AN-003
APPLICATION NOTE

SLIMbus Audio Bridge
Clock Management

LnK
44, rue des Combattants
B-4624 Romsée
Belgium
www.lnk-tools.com
info@lnk-tools.com

Table of Content

1. Purpose of the document

This document will detail the clock topology of a SLIMbus system and will explain the various options that are left to the user. More specifically, it will analyze how to best use the SLIMbus Audio Bridge clock system.

We will use the **LnK** SLIMbus Audio Bridge and the Audio Precision APx585 audio analyzer for the purpose of the exercise.

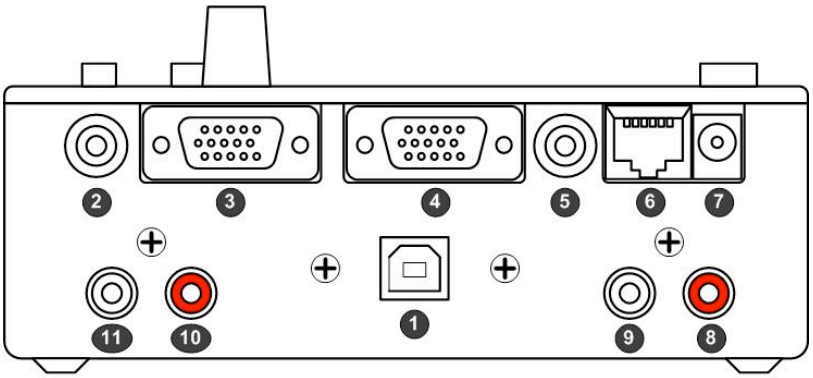
A basic knowledge of the SLIMbus specification and of the operation of the SLIMbus Audio Bridge is required to follow the various explanations given in this document. Refer to the SLIMbus Specification document and to the SLIMbus Audio Bridge HW and SW user manuals for more information.

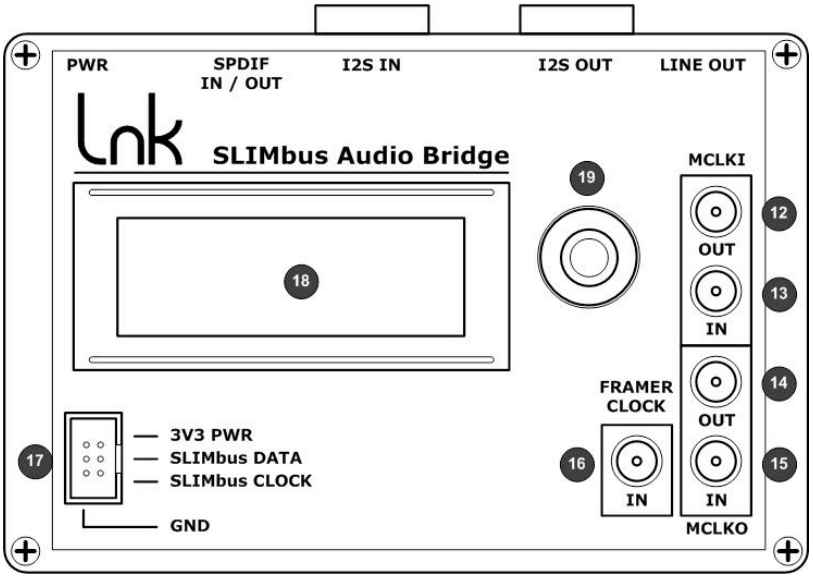
The lecture of the application notes AN-001 and AN-002 relative to the channel setup and the APx configuration is highly recommended.

2. Presentation of the equipments

The tests described in this document will require the SLIMbus Audio Bridge. An audio analyzer will also be needed for the audio tests. Any kind of audio analyzer is suitable. However, the APx500 models from AudioPrecision are especially suited for these tests. For detailed information about these tools, please refer to their respective user manual.

2.1. SLIMbus Audio Bridge





1 - USB connector
2 - Analog line output (mini-jack)
3 - I2S multichannel output
4 - I2S multichannel input
5 - SPDIF IN / OUT (mini-jack)
6 - Factory Service
7 - Power supply
8 - SPDIF OUT
9 - SPDIF IN
10 - Analogue OUT (R)
11 - Analogue OUT (L)

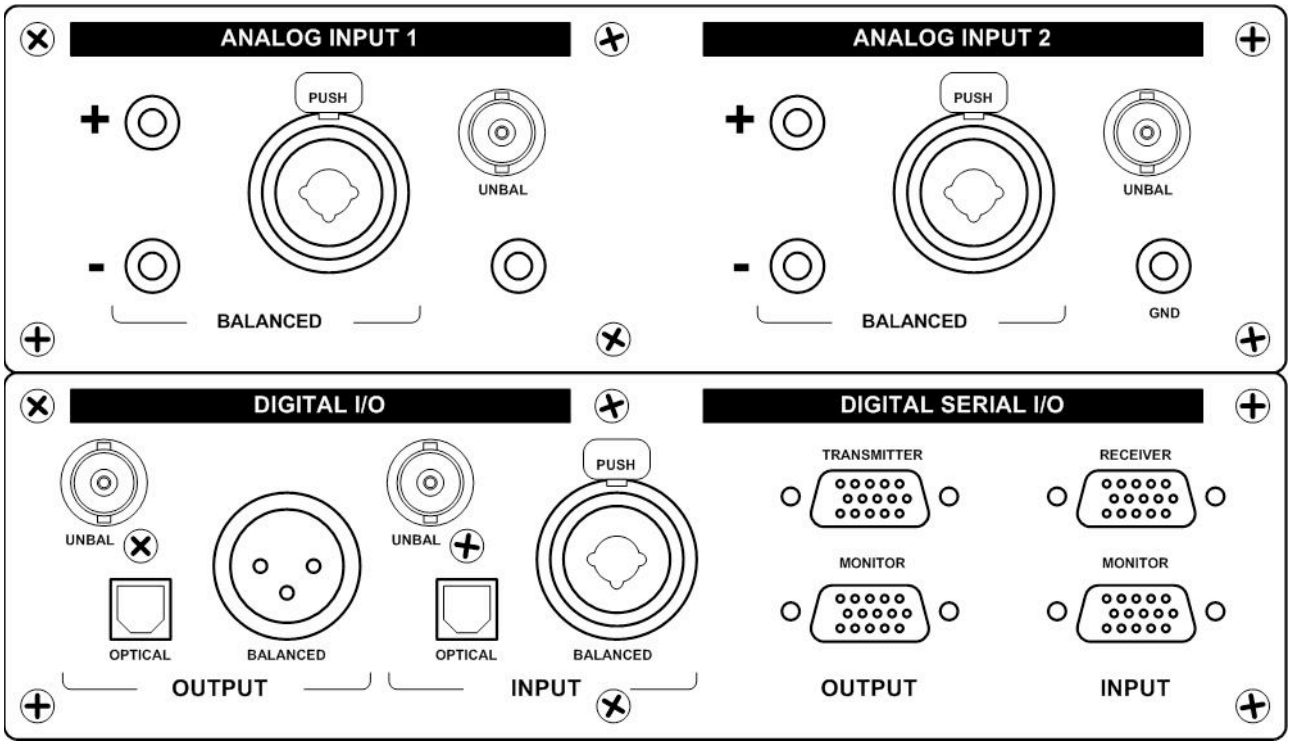
12 - Audio Master IN clock out
13 - Audio Master IN clock in
14 - Audio Master OUT clock out
15 - Audio Master OUT clock in
16 - Framer clock input
17 - SLIMbus connector
18 - Display
19 - Rotary knob + push

The SLIMbus Audio Bridge will transmit (or receive) digital audio streams on (from) SLIMbus. The bridge features SPDIF input and output and I2S multichannel input and output. The I2S interface use the Left Justified data format and 64 bits per frame. The I2S IN/OUT multichannel interface is pin/signal compatible with the Audio Precision DSIO interface of the APx500 machines when using the special cables delivered with the bridge.

