Sample clock jitter and real-time audio over the IEEE1394 high performance serial bus.

Julian Dunn
Nanophon Limited
Cambridge, England
Email: njd@nanophon.com
http://www.nanophon.com

ABSTRACT

The asynchronous clocks within the IEEE1394 high performance serial bus present challenges for sample timing recovery in conjunction with the transmission of digital audio. Moreover the systematic nature of the jitter produced means that prototype systems are unlikely to produce worst-case performance. This is analysed and, with jitter audibility models developed elsewhere, a sampling jitter attenuation requirement is estimated.

1.0 INTRODUCTION

The IEEE1394 High Performance Serial Bus [1, 2, 3] has been used in the specification of the new A/M protocol - IEC-PAS 61883-6 [4,5,6]. This has the capability to transmit 24 bit word length, 96kHz sampling frequency multi-channel audio and so, potentially, it is a very high quality interface format.

However in addition to the accurate transmission of sample data, a complete real-time digital audio connection also needs to pass timing information, such as a sample synchronisation signal, to define the sampling instant at which the data is valid. Error in this timing, or jitter, is known to distort the signal at the point of conversion of the data to or from the analogue domain.

The IEEE1394 format uses asynchronous clocks at each node. The interaction of these clocks with each other and with the sample (word) clock generates jitter. This paper uses a simulation of a single bus IEEE1394:1995 network to examine the nature of this jitter.

As multi-channel high quality digital audio is becoming more common there is a requirement for an appropriate digital interface format. The IEC-PAS-61883-6 format for using IEEE1394 has been proposed as satisfying this need.

Does it ?

2.0 JITTER

Jitter is the variation in the timing of a periodic event - such as a signal transition - from an ideal timing that the event would have if were perfectly regular. For example, a perfect jitter-free square wave has an exactly constant time delay between transitions. In practice each transition of a real square wave, with exactly the same mean frequency will occur slightly before or after the ideal. This variation is called jitter.

In this paper the term jitter is used to mean:

> "Deviation in timing of transitions when measured with respect to an ideal clock."

There can be confusion between levels of jitter measured in this way and measurements of variations in the time between successive transitions of a clock. For example in examination of a 48kHz sample clock it may be possible to measure a variation in the period of, say, 500ps when the variation from the ideal clock timing

is an integration of these period errors, which after 20 successive periods may be over 10ns. The simulation later in the paper gives an example of the large difference that can often exist between this period variation measure (sometimes called edge to edge jitter) and the “deviation from an ideal clock”. measure.

It is important to use the correct measurement for jitter when calculating the modulation effect it will have.

2.1 Sampling Jitter

In this paper the term sampling jitter is applied to the variation in the timing an audio signal through jitter in an analogue to digital (ADC), digital to analogue (DAC), or asynchronous sample rate converter (ASRC). In the former two cases this can often be associated with an observable clock signal but in an ASRC it may be a totally numerical process as the samples of a signal are regenerated to correspond with new sampling instants.

Jitter will only affect the audio signal contents when it is being sampled or re-sampled. This occurs when a signal is passing from the continuous time domain to the sampled-signal domain in an ADC, from the sampled-signal domain to the continuous time domain in a DAC, or while remaining in the sampled-signal domain but with the sampling intervals re-determined such as in the ASRC.

The effect of sampling jitter is to modulate the signal being sampled. This modulation causes unwanted modulation products to be produced. This may produce an undesirable change - particularly if the products may be perceived as making an audible difference. In some cases the signal with jitter is preferred but as the effect is often uncontrolled it is generally felt to be undesirable.

The amplitude of the jitter modulation products is proportional to the amplitude of the jitter (where jitter is defined according to the previous section), and the rate of change of the signal that is being affected by the jitter. For an audio tone of frequency f and sinusoidal jitter of peak amplitude J the modulation sidebands produced are at a relative level (with respect to the audio tone) of $20 \log(\pi f J)$, derived in [7].

For example with sinusoidal jitter of 10ns rms (14ns peak) on a 1kHz tone the level of each sideband will be -87dB. The same jitter on a 10kHz tone will be at -67dB with respect to the tone.

Of course real jitter and signals are not sinusoidal. However accurately this illustrates the magnitudes of the effect.

It should be noted that some delta-sigma converters with high levels of ultrasonic noise crossing between the sampled-signal and continuous-time domains have the problem that jitter modulation of the ultrasonic noise causes the audio band noise floor to be raised. However most integrated converters of this type filter out the ultrasonic noise using switched capacitor filters in the sampled domain to avoid this.

2.2 Sampling Jitter Audibility

A recent paper [8] describes practical research that found the lowest jitter level at which the jitter made a noticeable difference to be about 10ns rms. This was with a high level test sine tone at 17kHz. With music none of their subjects found jitter below 20ns rms to be audible.

In [7] the author developed a model for jitter audibility based on worst case audio single tone signals and including the effects of masking. This concluded:

> “Masking theory suggests that the maximum amount of jitter that will not produce an audible effect is dependent on the jitter spectrum. At low

frequencies this level is greater than 100ns, with a sharp cut-off above 100Hz to a lower limit of approximately 1ns (peak) at 500Hz falling above this frequency at 6dB per octave to approximately 10ps (peak) at 24 kHz for systems where the audio signal is 120dB above the threshold of hearing."

In the view of the more recent research cited above this may be considered to be over cautious. However the indication that jitter below 100Hz is more than 40dB less audible than jitter above 500Hz is useful when determining the properties of jitter attenuation devices.

2.3 Acceptable Levels Of Sampling Jitter

The market for higher quality audio equipment and the association of very low levels of jitter with audio quality appears to require devices to try to have sampling jitter level commensurate with producing modulation products below the levels corresponding with the quantisation noise of the system. Audibility criteria may not be an issue in the marketplace.

For this reason sampling jitter levels that may be derived from the interface may need to approach lower levels than 10ns.

For example it may be important that devices can have a full scale total harmonic distortion and noise (THD+N) performance of at least 100dB. This would imply sampling jitter levels of below 1.6ns rms (for a conventional 1kHz tone stimulus).

3.0 IEEE 1394 HIGH PERFORMANCE SERIAL BUS

The IEEE1394 bus has many applications including the interconnection of computers, computer peripherals, and A/V equipment. It is an asynchronous serial bus that supports real time data transmission on isochronous channels. Individual nodes on the bus are connected by one-one links. Each node has a 24.576MHz clock that has to be within 100ppm of that nominal rate.

The 1995 version of the standard [1] defines 3 speed grades with data bit rates that are multiples of this clock. These are nominally 98.304MHz, 196.608MHz and 393.216MHz for speeds known as S100, S200, and S400.

The bus is extended by nodes having more than one port. A node will re-transmit packets received on one port through other ports. This re-transmission is synchronised to a free-running clock on that node with the result that the repeater data delay varies as a sawtooth due to the changing phase of the local clock with respect the timing of the incoming data.

This variation in repeater data delay is called, in this paper, repeater jitter. How is it specified?

In IEEE1394:1995 the only restriction to the repeater data delay is implied by the only allowable value for the PHY_DELAY parameter being one that indicates that the delay is no larger than 144ns.

IEEE1394:1995 page 87, Part of table 4-29 — Self-ID packet fields

Field	Derived from	Comment
del	PHY_DELAY	worst case repeater data delay:
		00 ≤ 144 ns (~14/BASE_RATE)
		01 reserved
		10 reserved
		11 reserved

There is no reference to a minimum delay in that document. The draft P1394a supplement has more information:

P1394a Draft 2.0, page 90:
Table 7-14 — Cable interface timing constants

Timing constant	Minimum	Maximum	Comment
PHY_DELAY	0.06 µs	See PHY registers	Best-case repeater data delay has a fixed minimum.

The supplement also provides for indication of the PHY jitter but provides no limits for this (beyond the limit of 84ns implied by the difference between maximum and minimum delays).

P1394a Draft 2.0, page 70:
Table 6-1 — PHY register fields for the cable environment

Field	Size	Type	Power reset value	Description
Delay	4	r	vendor-dependent	Worst-case repeater delay, expressed as 144 + (delay * 20) ns
Jitter	3	r	vendor-dependent	The difference between the fastest and slowest repeater data delay, expressed as (jitter + 1) * 20 ns.

These specifications relate to the repeater jitter at each node. The number of nodes that can be cascaded is also relevant to the jitter build up.

There is an upper limit on bus size of 63 nodes. (With the use of bridges to connect buses together this does not represent an upper limit on the number of nodes in a network; but that is beyond the scope of this paper.) The timing requirements mean that these cannot be all in one cascade.

Some of the references [page 127 of 3, 9, and 10] have asserted that the maximum number of hops permitted is 16. This is a conclusion may have been inferred from the original standard but it is not a normative requirement. The P1394a draft supplement [2] makes this point on page 21. The maximum hop count depends on several parameters including the cable length but primarily the PHY_DELAY of the nodes. For 1m cable lengths and a 65ns maximum PHY_DELAY more than twice this number could be cascaded.

Table C-2 on page 166 of P1394a also indicates that in circumstances “where the bus manager knows that the maximum cable length used in the topology is 4.5 meters and that the maximum PHY delay is 0.144 ms” the default and maximum gap count permits up to 23 hops. (This table corrects errors “known to exist” in the informative clause E.1 of IEEE Std 1394-1995.)

4.0 IEC61883-1 CONSUMER AUDIO/VIDEO EQUIPMENT - DIGITAL INTERFACE

IEC 61883-1 defines a method of transmission of real-time digital audio/video data over IEEE1394. In particular it defines a common isochronous packet (CIP) format that standardises support for real time audio and video streams.

The CIP format provides for a time stamp (SYT) based on the lower 16 bits of the IEEE1394 CYCLE_TIME register of the transmitter. The SYT may be used to determine the presentation time of the signal at the receiver. The SYT has the same resolution as the CYCLE_TIME register of 1/24.576µs, or 40ns.

5.0 IEC-PAS 61883-6 A/M PROTOCOL

IEC-PAS 61883-6 “Specification For The Transmission Of Audio And Music Data” or Audio/Music (A/M) Protocol is a specification for the transmission of iscochronous music and audio streams using the CIP format. This was initially developed by Yamaha and was published by the IEC and the 1394 Trade Association as a “dual logo” publicly available specification in 1998. It is currently in the process of being developed into an international standard (when it would have the same status as IEC61883-1). This paper is intended as a contribution to that development.

6.0 IEC-PAS 61883-6 SAMPLE CLOCK SYNCHRONISATION

The free-running 24.576 MHz clock on each node is used to increment a CYCLE_TIME register on each node. The node defined as the bus Cycle Master transmits a cycle start packet at intervals of 125us (8kHz). This defines the start of the IEEE1394 isochronous cycle. This packet has a value that allows the other nodes on the bus to align their CYCLE_TIME registers to correct any drift due to the slightly different clock frequencies. As the cycle start packet is transmitted on the bus it suffers from reclocking jitter and, in addition, any correction has to be applied with the resolution of the 24.576MHz clock so there is jitter on the CYCLE_TIME register alignment.

In some circumstance it is possible to synchronise digital audio signals to the cycle master. In many circumstances that would not be possible. The 61883-6 protocol requires the SYT field in the CIP to define the presentation time of the signal. This allows the transmission of timing information that supports sample rates that are not multiples of 8kHz or are not synchronous with the cycle master node [11]. Figures 1 and 2 shows the flow of timing information that provides this flexibility.

It should be noted that the published IEC-PAS 61883-6 apparently allows consumer or cost sensitive receivers to ignore the SYT. (This may be taking the text on page 7 of [5] too literally.) Alternative methods of synchronisation are described in [10] but are beyond the scope of this paper.

IEC-PAS 61883-6 [5] has an informative annex on synchronisation which states that

> “A receiver can reproduce the "synchronization clock" in terms of the pulse generated when the SYT equals its own CYCLE_TIME. The resolution of the time stamp is 1/(24.576 MHz), or approximately 40ns, and CYCLE_TIME may have 40 ns of jitter.”

The 40ns figure is the resolution of the CYCLE_TIME register rather that the jitter of the CYCLE_TIME register. If that was corrected then the statement would be true. However in the context of an informative annex about synchronisation it would not be very useful without an indication of the jitter introduced by the variable repeater delay through intermediate nodes.

A more accurate statement may be:

> “A receiver can reproduce the "synchronisation clock" in terms of the pulse generated when the SYT equals its own CYCLE_TIME. The resolution of the time stamp and of CYCLE_TIME adjustment values is 1/24.576 μs, or approximately 40ns. Quantisation of the time stamp, and the quantisation of the CYCLE_TIME adjustment for the receiver and the transmitter will produce jitter. If the cycle master, synchronisation source, and receiver are on different nodes then these three quantisations will add 120ns of peak to peak jitter to the reproduced synchronisation clock. This is in addition to the jitter introduced by the variable repeater data delay between the cycle master and the receiver, and between the cycle master and the transmitter”

7.0 JITTER ANALYSIS SIMULATION

7.1 Topology

A simulation was developed to analyse the mechanisms for the introduction of jitter into an IEEE1394 based IEC61883-6 digital audio stream. The network topology used is shown in figure 3. This has 15 repeaters in the paths between each of the three nodes of interest, cycle master, (sample synchronisation) transmitter, and (sample synchronisation) receiver.

