Effect

base class

Effect

```
processFrame(float& input, float& output);
setDryWet(float dryWet);
setBypass(bool bypass);

…
```

Effect

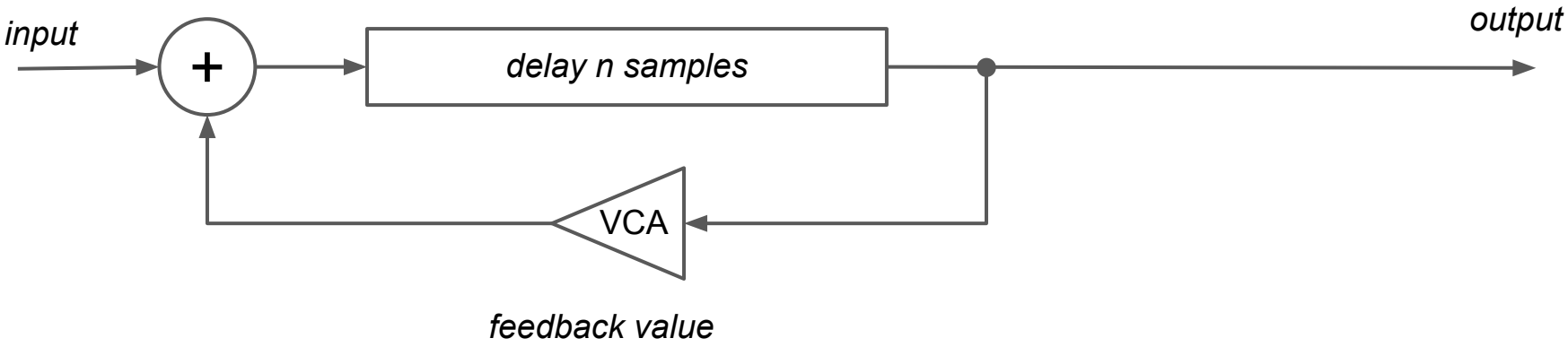
```
processFrame(float& input, float& output);
setDryWet(float dryWet);
setBypass(bool bypass);

…
```

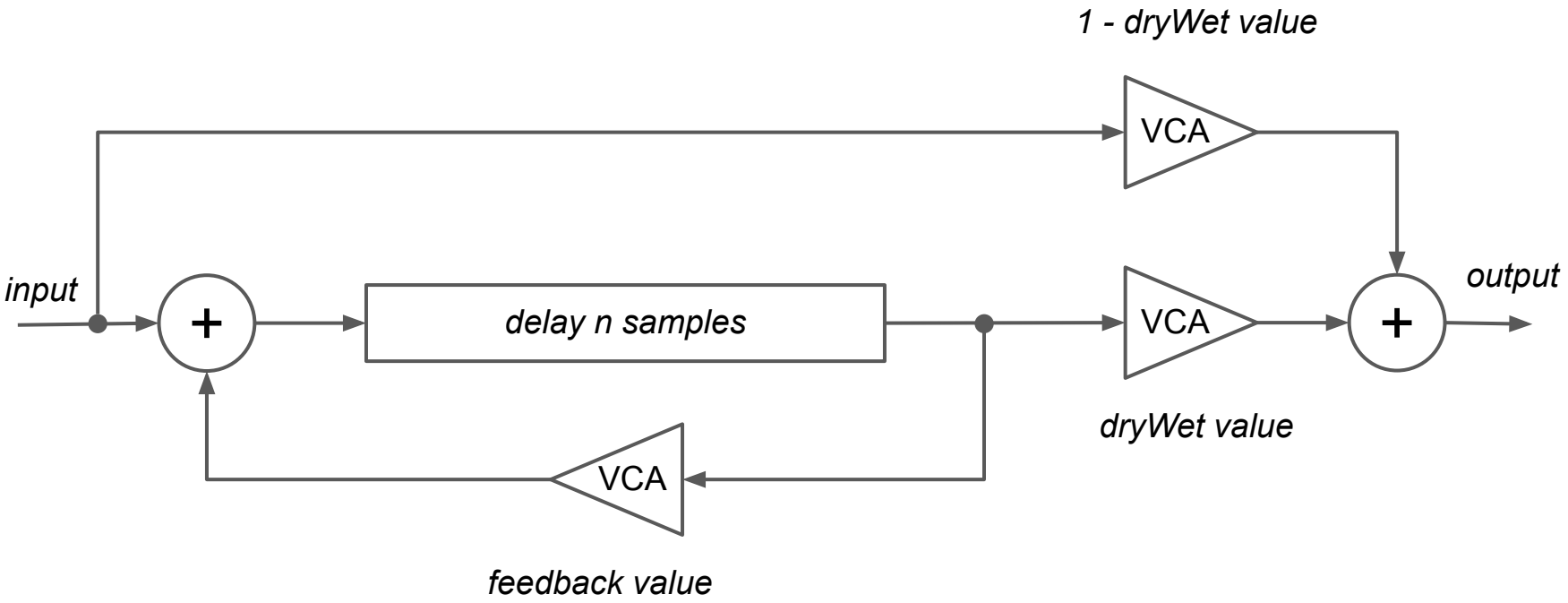
ASubclass

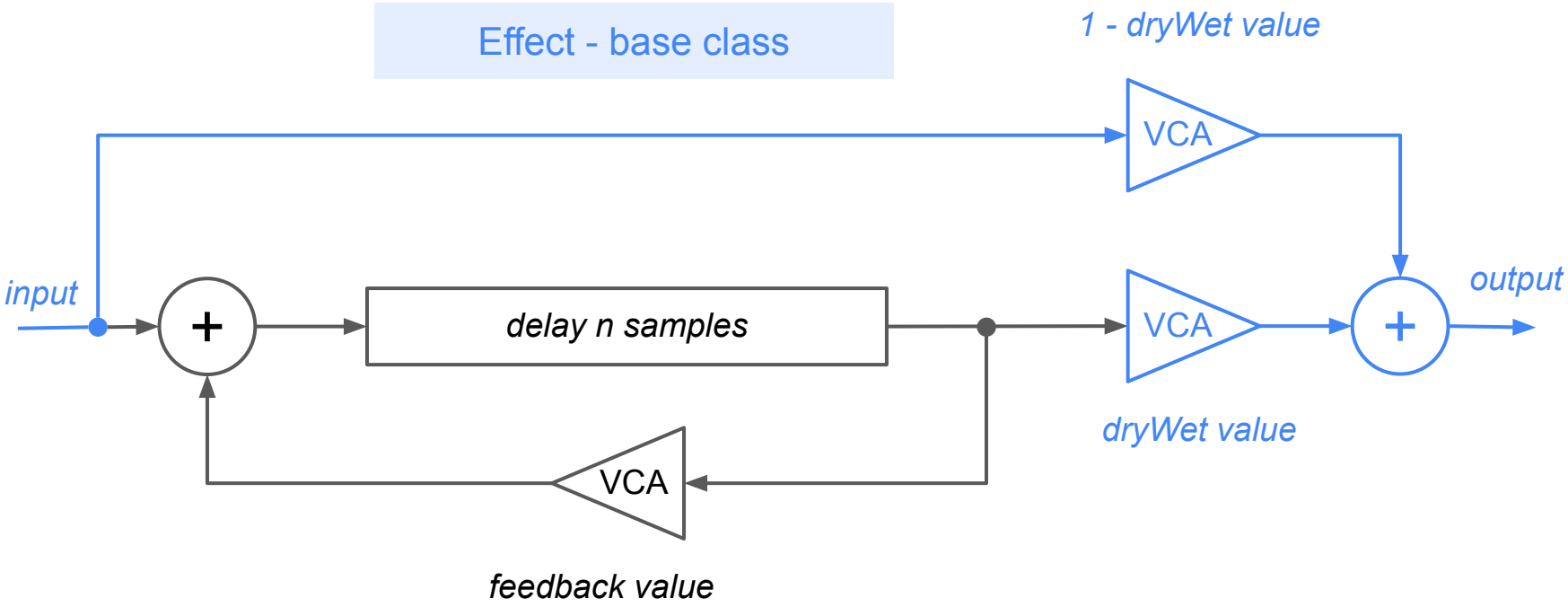
```
applyEffect(float& input, float& output);

…
```

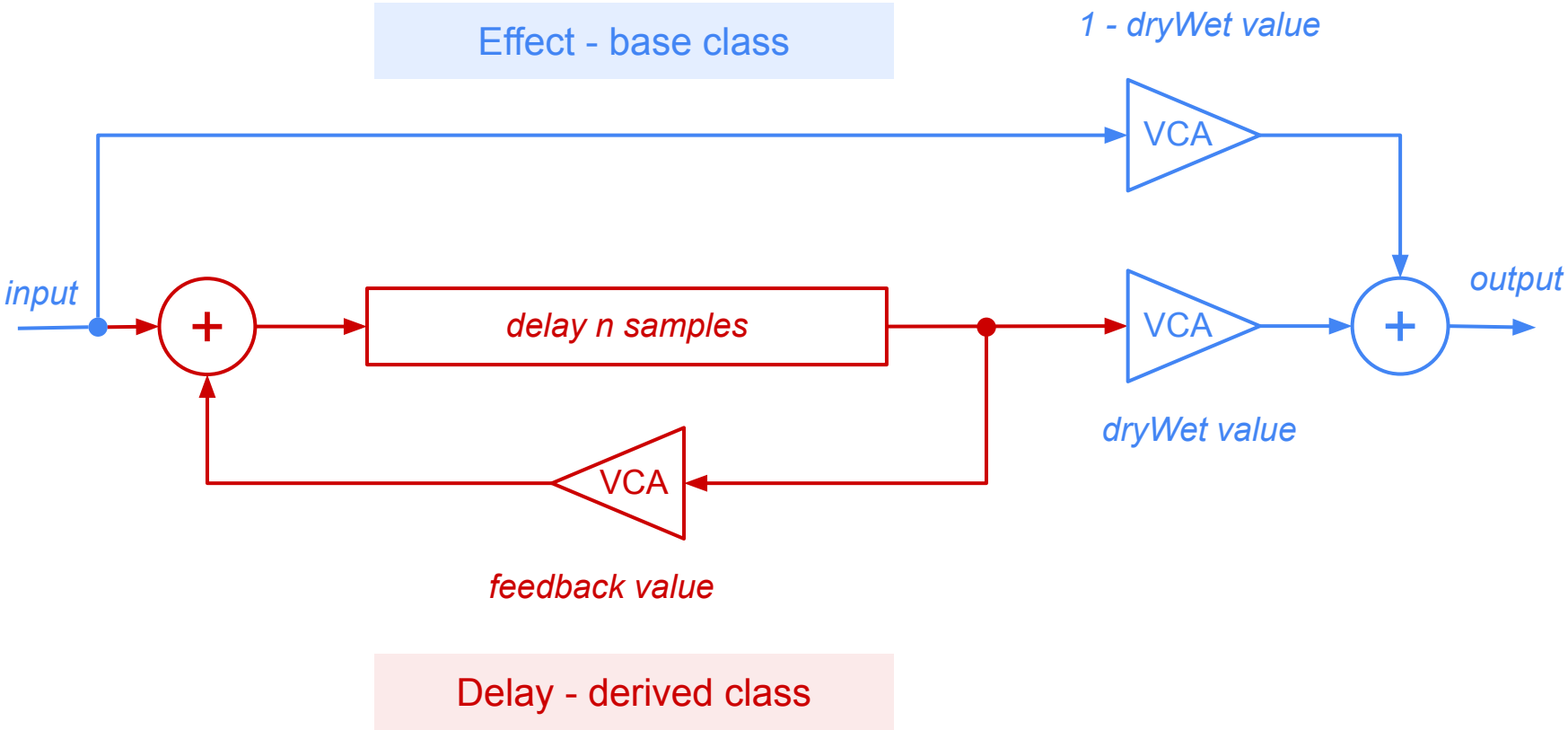
Delay

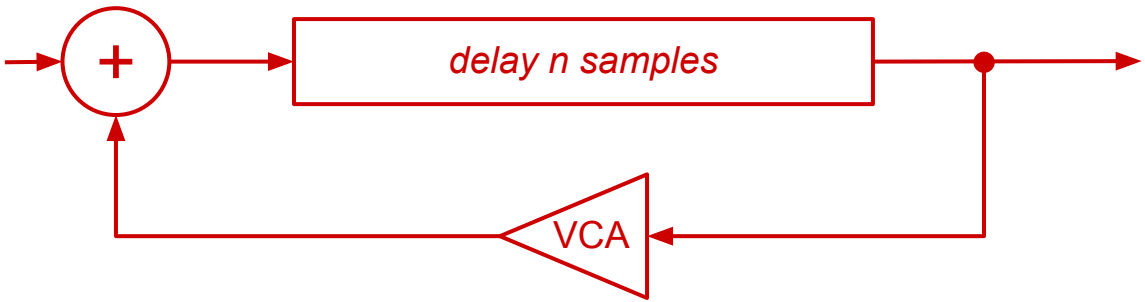
derived from Effect


input
+
delay n samples
output
VCA
feedback value

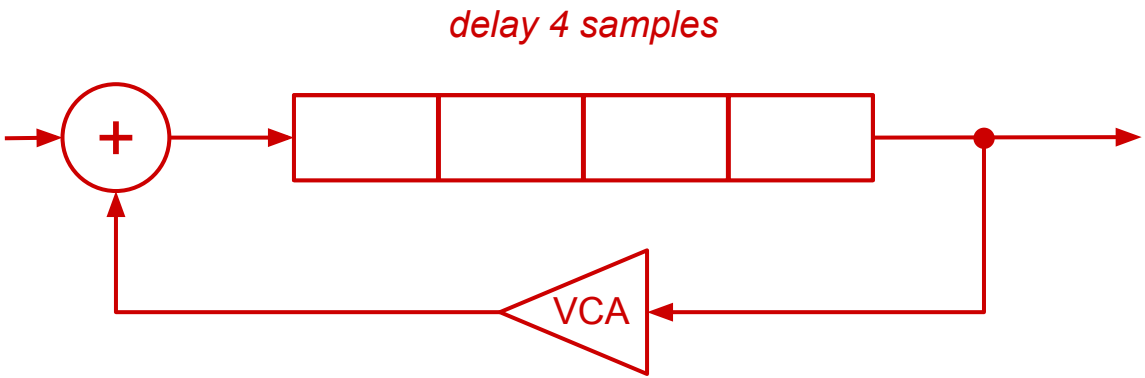

1 - dryWet value
VCA
input
+
delay n samples
VCA
+
output
dryWet value
VCA
feedback value