The 8 port bridge needs a power supply and needs to be connected to a PC through USB to be able to feed the SLIMbus messages on the bus.

2.2. Audio Analyzer

The audio Analyzer shall provide at a minimum a SPDIF input, a SPDIF output on coaxial connector (unbalanced) and 2 unbalanced analog inputs. For the multichannel tests, a Digital Serial I/O interface (I2S multichannel) is needed.


ANALOG INPUT 1
PUSH
UNBAL
BALANCED
ANALOG INPUT 2
PUSH
UNBAL
GND
BALANCED
DIGITAL I/O
UNBAL
OPTICAL
BALANCED
OUTPUT
UNBAL
PUSH
OPTICAL
BALANCED
INPUT
DIGITAL SERIAL I/O
TRANSMITTER
RECEIVER
MONITOR
MONITOR
OUTPUT
INPUT

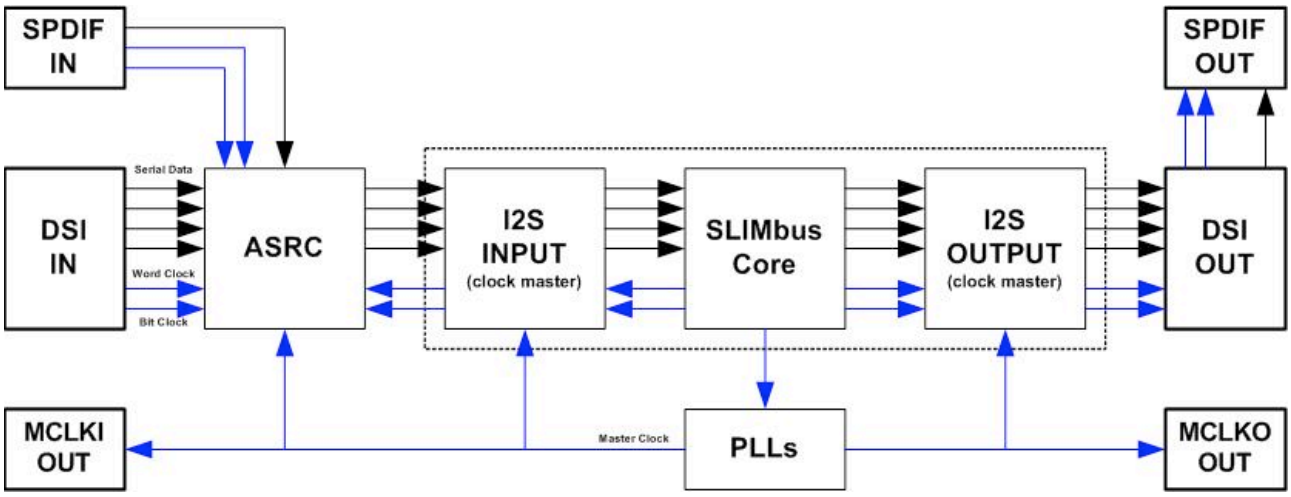
3. Clock Architecture

There are two ways of generating the sampling rate clocks:
- Derived from the SLIMbus clock (using PLLs or fractional dividers)
- Externally fed

Both methods implies different system architecture constraints. However, there is a common rule that applies in every case:

The source and the sink must use the same sampling clock.

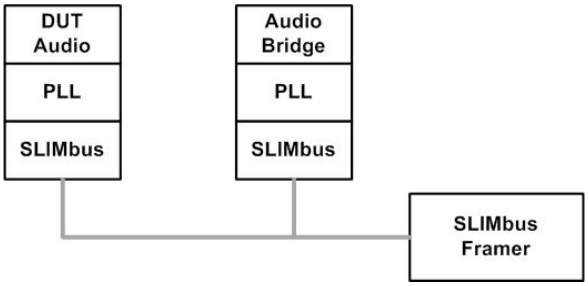
3.1. Sample rate clocks are frequency locked to the SLIMbus clock



The bridge input and output sampling rates are derived from the SLIMbus clock by using a PLL. The audio inputs of the bridge are not necessarily using that sampling rate. The SPDIF and DSI input signals are using their own clocks. Therefore, the Asynchronous Sampling Rate Converters (ASRC) are used to bridge the two clock domains and to compensate for the (slight) difference between the 2 clock frequencies.

Note that the bridge sampling rates being derived from the SLIMbus clock, they will not drift apart from the channel rate. If a SLIMbus channel is set to have 48k segments per second (e.g. channel rate) and has to carry a 48 kHz sampling rate audio stream, the match will be perfect. The channel rate and the sampling rate are frequency locked. This is the typical case where the **Isochronous transport protocol** is used to carry audio samples.

In order to use that mode of operation, the other end of the link must also use a similar sampling clock generation. As long as the ratio between the SLIMbus clock and sampling rate clock is rigorously the same on both ends, the 2 clocks will be frequency locked.

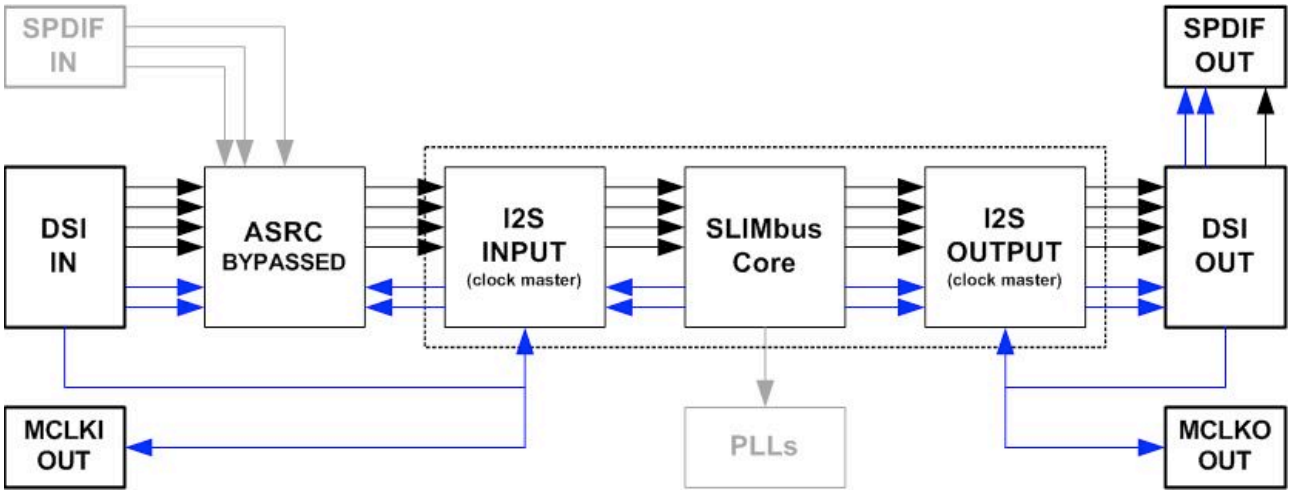


The PLL operation with the ASRCs enabled is handy when no special care is given to the clock structure of the test bench. **It works in all the cases**.
The bridge can be fed with a SPDIF signal or with the serial digital data (DSI - I2S input).

