TA Document 1999014

Enhancement to Audio and Music Data Transmission Protocol 1.0

July 10, 2000

Sponsored by:
1394 Trade Association

Accepted for Release by:
1394 Trade Association Board of Directors.

Abstract:
This specification defines enhancements to the 1394 TA specification "Audio and Music Data Transmission Protocol Ver.1.0" (or IEC PAS 61883-6).

Keywords:
Audio and Music, IEC PAS 61883-6, DVD-Audio, SACD, MIDI.

1394 Trade Association Specifications are developed within Working Groups of the 1394 Trade Association, a non-profit industry association devoted to the promotion of and growth of the market for IEEE 1394-compliant products. Participants in working groups serve voluntarily and without compensation from the Trade Association. Most participants represent member organizations of the 1394 Trade Association. The specifications developed within the working groups represent a consensus of the expertise represented by the participants.

Use of a 1394 Trade Association Specification is wholly voluntary. The existence of a 1394 Trade Association Specification is not meant to imply that there are not other ways to produce, test, measure, purchase, market or provide other goods and services related to the scope of the 1394 Trade Association Specification. Furthermore, the viewpoint expressed at the time a specification is accepted and issued is subject to change brought about through developments in the state of the art and comments received from users of the specification. Users are cautioned to check to determine that they have the latest revision of any 1394 Trade Association Specification.

Comments for revision of 1394 Trade Association Specifications are welcome from any interested party, regardless of membership affiliation with the 1394 Trade Association. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally, questions may arise about the meaning of specifications in relationship to specific applications. When the need for interpretations is brought to the attention of the 1394 Trade Association, the Association will initiate action to prepare appropriate responses.

Comments on specifications and requests for interpretations should be addressed to:

Editor, 1394 Trade Association
Regency Plaza Suite 350
2350 Mission College Blvd.
Santa Clara, Calif. 95054, USA

1394 Trade Association Specifications are adopted by the 1394 Trade Association without regard to patents which may exist on articles, materials or processes or to other proprietary intellectual property which may exist within a specification. Adoption of a specification by the 1394 Trade Association does not assume any liability to any patent owner or any obligation whatsoever to those parties who rely on the specification documents. Readers of this document are advised to make an independent determination regarding the existence of intellectual property rights, which may be infringed by conformance to this specification.

Table of contents

List of figures

List of tables

1. Overview

1.1 Purpose

The purpose of this document is to define enhancements to the "Audio and Music Data Transmission Protocol Ver. 1.0" (hereafter A&M Protocol Ver. 1.0), also known as IEC PAS 61883-6, for new audio centric applications. In order to enhance A&M protocol Ver. 1.0, a new interpretation or definition will be given to some of the definitions in this enhancement specification without changing the original meanings in the context of A&M Protocol Ver. 1.0.

This new specification will introduce a more generic model for the packetization process. Such a generalization will produce new parameters in the generic model. The parameter values are arranged so that the previous model can be interpreted as a special case or instance of the new generic model.

The "Adaptation Layers" clause includes references to the specification and a brief informative description of the applications that uses this specification.

Also this document provides examples and informative clauses for reducing the ambiguity of the A&M Protocol Ver. 1.0 for maintaining minimum connectivity.

1.2 Scope

This document defines enhancements to 1394 TA Specification "Audio and Music Data Transmission Protocol Ver. 1.0. The enhancements are focused on the AM824 Data type.

2. References

The following standards contain provisions, which through reference in this document, constitute provisions of this standard. All the standards listed are normative references. Informative references are given in Annex A. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

[R1] IEEE Std 1394-1995, Standard for a High Performance Serial Bus.

[R2] IEC 61883-1, Consumer audio/video equipment – Digital interface – Part 1: General.

[R3] IEC 61883-6 PAS, Audio and music data transmission protocol

[R4] IEC 60958-1, Digital audio interface - Part 1: general

[R5] IEC 60958-3, Digital audio interface - Part 3: Consumer applications

[R6] IEC 60958-4, Digital audio interface - Part 4: Professional applications

[R7] IEC 61937, Digital audio - Interface for non-linear PCM encoded audio bitstreams applying IEC 60958

[R8] IEEE Std 754-1985, Binary Floating-Point Arithmetic

[R9] RP-027, Specification for MIDI Media Adaptation Layer for IEEE1394

[R10] Complete MIDI 1.0 Detailed Specification

[R11] TA 1999024, SMPTE Time Code and Sample Count Transmission Protocol Ver.1.0

[R12] TA 1998003, AV/C Digital Interface Command Set General Specification, Version 3.0.

[R13] TA 1999015, AV/C Command Set for Rate Control of Isochronous Data Flow 1.0

3. Definitions

3.1 Conformance levels

3.1.1 expected: A key word used to describe the behavior of the hardware or software in the design models *assumed* by this Specification. Other hardware and software design models may also be implemented.

3.1.2 may: A key word that indicates flexibility of choice with *no implied preference*.

3.1.3 shall: A key word indicating a mandatory requirement. Designers are *required* to implement all such mandatory requirements.

3.1.4 should: A key word indicating flexibility of choice with a strongly preferred alternative. Equivalent to the phrase *is recommended*.

3.1.5 reserved fields: A set of bits within a data structure that are defined in this specification as reserved, and are not otherwise used. Implementations of this specification shall zero these fields. Future revisions of this specification, however, may define their usage.

3.1.6 reserved values: A set of values for a field that are defined in this specification as reserved, and are not otherwise used. Implementations of this specification shall not generate these values for the field. Future revisions of this specification, however, may define their usage.

The IEEE is investigating whether the “may, shall, should” and possibly “expected” terms will be formally defined by IEEE. If and when this occurs, draft editors should obtain their conformance definitions from the latest IEEE style document.

3.2 Glossary of terms

3.2.1 AM824: A 32-bit data field that has an 8-bit label and 24-bit data field defined in Audio and Music Data Transmission Protocol Ver. 1.0.

3.2.2 Audio Channel Cluster: Group of logical audio channels that carry tightly related synchronous audio information. A stereo audio stream is a typical example of a two-channel audio channel cluster.

3.2.3 Audio data stream: Transport medium that can carry audio information.

3.2.4 Byte: Eight bits of data.

3.2.5 Compound Data Block: The name for the Data Block that consists of AM824 data in any combination.

3.2.6 Conformant Data: A type of AM824 data that carries information equivalent to that defined in external specification such as IEC60958 or MIDI.

3.2.7 IEEE: The Institute of Electrical and Electronics Engineers, Inc.

3.2.8 Isochronous: A term that indicates the essential characteristic of a time-scale or signal, such that the time intervals between consecutive instances either have the same duration or duration’s that are integral multiples of the shortest duration. In the context of Serial Bus, “isochronous” is taken to mean a bounded worst-case latency for the transmission of data; physical and logical constraints that introduce jitter preclude the exact definition of “isochronous".

3.2.9 MIDI: Musical Instrument Digital Interface - an industry standard for the interconnection of music processing devices (e.g. keyboards, signal processors) and computers together. MMA (MIDI Manufactures Association, http://www.midi.org) or AMEI (Association of Musical Electronics Industry, http://www.amei.or.jp/) are contact points for the standard.

3.2.10 Music data: Data generally used for controlling a tone generator. The data defined in the MIDI specification, which may be called MIDI data, is an example of music data.

.**3.2.11 Node:** An addressable device attached to Serial Bus with at least the minimum set of control registers defined by IEEE Std 1394–1995.

3.2.12 Quadlet: Four bytes of data.

3.2.13 Stream: A time-ordered set of digital data originating from one source and terminating at zero or more sinks. A stream is characterized by bounded bandwidth requirements and by synchronization points, or time stamps, within the stream data.

3.3 Acronyms and abbreviations

A/M Protocol	Audio and Music Data Transmission Protocol.
AV/C	Audio Video Control
DVD	Digital Versatile Discs (See http://www.dvdforum.org/index.htm)
SACD	Super Audio CD (See http://www.licensing.philips.com/)

4. Reference model for AM824 data transmission

This clause describes a reference model for AM824 data transmission. This reference model may be applied to other data types defined in the A&M Protocol, if the sequence denoted as AM824 is replaced with a sequence of the desired data type.

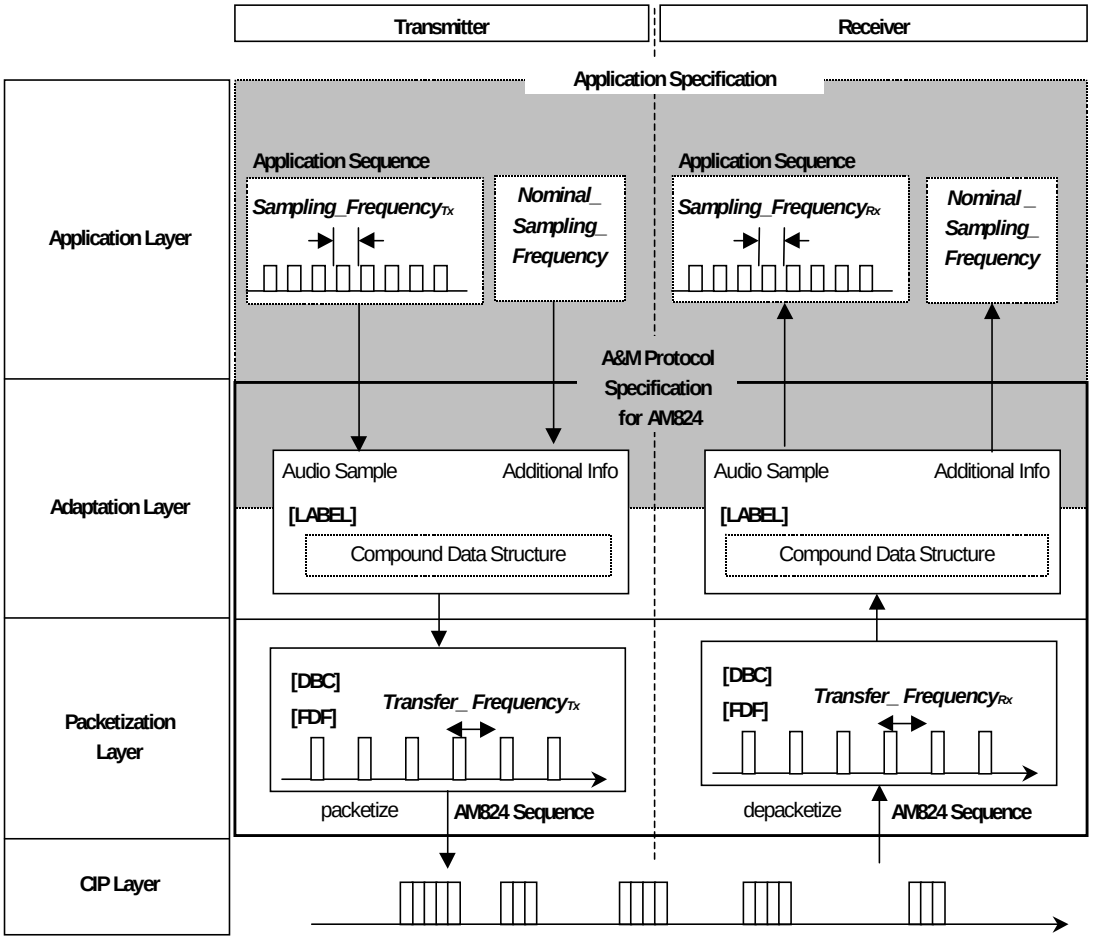

Figure 4.1 – Reference model for AM824 data transmission

This model gives an outline for audio data transmission from a transmitter to a receiver using the AM824 data type. It has four major layers denoted as CIP (Common Isochronous Packet) Layer, Packetization Layer, Adaptation Layer and Application Layer.

4.1 Application layer

Each application defines its own application sequence and the interface to the adaptation layer. The Application Sequence in Figure 4.1 – Reference model for AM824 data transmission is data in a format such as an audio signal format. The *Nominal_Sampling_Frequency* is the ideal sampling frequency for the Application Sequence. The range of *Sampling_Frequency* should be defined by the application. The audio signal at *Nominal_Sampling_Frequency* can be reproduced at the actual rate of *Sampling_Frequency* in operation. This means that the value of *Sampling_Frequency* may have some deviation and/or may vary in time in contrast with *Nominal_Sampling_Frequency*.

Additional Info in Figure 4.1 – Reference model for AM824 data transmission is any information other than events of a sequence (audio samples) being transmitted at a given rate

4.2 Adaptation layer

Adaptation Layer defines a process to convert an Application Sequence to an AM824 Sequence and vice versa. If the bit length of an audio sample of the Application Sequence is not 24 bits, some conversion between *Sampling_Frequency* and *Tranfer_Frequency* may be required (see clause 6.1). The *Transfer_Frequency* represents the frequency of occurrence of a Data Block, which is equivalent to a Cluster Event. *Transfer_Frequency* is used for describing conceptual transmission model.