The figure 3 topology was chosen as the same as that used in [9]. It was called a worst case topology in that paper.

For the reasons stated above in connection with the IEEE1394 specification this does not appear to be the case. It compares with the following similarly shaped topologies but with more intermediate hops:

1. With 4.5m cable and 144ns repeater delays - 22 hops are permitted between each pair of nodes.

2. In the extreme case with the shorter repeater delays of faster PHY devices and shorter cables (as would be used in an equipment rack, for example) a maximally sized bus of 63 nodes could be fitted to the same shape. This would leave a functional bus with 42 repeaters from the cycle master to the transmitter and receiver nodes.

7.2 Conditions

The simulation model uses the following parameters:

CYCLE_COUNT adjustment resolution = 1/24.576 μs

CYCLE_COUNT resolution = 1/24.576 μs

time stamp quantisation error = 1/24.576 μs

repeater jitter = 2/(data bit rate)

≅ 20ns for the S100 simulation

≅ 5ns for the S400 simulation

These first three quantities are as defined by IEEE1394:1995 while the repeater jitter is based on the value used in [9] (which was assumed to be S100). The S400 value is scaled down from this. As explained in section 3.0 the repeater jitter is not well defined in the standard so these values are not worst case.

The simulation was repeated to gather sets of results for the following conditions:

1. All nodes of S100 speed and the frequency differences between nodes selected by a Gaussian distribution random number generator and resulting in a standard deviation of 30ppm.

2. As 1 but with the frequency differences between nodes reduced by a factor of ten to a standard deviation of 3ppm.

3. As 1 but all nodes of S400 speed.

The simulation was performed using a mathematical spreadsheet package, MathCad. Details of this are shown in the annex to this paper. An explanation of the simulation is interspersed with the equations and results in that annex.

The simulation includes an examination of the effect of the jitter on a simple audio signal. A clock multiplying PLL is modelled with a 3rd order response and a corner frequency at 500Hz. The output of this PLL simulation is used to modulate the

timing of an audio test signal. A Fourier transform of the modified audio signal is presented and the signal to noise ratio resulting from this jitter modulation is calculated.

The main simulation results are tabulated here:

	PHY speed and frequency deviation		
	S100 (30ppm)	S100 (3ppm)	S400 (30ppm)
Jitter (ns), refer to:	A.8	A.9	A.10
SYT match rms	38	38.5	22
SYT match peak (over time=1s)	160	145	75
Word clock rms (after PLL)	12	21	6.2
Period variation (ns), refer to:	A16.1 and A16.2		
SYT match rms	51.3	43.6	27.5
Word clock rms	0.43	0.50	0.30
Audio performance (with 1kHz tone)	A.11,A.12	A.13	A.14
Signal to noise ratio	82.6 dB	77.5 dB	88.1 dB
Peak spurious component	-95 dB	-88 dB	-95 dB
at frequency offset	+/- 400Hz	+/- 40Hz	+/- 400Hz

In addition to these results it is apparent from examination of the spectra that the jitter products are based on discrete tonal components. These include some at an amplitude of 9ns rms that correspond with the beat frequencies of the clocks directly determining the quantisation of the two CYCLE_COUNT adjustments and of the time stamp. These three quantisations all have a magnitude of 40ns peak-peak. (A sawtooth of peak to peak amplitude 40ns contains a fundamental amplitude of approximately 9ns rms)

8.0 DISCUSSION

8.1 Comparison With Published Jitter Figures

8.1.1 Fujimori and Kakiuchi [12] report

> "the jitter of the cycle out pulse generated by a link chip is around 1ns when the observed node is not a cycle master."

This statement is difficult to understand without concluding that what is being measured is not the sort of jitter that is being discussed in this paper.

8.1.2 Moses and Bartlett [10] reported the worst case difference between the CYCLE_TIME registers of sender and receiver as "25ns over one isochronous cycle".

This figure is the deviation of cycle start period between any two nodes where each node may have a frequency error of 100ppm. The cycle start period is nominally 125us so a deviation of 100ppm corresponds with 12.5ns. This doubles up to 25ns when you consider that one node could have the deviation in the opposite sense to the other.

8.1.3 Kuribayashi et al. report [9] jitter in the SYT match clock of 38.9 ns RMS, with a peak to peak range of 367 ns, and on the word clock of a simple PLL locked to the SYT match clock they report jitter of only 548 ps RMS. This is with the same topology as used in the simulation.

The SYT match rms jitter value is apparently very similar to the jitter results of the simulation for S100 nodes and 30ppm node clock frequency spread.

The oscilloscope plot in figure 11 of that paper shows the jitter has been quantised into 40 ns intervals. This indicates that the measurement is being made referenced to a clock on the same node. If this measurement of a timing variation on the receiver node was being referenced to the transmitter node - the synchronisation source - then the drift between the two clocks would ensure that intermediate values of jitter would be shown. This author concludes that this is a period variation measurement. Given that assumption the closest match to this amplitude from the simulation results corresponds with the narrow frequency range S100 simulation. This produces a SYT match period variation of 43.6ns.

The word clock figure of 0.55ns rms is much lower than the word clock jitter simulation level of 21ns rms for the same conditions. A similar explanation may apply. The simulation for the narrow frequency range S100 produces a word clock period variation of 0.50ns.

Taking into account this correction then these results show good agreement with the simulation.

8.2 Comparison With Jitter Audibility Criteria

In section 2.2 it was indicated that jitter components below 500Hz have a significantly reduced audibility.

As the jitter spectrum is strongly tonal with frequencies determined by the relative frequencies of node clocks the jitter energy may be concentrated at any part of the spectrum. Hence for worst case considerations the jitter can be assumed to be all at the worst point in the spectrum - for jitter attenuation purposes (and where the slope of the attenuation characteristic is at least 1st order) this is the lowest frequency of interest, 500Hz.

In order to have jitter levels reduced in amplitude to below 1ns the SYT match jitter of 20 to 40ns rms needs to be attenuated by 26 to 32dB. This needs to be for all frequencies above 500Hz.

8.3 Comparison With Jitter “Marketing” Criterion

In section 2.3 a sampling jitter figure of 1.6ns rms was suggested as a target.

With incoming SYT match jitter levels of 38ns rms the jitter needs to be attenuated by 28dB to achieve this target.

If the jitter spectrum was noise-like it would be possible to attenuate the SYT match jitter by 32 dB by reducing the jitter attenuation filter noise bandwidth to:

4kHz*10^(-28/10) = 6.3Hz

However as the jitter components are discrete and arbitrary this method would not be reliable. When node clocks were unfortunately closely matched the jitter would fall at very low frequencies. Then even a significant reduction in the jitter attenuation PLL bandwidth would make very little difference.

This very low frequency jitter would not be audible but it may be a serious marketing disadvantage.

8.4 Engineering Solutions

There are several techniques that can be applied to this problem:

8.4.1 Attenuate the SYT match jitter as recommended in the previous sections. This will require a long time constant and a quartz crystal oscillator-based time-base. The design of this circuit to achieve jitter rejection of 32dB from 500Hz would not be simple - particularly if it was to avoid a long lock-up time and a limited synchronisation range. If it used a conventional quartz crystal based oscillator to generate the clock directly then the system would have a limited pull range that would require a change of oscillator when switching between 44.1kHz related and 48kHz related sources, and then might not be able to lock to sources more than 100ppm away.

It would also require an attenuation characteristic starting at considerably lower frequencies than recent literature has targeted [13,14].

8.4.2 Use an asynchronous sample rate converter to re-sample the signal with the appropriate sampling jitter attenuation function. This would result in non-bit-transparent interface. Current state of the art sample rate converters have a performance no better than a 20 bit wide data channel, [15,16]

This would be adequate for consumer low cost applications or in professional applications where a digital output is not required.

8.4.3 Modify the jitter spectrum so that the attenuation task is simpler. There are methods of controlling the frequencies of the jitter spectral components. This approach could benefit from standardisation through incorporation into IEC-PAS61883-6 so that devices had complementary jitter behaviour.

8.4.4 Modify the IEEE1394 bus physical layer devices to allow node local clocks to be pulled to phase lock to the cycle master. This would eliminate the jitter mechanism. If there were any intermediate devices that did not support “PHY clock phase lock” then those nodes would add repeater jitter. However at S400 rates this jitter may be expected to be 5ns, rather than the 40ns due to CYCLE_COUNT resolution.

8.4.5 For applications with only one destination and with a source, such as a playback device, that can have it’s transmission rate manipulated the destination receiver could control the synchronisation. This mode of operation is used in some consumer audio equipment now in order to avoid jitter from clock recovery circuits. This control could be based on flow control instructions (software handshaking) or on the timing of another isochronous “sync-channel” from the signal receiver back to the signal transmitter.

Both styles of this solution solve the jitter problem for only one receiver for each stream and do not work for sources whose synchronisation is not controllable locally, such as broadcasts.

The flow control method does not provide a synchronisation clock on the bus. The sync-channel method would permit other receiver devices to recover synchronisation - in this case locking to the new synchronisation master receiver - but that would still leave the original jitter problem.

8.4.6 A separate low jitter synchronisation timing reference. This method is described in AES11 [17]. The reference signal could be a word clock or a synchronously clocked digital audio interface signal, such as IEC958 [18] or AES3 [19].

This method would control the synchronisation of all the audio devices so that the IEEE1394 bus was only used for the transfer of the audio data. This approach is similar to that already used for many professional installations. A variant of this mode would have the synchronisation master also generate a synchronisation reference for the bus. This could be used for devices, such as storage devices, that do not require low jitter. The separate synchronisation timing reference could then be used to only feed the devices that need to regenerate a low jitter sample clock.

For mixed sample rate environments where the rates are not related then a separate synchronisation reference signal could be required for each sample rate. It would be possible to avoid this requirement if the reference carried timing information common to both. For example a 48kHz AES3/IEC60958 reference could also carry an embedded 300Hz marker signal that could be used to generate a secondary 44.1kHz sample clock at nodes that require it.

8.5 Alternative Synchronous Interface for Multichannel Linear PCM Audio

The proposed Presto interface [20,21] supports multi-channel digital audio using synchronous data transfer. This is intended for professional applications and could be considered as an alternative approach. It does not have the networking flexibility of IEEE1394 but it does have support for video and control information as well as multichannel audio. It also has a long transmission distance and the synchronous data transmission means that the reclocking jitter generation mechanisms are absent.

9.0 CONCLUSION

Digital audio interfacing using IEEE1394 is possible. Sample clock synchronisation information transmitter over the same interface will have levels of jitter that will not be acceptable for high quality applications without high levels of attenuation in clock recovery systems.

There are several engineering solutions to the problem. These need to be addressed for the “Specification For The Transmission Of Audio And Music Data” over IEEE1394, IEC-PAS 61883-6, to be able to support high quality audio transmission.

10.0 ACKNOWLEDGMENTS

The author wishes to thank Chris Travis of Sonopsis and Steve Harris of Cirrus Logic for assistance and corrections for this paper.

10.0 REFERENCES

[1] IEEE Std 1394:1995 - 'IEEE Standard for a High Performance Serial Bus', IEEE, August 1996

[2] P1394a Draft 2.0 - 'Draft Standard for a High Performance Serial Bus (Supplement)', IEEE, March 15 1998

[3] Don Anderson, 'Firewire System Architecture', 2nd Ed., Addison Wesley December 1998

[4] IEC 61883-1:1998 - 'Consumer audio/video equipment - Digital interface - Part 1: General', IEC, Geneva 1998

[5] IEC-PAS 61883-6:1998 - 'Consumer audio/video equipment - Digital interface - Part 6: Audio and music data transmission protocol', 1394 Trade Association, Austin, 1997 and IEC, Geneva, 1998

[6] Jun-ichi Fujimori and Yoshio Osakabe, 'Digital Audio and Performance Data Transmission Protocol Over IEEE 1394', Pre-print 4346 of the 101st AES Convention, Los Angeles, November 1996

[7] Julian Dunn - `Considerations for Interfacing Digital Audio Equipment to the Standards AES3, AES5, AES11' Published in `Images of Audio', the *Proceedings of the 10th International AES Conference*, London, September 1991. pp 115-126.

[8] Eric Benjamin and Benjamin Gannon, 'Theoretical and Audible Effects of Jitter on Digital Audio Quality', Pre-print 4826 of the 105th AES Convention, San Francisco, September 1998

[9] Hirotaka Kuribayashi, Yasushi Ohtani, and Jun-ichi Fujimori, 'A supplement to Audio and Music Data Transmission Protocol over IEEE 1394: SMPTE time code transmission, Annex A - Delay and jitter evaluation of an audio clock on IEEE1394', Yamaha Corporation, September 1998, presented to 105th AES Convention, San Francisco 1998 (Note: the pre-print for the presentation does not have annex A).

