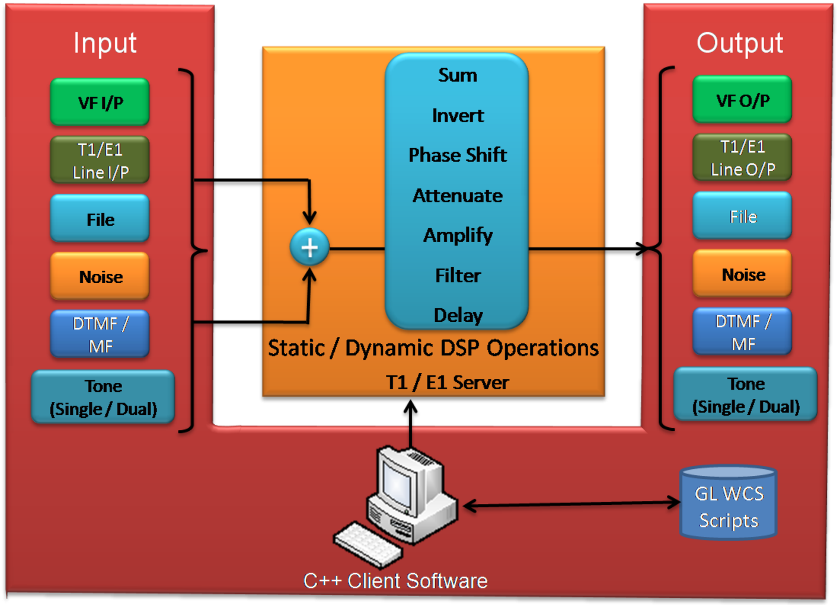
Static and Dynamic DSP Operations

GL Communications Inc.

818 West Diamond Avenue - Third Floor, Gaithersburg, MD 20878
Phone: (301) 670-4784 Fax: (301) 670-9187 Email: info@gl.com
Website: https://www.gl.com

Digital Signal Processing (DSP)



- DSP capability in Windows Client Server (WCS) is categorized into **Static Operators** and **Dynamic Operators**
- **Static Operators** - provide the ability to specify a sequence of digital signal processing steps to be performed on incoming and/or outgoing timeslots
- **Dynamic Operators** - perform dynamic or time-varying operations via schedules, which specifies a sequence of digital signal processing steps to be performed at specified time offset for each operator on incoming and/or outgoing signals

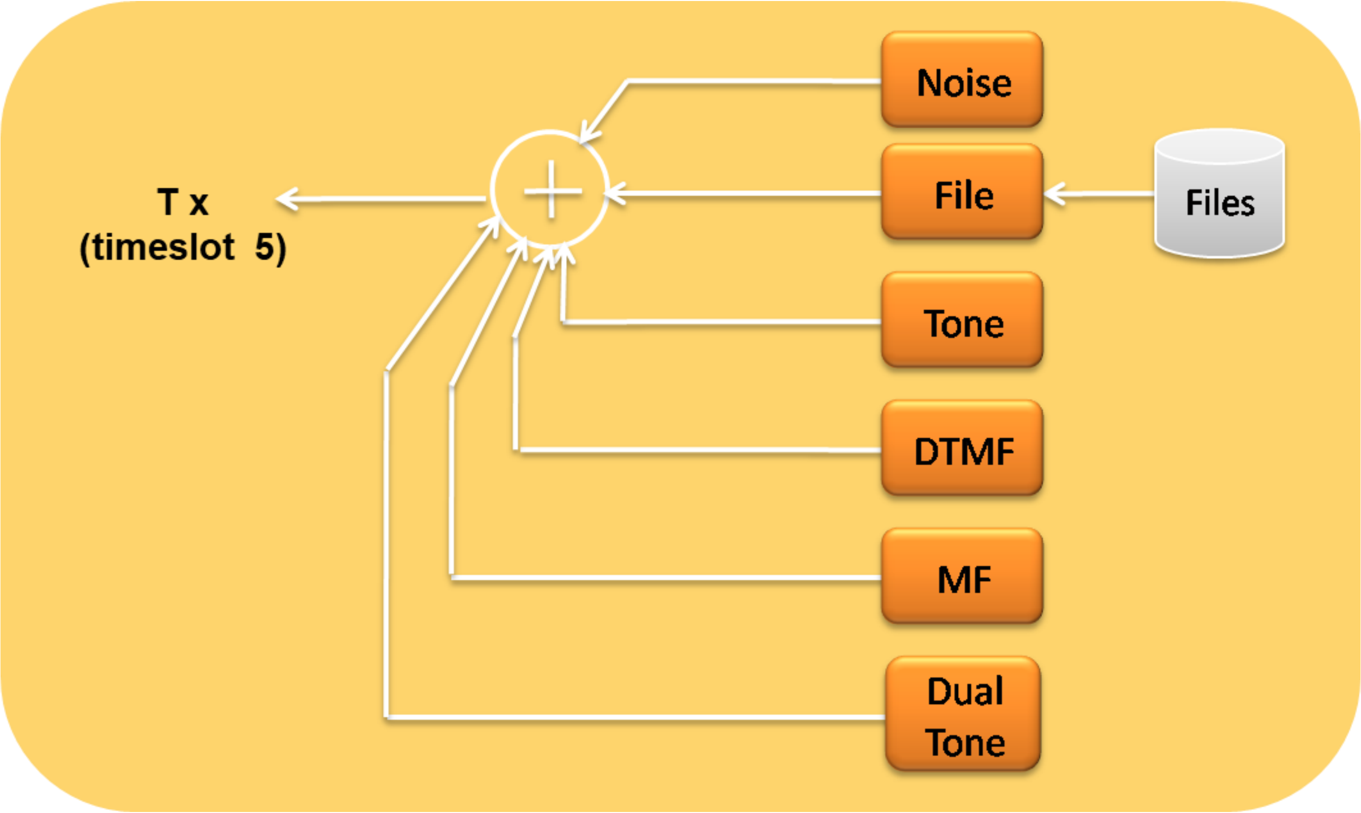
Static DSP Operations

- **Following functions can be performed using Static DSP operators:**
  - Sum
  - Invert
  - Filter
  - Delay
  - Amplify
  - Attenuate
  - Bxor, bor, brev, bnot, band
  - Infile, outfile
  - White noise, tone, dual tone, phase shift, dtmf digits, mf digits, mfcr2 digits
  - Power monitor, signaling bits monitor, const, bytes and many others

Static DSP Operations

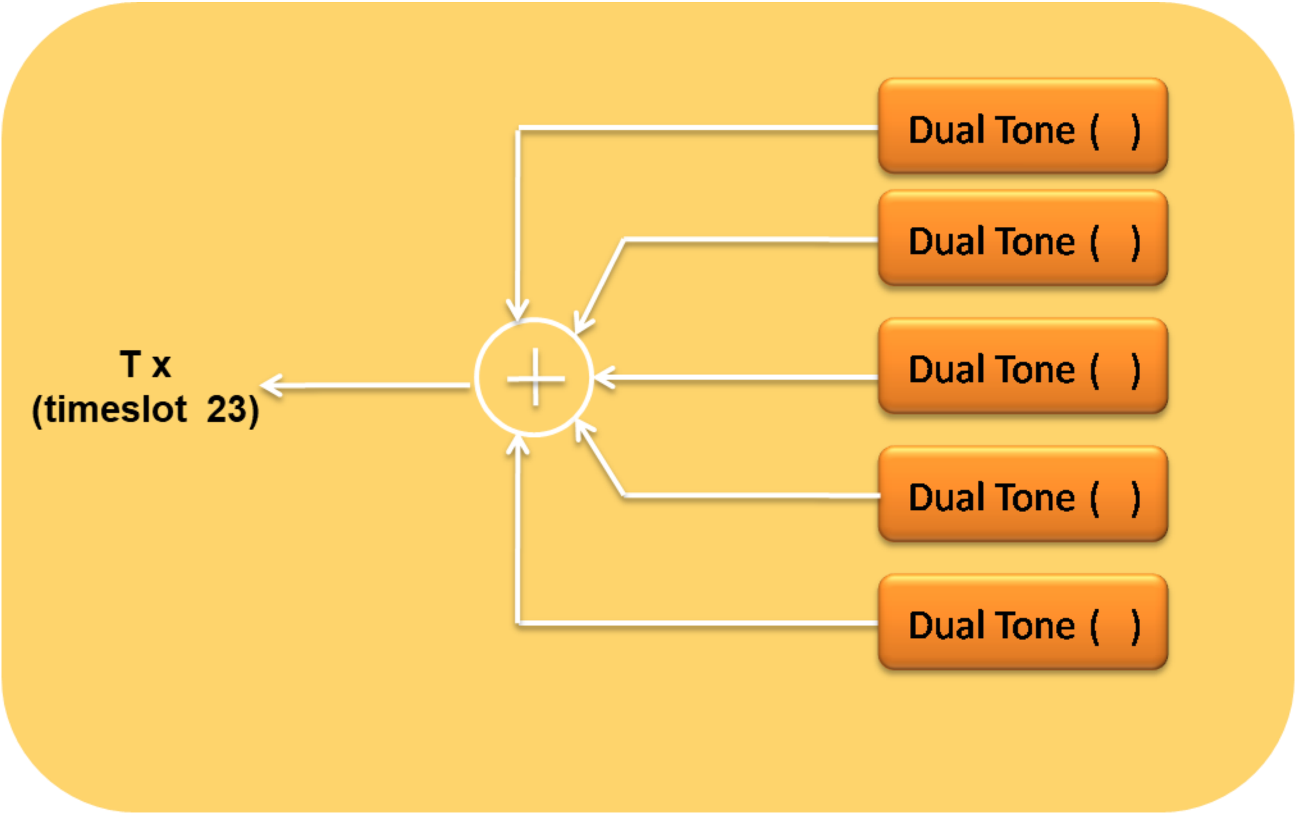
- Basic Static Operations for Echo Paths Simulation
  - Sum digitally synthesized sources
  - Sum multiple dual tone generators
  - Sum signal with delayed and attenuated version of itself
  - Parallel echo paths summed with digitally synthesized tone / noise / file
  - Sum signal with inverted version of itself
- Static Operations using C++ Client
  - Transmit filtered tones and white noise
  - Adding speech and noise to the receive data
  - Adding noise and phase shift tone to the speech data
  - Testing Arithmetical Functions on Incoming Bit Stream
  - Double talk simulation for echo canceller testing

Echo Paths Simulated using Functions: SUM



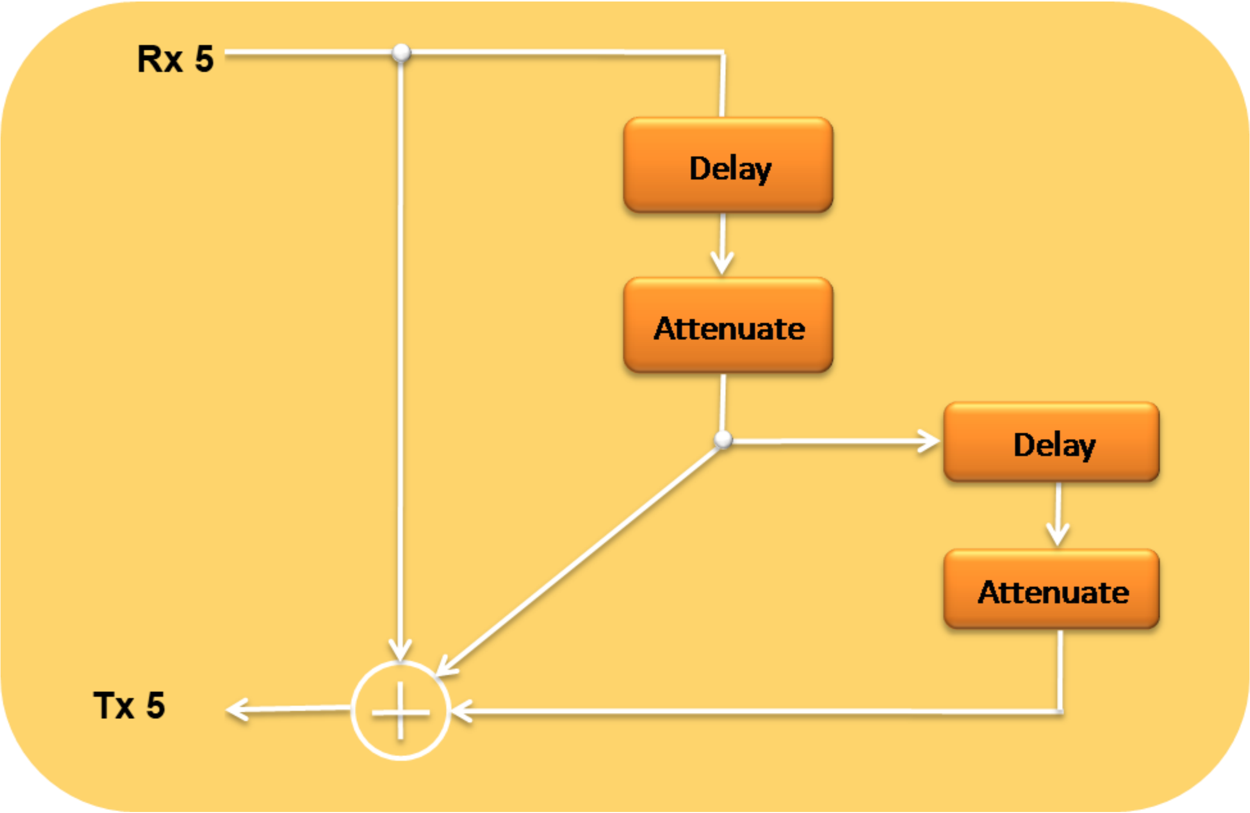
- Digitally synthesized generators of tone, noise, DTMF digits, MF digits, and dual tone are summed and transmitted into timeslot

Echo Paths Simulated using Functions: SUM



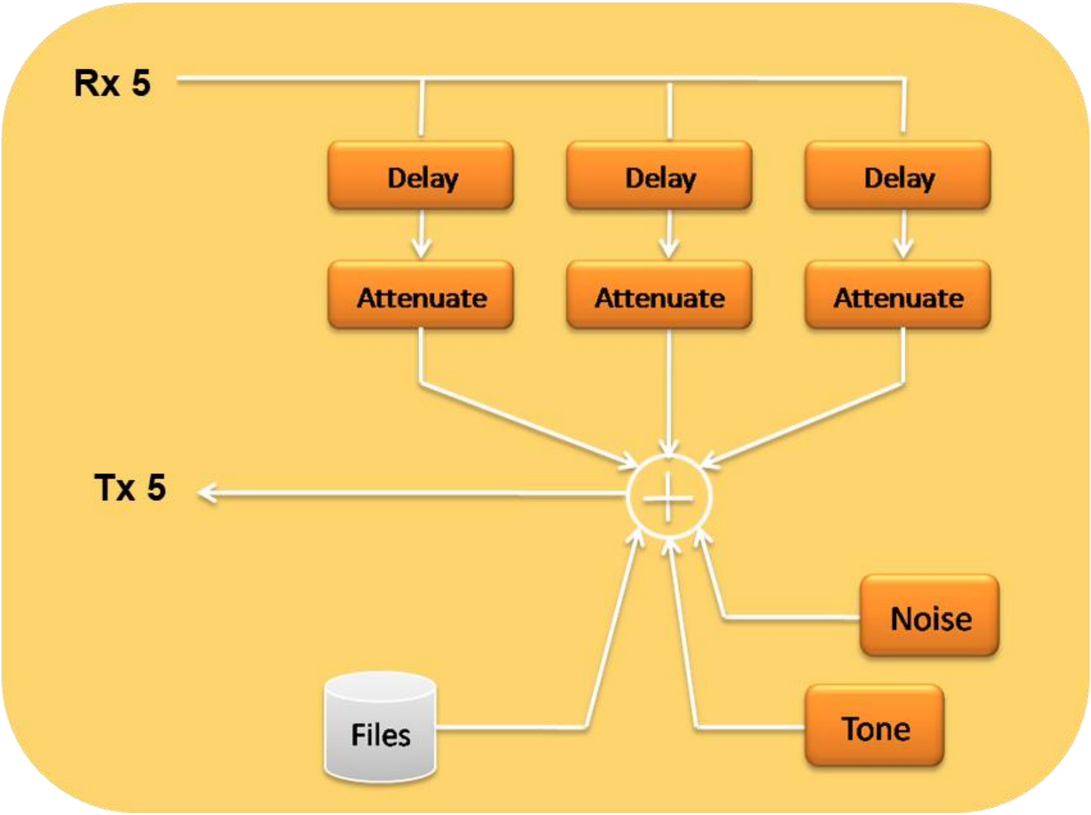
- Multiple dual tone generators with possibly different parameters are summed and transmitted into timeslot

