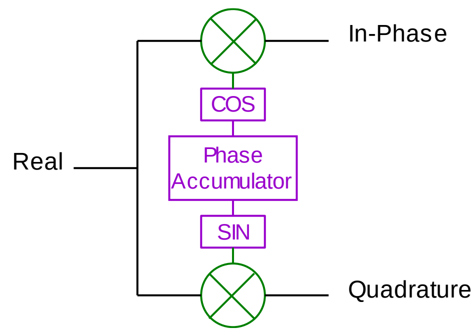


Figure 3-2 Phase Accumulator Hardware

Mapping the binary count to an actual sine and cosine equivalent value can consume considerable hardware resources. One common approach is a simple look-up table where precomputed values are stored in a memory addressable by the counter. Each value is accessible within a single clock cycle. Another approach is to compute the values algorithmically, which typically requires multiple cycles to complete.

3.1 Type 0 (Quadrature Mixer)

Architecture Features



Description	Quadrature mixer using CORDIC generated sine/cosine values.		
Phase Accum Size	23 bits	**Tuning Resolution**	Fs / 2^{23}
Input Data Size	16 bits	**Output Data Size**	17 bits
Sine/Cosine Size	24 bits	**Max Input Rate**	Fs