Effect - base class
1 - dryWet value
VCA
input
delay n samples
VCA
output
dryWet value
VCA
feedback value

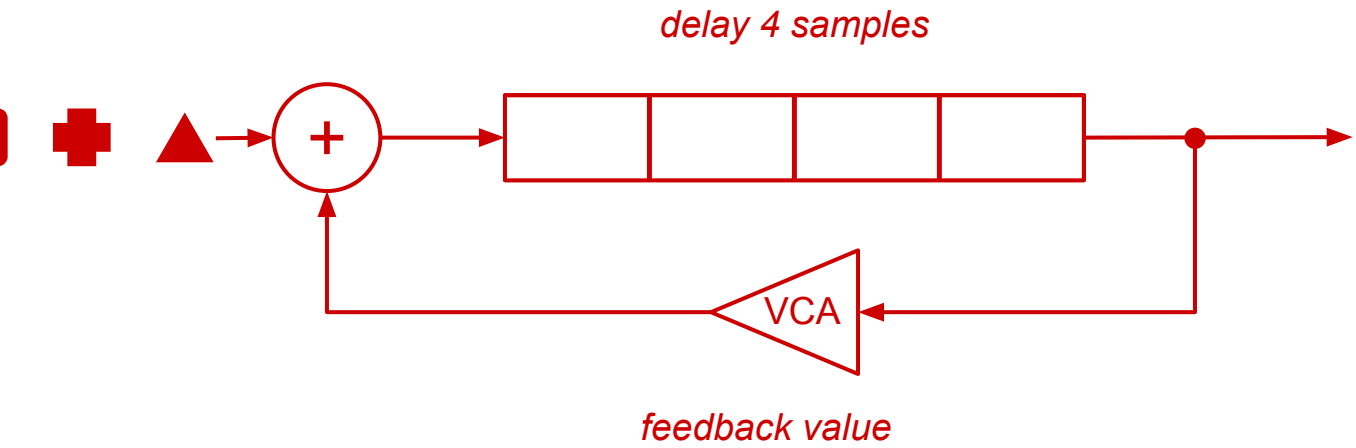

Effect - base class
1 - dryWet value
VCA
input
+
delay n samples
VCA
+
output
dryWet value
VCA
feedback value
Delay - derived class