The transfer rate of an AM824 Sequence is 24 * *Transfer_ Frequency* [bits/sec].

Generally Adaptation Layer is designed such that both the Application Sequence at *Sampling_Frequency* and its *Nominal_Sampling_Frequency* are carried. In this specification, *Nominal_Sampling_Frequency*, which would usually be one of the ancillary data items, is carried by SFC which is defined in clause 5.2, not by the AM824 Ancillary Data type defined in clause 7.1. In future extensions of this specification, *Nominal_Sampling_Frequency* could be carried by one of the AM824 Ancillary Data types. The information in *Nominal_Sampling_Frequency* is necessary for using command based rate control or making a copy. On the other hand, *Sampling_Frequency* is necessary for clock based rate control. Although *Sampling_Frequency* is not explicitly transmitted, it can be estimated from SYT_INTERVAL and time stamps by the algorithm specified for the AM824 Data type.

An application specification defines the process (shown in the gray shaded area in Figure 4.1 – Reference model for AM824 data transmission) to convert the application's signal (Application Sequence) to an AM824 Sequence. This document assumes that the application specification is an external document using the definition of the AM824 Label for the adaptation process. For several generic data types this document also defines the Adaptation Layer.

The adaptation to an AM824 Sequence is the point at which the packetization process interfaces to the application. The packetization process can be described as IEEE 1394 adaptation from the point of view that the data stream utilizes IEEE 1394 as its transport. The packetization process in the sense of A&M Protocol Ver. 1.0 consists of the adaptation process and AM824 Sequence packetization process. The AM824 Sequence is an application-neutral sequence.

In the sense of A&M Protocol Ver. 1.0, only straightforward adaptation processes are defined.

This model helps when new application is introduced, especially if the application uses AM824 Sequence.

More details of this layer are described in clause 6

4.3 Packetization layer

The AM824 Sequence is directly packetized to CIP or depacketized from CIP in the Packetization Layer.

Transfer_ Frequency can be implicitly expressed by the output of a locked PLL circuit as shown in Figure 4.2 – Implementation example of receiver, instead of being explicitly denoted in the Packetization Layer.

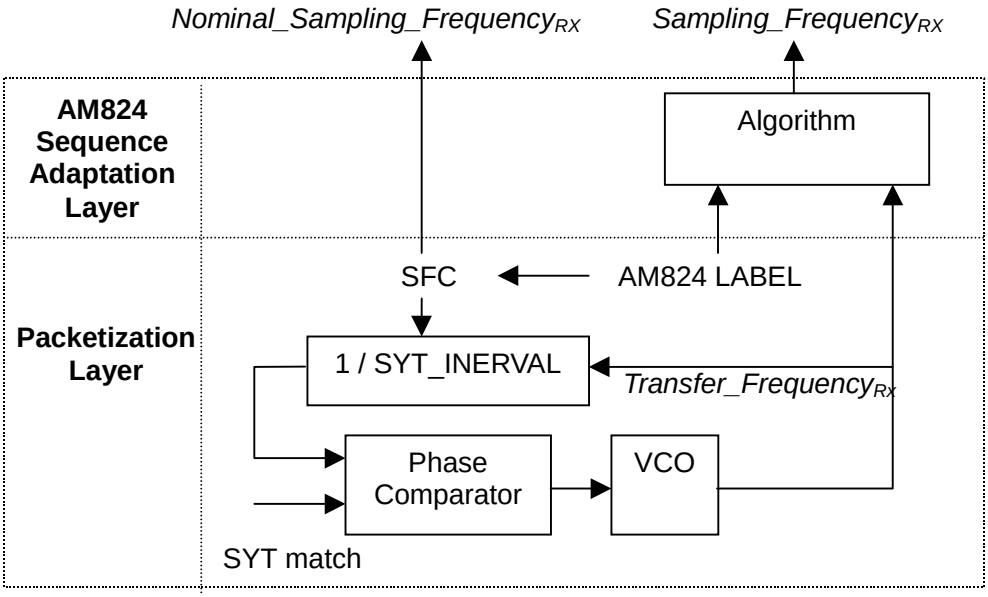

Figure 4.2 – Implementation example of receiver

5. Supplementary FDF definition

Items with gray shading are defined in this specification.

Table 5.1 – FDF definition

Value	Description
0000 $0xxx_2$	Basic format for AM824
0000 $1xxx_2$	Basic format for AM824.Transmission rate may be controlled by an AV/C command set.
0001 $0xxx_2$	Basic format for 24-bit*4 Audio Pack
0001 $1xxx_2$	- reserved -
0010 $0xxx_2$	Basic format for 32-bit Floating-Point Data
0010 $1xxx_2$	- reserved -
0011 $0xxx_2$	Reserved for basic format
0011 $1xxx_2$	- reserved -
0100 $0xxx_2$ - 1111 1110_2	- reserved -
1111 1111_2	Packet for NO-DATA

New FDF space is allocated for the AM824 data type as described in Table 5.1 – FDF definition Consequently the FDF space for the AM824 data type becomes 0000 $xxxx_2$ and its defined as shown in Figure 5.1 – New structure of FDF for AM824 data type without changing the original definition for FDF = 0000 $0xxx_2$.

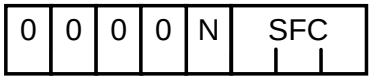

Figure 5.1 – New structure of FDF for AM824 data type

5.1 N-flag

The N-flag as shown in Figure 5.1 – New structure of FDF for AM824 data type shall be used to select the AM824 LABEL space and adaptation process described in clause 5.4.

Any AM824 data type shall occupy the same space in both LABEL spaces. An application may use only one of two LABEL spaces by giving a fixed value to the N-flag. Only an AM824 data type that owns the LABLE space or Application Specific Ancillary Data, which is defined in clause 7.1.2, can inhibit the use of one of the LABEL spaces.

5.2 Supplementary SFC definition

In A&M Protocol Ver. 1.0, there is only one SFC table that specifies both *Nominal_Sampling_Frequency* and SYT_INTERVAL.

In this specification, the SFC definition is changed so that a new AM824 Data Type which is defined after A&M Protocol Ver. 1.0 may define its own SFC table. In order to keep compatibility with the A&M Protocol Ver. 1.0, in the case of FDF = 0000 $0xxx_2$, the default SFC table shall be identical to the table defined in A&M Protocol Ver. 1.0. Only a new AM824 Data type may override the default SFC table.

The empty packet defined in [R2] shall use the default SFC table.

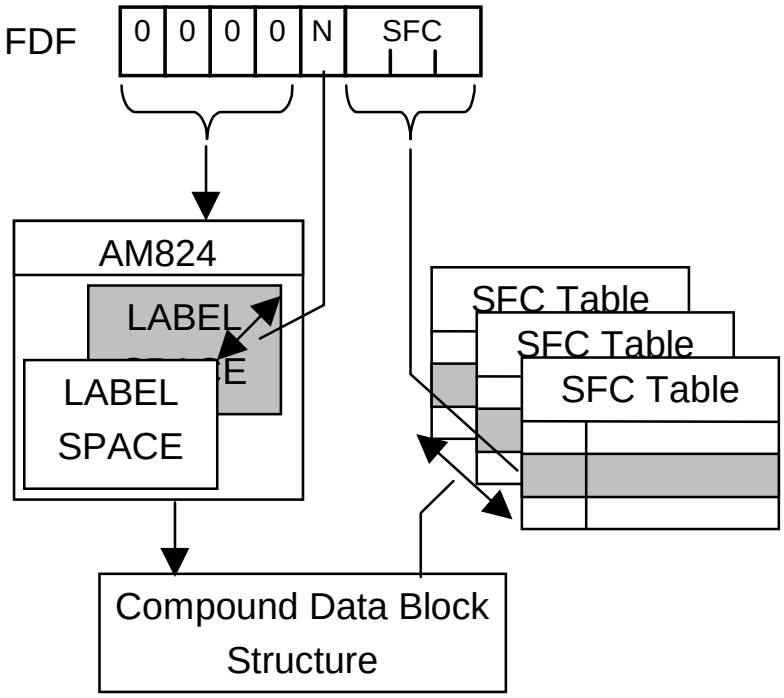

Figure 5.2 – SFC interpretation

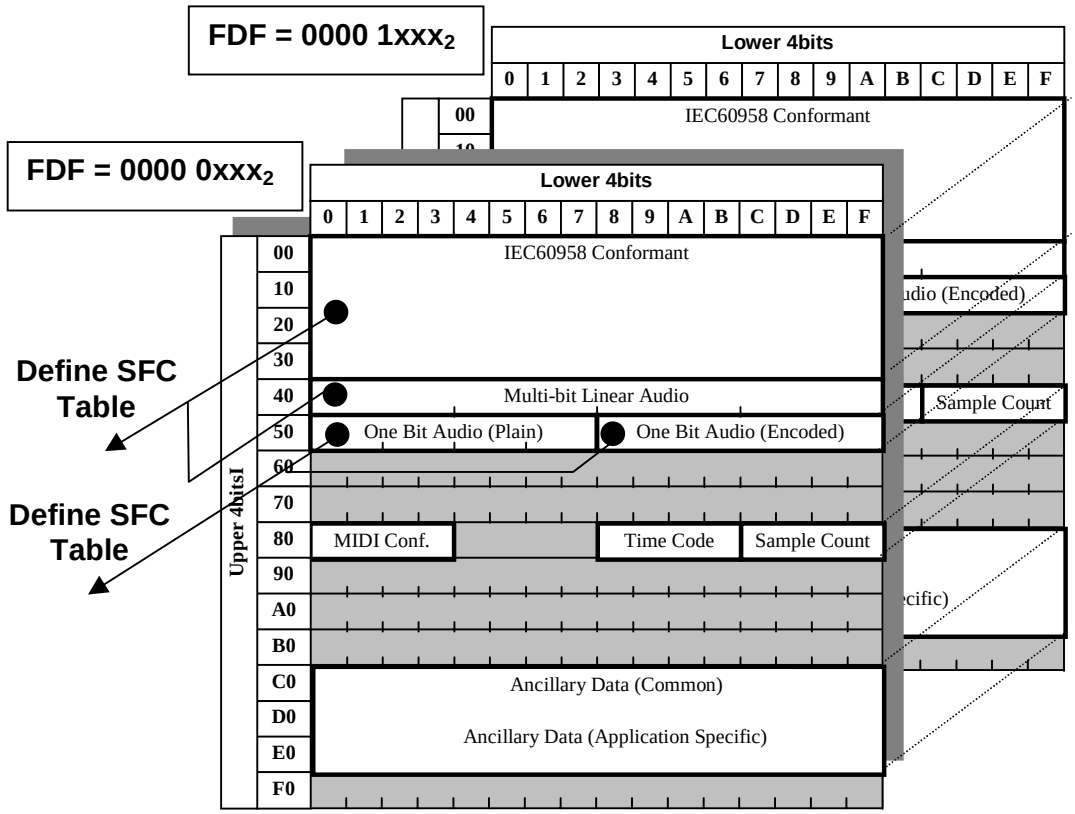

Figure 5.3 – FDF for AM824 and AM824 LABEL space (informative)

5.3 Clock based rate control mode (FDF = 0000 0xxx$_2$)

This FDF value, which is defined in A&M Protocol Ver. 1.0, is interpreted to indicate that a transmission clock reproduced by means of a timestamp controls the data transmission rate.

The meaning of this FDF value is not changed.

5.3.1 Default SFC table for FDF = 0000 0xxx$_2$

The items in Table 5.2 – Default SFC table for FDF = 0000 0xxx$_2$ with gray shading are defined in this specification.

Table 5.2 – Default SFC table for FDF = 0000 0xxx$_2$

Value	Description	
	SYT_INTERVAL	*Nominal_Sampling_Frequency*
00_{10}	8	32kHz
01_{10}	8	44.1kHz
02_{10}	8	48kHz
03_{10}	16	88.2kHz
04_{10}	16	96kHz
05_{10}	32	176.4kHz
06_{10}	32	192kHz
07_{10}	- reserved -	- reserved -

The TRANSFER_DELAY for Blocking Transmission, in the case of DEFAULT_TRANSFER_DELAY = 352, corresponds to the default SFC table as given in Table 5.3 – TRANSFER_DELAY for blocking transmission.

Table 5.3 – TRANSFER_DELAY for blocking transmission

Value	TRANSFER_DELAY
00_{10}	352 + 250.0 = 602.0[µ sec]
01_{10}	352 + 181.4 = 533.4[µ sec]
02_{10}	352 + 166.7 = 518.7[µ sec]
03_{10}	352 + 181.4 = 533.4[µ sec]
04_{10}	352 + 166.7 = 518.7[µ sec]
05_{10}	352 + 181.4 = 533.4[µ sec]
06_{10}	352 + 166.7 = 518.7[µ sec]
07_{10}	- reserved -

Items with gray color shading in Table 5.3 – TRANSFER_DELAY for blocking transmission are defined in this specification.