However, it has some limitations when it is about running accurate audio testing of the DUT. The ASRCs are contributing in a very significant manner to the in-band ripple. Amplitude variations of +/- 0.012 dB are observable in the worst cases. Knowing that the oversampling and decimation filters of high end DACs are often given for less than 0.001 dB ripple, the ASRCs will completely hide the DAC performances.
ASRCs also contribute to the overall THD+N figures of the measurement chain. But it is not as critical as the in-band ripple. ASRCs THD+N contribution is most of the time below -125 dB.
The use of the ASRCs does not provide a fully transparent link (e.g. bit copy).

When a transparent link is required, a slightly different setup around the SLIMbus Audio Bridge is necessary. The MCLKI (audio master clock at 256 Fs) output on SMA connector must be used to synchronize the APx585 (or any other signal source used for the test). The transmitter (SPDIF or I2S) and the Audio Bridge are now running on the same clock domain and the ASRCs can be disabled.

3.2. Sample rate clocks are externally fed



In this case, the sample rates and the SLIMbus clock are completely unrelated. The I2S interface master clocks are provided through the DSI (I2S) interfaces or through the SMA connectors (MCLKI IN and MCLKO IN). The rest of the document will only deal with the DSI (I2S) interface as the mechanisms are rigorously identical.

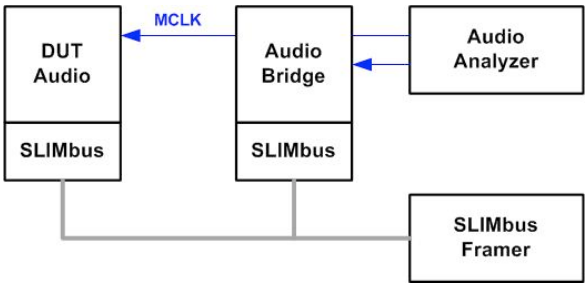
The Bridge and the DUT must both receive the same reference clock. It is mandatory to get an error free audio link.
The Bridge can get MCLKI from the APx585 on the DSI transmitter connector and MCLKO from APX585 on the DSI receiver connector. That guaranties a perfect synchronization between the Audio Bridge and the Audio Analyzer.

As we assumed that the channel rate and the sampling rates are uncorrelated, the transport protocols to be used are either the **Pushed protocol** or the **Pulled protocol**, depending on what protocol is supported by the DUT.

The use of these protocols implies that the channel rate must be greater than the sampling rate, always.

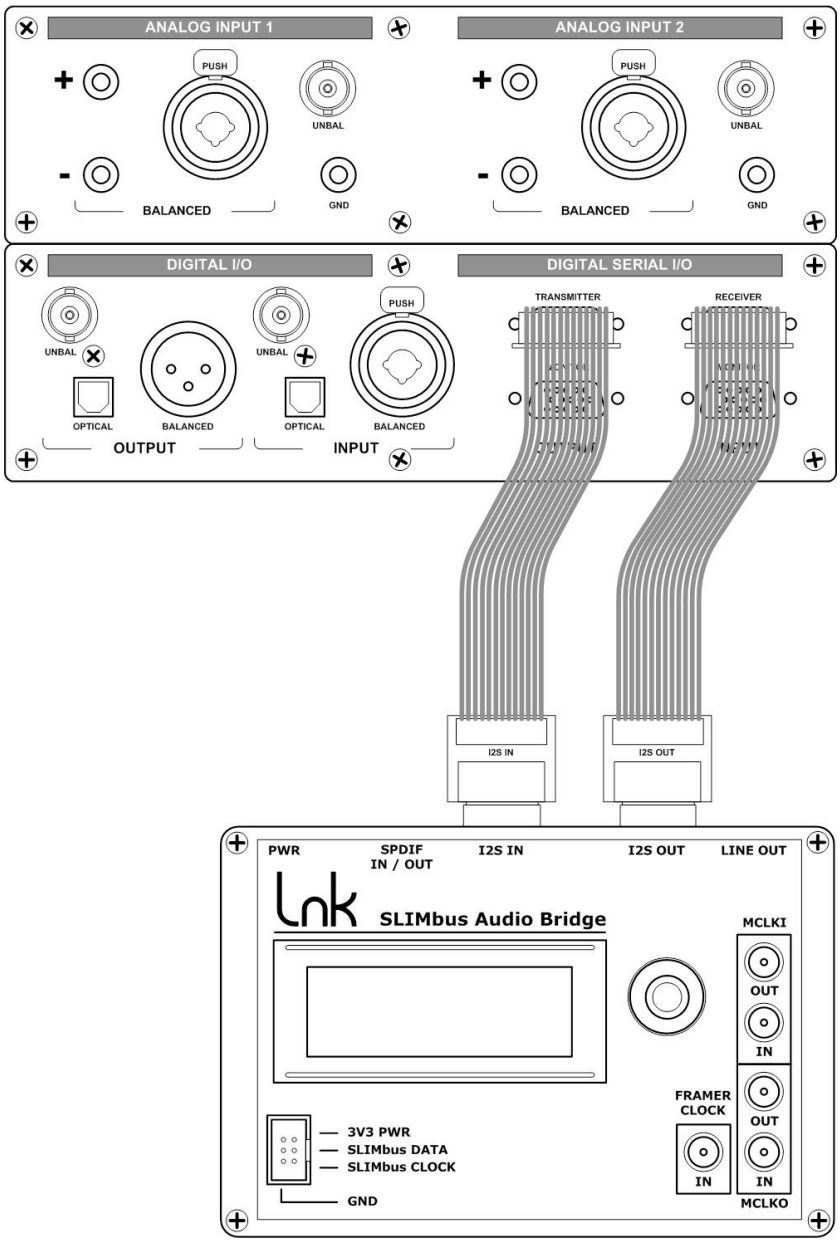
If the channel rate and the sampling rate are both given for 48 kHz, there might be a slight difference in frequency.
For instance, the channel rate could be equal to 47.999 kHz and the sampling rate equal to 48.001 kHz. This situation would result in a loss of 2 samples every seconds (in average). This is not acceptable. So the channel rate must be strictly superior to the sampling rate, say 48.001 kHz.
However, the SLIMbus channel rate can't take any arbitrary values. With a bus running at a root frequency of 24.576 MHz, the closest usable channel rate value is 64 kHz. It is a waste of bandwidth (about 25%), just because the audio clock domain and the SLIMbus clock domain are unrelated. For test purposes, this waste of bandwidth is irrelevant but in a real application, care must be given to avoid this type of situation.

A 26 MHz root frequency would give a channel rate of 50.78125 kHz to effectively carry 48 kHz audio. It is much more efficient and reflects more accurately how SLIMbus can be used.



4. Script examples

The scripts will simply loop through 2 audio channels in the SLIMbus Audio Bridge.



The APx585 is connected to the SLIMbus Audio Bridge via the DSIO interfaces. The Bridge will be controlled by its PC software application. The bridge configuration (clock source, ASRC control, ...) will be set by the mean of SLIMbus messages. The bridge Framer will be used for the purpose of the exercise.

4.1. Isochronous Streaming

We will configure the bridge to operate with the DSI ASRC enabled and to get the sampling clocks from the SLIMbus clock. We will use the Isochronous transport protocol as the channel rate and sampling rate will be identical.

The bridge framer will boot the bus with the Subframe Mode SM=0 (100% control bandwidth) and the Clock Gear CG=9 (Root frequency divided by 2). We use 24.576 MHz

as Root Frequency. Verify that the bridge will use RF=1 (24.576 MHz) as boot frequency before running the script.