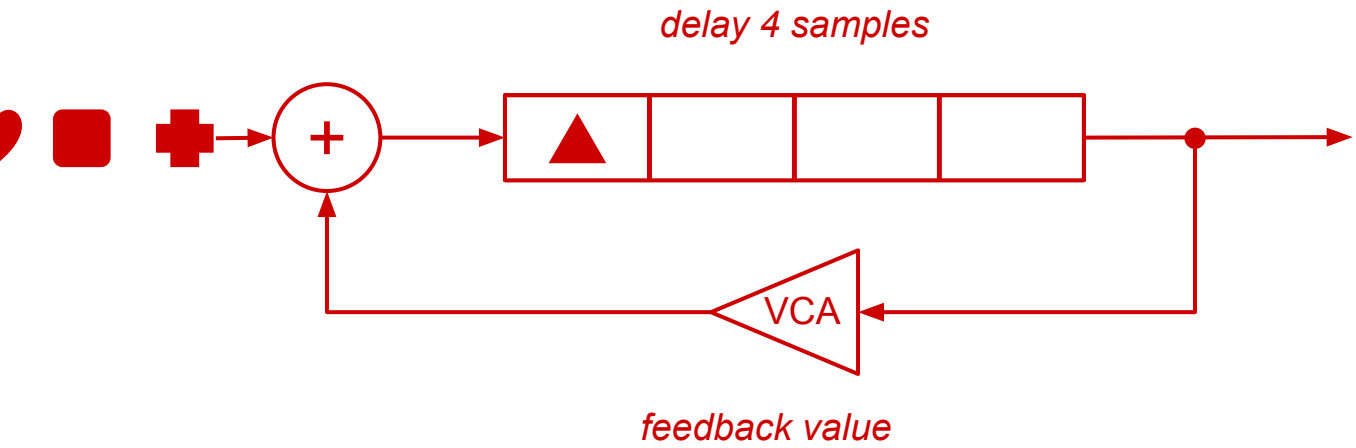
Delay - derived class



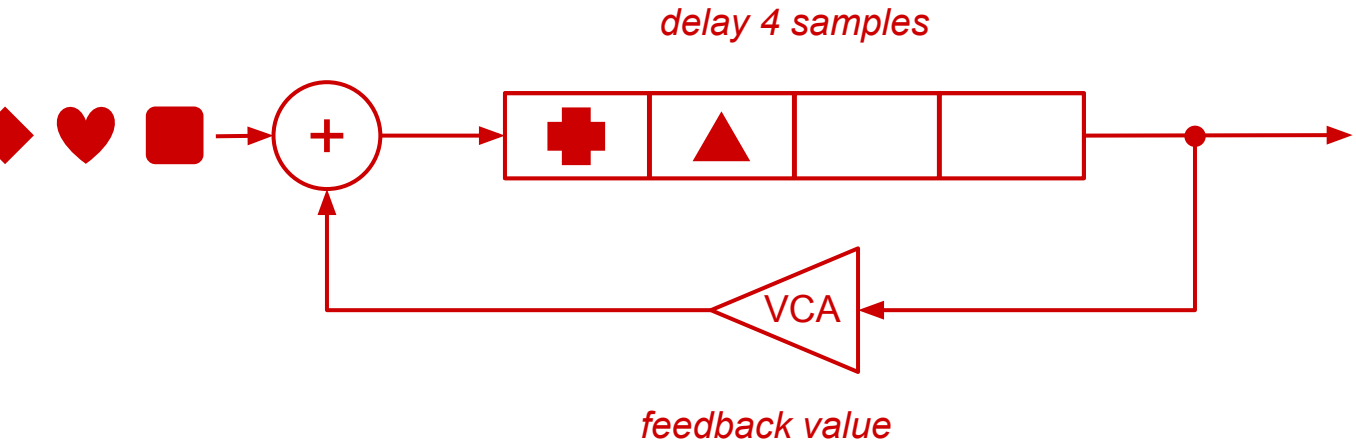
Delay - derived class



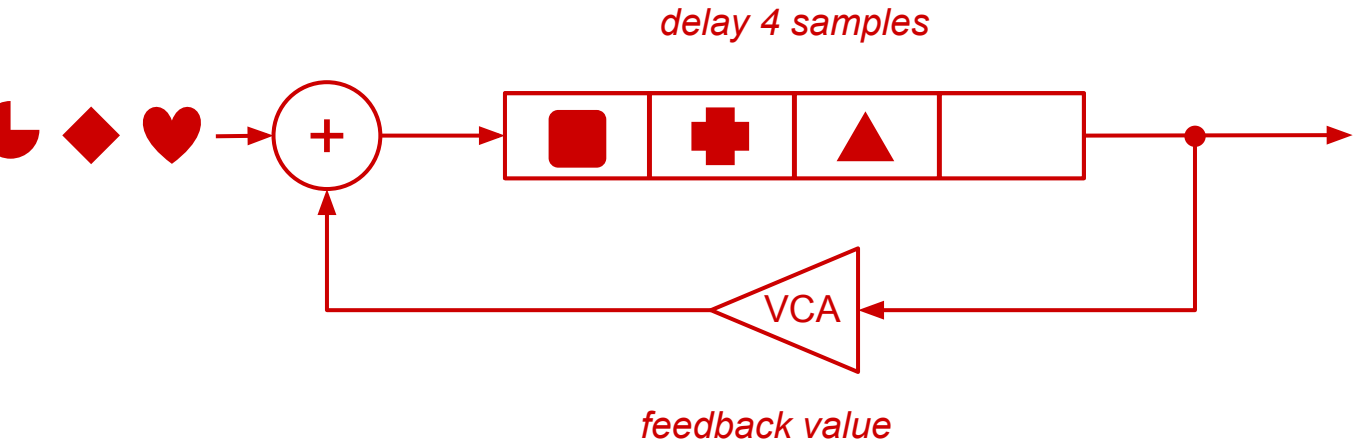
Delay - derived class



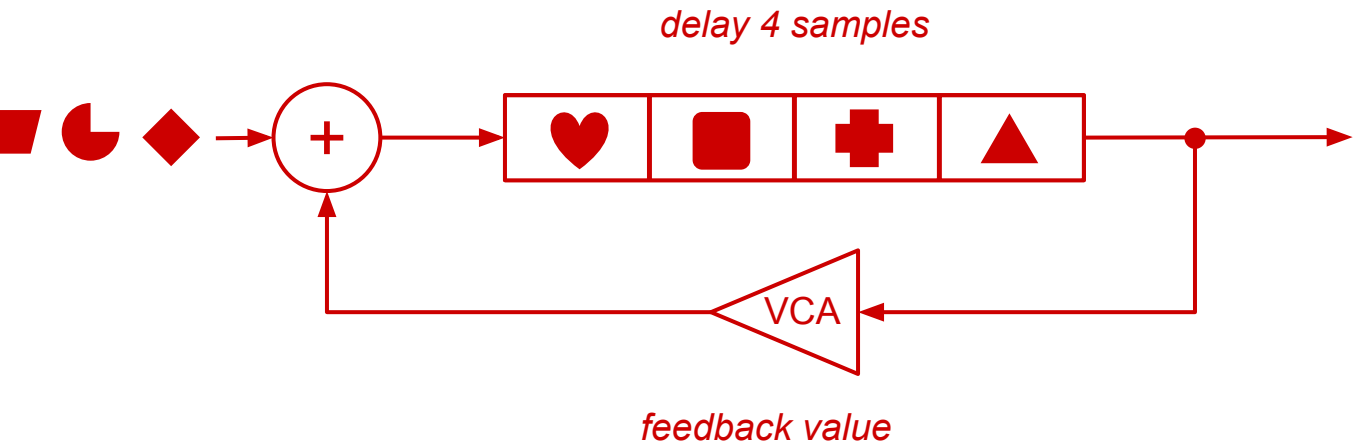
Delay - derived class



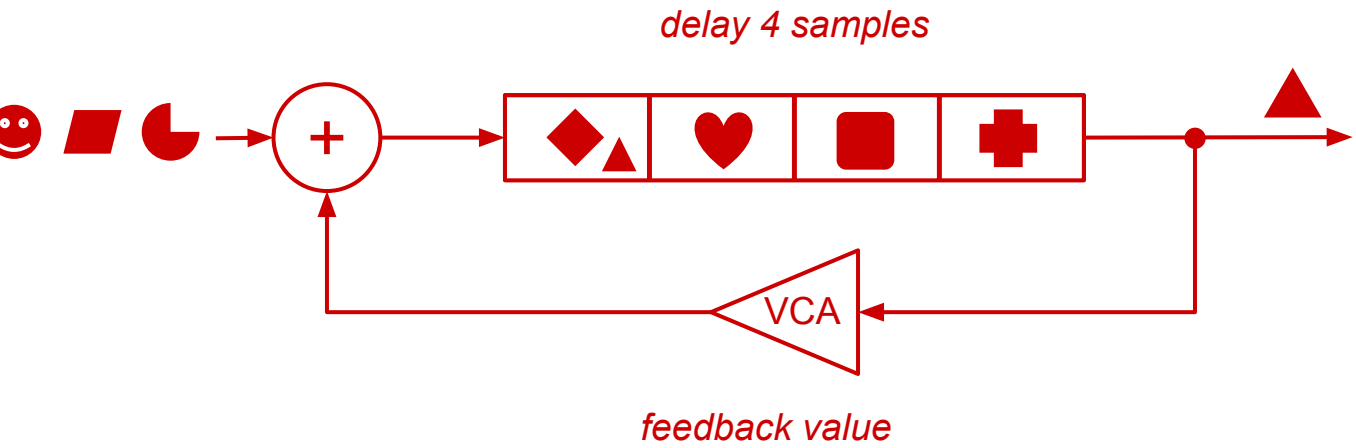
Delay - derived class



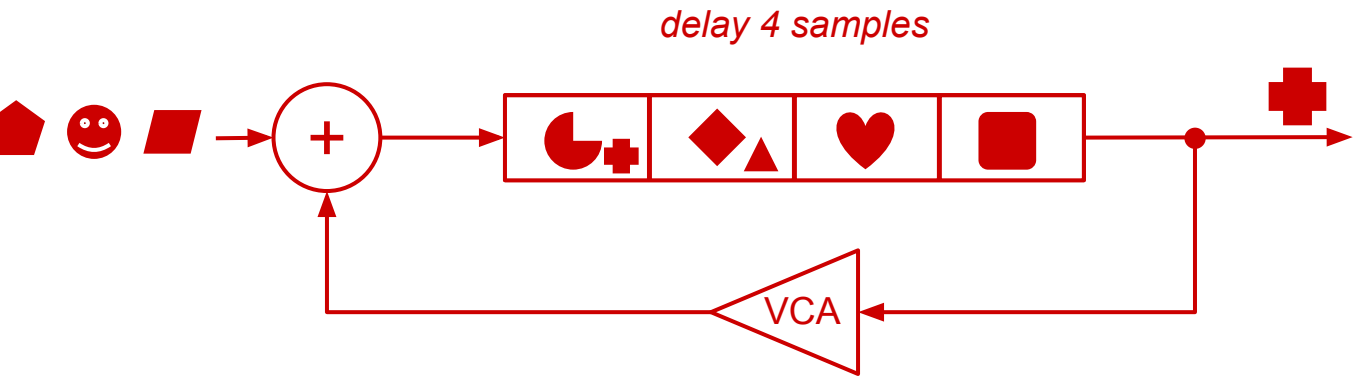
Delay - derived class



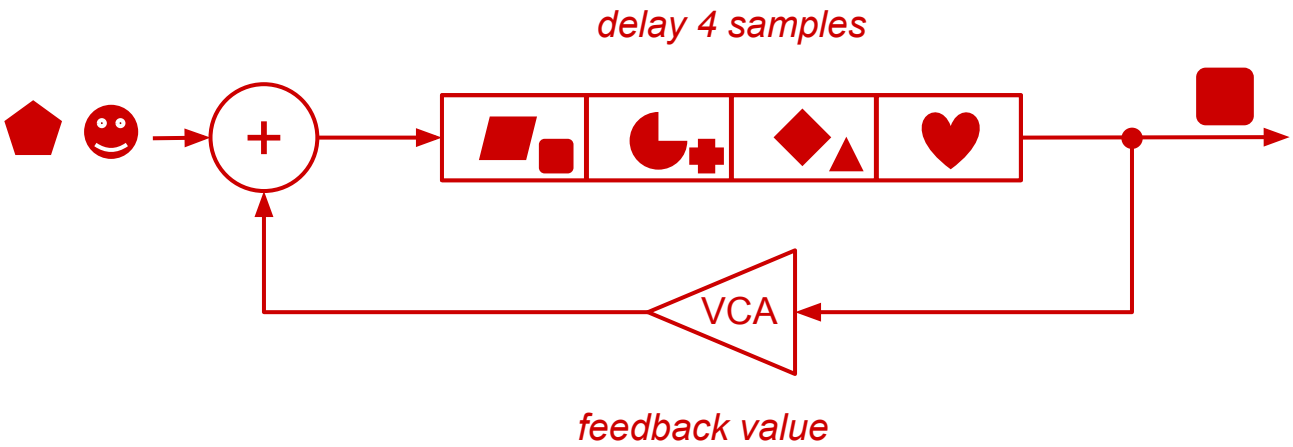
Delay - derived class



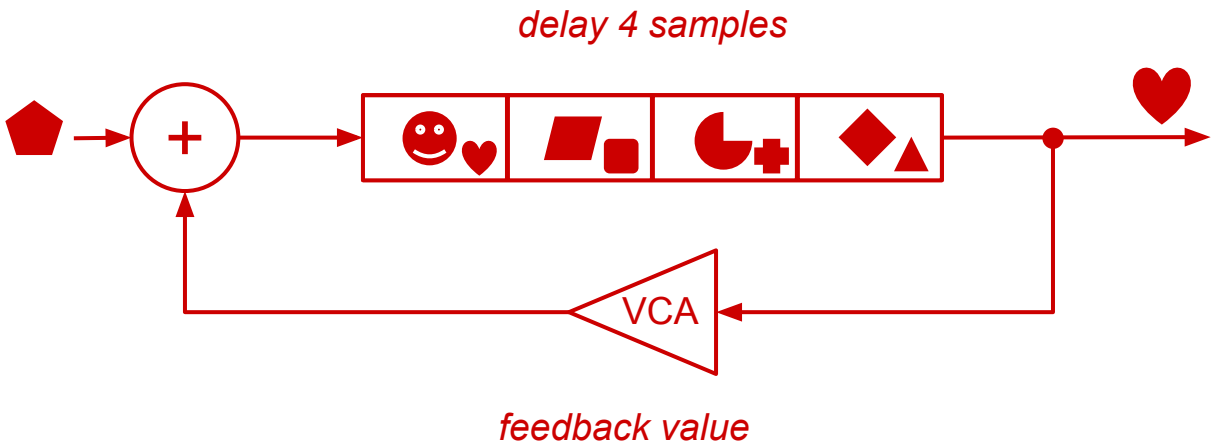
Delay - derived class



