JEDEC STANDARD

Serial Interface for Data Converters

JESD204B

(Revision of JESD204A, April 2008)

JULY 2011

JEDEC SOLID STATE TECHNOLOGY ASSOCIATION

NOTICE

JEDEC standards and publications contain material that has been prepared, reviewed, and approved through the JEDEC Board of Directors level and subsequently reviewed and approved by the JEDEC legal counsel.

JEDEC standards and publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for use by those other than JEDEC members, whether the standard is to be used either domestically or internationally.

JEDEC standards and publications are adopted without regard to whether or not their adoption may involve patents or articles, materials, or processes. By such action JEDEC does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the JEDEC standards or publications.

The information included in JEDEC standards and publications represents a sound approach to product specification and application, principally from the solid state device manufacturer viewpoint. Within the JEDEC organization there are procedures whereby a JEDEC standard or publication may be further processed and ultimately become an ANSI standard.

No claims to be in conformance with this standard may be made unless all requirements stated in the standard are met.

Inquiries, comments, and suggestions relative to the content of this JEDEC standard or publication should be addressed to JEDEC at the address below, or refer to www.jedec.org under Standards and Documents for alternative contact information.

Published by
©JEDEC Solid State Technology Association 2011
3103 North 10th Street
Suite 240 South
Arlington, VA 22201-2107

PRICE: Contact JEDEC

Printed in the U.S.A.

PLEASE!

DON'T VIOLATE
THE
LAW!

SERIAL INTERFACE FOR DATA CONVERTERS

(From JEDEC Board Ballot JCB-08-01 and JCB-11-47, formulated under the cognizance of JC-16 Committee on Interface Technology.)

1 Scope

This specification describes a serialized interface between data converters and logic devices. It contains normative information to enable designers to implement devices that communicate with other devices covered by this specification. Informative annexes are included to clarify and exemplify the specification.

Due to the range of applications involved, the intention of the document is to completely specify only the serial data interface and the link protocol. Certain signals common to both the interface and the function of the device, such as device clocks and control interfaces, have application-dependent requirements. Devices may also have application-dependent modes, such as a low power / shutdown mode that will affect the interface. In these instances, the specification merely constrains other device properties as they relate to the interface, and leaves the specific implementation up to the designer.

Revision A of the standard was expanded to support serial data interfaces consisting of single or multiple lanes per converter device. In addition, converter functionality (ADC or DAC) can be distributed over multiple devices:

- All parallel running devices are implemented or specified to run synchronously with each other using the same data format.
- Normally this means that they are part of the same product family.

Revision B of the standard now supports the following additional functions:

- Mechanism for achieving repeatable, programmable deterministic delay across the JESD204 link.
- Support for serial data rates up to 12.5 Gbps.
- Transition from using frame clock as the main clock source to using device clock as the main clock source. Device clock frequency requirements offer much more flexibility compared to requiring a frame clock input.

The logic device (e.g. ASIC or FPGA) is always assumed to be a single device.
Figure 1 compares the scope of the original JESD204 specification and its revisions.

1 Scope (cont'd)

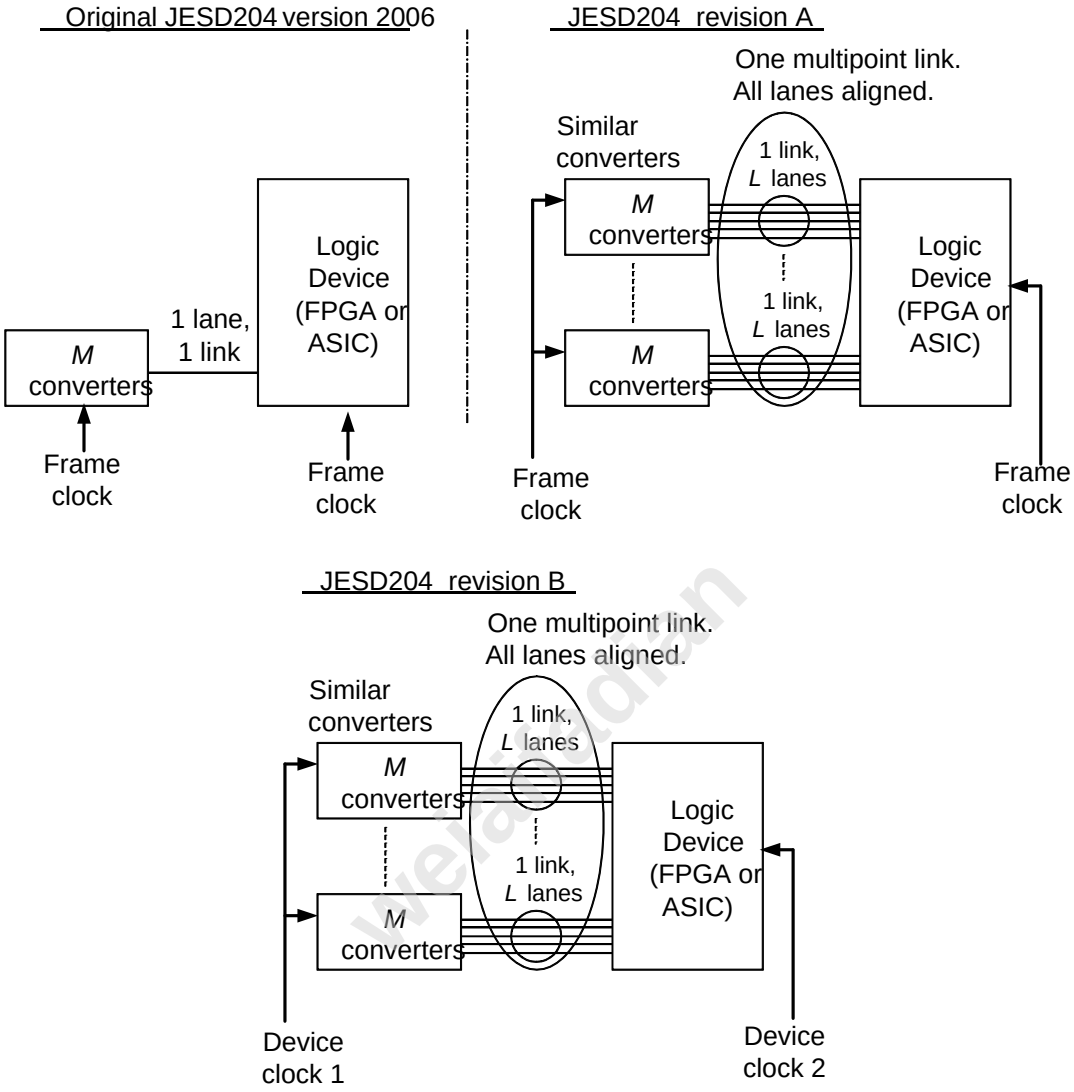


Figure 1 — Scope of original JESD204 and revisions A and B

Although not illustrated in the figure, it is possible to apply multiple, independent instances of the JESD204 standard to the same device.

2 References

2.1 Normative

The following normative documents contain provisions that, through reference in this text, constitute provisions of this standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies.