Structure for Echo Path Modeling and Testing EC



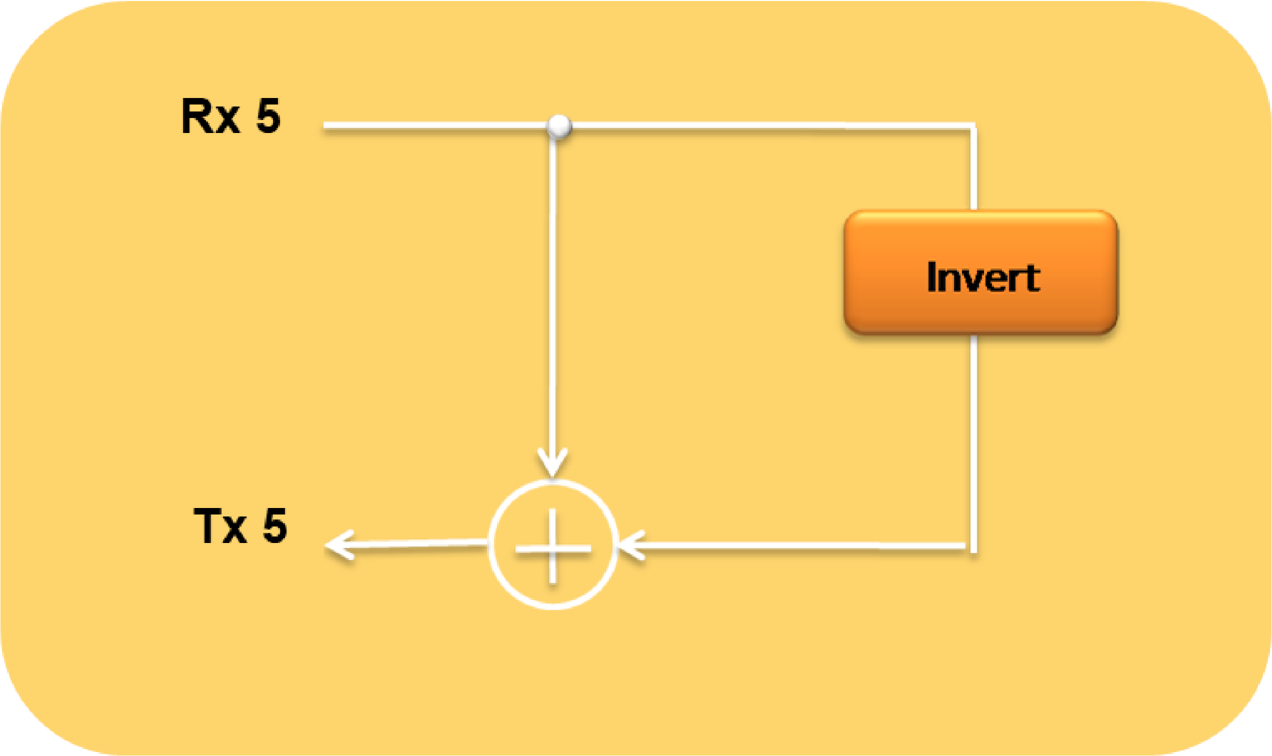
- Receive timeslot is summed with delayed and attenuated versions of itself and transmitted back

Sum and Attenuate Operators



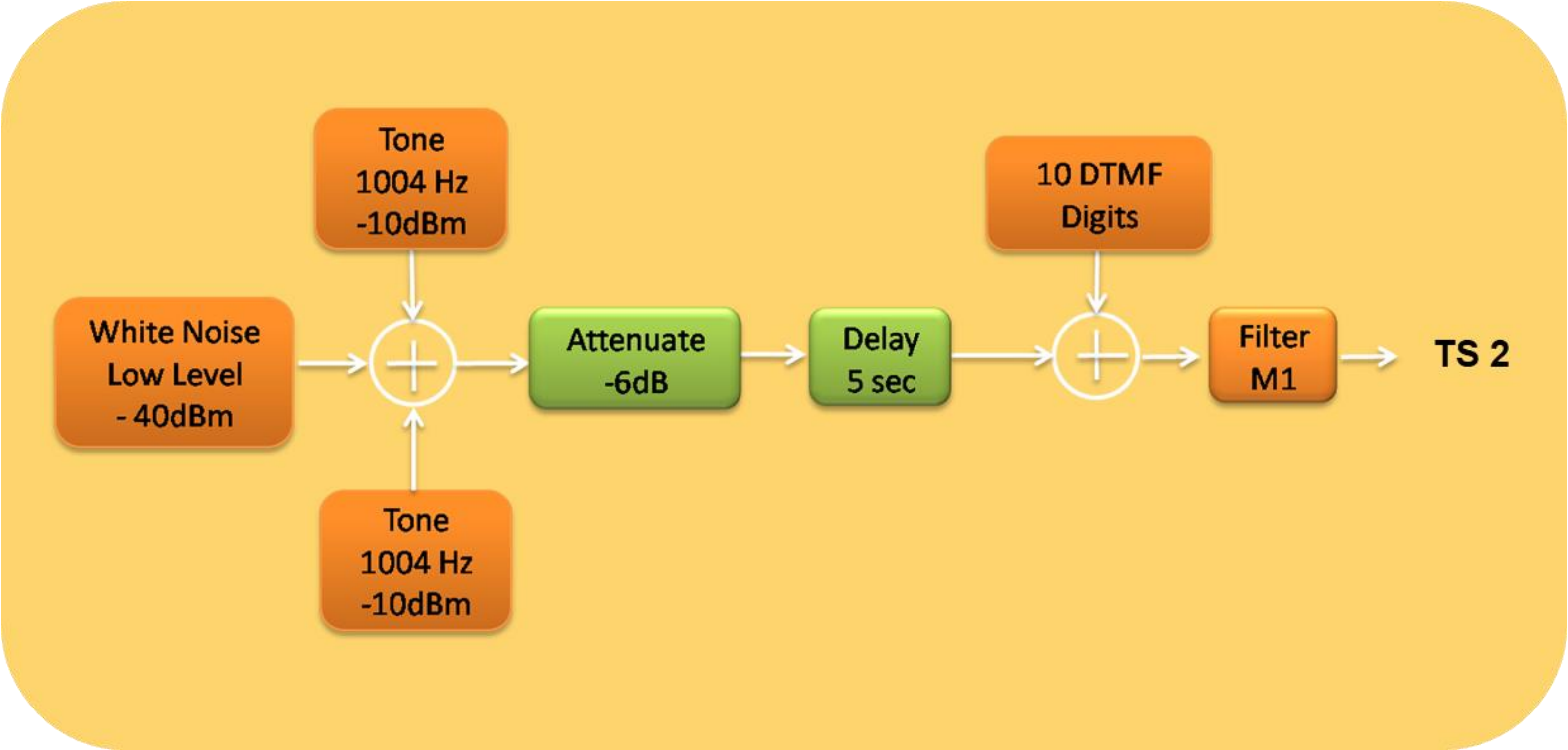
- Three parallel echo paths are summed with a digitally synthesized tone and noise and a PCM file, a more complex structure for echo path modeling

Invert and Sum Operators



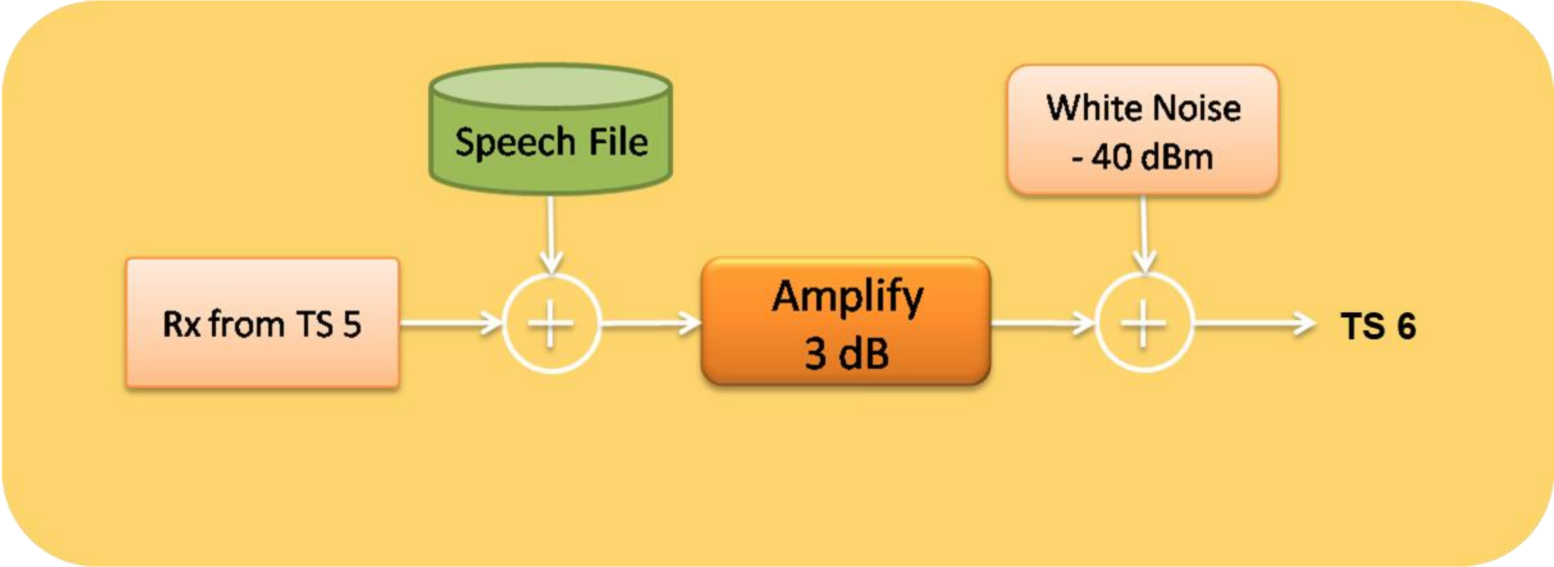
- Receive timeslot 5 is inverted, summed with itself, and transmitted into timeslot 5. This is an example of a perfect canceller

Transmit Filtered Tones and White Noise



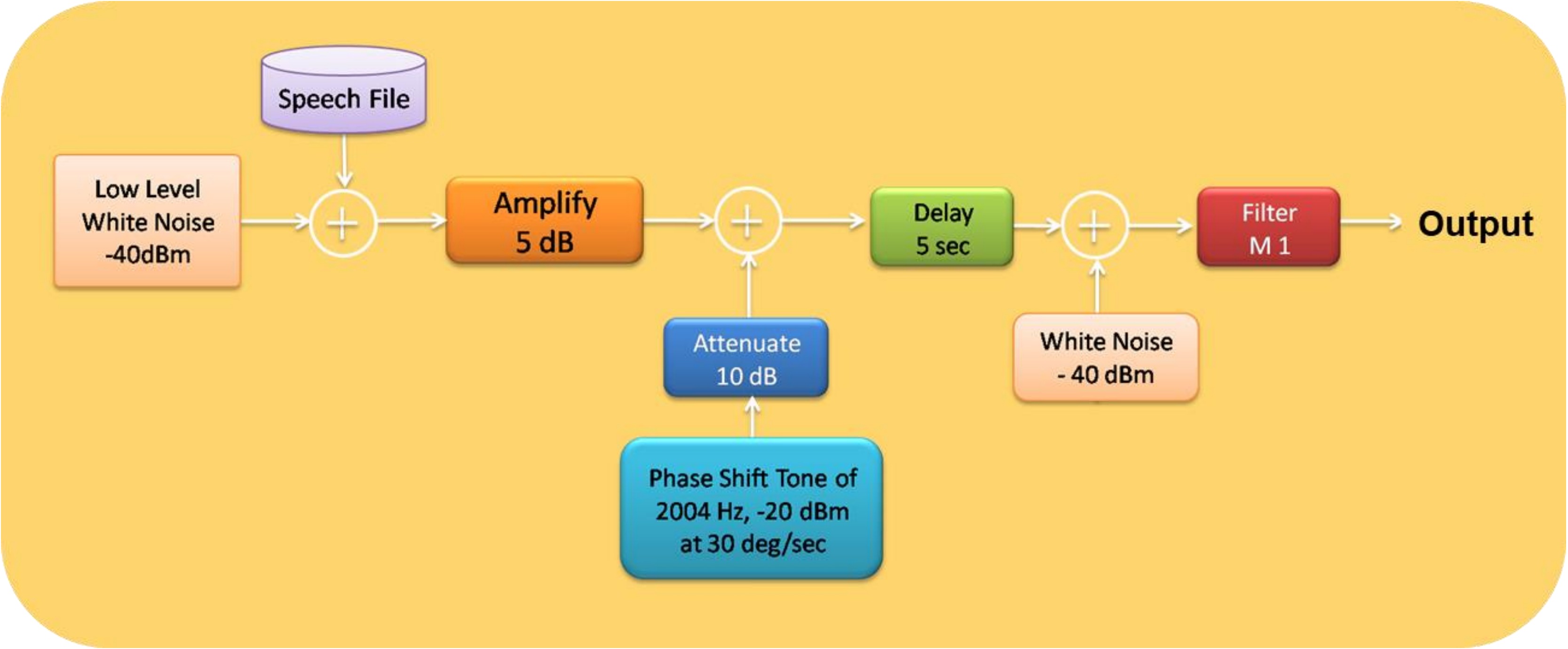
- Script used for the operation transmits two tones at different frequencies along with white noise and DTMF digits. The output is obtained through a filter

Adding Speech and Noise to the Receive Data



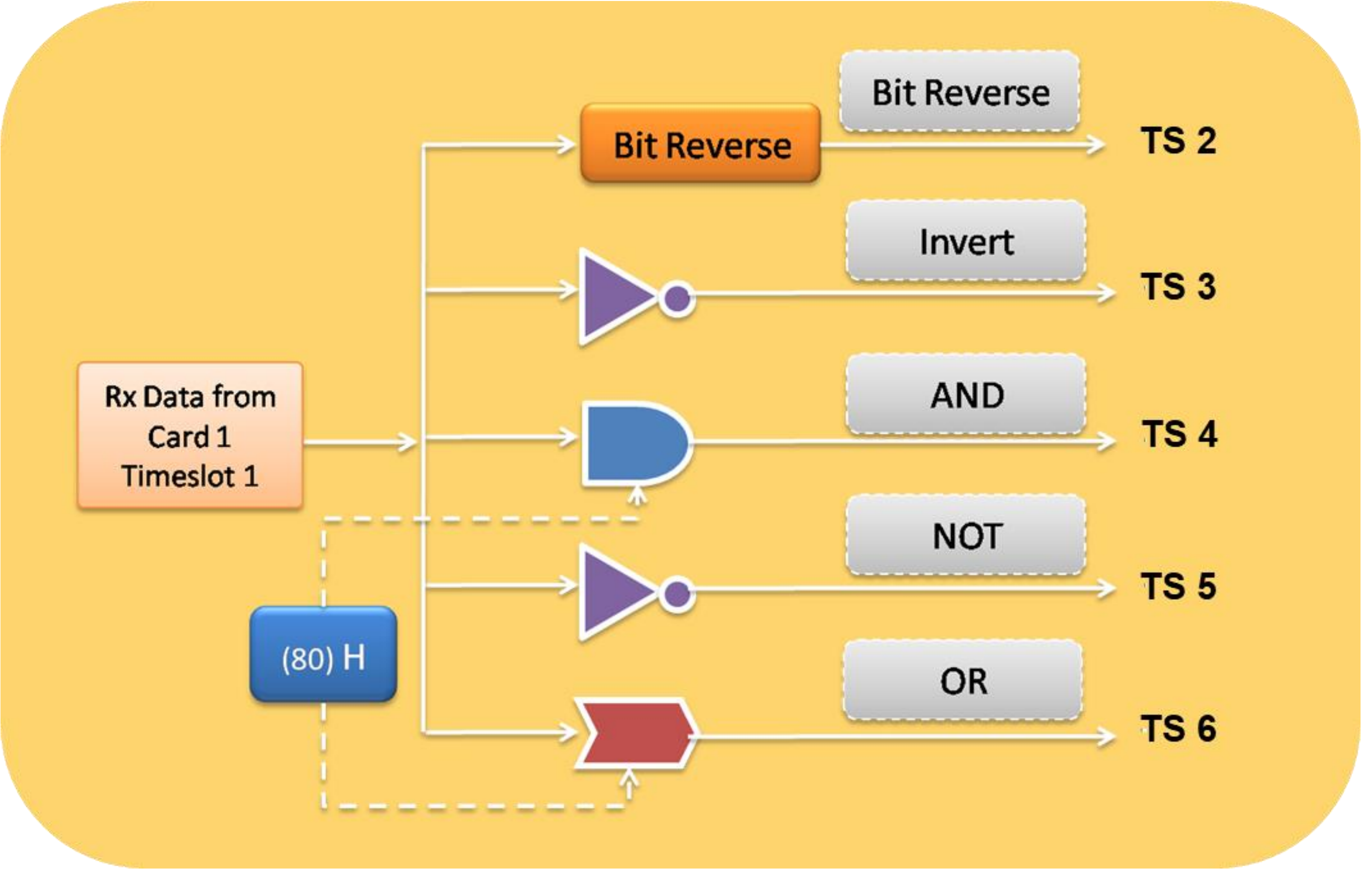
- Script used for the operation transmits the amplified speech file with white noise and the data received on a specified timeslot

Adding Noise and Phase-shift Tone to Speech



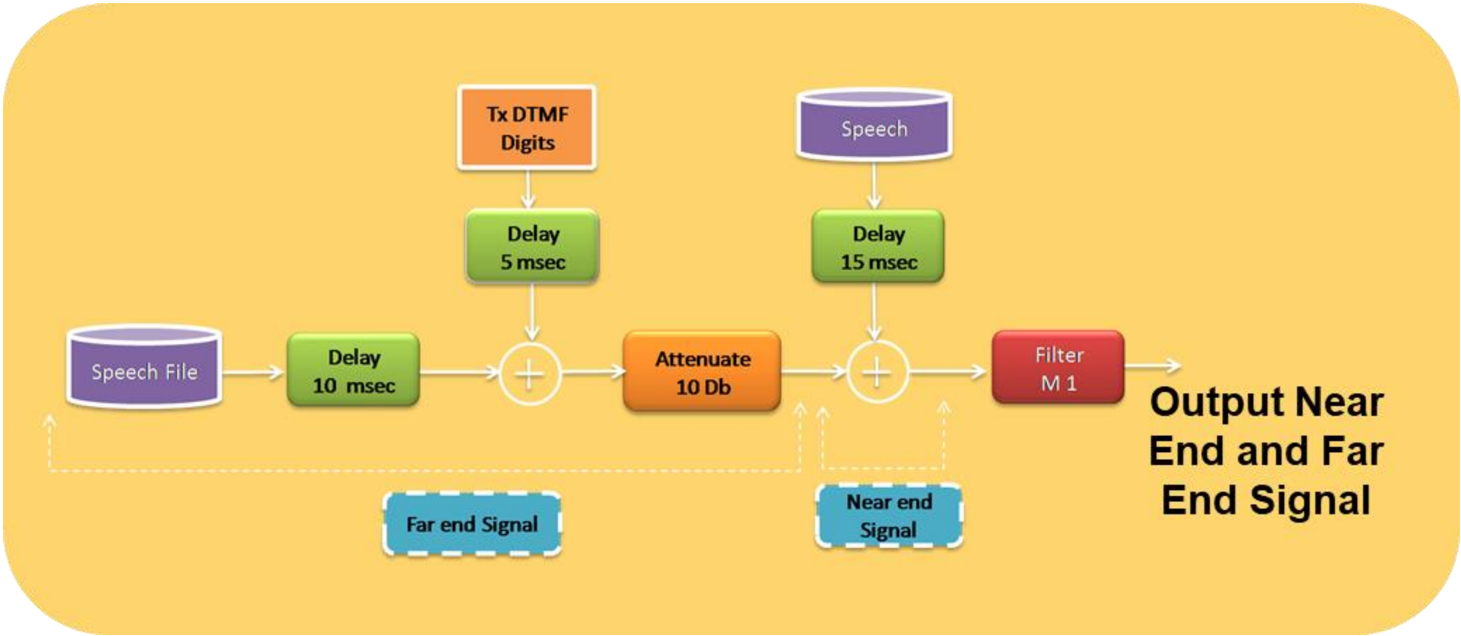
- Script used for the operation transmits an amplified speech file with a continuously phase shifted tone combined with a white noise through a filter