5.4 Command based rate control mode (FDF = 0000 1xxx$_2$)

This new allocated FDF value indicates that the data transmission rate is controlled by a command set such as *AV/C Command Set for Rate Control of Isochronous Data Flow* [R13].

This transmission mode can be used for reproducing an application sequence at a receiver or for high-speed data transfer without using a timestamp in the SYT field.

If the timing information is available, the transmitter should provide the correct timestamp in the SYT field according to the integer multiplier *n* so that the clock based rate controlled receiver can receive the data transmitted in this mode.

$SYT\_INTERVAL_{N\text{-}flag=1} = SYT\_INTERVAL_{N\text{-}flag=0} * n$ ($n >= 1$)

Where $SYT\_INTERVAL_{N\text{-}flag=1}$ and $SYT\_INTERVAL_{N\text{-}flag=0}$ denote SYT_INTERVAL specified by the SFC table in the cases in which FDF = 0000 $1xxx_2$ and FDF = 0000 $0xxx_2$ respectively. The integer multiplier *n* is obtained by a command.

5.4.1 Default SFC table for FDF = 0000 $1xxx_2$

Table 5.4 – Default SFC table for FDF = 0000 $1xxx_2$

Value (decimal)	*Nominal_Sampling_Frequency*	SYT_INTERVAL	*Samling_Frequency*
0	32 kHz	8 * *n*	32 kHz * *n*
1	44.1 kHz	8 * *n*	44.1 kHz * *n*
2	48 kHz	8 * *n*	48 kHz * *n*
3	88.2 kHz	16 * *n*	88.2 kHz * *n*
4	96 kHz	16 * *n*	96 kHz * *n*
5	176.4 kHz	32 * *n*	176.4 kHz * *n*
6	192 kHz	32 * *n*	192 kHz * *n*
7	–	–	–

The DBS of an event is independent of the transfer speed.

6. Adaptation processes for AM824 sequence

This clause describes typical methods of adaptation to an AM824 Sequence.

6.1 Basic sequence conversion

Transfer_Frequency is identical to *Sampling_Frequency* (transfer frequency of the application sequence such as audio) to be packetized if each event in the application sequence (each audio sample) is stored in one unit such as one AM824 Data of an AM824 Sequence.

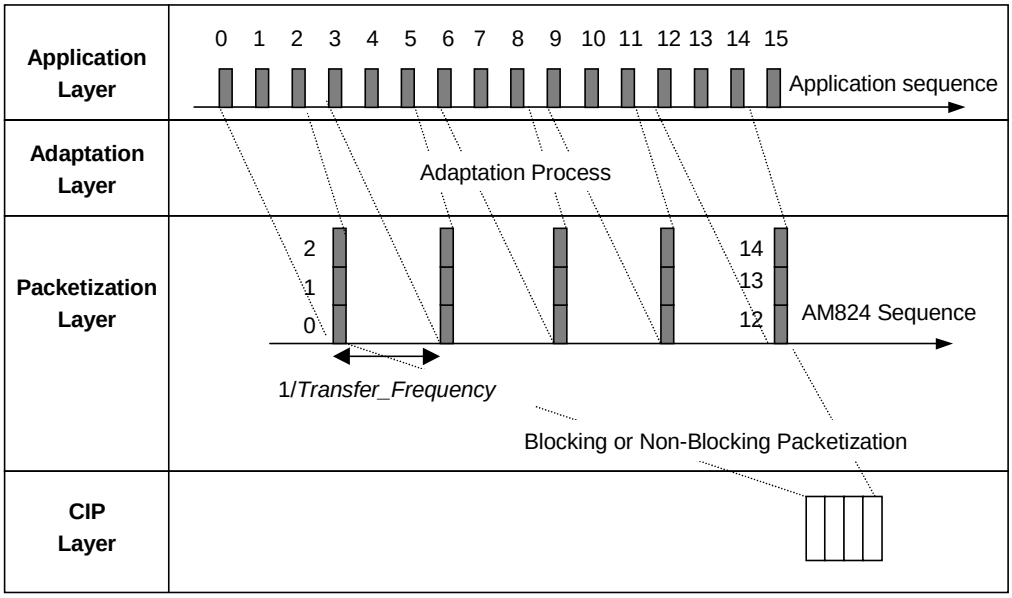

Figure 6.1 – Adaptation to AM824 sequence

Figure 6.1 – Adaptation to AM824 sequence describes an example of an adaptation process in which each event of the application sequence is 8 bits in length and three events are stored in a single AM824 data, which has a 24-bit payload. In this case, the relation between *Sampling_Frequency* and *Transfer_Frequency* is expressed by

Sampling_Frequency = L * *Transfer_Frequency*

Where L = 3.

The parameters *Sampling_Frequency*, *Transfer_Frequency* and L can not be specified independently. All of them are specified by the SFC code selected by the AM824 data type.

6.2 Sequence multiplexing

If the event occurrence rate of an application sequence is less than half of the rate of the Compound Data Block, one single A&M sequence can carry more than one application sequence by multiplexing the application sequence into a single A&M sequence assigned to the Compound Data Block. In this case each multiplexed application sequence is identified by its DBC (Data Block Count).

If the AM824 Sequence defines No-Data for padding, even an application sequence, which is asynchronous to *Transfer_Frequency*, can be adapted to the AM824 sequence. One significant example of this case is adaptation of a MIDI data stream (Application Sequence) to a MIDI Conformant sequence (AM824 Sequence).

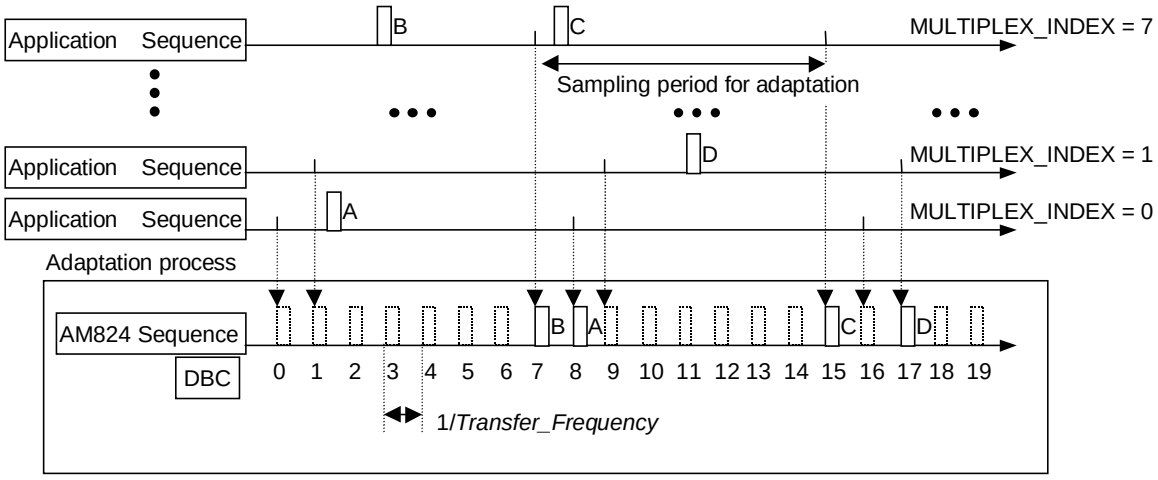

Figure 6.2 – Asynchronous sequence multiplexing

An application that uses this multiplexing shall define MULTIPLEX_NUMBER to be a power of 2. The MULTIPLEX_NUMBER is defined in conjunction with the LABEL definition because the place for carrying MULTIPLEX_NUMBER information is not defined in this document. This definition will be overridden by a future specification if it defines a method of carrying MULTIPLEX_NUMBER.

The identifier for a multiplexed sequence denoted by MULTIPLEX_INDEX is given by MULTIPLEX_INDEX = mod (DBC, MULTIPLEX_NUMBER).

7. Supplementary AM824 data types

Items with gray color shading are defined in this specification.

Table 7.1 – LABEL definition

Value	Description
00_{16} - $3F_{16}$	IEC60958 Conformant
40_{16} - $4F_{16}$	Multi-bit Linear Audio
50_{16} - 57_{16}	One-bit Audio (Plain)
58_{16} - $5F_{16}$	One-bit Audio (Encoded)
60_{16} - $7F_{16}$	- reserved -
80_{16} - 83_{16}	MIDI Conformant
84_{16} - 87_{16}	- reserved -
88_{16} - $8B_{16}$	SMPTE Time Code Conformant
$8C_{16}$ - $8F_{16}$	Sample Count
90_{16} - BF_{16}	- reserved -
$C0_{16}$ – EF_{16}	Ancillary Data
$F0_{16}$ - FF_{16}	- reserved -

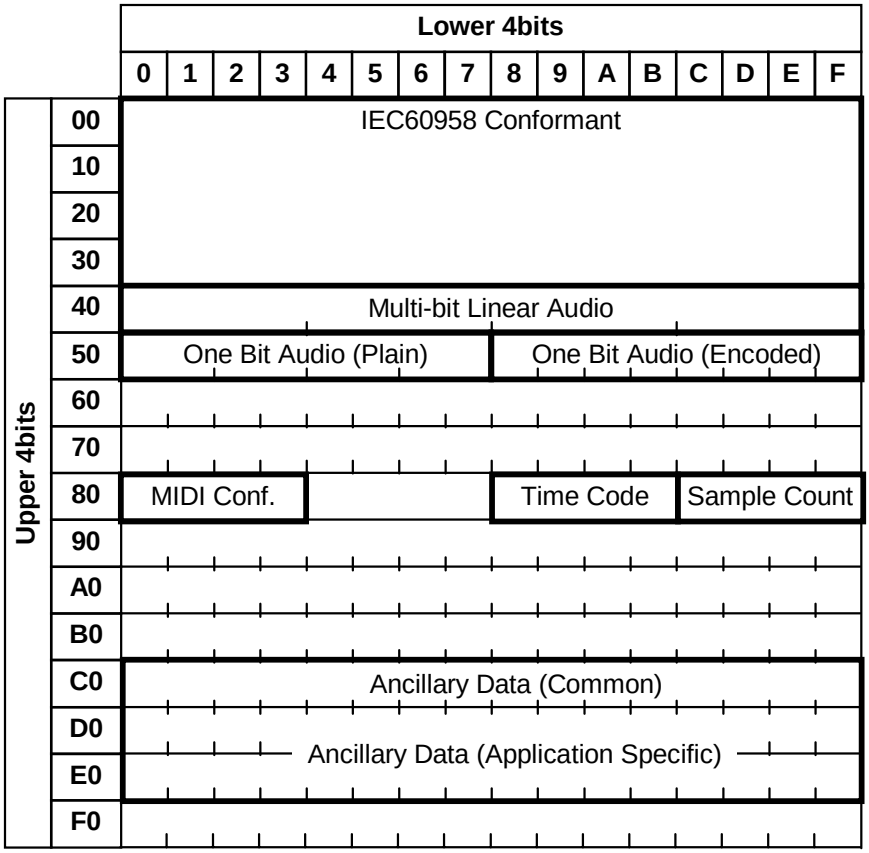

Figure 7.1 – AM824 LABEL allocation map (informative)

7.1 Ancillary data

Generic Ancillary data is illustrated in Figure 7.2 – Generic ancillary data. The definition of BYTE1, BYTE2 and BYTE3 and transmission method, timing accuracy and interval for instance, should be given by each instance of Ancillary Data. It is recommended that every information carried by Ancillary Data be transmitted repeatedly in a reasonably short interval of time while the information is valid so that the receiver does not have to wait for the information. It is recommended that BYTE1 be defined as a sub-label that specifies BYTE2 and BYTE3.

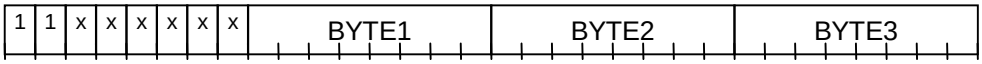

Figure 7.2 – Generic ancillary data

Table 7.2 – LABEL definition for ancillary data type

Value	Description
$C0_{16}$ - CF_{16}	Common Ancillary Data
$D0_{16}$ - EF_{16}	Application Specific Ancillary Data

7.1.1 Common ancillary data

Common Ancillary Data carries information common to all applications under a category such as copyright information.

Table 7.3 – LABEL definition for common ancillary data

Value	Description
$C0_{16}$ - CE_{16}	- reserved -
CF_{16}	Ancillary No-Data

7.1.1.1 Ancillary no-data

Ancillary No-Data provides a No-Data event only for AM824 data that does not define its own No-Data. AM824 data types that define their own No-Data shall not use this Ancillary No-Data.

In order to determine whether the AM824 data type carries valid information, it is required that No-Data specifies the AM824 data type to which it belongs. For this reason, the AM824 data type derived from a given No-Data should be identical to the AM824 data that carries valid information. The A&M Protocol Ver. 1.0 allows the use of No-Data defined in MIDI Conformant data.