1. IEEE Std 802.3-2008®, Part 3, Section Three, Local and metropolitan area networks - CSMA/CD access methods and Physical Layer specifications, 2008. http://standards.ieee.org/getieee802/

2. JEDEC JESD99, Terms, Definitions, and Letter Symbols for Microelectronic Devices.

3. OIF-SxI-5-01.0, System Interface Level 5 (SxI-5): Common Electrical Characteristics for 2.488 – 3.125Gbps Parallel Interfaces, Optical Internetworking Forum, October 2002. www.oiforum.com/public/documents/OIF-SxI5-01.0.pdf

4. OIF-CEI-02.0, Common Electrical I/O- Electrical and Jitter Interoperability agreements for 6G+ bps and 11G+ bps I/O, Optical Internetworking Forum, February 2005. www.oiforum.com/public/documents/OIF_CEI_02.0.pdf

2.2 Informative

The following standards contain provisions that, through references in the text, are informative in this standard.

5. ANSI T1.523-2001, ATIS Telecom Glossary 2000, February 2001. http://www.atis.org/tg2k/

6. IEEE Std 802.3-2008®, Part 3, Section Four, Local and metropolitan area networks - CSMA/CD access methods and Physical Layer specifications, 2008. http://standards.ieee.org/getieee802/

7. ANSI/IEEE Std 91a-1991, Graphic symbols for logic functions, IEEE 1991, ANSI 1994. (Summary available at e.g. http://en.wikipedia.org/wiki/Logic_gate)

8. INCITS 450-2009, Information technology - Fibre Channel - Physical Interface - 4 (FC-PI-4), available from http://webstore.ansi.org

9. INCITS TR-35-2004 (R2009), Fibre Channel - Methodologies for Jitter and Signal Quality Specification (FC-MJSQ) , available from http://webstore.ansi.org

3 Terminology

For the purposes of this standard, the terms and definitions given in JESD99 (reference 2) and the following apply:

3.1 Terms and definitions

8B/10B code: A DC-balanced octet-oriented data encoding specified in reference 1, clause 36.2.4. (Ref. IEEE 802.3)

ceil(x): The smallest integer greater than or equal to x.

character: A symbol produced by 8B/10B encoding of an octet.

NOTE 1 While all octets can be encoded as data characters, certain octets can also be encoded as control characters.

NOTE 2 The same character may exist as two different code groups, depending on running disparity.

character clock: A signal used for sequencing the 8B/10B characters or octets.

clock generator: A circuit used to generate synchronous, phase aligned device clocks to various devices in the JESD204B system.

NOTE A clock generator circuit can include one or more clock generator devices, but they must use a common source clock.

code group: A set of ten bits that, when representing data, conveys an octet. (Ref. IEEE 802.3)

control interface: An application-specific interface used to pass information (usually status and control information) between a converter device and a logic device and/or between a device and a higher layer application level.

NOTE The details of the control interface are outside the scope of the serial interface described by this standard.

conversion clock: A signal used to define the analog sampling moments in a converter.

NOTE Usually the conversion clock is the same as the sample clock, except in case of interpolating DACs or decimating ADCs, where the conversion clock is faster than the sample clock. In all cases, the conversion clock is derived from the device clock.

converter: An analog-to-digital converter (ADC) or digital-to-analog converter (DAC).

NOTE In this standard, a converter is assumed to interface via a single stream of digital samples.

converter device: A component package containing one or more converters.

NOTE This standard specifies the interactions between one logic device and one or more converter devices.

3.1 Terms and definitions (cont'd)

data link: An assembly, consisting of parts of two devices and the interconnecting data circuit, that is controlled by a link protocol enabling data to be transferred from a data source to a data sink. ("terminal" replaced by "device" in ANSI T1.523-2001.)

descrambler: The inverse of a scrambler. (Ref. ANSI T1.523-2001)

NOTE The descrambler output is a signal restored to the state that it had when it entered the associated scrambler, provided that no errors have occurred.

device clock: A master clock signal from which a device must generate its local clocks.

floor(x): The greatest integer less than or equal to x.

frame: A set of consecutive octets in which the position of each octet can be identified by reference to a frame alignment signal. (Adapted from ANSI T1.523-2001.)

NOTE 1 The frame alignment signal does not necessarily occur in each frame.

NOTE 2 In JESD204, a frame consists of *F* octets and is transmitted over a single lane.

frame clock: A signal used for sequencing frames or monitoring their alignment.

frame period: One period of the frame clock, i.e. the duration of one frame.

NOTE During one frame period, one frame is transmitted over each lane of a multilane link.

idle mode: An operating mode used for a converter that is not currently sampling data.

interconnect: The transmission path along which a signal propagates. (Synonym for "medium" in ANSI T1.523-2001.)

invalid code group: A code group that is not found in the proper column of the 8B/10B decoding tables, according to the current running disparity. (Ref. IEEE 802.3)

lane: A differential signal pair for data transmission in one direction.

line clock: A signal used for sequencing the serial bits on an electrical interface.

link: Synonym for "data link".

local clock: A clock derived inside a device from the device clock and used in the implementation of the JESD204B link within the device.

NOTE 1 It is possible to align a local clock to an external signal, e.g. SYSREF.

NOTE 2 An internal copy of the device clock is not a local clock.

3.1 Terms and definitions (cont'd)

logic device: A component package containing exclusively or primarily digital logic; e.g., an ASIC or FPGA.

NOTE This standard specifies the interactions between one logic device and one or more converter devices.

max(x, y): The largest of x and y.

min(x, y): The smallest of x and y.

mod(x, y): The remainder after dividing x by y (x modulo y).

multiframe: A set of consecutive frames in which the position of each frame can be identified by reference to a multiframe alignment signal. (Ref. ANSI T1.523-2001)

NOTE 1 The multiframe alignment signal does not necessarily occur in each multiframe.

NOTE 2 In JESD204, a multiframe consists of *K* frames and is transmitted over a single lane.

multiframe clock: A signal used for sequencing multiframes or monitoring their alignment.

multipoint link: A data communications link that interconnects three or more devices. ("terminal" replaced by "device" in ANSI T1.523-2001.)

nibble: A group of four data bits. (Ref. IEEE 802.3)

octet: A group of eight adjacent binary digits, serving as the input to an 8B/10B encoder or the output of an 8B/10B decoder.

receiver: A circuit attached to a lane for reconstructing a serial bit stream into time-aligned frames.

NOTE A receiver consists of one physical layer block and one link layer block.

receiver block: The combination of the receiver transport layer and all receiver link layer and physical layer blocks connected to a link.

receiver device: A component package containing one or more receiver blocks.

rising edge of a differential signal(P,N): The simultaneous transitions that occur when signal(P) changes from the low logic level to the high logic level and signal(N) changes from the high logic level to the low logic level.

running disparity: A binary parameter having a value of + or –, representing the imbalance between the number of ones and zeros in a sequence of 8B/10B code-groups. (Ref. IEEE 802.3)

sample: The instantaneous value of a signal measured or determined at a discrete time. (Adapted from ANSI T1.523-2001, "sampled data".)

NOTE In the context of JESD204, a sample is always the digital representation of a signal.

3.1 Terms and definitions (cont'd)

sample clock: A signal used to define the sample boundaries within a frame.