Testing Arithmetical Functions on Incoming Bit Stream



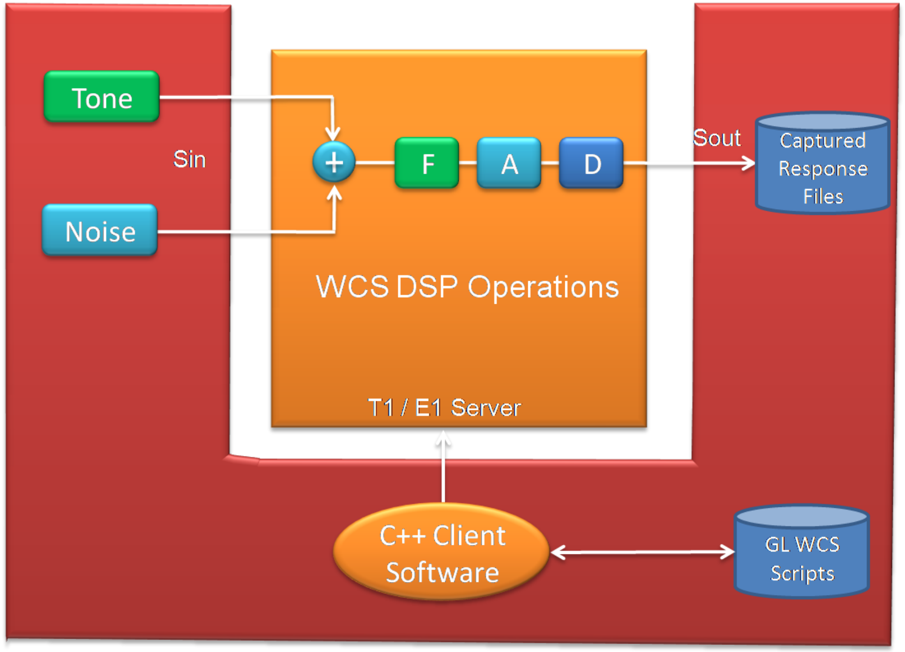
- Script used for the operation to perform various types of arithmetical functions on the incoming bit stream

Double-talk Simulation for Echo Canceller Testing



- Script used for the operation emulates the far-end and near-end call for echo canceller testing

Dynamic Digital Signal Processing (DSP)



- Scripted DSP commands provide the ability to specify a sequence of digital signal processing steps to be performed on incoming and/or outgoing timeslots
  - ➢ The operations can be made dynamic or time-varying via schedules
  - ➢ Schedules are categorized into **Time, Operators, Transition**, and **Value**

Dynamic DSP Operations

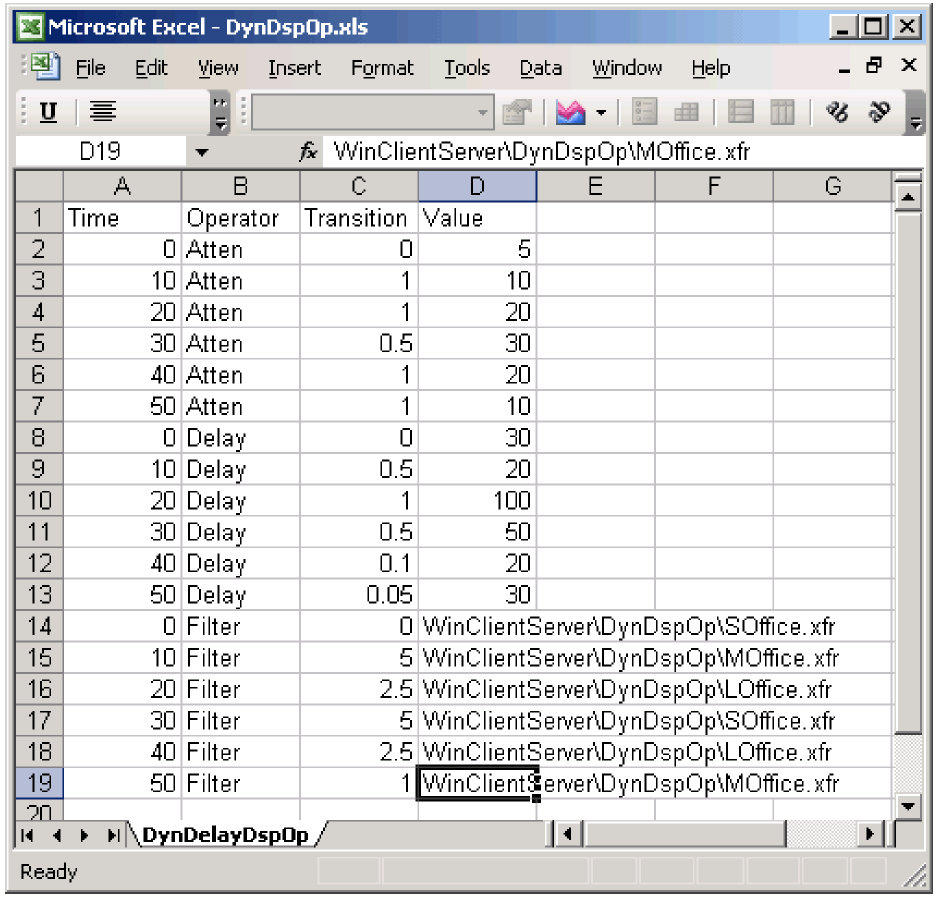
Offline Dynamic DSP Operations

- Amplify ("AmplifyDspOp" - dynamic amplification)
- Attenuate ("AttenDspOp " - dynamic attenuation)
- Delay ("DelayDspOp" - dynamic delay)
- Filter ("FiltDspOp" - dynamic filter models)

Real-time Dynamic DSP Operations

- Delay / Attenuate ("AttenDspOp " and "DelayDspOp")
- Filter ("FiltDspOp")

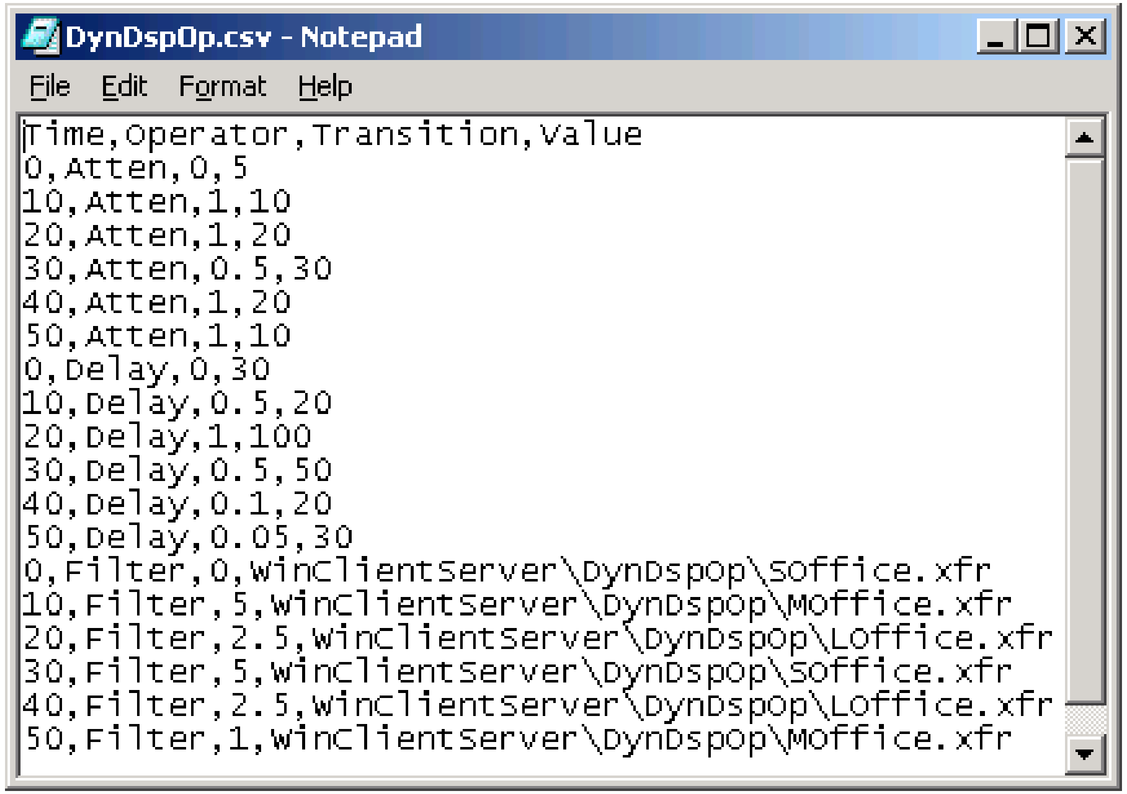
DSP Operations Schedule in Microsoft Excel


Microsoft Excel - DynDspOp.xls
File
Edit
View
Insert
Format
Tools
Data
Window
Help
D19
fx
WinClientServer\DynDspOp\MOffice.xfr
DynDelayDspOp
Ready

Time	Operator	Transition	Value
0	Atten	0	5
10	Atten	1	10
20	Atten	1	20
30	Atten	0.5	30
40	Atten	1	20
50	Atten	1	10
0	Delay	0	30
10	Delay	0.5	20
20	Delay	1	100
30	Delay	0.5	50
40	Delay	0.1	20
50	Delay	0.05	30
0	Filter	0	WinClientServer\DynDspOp\SOffice.xfr
10	Filter	5	WinClientServer\DynDspOp\MOffice.xfr
20	Filter	2.5	WinClientServer\DynDspOp\LOffice.xfr
30	Filter	5	WinClientServer\DynDspOp\SOffice.xfr
40	Filter	2.5	WinClientServer\DynDspOp\LOffice.xfr
50	Filter	1	WinClientServer\DynDspOp\MOffice.xfr

- Schedule is a file with a sequence of settings to be performed at specified run time offsets for each DSP operator
- Time, Operator, Transition, and Values are the columns in the schedule file

DSP Operations Schedule in CSV Format (DynDspOp.csv)

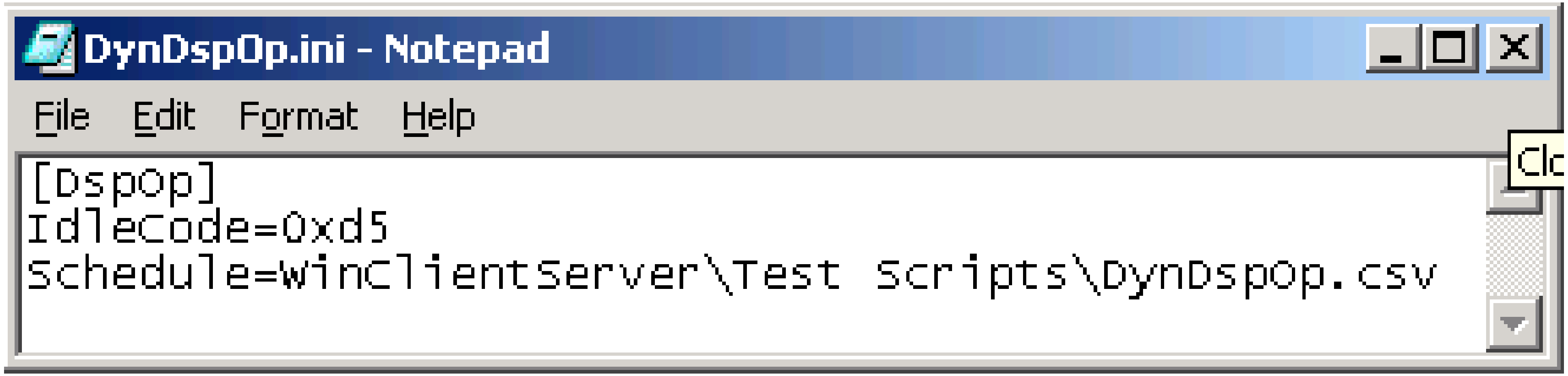


```
Time,Operator,Transition,Value
0,Atten,0,5
10,Atten,1,10
20,Atten,1,20
30,Atten,0.5,30
40,Atten,1,20
50,Atten,1,10
0,Delay,0,30
10,Delay,0.5,20
20,Delay,1,100
30,Delay,0.5,50
40,Delay,0.1,20
50,Delay,0.05,30
0,Filter,0,WinClientServer\DynDspOp\SOffice.xfr
10,Filter,5,WinClientServer\DynDspOp\MOffice.xfr
20,Filter,2.5,WinClientServer\DynDspOp\LOffice.xfr
30,Filter,5,WinClientServer\DynDspOp\SOffice.xfr
40,Filter,2.5,WinClientServer\DynDspOp\LOffice.xfr
50,Filter,1,WinClientServer\DynDspOp\MOffice.xfr
```

- Schedule must be saved in a CSV file format, in order to be used in a dynamic DSP operation

  Ex: DynDspOp.csv

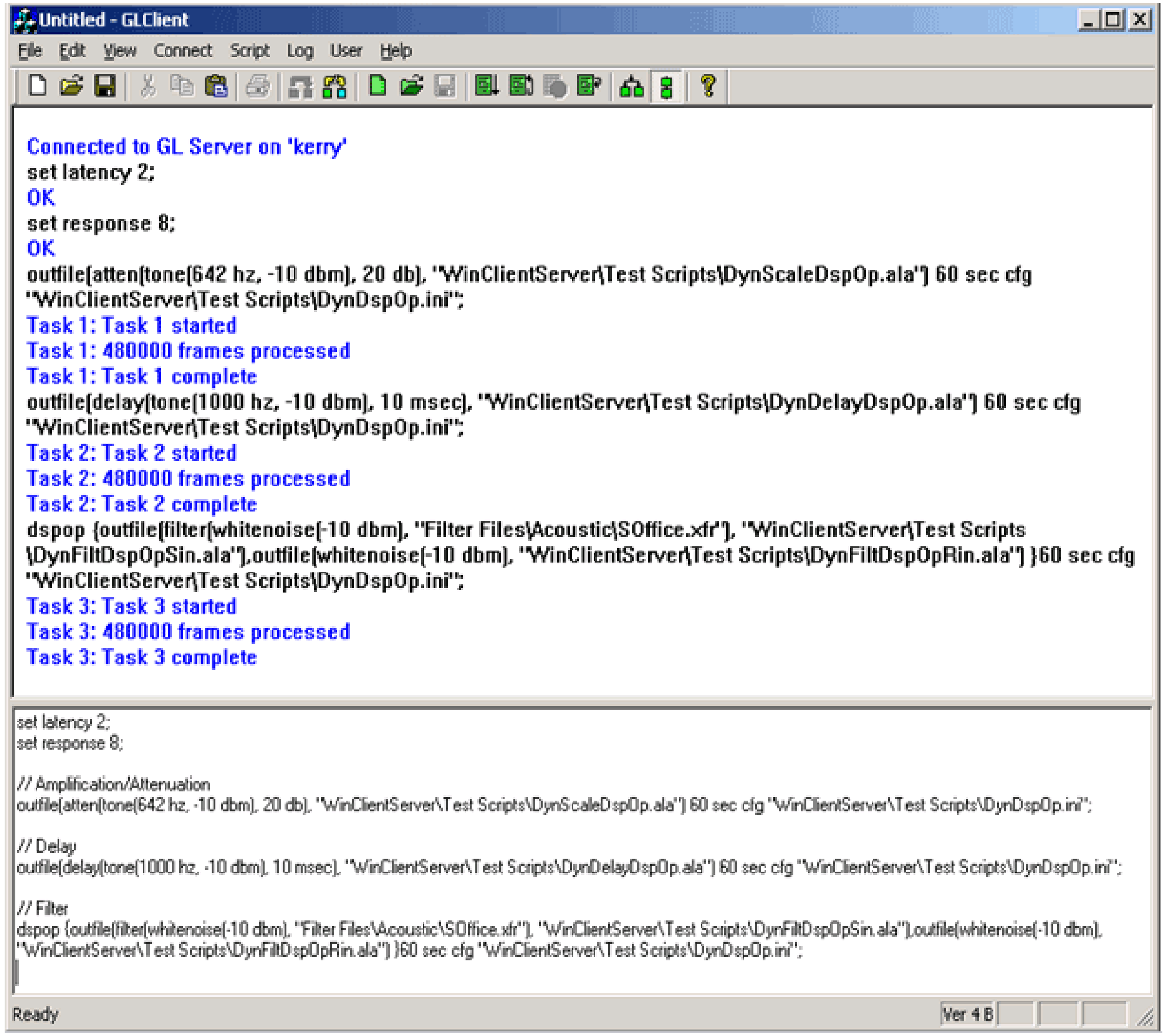
Invoking Schedule Via Configuration File



```
[DspOp]
IdleCode=0xd5
Schedule=WinClientServer\Test Scripts\DynDspOp.csv
```

- DSP operator or DSP operation obtains Schedule via a configuration file (*.ini)
- Configuration file invokes the schedule through a "schedule=" entry

