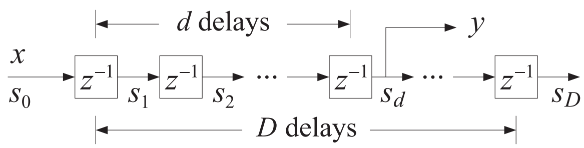
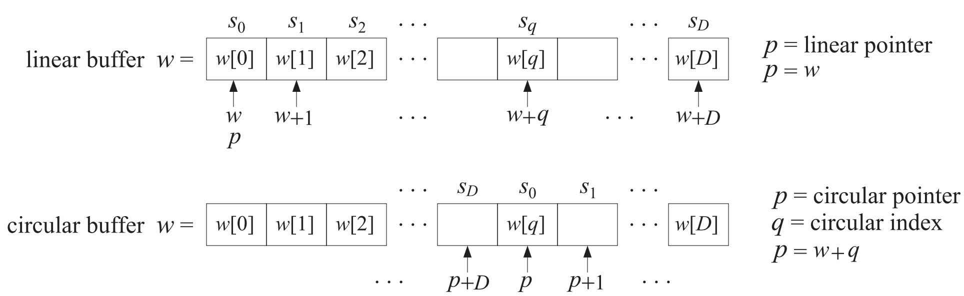
# *Summary of Delay-Based Effects*

## *Delays*




$$H(z) = z^{-d}$$

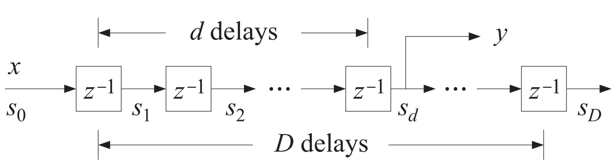
There are $D$ registers whose contents are the "internal" states of the delay line. The $d$th state $s_d$, i.e., the content of the $d$th register, represents the $d$-fold delayed version of the input, that is, at time $n$ we have: $y(n) = s_d(n) = x(n-d)$, for $d = 1, \dots, D$; the case $d = 0$ corresponds to the input $s_0(n) = x(n)$. The states $s_0, s_1, \dots, s_D$ are stored in memory in a $(D+1)$-dimensional array or buffer $w$. But the manner in which they are stored and retrieved depends on whether a linear or a circular buffer is used. The two cases are depicted below.




The circular pointer $p$ is related to the circular index $q$ by, $p = w + q$. In a MATLAB implementation, one can use the following anonymous function to wrap the circular index modulo-$(D+1)$,

```
Q = @(D,q) q + (D+1)*((q<0) - (q>D));     % substitute for qwrap.m/qwrap.c
```

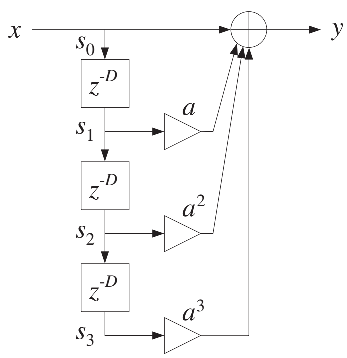
and $q$ is to be restricted to the range, $-1 \le q \le 2D$, which is sufficient for retrieving the states in all filtering operations. The sample processing algorithm of a $d$-fold delay $y(n) = x(n-d)$, expressed in terms of the pointer $p$, or the index $q$, is as follows,




| | |
|---|---|
| *for each x do:*<br>$y = *(p+d)$<br>$*p = x$<br>$p--$ | *for each x do:*<br>$y = w[Q(q+d)]$<br>$w[q] = x$<br>$q = Q(q-1)$ |

where in MATLAB, the array indices must be shifted by 1, that is, $w[i] \equiv w_{\text{MAT}}(i+1)$, $i = 0, 1, \dots, D$.