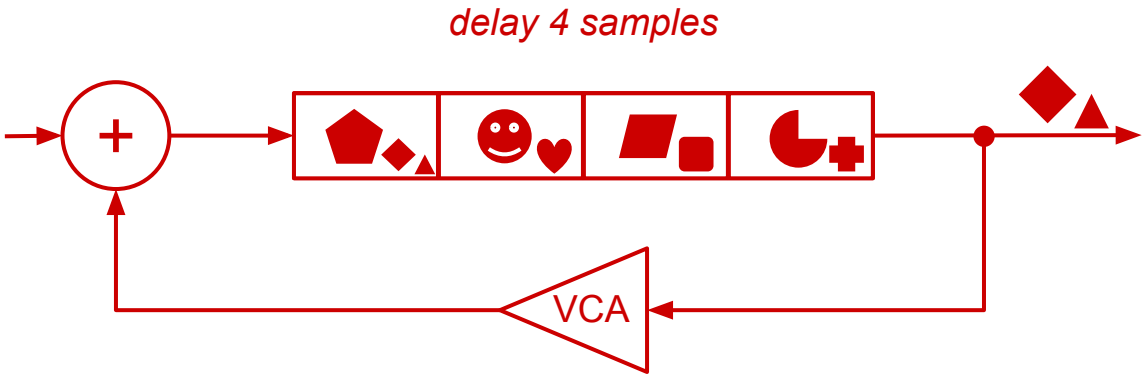
Delay - derived class



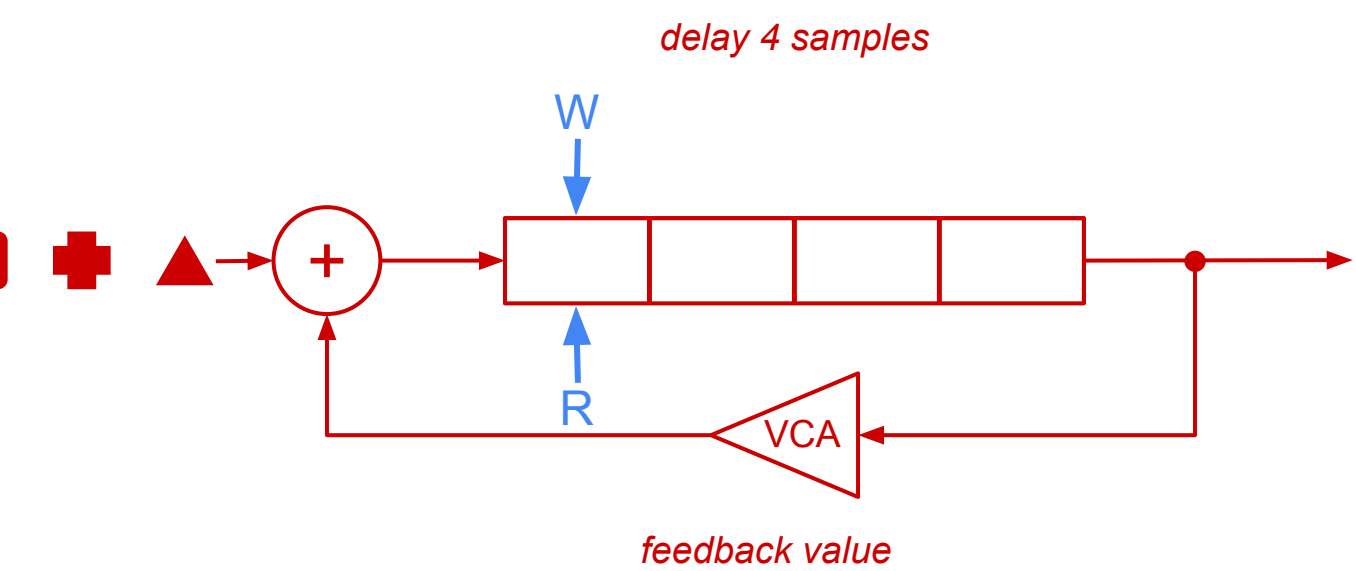
Delay - derived class



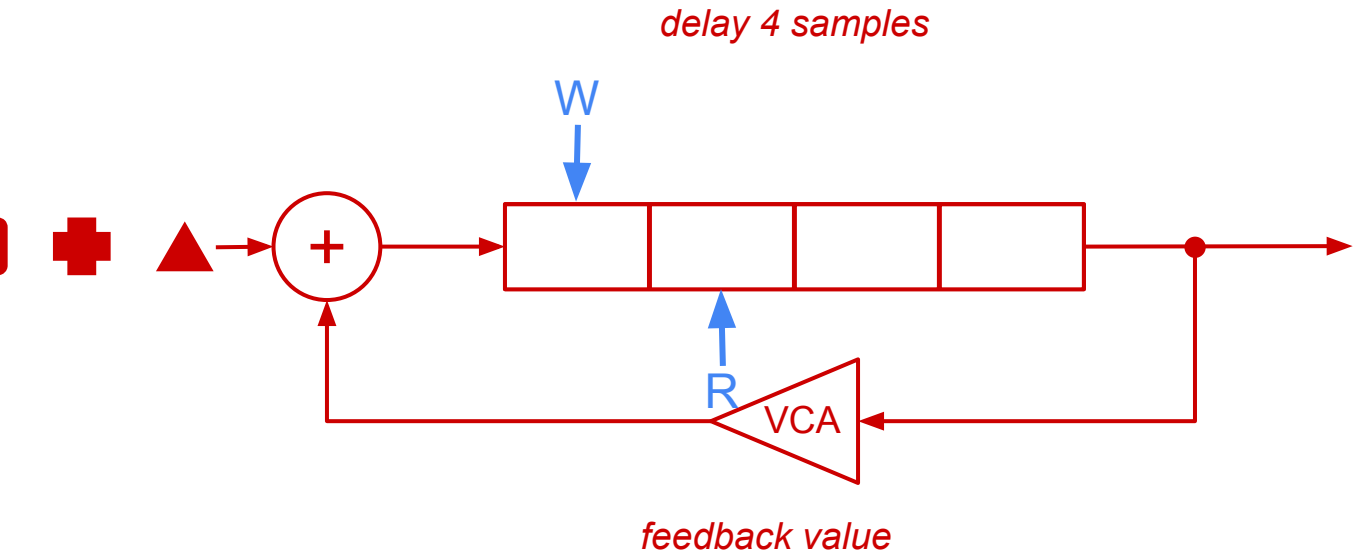
Delay - derived class



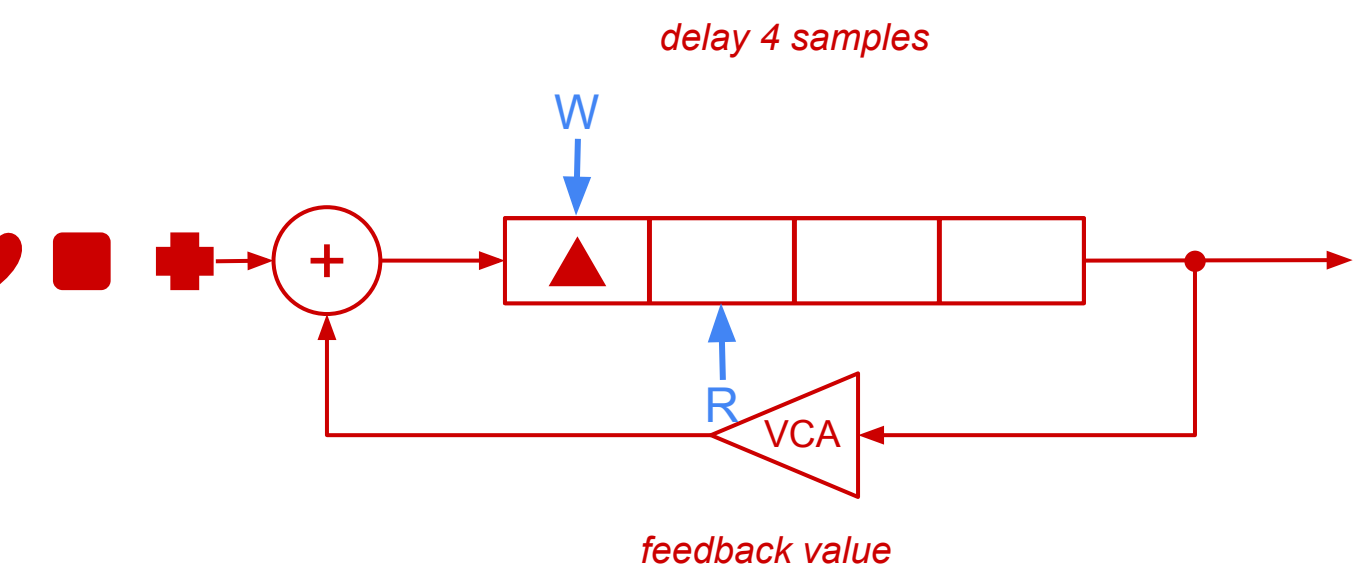
Delay - derived class



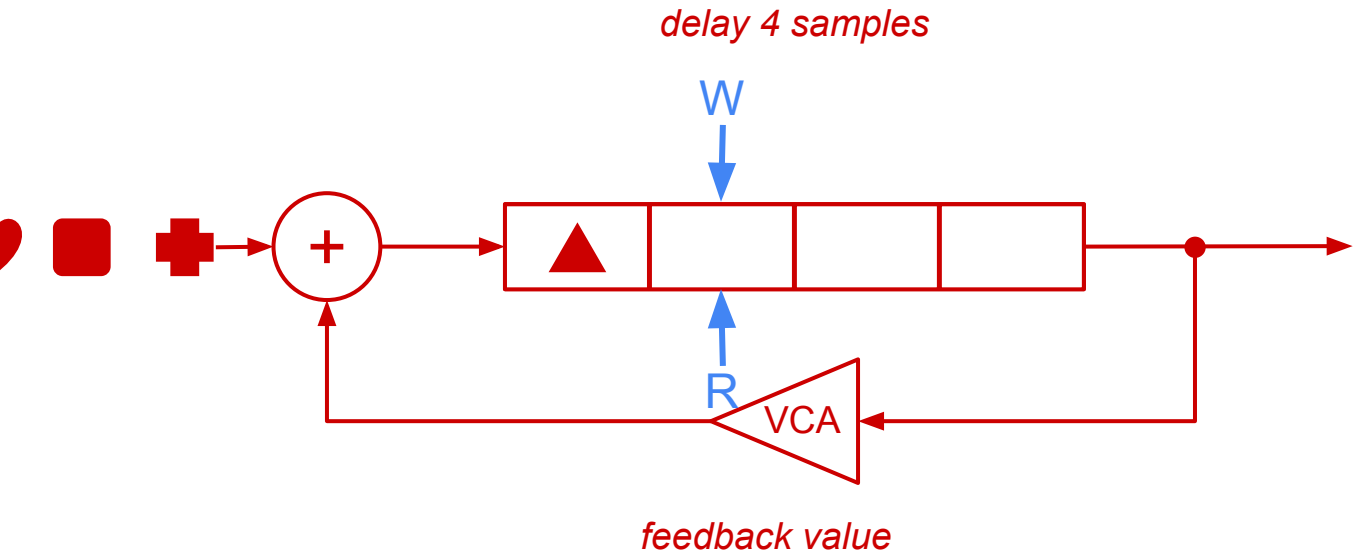
Delay - derived class



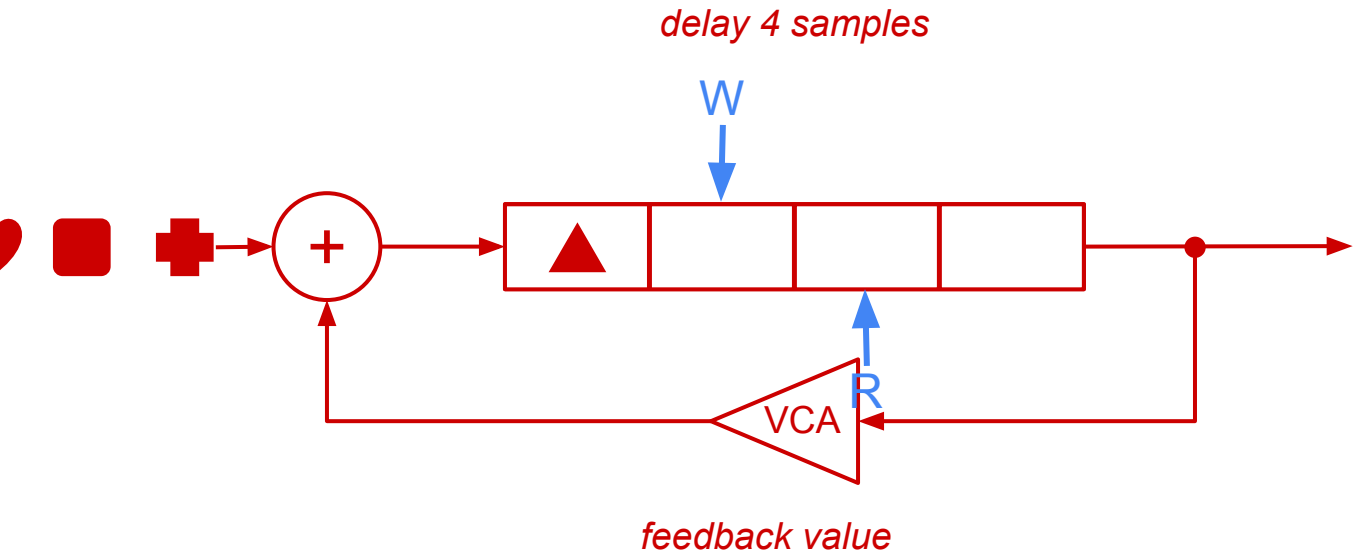
Delay - derived class



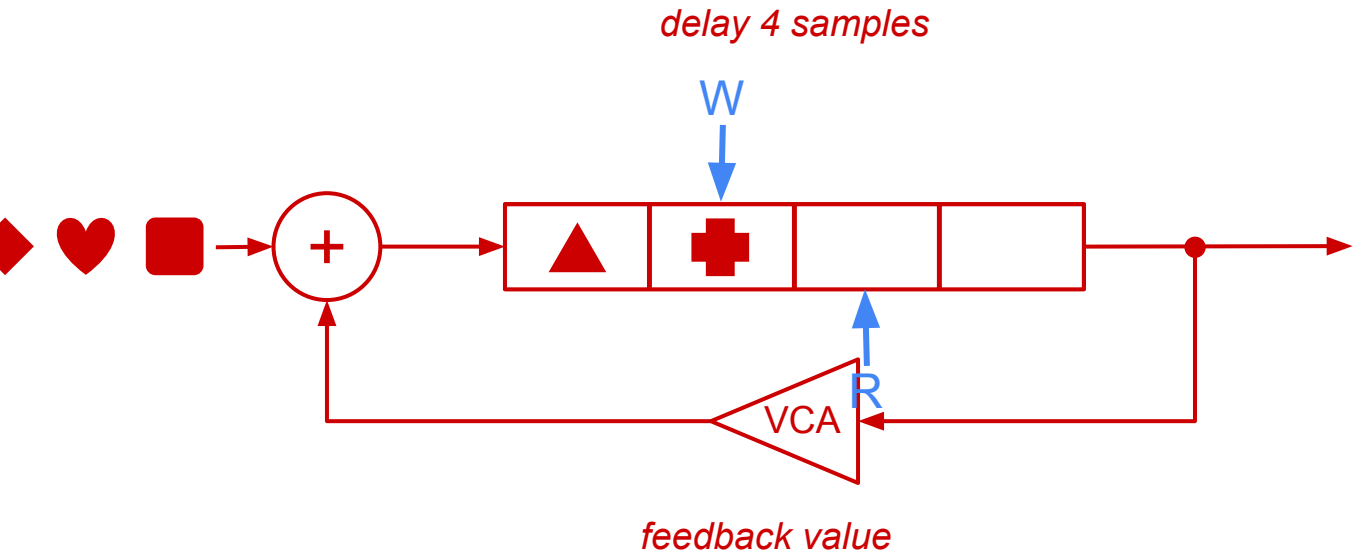
Delay - derived class



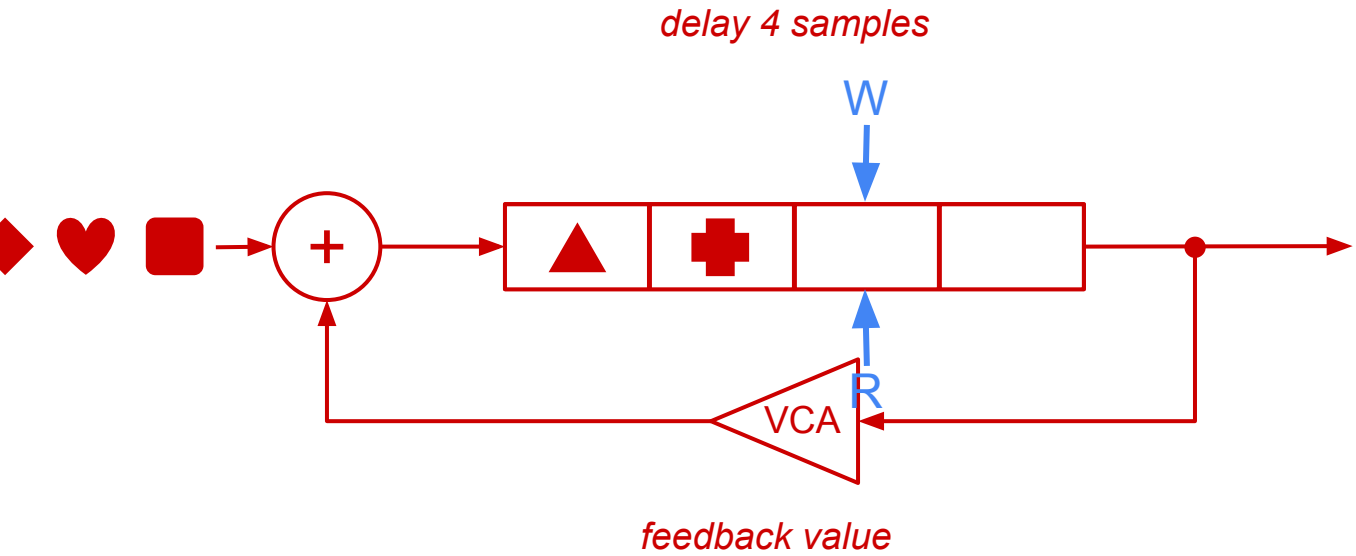
Delay - derived class