Script invoking Schedule via Configuration File

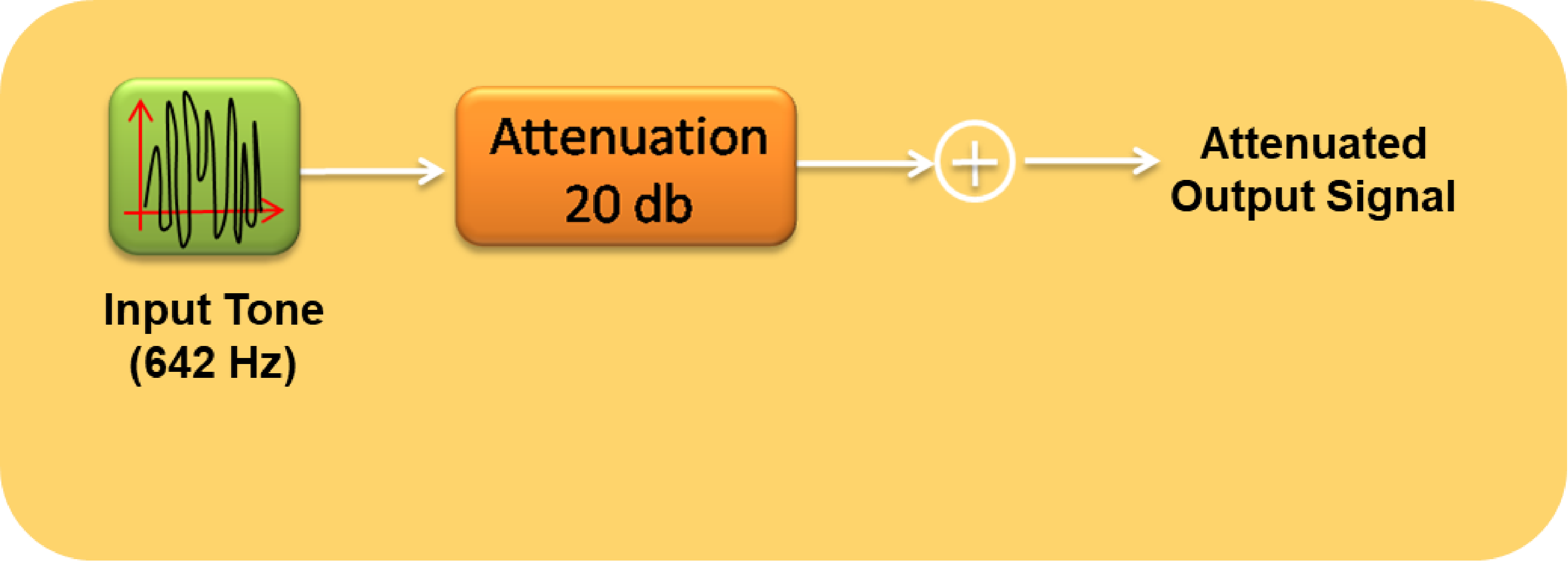


```
Connected to GL Server on 'kerry'
set latency 2;
OK
set response 8;
OK
outfile(atten(tone(642 hz, -10 dbm), 20 db), "WinClientServer\Test Scripts\DynScaleDspOp.ala") 60 sec cfg
"WinClientServer\Test Scripts\DynDspOp.ini";
Task 1: Task 1 started
Task 1: 480000 frames processed
Task 1: Task 1 complete
outfile(delay(tone(1000 hz, -10 dbm), 10 msec), "WinClientServer\Test Scripts\DynDelayDspOp.ala") 60 sec cfg
"WinClientServer\Test Scripts\DynDspOp.ini";
Task 2: Task 2 started
Task 2: 480000 frames processed
Task 2: Task 2 complete
dspop {outfile(filter(whitenoise(-10 dbm), "Filter Files\Acoustic\SOffice.xfr"), "WinClientServer\Test Scripts
\DynFiltDspOpSin.ala"),outfile(whitenoise(-10 dbm), "WinClientServer\Test Scripts\DynFiltDspOpRin.ala") }60 sec cfg
"WinClientServer\Test Scripts\DynDspOp.ini";
Task 3: Task 3 started
Task 3: 480000 frames processed
Task 3: Task 3 complete
```

```
set latency 2;
set response 8;

// Amplification/Attenuation
outfile(atten(tone(642 hz, -10 dbm), 20 db), "WinClientServer\Test Scripts\DynScaleDspOp.ala") 60 sec cfg "WinClientServer\Test Scripts\DynDspOp.ini";

// Delay
outfile(delay(tone(1000 hz, -10 dbm), 10 msec), "WinClientServer\Test Scripts\DynDelayDspOp.ala") 60 sec cfg "WinClientServer\Test Scripts\DynDspOp.ini";

// Filter
dspop {outfile(filter(whitenoise(-10 dbm), "Filter Files\Acoustic\SOffice.xfr"), "WinClientServer\Test Scripts\DynFiltDspOpSin.ala"),outfile(whitenoise(-10 dbm),
"WinClientServer\Test Scripts\DynFiltDspOpRin.ala") }60 sec cfg "WinClientServer\Test Scripts\DynDspOp.ini";
```

Dynamic Offline Attenuation ("AttenDspOp")

Attenuation of the tone input



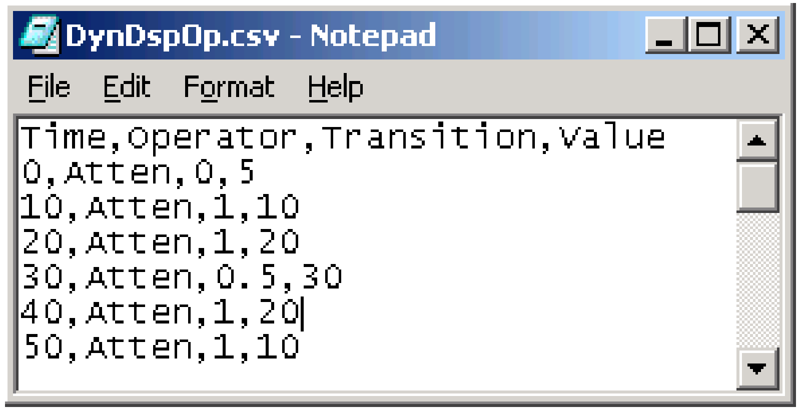
- Example script demonstrates attenuation of input tone (642 Hz) by 20 db to obtain attenuated output signal

Offline AttenDspOp Testing ("attenuate")

AttenDspOp WCS Test Script

```
set latency 4;
set response 6;

// (1) Amplification/Attenuation
outfile(atten(tone(642 hz, -10 dbm), 20 db),  "WinClientServer\DynDspOp\
DynScaleDspOp.ala") 60 sec cfg "WinClientServer\DynDspOp\DynDspOp.ini";
```

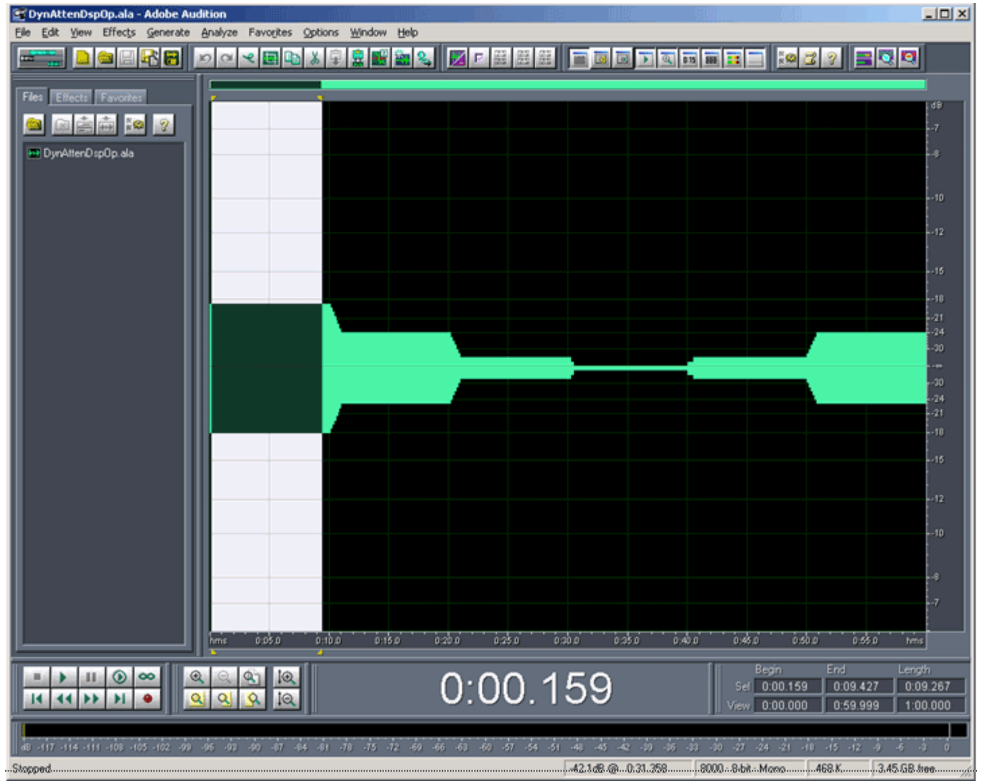
Applicable Schedule



```
Time,Operator,Transition,Value
0,Atten,0,5
10,Atten,1,10
20,Atten,1,20
30,Atten,0.5,30
40,Atten,1,20
50,Atten,1,10
```

- Input tones are attenuated as per the specified Time, Transition, and Values defined in the Schedule *.csv file

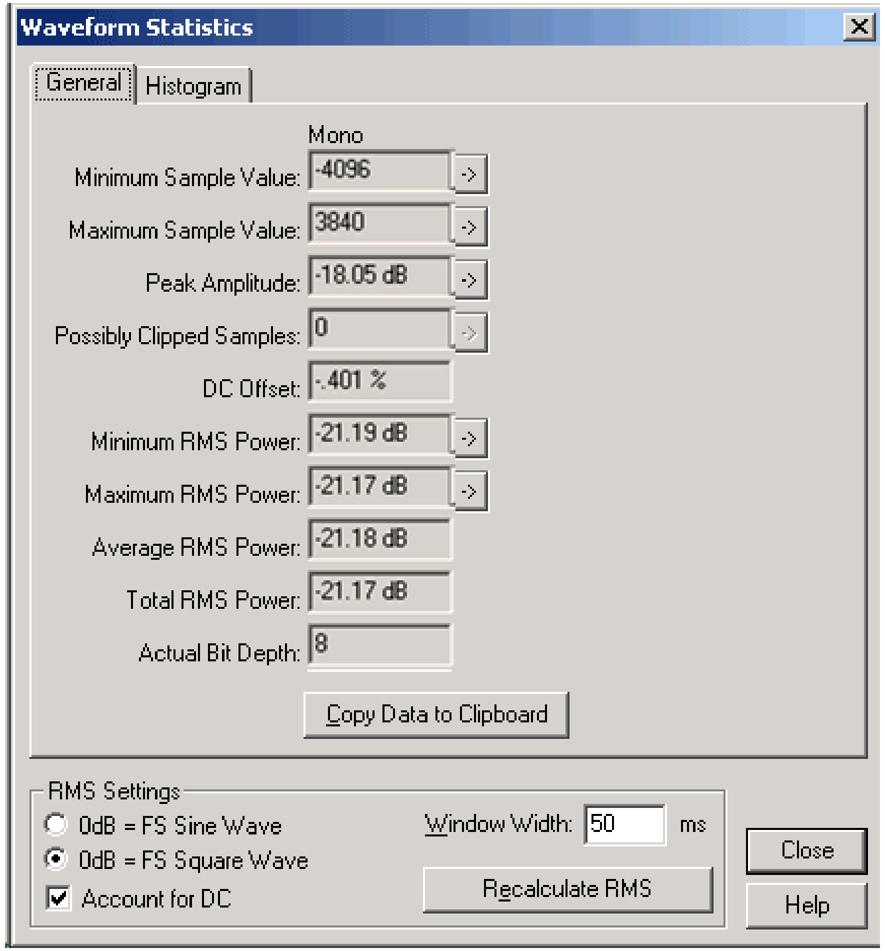
Output Analysis


DynAttenDspOp.ala - Adobe Audition
File Edit View Effects Generate Analyze Favorites Options Window Help
Files Effects Favorites
DynAttenDspOp.ala
dB
hms

0:00.159

	Begin	End	Length
Sel	0:00.159	0:09.427	0:09.267
View	0:00.000	0:59.999	1:00.000

Stopped -42.1dB @ 0:31.358 8000 8-bit Mono 468 K 3.45 GB free



	Mono
Minimum Sample Value:	-4096
Maximum Sample Value:	3840
Peak Amplitude:	-18.05 dB
Possibly Clipped Samples:	0
DC Offset:	-.401 %
Minimum RMS Power:	-21.19 dB
Maximum RMS Power:	-21.17 dB
Average RMS Power:	-21.18 dB
Total RMS Power:	-21.17 dB
Actual Bit Depth:	8

RMS Settings

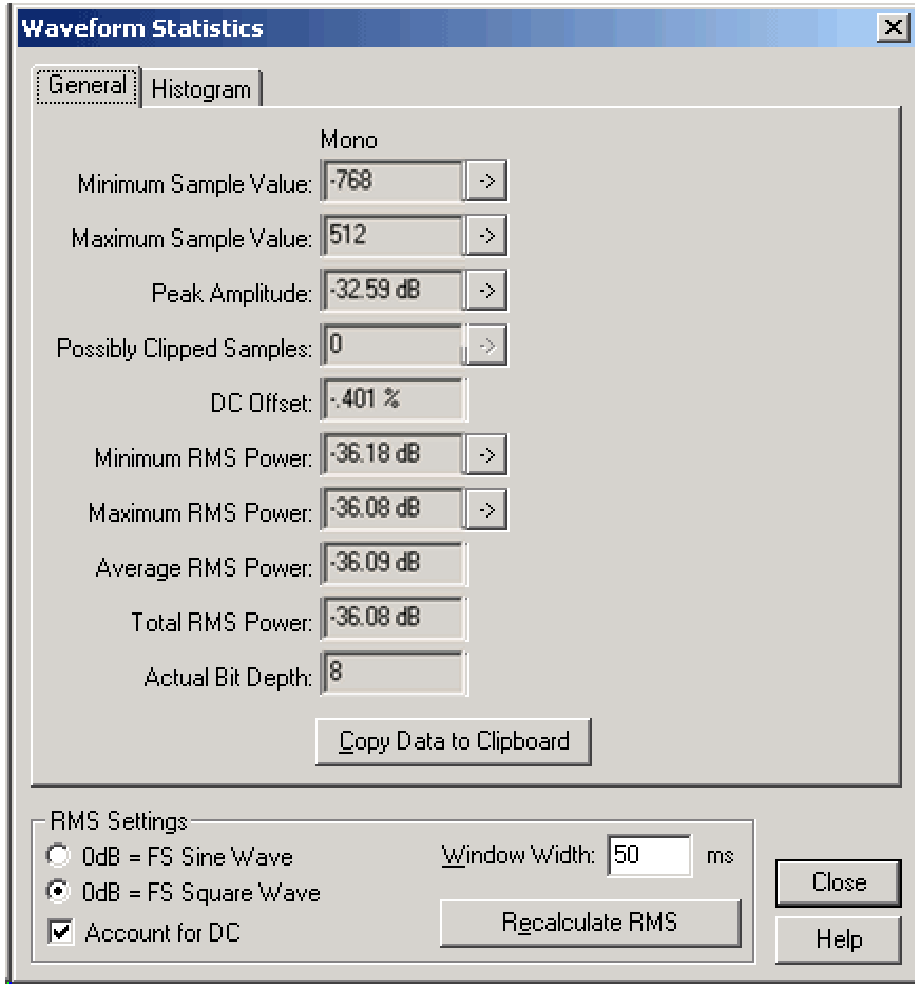
○ 0dB = FS Sine Wave

● 0dB = FS Square Wave

☑ Account for DC

Window Width: 50 ms

- **0 - 10 sec:** Power = -21.17 dBov = -15.02 dBm. Target power is -10 dBm (source signal power) -5 dB (attenuation) = -15 dBm



Waveform Statistics

	Mono
Minimum Sample Value:	-768
Maximum Sample Value:	512
Peak Amplitude:	-32.59 dB
Possibly Clipped Samples:	0
DC Offset:	-.401 %
Minimum RMS Power:	-36.18 dB
Maximum RMS Power:	-36.08 dB
Average RMS Power:	-36.09 dB
Total RMS Power:	-36.08 dB
Actual Bit Depth:	8