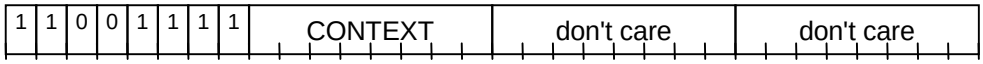

Figure 7.3 – Ancillary no-data

Table 7.4 – CONTEXT definition

Value	Description
00_{16}	No-Data for IEC60958 Conformant
01_{16} – $3F_{16}$	-reserved-
40_{16}	No-Data for Multi-bit Linear Audio
41_{16} – $4F_{16}$	-reserved-
50_{16}	No-Data for One-bit Audio (Plain)
51_{16} – 57_{16}	-reserved-
58_{16}	No-Data for One-bit Audio (Encoded)
59_{16} – $5F_{16}$	-reserved-
60_{16} – $7F_{16}$	-reserved-
80_{16} – 83_{16}	-reserved-
84_{16} –87_{16}	-reserved-
88_{16} –$8F_{16}$	-reserved-
$C0_{16}$ – CE_{16}	No-Data for each 7 different common ancillary data
CF_{16}	No-Data for unspecified type. This shall be used only for the purpose described in 8.1
$D0_{16}$ – EF_{16}	No-Data for each 32 different application specific ancillary data
$F0_{16}$ – FF_{16}	-reserved-

7.1.2 Application specific ancillary data

Application Specific Ancillary Data carries information specific to an application, which is transmitted along with the audio and music data. Examples are: mapping of sequence of a Compound Data Block to speaker location, microphone location or signal name.

Table 7.5 – LABEL definition for application specific ancillary data

Value	Description
$D0_{16}$	DVD-Audio
$D1_{16}$	SACD
$D2_{16}$ – EF_{16}	- reserved -

The general format for Application-specific Ancillary Data is shown in Figure 7.4:

Figure 7.4 – General Format for Application-specific Ancillary Data

The first byte (“LABEL”) indicates that this data is for application-specific ancillary data of the type shown in Table 7.5. The second byte (“Sub LABEL”) further identifies the particular data that follows. For details, see section 9.2 (for DVD Audio) and section 9.3 for SACD

8. Compound data block structure

Compound Data Block is the name for the Data Block that consists of AM824 data in any combination, if all the AM824 data in the Data Block specify the same SFC table. (Note that the SFC value in a CIP specifies the entry of the SFC table selected according to AM824 Data type that defines the SFC table.)

Thus the cluster, which is equivalent to a Data Block in the context of AM824 data, can be referred to as a Compound Cluster.

Each sequence carried by a Compound Data Block is uniquely identified by the location of events in the Compound Block.

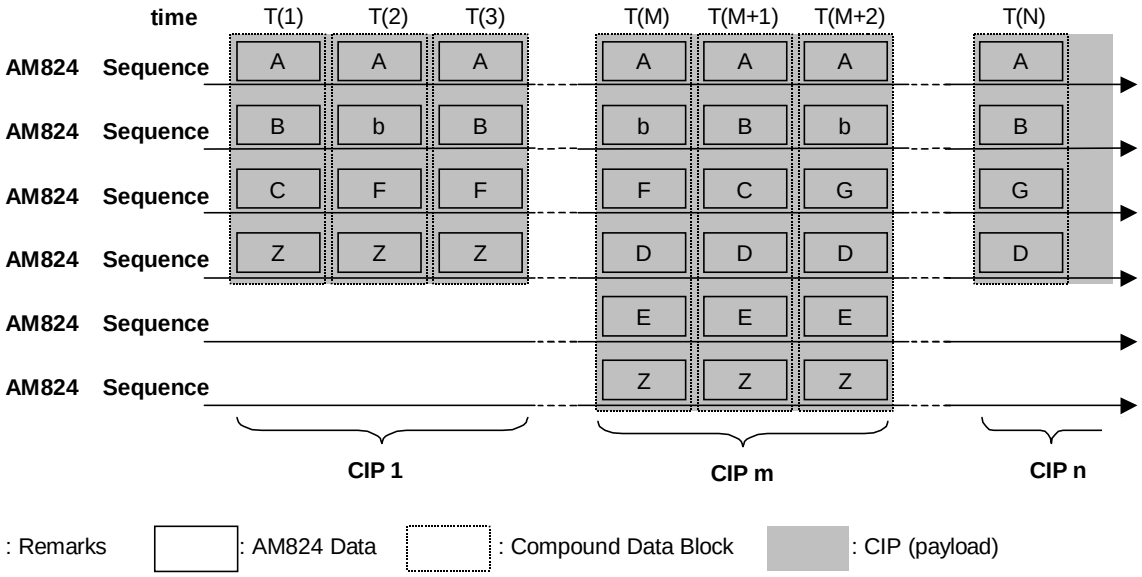

Figure 8.1 – Example of compound data block

An example of usage of Compound Data Block is illustrated in Figure 8.1 – Example of compound data block.

The capital letter, 'B' for example, in the box of AM824 Data, represents the box's data type. The small letter, 'b' for example, in the box of AM824 Data, denotes "No Data" for same data type.

DBS (Data Block Size) or CLUSTER_DIMENSION may vary in time. Also, the AM824 Data type described in the LABEL field of each event may vary in time.

8.1 Compound data structure rule

A&M Protocol Ver. 1.0 allows any order of AM824 data type in a Compound Data Block. In order to maintain minimum connectivity, this clause defines rules for the Compound Data structure, or in other words, a rule for AM824 sequence configuration. Also, some recommendations for implementation are described.

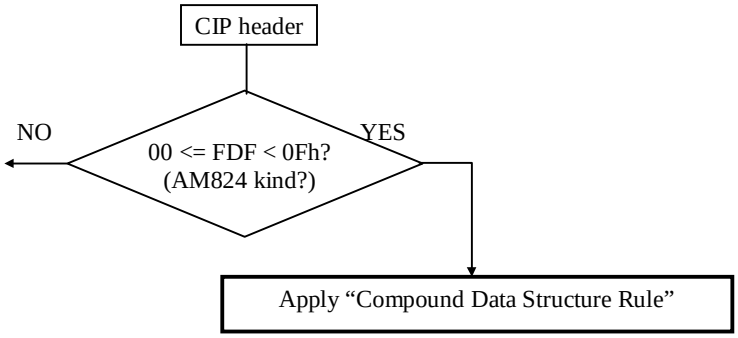

Figure 8.2 – Condition of AM824 rule

8.1.1 Size rule

The size of Compound Data should be an even number of quadlets.

If the number of quadlets in a sequence required by an application is not an even number, an unspecified sequence (sequence of Ancillary No-Data with CONTEXT = CF_{16}) should be added to make the number of quadlets in the sequence even. Figure 8.1 – Example of compound data block shows a Compound Data Block compliant to this rule where the event denoted by “Z” is interpreted as Ancillary No-Data with CONTEXT = $CF_{16.}$ As long as the number of quadlets in a sequence is even, any number of unspecified sequences may be added.

8.1.2 Order rule

Application Specifier is either Application Specific Ancillary Data or any Common Ancillary Data except Ancillary No-Data for non-Ancillary Data. *Content Data* is any AM824 data other than *Application Specifier*.

A Compound Data Block starts with zero or only one *Unspecified Region* followed by zero or one or more *Specified Region*(s). *Unspecified Regi*on includes only *Content Data*. *Specified Region* starts with one or more *Application Specifiers* followed by one or more *Content Data* before encountering the next *Application Specifier* or the end of the Compound Data Block.

A sequence of *Application Specifiers* may contain both Common Ancillary Data and Application Specific Ancillary Data by multiplexing.

The order of the Content Data in an Unspecified Region shall be determined by following formula:

IEC60958 Conformant Data < Multi-bit Linear Audio < MIDI Conformant Data < SMPTE Time Code < Sample Count.

Within an Unspecified Region the same data type should occupy a contiguous area.

The order inside a Specified Region is defined by the application specified in Application Specific Ancillary Data. The Specified Region shall have none or only one Common Ancillary Data or one or more Application Specific Data for the same application.

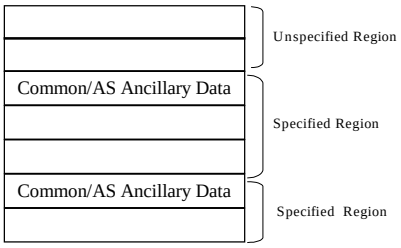

Figure 8.3 – Generic compound data block structure

IEC60958 Conformant L-ch
IEC60958 Conformant R-ch
MBL Audio Data 1-ch
MBL Audio Data 2-ch
MBL Audio Data 3-ch
MBL Audio Data 4-ch
MIDI Conformant Data

Figure 8.4 – Example of unspecified region structure

8.2 Recommendation: general

Because 2-channel stereo audio is widely accepted, it is highly recommended that for devices, which transmit audio in any format, the first 2 sequences be linear audio either in IEC60958 Conformant or Raw Audio. The first sequence should be left and second should be right. If a transmitter is a monaural audio device, it may send the audio in left channel and silent data in the right, or send the audio in both left and right. It is implementation-dependent.

If a transmitter is a multi-channel audio device, it may send downmixed in 2-channel stereo audio in addition to the multi-channel audio.

8.3 Recommendation for transmitter

1) DBS (Data block size in quadlets) should be greater or equal to 2. An even number is the most preferable.

2) At the top of the Data Block of mixed Audio and Music data, Stereo Left Channel, then Right Channel should be transmitted.

3) In Data Blocks of multichannel audio data, the first two quadlets should be the main channels corresponding to Stereo Left and Right Channel.

4) Recommendation for stream change method is as follows.

   When the content stream is changed, it is preferable to insert Ancillary no-data or Empty packets at the change point of the stream.

   The change point of the stream is not a pause of each tune in the CD album, but it implies that at the point, some change of e.g. compression methods occurs.

   The purpose of insertion of Ancillary no-data or Empty packets is to prevent losing the end portion of the previous stream and the beginning of the next stream.

The general recommendation method is described as follows.

It is desirable to output Ancillary no-data with the previous CONTEXT of 10ms or more following the previous stream.

Afterwards, when the next stream can be recognized beforehand, the insertion of Ancillary no-data with the next CONTEXT is recommended.

Otherwise, the insertions of Ancillary no-data with next CONTEXT are not needed.

That is, the Ancillary no-data with previous CONTEXT can be changed to the following stream directly.

And, when the transmission device does not have the capability of outputting the Ancillary no-data with previous CONTEXT and the Ancillary no-data with next CONTEXT, the transmission device can output MIDI no-data or Empty packets or stop the stream output.

When the Empty packets is output to prevent losing the beginning of the following Content, it is preferable to add time stamp information in SYT.

8.4 Recommendation for receiver

1) Stereo products that receive multichannel streams with DBS >= 2 should reproduce the sound of the first two channels of the Data Block as Stereo Left and Right channels.

2) Stereo products which have no non-linear PCM decoder should reproduce no (muted) sound when they receive Validity Flag = '1' in IEC 60958 Conformant Data.

9. AM824 sequence adaptation layers

The transport mechanism using CIP may be used as an alternative transport layer for an existing data transmission protocol such as IEC60958 and MIDI.

This adaptation layer definition defines only one-to-one mapping between an application data structure and an AM824 data structure and a procedure for transporting the application data only with a constant time shift.

The definition of the adaptation to CIP can be described and maintained by either organization responsible for the adaptation.

The adaptation layer definition described in this document provides only an alternative transport. The meaning of the data carried by the transport should be given in the original specification. Also, the transmission rate should be identical to that which is originally specified when the "nonidentical to sampling frequency" indication flag is off.

The Adaptation Layer definition falls into two categories. One is generic that can be used in applications and does not define Application Specific Ancillary Data. Another is application specific that defines the structure of the Compound Data Block and Application Specific Ancillary Data.

9.1 General

9.1.1 IEC60958 bit stream ([R4][R5][R7])

Any modification or enhancement is prohibited in this adaptation layer, although an increase in the number of audio channels or transmission rate can be easily done. Multiple IEC60958 bit streams may be allowed.

9.1.2 Multi-bit linear audio (MBLA)

The "Raw Audio" data type is merged with a new "Multi-bit Linear Audio" data type that can give a clearer meaning. The original definition of "Raw Audio" remains the same.

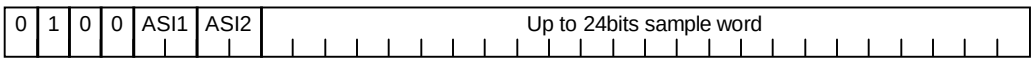

Figure 9.1 – MBLA data

The label field of MBLA has two fields for ASI (Application Specific Information). The definition of ASI2 depends on ASI1 value described in Table 9.1 – ASI1 definition.

Table 9.1 – ASI1 definition

Value	Description
00_2	Raw Audio. Sample word can be fed directly to a D/A converter. Ancillary Data may accompany. The definition of ASI2 is identical to VBL (Valid Bit Length) defined in A&M Protocol Ver.1.0.
01_2 - 11_2	Application Specific Information. Sample word may be fed directly to a D/A converter but in some processing required according to the application identified by application specific Ancillary Data which shall appear in the same Data Block. The definition of ASI2 field also shall be given by the application such as DVD-Audio described in 9.2.1

9.1.3 One-bit audio

In this clause, the format of One-bit Audio is described.