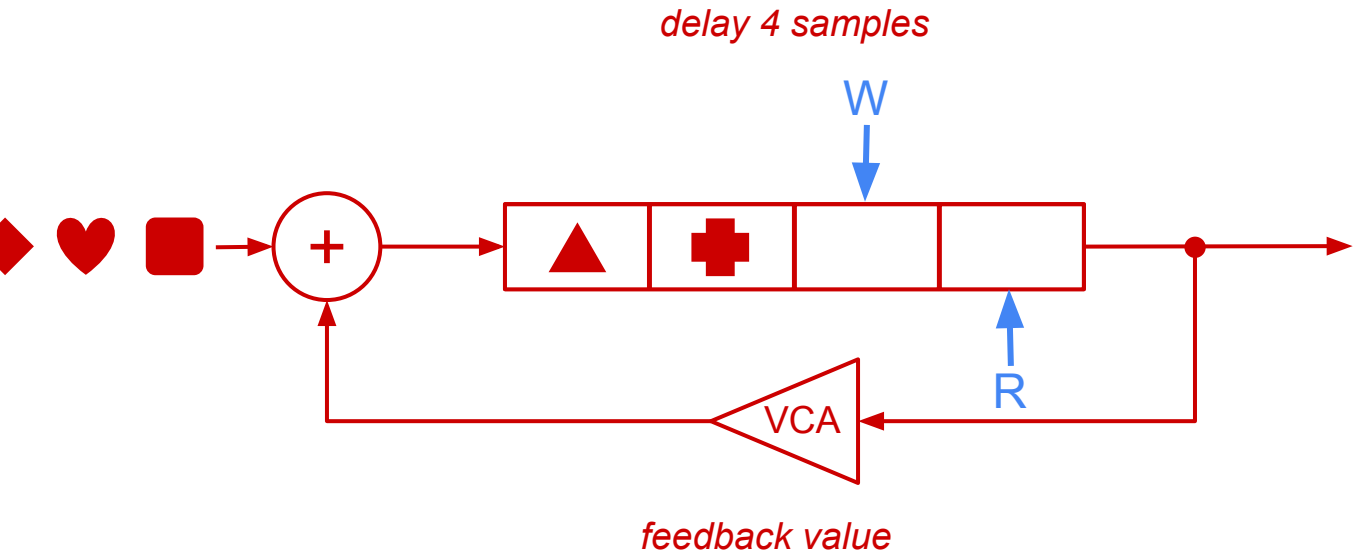
Delay - derived class



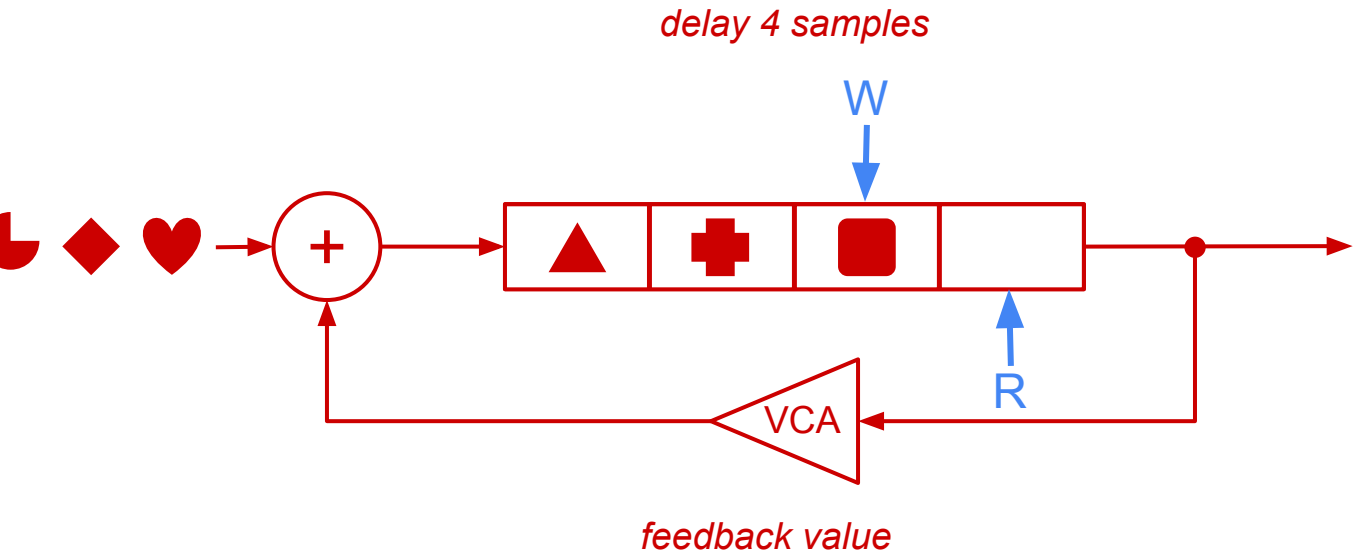
Delay - derived class



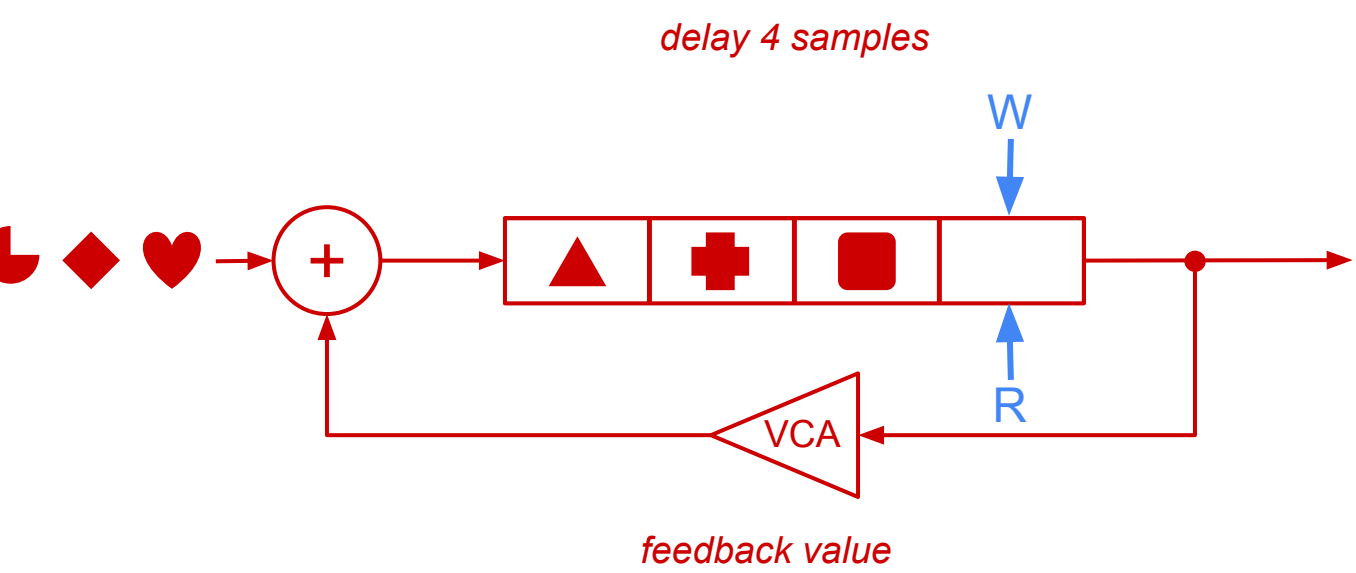
Delay - derived class



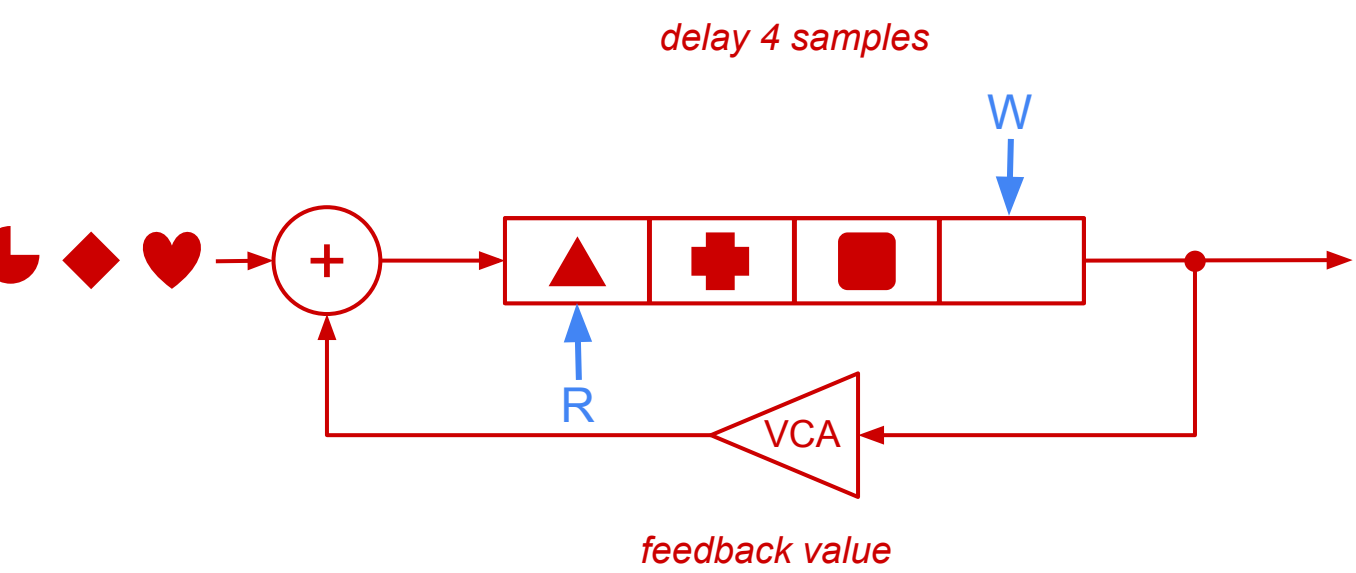
Delay - derived class



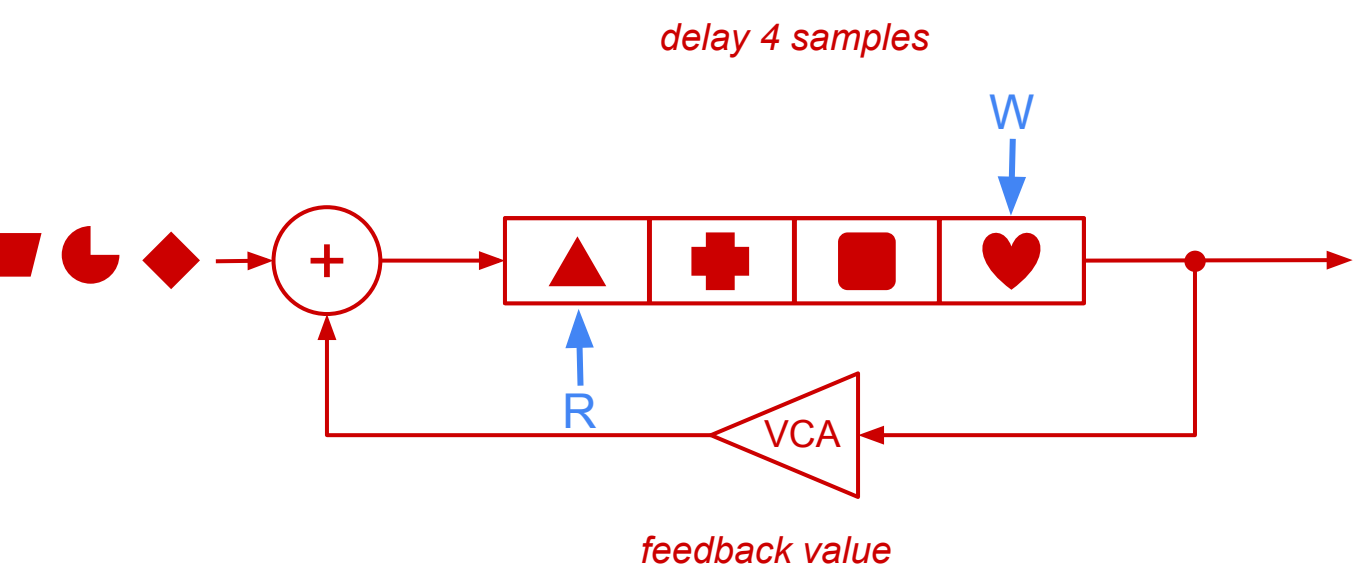
Delay - derived class



Delay - derived class

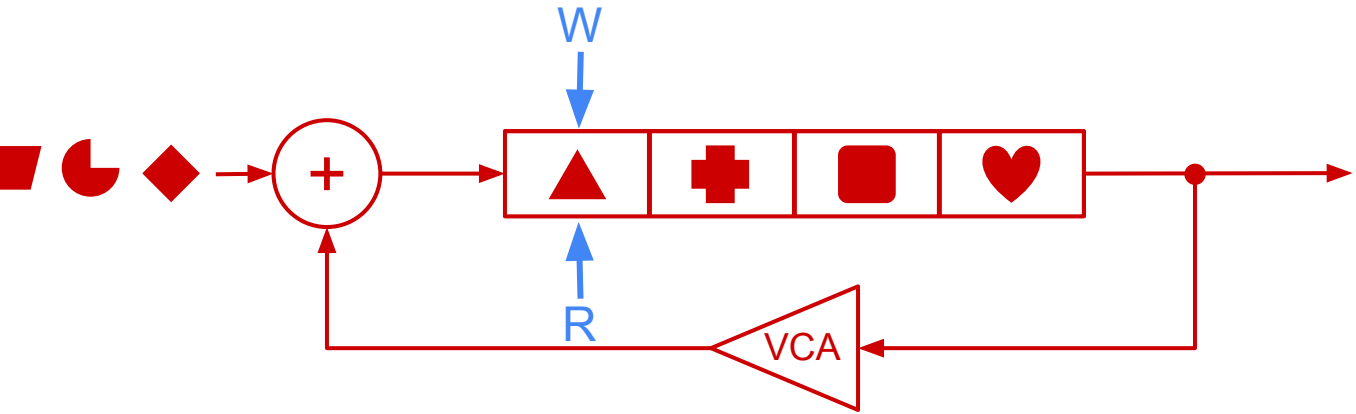


Delay - derived class



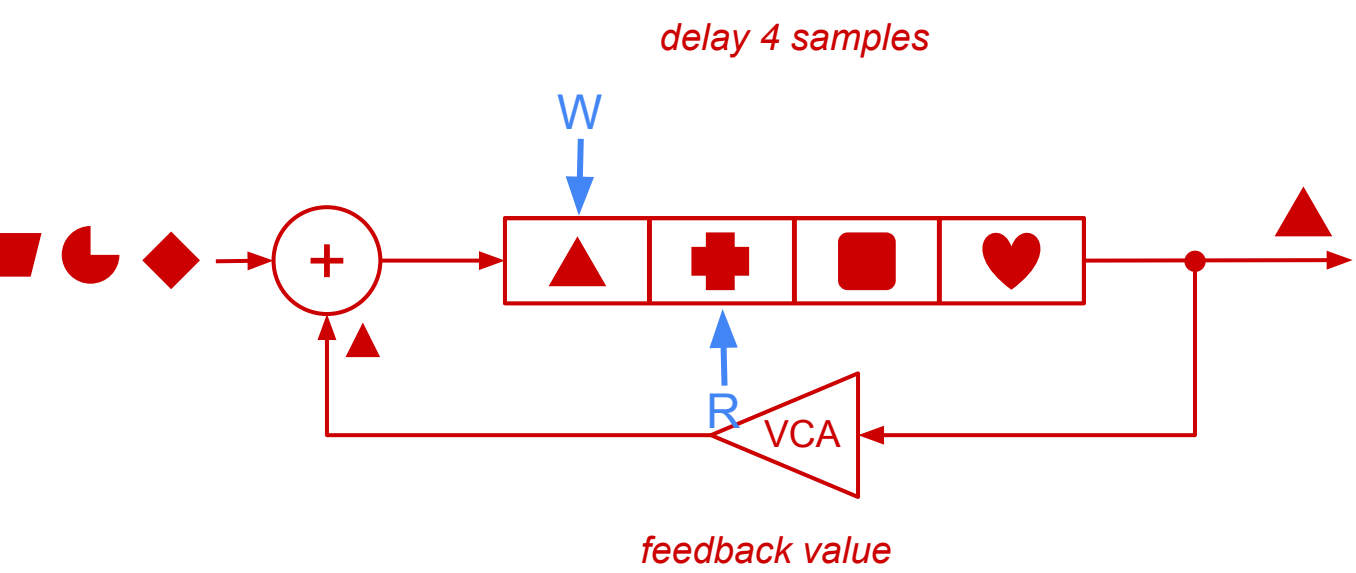
Delay - derived class

delay 4 samples

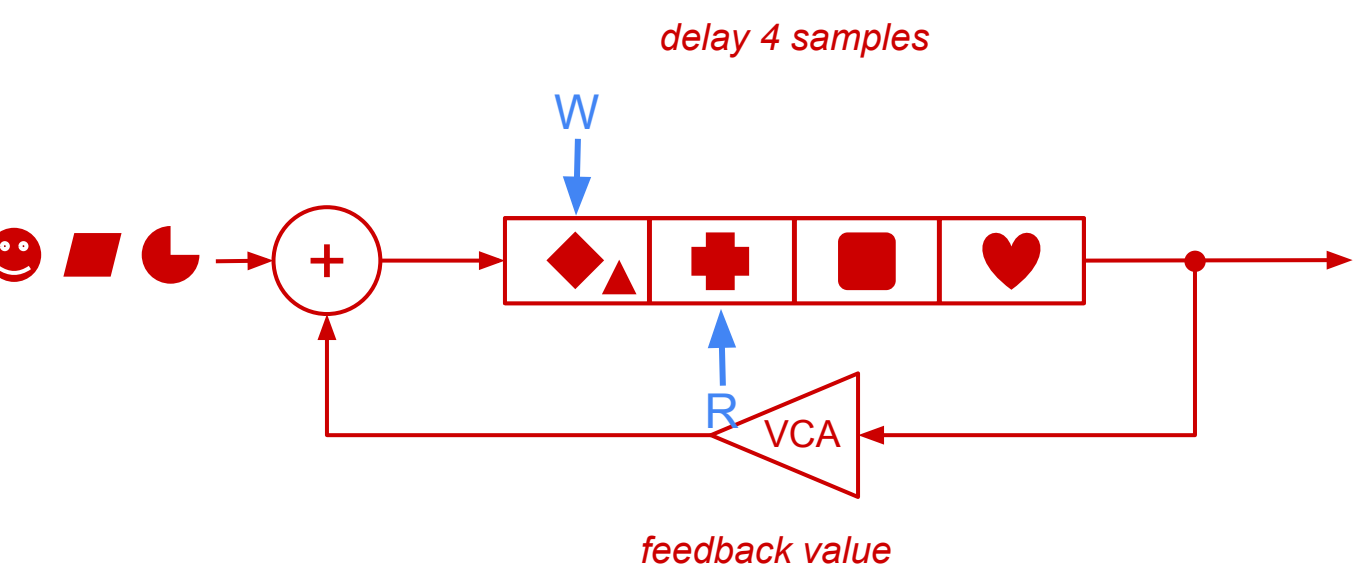