[10] Bob Moses and Greg Bartlett, 'Audio Distribution and Control Using the IEEE 1394 Serial Bus', Pre-print 4548 of the 103rd AES Convention, New York, September 1997

[11] Bob Moses, 'Implementing Digital Audio Devices for the IEEE 1394 High Performance Serial Bus', Preprint 4761 of the 105th AES Convention, San Francisco, September 1998

[12] Jun-ichi Fujimori and Shizuo Kakiuchi, 'Digital Audio Transmission Over IEEE 1394: Protocol Design and Implementation', Preprint 4547 of the 103rd AES Convention, New York, September 1997

[13] Wai-Ki Wong, 'High-Performance Jitter-Reduction Circuit for Digital Audio', Preprint 3888 of the 97th AES Convention, San Francisco, November 1994

[14] M. Wood, 'Analysis of Jitter Rejection of SRCs and DACs Using an NCO', Preprint 4261 of the 100th AES Convention, Copenhagen, May 1996

[15] Robert Adams, Khiem Nguyen and Karl Sweetland, 'An Integrated AES/EBU Receiver/Sample-Rate Converter', Preprint 4536 of the 103rd AES Convention, New York, September 1997

[16] Steven Harris, Gautham Kamath and Eric Gaalaas, 'A Monolithic 24-Bit, 96-kHz Sample Rate Converter with AES3 Receiver and AES3 Transmitter', Preprint 4771 of the 105th AES Convention, San Francisco, September 1998

[17] AES11-1991 - ` Recommended Practice for Digital Audio Engineering - Synchronisation of Digital Audio Equipment in Studio Operations' J. Audio Eng. Soc. (Standards and Information Documents), vol. 39, pp 155-162 (1991 March)) (The latest version including amendments is available from www.aes.org)

[18] IEC-958 - `Digital Audio Interface' First Edition, International Electrotechnical Commission, Geneva, Switzerland (1989 March)

[19] AES3-1992 - ` Recommended Practice for Digital Audio Engineering - Serial Transmission Format for Two-Channel Linearly Represented Digital Audio Data' J. Audio Eng. Soc., vol. 40 No. 3, pp 147-165 (June 1992) (The latest version including amendments is available from www.aes.org)

[20] Tim Thompson and Hal Chamberlin, ‘Presto: A High-Performance Transport for Digital Audio, Video, Control, and Synchronization’, Preprint 4550 of the 103rd AES Convention, New York, September 1997

[21] Presto web page - http://std.yrdi.com/presto

11.0 LIST OF FIGURES

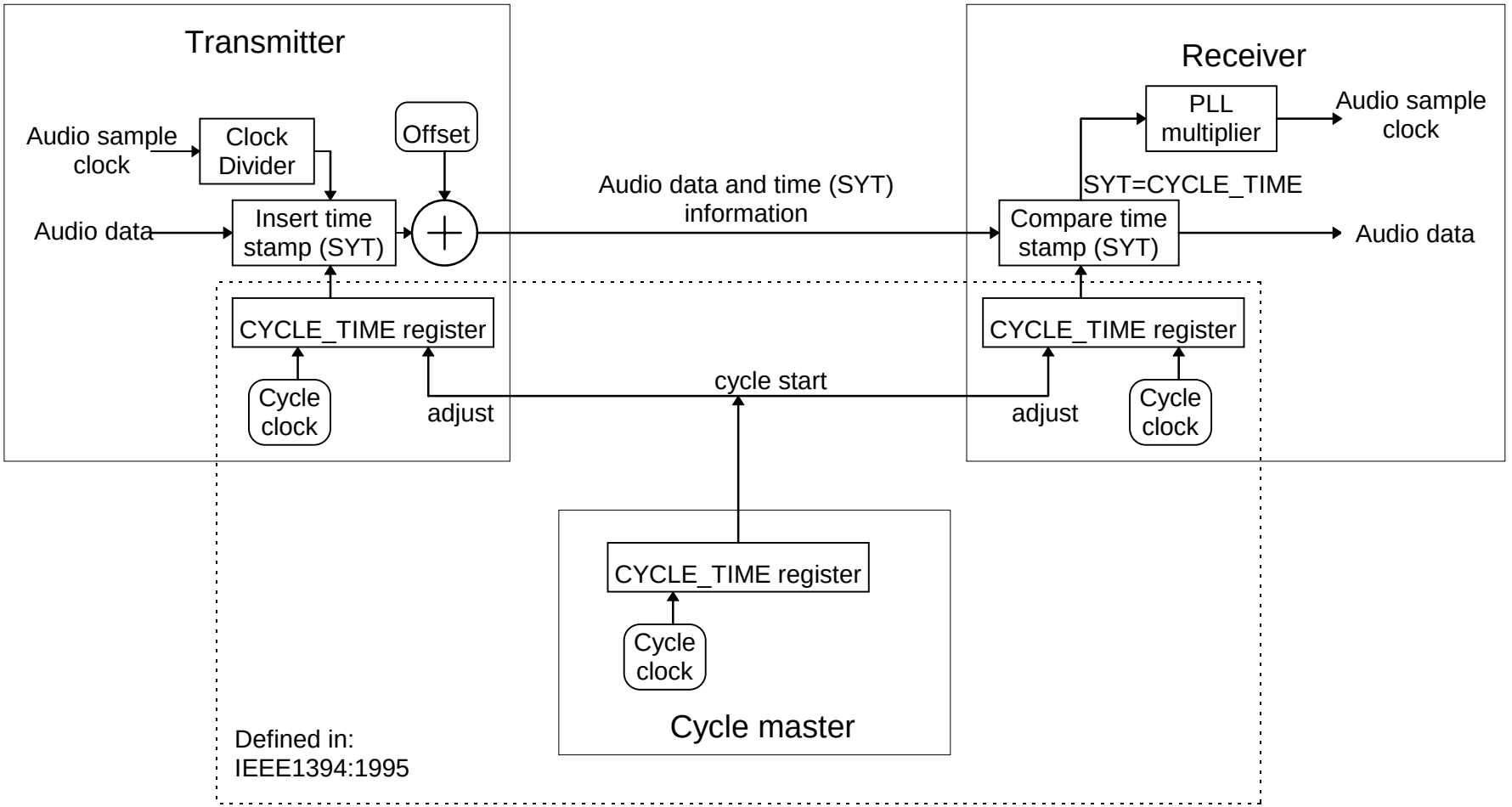


Figure 1 Timing information model for time-stamp method of sample clock synchronisation

(After Kuribayashi, Ohtani and Fujimori 1998)

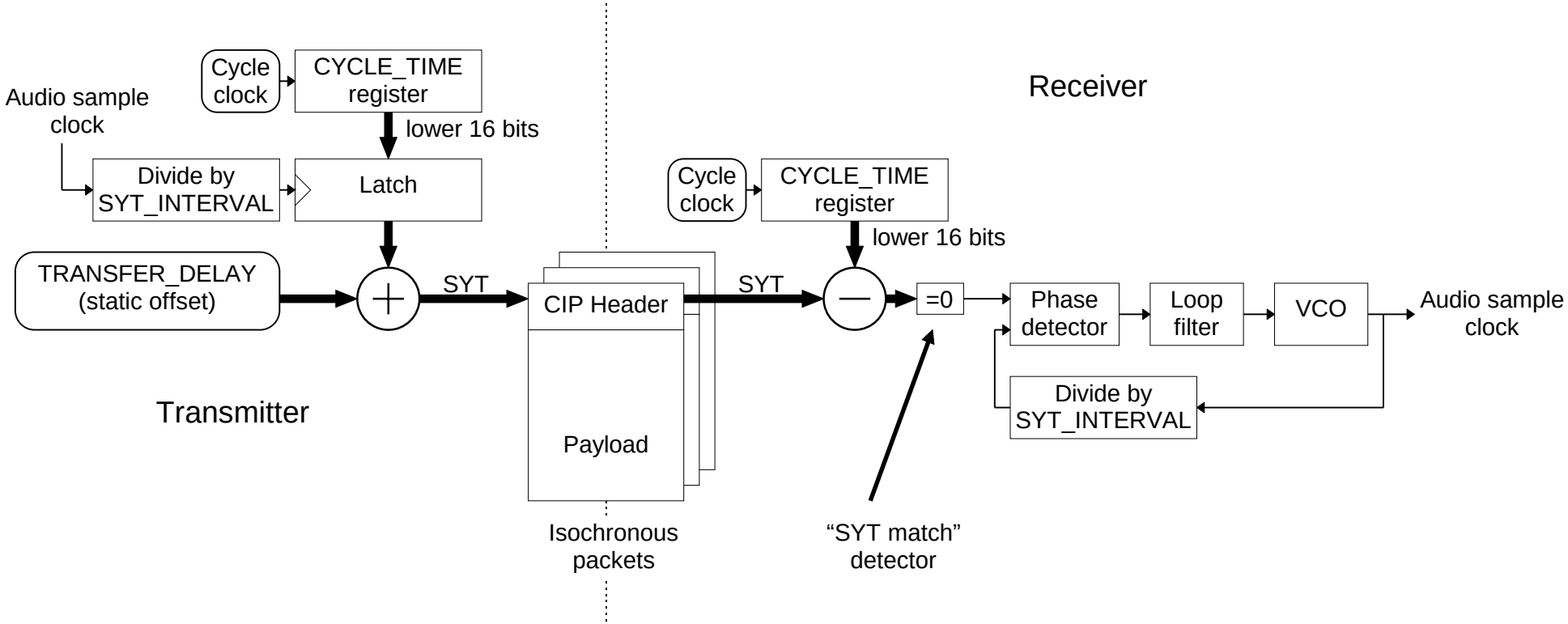


Figure 2 Time stamp model for sample clock synchronisation

(After Moses 1998)

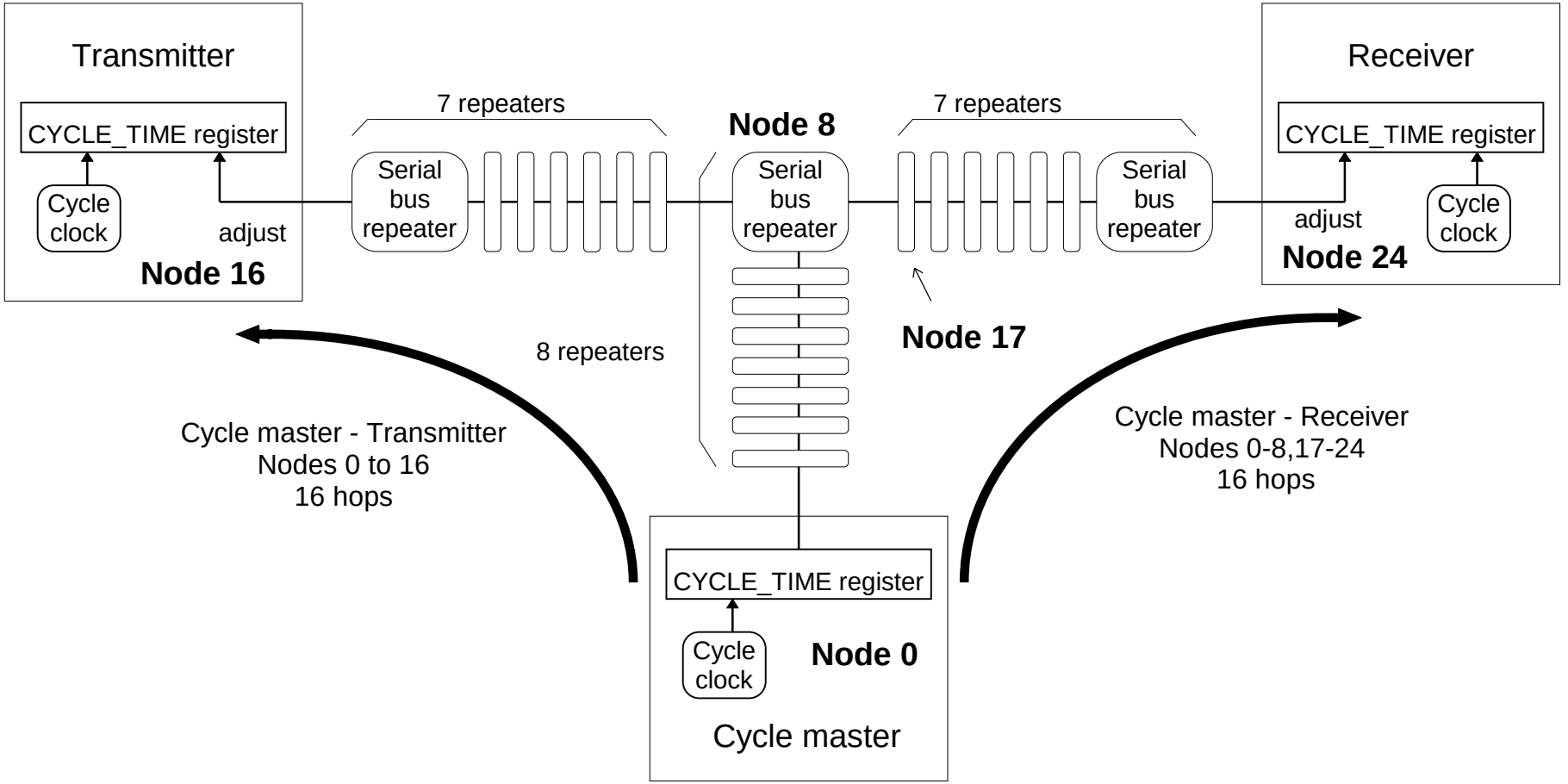


Figure 3 Cycle start timing flow - three-way 16 hop topology

(After Kuribayashi, Ohtani and Fujimori 1998)