Table 9.2 – LABEL definition for one-bit audio (plain)

Value	Description
50_{16}	One-bit Audio Stream: Multi-Channel Cluster Start data
51_{16}	One-bit Audio Stream: Multi-Channel Cluster Continuation data
52_{16} - 57_{16}	- reserved -

Table 9.3 – LABEL definition for one-bit audio (encoded)

Value	Description
58_{16}	DST: Encoded One-bit Audio stream
59_{16} - $5F_{16}$	- reserved -

9.1.3.1 One-bit audio (plain) : LABEL=50_{16}-51_{16}

The data of the One-bit Audio (Plain) has one-bit length data stream, and can be directly played back through the analog low pass filter bit by bit (MSB First). The data stream is packed in 24-bit data fields of an AM824 quadlet with MSB First per audio channel.

The sampling frequency of the One-Bit Audio (50_{16}- 51_{16}) is defined in Table 9.4 with its own SFC table.

Table 9.4 – Sampling frequency definition of one-bit audio (50_{16},51_{16},58_{16})

Value of SFC	SYT_INTERVAL	Sampling Frequency
00	16	2.048MHz
01	16	2.8224MHz
02	32	3.072MHz
03	32	5.6448MHz
04	64	6.144MHz
05	64	11.2896MHz
06	128	12.288MHz
07	- reserved -	- reserved -

One-Bit Audio (Plain) can transmit Multi-Channel Cluster. Each AM824 quadlet carries the data for one channel of the cluster. Two AM824 Labels are used to indicate the Start and Continuation of the data in the cluster.

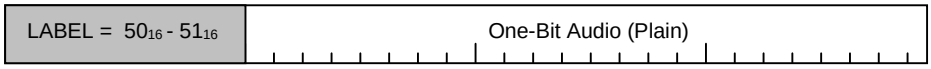

Figure 9.2 – Generic one-bit audio quadlet

The channel number shall start with No.1 and be sequential:

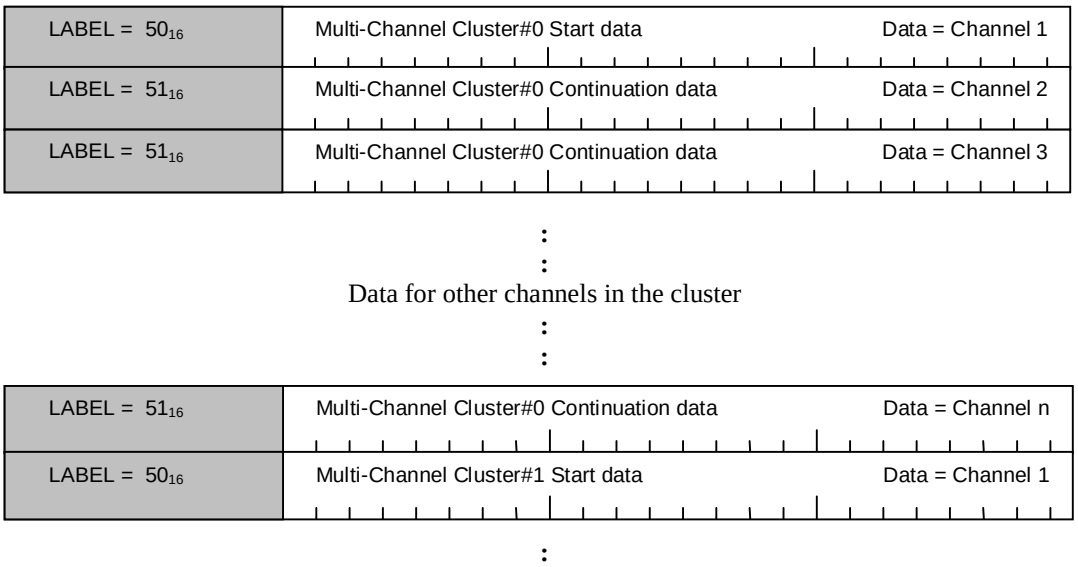

Figure 9.3 – Generic one-bit audio quadlet sequence

9.1.3.2 One-bit audio (encoded)

The data of the One-Bit Audio (Encoded) is the encoded data stream.

9.1.3.2.1 DST : LABEL=58_{16}

DST (Direct Stream Transfer) is the loss-less coding technique used for One-bit Audio in SACD, and is defined in the [B1]Part 2.

The encoded data stream is packed in 24-bit data fields of AM824 Data with MSB First.

For decoding the stream, SACD Ancillary data is needed. DST supports multi channel One-bit Audio and carries each data stream in one mixed stream.

DST encodes the One-bit Audio data stream Frame by Frame. The Frame is defined in the [B1]Part 2.

The sampling frequency of the DST is defined in Table 9.4 with its own SFC table.

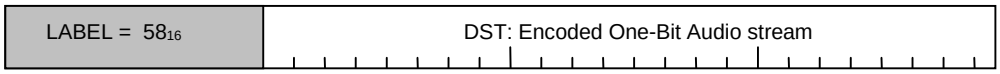

Figure 9.4 – One-bit Audio DST encoded quadlet

9.1.3.3 High speed transfer for one-bit audio(LABEL:50_{16},51_{16},58_{16})

As far as One-bit Audio (LABEL: 50_{16}, 51_{16} ,58_{16}) ,the transfer frequency and SYT_INTERVAL for the high speed AM824-data transfer are defined depending on the speed as shown in Table 9.5 if N-flag in the FDF is 1. In this table, an integer value of *n* (>1) indicates the number of times faster than normal speed.

Table 9.5– SFC definition of one-bit audio (LABEL: 50_{16}, 51_{16} ,58_{16}) for high speed AM824-data transfer

Value of SFC	*Nominal_Sampling _Frequency*	SYT_INTERVAL	*Sampling_Frequency*
0	2.048 MHz	16 * *n*	2.048 MHz * *n*
1	2.8224 MHz	16 * *n*	2.8224 MHz * *n*
2	3.072 MHz	32 * *n*	3.072 MHz * *n*
3	5.6448 MHz	32 * *n*	5.6448 MHz * *n*
4	6.144 MHz	64 * *n*	6.144 MHz * *n*
5	11.2896 MHz	64 * *n*	11.2896 MHz * *n*
6	12.288 MHz	128 * *n*	12.288 MHz * *n*
7	- reserved -	–	–

The DBS of an event is independent of the transfer speed.

9.1.4 Non-linear audio data stream

Any non-linear audio data carried by a IEC61937 bitstream can be transmitted by using the IEC60958 Conformant data sequence.

9.1.5 MIDI data stream

Any modification or enhancement is prohibited on this adaptation layer although increase of transmission rate for instance can be easily done. The specification that uses this adaptation layer is given in [R9].

This specification restricts the packetization of MIDI data stream so that a single MIDI Conformant sequence can carry multiple MIDI data streams by multiplexing. MIDI Conformant data defines MULTIPLEX_NUMBER = 8.

NOTE — The Default MULTIPLEX_NUMBER for MIDI Conformant AM824 types may be incompatible with some applications conforming to IEC PAS 61883-6.

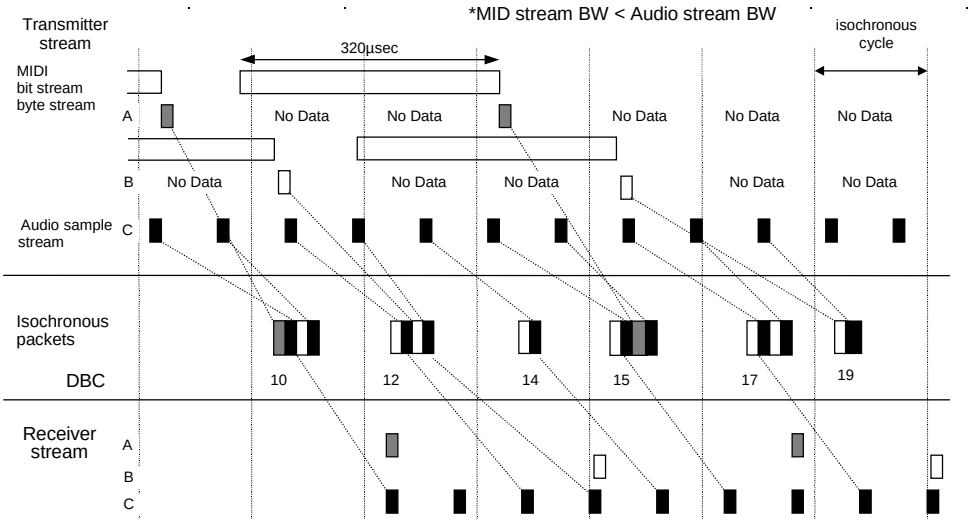

Figure 9.5 – Multiplexing of MIDI data streams (informative)

NOTE — Figure 9.5 – Multiplexing of MIDI data streams (informative) shows how two MIDI data streams, which should flow in different MIDI cables, are multiplexed in a single MIDI Conformant sequence with an audio stream. This figure is intended to give only the sequence multiplexing scheme. The parameters of this example such as the number of multiplexed sequences and the audio sampling rate were chosen so that the figure would be readable. Consequently, not all the parameters are valid for this specification and its predecessor.

9.1.6 SMPTE time code and sample count

SMPTE time code and sample count transmission are defined in a separate document [R11]

9.2 DVD-audio

The compound data for DVD-Audio consists of Multi-bit Linear Audio data, Common Ancillary and DVD-Audio specific ancillary data.

9.2.1 Multi-bit linear audio data

DVD-Audio data use the LABEL from 48_{16} to $4F_{16}$ of Multi-bit Linear Audio and use ASI2 for scaleable contents.

Table 9.6– ASI2 definition for DVD-Audio

Value	Description
00_2	24 bits
01_2	20 bits
10_2	16 bits
11_2	Previous Sample Word Data Hold

9.2.2 DVD-Audio specific ancillary data

This clause specifies private header data that are carried by DVD-Audio specific ancillary data (informative).

9.2.2.1 Data transmitted at starting point

This ancillary data is used at the starting point of audio data when performing play start or search for a track number.

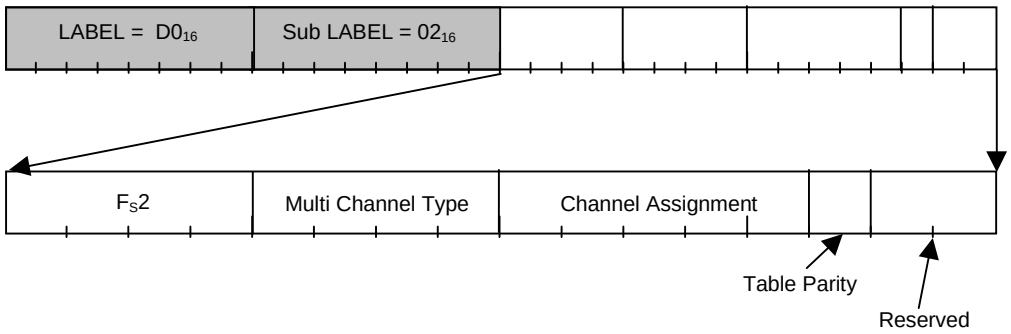

Figure 9.6 – Data transmitted at data starting point

Table 9.7 – Data transmitted at starting point

Data	Bits	Description
Fs2	4	Sampling Frequency Group2
Multi Channel Type	4	Fs, Bit combination table
Channel Assignment	5	Channel combination of Group1 and 2
Table Parity	1	Table Parity of audio data

9.2.2.2 DVD-Audio Specific Ancillary Data

This ancillary data is transmitted at every data block.

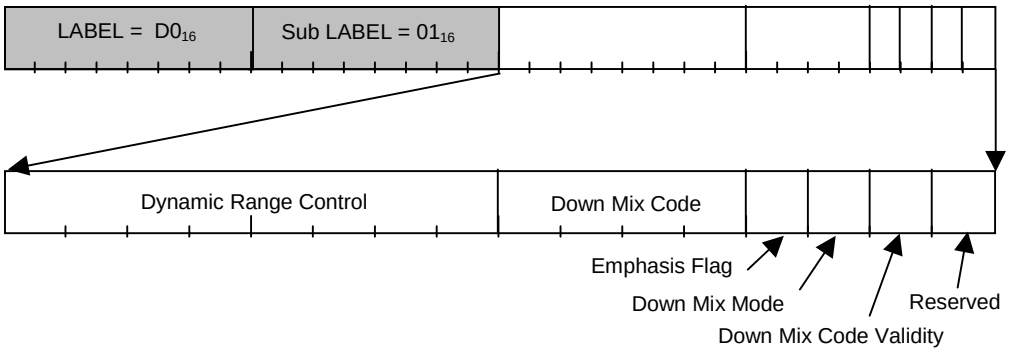

Figure 9.7 – Data transmitted at every data block

Table 9.8 – Data transmitted at every data block

Data	Bits	Description
Dynamic Range Control	8	Adaptive compression coefficient
Down Mix Code	4	Down Mix Table number
Emphasis Flag	1	Enhances on or off
Down Mix Mode	1	Down Mix permission
Down Mix Code Validity	1	Down Mix Code validity

9.2.3 Example of DVD-Audio stream

The following figure illustrates a typical multi-channel DVD-Audio stream carried over the 1394 bus.