NOTE Usually the sample clock is the same as the frame clock, except in cases where there are multiple samples per converter within a frame, where the sample clock is an integer multiple of the frame clock. In all cases, the sample clock is derived from the device clock.

scrambler: A randomizing mechanism that is used to eliminate long strings of consecutive identical transmitted symbols and avoid the presence of spectral lines in the signal spectrum without changing the signaling rate. (Ref. IEEE 802.3)

source clock: An oscillator from which the various other clock signals are derived. This oscillator is typically a VCO inside a clock generator device, or an external VCO within a clock generator circuit.

symbol: The smallest unit of coded data on the medium. (Simplified from IEEE 802.3.)

NOTE In this standard used as synonym to "character".

SYSREF: A periodic, one-shot (strobe-type), or "gapped" periodic signal used to align the boundaries of local clocks in JESD204B Subclass 1 devices. SYSREF must be source synchronous with the device clock.

transmitter: A circuit that serializes the input frames and transports the resulting bit stream across a lane.

NOTE A transmitter consists of one link layer block and one physical layer block.

transmitter block: The combination of the transmitter transport layer and all transmitter link layer and physical layer blocks connected to a link.

transmitter device: A component package containing one or more transmitter blocks.

valid code group: A code group that is found in the proper column of the 8B/10B decoding tables, according to the current running disparity. (Ref. IEEE 802.3)

word: A character string or a binary element string that it is convenient to consider as an entity. (Adapted from ANSI X3.172.)

3.2 Meaning of symbols and abbreviations

/A/: K28.3 character, lane align.

ADC: Analog-to-Digital Converter

C: Control bit (used in figures).

***CF*:** Number of control words per frame clock period per link.

Cr: Converter (used in figures).

CGS: Code Group Synchronization, see 5.3.3.1.

***CS*:** Number of control bits per conversion sample.

D: Dx.y, any 8B10B data symbol (used in figures).

DAC: Digital-to-Analog Converter

/Dx.y/: Data code group Dx.y encoded according to the current running disparity. (Ref. IEEE 802.3)

/F/ (as abbreviation): K28.7 character, frame sync.

***F* (as symbol):** Number of octets per frame.

***HD*:** High Density user data format, see 5.1.3.

ILA: Initial Lane Alignment, see 5.3.3.5.

/K/ (as abbreviation): K28.5 character, code group synchronization.

***K* (as symbol):** Number of frames per multiframe.

/Kx.y/: Control code group Kx.y encoded according to the current running disparity. (Ref. IEEE 802.3)

***L*:** Number of lanes per converter device.

L_{MP}**:** Total number of lanes per multipoint link.

LMFC: Local Multi Frame Clock.

***M*:** Number of converters per device.

MCDA: Multiple Converter Device Alignment (see clause 9)

ML: Multiple Lane (see clause 9)

3.2 Meaning of symbols and abbreviations (cont'd)

mod: Modulo. x mod y is used as alternative notation for mod(x, y).

***N*:** Converter resolution.

N': Total number of bits per sample in the user data format.

NG: Nibble Group

NMCDA: No Multiple Converter Device Alignment (see clause 9).

PVT: A term denoting the variations of a device across process corner, supply voltage, and temperature.

/R/: K28.0 character.

RBD : RX Buffer Delay, see 6.1.

RD: Running Disparity

RX: Receiver

***S*:** Number of samples transmitted per single converter per frame cycle.

SERDES: Serializer/Deserializer

SL: Single Lane (see clause 9).

T: Tail bit

***T_f*:** Frame Period

TX: Transmitter

UI: Unit Interval = the duration of a serial bit.

Table 1 — Special symbols and operators

Printed character	Meaning	Usage
|	Logical OR operator	In program code
!	Logical NOT operator	In program code
!=	Relational inequality operator	In program code
&	Logical AND operator	In program code
=	Assignment operator	In program code
==	Relational equality operator	In program code
~	Active low signal	Suffix to a name

4 Electrical specification

4.1 Electrical specification overview

The electrical layer of this specification supports three physical layer variants for unidirectional, point-to-point, serial coded data rates:

- LV-OIF-SxI5 – based operation: from 312.5 Mbps to 3.125 Gbps (adapted from OIF SxI-5, Ref. 3).
- LV-OIF-6G-SR – based operation: from 312.5 Mbps to 6.375 Gbps (adapted from CEI-6G-SR, Ref. 4).
- LV-OIF-11G-SR – based operation: from 312.5 Mbps to 12.5 Gbps (adapted from CEI-11G-SR, Ref. 4).

The converter devices are connected to a logic device in one of three ways:

- All devices are on the same printed circuit board and communicate at up to 3.125, 6.375, or 12.5 Gbps (depending on the physical layer variant in use) using copper signal traces
- The devices are on separate cards or boards that are connected by a backplane using one or more controlled-impedance connectors,
- The devices are on separate cards or boards that are connected by one or more cables.

When the converter devices and the logic device communicate across a controlled-impedance connector or a cable, the implementer must ensure the reliability of the link at the implementer-determined maximum signaling rate (up to 3.125, 6.375, or 12.5 Gbps depending on the physical layer variant in use).

The interconnect is point-to-point and unidirectional. The signaling defined in this specification is low-swing, differential, suitable for low-voltage IC technologies, and generally represents what is commercially referred to as "CML".

Compliant Transmitters and receivers are expected to achieve a BER less than 1e-12 for the LV-OIF-SxI5 physical layer, and less than 1e-15 for the LV-OIF-6G-SR and LV-OIF-11G-SR physical layers.

Compliance of the LV-OIF-6G-SR and LV-OIF-11G-SR variants may be verified at a BER of 1e-12 by applying adjustments to the Gaussian Jitter (GJ) portion of the transmit and receive masks, and to the GJ portion of the reference transmitter test signal to compensate for the lower test populations (consult OIF-CEI-02.0, Ref. 4, Clause 2).

4.1 Electrical specification overview (cont'd)

Voltage specifications in this document follow the convention shown in Figure 2.

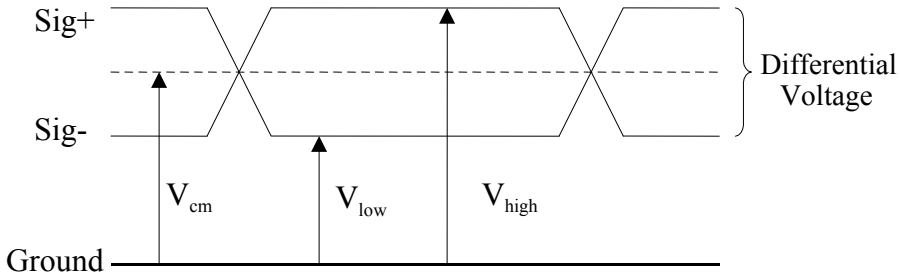


Figure 2 — Differential peak-to-peak Voltage = 2 * (V_{high} – V_{low})

4.2 Compliance types

Compliance is measured at the system-level integration of Transmitter, Receiver and Interconnect and is a characteristic of the Transmitter and Receiver devices' external behavior.

For Transmitter and Receiver devices, there are two *types* of compliance - one of which may be claimed by device manufacturers:

- DC-compliance: DC parameters of the device conform to the specifications in 4.4, 4.5, or 4.6 (depending on the physical layer variant in use).
- AC-compliance only: DC parameters of the device do not fully conform to the specifications in 4.4, 4.5, or 4.6 (depending on the physical layer variant in use).

If either the Transmitter or Receiver device or both are not DC-compliant, the devices must be coupled via external capacitors. External coupling capacitors may also be used with DC-compliant devices unless the device specification specifically prohibits it. Note that a device may internally use AC-coupling in order to meet the DC electrical specifications and thus claim DC-compliance at its pins.