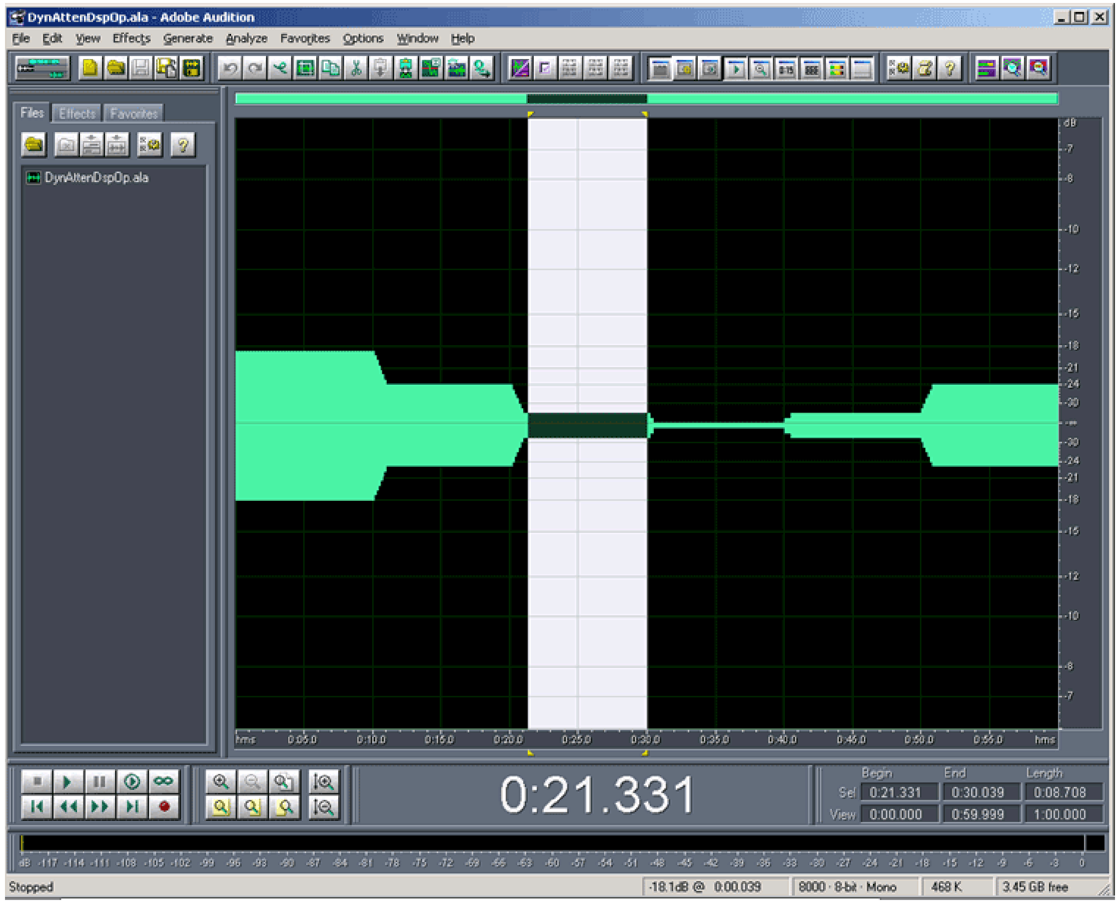
RMS Settings

○ 0dB = FS Sine Wave

● 0dB = FS Square Wave

☑ Account for DC

Window Width: 50 ms



DynAttenDspOp.ala - Adobe Audition

File Edit View Effects Generate Analyze Favorites Options Window Help
Files Effects Favorites
DynAttenDspOp.ala
hms 0:05.0 0:10.0 0:15.0 0:20.0 0:25.0 0:30.0 0:35.0 0:40.0 0:45.0 0:50.0 0:55.0 hms

0:21.331

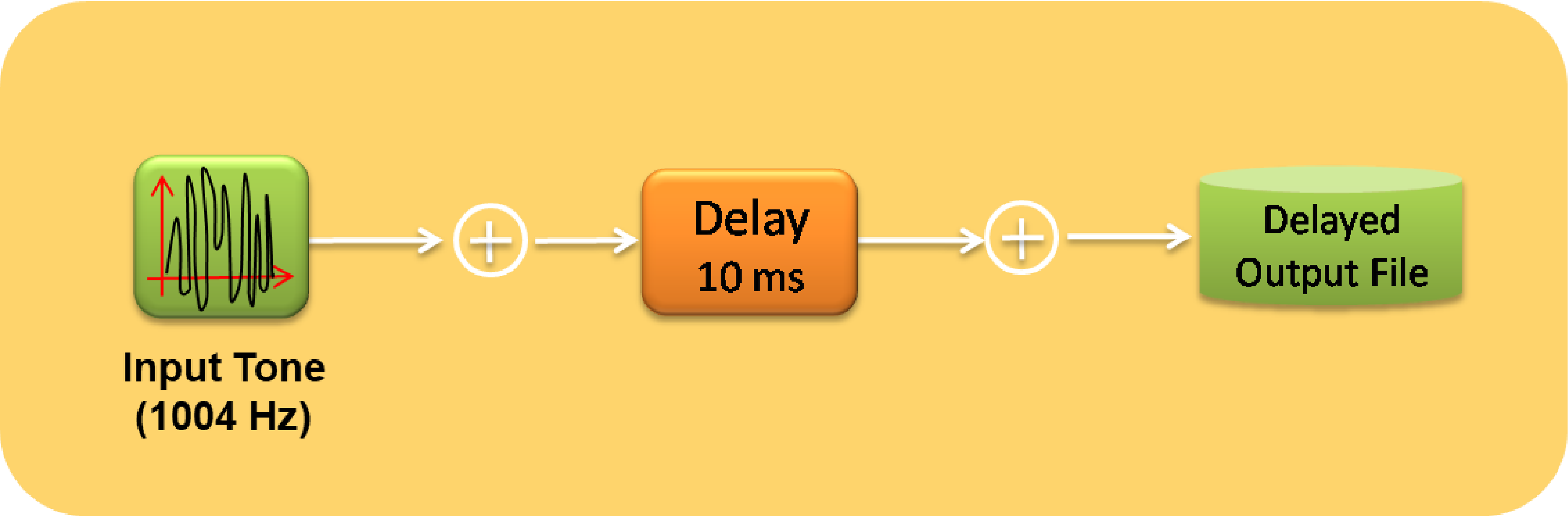
	Begin	End	Length
Sel	0:21.331	0:30.039	0:08.708
View	0:00.000	0:59.999	1:00.000

Stopped | -18.1dB @ 0:00.039 | 8000 · 8-bit · Mono | 468 K | 3.45 GB free

- 10 - 20 sec: Power = -26.21 dBov = -20.07 dBm. Target power is -10 dBm (source signal power) -10 dB (attenuation) = -20 dBm. Note power tapers down over transition interval of 1 second

Dynamic Offline Delay ("DelayDspOp")

Delaying the tone input



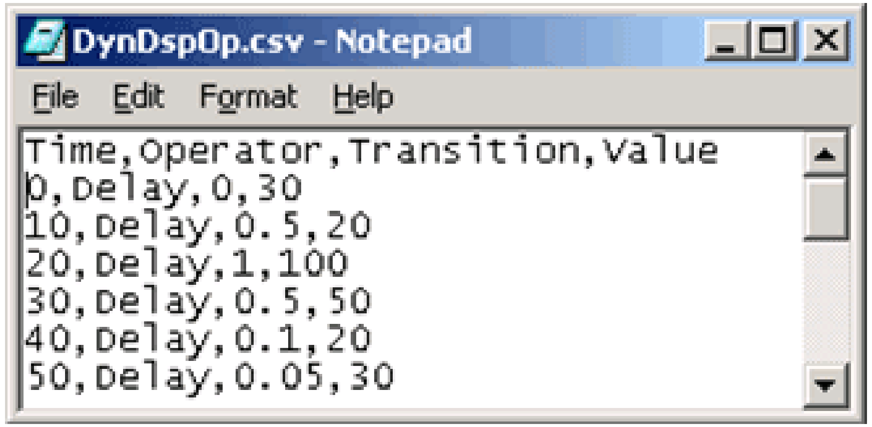
- Example script demonstrates delaying of Input Tone (1004 Hz) by 10 milliseconds to obtain delayed output file

Offline Delay Testing ("DelayDspOp")

DelayDspOp WCS Test Script

```
set latency 4;
set response 6;

// (2) Delay
outfile(delay(tone(1000 hz, -10 dbm), 10 msec), "WinClientServer\DynDspOp\
DynDelayDspOp.ala") 60 sec cfg "WinClientServer\DynDspOp\DynDspOp.ini";
```

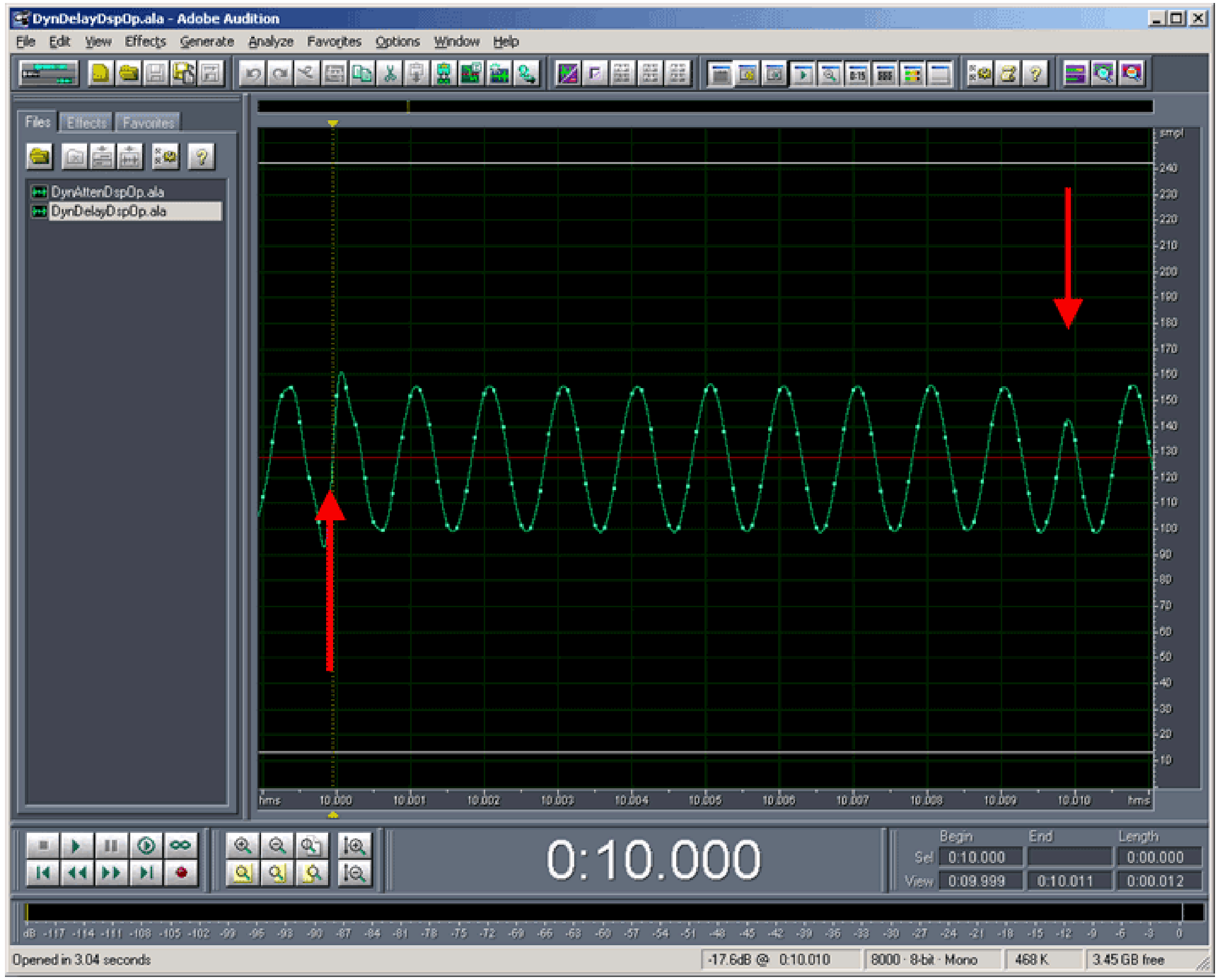
Applicable Schedule



```
Time,Operator,Transition,Value
0,Delay,0,30
10,Delay,0.5,20
20,Delay,1,100
30,Delay,0.5,50
40,Delay,0.1,20
50,Delay,0.05,30
```

- Input tone is delayed as per the specified Time, Transition, and Values defined in the Schedule *.csv file

Output Analysis



DynDelayDspOp.ala - Adobe Audition

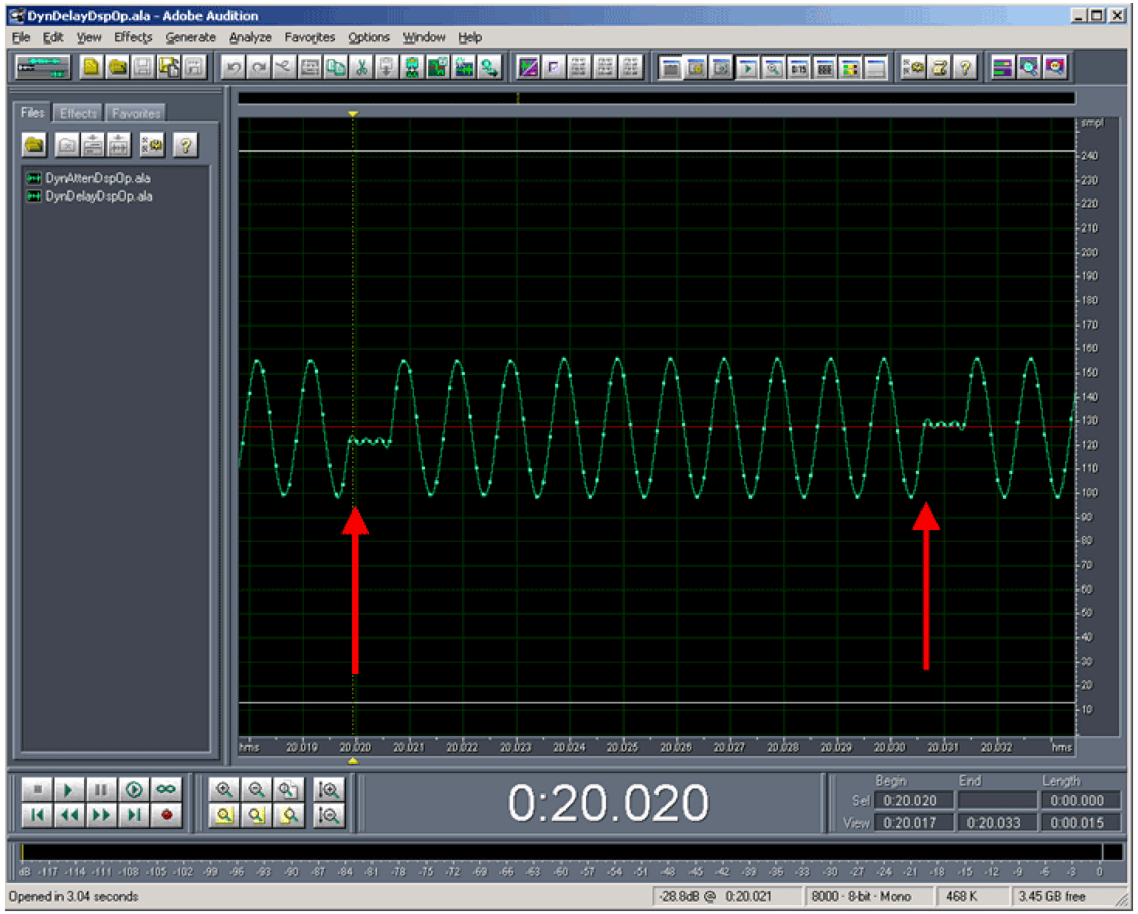
DynAttenDspOp.ala
DynDelayDspOp.ala

0:10.000

	Begin	End	Length
Sel	0:10.000		0:00.000
View	0:09.999	0:10.011	0:00.012

Opened in 3.04 seconds | -17.6dB @ 0:10.010 | 8000 · 8-bit · Mono | 468 K | 3.45 GB free

- Dynamic delay at 10.0 sec., and 10.1 sec., as samples are removed to shorten the delay


DynDelayDspOp.ala - Adobe Audition
File Edit View Effects Generate Analyze Favorites Options Window Help
Files Effects Favorites
DynAttenDspOp.ala
DynDelayDspOp.ala
smpl
240
230
220
210
200
190
180
170
160
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
hms 20.019 20.020 20.021 20.022 20.023 20.024 20.025 20.026 20.027 20.028 20.029 20.030 20.031 20.032 hms
dB -117 -114 -111 -108 -105 -102 -99 -96 -93 -90 -87 -84 -81 -78 -75 -72 -69 -66 -63 -60 -57 -54 -51 -48 -45 -42 -39 -36 -33 -30 -27 -24 -21 -18 -15 -12 -9 -6 -3 0

0:20.020

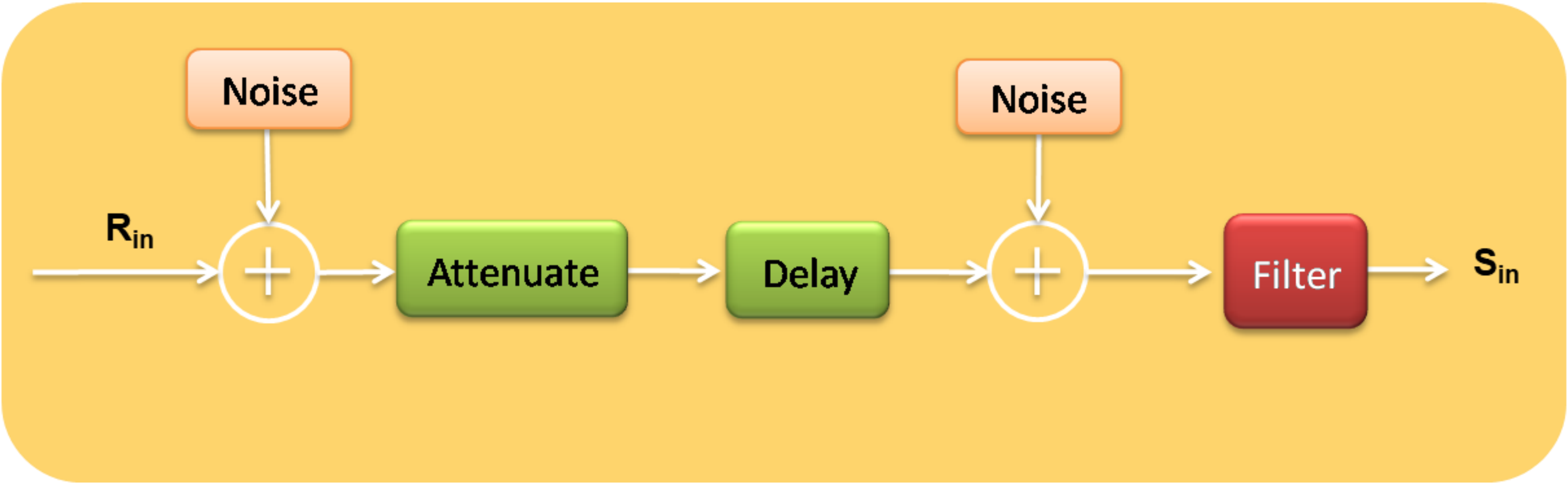
	Begin	End	Length
Sel	0:20.020		0:00.000
View	0:20.017	0:20.033	0:00.015