Data on the disc is organized into a series of blocks. The data for each channel is packed into one block.

Each data block should be ordered by increasing channel number.

DVD-Audio ancillary data is immediately followed by the data block. The first ancillary data is “the data transmitted at every data block,” and the second ancillary data is “the data transmitted at the data starting point” or “ Table Parity” or “ DMCT (Down Mix Coefficient Table)” or something similar.

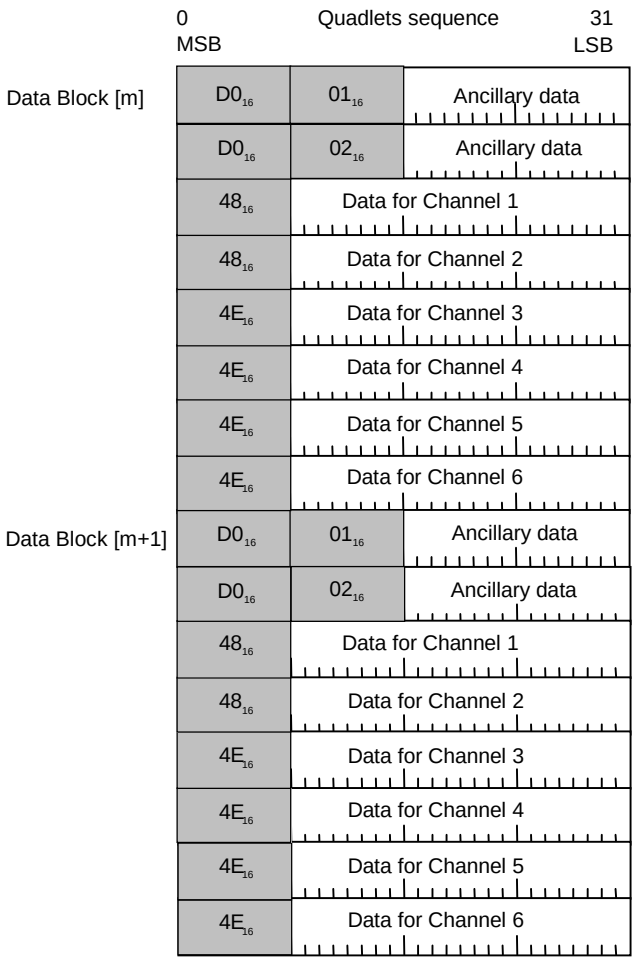

Figure 9.8 – Example of DVD-Audio compound data

9.3 SACD

The compound data for SACD consists of One-bit Audio data, Common Ancillary and SACD specific ancillary data.

9.3.1 SACD ancillary data : AM824 LABEL=$D1_{16}$: Sub LABEL=00_{16}

The SACD player transmits SACD Ancillary data at the starting point of every Frame. The Frame is defined in the [B1]Part 2. The SACD Ancillary Data contains the information about the data within the Frame.

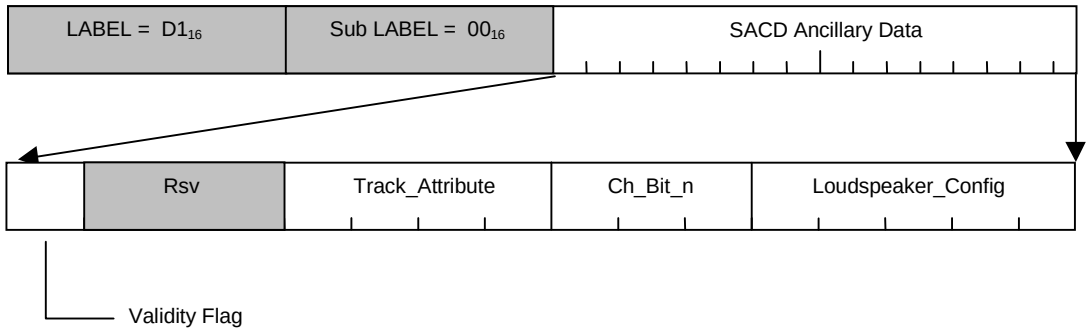

Figure 9.9 – SACD ancillary data

Table 9.9 – data information (informative)

Data	Bits	Description
Validity Flag	1	Valid or not Valid
Track_Attribute	4	Copy Control Information
Ch_Bit_n	3	Number of channels
Loudspeaker_Config	5	Loudspeaker set-up

The Validity Flag shows the validity of the data within the Frame.

If a disc read error occurs, the SACD player shall replace the error data with safe data, such as a mute signal, and set the Validity Flag to 1_2.

Table 9.10 – Validity flag definition

Value(binary)	Description
0_2	Valid
1_2	Not valid

Rsv is the reserved area and the default value is 000_2.

The Track_Attribute shows copy control information dedicated to Super Audio CD, and is defined in [B1] Part 2. This information shall be copied from the Super Audio CD track by track.

Ch_Bit_n shows the total number of channels, and is defined in [B1] Part 2. This information shall be copied from the Super Audio CD Frame by Frame.

Loudspeaker_Config shows the loudspeaker set-up, and is defined in [B1] Part 2. This information shall be copied from Super Audio CD track by track.

9.3.2 SACD supplementary data : AM824 LABEL=$D1_{16}$: Sub LABEL=01_{16}

SACD Supplementary data is a synchronized stream along with the Audio data from the SACD. It has several data lengths as defined in the [B1] Part 2 “Supplementary data”. Audio Data and Supplementary data are synchronized on a Frame by Frame basis.

For decoding the stream, SACD Ancillary data is needed.

Figure 9.10 – SACD supplementary data

9.3.3 Example of SACD streams

Figure 9.11 – Example of plain one-bit audio with an even number of channels illustrates a typical multi-channel Plain One-Bit Audio stream carried over the 1394 bus from SACD for the case where the value of SFC in FDF is 001_2. The data on the disc is organized into a series of frames, with 75 frames for each second of audio. Each frame contains a total of 1568 * 3 bytes of Audio Cluster Data per channel. Quadlets are organized according to the “Order Rule”, so that the order is Ancillary Data first, Multi-Channel Cluster data next, and a Ancillary No-Data with CONTEXT = CF_{16} last (if it is needed in a cluster).

The SACD Ancillary Data starts, and is followed by the first group of Multi-Channel Cluster data. In this example, the first quadlet contains the Ancillary Data for the whole of Frame #0. If, for example, there is a disc error, the SACD player sets the Validity Flag in the Ancillary Data for this Frame (Frame #0) which remains valid until the next SACD Ancillary Data (Frame #1). This also applies to the Track_Attribute, Ch_Bit_n and Loudspeaker_Config contained in the Ancillary Data for Frame #0.

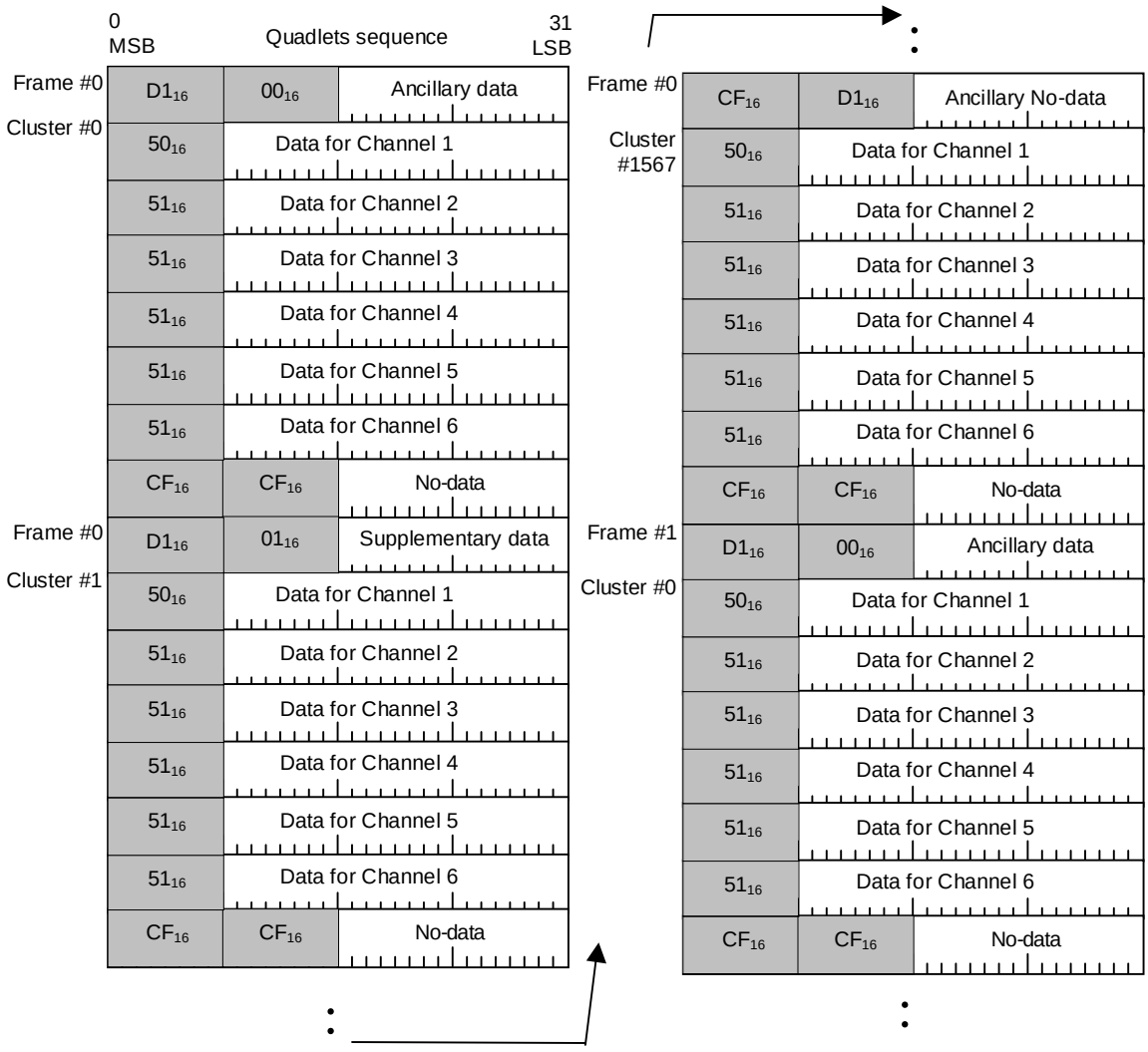

Figure 9.11 – Example of plain one-bit audio with an even number of channels

In the example of Figure 9.11 – Example of plain one-bit audio with an even number of channels, there is an even number of channels in the Multi-Channel Cluster, so a Ancillary No-Data with CONTEXT = CF may be added to the last of the cluster data so that the total number of quadlets in the cluster is kept even. SACD Supplementary data is transmitted at the same location as SACD Ancillary Data (after the SACD Ancillary Data has already been transmitted). After all the SACD Supplementary data has been transmitted, an Ancillary No-Data or other Ancillary data quadlet may be put in the same location.

Figure 9.12 – Example of plain one-bit audio with an odd number of channels shows an odd number of channels, with supplementary data. Here, Ancillary No-Data is not required.

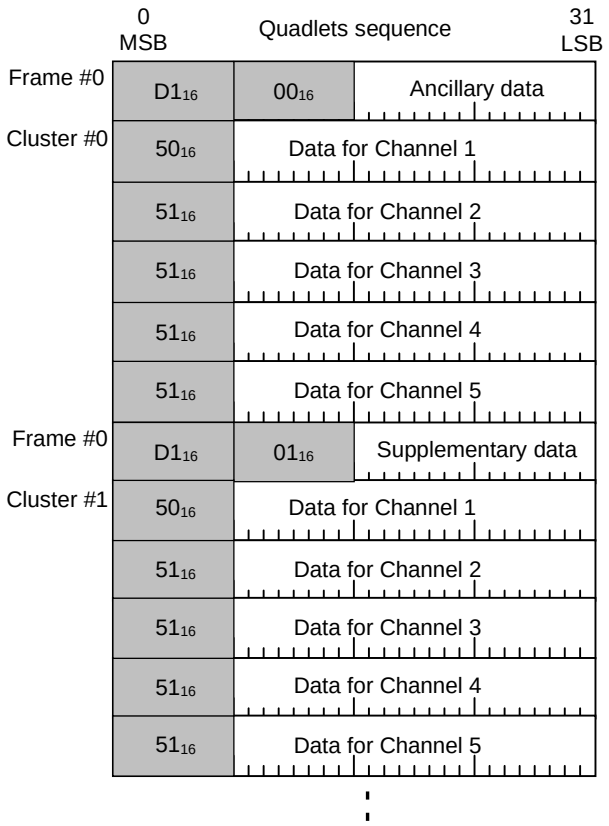

Figure 9.12 – Example of plain one-bit audio with an odd number of channels

The same rules apply to DST Encoded data. DST data should always consist of an even number of quadlets, with dummy data added to Ancillary and Supplementary data as needed.