0	ASSIGN_LOGICAL_ADDRESS	Dst=0x01C100010000, LA=0x00
1	ASSIGN_LOGICAL_ADDRESS	Dst=0x01C100010100, LA=0x01
2	ASSIGN_LOGICAL_ADDRESS	Dst=0x01C100010200, LA=0x02
3	CHANGE_VALUE	Src=0xFF, Dst=0x02, AM=1, BA=1, SS=0, VU=0xF0
4	WAIT	ms=20
5	CHANGE_VALUE	Src=0xFF, Dst=0x02, AM=1, BA=2, SS=0, VU=0xBA
6	WAIT	ms=20
7	BEGIN_RECONFIGURATION	
8	- NEXT_SUBFRAME_MODE	SM=19
9	RECONFIGURE_NOW	
10	WAIT	ms=5
11	CONNECT_SINK	Src=0xFF, Dst=0x02, CN=0, PN=2
12	CONNECT_SINK	Src=0xFF, Dst=0x02, CN=1, PN=3
13	CONNECT_SOURCE	Src=0xFF, Dst=0x02, CN=0, PN=0
14	CONNECT_SOURCE	Src=0xFF, Dst=0x02, CN=1, PN=1
15	BEGIN_RECONFIGURATION	
16	- NEXT_DEFINE_CHANNEL	CN=0, SD=3140, TP=0, SL=6
17	- NEXT_DEFINE_CONTENT	CN=0, FL=1, PR=3, AF=0, DT=1, CL=0, DL=6
18	- NEXT_ACTIVATE_CHANNEL	CN=0
19	- NEXT_DEFINE_CHANNEL	CN=1, SD=3146, TP=0, SL=6
20	- NEXT_DEFINE_CONTENT	CN=1, FL=1, PR=3, AF=0, DT=1, CL=0, DL=6
21	- NEXT_ACTIVATE_CHANNEL	CN=1
22	RECONFIGURE_NOW	

The first three messages of the scripts will assign a Logical Address to the devices of the Audio Bridge. The Generic Device, the one with the data ports, will get Logical Address **2**.

The next step consists in configuring the Audio Bridge accordingly to the streaming we want to achieve. We use the CHANGE_VALUE messages to modify the bridge parameters.

The Register **1** (also called Value Element 0x001) controls the clock switches and the sample rate converters.

b7	b6	b5	b4	b3	b2	b1	b0
VALIDATE	BH_ENA	DSI ASRC	SPDIF ASRC	MCLKO SEL1	MCLKO SEL0	MCLKI SEL1	MCLKI SEL0
1	**1**	**1**	**1**	**0**	**0**	**0**	**0**

MCLKI_SEL: 0b00 = PLL, 0b01 = DSI, 0b10 = SMA, 0b11 = Reserved
MCLKO_SEL: 0b00 = PLL, 0b01 = DSI, 0b10 = SMA, 0b11 = Reserved
DSI_ASRC: 0b0 = ASRC disabled, 0b1 = ASRC enabled
SPDIF_ASRC: 0b0 = ASRC disabled, 0b1 = ASRC enabled
BH_ENA: 0b0 = Bus Hold disabled, 0b1 = Bus Hold enabled
VALIDATE: 0b0 = Register value ignored, 0b1 = Configuration executed

We want to use the bridge PLLs to generate the sample rate clocks. We will set MCLKI_SEL to 0b00 and MCLKO_SEL to 0b00.
The sample rate converters shall be enabled. We set DSI_ASRC and SPDIF_ASRC.
The SLIMbus bus hold shall be active. We set BH_ENA.
At last, VALIDATE will indicate the bridge what to do with the new register content. We set VALIDATE to force a reconfiguration.

The register **1** content shall then be **0b11110000 = 0xF0**.

The CHANGE_VALUE message has the following parameters:
- AM=1 : Access Mode = Byte Access 1)
- BA=1 : Register address = 1
- SS=0 : Slice Size = Size of the register in bytes = 1 byte
- VU=0xF0 : Value Update = new register content = 0xF0

See the SLIMbus Specification (section 7, page 77) for more information about these parameters.

A delay of 20 ms - **WAIT(ms=20)** - is inserted after the message to let the time to the Audio Bridge to reconfigure.

The Register **2** (also called Value Element 0x002) controls the loopback mode, the DAC signal selection and the DAC output level. Setting-up that register is optional. It will only enable the analog output and SPDIF output with the content of the audio channels. The DAC output level is attenuated by 14 dB (just set arbitrarily).

b7	b6	b5	b4	b3	b2	b1	b0
VALIDATE	DAC ATTN3	DAC ATTN2	DAC ATTN1	DAC ATTN0	DAC_SD SEL1	DAC_SD SEL0	LOOP BACK
1	**0**	**1**	**1**	**1**	**0**	**1**	**0**

LOOPBACK: 0b0 = No internal Loopback, 0b1 = internal loopback enabled
DAC_SD_SEL: 0b00 = SDOUT1, 0b01 = SDOUT2, 0b10 = SDOUT3, 0b11 = SDOUT4
DAC_ATTN: 0b0000 = No Attenuation, ... , 0b1111 = 30 dB attenuation (step 2dB)
VALIDATE: 0b0 = Register value ignored, 0b1 = Configuration executed

The internal loopback is disabled.
The DAC input stream is set on SDOUT2 as the sink ports are the one feeding the I2S stream 2.
DAC_ATTN = 0b0111 = 7, leading to 7 x 2dB = 14 dB attenuation.
At last, VALIDATE will indicate the bridge what to do with the new register content. We set VALIDATE to force a reconfiguration.

The register **2** content shall then be **0b10111010 = 0xBA**.
Note that the use of Register 2 is optional. But it allow use to also stream out the data through the SPDIF output of the bridge. So let's use it.
Again, a delay of 20 ms - **WAIT(ms=20)** - is inserted after the message to let the time to the Audio Bridge to reconfigure.

Now that the bridge HW is reconfigured, we can start configuring SLIMbus to host the two 48 kHz channels.

Some bandwidth needs to be freed in order to create data channels. The Subframe Mode is changed in a reconfiguration sequence to take the value 19. This mode corresponds to 12.5% of the SLIMbus bandwidth dedicated to control and the rest being available for data channels.