Opened in 3.04 seconds | -28.8dB @ 0:20.021 | 8000 · 8-bit · Mono | 468 K | 3.45 GB free

- Between 20.0 and 21.0 sec., samples are repeated to insert more delay. Note the processing blocksize is 10 ms. For unclocked DSP operations (no tx or rx DSP operations), the processing block size is the sum of the latency and the response time

Dynamic Filter ("FiltDspOp")

Transmit Filtered R_{in} and White Noise



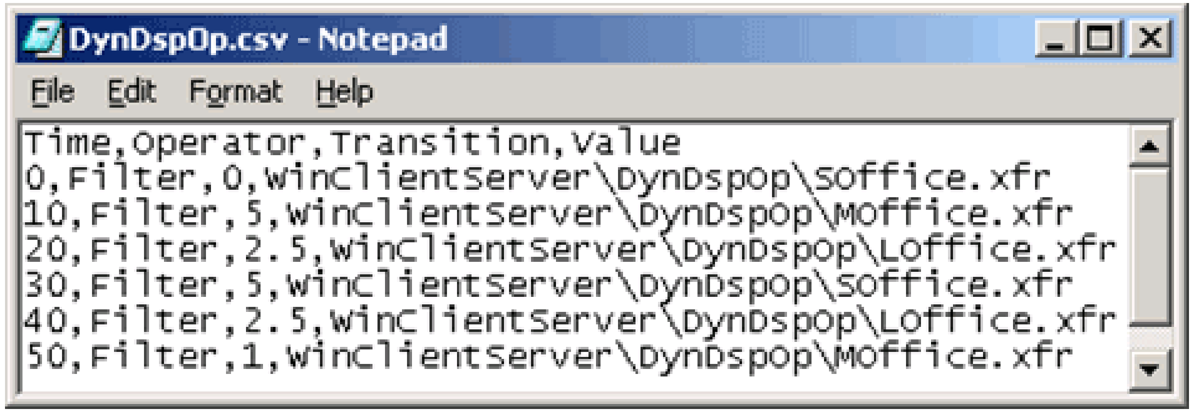
- Example script demonstrates transmission of R_{in} along with noise, to obtain a filtered S_{in}

Offline Delay Testing ("FiltDspOp")

FiltDspOp WCS Test Script

```
set latency 4;
set response 6;

// (3) Filter
Dspop {outfile(filter(whitenoise(-10 dbm), "WinClientServer\DynDspOp\SOffice.xfr"),
"WinClientServer\DynDspOp\DynFiltDspOpSin.ala"), outfile(whitenoise(-10 dbm),
"WinClientServer\DynDspOp\DynFiltDspOpRin.ala") } 60 sec cfg "WinClientServer\ DynDspOp\DynDspOp.ini";
```

Applicable Schedule



```
Time,Operator,Transition,Value
0,Filter,0,WinClientServer\DynDspop\SOffice.xfr
10,Filter,5,WinClientServer\DynDspop\MOffice.xfr
20,Filter,2.5,WinClientServer\DynDspop\LOffice.xfr
30,Filter,5,WinClientServer\DynDspop\SOffice.xfr
40,Filter,2.5,WinClientServer\DynDspop\LOffice.xfr
50,Filter,1,WinClientServer\DynDspop\MOffice.xfr
```

- Input tone is delayed as per the specified Time, Transition, and Values defined in the Schedule *.csv file

Types of Filters Used

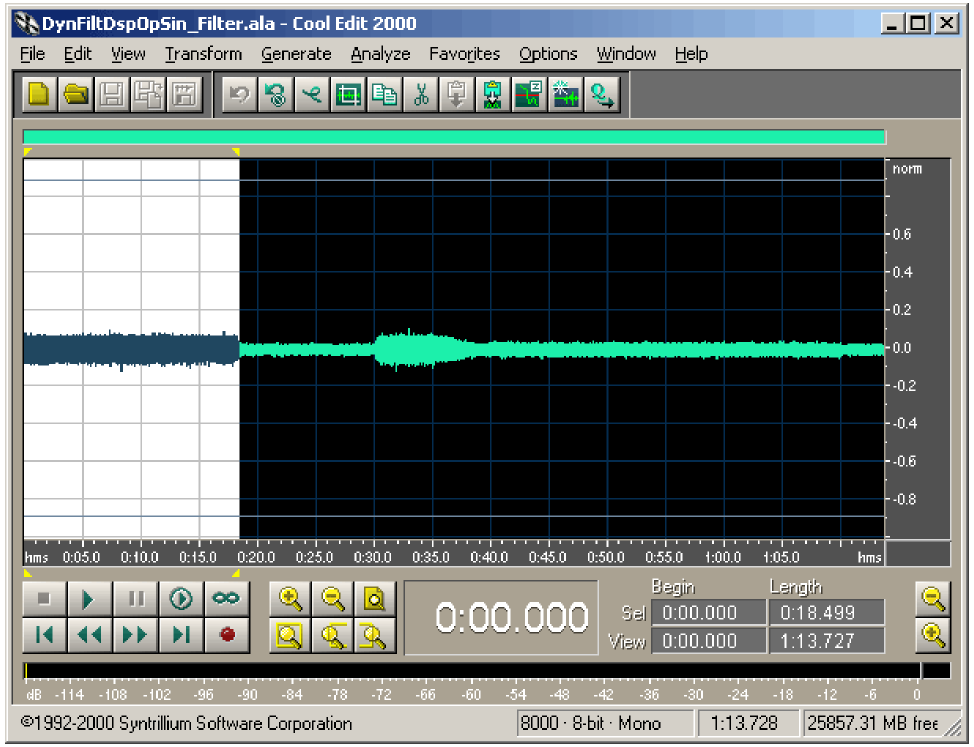
Filter DSP Operations

Filter	Description	Broadband ERL
SOffice.xfr	Small office environment	16.36 dB
MOffice.xfr	Medium-sized office environment	25.48 dB
LOffice.xfr	Large office environment	23.71 dB

Note that the FiltDspOp WCS Test script produces two output files:

- DynFiltDspOpRin.ala: The original noise signal in A-Law compressed form
- DynFiltDspOpSin.ala: The dynamically filtered noise signal in A-Law compressed form

Output Analysis

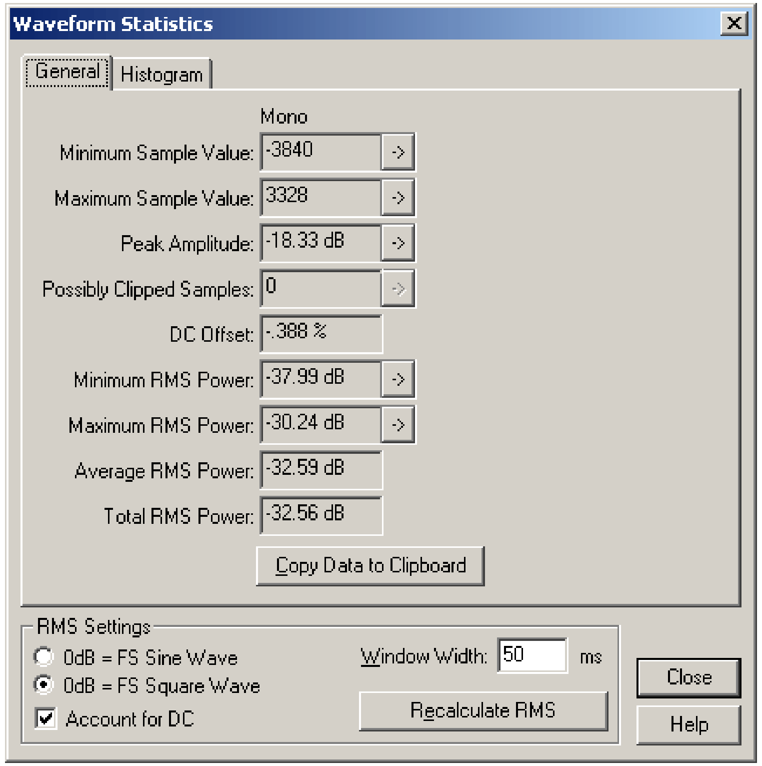

DynFiltDspOpSin_Filter.ala - Cool Edit 2000

File
Edit
View
Transform
Generate
Analyze
Favorites
Options
Window
Help
norm
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
hms 0:05.0 0:10.0 0:15.0 0:20.0 0:25.0 0:30.0 0:35.0 0:40.0 0:45.0 0:50.0 0:55.0 1:00.0 1:05.0 hms
dB -114 -108 -102 -96 -90 -84 -78 -72 -66 -60 -54 -48 -42 -36 -30 -24 -18 -12 -6 0

0:00.000

	Begin	Length
Sel	0:00.000	0:18.499
View	0:00.000	1:13.727

©1992-2000 Syntrillium Software Corporation | 8000 · 8-bit · Mono | 1:13.728 | 25857.31 MB free


Waveform Statistics

	Mono
Minimum Sample Value:	-3840
Maximum Sample Value:	3328
Peak Amplitude:	-18.33 dB
Possibly Clipped Samples:	0
DC Offset:	-.388 %
Minimum RMS Power:	-37.99 dB
Maximum RMS Power:	-30.24 dB
Average RMS Power:	-32.59 dB
Total RMS Power:	-32.56 dB

RMS Settings

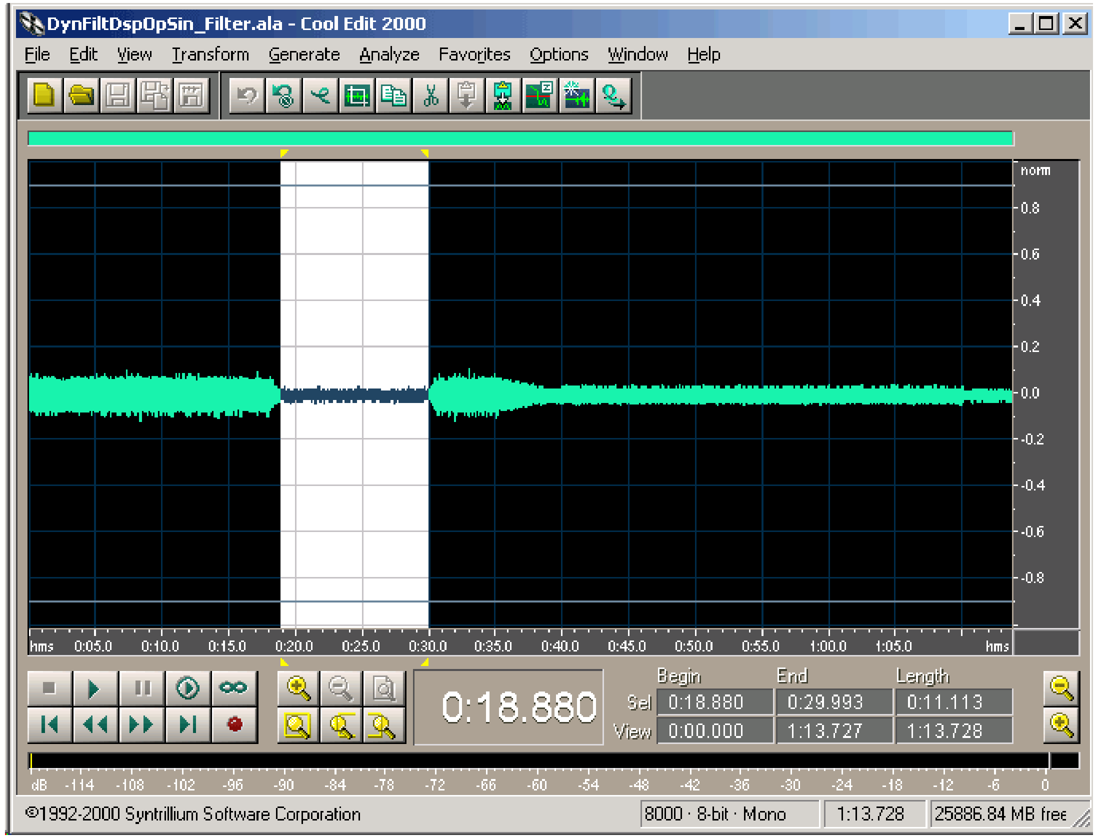
○ 0dB = FS Sine Wave

● 0dB = FS Square Wave

☑ Account for DC

Window Width: 50 ms

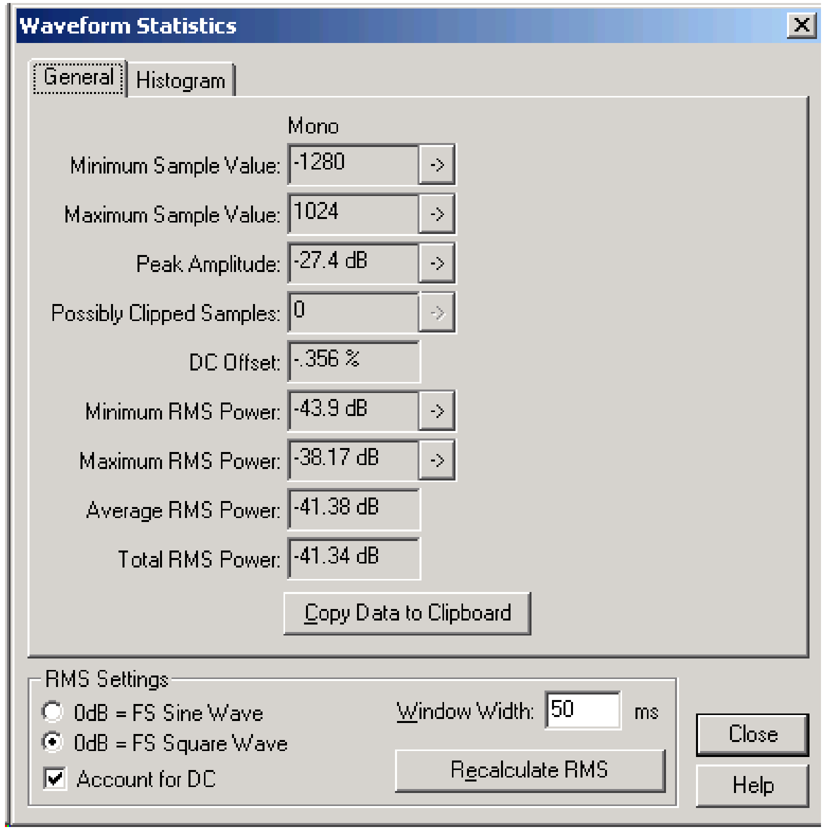
- DynFiltDspOpRin.ala: Power = -32.58 dBov = -26.58 dBm, Target power is -10 dBm (source signal power) -16.36 dB (SOffice.xfr attenuation) = -26.36 dBm


DynFiltDspOpSin_Filter.ala - Cool Edit 2000
File Edit View Transform Generate Analyze Favorites Options Window Help
norm
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
hms 0:05.0 0:10.0 0:15.0 0:20.0 0:25.0 0:30.0 0:35.0 0:40.0 0:45.0 0:50.0 0:55.0 1:00.0 1:05.0 hms
dB -114 -108 -102 -96 -90 -84 -78 -72 -66 -60 -54 -48 -42 -36 -30 -24 -18 -12 -6 0

0:18.880

	Begin	End	Length
Sel	0:18.880	0:29.993	0:11.113
View	0:00.000	1:13.727	1:13.728

©1992-2000 Syntrillium Software Corporation | 8000 · 8-bit · Mono | 1:13.728 | 25886.84 MB free



	Mono
Minimum Sample Value:	-1280
Maximum Sample Value:	1024
Peak Amplitude:	-27.4 dB
Possibly Clipped Samples:	0
DC Offset:	-.356 %
Minimum RMS Power:	-43.9 dB
Maximum RMS Power:	-38.17 dB
Average RMS Power:	-41.38 dB
Total RMS Power:	-41.34 dB

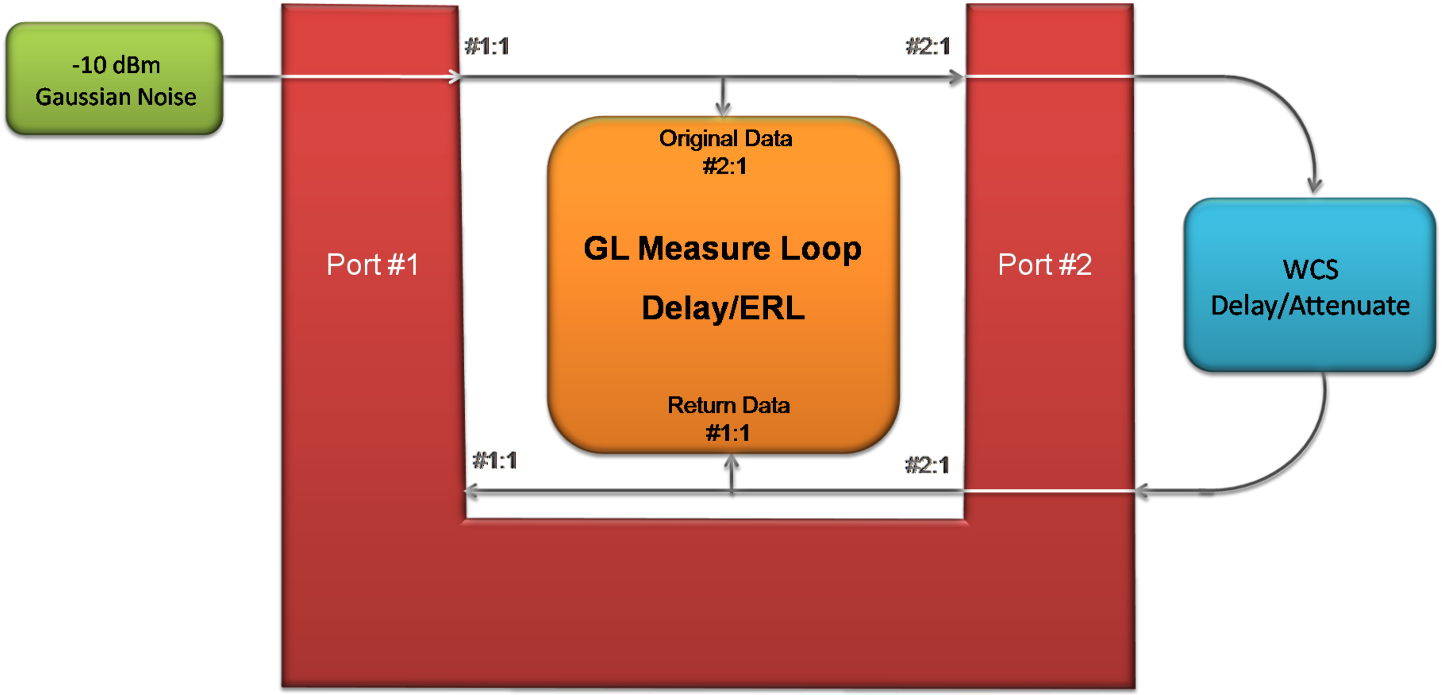
RMS Settings
○ 0dB = FS Sine Wave
● 0dB = FS Square Wave
☑ Account for DC
Window Width: 50 ms