Sample clock jitter and real-time audio over the IEEE1394 high performance serial bus - annex describing the simulation

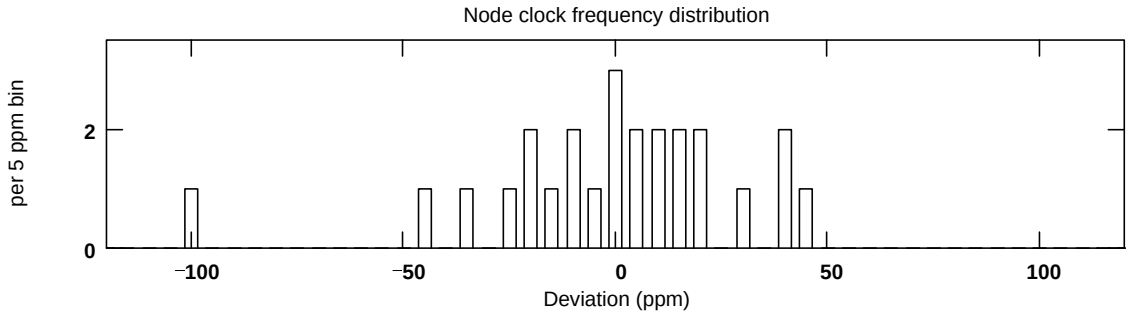
The following sections describe a simulation of the mechanism for jitter generation at the physical layer (PHY) of IEEE1394 when used as described in the A/M protocol (IEC-PAS 61883-6)

A.1 Random selection of PHY node frequencies

The jitter is produced as a result of the interaction of asynchronous clocks at nodes on the bus. A random distribution of frequencies around the nominal is selected in order to simulate this. The IEEE1394 specification requires the PHY clock to be within 100ppm of the nominal frequency.

A gaussian random distribution with a defined standard deviation is simulated. The standard deviation is selected so that 99.7% of node clocks should have a deviation of less than 100ppm.

$$\text{Freq\_error}_{node} := \text{Gaussian\_distribution\_rnd}\left(\frac{100 \cdot \text{ppm}}{3}\right)$$



$\max(\text{Freq\_error}) = 43.8 \cdot \text{ppm}$ $\quad$ $\text{standard\_deviation}(\text{Freq\_error}) = 30.2 \cdot \text{ppm}$

$\min(\text{Freq\_error}) = -100.4 \cdot \text{ppm}$ $\quad$ $\text{mean}(\text{Freq\_error}) = -1.387 \cdot \text{ppm}$

The nominal clock frequencies at each PHY are based on the speed code. IEEE1394:1995 defines three speed codes for the cable medium. The PHY repeater re-clocking depends on the implementation. The simulation uses a frequency corresponding to half the data bit rates for each speed code. This follows Kuribayashi (1998):

$\text{Fclk\_S100} = 49.152 \cdot \text{MHz}$ $\quad$ $\text{Fclk\_S200} = 98.304 \cdot \text{MHz}$ $\quad$ $\text{Fclk\_S400} = 196.608 \cdot \text{MHz}$

A.2 Reclocking at intermediate nodes

The re-clocking of a previously clocked signal by an asynchronous clock adds a variable delay that depends on the relative phases of the two clocks.

$$\text{Clocking\_delay}(t, f1, f2) := \text{mod}\left(\frac{\text{floor}(t \cdot f2)}{f2}, \frac{1}{f1}\right)$$

Delay introduced by a repeater node has a constant element (which is ignored) and a variable element that depends on the relative phases of the clock in the preceding and current nodes. This is simulated by this function of time, t, and the two frequencies, f1 and f2.

The reclocking delay, at time T, between nodes, nodeA and nodeB, is calculated using the frequency offsets (previously determined for each node) and the nominal frequency, F.

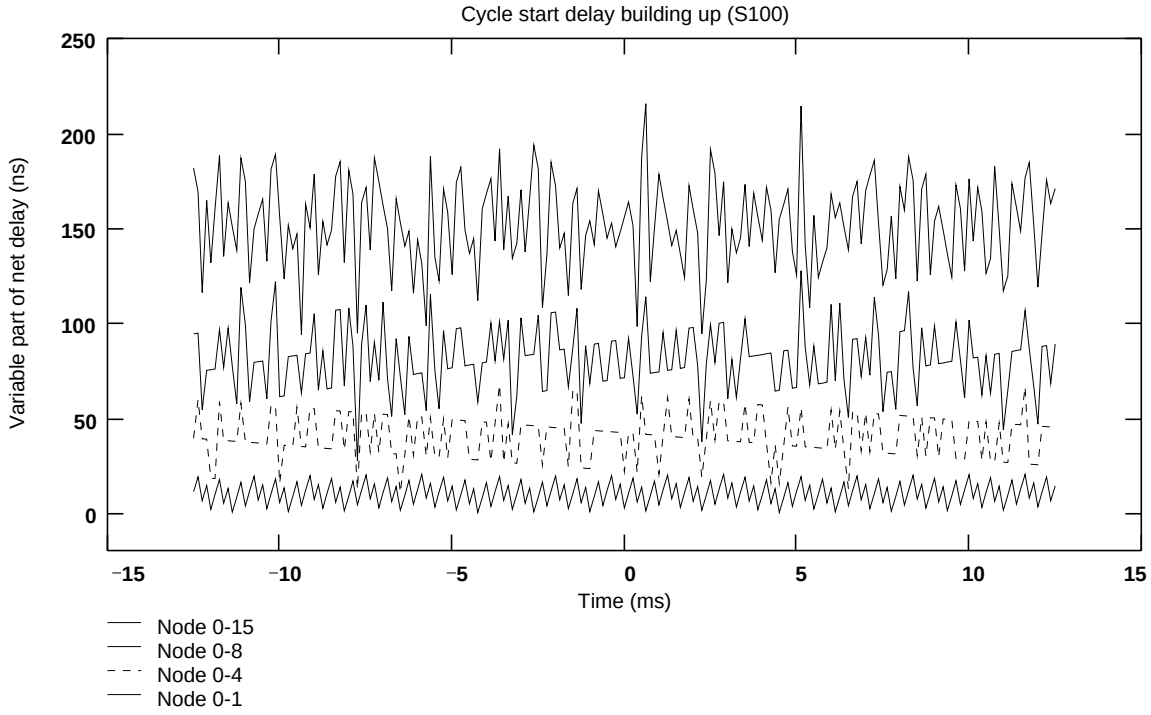
$$\text{Reclock\_delay}(t, \text{nodeA}, \text{nodeB}, F) := \text{Clocking\_delay}\left(t, \text{Freq\_error}_{\text{nodeA}} \cdot F + F, \text{Freq\_error}_{\text{nodeB}} \cdot F + F\right)$$

A.3 Jitter on the arrival of cycle start packets to the transmit node

We are interested in the timing of the transmitter at node 16, where the time-stamp value is determined, and the receiver at node 24 where the presentation time is determined based on that time-stamp. Node 0 is the cycle master node that originates the cycle start and controls the cycle timing. (This ignores the effect of delayed cycle start transmission from the cycle master as that can be corrected). The cycle start gets to node 16 by passing through each node between 0 and 15 and is asynchronously re-clocked at each of the 15 intermediate nodes.

$$\text{Cycle\_start\_delay\_15}(t, \text{Fclk}) := \sum_{\text{node}=1}^{15} \text{Reclock\_delay}(t, \text{node} - 1, \text{node}, \text{Fclk})$$

The build up of this delay is illustrated with samples corresponding to every cycle start



A.4 Transmitter CYCLE_TIME register jitter

The cycle start arrival at node 16 will cause the CYCLE_TIME register to be adjusted. This has a resolution corresponding to the Cycle_clock for that node. Cycle clocks are derived from the PHY clocks - so they have the same frequency error:

$$\text{Cycle\_clock}_{\text{node}} := 24.576 \cdot \text{MHz} \cdot \left(1 + \text{Freq\_error}_{\text{node}}\right)$$

The total delay of the node 16 CYCLE_TIME register, Node_16_Delay, is therefore a quantisation of the time of the the sum of the delays in the cycle start through reclocking at each node plus the delay added by the CYCLE_TIME register resolution.

The cycle start delays used to calculate the CYCLE_TIME register delay should be calculated for the last cycle start before time t.

$$T(t) := \frac{\text{floor}\left[(t) \cdot \text{Cycle\_clock}_0\right]}{\text{Cycle\_clock}_0}$$

As the source of the packets in this simulation the Node_16_Delay is calculated for the time TRANSFER_DELAY in advance of time t, in order to allow for data propagation delays so that they do not allow the data to miss the presentation time.

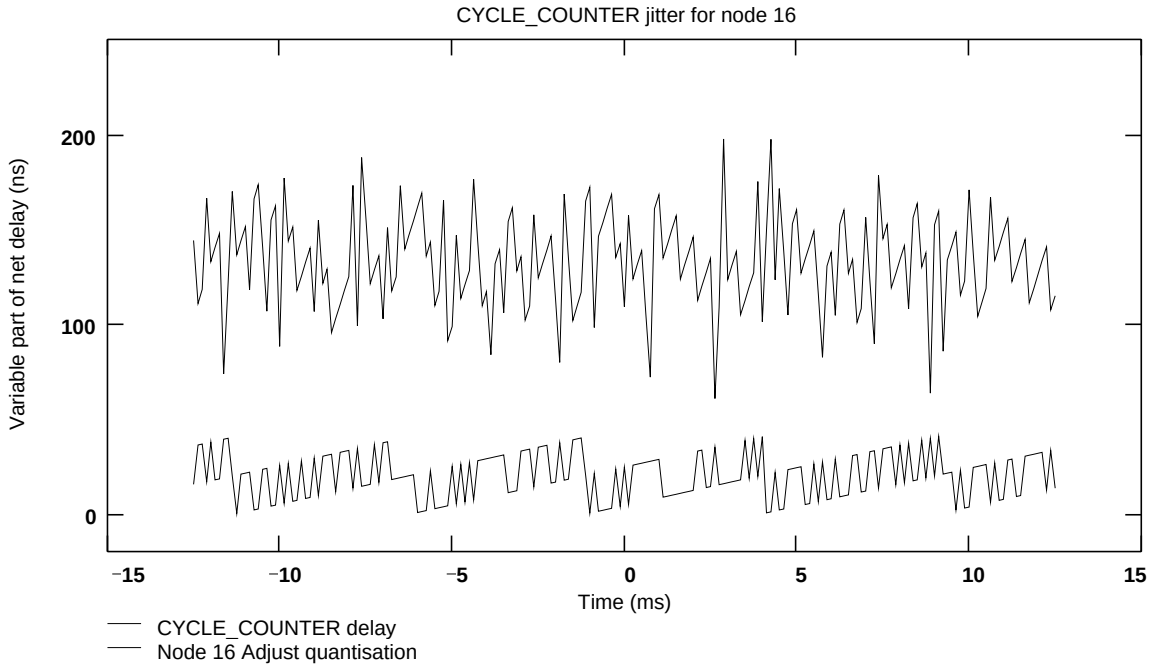
$$TRANSFER\_DELAY := 352 \cdot \mu s$$

(This is the default delay and in some circumstances may be reduced)

Hence $Tadv(t) := T(t - TRANSFER\_DELAY)$

The quantisation of the input cycle start delay by the node 16 cycle clock is calculated here:

$$Node\_16\_Delay(t, Fclk) := \frac{floor\left[(Cycle\_start\_delay\_15(Tadv(t), Fclk) + Tadv(t)) \cdot Cycle\_clock_{16} \right]}{Cycle\_clock_{16}} \ldots + -Tadv(t)$$



A.5 Jitter of the receiver CYCLE_TIME register

The cycle start arrives at node 24 after passing through each node from 1 to 8, then from 17 to 23. The Cycle_start delay is calculated in a similar fashion to the node 16 cycle start delay.