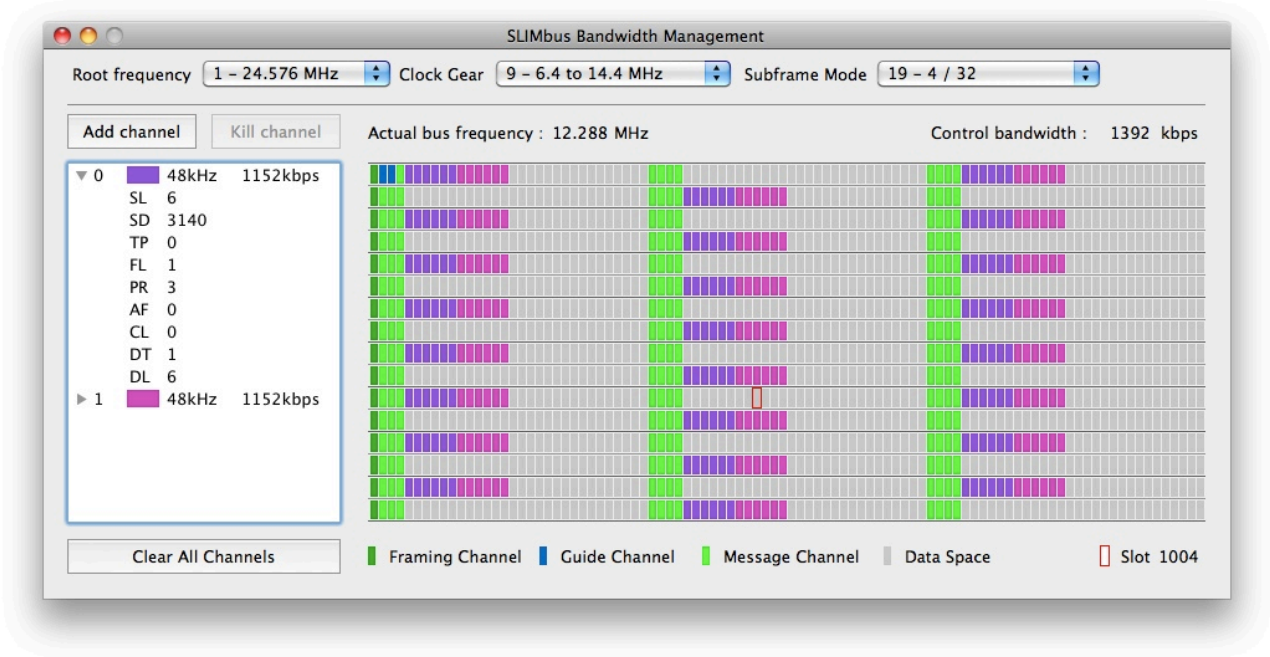
BEGIN_RECONFIGURATION
NEXT_SUBFRAME_MODE(19)
RECONFIGURE_NOW

It is important to make sure that there will only be one and one only reconfiguration sequence in a superframe. It is a requirement of the SLIMbus specification.
Therefore, we added a delay of 5 ms after the reconfiguration sequence messages.
The duration of the superframe at 12.288 mHz is 500 us. Adding a delay of 5ms is more than enough to guarantee that the condition will be met.

The next messages are setting up ports and channels.

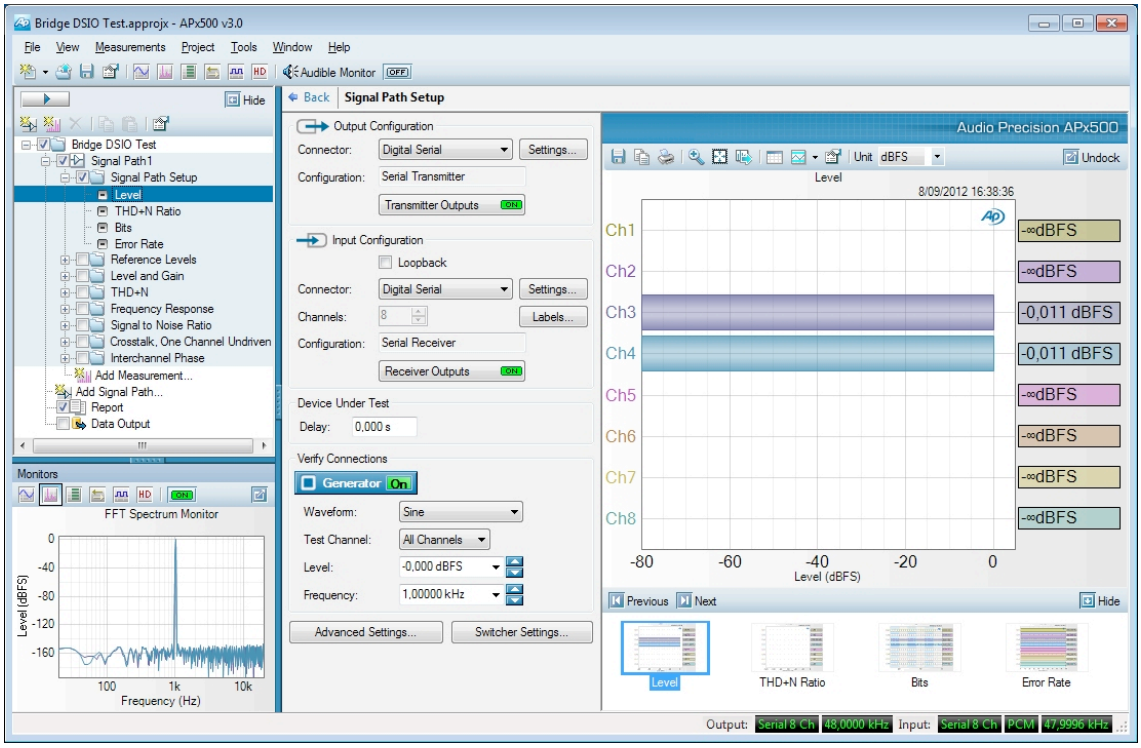
The CONNECT_SINK messages are defining the ports 2 and 3 as **sink** ports.
The CONNECT_SOURCE messages are defining the ports 0 and 1 as **source** ports.

The channels are both having 48kHz rate and are using the **Isochronous** protocol. They use a 24 bits (6 slots) segment.
At the end of the reconfiguration sequence, the channels will be activated and the link between the DSI (I2S) input stream 1 and the DSI (I2S) output stream 2 will be effective.

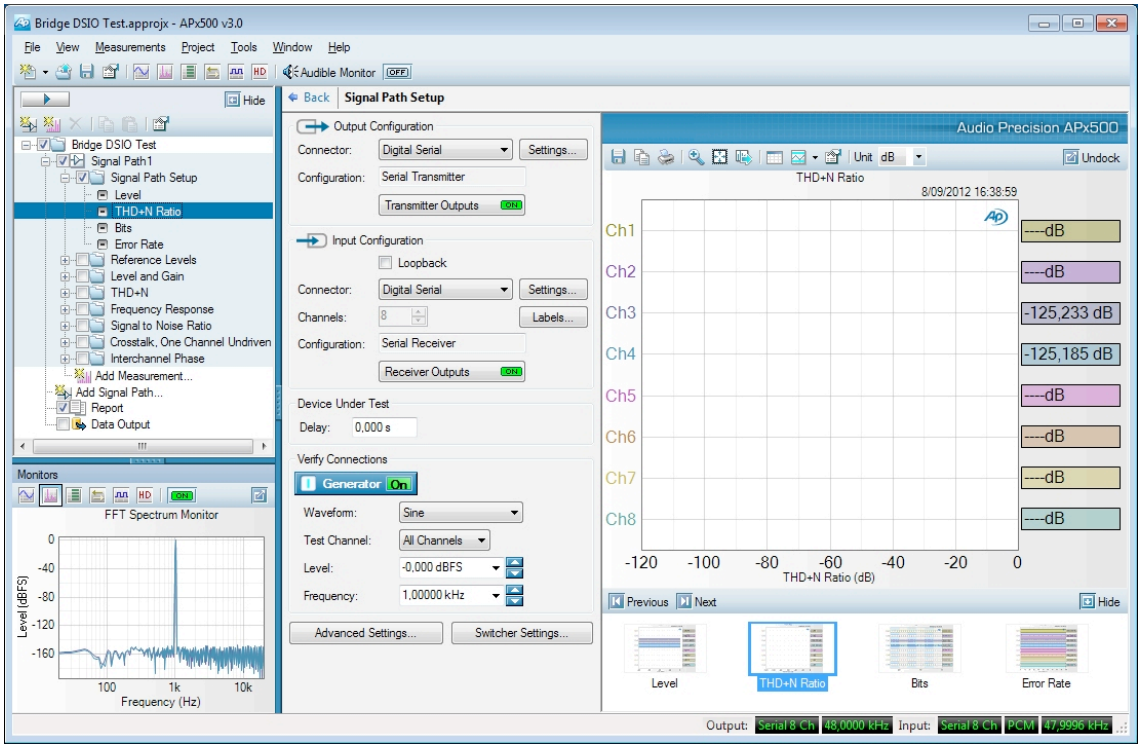


Note that the SPDIF input and SPDIF output could also be used instead of the DSIO interfaces, with the same results.

An audio measurement with the APx585 shows the expected figures for the THD+N and the level.



The frequency ripple of the ASRC gives an attenuation of **-0.011 dB** at 1 kHz. Note that as it is a ripple, the attenuation vary with the ratio *Tone Frequency / Fs* .



The THD+N is equal to -125 dB (without any weighting), which is in line with the specification of the ASRC.