- DynFiltDspOpSin.ala: Power = -41.38 dBov = -35.38 dBm, Target power is -10 dBm (source signal power) -25.48

  dB (MOffice.xfr attenuation) = -35.48 dBm

Real-time Delay/Attenuate Operations

Real-time Dynamic Delay / Attenuation Testing



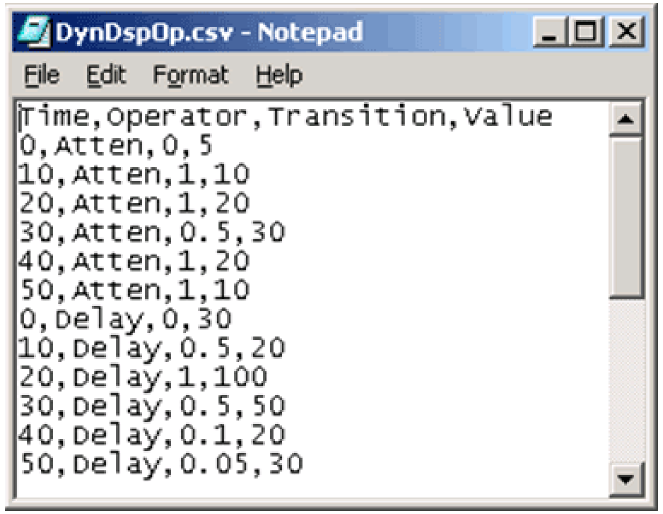
- On port# 1, -10dBm noise is input, the original data is monitored using GL's Measure Loop Delay/ERL. The original data at port# 2 is delayed or attenuated as per defined parameters in scheduled file and sent back on port# 2. The returned data is used to verify the delay, which agrees closely with the programmed values

Real-time Delay/Attenuation Testing

Dynamic Echo path implemented in WCS Test Script

```
set latency 4;
set response 6;

// (4) Delay/Attenuate
tx(atten(delay(rx(#2:1), 10 msec), 5 db), #2:1) 60 sec cfg "WinClientServer\DynDspOp\
DynDspOp.ini" priority 1;
```

Applicable Schedule

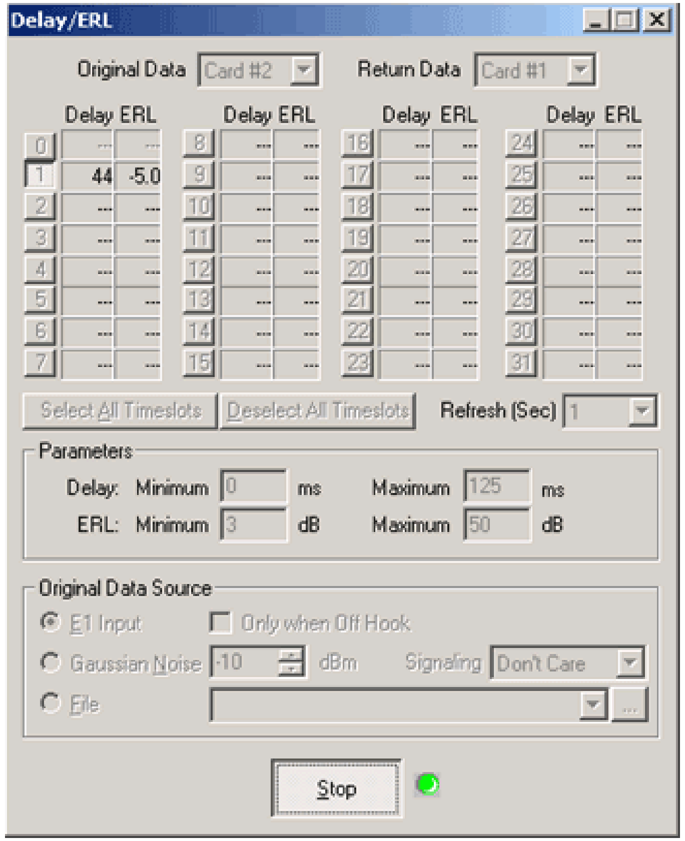


```
Time,Operator,Transition,Value
0,Atten,0,5
10,Atten,1,10
20,Atten,1,20
30,Atten,0.5,30
40,Atten,1,20
50,Atten,1,10
0,Delay,0,30
10,Delay,0.5,20
20,Delay,1,100
30,Delay,0.5,50
40,Delay,0.1,20
50,Delay,0.05,30
```

- In real-time, the input signal is delayed/attenuated as per the specified Time, Transition, and Values defined in the Schedule *.csv file

Observations

0 - 10Sec



Original Data: Card #2 Return Data: Card #1

	Delay	ERL		Delay	ERL		Delay	ERL		Delay	ERL
0	---	---	8	---	---	16	---	---	24	---	---
1	44	-5.0	9	---	---	17	---	---	25	---	---
2	---	---	10	---	---	18	---	---	26	---	---
3	---	---	11	---	---	19	---	---	27	---	---
4	---	---	12	---	---	20	---	---	28	---	---
5	---	---	13	---	---	21	---	---	29	---	---
6	---	---	14	---	---	22	---	---	30	---	---
7	---	---	15	---	---	23	---	---	31	---	---

Refresh (Sec): 1

Parameters

Delay: Minimum 0 ms Maximum 125 ms

ERL: Minimum 3 dB Maximum 50 dB

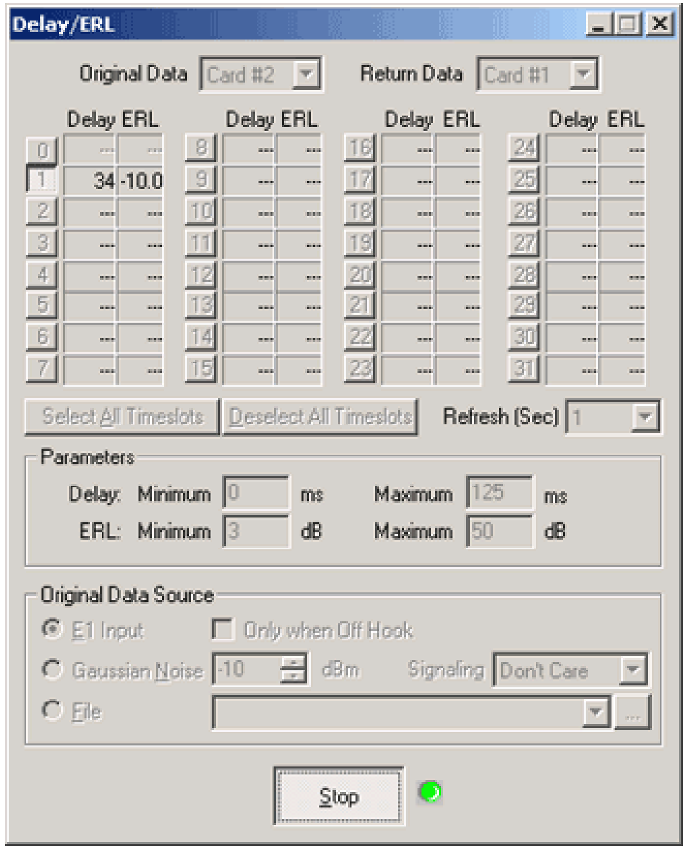
Original Data Source

● E1 Input □ Only when Off Hook

○ Gaussian Noise -10 dBm Signaling Don't Care

○ File

10 – 20Sec



Original Data: Card #2 Return Data: Card #1

	Delay	ERL		Delay	ERL		Delay	ERL		Delay	ERL
0	---	---	8	---	---	16	---	---	24	---	---
1	34	-10.0	9	---	---	17	---	---	25	---	---
2	---	---	10	---	---	18	---	---	26	---	---
3	---	---	11	---	---	19	---	---	27	---	---
4	---	---	12	---	---	20	---	---	28	---	---
5	---	---	13	---	---	21	---	---	29	---	---
6	---	---	14	---	---	22	---	---	30	---	---
7	---	---	15	---	---	23	---	---	31	---	---

Refresh (Sec): 1

Parameters

Delay: Minimum 0 ms Maximum 125 ms

ERL: Minimum 3 dB Maximum 50 dB

Original Data Source

● E1 Input □ Only when Off Hook

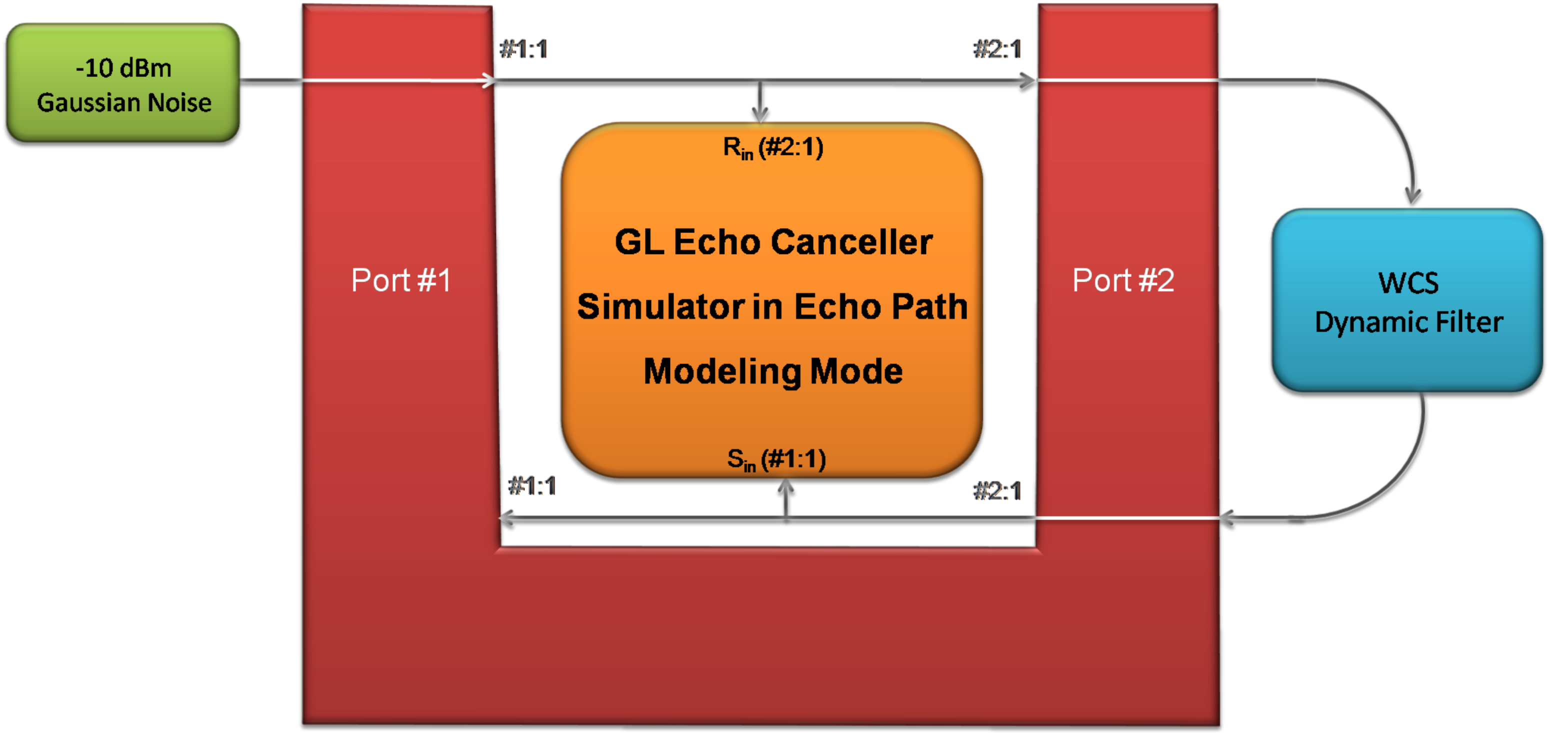
○ Gaussian Noise -10 dBm Signaling Don't Care

○ File

- Observe the filter transitions from one to the next at 10 - seconds interval

Real-time Filter Operations

Real-time Dynamic Filter Testing



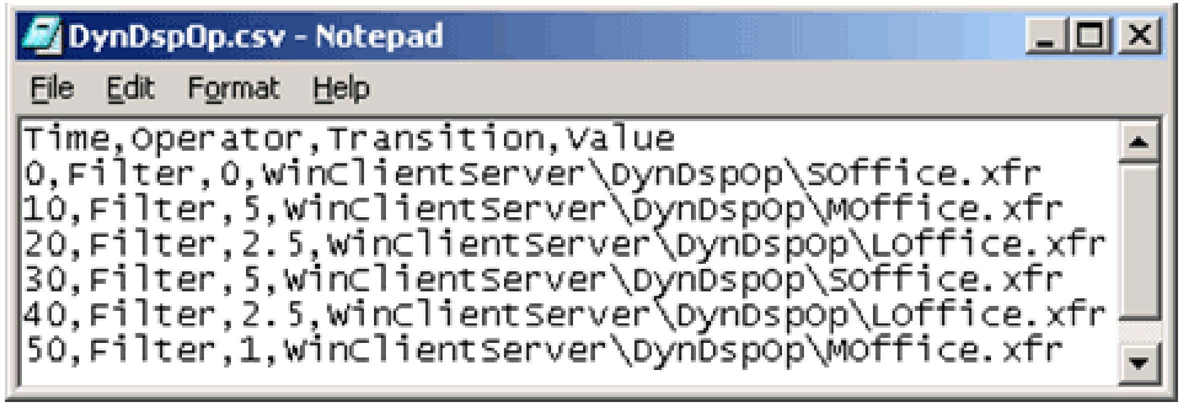
- On port# 1, -10dBm noise is input, the R_{in} is monitored using GL's Delay Echo Canceller. The R_{in} at port# 2 is filtered as per parameters in scheduled file and sent back on port# 2. The returned S_{in} is used to verify the filter, which agrees closely with the programmed values

Real-time Filter Testing

Dynamic Echo path implemented in WCS Test Script

```
set latency 4;
set response 6;

// (4) Realtime Filter
tx(filter(rx(#2:1), " WinClientServer\DynDspOp\SOffice.xfr"), #2:1) 60 sec cfg
"WinClientServer\DynDspOp\DynDspOp.ini"  priority 1;
```

Applicable Schedule

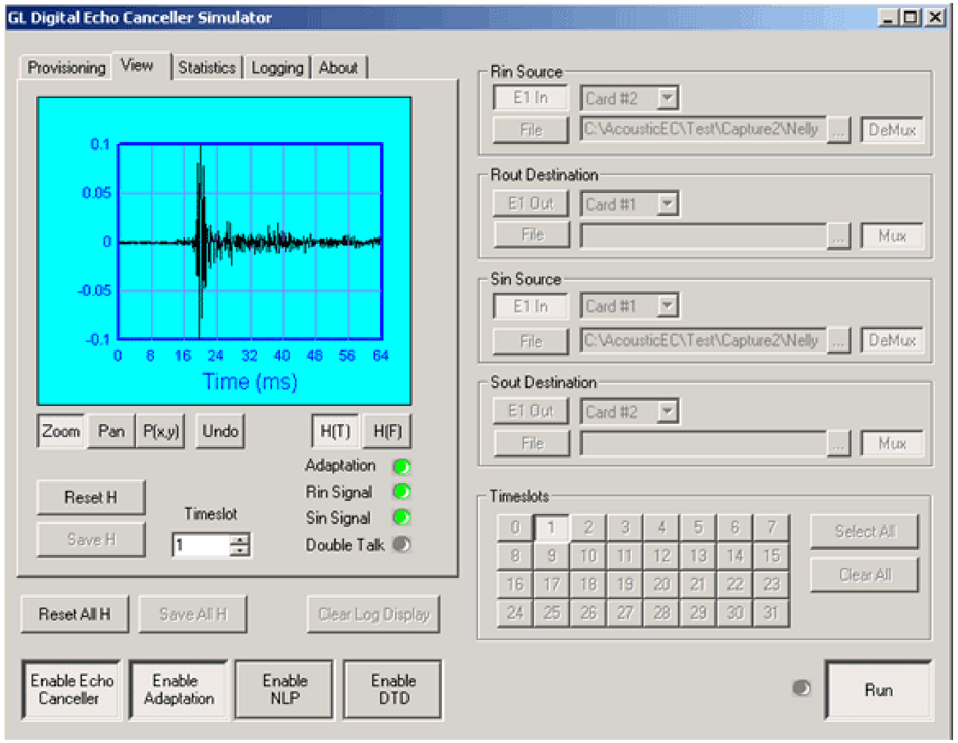


```
Time,Operator,Transition,Value
0,Filter,0,WinClientServer\DynDspOp\SOffice.xfr
10,Filter,5,WinClientServer\DynDspOp\MOffice.xfr
20,Filter,2.5,WinClientServer\DynDspOp\LOffice.xfr
30,Filter,5,WinClientServer\DynDspOp\SOffice.xfr
40,Filter,2.5,WinClientServer\DynDspOp\LOffice.xfr
50,Filter,1,WinClientServer\DynDspOp\MOffice.xfr
```