feedback value

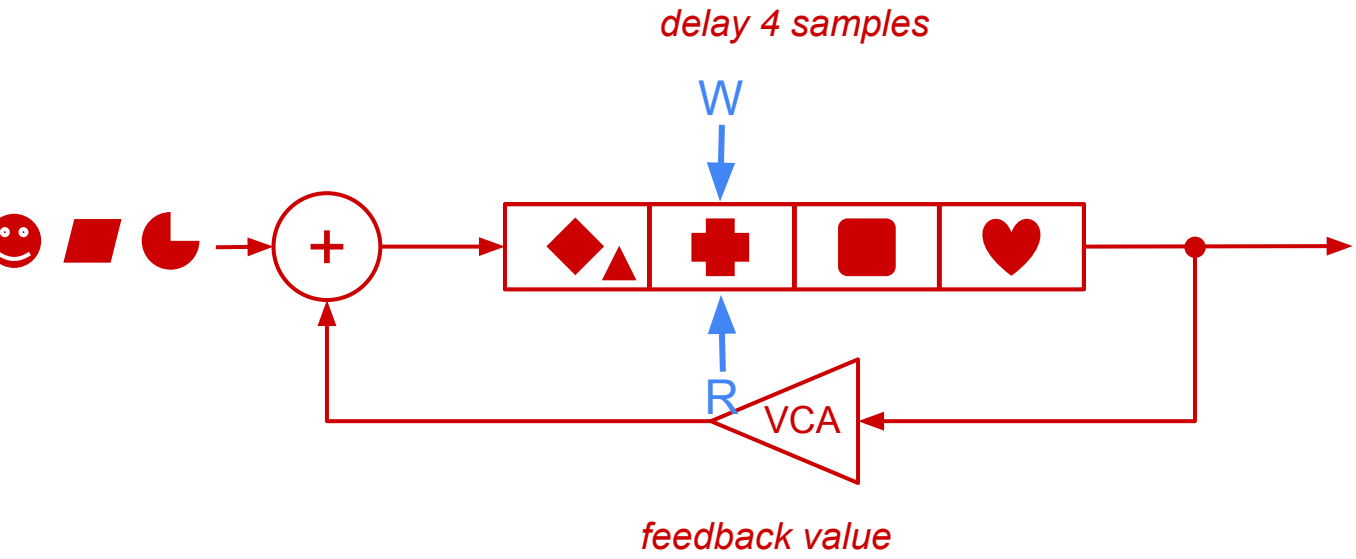
Delay - derived class



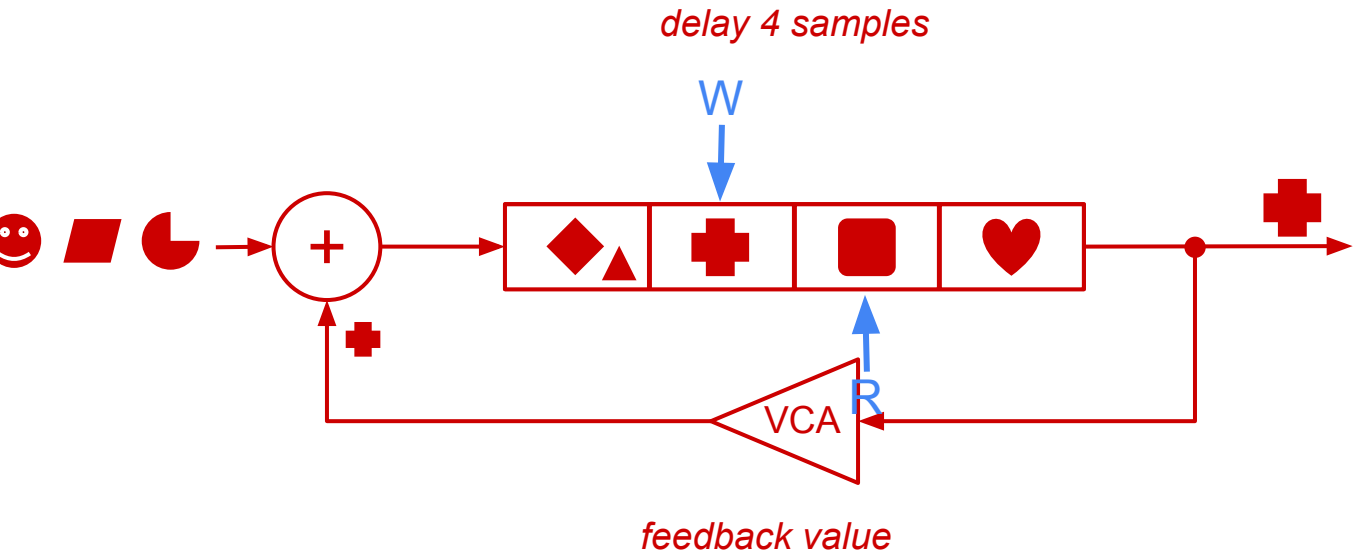
Delay - derived class



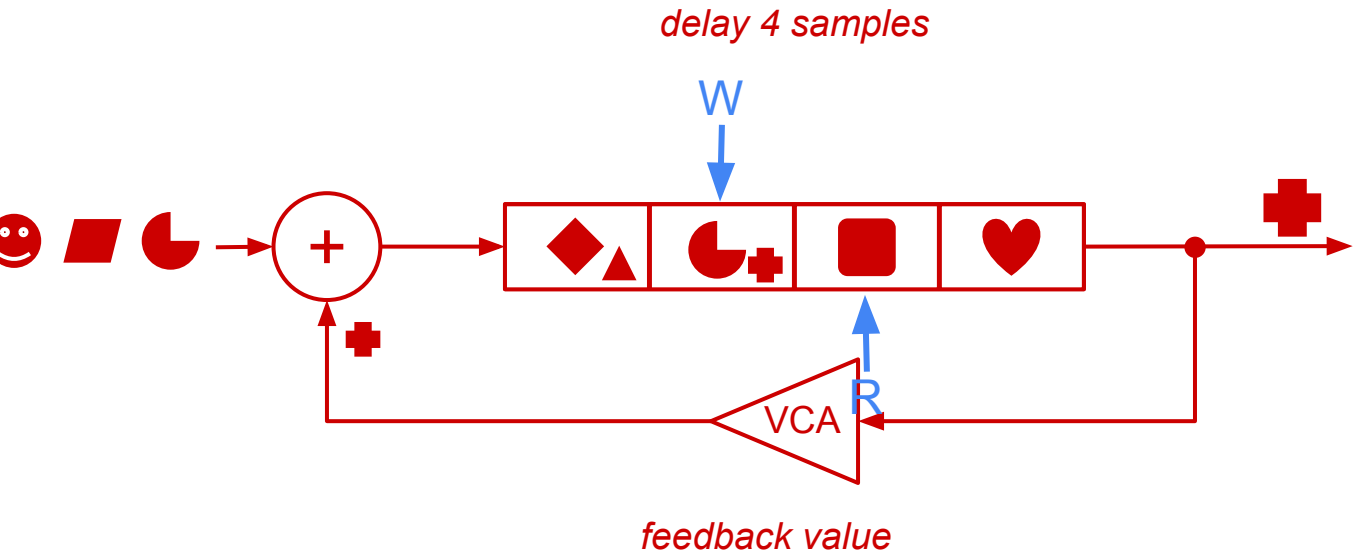
Delay - derived class



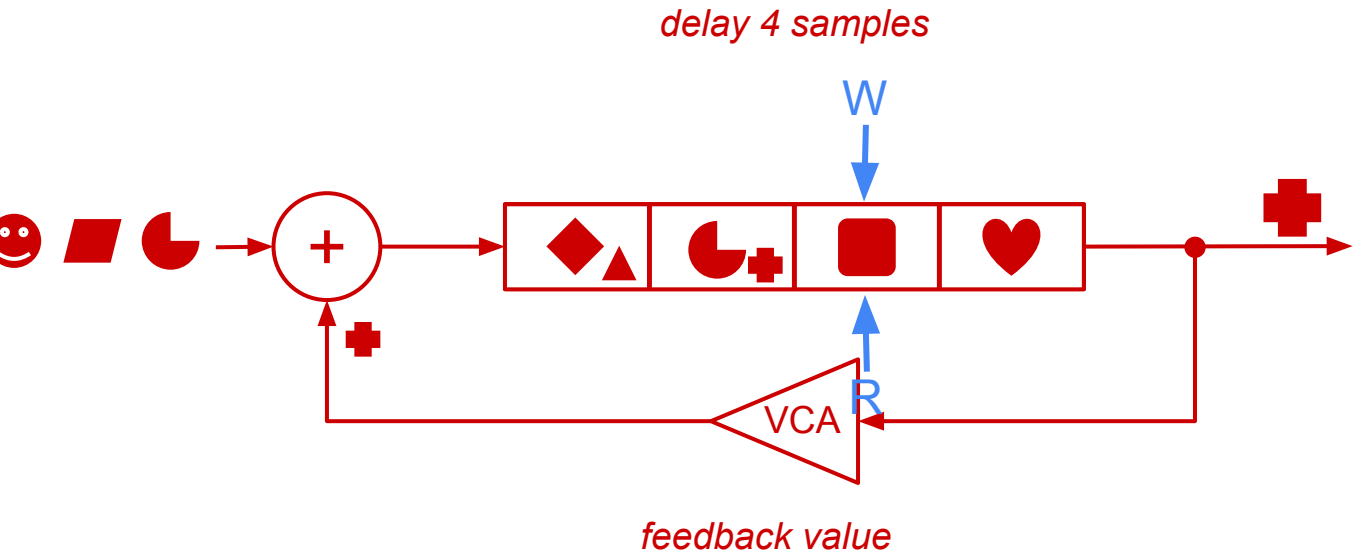
Delay - derived class



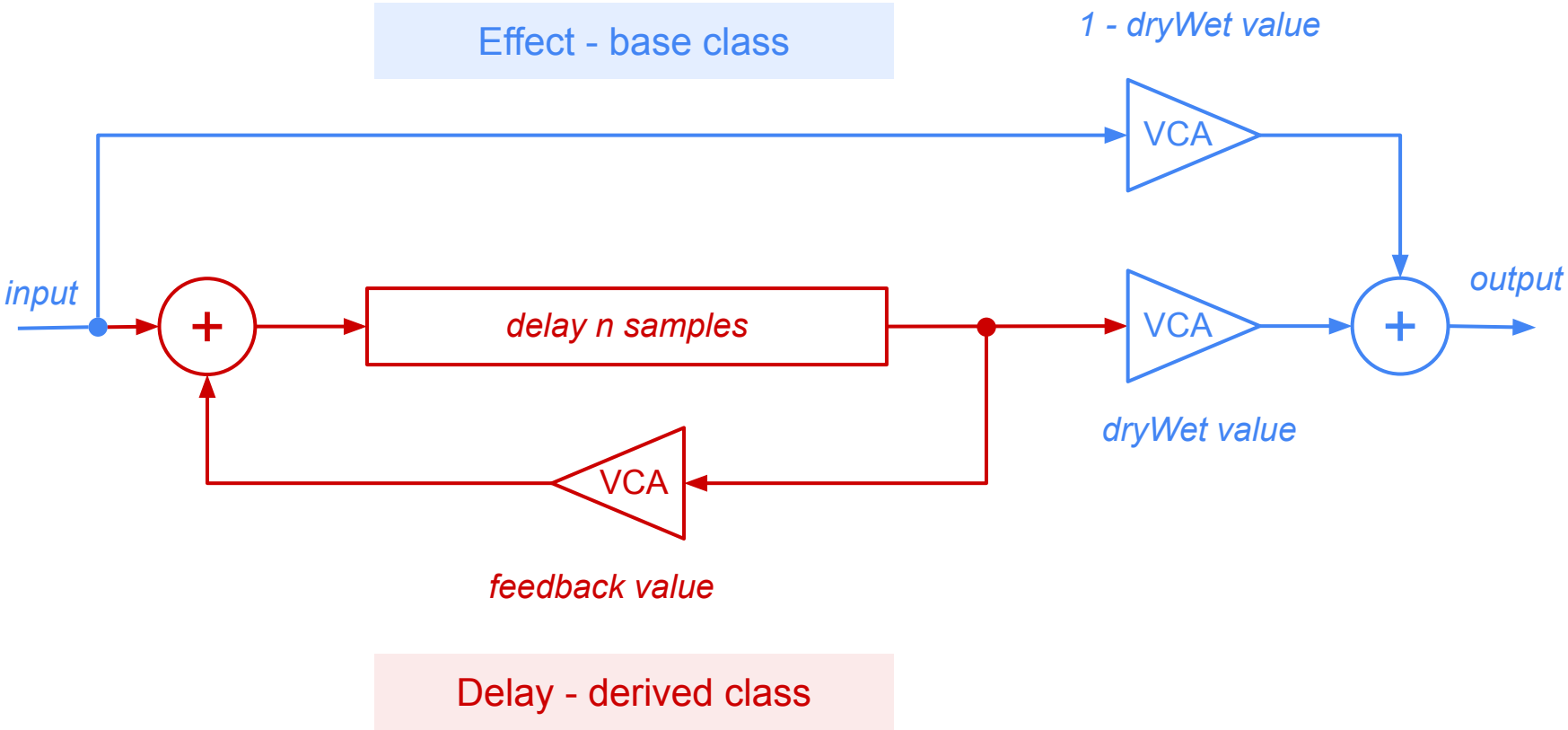
Delay - derived class



Delay - derived class



Delay - derived class


Effect - base class
1 - dryWet value
VCA
input
+
delay n samples
VCA
+
output
dryWet value
VCA
feedback value
Delay - derived class