The system integrator must ensure that the Transmitter and Receiver compliance types match and that the Interconnect is appropriate for the compliance type chosen.

For the Transmitter and receiver circuits, and for the system, there is only one *level* of compliance recognized for each of the three defined physical layer variants- i.e., full compliance with performance specifications detailed in this clause for the particular physical layer variant being used. There is only one exception to this: a device may specify a range of data transfer rates that is a subset of the full range of data rates supported by one of the physical layer variants. A compliant device does not have to support the full data rate range.

4.3 Interconnect

The data interface supports unidirectional, point-to-point, serial coded data rates from 312.5 Mbps to 3.125, 6.375, or 12.5 Gbps (depending on the physical layer variant in use) between converter devices and a logic device using controlled impedance traces on printed circuit boards (PCBs). The data interface may also be implemented across a backplane (using controlled-impedance connectors) or across a cable. Exactly one Transmitter and one Receiver are allowed to be present at the ends of the interconnect (point-to-point unidirectional connection).

The primary intended application is as a point-to-point interface of up to approximately 200mm (≈8") between integrated circuits. Up to one connector is expected for operation above 3.125 Gbps and up to two connectors for operation below 3.125 Gbps. The performance of an actual transceiver interconnect is highly dependent on the implementation. The link performance depends on effective channel characteristics like attenuation rather than on physical length. For operation below 6.375 Gbps a 200 mm reach is expected to be possible with a low-cost material, but above 6.375 Gbps a premium material will likely be necessary to achieve the same reach. (See Ref. 3 and Ref. 4.)

The Interconnect requirements are as follows:

- The nominal characteristic impedance of the signal traces shall be 100 Ω differential.
- Depending upon the Transmitter's and Receiver's compliance capabilities, AC-coupling capacitors may be used in the interconnect. The value of the capacitor depends on the minimum frequency present in the code used. Therefore the value of the AC-coupling capacitors is not specified in this document and is determined by the application.

4.3.1 Interconnect Insertion Loss

The interconnect insertion loss is a measure of the signal quality of the interconnect itself. Insertion loss will be affected by channel length, dielectric material, number of connectors, and board topology.

- The fitted interconnect insertion loss (when calculated using the linear method) shall meet the mask requirements of the transfer model shown in Figure 3 from 50MHz to 0.75 times the Baud Rate. For details on calculating the linear fitted insertion loss of the channel, please refer to Appendix E.
- The Insertion Loss Deviation (calculated as the absolute difference between the fitted insertion loss and the measured insertion loss) shall not exceed 1.5dB from 50MHz to 0.75 times the baud rate.

4.3.1 Interconnect Insertion Loss (cont'd)

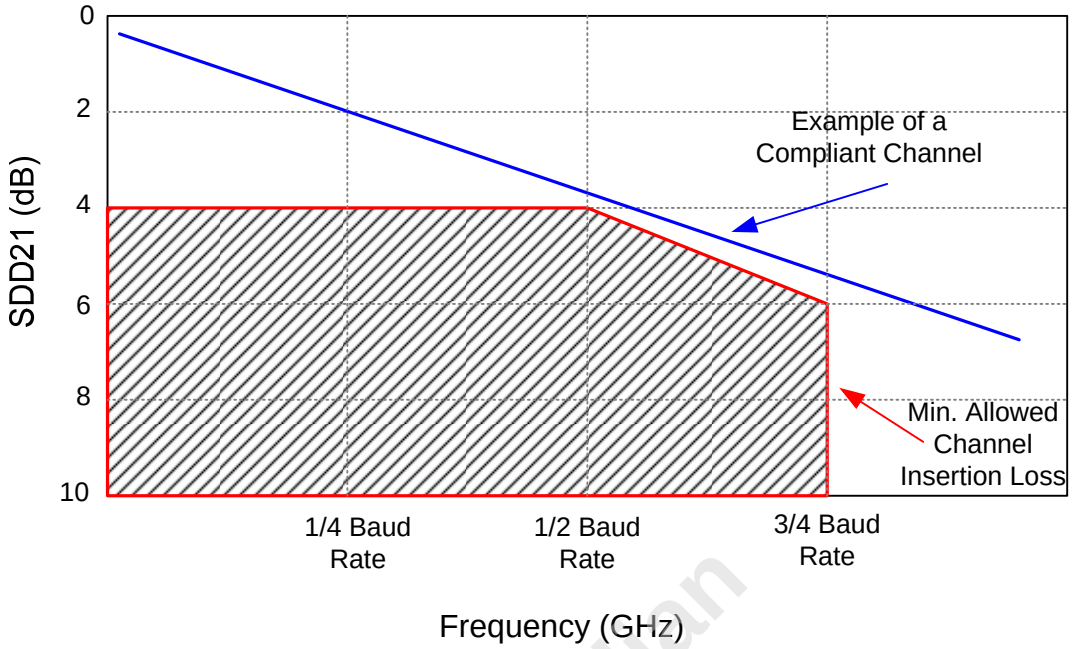


Figure 3 — Interconnect Insertion Loss Mask

4.4 LV-OIF-SxI5 Data interface signals (Differential)

4.4.1 Compliance verification

The device implementer shall consult OIF SxI-5 (Ref. 3) regarding the exact interpretation of the items specified in 4.4.2 and 4.4.3 and OIF-CEI-02.0 (Ref. 4) regarding their measurement techniques. However, the PRBS31 sequence is not used for compliance testing in JESD204. Instead of PRBS31, JESD204 transmitters shall be compliance tested with either the "modified RPAT" or the JSPAT pattern and JESD204 receivers with the JTSPAT pattern, see 5.3.3.8.2. INCITS TR-35-2004 (Ref. 9) is recommended as a useful resource for signal quality and jitter tolerance measurement methodologies.

4.4.2 LV-OIF-SxI5 Transmitter Electrical Specifications

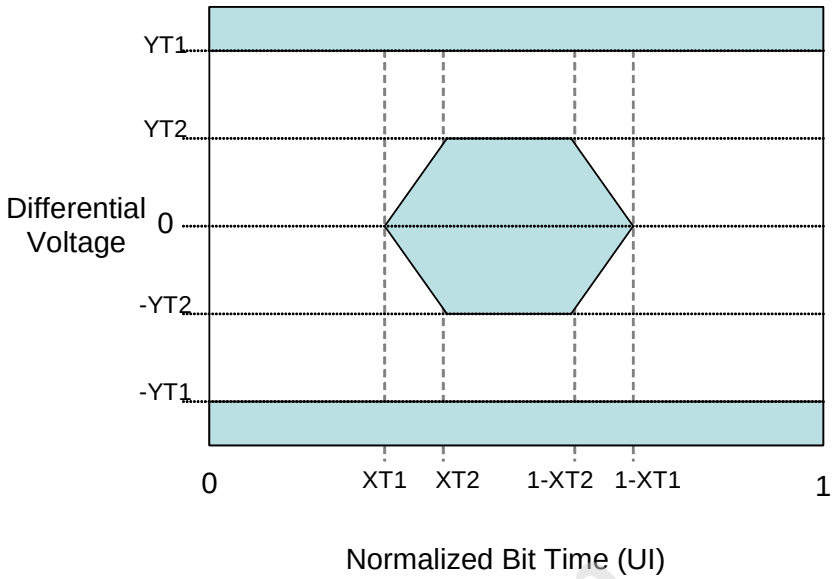
LV-OIF-SxI5 Compliant transmitters shall meet all the requirements specified in 4.4.2.

