# Application Note AN3101-11: Soft Mute

By Shultz Wang

## Introduction

Oftentimes, when switching between different programs in the DSP-1K, it is desirable to gently mute the output in order to avoid spurious noise, and then gently unmute after the program switchover is complete. This application note provides a basic soft mute algorithm to accomplish this.

## Algorithm

Soft muting is the changing of the output volume in a continuous exponential fashion, which avoids the abrupt on-off discontinuity of a hard mute. The soft mute algorithm is based on the smoothing equation, which is derived from the universal time constant formula for RC and RL circuits:

$$\text{Change} = (\text{Final-Start})[1-1/e^{(t/\tau)}].$$

To get the value at the next timestep, add the current value to both sides:

$$\text{Change} + \text{Start} = (\text{Final-Start})[1-1/e^{(t/\tau)}] + \text{Start}.$$

Rephrased,

Next = Current + k(Target-Current), where k is a fractional multiplier which determines the smoothing rate.

The time constant $\tau$ gives the amount of time it takes for the curve to rise 63% closer from its current value to its final value. Solving k for $\tau$:

$$k = 1 - 1/e^{(t/\tau)}$$

$$e^{(t/\tau)} = 1/(1-k)$$

$$t/\tau = \ln[1/(1-k)] = -\ln(1-k)$$

$\tau = -t/\ln(1-k)$, where t is one sample period.

The basic smoothing equation, applied as an output gain, generates the soft mute. The soft mute code may be placed at the end of the program space, and thus not interfere with any changes in the instructions above when program reloads occur. If the target value is set to a small non-zero number, this algorithm may be used to implement a soft duck as opposed to a mute.

The following table can serve as a guide for selecting the proper fractional value for good time constants (sampling period = 1/48kHz).

| Shift | Fractional | Hex | τ = -(Sample Pd)/ln(1-Fractional) |
|---|---|---|---|
| 1 bit | 1/2 | $0800000 | 0.030ms |
| 2 bits | 1/4 | $0400000 | 0.072ms |
| 3 bits | 1/8 | $0200000 | 0.156ms |
| 4 bits | 1/16 | $0100000 | 0.323ms |
| 5 bits | 1/32 | $0080000 | 0.656ms |
| 6 bits | 1/64 | $0040000 | 1.323ms |
| 7 bits | 1/128 | $0020000 | 2.656ms |
| 8 bits | 1/256 | $0010000 | 5.323ms |
| 9 bits | 1/512 | $0008000 | 10.656ms |
| 10 bits | 1/1024 | $0004000 | 21.323ms |
| 11 bits | 1/2048 | $0002000 | 42.656ms |
| 12 bits | 1/4096 | $0001000 | 85.323ms |
| 13 bits | 1/8192 | $0000800 | 0.170s |
| 14 bits | 1/16384 | $0000400 | 0.341s |
| 15 bits | 1/32768 | $0000200 | 0.682s |
| 16 bits | 1/65536 | $0000100 | 1.365s |
| 17 bits | 1/131072 | $0000080 | 2.731s |
| 18 bits | 1/262144 | $0000040 | 5.461s |
| 19 bits | 1/524288 | $0000020 | 10.923s |
| 20 bits | 1/1048576 | $0000010 | 21.845s |

For testing the soft mute, the 1KLoader program may be used to load the General Purpose Registers with the required Target and k values.

## Source Code

```
; Program code goes above

; $40C: Target value
; $40D: Fractional constant k
; $40E: Current value
; $40F: Temporary output storage

SCA    0      $40F   ; Store Output in $403

CM     1.0    $40C   ; Read Target
LCMA   -1.0   $40E   ; Acc = Target-Current, B = Current
AMB           $40D   ; Acc = (Target-Current)k + Current = Smoothing coefficient
SCA    1.0    $40E   ; Save smoothing coefficient

AMC    0      $40F   ; Acc = Smoothing coefficient * Output
SCA    0      $410   ; Output to channel 0
```

NOTICE



**Contact Information:**

Wavefront Semiconductor
200 Scenic View Drive
Cumberland, RI 02864 U.S.A.
Tel: +1 401 658-3670
Fax: +1 401 658-3680
On the web at www.wavefrontsemi.com
Email: info@wavefrontsemi.com



Application note revised March, 2005