- In real-time, the input signal is filtered as per the specified Time, Transition, and Values defined in the Schedule *.csv file

Observations

SOffice.xfr



GL Digital Echo Canceller Simulator

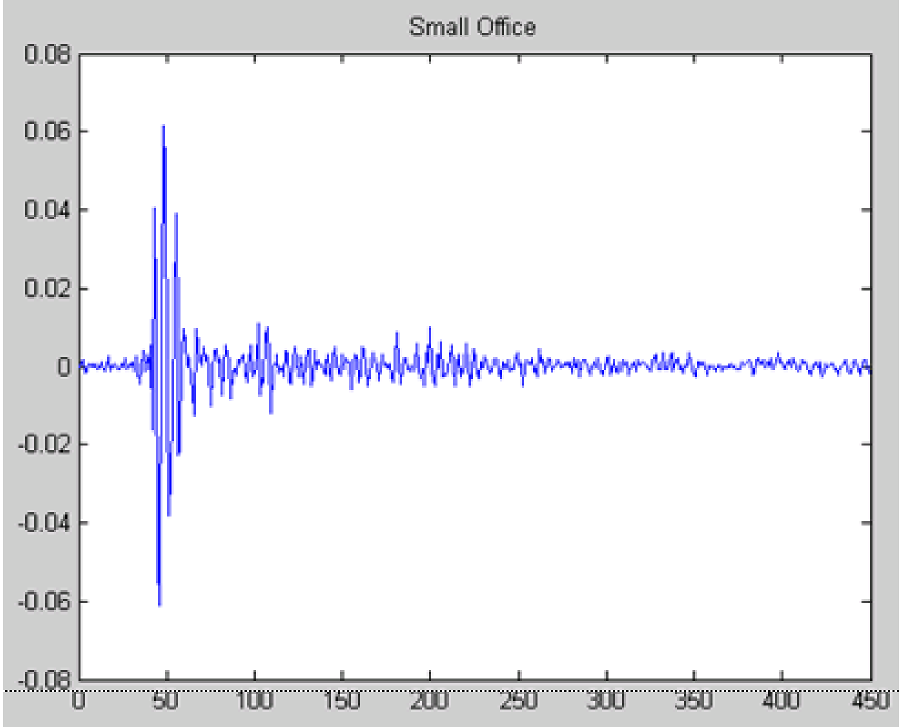
Rin Source: E1 In, Card #2, File: C:\AcousticEC\Test\Capture2\Nelly, DeMux

Rout Destination: E1 Out, Card #1, File, Mux

Sin Source: E1 In, Card #1, File: C:\AcousticEC\Test\Capture2\Nelly, DeMux

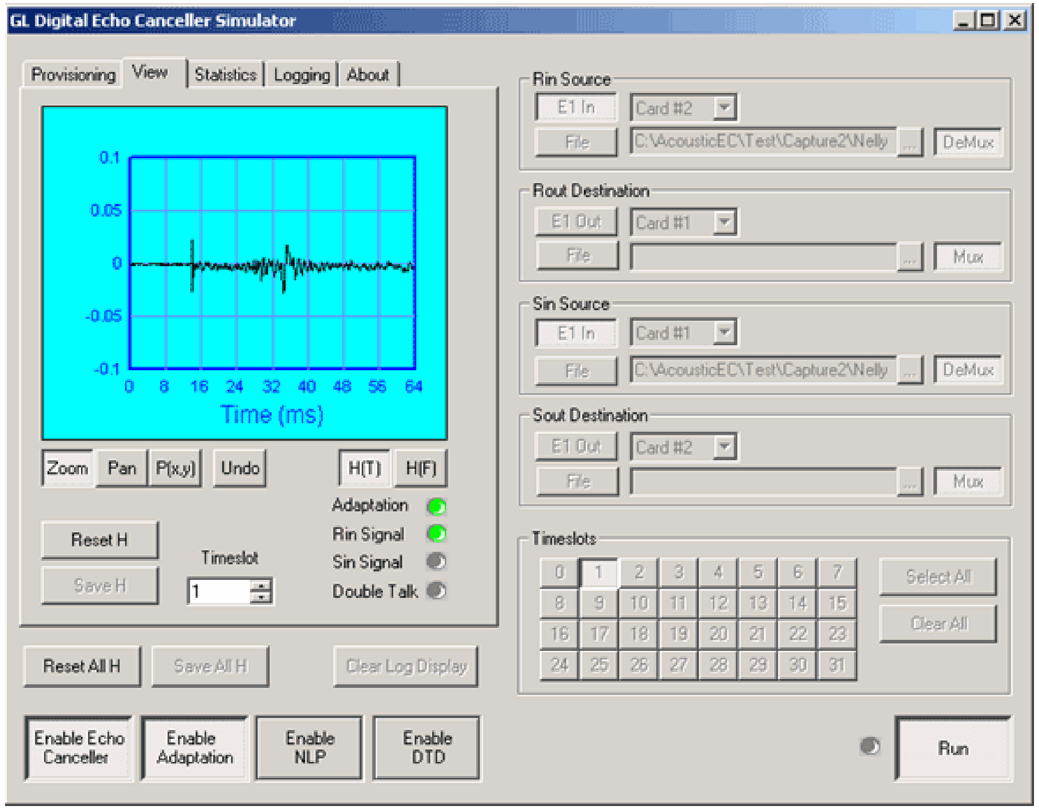
Sout Destination: E1 Out, Card #2, File, Mux

Timeslot: 1



- Small, Medium and Large office environment filter contains 450 taps, corresponding to 56.25 ms
- Successive filters evolves in the DEC View

MOffice.xfr



GL Digital Echo Canceller Simulator

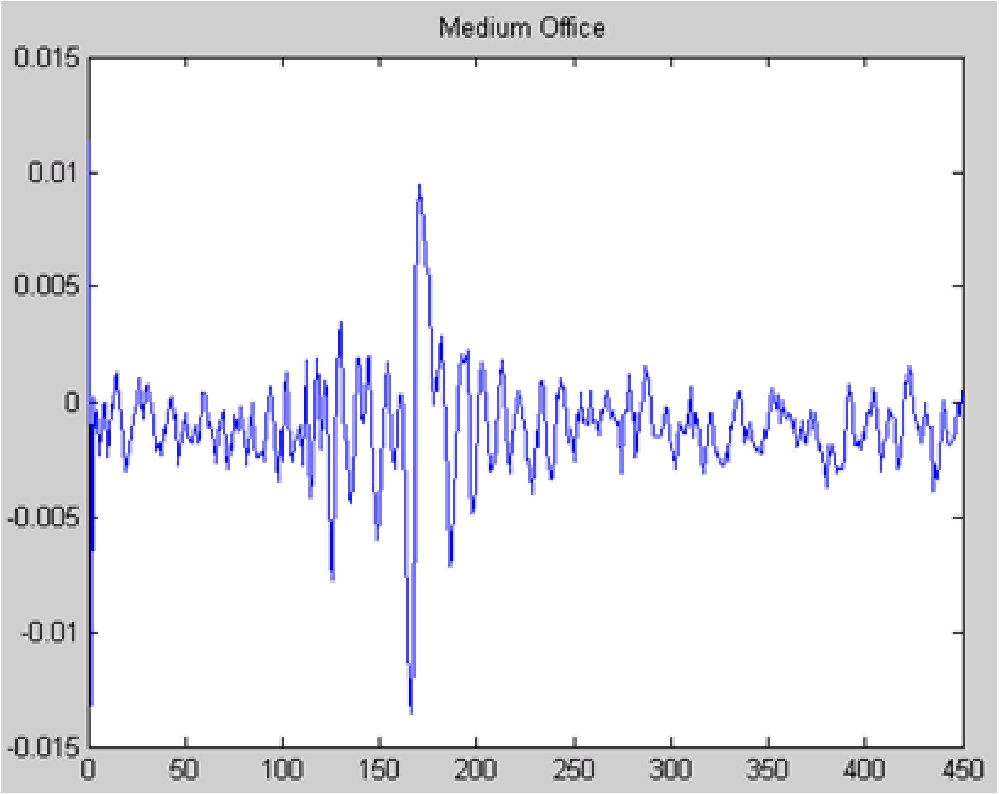
Rin Source: E1 In, Card #2, File C:\AcousticEC\Test\Capture2\Nelly

Rout Destination: E1 Out, Card #1

Sin Source: E1 In, Card #1, File C:\AcousticEC\Test\Capture2\Nelly

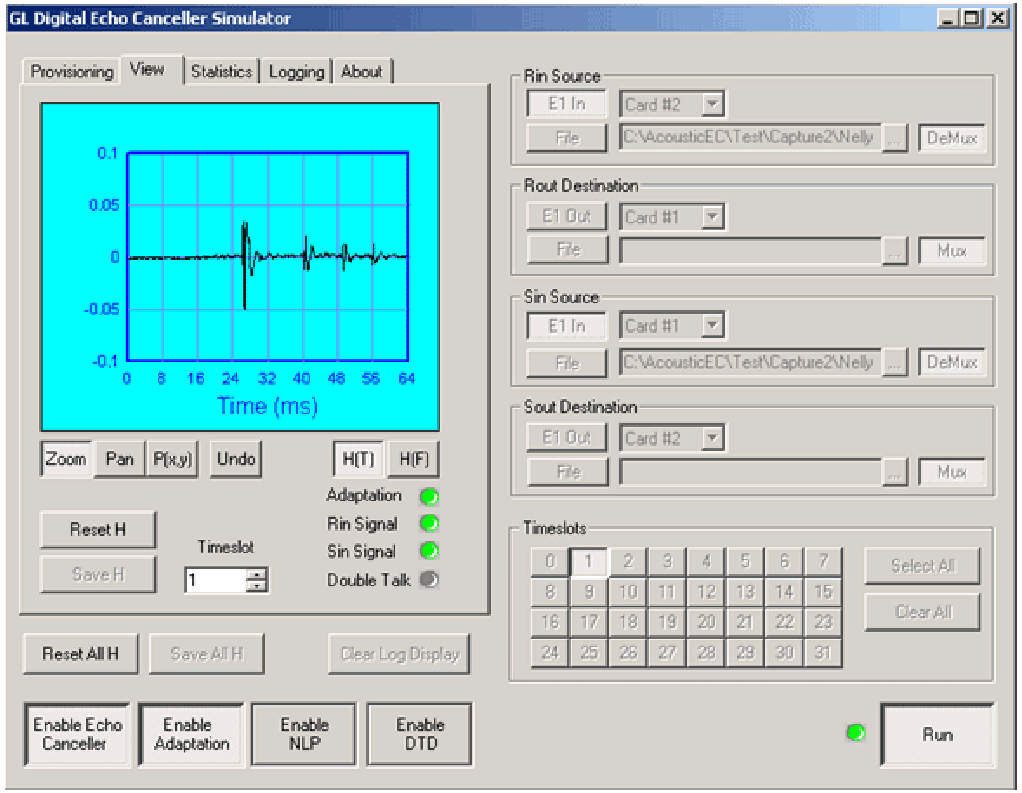
Sout Destination: E1 Out, Card #2

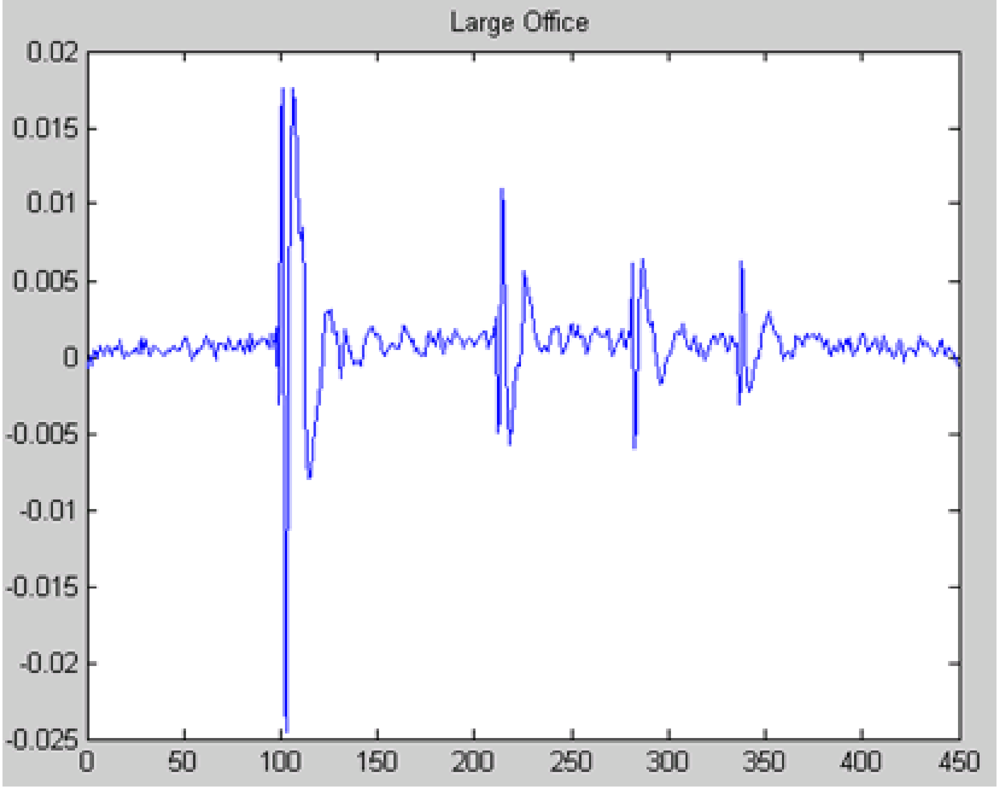
Timeslot: 1



- The 14 ms offset due to WCS transmit latency, processing block size, and hardware buffering is displayed

LOffice.xfr

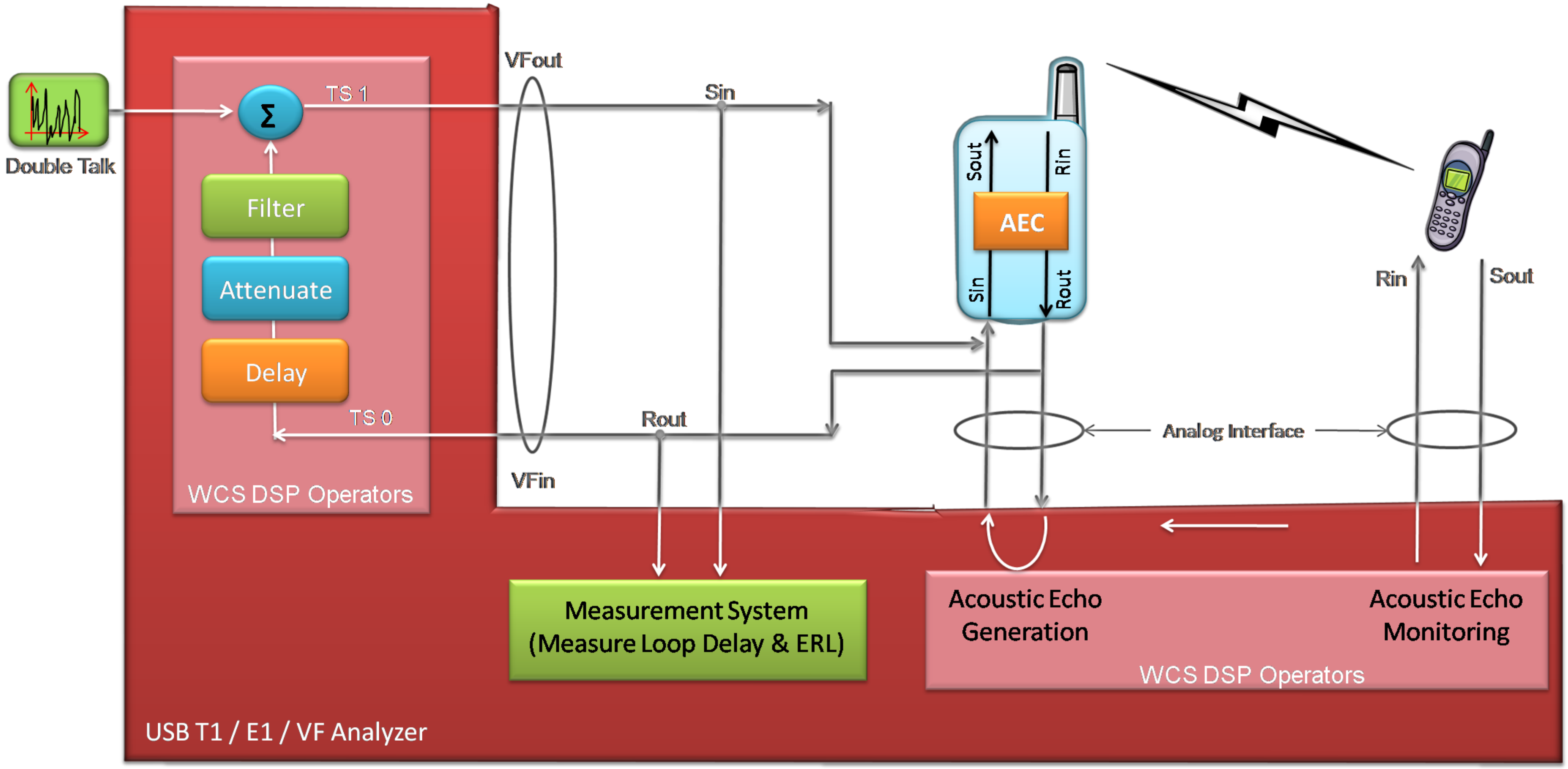




- Note the smooth transition between filters at 10-second intervals
- The speed of the transition is governed by the transition times specified in the Schedule File

GL Communications

Acoustic Echo Simulation



Thank You