Table 2 — General differential output DC and AC characteristics for LV-OIF-SxI5 – based operation

Symbol	Parameter	Conditions	Min	Max	Units
UI	Unit Interval	(Note 1)	320	3,200	ps
Trise/Tfall	Rise and Fall Times	20% - 80% into 100 Ω load	50	(Note 2)	ps
Vdcm	Transmitter Common Mode Voltage	Required only if DC-Compliance is claimed. Termination circuit of Figure 5 applied at the Transmitter terminals and Transmitter Ground with 1.05 V < Vtt < 1.35 V; 75 < Zrdiff < 125 Ω, 0 < Ztt < 30 Ω.	0.72	1.23	V
Idshort	Transmitter Short Circuit Current	Transmitter terminal(s) shorted to any voltage between -0.25 V and 1.45 V, power on or off.	-50	+50	mA
Zdse	Single-ended Output Impedance	At DC	35	65	Ω
Zddiff	Differential Impedance	At DC	75	125	Ω
RLdse	Single-ended Return Loss	From 0.004*baud rate to 0.75*baud rate relative to 50 Ω.	7.5		dB
RLddiff	Differential Return Loss	From 0.004*baud rate to 0.75*baud rate relative to 100 Ω.	7.5		dB
NOTE 1 Baud Rate = 1 / UI. A subset of this range may be supported. NOTE 2 Maximum rise time is specified by the Eye Mask.					

The transmit eye mask specifies the signal amplitude and jitter. The eye mask is measured into a differential 100 Ω load with more than 20 dB return loss between DC and 1.6*baud rate.

4.4.2 LV-OIF-SxI5 Transmitter Electrical Specifications (cont'd)



XT1 (UI)	XT2 (UI)	YT1 (V)	YT2 (V)	DJ (p-p UI)	TJ (p-p UI)
0.175	0.45	0.50	0.25	0.17	0.35

NOTE 1 Total Jitter (TJ) = Deterministic Jitter (DJ) + Random Jitter (RJ)
NOTE 2 XT1 = TJ / 2.
NOTE 3 Unit Interval (UI) is specified in Table 2.
NOTE 4 Random Jitter (RJ) is defined with respect to a BER of 1e-12 (Q=7.04).

Figure 4 — Transmit Eye Mask for LV-OIF-SxI5 – based operation

4.4.3 LV-OIF-SxI5 Receiver Electrical Specifications

Compliant receivers shall meet all the requirements specified in 4.4.3. The measurement points are at the receive device pins. The line termination shown in Figure 5 is considered part of the Receiver.

Table 3 — General Differential Input DC and AC Characteristics for LV-OIF-SxI5 – based operation

Symbol	Parameter	Conditions	Min	Max	Units
Vrcm	Input Common Mode Voltage	Required only if DC-Compliance is claimed.	0.70	Vtt	V
Vtt	Termination Voltage	Required only if DC-Compliance is claimed.	1.10	1.30	V
Ztt	Vtt Source Impedance	At DC.	-	30	Ω
Zrdiff	Receiver Differential Impedance	At DC.	75	125	Ω
RLrdiff	Differential Return Loss	From 0.004*baud rate to 0.75*baud rate relative to 100 Ω.	10		dB

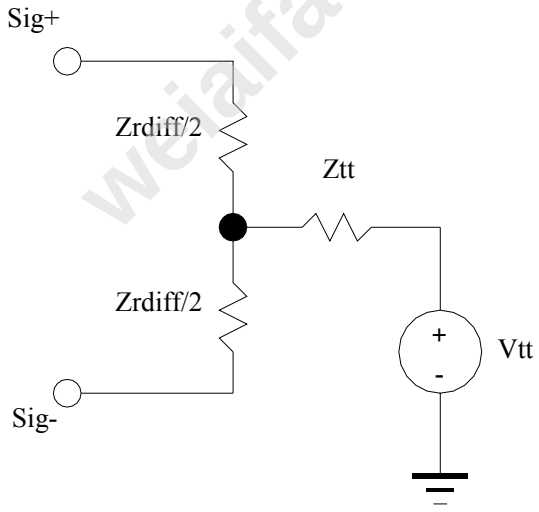
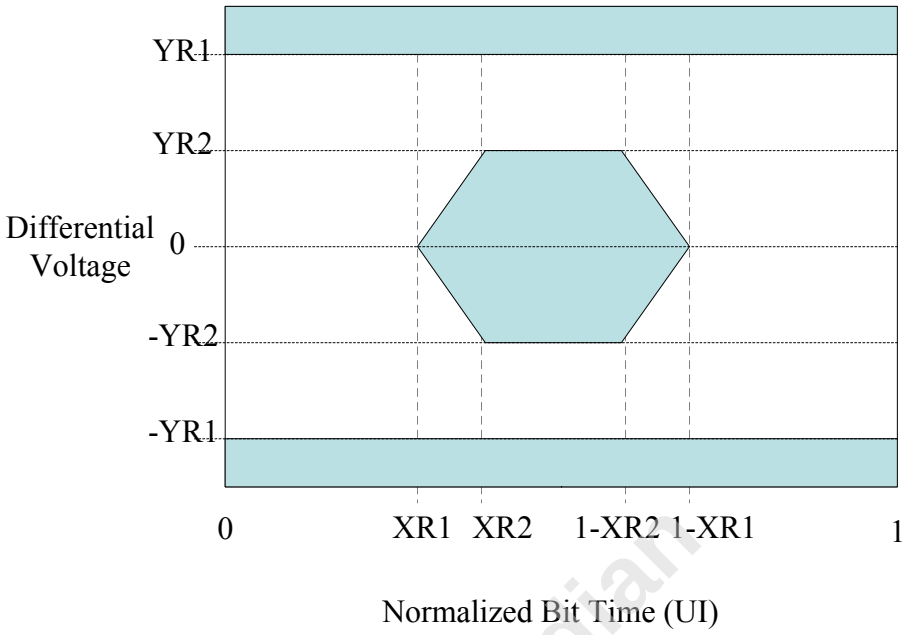


Figure 5 — Line termination at Receiver

The Receive Eye Mask specifies the signal amplitude and jitter tolerance requirements for the Receiver. The eye mask is measured into a differential 100 Ω load with more than 20 dB return loss between DC and 1.6*baud rate.

4.4.3 LV-OIF-SxI5 Receiver Electrical Specifications (cont'd)



XR1 (UI)	XR2 (UI)	YR1 (V)	YR2 (V)	DJ (p-p UI)	TJ (p-p UI)
0.28	0.39	0.50	0.0875	0.32	0.56

NOTE 1 Total Jitter (TJ) = Deterministic Jitter (DJ) + Random Jitter (RJ)
NOTE 2 XR1 = TJ / 2.
NOTE 3 Unit Interval (UI) is specified in Table 2.
NOTE 4 Random Jitter (RJ) is defined with respect to a BER of 1e-12 (Q=7.04).

