

# Equalizer Example

Karl W. Pfalzer (kpfalzer@sbcglobal.net)

A 10-band equalizer system example is given in ROOT/filter/examples/equalizer/src/equalizer.cxx.

A schematic of the system is given below:

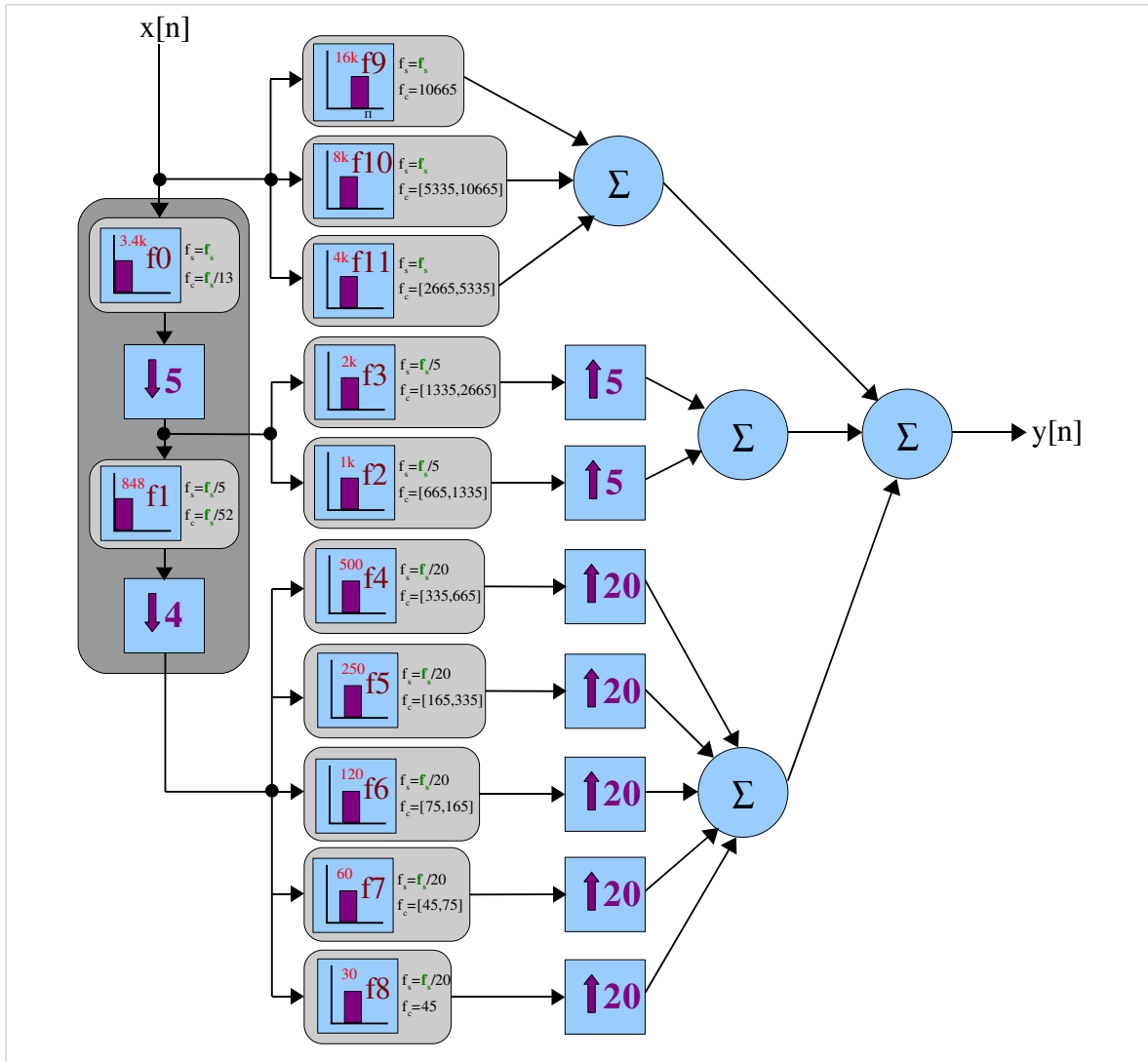


Figure 1: 10-band equalizer system

The following *components* are used in the equalizer system:

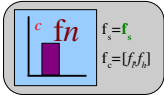



	<p>FIR filter. <math>f_s</math>=filter sampling frequency (as a function of primary sampling frequency, <math>f_s</math>). <math>f_c</math> = filter cutoff bounds: <math>f_l</math> is low cutoff and <math>f_h</math> is high cutoff. <math>c</math> = center frequency of filter.  <math>fn</math> = filter id <math>fn[n]</math> in equalizer.cxx code.</p>	 <p>Decimation by integer factor <math>n</math>.</p>
	<p>Interpolation by integer factor <math>n</math>.</p>	 <p>Summation of inputs.</p>

Table 1: Symbols used in 10-band equalizer system

The filter package (i.e., namespace `filter`) provides a class for each of these components:

- `TFilter`
- `TInterpolate`
- `TDecimate`
- `TSummer`

All of these classes are derived from `TBlock` which is the base class for all *building blocks* or components. The `>>` operator is overloaded similar to

```
TBlock* operator >> (TBlock *source, TBlock *rcvr)
```

allowing a series of `TBlocks` to be connected modeling a left-to-right dataflow:

```
source >> block1 >> block2 >> block3 ...
```

(where all of the operands of `>>` are type `TBlock*`).

The Figure 1: 10-band equalizer system contains two *decimators* (shown on right): a factor 5 followed by a factor 4. These decimators are created in `equalizer.cxx` as:

```
fn[0] = new TFilter(new TLpf(...));
dn[0] = new TDecimate(5);
fn[1] = new TFilter(new TLpf(...));
dn[1] = new TDecimate(4);
```

and connected using the `>>` operator:

```
fn[0] >> dn[0] >> fn[1] >> dn[1];
```