It's interesting to notice that the AP transmitter sampling rate frequency is 48KHz and the AP receiver sampling rate frequency is 47.9996 kHz. Both frequencies are close but not equal. This confirms what was said before about the 2 clock domains and the need for a sample rate converter.

4.2. Pushed Streaming

In this example, the sample rate clocks are **NOT** derived from the SLIMbus clock. The APx will be the clock source of both input and output serial digital interfaces. The script will therefore be a bit different.

```
0   ASSIGN_LOGICAL_ADDRESS    Dst=0x01C100010000, LA=0x00
1   ASSIGN_LOGICAL_ADDRESS    Dst=0x01C100010100, LA=0x01
2   ASSIGN_LOGICAL_ADDRESS    Dst=0x01C100010200, LA=0x02
3   CHANGE_VALUE              Src=0xFF, Dst=0x02, AM=1, BA=1, SS=0, VU=0xC5
4   WAIT                      ms=20
5   CHANGE_VALUE              Src=0xFF, Dst=0x02, AM=1, BA=2, SS=0, VU=0xBA
6   WAIT                      ms=20
7   BEGIN_RECONFIGURATION
8   - NEXT_SUBFRAME_MODE      SM=18
9   RECONFIGURE_NOW
10  WAIT                      ms=5
11  CONNECT_SINK              Src=0xFF, Dst=0x02, CN=0, PN=2
12  CONNECT_SINK              Src=0xFF, Dst=0x02, CN=1, PN=3
13  CONNECT_SOURCE            Src=0xFF, Dst=0x02, CN=0, PN=0
14  CONNECT_SOURCE            Src=0xFF, Dst=0x02, CN=1, PN=1
15  BEGIN_RECONFIGURATION
16  - NEXT_DEFINE_CHANNEL     CN=0, SD=20, TP=1, SL=7
17  - NEXT_DEFINE_CONTENT     CN=0, FL=0, PR=3, AF=0, DT=1, CL=0, DL=6
18  - NEXT_ACTIVATE_CHANNEL   CN=0
19  - NEXT_DEFINE_CHANNEL     CN=1, SD=27, TP=1, SL=7
20  - NEXT_DEFINE_CONTENT     CN=1, FL=1, PR=3, AF=0, DT=1, CL=0, DL=6
21  - NEXT_ACTIVATE_CHANNEL   CN=1
22  RECONFIGURE_NOW
```

We will only analyze the differences with the isochronous streaming script.
Register **1** gets a different value. This time, the ASRCs are disabled. The clock source is set to the DSI interface, both for MCLKI and MCKLO

b7	b6	b5	b4	b3	b2	b1	b0
VALIDATE	BH_ENA	DSI ASRC	SPDIF ASRC	MCLKO SEL1	MCLKO SEL0	MCLKI SEL1	MCLKI SEL0
1	**1**	**0**	**0**	**0**	**1**	**0**	**1**

The register **1** content shall now be **0b11000101 = 0xC5**.
The register **2** content is not modified.

Some bandwidth needs to be freed in order to create data channels. The Subframe Mode is changed from SM=0 to the value 18. We chose a different value than for the isochronous streaming because the channel rate will be different and the Subframe Mode 18 is more suitable for this channel configuration.

BEGIN_RECONFIGURATION
NEXT_SUBFRAME_MODE(18)
RECONFIGURE_NOW

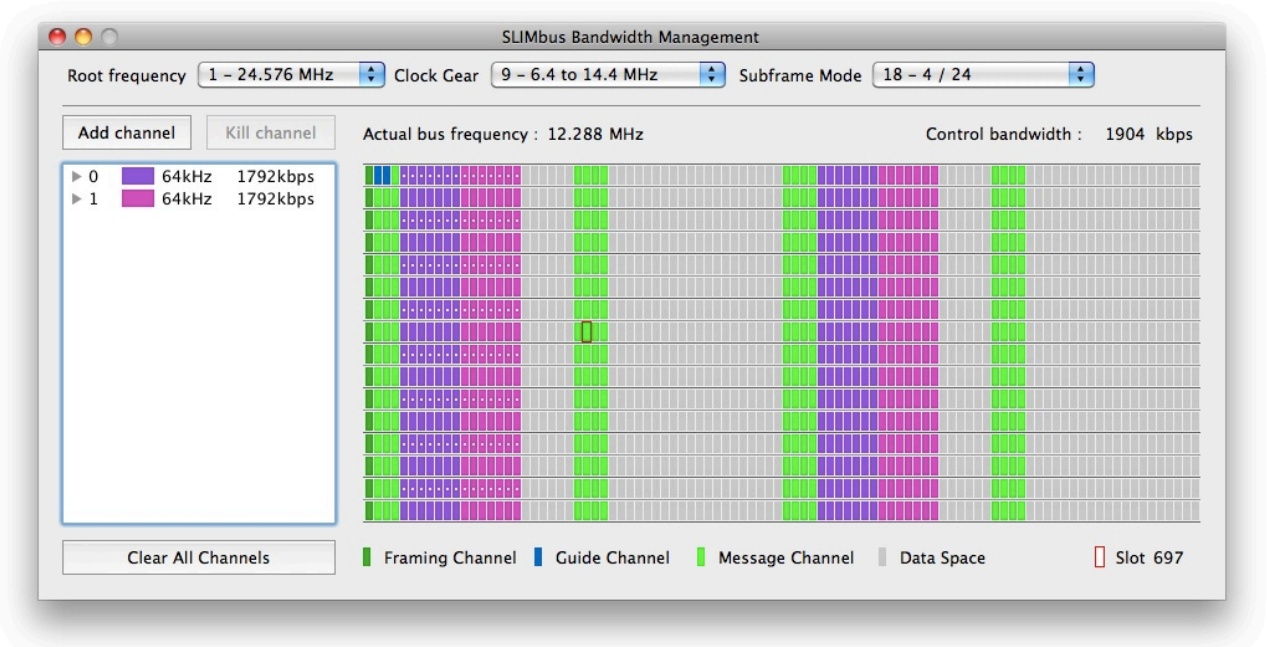
The port assignment is identical to the isochronous streaming script.

The biggest difference is in the channel definition. As the SLIMbus clocks and the sampling rate clocks are not correlated anymore, we must make sure that the peak rate of the audio streams will **ALWAYS** be smaller than the SLIMbus channel rate.
With the Root Frequency 1 (24.576 MHz), the closest but higher rate to 48 kHz is equal to 64 kHz. So we will create 2 channels with a rate of 64 kHz to carry audio streams having an average rate of 48 kHz.

The protocol used is the **Pushed** protocol. Note that using the **Pulled** protocol would have had the exact same result.
As we need to add one TAG slot to the segment, the segment size will grow from 6 slots to 7 slots.