Figure 6 — Receive Eye Mask for LV-OIF-SxI5 – based operation

4.5 LV-OIF-6G-SR Data interface signals (Differential)

4.5.1 Compliance verification

The device implementer shall consult OIF CEI 2.0 (Ref. 4) regarding the exact interpretation of the items specified in 4.5.2 and 4.5.3 and their measurement techniques. However, the PRBS31 sequence is not used for compliance testing in JESD204. Instead of PRBS31, JESD204 transmitters shall be compliance tested with either the "modified RPAT" or the JSPAT pattern and JESD204 receivers with the JTSPAT pattern, see 5.3.3.8.2. INCITS TR-35-2004 (Ref. 9) is recommended as a useful resource for signal quality and jitter tolerance measurement methodologies.

4.5.2 LV-OIF-6G-SR Transmitter Electrical specifications

Compliant transmitters shall meet all the requirements specified in 4.5.2.

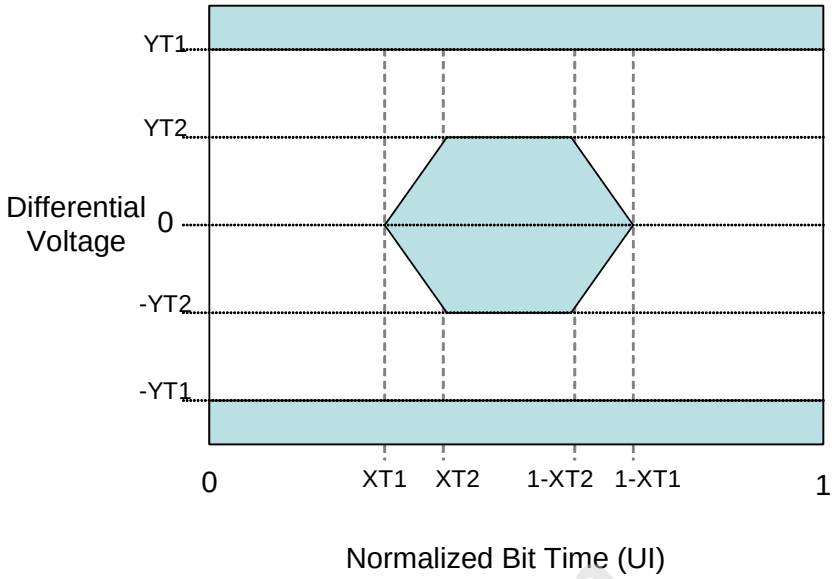
Table 4 — General differential output DC and AC characteristics for LV-OIF-6G-SR – based operation

Symbol	Parameter	Conditions	Min	Max	Units
UI	Unit Interval	(Note 1)	156.9	3,200	ps
Trise/Tfall	Rise and Fall Times	20% - 80% into 100 Ω load	30	(Note 2)	ps
T_Vcm	Output Common Mode Voltage	Applies only to AC coupling. Termination circuit of Figure 5 applied at the Transmitter terminals and Transmitter Ground with parameters as per (Note 3).	0	1.8	V
		Required only if DC-Compliance is claimed. Termination circuit of Figure 5 applied at the Transmitter terminals and Transmitter Ground with parameters as per (Note 4).	735	1135	mV
		Required only if DC-Compliance is claimed. Termination circuit of Figure 5 applied at the Transmitter terminals and Transmitter Ground with parameters as per (Note 5).	550	1060	mV
		Required only if DC-Compliance is claimed. Termination circuit of Figure 5 applied at the Transmitter terminals and Transmitter Ground with parameters as per (Note 6).	490	850	mV
Vdiff	Transmitter Differential Voltage	Into floating 100 Ω load	400	750	mVppd
Idshort	Transmitter Short Circuit Current	Transmitter terminal(s) shorted to each other or ground, power on.	-100	+100	mA
Zddiff	Differential Impedance	At DC	80	120	Ω
RLddiff	Differential Output Return Loss	From 100MHz to 0.75*Baud Rate	8		dB
RLdcm	Common Mode Return Loss	From 100MHz to 0.75*Baud Rate	6		dB

NOTE 1 Baud Rate = 1 / UI. A subset of this range may be supported.
NOTE 2 Maximum rise time is specified by the Eye Mask.
NOTE 3 Ztt ≥ 1kΩ
NOTE 4 Ztt ≤ 30 Ω, Vtt = 1.2V Nominal
NOTE 5 Ztt ≤ 30 Ω, Vtt = 1.0V Nominal
NOTE 6 Ztt ≤ 30 Ω, Vtt = 0.8V Nominal

The transmit eye mask specifies the signal amplitude and jitter. The eye mask is measured into a differential 100 Ω load with more than 20 dB return loss between DC and 1.6*baud rate.

4.5.2 LV-OIF-6G-SR Transmitter Electrical specifications (cont'd)



XT1 (UI)	XT2 (UI)	YT1 (V)	YT2 (V)	T_UBHPJ (p-p UI)	T_DCD (p-p UI)	TJ (p-p UI)
0.15	0.4	0.375	0.20	0.15	0.05	0.3

NOTE 1 T_UCHPJ = Transmit Uncorrelated Bounded High Probability Jitter
NOTE 2 T_DCD = Transmit Duty Cycle Distortion
NOTE 3 XT1 = TJ / 2.
NOTE 4 Unit Interval (UI) is specified in Table 4.
NOTE 5 The Gaussian Jitter (GJ) portion of the Total Jitter (TJ) is defined with respect to a BER of 1e-15 (Q=7.94).

Figure 7 — Transmit Eye Mask for LV-OIF-6G-SR – based operation

4.5.3 LV-OIF-6G-SR Receiver specifications

Compliant receivers shall meet all the requirements specified in 4.5.3. The measurement points are at the receive device pins. The line termination shown in Figure 5 is considered part of the Receiver.

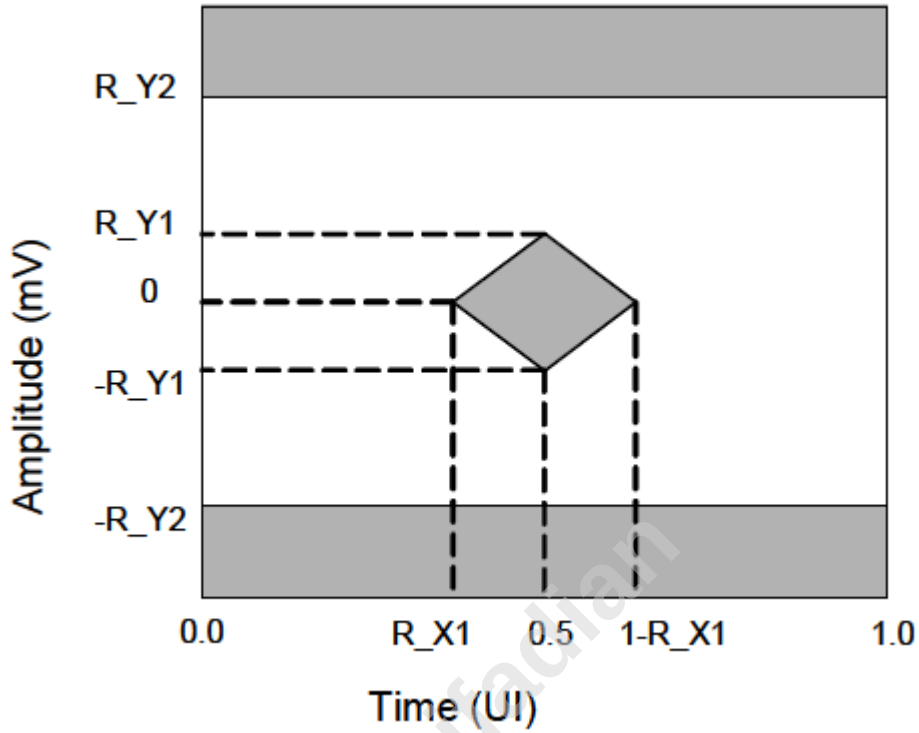
Table 5 — General Differential Input DC and AC Characteristics for LV-OIF-6G-SR – based operation