## *Comb Filters*

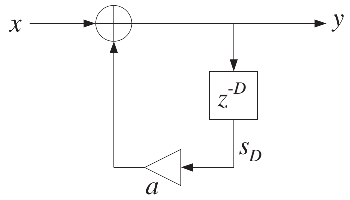



*for each input x do:*

$s_0 = x$
$s_1 = *(p+D)$
$s_2 = *(p+2D)$
$s_3 = *(p+3D)$
$y = s_0 + as_1 + a^2 s_2 + a^3 s_3$
$*p = s_0$
$p--$

$$H(z) = 1 + az^{-D} + a^2 z^{-2D} + a^3 z^{-3D}$$

## Plain Reverb

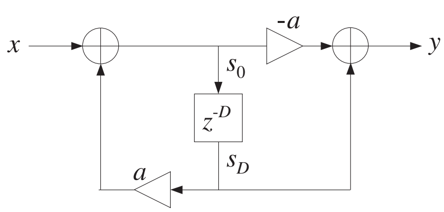



*for each input sample x do:*
$s_D = *(p+D)$
$y = x + as_D$
$*p = y$
$p--$

$$y(n) = ay(n-D) + x(n), \qquad H(z) = \frac{1}{1 - az^{-D}}$$

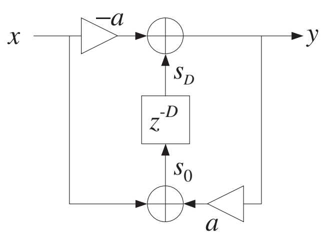
## Allpass Reverb

Canonical realization:




*for each input sample x do:*
$s_D = *(p+D)$
$s_0 = x + as_D$
$y = -as_0 + s_D$
$*p = s_0$
$p--$

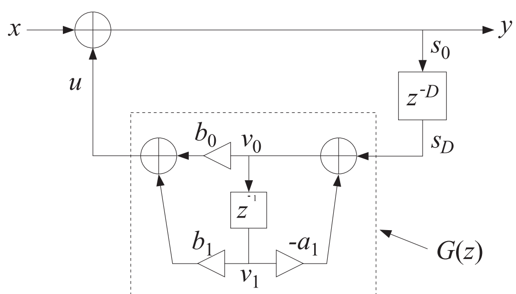
Transposed realization:




*for each input x do:*
$s_D = *(p+D)$
$y = s_D - ax$
$*p = s_0 = x + ay$
$p--$

$$H(z) = \frac{-a + z^{-D}}{1 - az^{-D}}$$

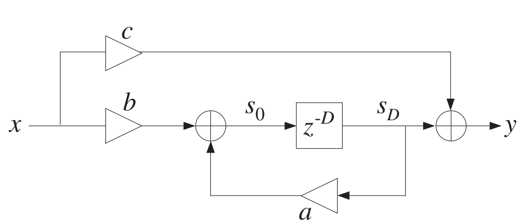
## Lowpass Reverb / Guitar Algorithm




*for each input sample x do:*
$s_D = *(p+D)$
$v_0 = -a_1 v_1 + s_D$
$u = b_0 v_0 + b_1 v_1$
$y = x + u$
$v_1 = v_0$
$*p = y$
$p--$

$$H(z) = \frac{1}{1 - z^{-D}G(z)}, \qquad G(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}}$$

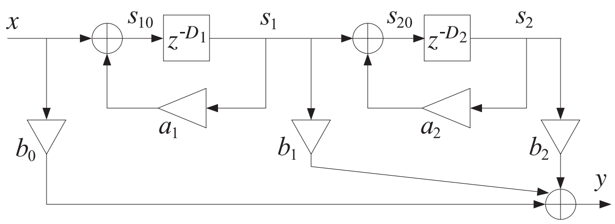
## Reverberating Delay




*for each input x do:*
$s_D = *(p+D)$
$y = cx + s_D$
$*p = s_0 = bx + as_D$
$p--$

