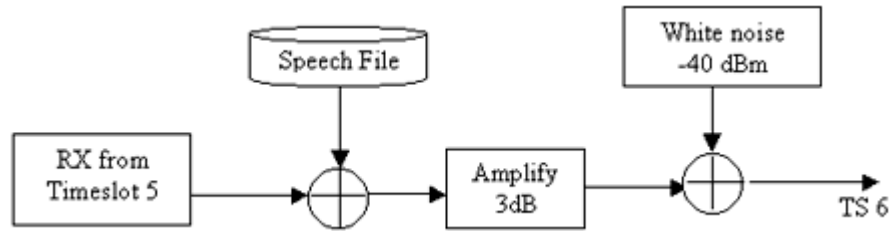


Adding speech and noise to the receive data



Script Description:

This script transmits the amplified speech file with white noise and the data received on a particular timeslot.

Variation 1:

- Connect two PCs A and B. Run WCS script on PC A and verify the output on PC B. Do not run the GUI on PC A.
- Add up the transmitted speech file 'b52_ula.pcm' and the data coming from timeslot 5
- Amplify it by 3db
- Add white noise of -40dBm to it.
- Transmit it through timeslot 6.

Script:

```
// Connection assumed: connect card1 and card2 from PC 'A' to card1 and card2 of PC 'B'
// respectively. This script should not be run simultaneously with GUI
//MODE:SEQUENTIAL
//
// User needs to comment the commands according to his requirement
//
//BOARD SETTINGS...
//
//INTERFACE SETTINGS
set rx interface terminate #*;
//set rx interface monitor #*;
//set rx interface bridge #*;
wait 3000;
//
//FRAME FORMAT SETTINGS FOR T1
set superframe format esf #*;
//set superframe format d4 #*;
//
//FRAME FORMAT SETTINGS FOR E1
//set signaling mode cas #*;
//set signaling mode ccs #*;
//set crc4 on#*;
//set crc4 off #*;
//
//CLOCK SETTINGS
set tx clock source internal #*;
//set tx clock source recovered #*;
```

```

//set tx clock source external #*;
//
//LOOPBACK SETTINGS
//set inward driver loopback on #*;
//set outward driver loopback on #*;
set outward driver loopback off #*;
set inward driver loopback off #*;
wait 3000;
//
//- - - - -Verification of initialization
//CHECKING FORMAT SETTINGS FOR T1
get superframe format #*;
//
//CHECKING FORMAT SETTINGS FOR E1
//get signaling mode #*;
//
//CHECKING OTHER SETTINGS FOR THE CARDS
get tx clock source #*;
get outward driver loopback #*;
get rx line frequency #*;
get rx line level #*;
wait 3000;
//
set latency 3;
set response 9;
//set priority default;
get response;
get latency;
get priority;
set latency default;
set response default;
//
//For T1 only
tx(sum(whitenoise(-40), amplify(sum (infile("mu-law samples/b52_ula.pcm"), rx
(#1:5)),3db)),#1:6);
//For E1 only
tx(sum(whitenoise(-40), amplify(sum (infile("a-law samples/b52_alaw.pcm"), rx
(#1:5)),3db)),#1:6);
//ENDING THE TASK
end task*;

```

Variation 2:

- op1 is the sum of the speech file and the data received on timeslot 5.
- op2 is sum of the amplified op1 and the white noise of -40dBm.
- op2 the resultant output which is transmitted on time slot 6.

Script:

```

// Connection assumed: connect card1 and card2 from PC 'A' to card1 and card2 of PC 'B'
respectively.
// this script should not be run simultaneously with GUI
//MODE: SEQUENTIAL
//
// User needs to comment the commands according to his requirement
//
//BOARD SETTINGS...
//
//INTERFACE SETTINGS
set rx interface terminate #*;

```

```

//set rx interface monitor #*;
//set rx interface bridge #*;
wait 3000;
//
//FRAME FORMAT SETTINGS FOR T1
set superframe format esf #*;
//set superframe format d4 #*;
//
//FRAME FORMAT SETTINGS FOR E1
//set signaling mode cas #*;
//set signaling mode ccs #*;
//set crc4 on#*;
//set crc4 off #*;
//
//CLOCK SETTINGS
set tx clock source internal #*;
//set tx clock source recovered #*;
//set tx clock source external #*;
//
//LOOPBACK SETTINGS
//set inward driver loopback on #*;
//set outward driver loopback on #*;
set outward driver loopback off #*;
set inward driver loopback off #*;
wait 3000;
//
// - - - - - --Verification of initialization
//CHECKING FORMAT SETTINGS FOR T1
get superframe format #*;
//
//CHECKING FORMAT SETTINGS FOR E1
//get signaling mode #*;
//
//CHECKING OTHER SETTINGS FOR THE CARDS
get tx clock source #*;
get outward driver loopback #*;
get rx line frequency #*;
get rx line level #*;
wait 3000;
//
set latency 3;
set response 9;
//set priority default;
get response;
get latency;
get priority;
set latency default;
set response default;
//
//For T1 only
dspop{ op1 = sum (infile("mu-law samples\b52_ula.pcm"), rx (#1:5)), op2 = sum
(amplify(op1,3db), whitenoise (-40 dbm)), tx (op2 , #1:6));
//For E1 only
dspop{ op1 = sum (infile("a-law samples\b52_alaw.pcm"), rx (#1:5)), op2 = sum
(amplify(op1,3db), whitenoise (-40 dbm)), tx (op2 , #1:6));
//ENDING THE TASK
end task*;

```