$$Cycle\_start\_delay\_23(T, Fclk) := \sum_{node=1}^{8} Reclock\_delay(T, node - 1, node, Fclk) \ldots + Reclock\_delay(T, 8, 17, Fclk) \ldots + \sum_{node=18}^{23} Reclock\_delay(T, node - 1, node, Fclk)$$

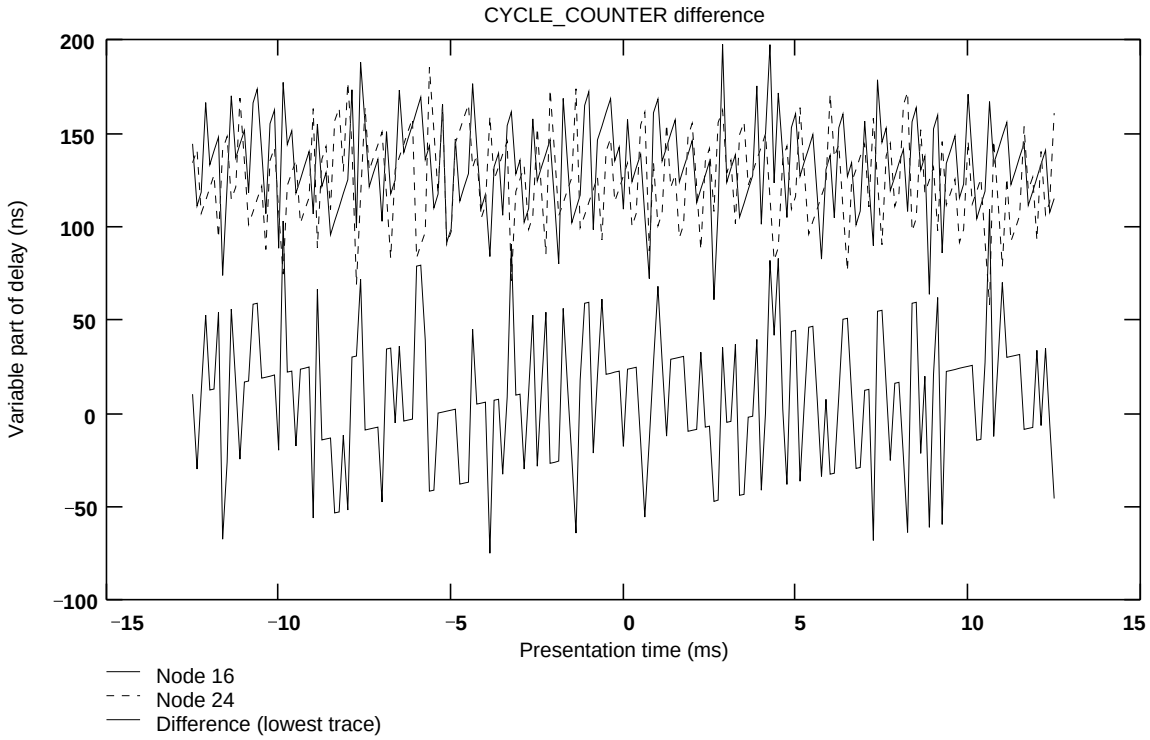
The quantisation effect of the node 24 CYCLE_COUNT register is similar to that for node 16 but as this is the receiver the TRANSFER_DELAY in not included.

$$\text{Node\_24\_Delay}(t, \text{Fclk}) := \frac{\text{floor}\left[(\text{Cycle\_start\_delay\_23}(T(t), \text{Fclk}) + T(t)) \cdot \text{Cycle\_clock}_{24} \right]}{\text{Cycle\_clock}_{24}} - T(t)$$

A.6 Receiver to transmitter CYCLE_COUNTER mismatch

The two cycle start packet paths from node 0 have a common route to node 8. As the difference in the timing at nodes 16 and 24 is of interest it may be thought that the route from node 0 to node 8 would have no net affect as the cycle start delays are the same However for a given audio packet the cycle start used on the transmitter is TRANSFER_DELAY earlier. This allows for a buffer to absorb variations in the transfer delay. The TRANSFER_DELAY is simulated with the calculation of the delay to the transmitter at node 16. It would be wrong to ignore the commonality as there will be some correlation between the delays to node 8 before and after TRANSFER_DELAY. This correlation will increase as the jitter frequency lowers.

The two CYCLE_COUNTER register delays are compared and the difference is illustrated here



A.7 Jitter to the time-stamp values at the transmitter

At the transmitter a time-stamp (SYT) will be applied for every SYT_INTERVAL samples of audio data. This time-stamp has the resolution of the cycle count register with a resulting time-stamp quantisation error. The effect is similar to the delay error due to reclocking but the quantization is more coarse.

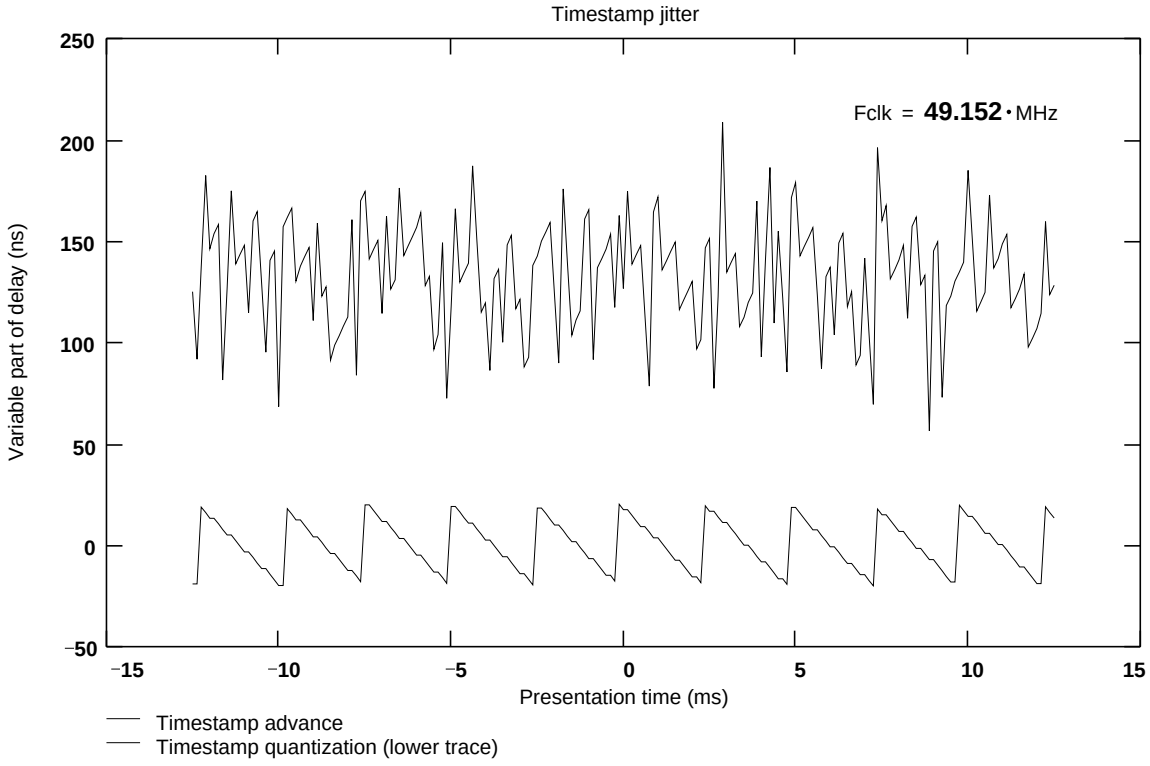
$$Audio\_sample\_rate := 48 \cdot kHz \qquad SYT\_INTERVAL := 8$$

$$Timestamp\_rate := \frac{Audio\_sample\_rate}{SYT\_INTERVAL}$$

$$Timestamp\_quantization(t, Clk) := \operatorname{mod}\left(\frac{\operatorname{floor}(t \cdot Timestamp\_rate)}{Timestamp\_rate}, \frac{1}{Clk}\right) - \frac{0.5}{Clk}$$

This time-stamp quantisation error adds to the error on the time-stamp due to the error in the local cycle timer clock at the transmitter node (node 16), $Cycle\_clock_{16}$. In a conventional implementation this error is due to the variable delay in the cycle start arrival from the cycle master node (node 0); previously calculated as the Node 16 delay.

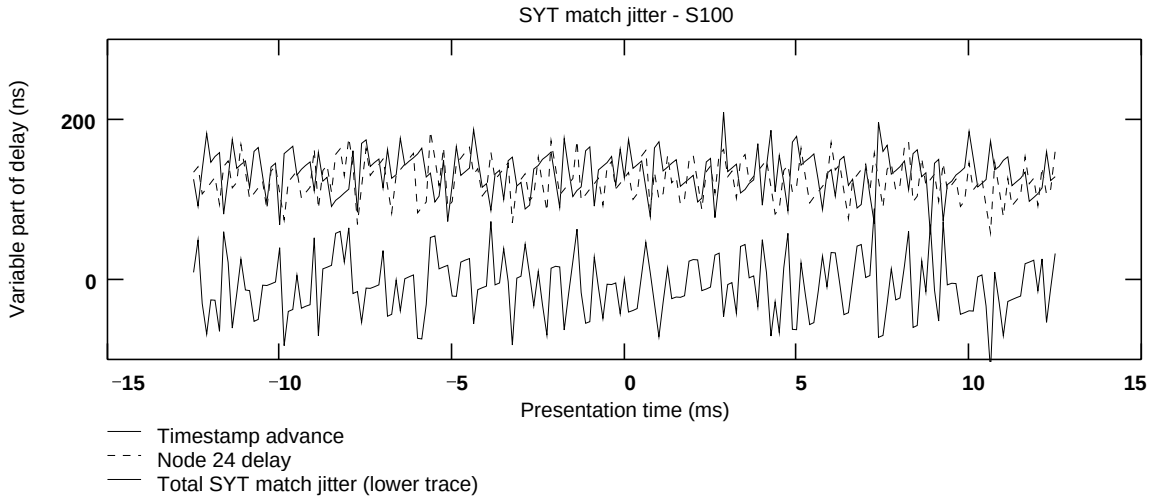
$$Timestamp\_advance(T, Fclk) := Node\_16\_Delay(T, Fclk) + Timestamp\_quantization\left(T, Cycle\_clock_{16}\right)$$



A.8 Time-stamp to SYT match jitter

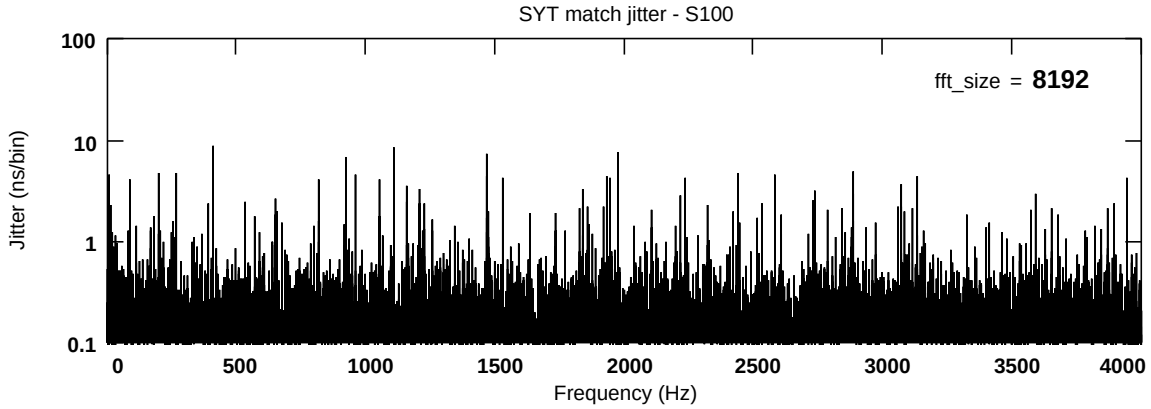
The combination of time-stamp jitter and the presentation time jitter (due to the delay at Node 24) produces the net jitter at the SYT match clock used to recover the audio sample clock in the receiver.

$$SYT\_match\_jitter(T, Fclk) := Node\_24\_Delay(T, Fclk) - Timestamp\_advance(T, Fclk)$$



The rms level of this jitter is:- $standard\_deviation(SYT\_jitter\_S100) = \mathbf{37.99} \cdot ns$

Peak jitter over this data length:- $\max\left(\overrightarrow{|SYT\_jitter\_S100|}\right) = \mathbf{159.6} \cdot ns$