Symbol	Parameter	Conditions	Min	Max	Units
Vtt	Termination Voltage	Required only if DC-compliance is claimed. Line Termination circuit of Figure 5 with parameters as per (Note 1).	1.2 – 8%	1.2 + 5%	V
		Required only if DC-compliance is claimed. Line Termination circuit of Figure 5 with parameters as per (Note 2).	1.0 – 8%	1.0 + 5%	V
		Required only if DC-compliance is claimed. Line Termination circuit of Figure 5 with parameters as per (Note 3).	0.8 – 8%	0.8 + 5%	V
Vrcm	Input Common Mode Voltage	Applies only to AC coupling	-0.05	1.85	V
		Required only if DC-compliance is claimed. Line Termination circuit of Figure 5 with parameters as per (Note 1).	720	Vtt – 10	mV
		Required only if DC-compliance is claimed. Line Termination circuit of Figure 5 with parameters as per (Note 2).	535	Vtt + 125	mV
		Required only if DC-compliance is claimed. Line Termination circuit of Figure 5 with parameters as per (Note 3).	475	Vtt + 105	mV
R_Vdiff	Input Differential Voltage		125	750	mVppd
Ztt	Vtt Source Impedance	At DC.	-	30	Ω
Zrdiff	Receiver Differential Impedance	At DC.	80	120	Ω
RLrdiff	Differential Input Return Loss	From 100 MHz to 0.75*baud rate relative to 100Ω.	8		dB
RLrcm	Common Mode Input Return Loss	From 100 MHz to 0.75*baud rate relative to 100Ω.	6		dB

NOTE 1 Vtt = 1.2V Nominal
NOTE 2 Vtt = 1.0V Nominal
NOTE 3 Vtt = 0.8V Nominal

The Receive Eye Mask specifies the signal amplitude and jitter tolerance requirements for the Receiver. The eye mask is measured into a differential 100 Ω load with more than 20 dB return loss between DC and 1.6*baud rate.

4.5.3 LV-OIF-6G-SR Receiver specifications (cont'd)



R_X1 (UI)	1-R_X1 (UI)	R_Y1 (V)	R_Y2 (V)	R-SJ-hf (p-p UI)	R-SJ-max (p-p UI)	R_BHPJ (p-p UI)	TJ (p-p UI)
0.30	0.70	0.0625	0. 375	0.05	5	0.45	0.60

NOTE 1 R-SJ-hf = Receive Sinusoidal Jitter, High Frequency
NOTE 2 R_BHPJ = Receive Bounded High Probability Jitter – Breakdown is 0.15 UIpp Uncorrelated, 0.30 UIpp Correlated.
NOTE 3 R_X1 = TJ / 2.
NOTE 4 Unit Interval (UI) is specified in Table 4.
NOTE 5 Total Jitter does not include Sinusoidal Jitter
NOTE 6 The Gaussian Jitter (GJ) portion of the Total Jitter (TJ) is defined with respect to a BER of 1e-15 (Q=7.94).

Figure 8 — Receive Eye Mask for LV-OIF-6G-SR – based operation

4.6 LV-OIF-11G-SR Data interface signals (Differential)

Whilst LV-OIF-11G-SR defines baud-rates between 9.95 Gsym/s to 11.1 Gsym/s, this data interface defines the maximum baud-rate as 12.5 Gsym/s. The device implementer shall consult OIF CEI 2.0 (reference 4), but increase the T_Baud maximum value to 12.5 Gsym/s in Table 8-1 and Table 8-4 of that document.

For baud-rates of up-to 11.1 Gsym/sec, the normalized bit time (UI) used for transmitter and receiver eye-masks shall be 1 / Baud Rate. However, for baud-rates greater than 11.1 Gsym/sec, the normalized bit time (UI) used for Transmit and Receive eye-masks shall be 1 / 11.1 Gsym/sec = 90.09ps.

4.6.1 Compliance verification

The device implementer shall consult OIF CEI 2.0 (reference 4) regarding the exact interpretation of the items specified in 4.6.2 and 4.6.3 and their measurement techniques. However, the PRBS31 sequence is not used for compliance testing in JESD204. Instead of PRBS31, JESD204 transmitters shall be compliance tested with either the “modified RPAT” or the JSPAT pattern and JESD204 receivers with the JTSPAT pattern, see 5.3.3.8.2. INCITS TR-35-2004 (reference 9) is recommended as a useful resource for signal quality and jitter tolerance measurement methodologies.

4.6.2 LV-OIF-11G-SR Transmitter Electrical specifications

Compliant transmitters shall meet all the requirements specified in 4.6.2.

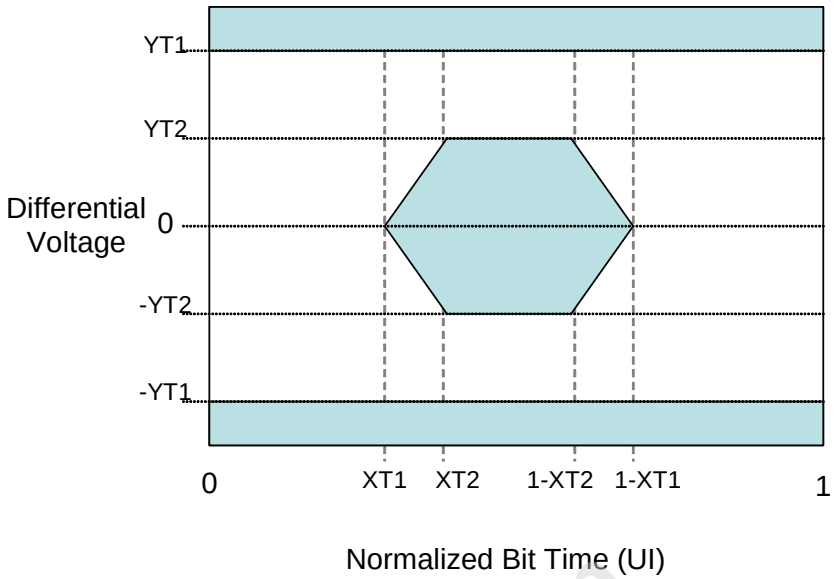
Table 6 — General differential output DC and AC characteristics for LV-OIF-11G-SR – based operation