Effect

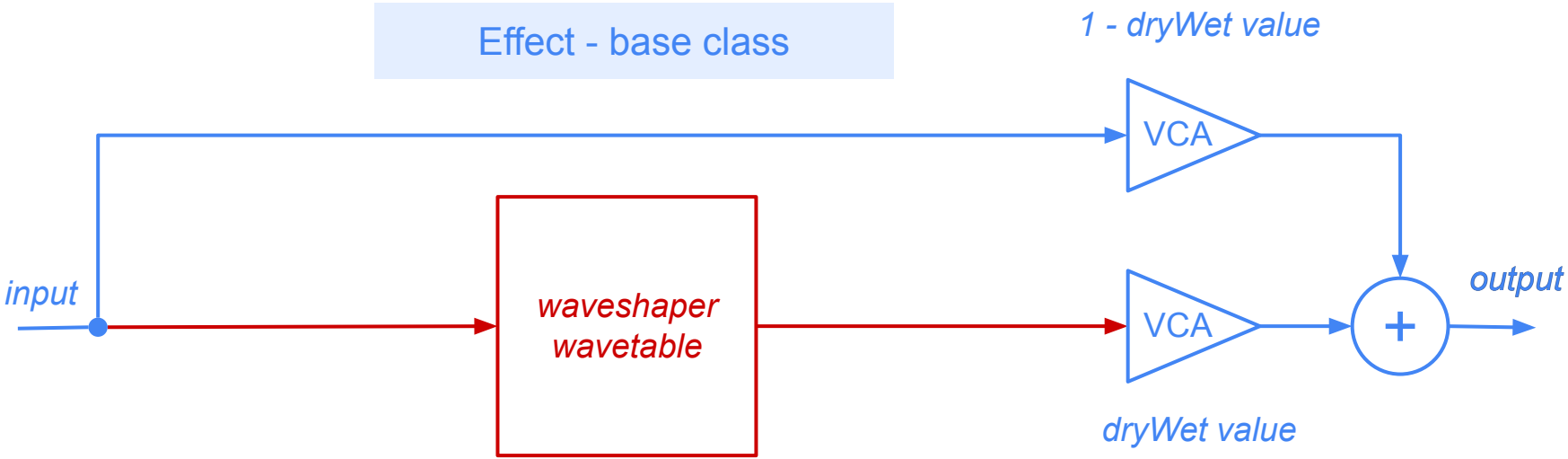
```
processFrame(float& input, float& output);
setDryWet(float dryWet);
setBypass(bool bypass);

…
```

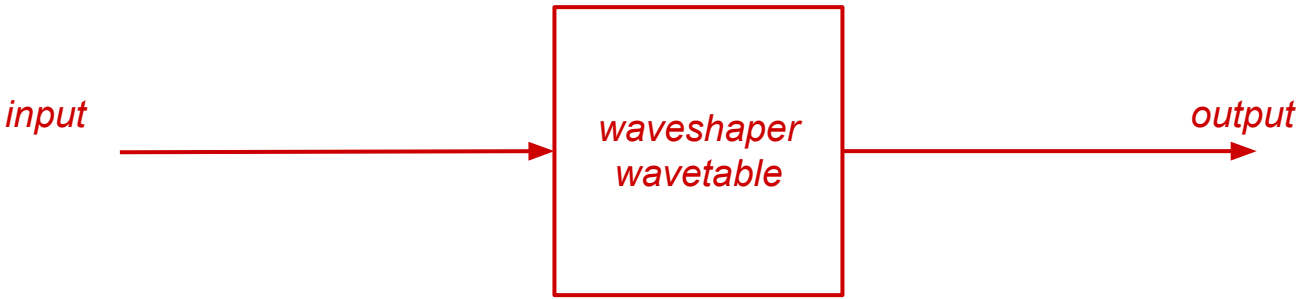
Delay

```
applyEffect(float& input, float& output);

float[] circBuffer;
int readH;
int writeH;
float feedback;
…
```

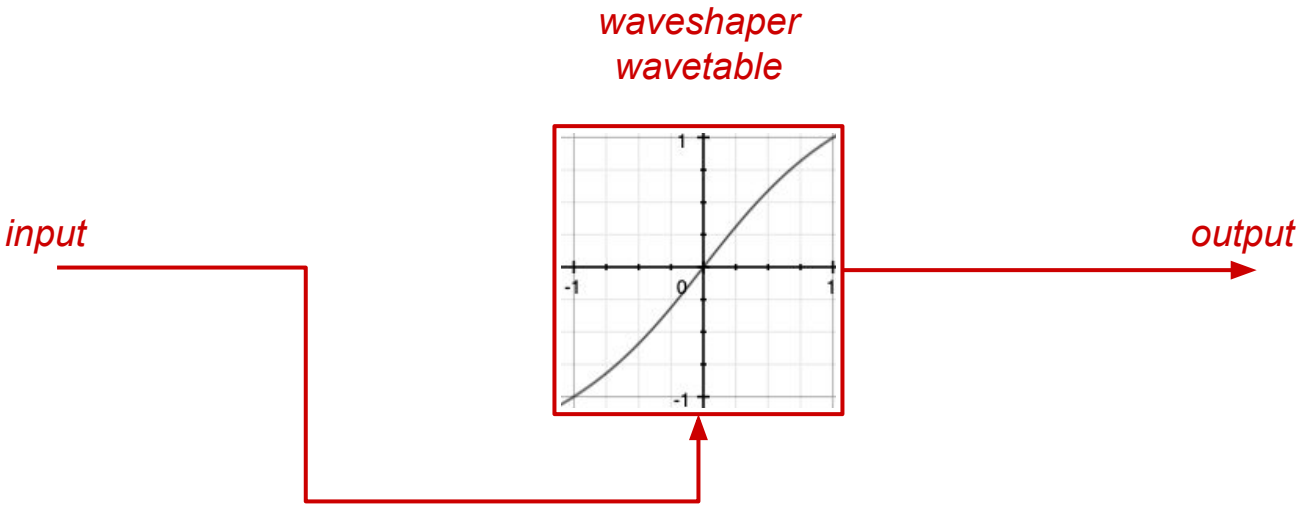
Waveshaper

derived from Effect


Effect - base class
1 - dryWet value
VCA
input
waveshaper
wavetable
output
VCA
+
dryWet value
Waveshaper - derived class