The spectrum here shows that there are several components of about 9 ns rms amplitude. One of these is at the beat frequency between the time-stamp rate and the cycle clock at node 16:

$$F\_beat := Cycle\_clock_{16} - Timestamp\_rate \cdot round\left(\frac{Cycle\_clock_{16}}{Timestamp\_rate}\right)$$

$$|F\_beat| = \mathbf{408.07} \cdot Hz$$

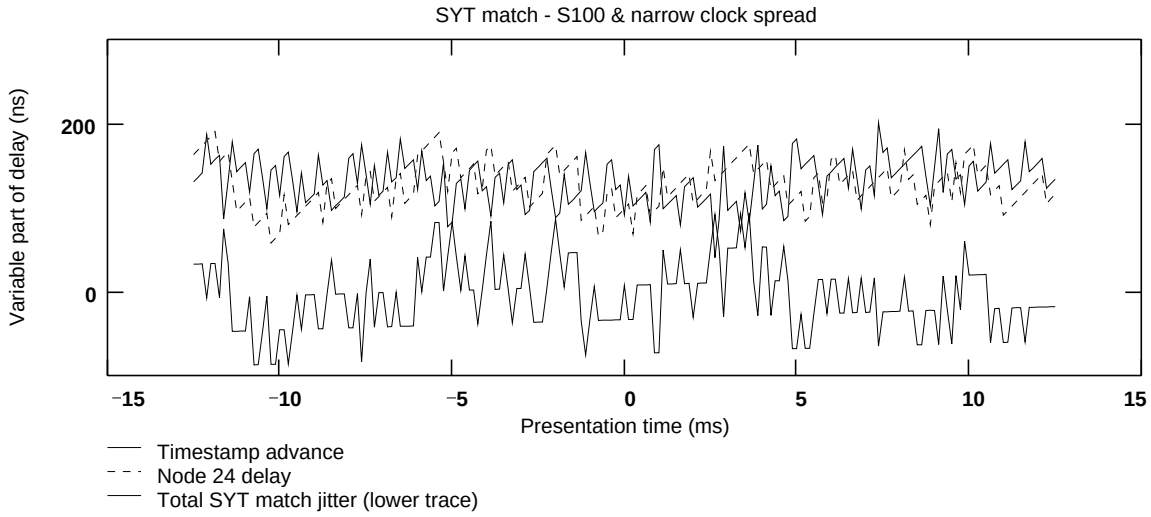
NOTE: The Fourier transforms used in this paper all use a rectangular window. This is convenient in maximising the resolution and also for making the relation between the amplitude of individual spuriae more obvious. As the signal to noise ratio of the jitter signal is low leakage is not a problem. For the high dynamic range audio spectra the audio tone stimulus has a period that is an integer submultiple of the fft size to avoid leakage.

A.9 The effect of narrow PHY clock frequency distribution

The jitter produced by the reclocking and time-stamp quantization is a consequence of the intermodulation of the clocks in the system. This means that the characteristics the jitter are determined by the frequency differences in the clocks. Closely matched clocks will produce predominantly low frequency jitter, and widely spaced clocks will produce higher frequency jitter. This is simulated by reducing the variation in frequency by a scale factor of 10:

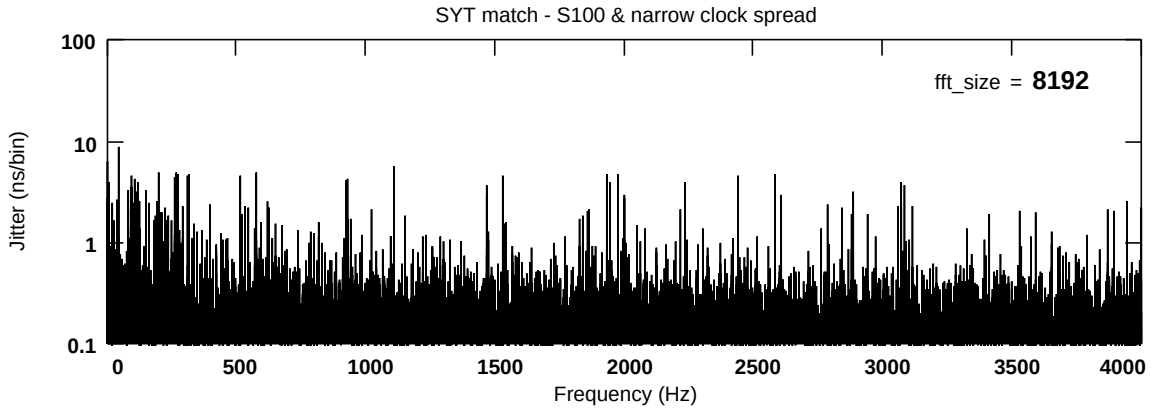
standard_deviation(Freq_error_lf) = **3.0** · ppm

The following plot can be compared with the previous SYT match jitter plot. Notice the closer match between time-stamp advance and node 24 delay. This is connected with the low pass filtering effect of the TRANSFER_DELAY on the jitter induced on the common path between nodes 0 and 8. This results in a slightly lower total SYT match jitter.



The rms level of this jitter is slightly lower: standard_deviation(SYT_jitter_lf) = **38.64** · ns

Peak jitter over this data length: $\max\left(\overrightarrow{|\text{SYT\_jitter\_lf}|}\right)$ = **145.4** · ns



Notice the main jitter components have moved to lower frequencies. The beat frequencies have scaled down by the same factor as the frequency distribution has narrowed.

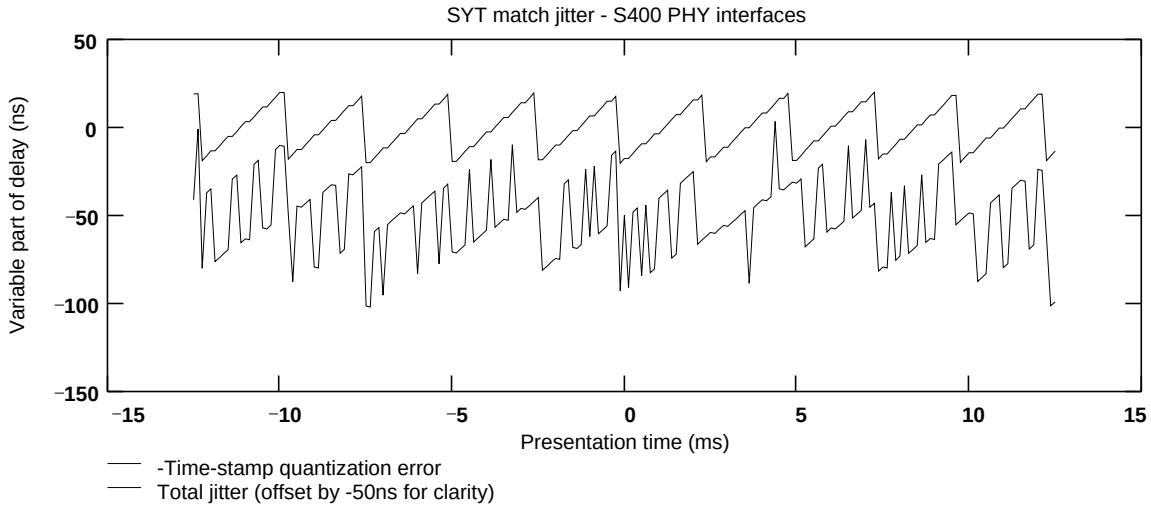
A.10 Simulation with a network using only fast (S400) PHY nodes

By the time the 1394 interface becomes widely used for audio interfacing the normal physical layer interface (PHY) may be operating at the S400 rate. The typical PHY may well then re-clock the data at four times the rate of the S100 PHY simulated above

The simulation was repeated to investigate the implications of this using the same PHY clock frequency deviations as the initial simulation.

The resulting jitter is now dominated by the time-stamp jitter. Note that while the finer resolution of the S400 transmission timing means that reclocking jitter is a factor of four less than for S100. However the time-stamp and the cycle counter adjustment resolution does not change so these components become dominant.

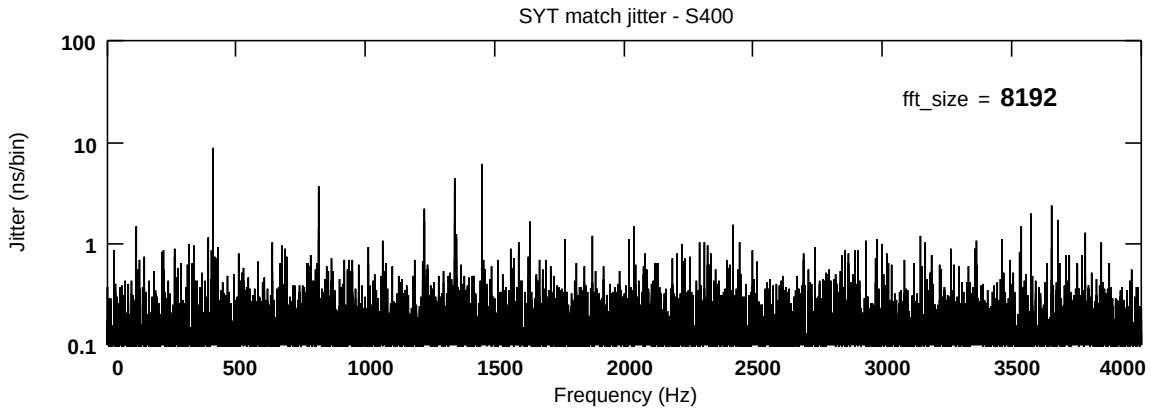
Fclk_S400 = **196.608** · MHz



The rms jitter from this S400 simulation is about 60% of that for the S100 simulation.

standard_deviation(SYT_jitter_S400) = **22.0** · ns

Peak jitter over this data length:- $\max\left(\overrightarrow{|\text{SYT\_jitter\_S400}|}\right)$ = **74.8** · ns



As the time-stamp quantisation jitter is now a predominant component to the jitter the beat frequency between sample clock and the transmitter node (node 16) cycle clock is the dominant spectral component to the resulting jitter. In this case at about 408 Hz and 9 ns rms. The amplitude of this component is unchanged from the S100 simulation.

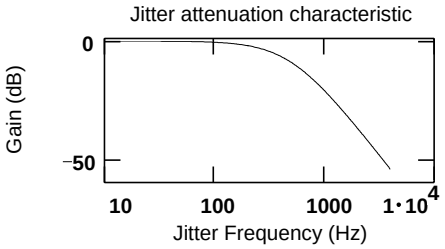
A.11 The effect of 1394 jitter on an audio signal

The SYT match clock is used to define the presentation time of the audio signal. Therefore the audio sample clock needs to be synchronised with it.. The audio sample clock will normally be at a much higher rate so a method is required to interpolate between SYT match instants. This can be achieved with an analogue phase locked loop (PLL) or using a numerical algorithm in conjunction with a variable delay digital interpolating filter, for example a sample rate converter, with the output at a constant sample rate that may be determined by a free-running clock

In both case there will be an associated sampling jitter attenuation characteristic. It is possible to have very long time constants using numerical approaches or with a PLL a low phase noise oscillator. However these have disadvantages When numerical techniques internal to an synchronous rate audio sample rate converter then it will result in a change in the data values that is not desired in many circumstances. In order to achieve good intrinsic jitter performance the low phase noise oscillator will have a restricted operating frequency range. For simple illustration this simulation models a jitter attenuation as a 3rd order roll-off with corner frequency, ω0

$$H(\omega,\omega0) := \frac{\omega0^2}{(j\cdot\omega)^2 + \frac{\omega0}{0.5}\cdot j\cdot\omega + \omega0^2}\cdot\frac{\omega0}{j\cdot\omega+\omega0}$$

Jitter attenuation function $\omega0 := 2\cdot\pi\cdot 500\cdot Hz$

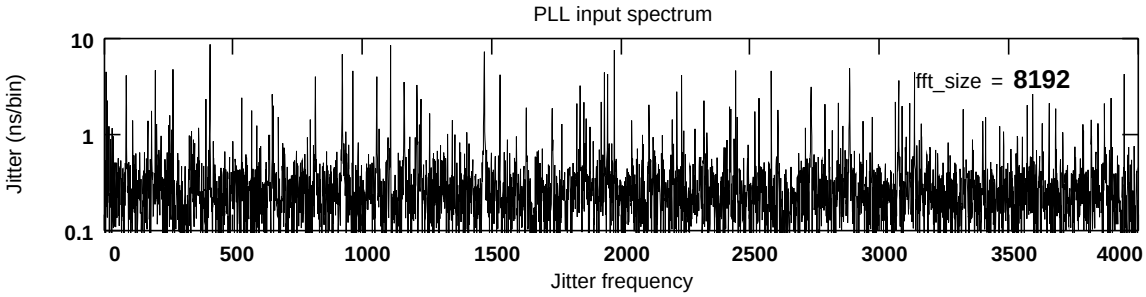


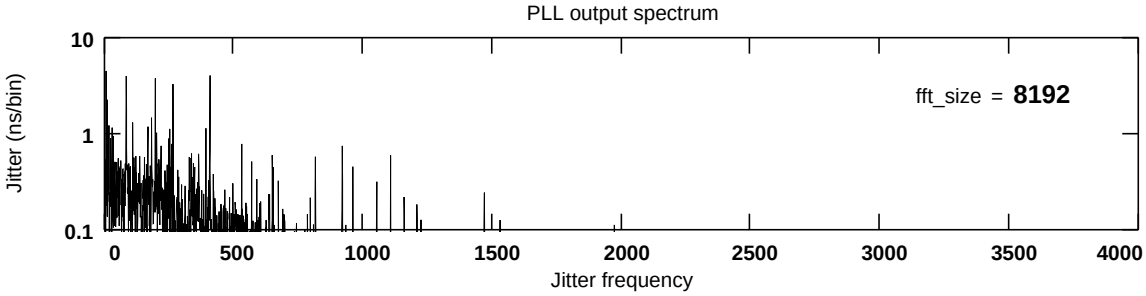
SYT match rms jitter level (S100)

standard_deviation(SYT_jitter_S100) = **37.99** ·ns

Filtered rms jitter level (S100)

standard_deviation(Filtered_jitter_S100) = **11.85** ·ns





These graphs show the jitter spectrum before and after filtering with the example PLL. The PLL filtered signal is now used to illustrate the effects of jitter on an audio signal