This page is left blank intentionally

Annexes

Annex A: Bibliography (informative)

[B1] Super Audio CD System Description Version 1.0

[B2] DVD Specifications for Read-Only Disc Part 4, Audio Specifications Version 1.0 March 1999

[B3] DVD Specifications for Read-Only Disc Part 4, Audio Specifications Version-up Information (from 1.0 to 1.1) May 1999

Annex B: Synchronization (informative)

Generally the term "synchronization" addresses these issues:

1) Rate matching between transmitter and receiver
2) Presentation time adjustment at the receiver side
3) Location adjustment at the transmitter side.

The rate matching between the transmitter and receiver can be done by one of two methods:

1) Clock-based rate control
2) Command-based rate control (see clause5.4).

Clock-based rate control may use sampling clock delivery in an isochronous stream or another clock delivery system such as a dedicated clock.

The presentation time adjustment of the application sequence at a receiver can be done since the time stamp of a CIP is defined such that it reflects the time when the corresponding audio sample goes out of a buffer for depacketization. If an application requires precise adjustment of the presentation time, the application should take into account the extra delay caused by signal processing or A/D and D/A conversion.

Annex C: Transport characteristics (informative)

C.1 Sampling clock jitter characteristics

Sampling clock jitter can degrade the accuracy of conversion processes in sampling devices. This part of the annex describes the jitter mechanisms in the exchange of sample timing information and derives worst-case jitter levels to be used for stressing sampling devices when making performance measurements.

This issue applies to systems that require a sample clock to be transferred across the bus to a sampling device. For example, it does not apply for devices that use flow control with a single sampling device acting as destination and synchronization master, or where the destination device is a non-sampling device such as a recorder.

C.1.1 Definitions

C.1.1.1 Sample clock

The reference used at a sampling device to define the instant at which an audio data sample word is valid. For oversampled conversion systems the sample clock is multiplied up to the oversampling rate. Inside an asynchronous sampling frequency converter (ASFC), one sample clock is represented numerically by the relationship it has to another sample clock.

C.1.1.2 Sampling frequency, Fs

This is the frequency of the sample clock.

C.1.1.3 Sample clock timing transfer

This is the mechanism by which the sample clock of one device can be derived from a clock on another device such as by using an embedded synchronization clock.

C.1.1.4 Embedded synchronization clock

An Embedded synchronization clock is the signal that carries information that is used by a sampling device to derive a sample clock. In the context of A/M protocol this synchronization clock is embedded in the SYT field of the CIP and carries timing information that refers to local CYCLE_TIME register values.

C.1.1.5 Synchronization clock frequency, Fsync

The embedded synchronization clock frequency using the A/M protocol has to be less than the isochronous cycle rate of 8kHz. The rate is defined as the following:

Fsync = Fs / SYT_INTERVAL

The SYT_INTERVAL value is defined in the CIP header for each sampling frequency.

C.1.1.6 Sampling device

A device that depends on the timing of a sample clock to modify an audio signal in some way as it is being converted between the analog and digital domains, or between two independent sampling frequencies. Examples of a sampling device are an analog to digital converter (ADC), a digital to analog converter (DAC) and an ASFC.

C.1.1.7 Non-sampling device

Devices that do not use clock timing in a way that may modify the analog or digital audio signal. Any clocks that they use do not affect the accuracy of that data in normal operation. (Compare with sampling device).

C.1.1.8 Synchronization clock source

A device that supplies an embedded synchronization clock that another device uses to derive a sample clock. This does not need to be a source device for audio data.

C.1.1.9 Synchronization clock destination

A device that supplies an embedded synchronization clock that another device uses to derive a sample clock. This does not need to be a source device for audio data.

C.1.1.10 Clock jitter

This is the deviation in the timing of clock transitions when compared with an ideal clock. The ideal clock can be considered to have a frequency of exactly the same long-term average frequency and aligned for zero means phase offset from the real clock. For a sample clock, the jitter amplitude defined in this way is directly related to the amplitude of the jitter modulation products produced in a sampling device.

C.1.1.11 Embedded synchronization clock jitter

Jitter in the embedded synchronization clock includes the effect of errors (including limited precision) in the embedded SYT data and jitter in the CYCLE_TIME register used to decode the SYT.

C.1.2 Sample clock transfer jitter mechanisms using A&M protocol

The A&M protocol and the IEEE1394 bus use asynchronous clocks to define and exchange timing and synchronization information. The changing phase relationships and limited timing resolution of these clocks, and in some circumstances, the changing phase relationship to an external sample clock, produce a variable error which introduces jitter into an embedded synchronization clock.

There are other sources of jitter including oscillator phase noise, variable gate delays and cable inter-symbol interference. These are normally small in comparison with the mechanisms considered here.

C.1.2.1 CYCLE_TIME register jitter

Embedded synchronization clock information is referenced to the CYCLE_TIME register value at the synchronization clock source. Jitter on this register value at the synchronization clock source and synchronization clock destination nodes contributes to embedded synchronization clock jitter.

C.1.2.1.1 Cycle start packet CYCLE_TIME resolution

The cycle start packet issued from the cycle master is used to align the CYCLE_TIME registers of any isochronous-capable nodes on an IEEE1394 bus. It is transmitted at or after cycle counter on the cycle master node is incremented. It carries the value of the cycle master node CYCLE_TIME register at the time the cycle start is initiated.

Asynchronous activities on the bus at the time of the cycle start event causes a delay in transmitting the cycle start packet. At the other isochronous nodes, the CYCLE_TIME register is loaded with the value carried on the cycle start packet. That compensates for the cycle start delay but only up to the resolution of that register. This resolution is 1/24.576MHz (which is approximated in this annex as 41ns).

The cycle start packet carries a value from the CYCLE_TIME register. If the transmission of the packet is timed so that it always occurs at a fixed time (after the moment that the CYCLE_TIME register updates to that value), then cycle start delays will be corrected without significant error. This means that asynchronous activity at the time of the cycle start event will not be a source of jitter.

However some IEEE1394 compliant implementations might introduce a variable delay between the time the CYCLE_TIME register is updated and cycle start packet transmission of that value. This will depend on the implementation but this delay may be limited to less than the 41ns CYCLE_TIME resolution or it could possibly be even greater than this.

C.1.2.1.2 Variable transport delay to cycle start packets

As a cycle start packet is passed through intermediate nodes on the bus it is delayed by a variable amount of repeater data delay.

The normal mechanism for the variation in this delay is the re-timing of the packet by the local clock at each node. The repeater data delay varies as the relative timing of the incoming transitions and the local clock changes. This change is a result of the frequency difference between the local clock and the clock on the previous node the packet has passed through. Jitter produced in this way is in the form of a ramping variation with a step correction in the opposite direction. The frequency of this 'sawtooth' is related to the frequency difference between the two node clocks.

IEEE1394 does not define explicit limits for repeater delay jitter. The draft supplement, P1394a, specifies a PHY register field 'Jitter' that can indicate values from 1/49.152MHz (which is approximated in this annex as 20ns) to 7/49.152MHz (approximately 163ns).

IEEE1394 PHY devices that resynchronize received data with a 49.152MHz clock will have repeater data delay jitter approximately 20ns peak-peak or 6ns RMS.

The jitter due to variable repeater delay jitter is cumulative. The total variable transport delay is the sum of the delay at each node. The total RMS jitter to the cycle start packet transport delay is the root sum of squares (RSS) of the RMS jitter at each intermediate repeater node.

C.1.2.1.3 Quantization of CYCLE_TIME register correction

The CYCLE_TIME registers at each isochronous node increment at a rate defined by the exact rate of the 24.576MHz clock in the local node. These registers are time aligned with similar registers in other nodes by being loaded with the value carried in the cycle start packet transmitted by the cycle master. As the CYCLE_TIME register incrementing clock has a slightly different frequency at each node there will be a gradually changing error between the updating of that register at the cycle master and the other nodes.

When there is a difference between the value in an incoming cycle start packet and the value in the local CYCLE_TIME registers, then a correction is made.

This correction is quantized to the CYCLE_TIME register resolution of 1/24.576MHz. The contribution of this mechanism to the CYCLE_TIME register jitter is normally a gradually increasing delay or advance with corrective step in the opposite direction. This jitter has an amplitude equivalent to the CYCLE_TIME resolution of 41ns peak to peak and 12ns RMS.

C.1.2.2 Time-stamp quantization jitter

The time stamp (SYT) carrying the sampling timing information has a resolution of 1/24.576MHz. The effect of quantization to this resolution is to add jitter to the embedded sample clock. This jitter has an amplitude equivalent to the SYT resolution of 41ns peak to peak and 12ns RMS. It will have frequency components related to the beat frequency between the time stamp rate (Fs/SYT_INTERVAL) and the 24.576MHz clock incrementing the CYCLE_TIME register.

C.1.3 Embedded sample clock jitter

C.1.3.1 Embedded sample clock jitter spectrum

The error in the values and timing of the embedded synchronization clock can be considered as a time-varying signal. This can be examined in the frequency domain through spectrum analysis. This jitter spectrum will relate to the jitter spectrum in the sample clock transfer mechanism and the jitter transfer function.

There are discrete frequency components corresponding to the fundamental and harmonic frequencies associated with each of the applicable jitter sources described in the previous clause. These frequencies depend on the frequency differences between the local PHY clocks on the nodes.

Any jitter source that produces a jitter signal similar to a sawtooth will have discrete jitter frequency components at the sawtooth frequency and multiples of that rate. Where the multiple is at a frequency above half the frequency that the timing information is updated then that component will be aliased to below that rate and the signal will no longer appear as a sawtooth.

C.1.3.2 Embedded sample clock jitter amplitude

The total amount of embedded sample clock jitter is dependent on the following:

— The number of nodes between the cycle master and sample clock source.

— The number of nodes between the cycle master and sample clock destination.

— The implementation of each node.

— Whether or not the sample clock source is synchronized to the bus.

C.1.3.2.1 Example One: Simple two-node bus

As an example examine the simplest two-node system. This has the cycle master as the sample clock source node (node 0), and the sample clock is locked to the sample clock source node PHY clock at a multiple of the cycle time rate. Asynchronous activity is low enough to ensure that the cycle start packet is never delayed.

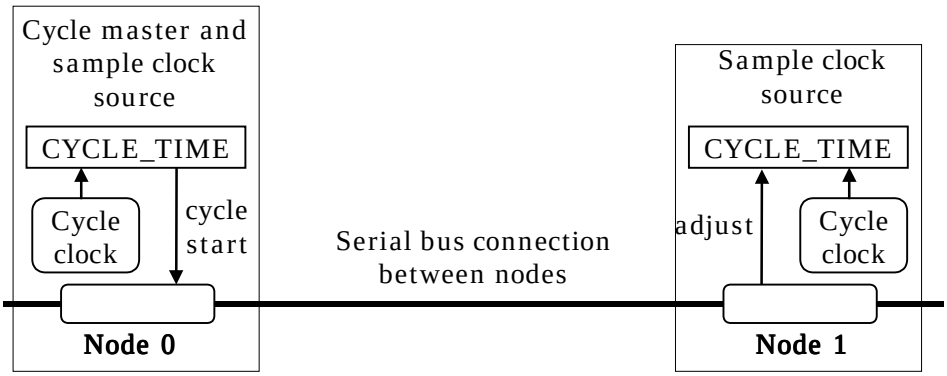

Figure C. 1 – Two-node bus

— There will not be any jitter due to cycle start packet CYCLE_TIME resolution as the cycle start packet is not being delayed due to asynchronous activity.

— There is no variable transport delay to cycle start packets as there are no intermediate nodes on the bus.

— Quantization of CYCLE_TIME register correction in the sample clock destination node will be a source of jitter in this example. This will be in the form of one sawtooth at a frequency determined by the offset between the cycle start rate and the sample clock destination PHY clock. This will have an amplitude of approximately 12ns RMS (41ns peak to peak).

— As the sample clock is frequency-locked to the cycle master PHY clock there is no time-stamp quantization jitter.

Therefore for the simple two-node system in this example the recovered embedded sample clock will have just one systematic jitter source. This will have a jitter amplitude of approximately 12ns RMS (41ns peak to peak) in the form of a sawtooth at a rate determined by the frequency offset between the two PHY node clocks.

C.1.3.2.2 Example two: three-node bus

For this example there are three nodes which are separately the cycle master node, sample clock source node and sample clock destination node.