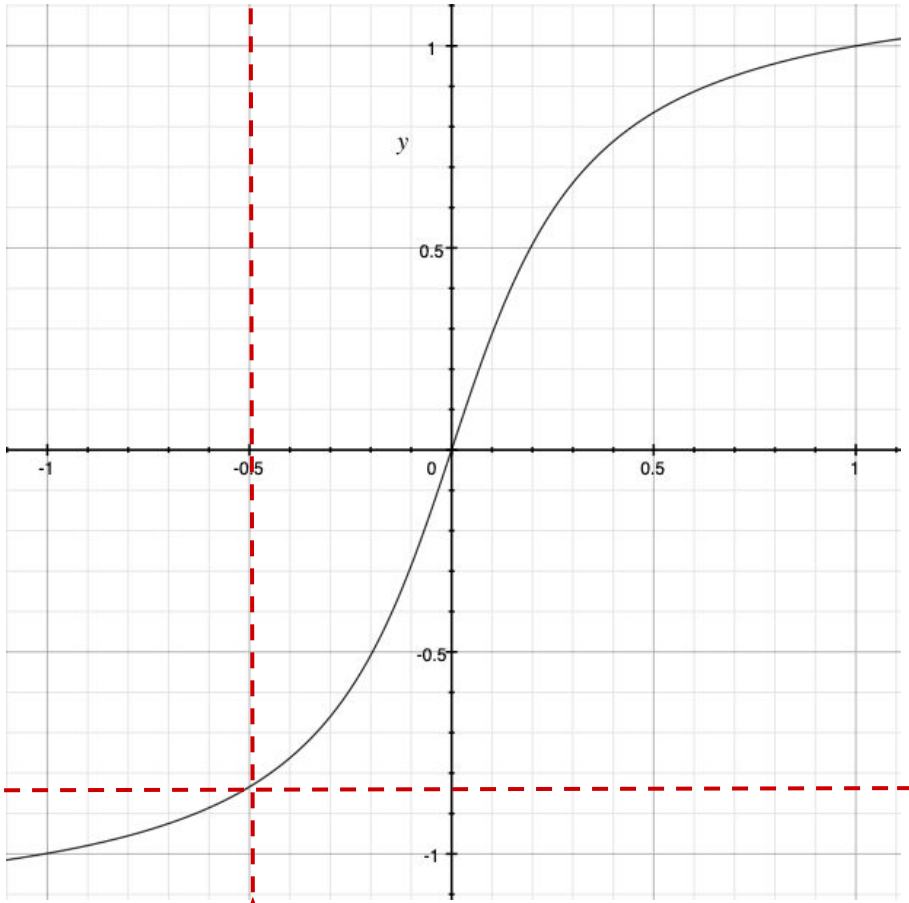


Waveshaper - derived class



Waveshaper - derived class

wavetable with arctan



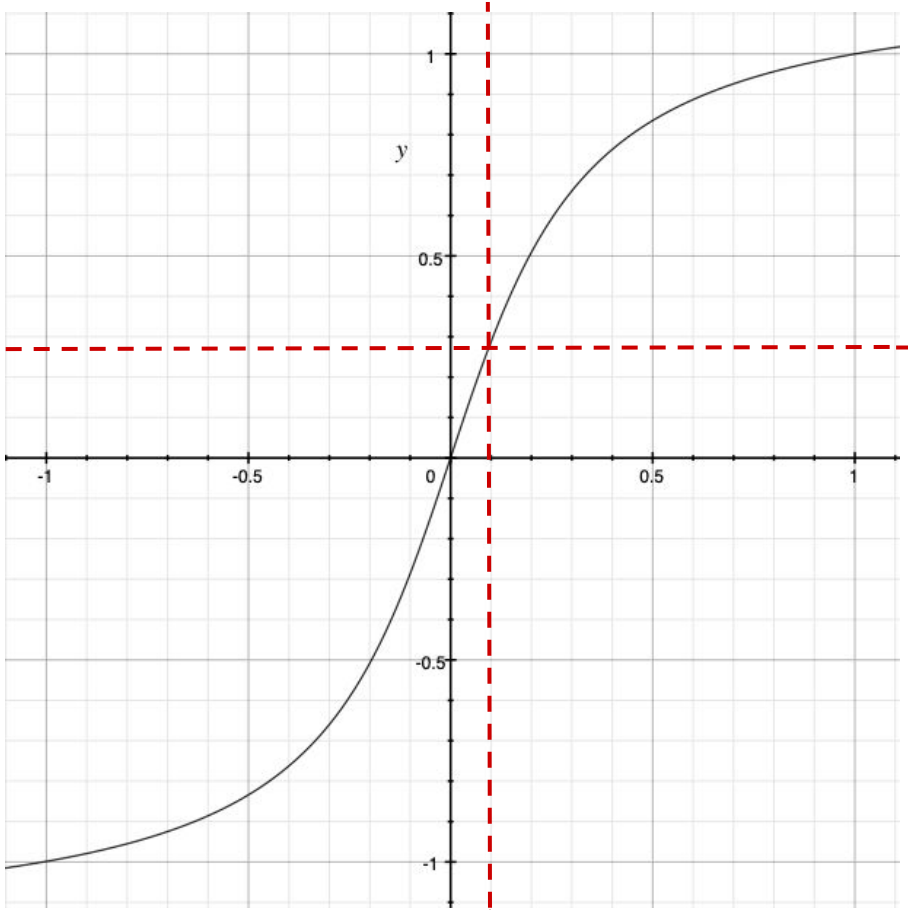
input

e.g. sample value = -0.5

output

sample value ≈ -0.83

wavetable with arctan



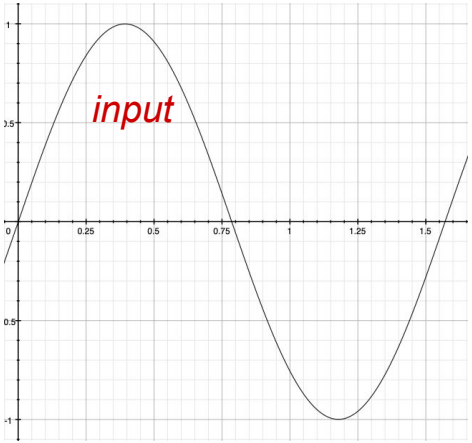
input

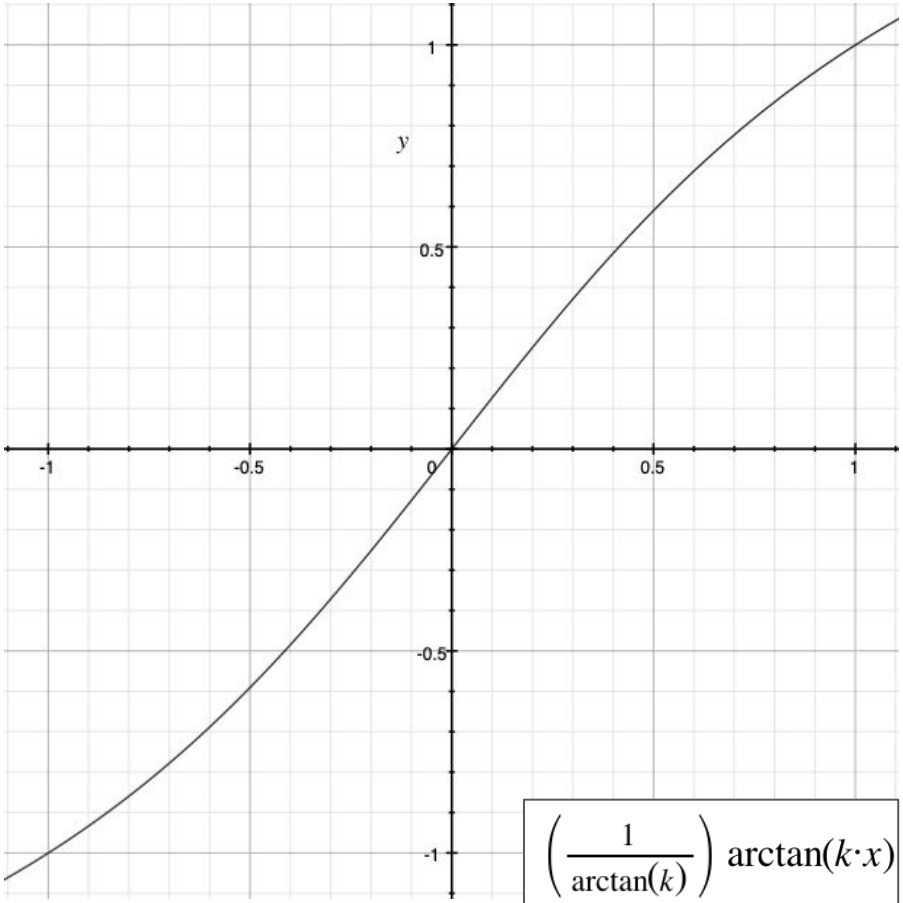
e.g. sample value = 0.1

output

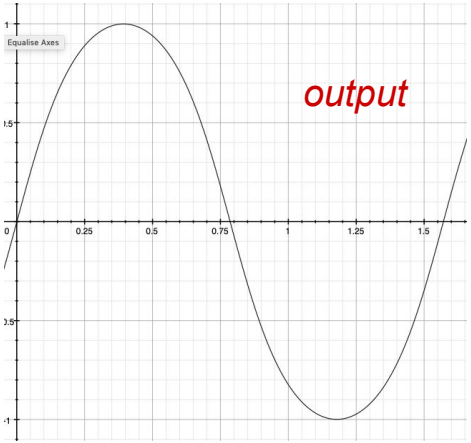
sample value ≈ 0.28

wavetable with arctan



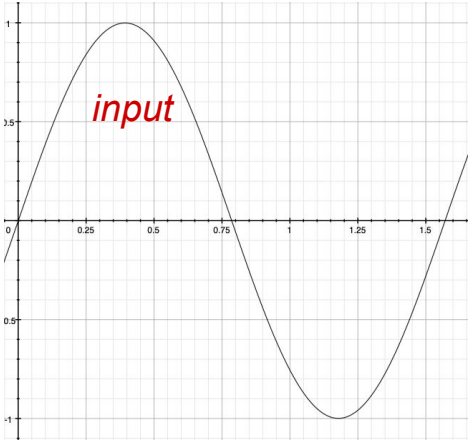


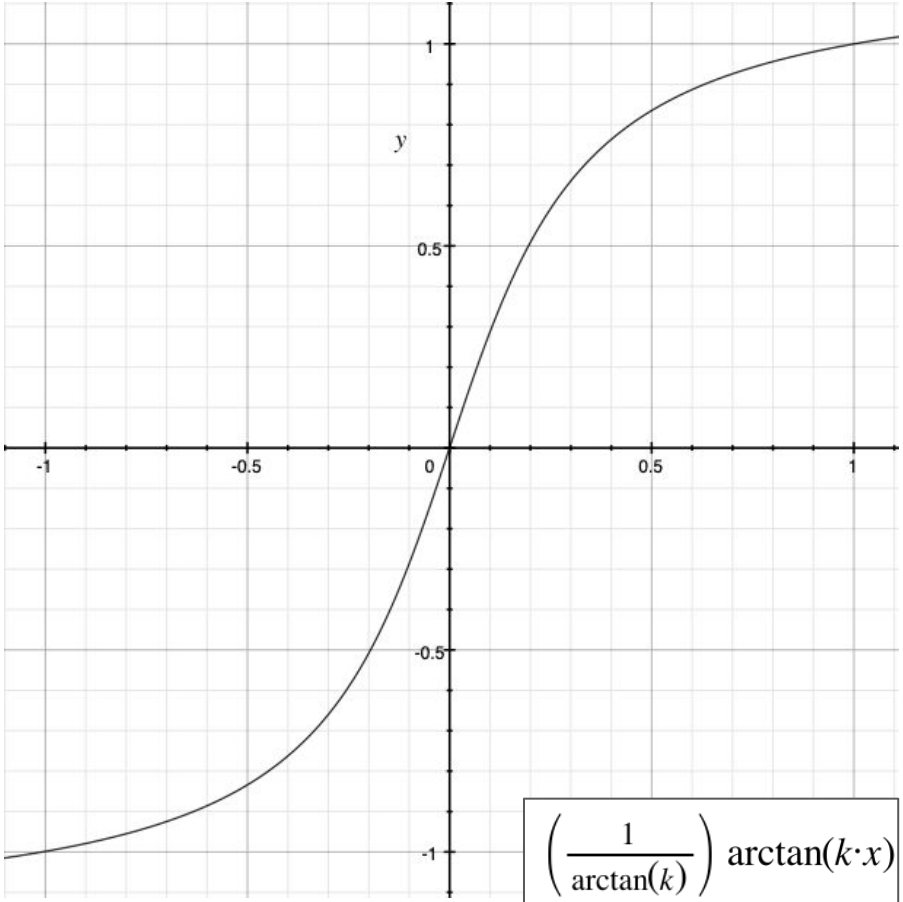
$$\left(\frac{1}{\arctan(k)}\right)\arctan(k\cdot x)$$



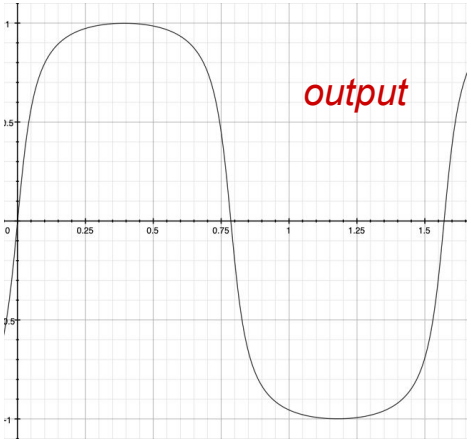
k = 1

wavetable with arctan

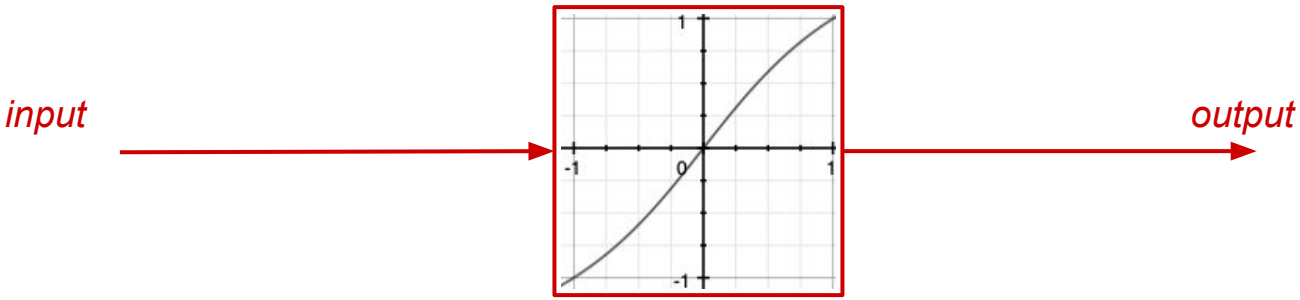




$$\left(\frac{1}{\arctan(k)}\right)\arctan(k \cdot x)$$



k = 4



Waveshaper - derived class

Waveshaper - derived class

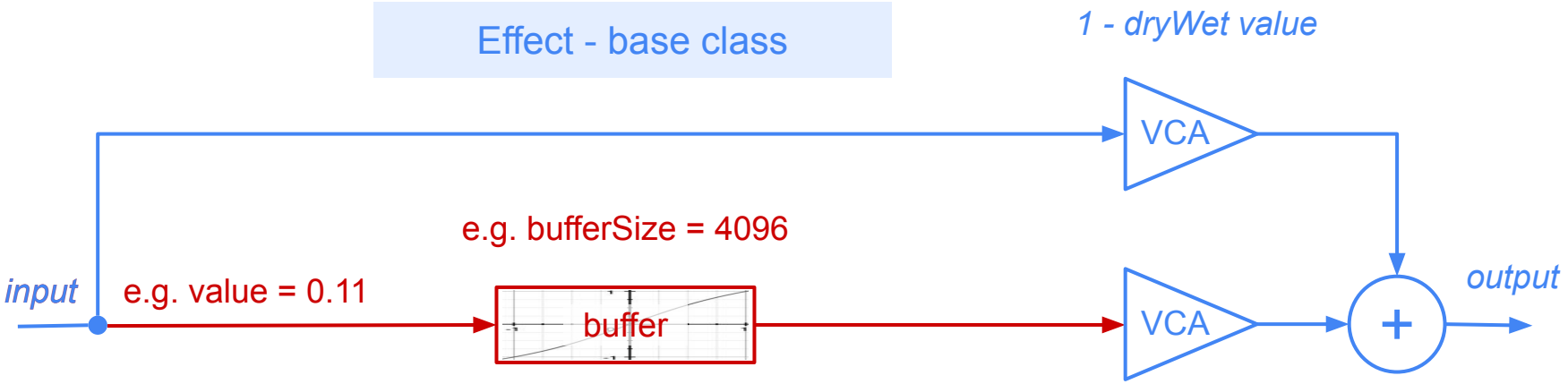

```
// map range [-1, 1] to [0, bufferSize - 1]
float index = (inputValue + 1.0) * 2048;
```

Waveshaper - derived class


```
// map range [-1, 1] to [0, bufferSize - 1]
float index = (inputValue + 1.0) * 2048;
// interpolate! index lies in between 2 indices
int i = (int) index;
float indexDecimal = index - (float)i;
outputValue = linMap(indexDecimal, buffer[i], buffer[i + 1]);
```

Waveshaper - derived class

Effect - base class

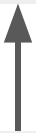



```
// map range [-1, 1] to [0, bufferSize - 1]
float index = (inputValue + 1.0) * 2048;
// interpolate! index lies in between 2 indices
int i = (int) index;
float indexDecimal = index - (float)i;
outputValue = linMap(indexDecimal, buffer[i], buffer[i + 1]);
```

Waveshaper - derived class

Effect

```
processFrame(float& input, float& output);
setDryWet(float dryWet);
setBypass(bool bypass);

…
```

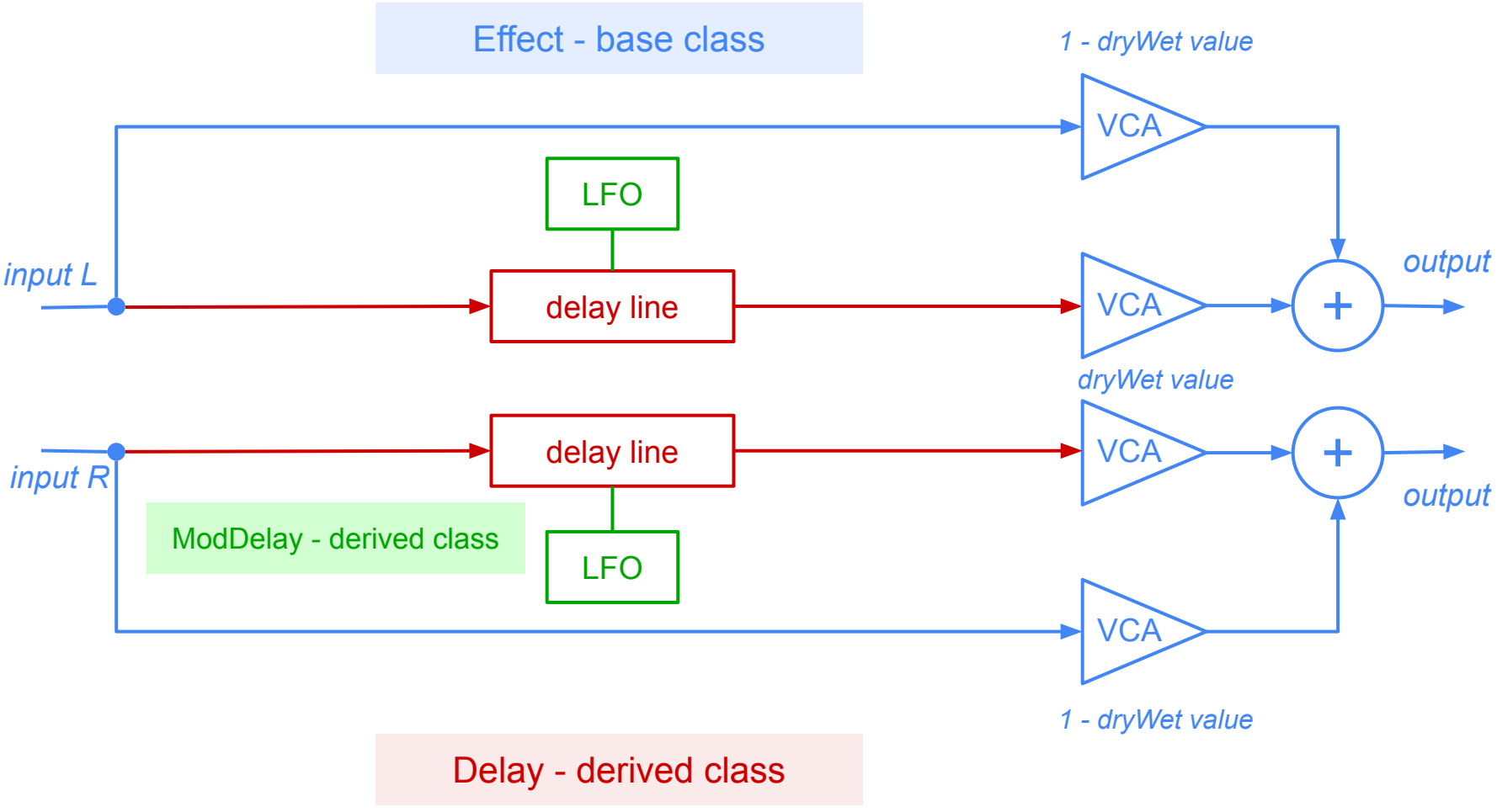


WaveShaper

```
applyEffect(float& input, float& output);

float[] buffer;
…
```

Chorus

derived from Delay


Effect - base class
1 - dryWet value
VCA
LFO
input L
delay line
VCA
output
dryWet value
input R
delay line
VCA
output
ModDelay - derived class
LFO
VCA
1 - dryWet value
Delay - derived class