A.12 Sampling jitter simulation from S100 jitter

The effect of sampling jitter is to modify the occurrence of the sampling instants and hence the timing of the signal. This can be simulated for a pure tone:

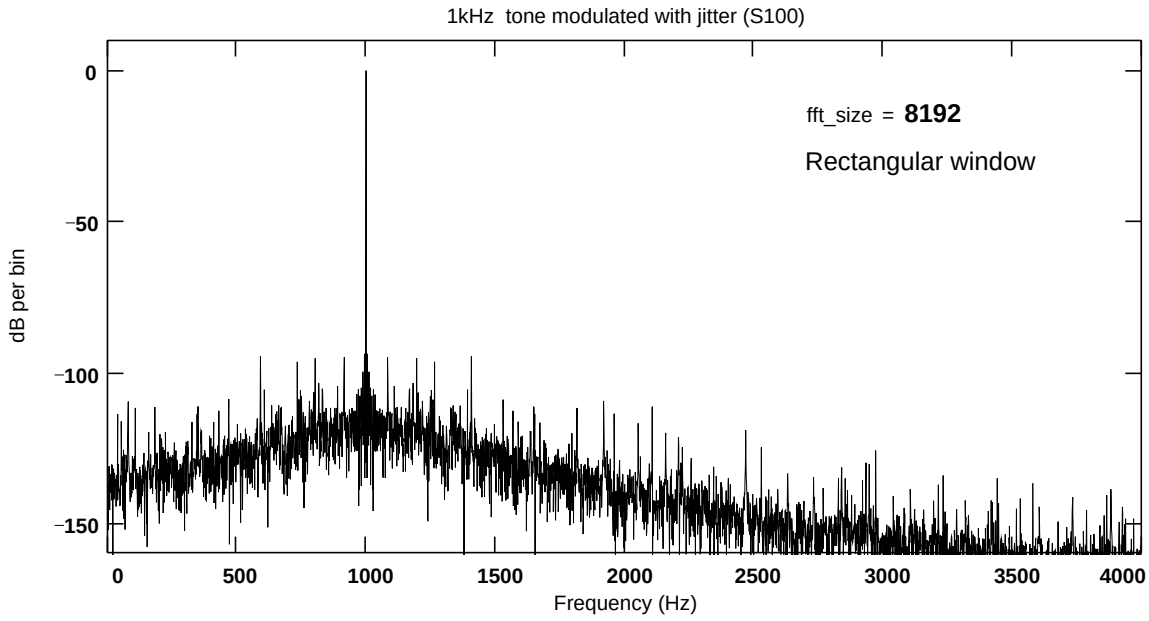
$$Audio\_signal_p := \sin\left(2 \cdot \pi \cdot Audio\_freq \cdot t(p)\right) \qquad Audio\_freq = 1 \cdot kHz$$

We can simulate the tone time-deviated by the results of the filtered PLL output jitter simulation:

$$Audio\_filtered\_jitter_p := \sin\left[2 \cdot \pi \cdot \left(t(p) + Filtered\_jitter\_S100_p\right) \cdot Audio\_freq\right]$$

Perform a Fourier transform to get the spectrum

$$Tone\_PLL\_S100 := fft(Audio\_filtered\_jitter) \cdot \sqrt{\frac{2}{fft\_size}}$$



Notice the modulation of the pure tone produces modulation sidebands corresponding with the jitter frequency components. As an effect of the filtering these are concentrated close to the original tone

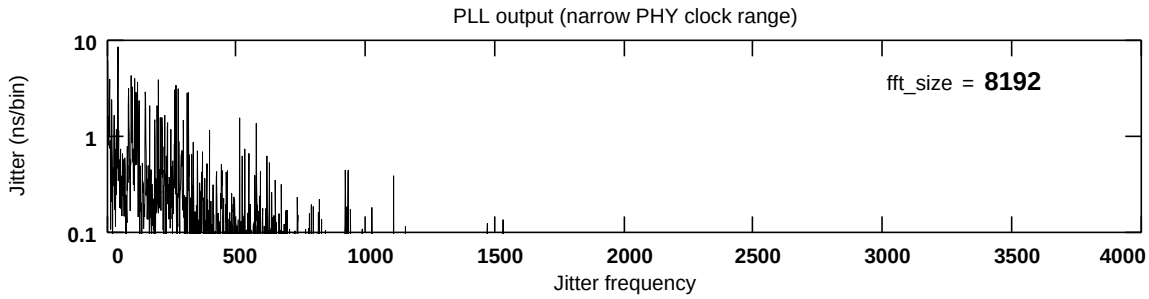
The total error can be calculated as a ratio with the signal, producing a signal to noise ratio.

$$Signal\_to\_Noise(Tone\_PLL\_S100) = -82.6 \cdot dB$$

This is only looking at frequencies from dc to 4 kHz, but as the error slopes down at high frequencies it is unlikely to be much worse for a 20 kHz bandwidth.

A.13 The sampling jitter effect of narrow frequency range PHY clocks

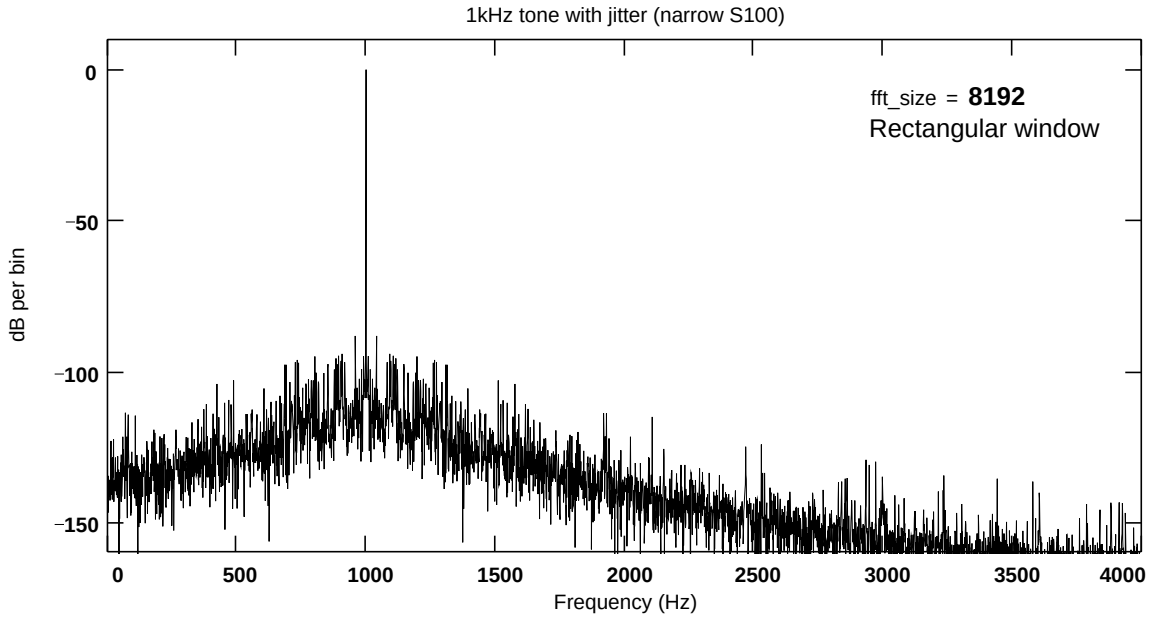
This is the jitter spectrum for the narrow narrow frequency spread PHY cycle clocks simulation after filtering with the PLL.



As was noted earlier the jitter is concentrated at lower frequencies and so more of the jitter is within the pass-band of the PLL filter. As a result the PLL output jitter is higher than for the previous simulation.

S100 rate with 10 ppm matched PHY clocks	S100 rate with 100ppm matched PHY clocks
standard_deviation(SYT_jitter_lf) = **38.6** · ns	standard_deviation(SYT_jitter_S100) = **38.0** · ns
standard_deviation(Filtered_jitter_lf) = **21.3** · ns	standard_deviation(Filtered_jitter_S100) = **11.9** · ns

The effect of this jitter modulation on an audio signal is illustrated here,

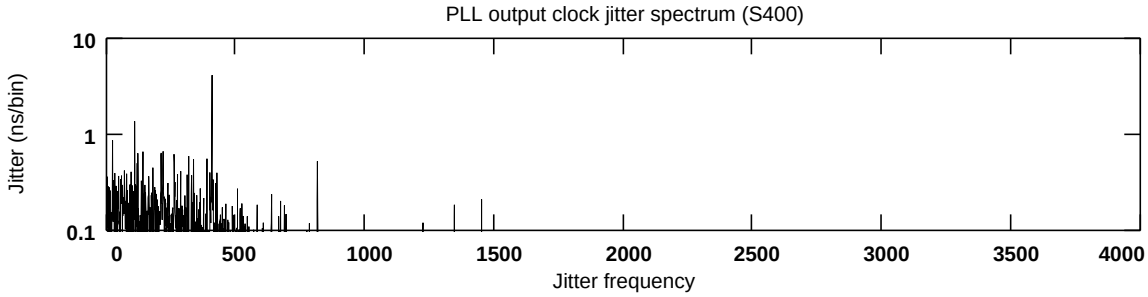


The higher level of low frequency jitter of this simulation results in a worse signal to noise measurement

Signal_to_Noise(Tone_PLL_lf) = −**77.5** · dB

A.14 The sampling jitter effect of high speed (S400) PHY clocks

The audio simulation was repeated with the broader frequency range PHY clocks of the first simulation but with faster S400 PHY nodes. The PLL output jitter spectrum is shown here:



It was noted earlier that the rms jitter (standard deviation of the jitter) from this S400 simulation is reduced to just below half that for the S100 simulation.

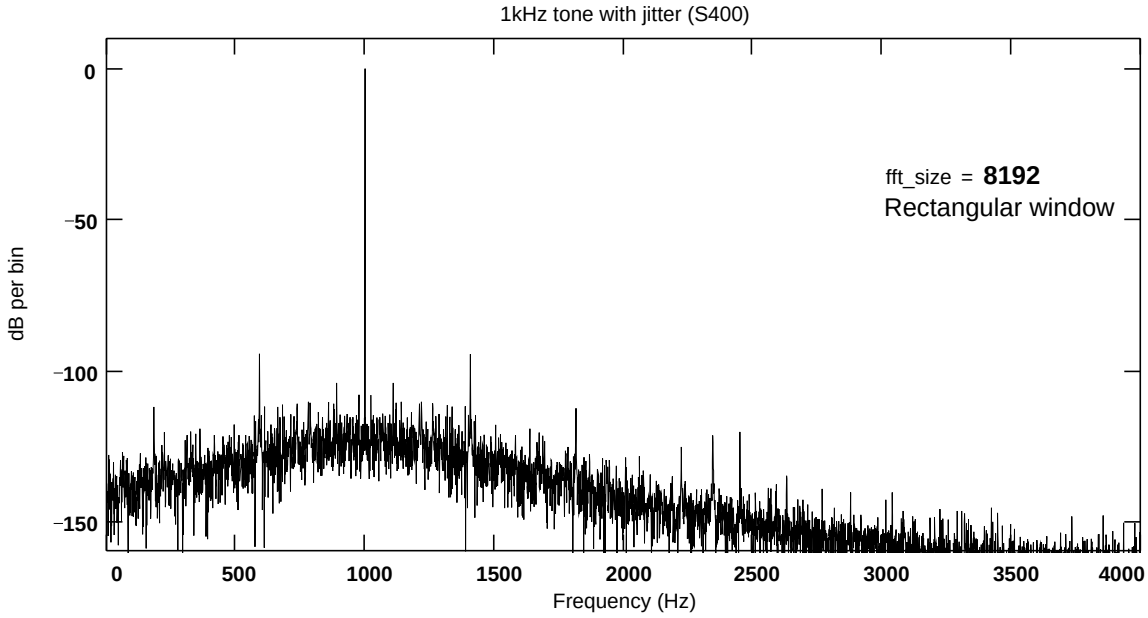
standard_deviation(SYT_jitter_S400) = **22.0** · ns

standard_deviation(SYT_jitter_S100) = **38.0** · ns

The word clock jitter at the output of the PLL filter is also reduced

standard_deviation(Filtered_jitter_S400) = **6.2** · ns

standard_deviation(Filtered_jitter_S100) = **11.9** · ns



Signal_to_Noise(Tone_PLL_S400) = −**88.1** · dB

The effect of the strongly tonal jitter modulation on the 1 kHz audio signal is a pair of 400 Hz sidebands at -94.6 dB FS corresponds with the 4.2 ns rms jitter component at 408 Hz that been attenuated from 9 ns by the PLL filter.