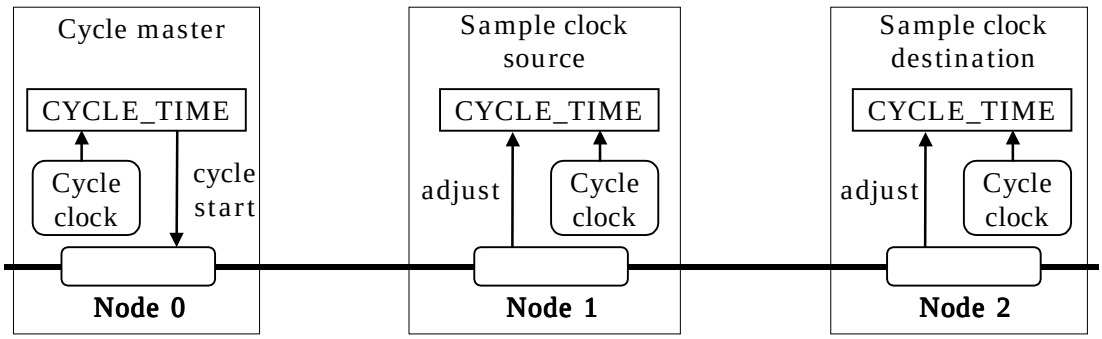

Figure C. 2 – Three-node bus

The following analysis also assumes that the sample clock is not synchronous to any of the bus clocks

— If the cycle start packet is sometimes delayed there may be some jitter caused when the cycle start packet CYCLE_TIME value does not exactly correspond with the delay to the transmission of the packet. This will have a peak amplitude that is dependent on the implementation of the cycle master cycle start transmission mechanism. (The amplitude of this mechanism is not included in the analysis.)

— In the path from cycle master (node 0) to node 1 there are no intermediate nodes. In the path from cycle master (node 0) to node 2 there is one intermediate node that will have a variable transport delay to cycle start packets. This will contribute to the jitter in the CYCLE_TIME value at that node. This jitter will be in the form of a sawtooth related to the beating of the node 0 and node 1 cycle clocks. The amplitude of this jitter mechanism depends on the implementation of the repeater function in this node. This analysis assumes that this repeater includes resynchronization with a 49.152MHz clock. This will contribute jitter of approximately 6ns RMS (20 ns peak to peak).

— Quantization of CYCLE_TIME register correction in nodes 1 and 2 will be a source of jitter. In each of these nodes this will be in the form of a sawtooth at a frequency determined by the offset between the cycle start rate and the node PHY clock. These two sources of jitter will each have an amplitude of approximately 12ns RMS (41ns peak to peak).

— At node 1 the sample clock timing is encoded into the SYT with the resolution of the CYCLE_TIME register. The sample clock is asynchronous to the update of the CYCLE_TIME register. The error due to the variation in relative phase of the clocks is a sawtooth with a frequency determined by the difference between the node 1 cycle clock frequency and the time stamp rate. This source of jitter will have an amplitude of approximately 12ns RMS (41ns peak to peak).

This illustrates how this system has four sources of periodic jitter (excluding the source of jitter related to asynchronous activity): Three of 12ns RMS and one of 6ns RMS. The sum total of the periodic jitter (excluding the component due to asynchronous activity) will be 21 ns RMS. (This would also have a peak to peak value of 132 ns. This value represents the infrequent coincidence of the peaks of all the contributing jitter components and would be an infrequent occurrence.)

C.1.3.2.3 Example: thirty-five-node system

This example illustrates a large bus configuration with 23 hops between the cycle master (node 0) and each of, sample clock source (node 23) and sample clock destination (node 34). (According to draft IEEE P1394a this configuration represents a maximum within the constraints of a maximum PHY delay of 144ns and maximum cable length of 4.5m.)

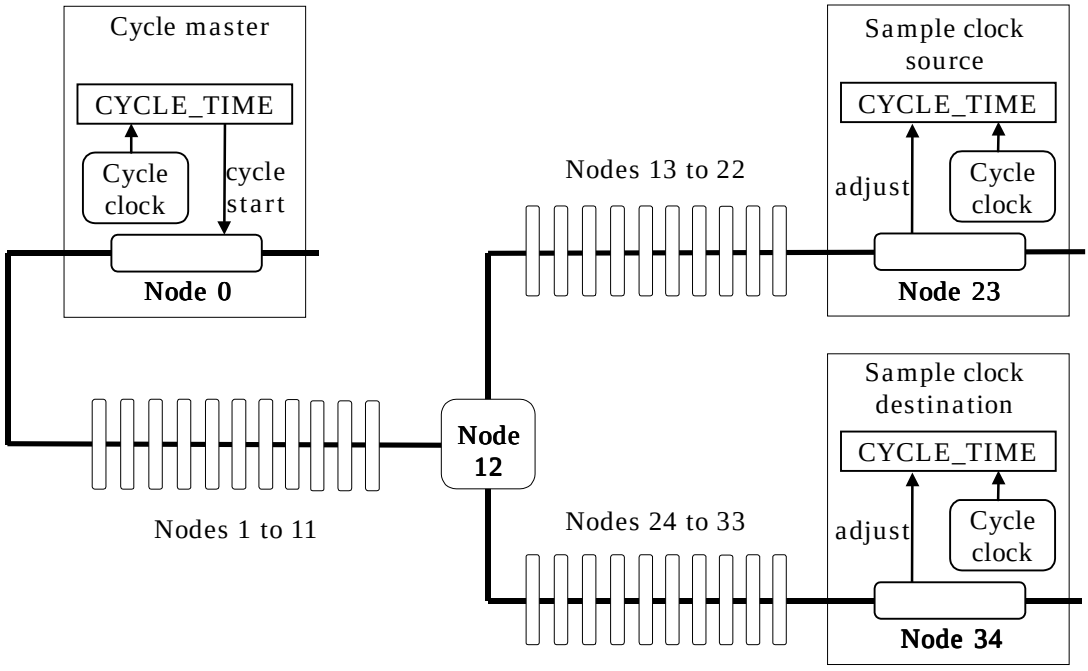

Figure C. 3 – Thirty-five-node bus

The following analysis also makes similar assumptions as for the 3-node example with respect to the sample clock.

- If there is asynchronous activity on the bus then the jitter mechanism due to cycle start packet delay is the same as for the three-node example. This is not included in the analysis.
- In the paths from the cycle master (node 0) to both the sample clock source (node 23) and sample clock destination (node 34) there are 22 intermediate nodes. Each of these will impose a variable transport delay on to cycle start packets in the same manner as the 3-node example. The peak jitter will scale in proportion to the number of hops (22) and the RMS jitter will scale with the square root of that number, 4.7. If each repeater applies re-synchronization with a local 49.152MHz clock then they will add a total of 28ns RMS of jitter to the arrival time of the cycle start packet at the sample clock source (node 23) and at the sample clock destination (node 34).
- As with the 3-node example, quantization of CYCLE_TIME register correction at the sample clock source and destination will be a source of jitter of amplitude 12ns RMS each.
- As with the 3-node example, the time-stamp quantization jitter will add 12ns RMS.

This illustrates how this system has three sources of periodic sawtooth jitter at 12ns RMS and two summed periodic components at 28ns RMS each. The sum total of the periodic jitter is 44 ns RMS.

This result does not represent a 'worst case'. The variable transport delay jitter at each intermediate node could be significantly greater than 20ns while remaining compliant with IEEE1394. The potential variable error in the CYCLE_TIME value in the cycle start packet (when the cycle start has been delayed by asynchronous activity) has also not been included.

C.1.4 Jitter attenuation

This occurs with the filtering function of the sample clock recovery device. This will have a low pass jitter attenuation characteristic. Sample clock jitter causes modulation of the sampled signal. These modulation products may become audible. For high quality applications it is recommended that the jitter attenuation characteristic of the sample clock recovery system satisfies the template shown in Figure C.4.

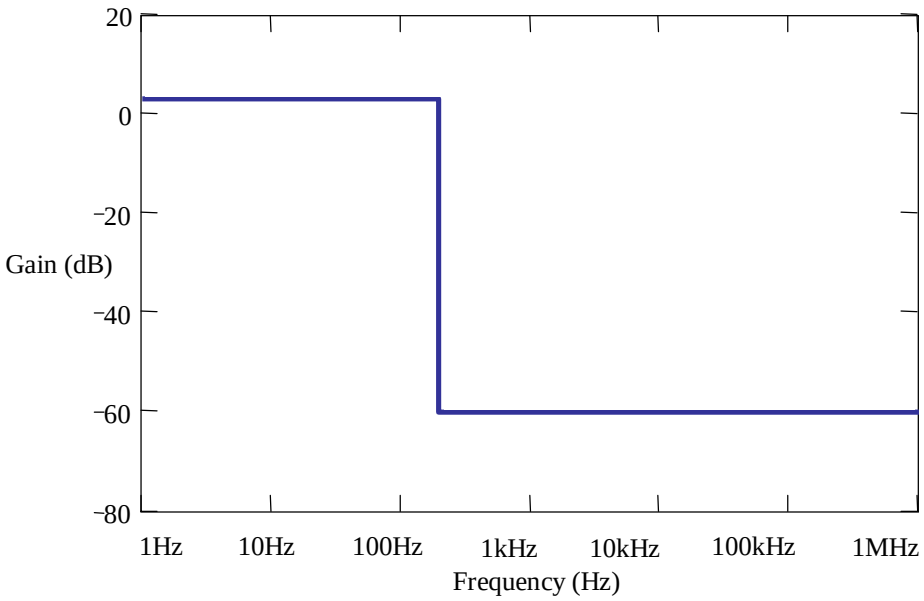

Figure C. 4 – Sample clock recovery jitter attenuation template

To satisfy this template the jitter attenuation plotted against jitter frequency shall fall below the shaded clauses of the graph. The attenuation shall exceed 60dB at jitter frequencies above 200Hz and up to half the recovered sample clock frequency. Below 200Hz the gain shall not exceed 3dB.

The jitter attenuation for received jitter at frequencies, f_r above half the SYT_MATCH clock rate, f_s is determined by the response to the images of the received jitter that may be present in the sampling clock. These will be present at image frequencies of:

$$f_i = N \cdot f_s \pm f_r$$

Where N is an integer.

C.1.5 Jitter measurement

Jitter meters approximate the long-term average frequency and phase of a signal that they are measuring. This will result in a high-pass characteristic. As the sample clocks derived using the A/M protocol have a strong low frequency jitter component the low frequency corner frequency of the jitter meter is important.

It is recommended that jitter measurements use the characteristics defined by the jitter measurement filter characteristic of Figure C.5.

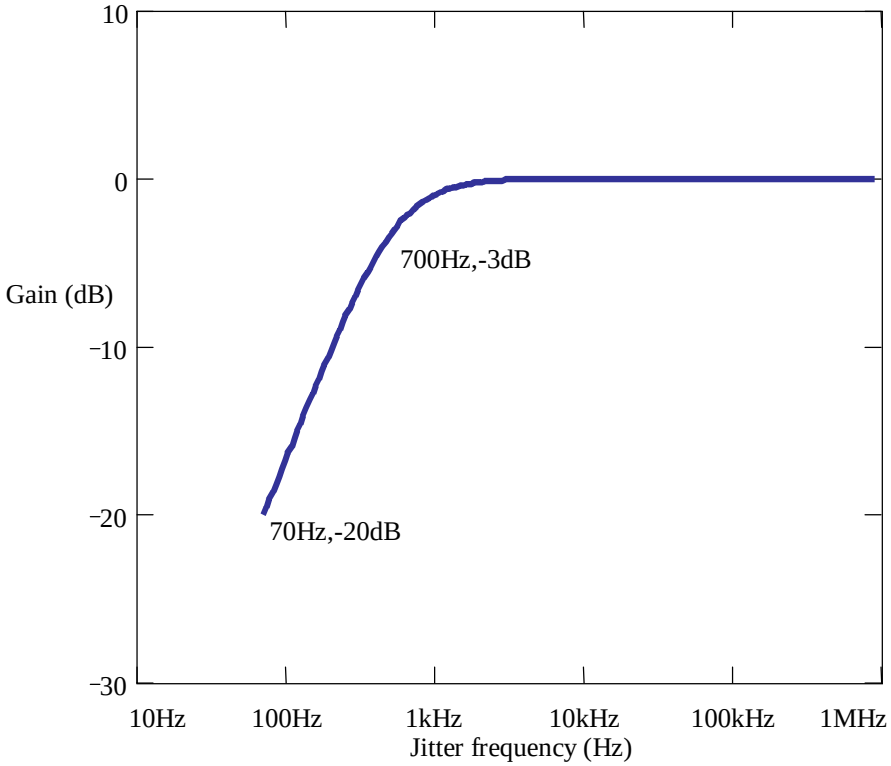

Figure C. 5 – Sample clock jitter measurement filter characteristic

This is a minimum- phase high pass filter with a -3 dB frequency of 700 Hz, a first order roll-off to 70 Hz and with a pass-band gain of unity. Note: This is compatible with the intrinsic jitter measurement filter characteristic used in IEC60958-3 and -4.