▼ 16	- NEXT_DEFINE_CHANNEL	CN=0, SD=20, TP=1, SL=7
	Channel Number (CN)	0
	Segment Distribution (SD)	20
	Transport Protocol (TP)	1 - Pushed Protocol
	Segment Length (SL)	7
▼ 17	- NEXT_DEFINE_CONTENT	CN=0, FL=0, PR=3, AF=0, DT=1, CL=0, DL=6
	Channel Number (CN)	0
	Frequency Locked (FL)	0 - Frequency unlocked
	Presence Rate (PR)	3 - 48kHz
	AUX format (AF)	0 - Not applicable
	Data Type (DT)	1 - LPCM audio
	Channel Link (CL)	0 - Channel not linked
	Data Length (DL)	6

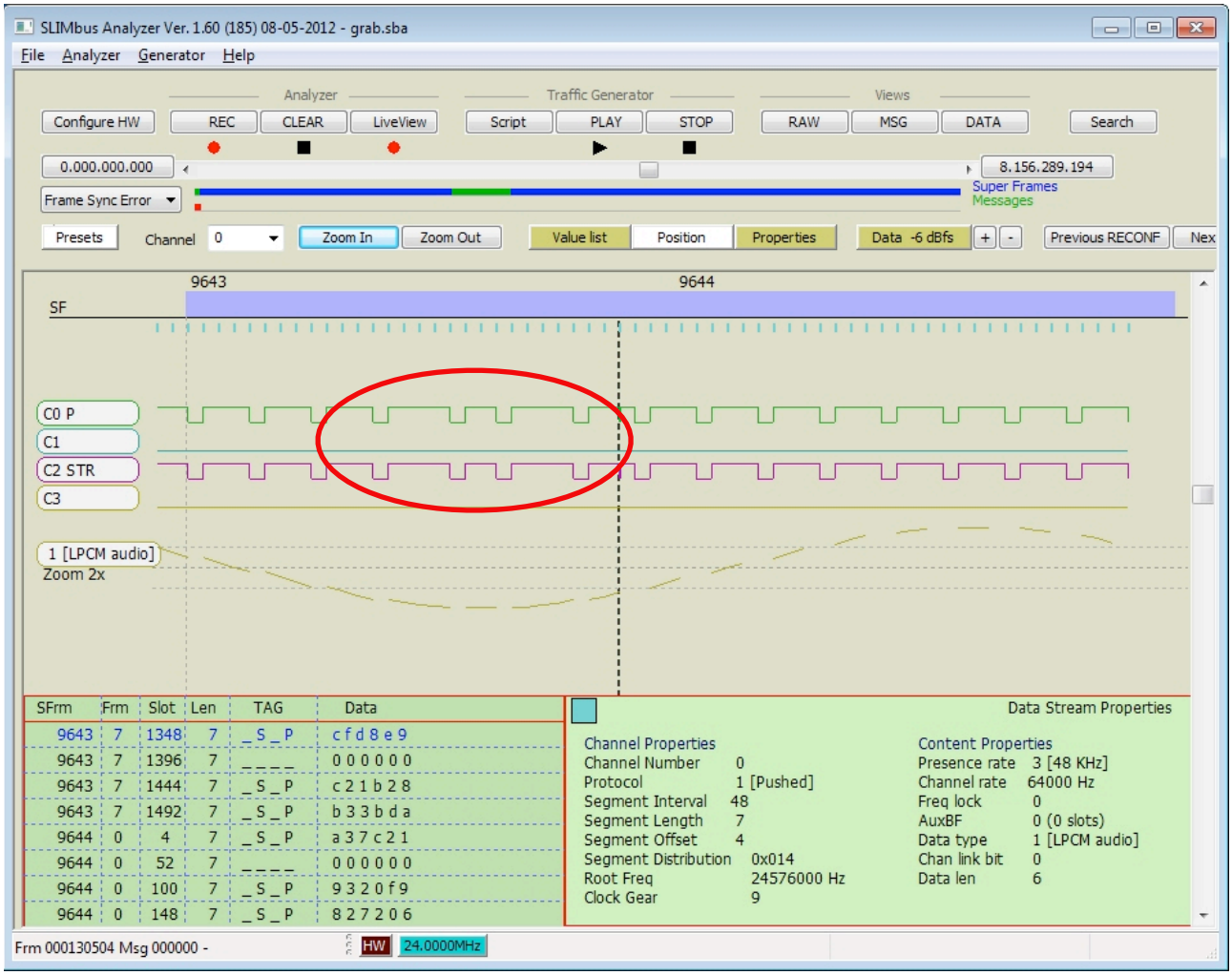
The sample size will remain 24 bits (6 slots). The Presence Rate (PR) is set to 48 kHz.



We are now getting three valid segments out of four.
It corresponds to the ratio 48kHz / 64kHz.

However, as we know that the sample clocks and the SLIMbus clock are not synchronized anymore, we can expect some hick-ups in that 3/4 scheme.

A capture of the data channels with the SLIMbus Protocol Analyzer is indeed showing such things.