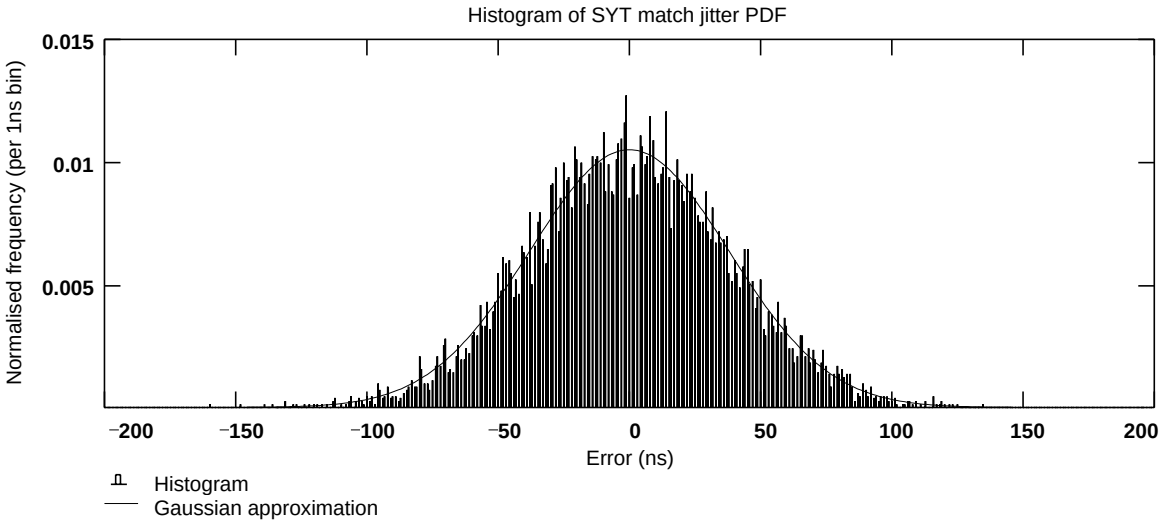
A.15 Statistical analysis of the simulation results

The probability density function of the SYT match jitter is derived and plotted as a histogram

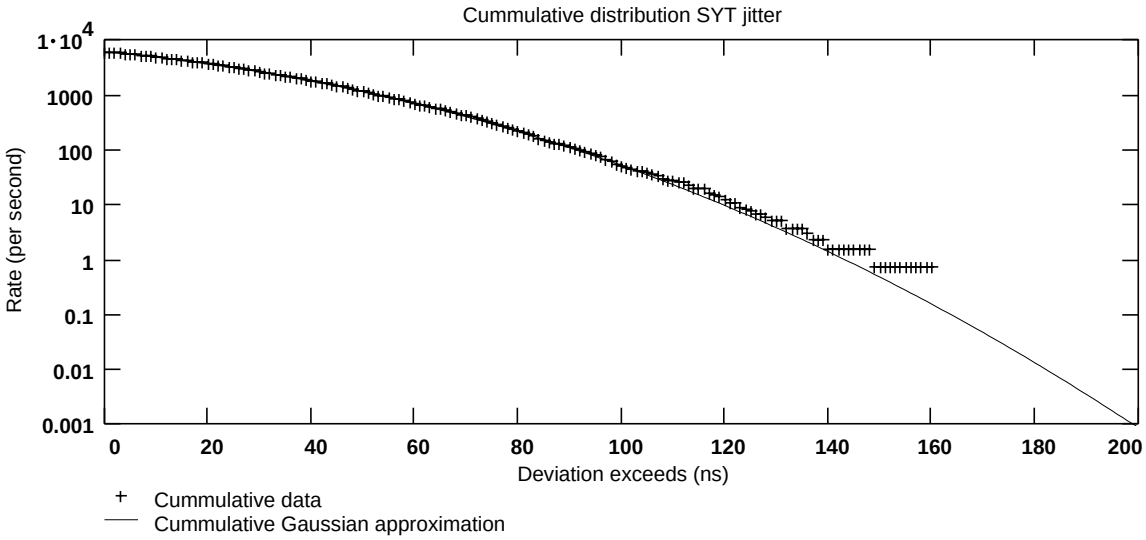
data_length = **8192**

standard_deviation(SYT_jitter_S100) = **38.0** · ns

$\max\left(\overrightarrow{|\text{SYT\_jitter\_S100}|}\right)$ = **159.6** · ns Peak sample over this data length



It may be important to be aware how often the jitter peaks occur. The following graph shows the cumulative probability of a jittered sample exceeding the deviation specified on the x axis. This cumulative Gaussian distribution with the same deviation is also plotted. The approximation appears to break down at 1 second - which is the length of the data set. However it is clear that peaks of 140 ns will occur at about once a second



A.16 Alternative "jitter" measurement - clock period variation

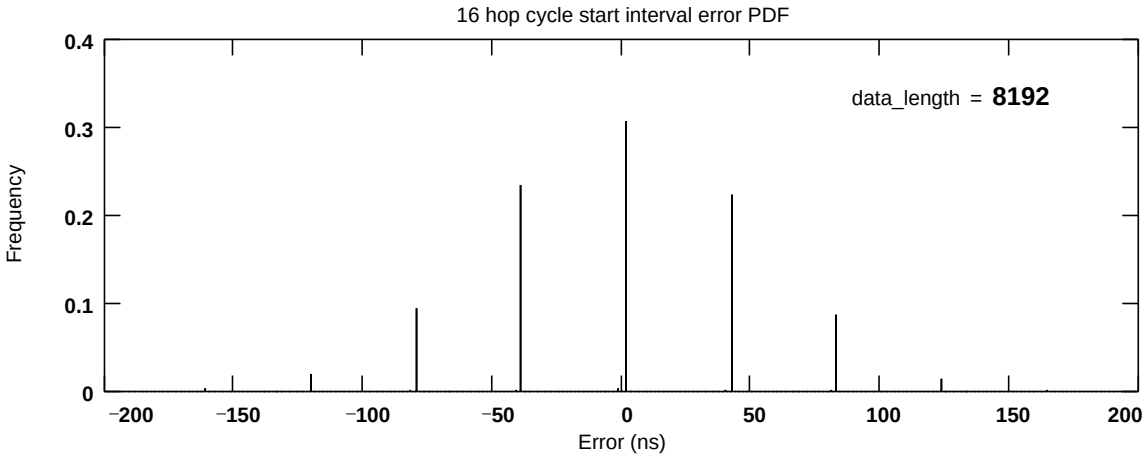
An alternative analysis of timing variation by looking at the error in the interval betweer successive SYT match instants. (This is what is called SYT match clock jitter in Kuribayashi et al. (1998) [9].)

$T := \frac{1}{Timestamp\_rate}$ $T = 166.67 \cdot \mu s$

For simplicity in the analysis the simulation has the time-stamp rate matching the cycle start rate

$Period\_err_N := SYT\_match\_jitter(t(N), Fclk\_S100) - SYT\_match\_jitter(t(N) - T, Fclk\_S100)$

Where the jitter value at the start and end of the period is uncorrelated this period variation measurement has a standard deviation 3 dB higher than a jitter measurement of the same signal. For jitter that is weighted towards lower frequencies the correlation increases and the amplitude will fall.



These calculations were repeated for all three simulations for comparison

standard_deviation(Period_err) = **51.3** · ns	S100 with first simulation (100ppm/3) frequency
standard_deviation(Period_err_lf) = **43.6** · ns	S100 with narrow (100ppm/30) frequency spread
standard_deviation(Period_err_S400) = **27.5** · ns	S400 simulation

This compares with the rms jitter measurements (calculated earlier) for the same signals

standard_deviation(SYT_jitter_S100) = **37.99** · ns	S100 with first simulation (100ppm/3) frequency
standard_deviation(SYT_jitter_lf) = **38.64** · ns	S100 with narrow (100ppm/30) frequency spread
standard_deviation(SYT_jitter_S400) = **22.02** · ns	S400 simulation

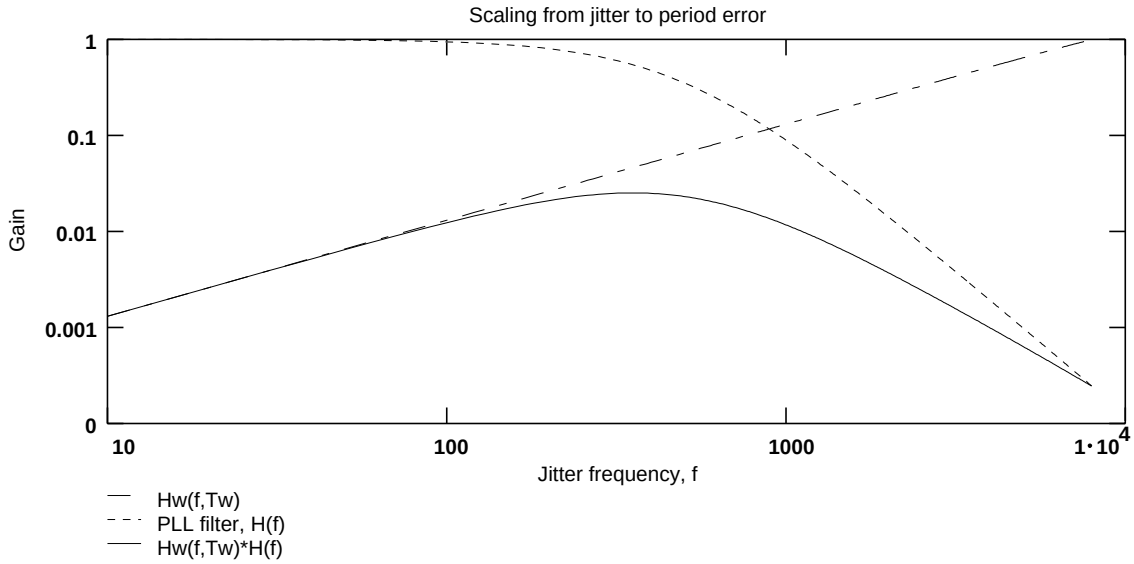
These results can be compared with the figure of 38.9ns from table 5 of Kuribayashi [9] which appears to be a SYT match interval variation measurement rather than a pure jitter measurement. With the strong jitter frequency dependence of this method of measurement it is difficult to draw conclusions about the match in the results but the narrow frequency spread S100 result is less than 10% high.

There is a transfer function between jitter and word clock period variation. A period variation measurement takes the jitter at the current transition and subtracts the jitter at the transition one period in the past. This is applying a comb filter to the jitter and has the following frequency domain transfer function:

$$Hw(f, Tw) := 2 \cdot \sin(\pi \cdot f \cdot Tw)$$

This is illustrated on the figure for the word clock period, $Tw = 20.833 \cdot \mu s$

This high pass filter would combine with any low pass PLL jitter transfer function to attenuate throughout the band as illustrated here for the 3rd order characteristic described earlier, with corner frequency at 500Hz. Note the under-reading of at least 20 dB for the jitter frequencies below 1 kHz.



Applying this function and the PLL filter gives a filtered rms word clock deviation for the simulations.

$$Period\_deviation(spectrum) := \sqrt{\sum_f \frac{\left(\left|spectrum_f \cdot H(\omega(f), \omega 0) \cdot Hw(Jitter\_Freq(f), Tw)\right|\right)^2}{length(spectrum)}}$$

$Period\_deviation(SYT\_spectrum\_S100) = 0.43 \cdot ns$		S100 wider clock range
$Period\_deviation(SYT\_spectrum\_lf) = 0.50 \cdot ns$		S100 narrow clock range
$Period\_deviation(SYT\_spectrum\_S400) = 0.30 \cdot ns$		S400

Compare with the filtered rms word clock jitter (i.e. referenced to absolute time) from S100 wide clock range simulation:

$$\sqrt{\left[\sum_f \frac{\left(\left|SYT\_spectrum\_S100_f \cdot H(\omega(f), \omega 0)\right|\right)^2}{length(SYT\_spectrum\_S100)}\right]} = 11.85 \cdot ns$$

This example highlights the difference between the two measures. The word clock deviation measurement is apparently much less than 1 ns. However, as that measure is insensitive to low frequency jitter it can be very misleading. Particularly where - as in this case - the measurement is on the output of a jitter attenuating PLL.