$$H(z) = c + b\frac{z^{-D}}{1 - az^{-D}}$$

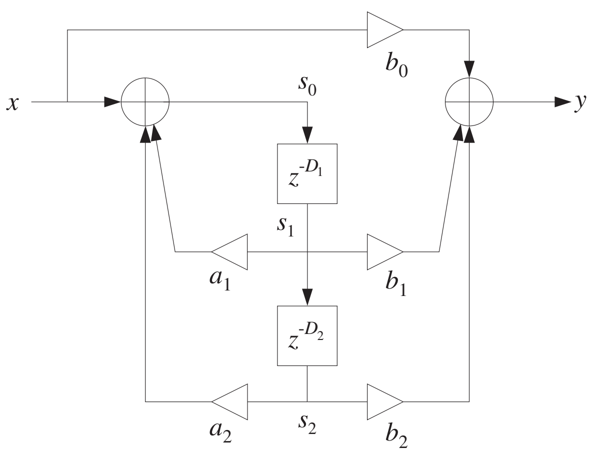
### *Multi-Delay Effects*




*for each input x do:*

$s_1 = *(p_1 + D_1)$
$s_2 = *(p_2 + D_2)$
$y = b_0 x + b_1 s_1 + b_2 s_2$
$*p_2 = s_{20} = s_1 + a_2 s_2$
$p_2{-}{-}$
$*p_1 = s_{10} = x + a_1 s_1$
$p_1{-}{-}$

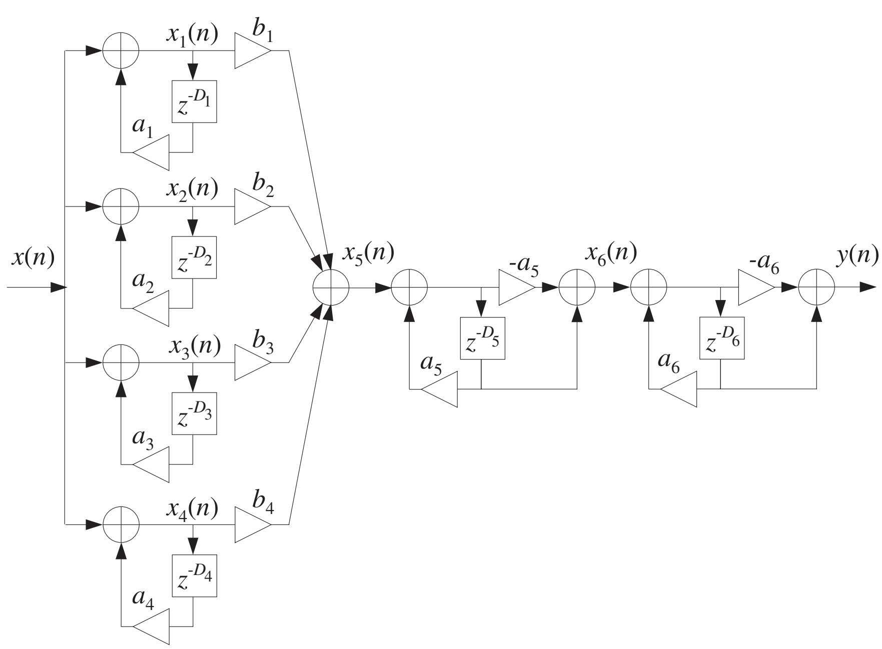
### *Multitap Delay Effects*




*for each input sample x do:*

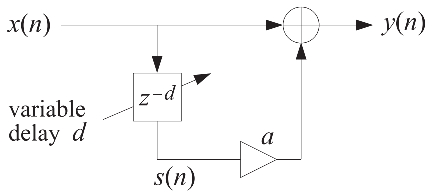
$s_1 = *(p + D_1)$
$s_2 = *(p + D_1 + D_2)$
$y = b_0 x + b_1 s_1 + b_2 s_2$
$s_0 = x + a_1 s_1 + a_2 s_2$
$*p = s_0$
$p{-}{-}$

### *Schroeder's Reverb Algorithm*




Its sample processing algorithm is built from four plain reverbs in parallel and two allpass reverbs in series, each requiring its own circular buffer and pointer.

### *Flanger*




$$y(n) = x(n) + a\,x\big(n - d(n)\big)$$

$$d(n) = \frac{D}{2}\left[1 - \cos(2\pi F_d n)\right]$$

where $F_d$ is in units of [cycles/sample]. The maximum delay $D$ corresponds typically to a few milliseconds, and the frequency $F_d$ to a couple of Hz. Here, the circular buffer must have length $(D+1)$, and the rounded version of $d(n)$ must be used in the sample processing algorithm,

*for each time sample n do:*
$s_d = *(p + d(n))$
$y(n) = x(n) + a\,s_d$
$*p = x(n)$
$p--$