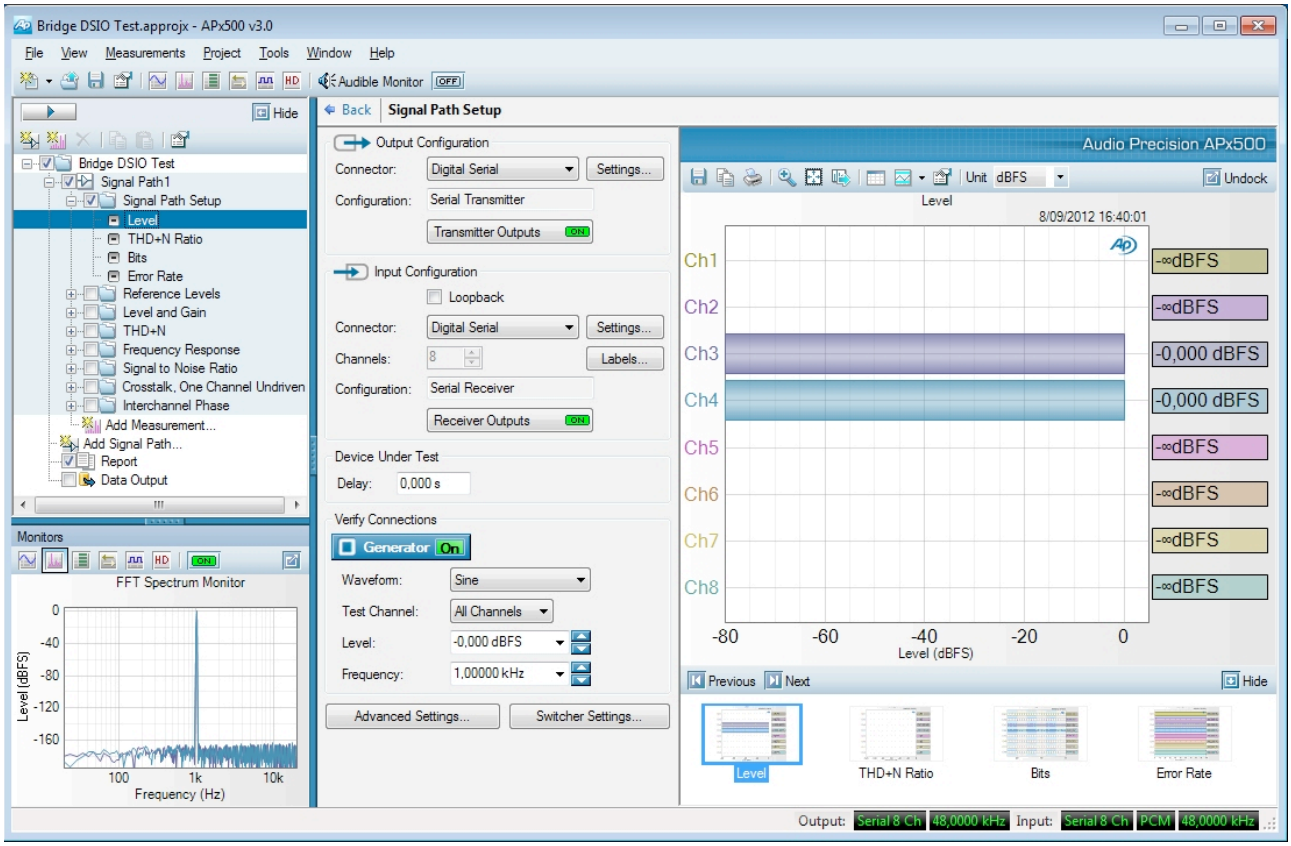


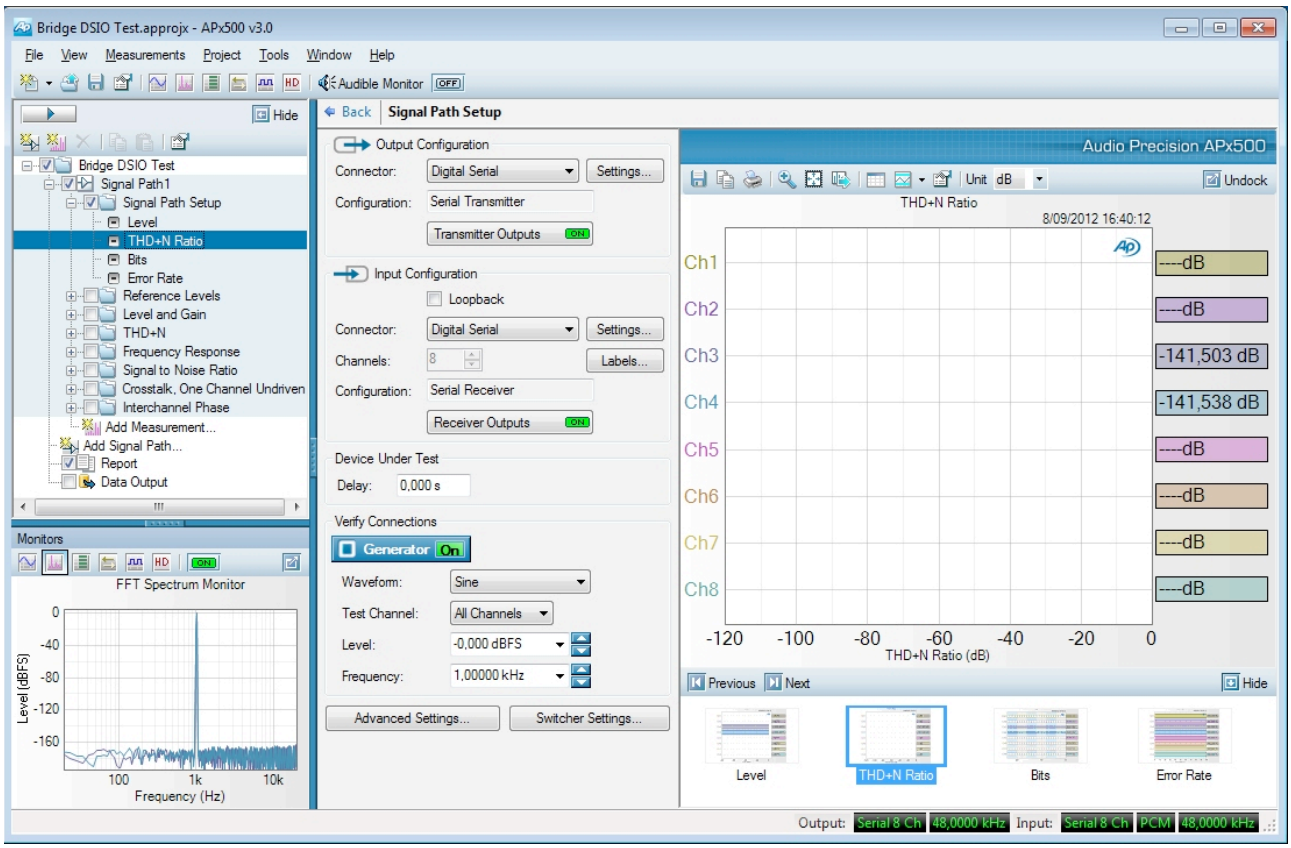
We can see on the capture that 5/4, 2/3, 5/4 events happened in the regular 3/4 pattern. This is because the clocks are not exactly what they are supposed to be. This hick-up happens every few seconds (corresponding to around 5000 superframes in this setup). These irregularities in the P bit pattern are actually compensating the clock frequency difference that could have caused a buffer overflow or under-run.

Now, let's have a look to the audio measurements.
The Level is reported to be exactly **0** dBFs. No attenuation anymore.
The THD+N level (unweighted) is below **-141** dB.
The audio link is now completely transparent.

Note that the SPDIF input is normally **NOT** usable with this example, as the APx585 DSIO clock outputs are disabled when the Unbalanced Digital (SPDIF) output is selected.

The output and input sample rate frequencies are now logically shown by the analyzer to be equal to 48 kHz.


Bridge DSIO Test.approjx - APx500 v3.0
File View Measurements Project Tools Window Help
Audible Monitor OFF
Hide
Bridge DSIO Test
Signal Path1
Signal Path Setup
Level
THD+N Ratio
Bits
Error Rate
Reference Levels
Level and Gain
THD+N
Frequency Response
Signal to Noise Ratio
Crosstalk, One Channel Undriven
Interchannel Phase
Add Measurement...
Add Signal Path...
Report
Data Output
Monitors
FFT Spectrum Monitor
Level (dBFS)
Frequency (Hz)
Back
Signal Path Setup
Output Configuration
Connector: Digital Serial
Settings...
Configuration: Serial Transmitter
Transmitter Outputs ON
Input Configuration
Loopback
Connector: Digital Serial
Channels: 8
Labels...
Configuration: Serial Receiver
Receiver Outputs ON
Device Under Test
Delay: 0,000 s
Verify Connections
Generator On
Waveform: Sine
Test Channel: All Channels
Level: -0,000 dBFS
Frequency: 1,00000 kHz
Advanced Settings...
Switcher Settings...
Audio Precision APx500
Unit dBFS
Undock
Level
8/09/2012 16:40:01
Ch1 -∞dBFS
Ch2 -∞dBFS
Ch3 -0,000 dBFS
Ch4 -0,000 dBFS
Ch5 -∞dBFS
Ch6 -∞dBFS
Ch7 -∞dBFS
Ch8 -∞dBFS
Level (dBFS)
Previous Next
Hide
Level
THD+N Ratio
Bits
Error Rate
Output: Serial 8 Ch 48,0000 kHz Input: Serial 8 Ch PCM 48,0000 kHz


Bridge DSIO Test.approjx - APx500 v3.0
File View Measurements Project Tools Window Help
Audible Monitor OFF
Hide
Bridge DSIO Test
Signal Path1
Signal Path Setup
Level
THD+N Ratio
Bits
Error Rate
Reference Levels
Level and Gain
THD+N
Frequency Response
Signal to Noise Ratio
Crosstalk, One Channel Undriven
Interchannel Phase
Add Measurement...
Add Signal Path...
Report
Data Output
Monitors
FFT Spectrum Monitor
Level (dBFS)
Frequency (Hz)
Back
Signal Path Setup
Output Configuration
Connector: Digital Serial
Settings...
Configuration: Serial Transmitter
Transmitter Outputs ON
Input Configuration
Loopback
Connector: Digital Serial
Channels: 8
Labels...
Configuration: Serial Receiver
Receiver Outputs ON
Device Under Test
Delay: 0,000 s
Verify Connections
Generator On
Waveform: Sine
Test Channel: All Channels
Level: -0,000 dBFS
Frequency: 1,00000 kHz
Advanced Settings...
Switcher Settings...
Audio Precision APx500
Unit dB
Undock
THD+N Ratio
8/09/2012 16:40:12
Ch1 ----dB
Ch2 ----dB
Ch3 -141,503 dB
Ch4 -141,538 dB
Ch5 ----dB
Ch6 ----dB
Ch7 ----dB
Ch8 ----dB
THD+N Ratio (dB)
Previous Next
Hide
Level
THD+N Ratio
Bits
Error Rate
Output: Serial 8 Ch 48,0000 kHz Input: Serial 8 Ch PCM 48,0000 kHz