Symbol	Parameter	Conditions	Min	Max	Units
UI	Unit Interval	(Note 1)	80	156.9	ps
Trise/Tfall	Rise and Fall Times	20% - 80% into 100 Ω load	24	(Note 2)	ps
T_Vcm	Output Common Mode Voltage	Applies only to AC coupling. Termination circuit of Figure 5 applied at the Transmitter terminals and Transmitter Ground with parameters as per (Note 3).	0	1.8	V
		Required only if DC-Compliance is claimed. Termination circuit of Figure 5 applied at the Transmitter terminals and Transmitter Ground with parameters as per (Note 4).	735	1135	mV
		Required only if DC-Compliance is claimed. Termination circuit of Figure 5 applied at the Transmitter terminals and Transmitter Ground with parameters as per (Note 5).	550	1060	mV
		Required only if DC-Compliance is claimed. Termination circuit of Figure 5 applied at the Transmitter terminals and Transmitter Ground with parameters as per (Note 6).	490	850	mV
Vdiff	Transmitter Differential Voltage	Into floating 100 Ω load	360	770	mVppd
Idshort	Transmitter Short Circuit Current	Transmitter terminal(s) shorted to each other or ground, power on.	-100	+100	mA
Zddiff	Differential Impedance	At DC	80	120	Ω
RLddiff	Differential Output Return Loss	From 100MHz to 0.75*Baud Rate	8		dB
RLdcm	Common Mode Return Loss	From 100MHz to 0.75*Baud Rate	6		dB

NOTE 1 Baud Rate = 1 / UI. A subset of this range may be supported.
NOTE 2 Maximum rise time is specified by the Eye Mask.
NOTE 3 Ztt ≥ 1kΩ
NOTE 4 Ztt ≤ 30 Ω, Vtt = 1.2V Nominal
NOTE 5 Ztt ≤ 30 Ω, Vtt = 1.0V Nominal
NOTE 6 Ztt ≤ 30 Ω, Vtt = 0.8V Nominal

The transmit eye mask specifies the signal amplitude and jitter. The eye mask is measured into a differential 100 Ω load with more than 20 dB return loss between DC and 1.6*baud rate.

4.6.2 LV-OIF-11G-SR Transmitter Electrical specifications (cont'd)



XT1 (UI)	XT2 (UI)	YT1 (V)	YT2 (V)	T_UBHPJ (p-p UI)	T_DCD (p-p UI)	TJ (p-p UI)
0.15	0.4	0.385	0.18	0.15	0.05	0.3

NOTE 1 T_UCHPJ = Transmit Uncorrelated Bounded High Probability Jitter
NOTE 2 T_DCD = Transmit Duty Cycle Distortion
NOTE 3 XT1 = TJ / 2.
NOTE 4 Unit Interval (UI) is specified in
Table 6. However, for baud-rates greater than 11.1 Gsym/sec, UI shall be 90.09ps.
NOTE 5 The Gaussian Jitter (GJ) portion of the Total Jitter (TJ) is defined with respect to a BER of 1e-15 (Q=7.94).

Figure 9 — Transmit Eye Mask for LV-OIF-11G-SR – based operation

4.6.3 LV-OIF-11G-SR Receiver specifications

Compliant receivers shall meet all the requirements specified in 4.6.3. The measurement points are at the receive device pins. The line termination shown in Figure 5 is considered part of the Receiver.

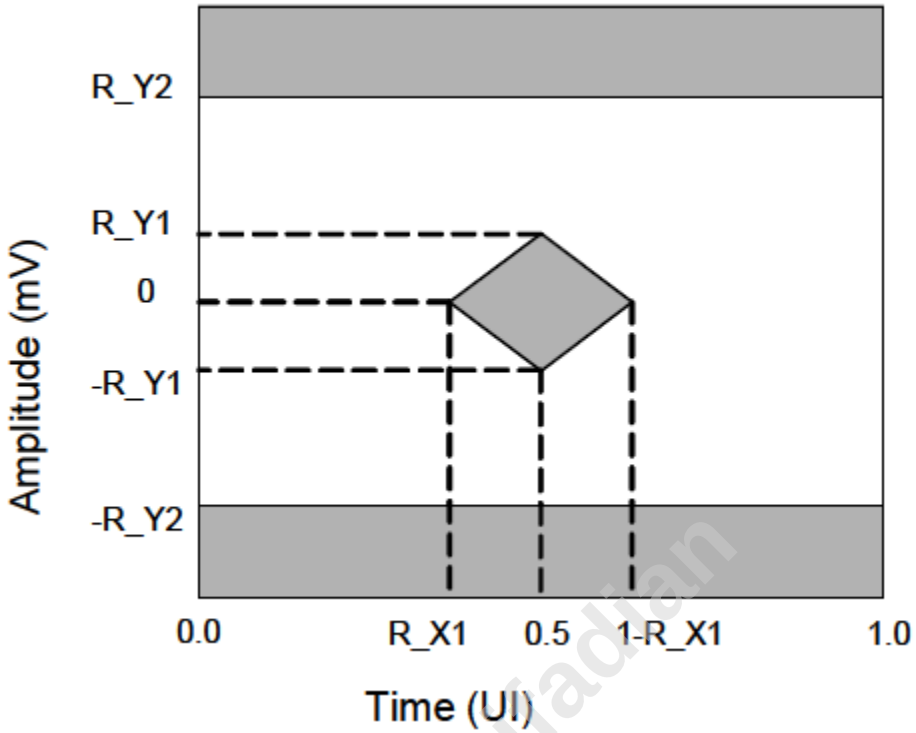
Table 7 — General Differential Input DC and AC Characteristics for LV-OIF-11G-SR – based operation

Symbol	Parameter	Conditions	Min	Max	Units
Vtt	Termination Voltage	Required only if DC-compliance is claimed. Line Termination circuit of Figure 5 with parameters as per (Note 1).	1.2 – 8%	1.2 + 5%	V
		Required only if DC-compliance is claimed. Line Termination circuit of Figure 5 with parameters as per (Note 2).	1.0 – 8%	1.0 + 5%	V
		Required only if DC-compliance is claimed. Line Termination circuit of Figure 5 with parameters as per (Note 3).	0.8 – 8%	0.8 + 5%	V
Vrcm	Input Common Mode Voltage	Applies only to AC coupling	-0.05	1.85	V
		Required only if DC-compliance is claimed. Line Termination circuit of Figure 5 with parameters as per (Note 1).	720	Vtt – 10	mV
		Required only if DC-compliance is claimed. Line Termination circuit of Figure 5 with parameters as per (Note 2).	535	Vtt + 125	mV
		Required only if DC-compliance is claimed. Line Termination circuit of Figure 5 with parameters as per (Note 3).	475	Vtt + 105	mV
R_Vdiff	Input Differential Voltage		110	1050	mVppd
Ztt	Vtt Source Impedance	At DC.	-	30	Ω
Zrdiff	Receiver Differential Impedance	At DC.	80	120	Ω
RLrdiff	Differential Input Return Loss	From 100 MHz to 0.75*baud rate relative to 100Ω.	8		dB
RLrcm	Common Mode Input Return Loss	From 100 MHz to 0.75*baud rate relative to 100Ω.	6		dB

NOTE 1 Vtt = 1.2V Nominal
NOTE 2 Vtt = 1.0V Nominal
NOTE 3 Vtt = 0.8V Nominal

The Receive Eye Mask specifies the signal amplitude and jitter tolerance requirements for the Receiver. The eye mask is measured into a differential 100 Ω load with more than 20 dB return loss between DC and 1.6*baud rate.

4.6.3 LV-OIF-11G-SR Receiver specifications (cont'd)



R_X1 (UI)	1-R_X1 (UI)	R_Y1 (V)	R_Y2 (V)	R_SJ-hf (p-p UI)	R_SJ-max (p-p UI)	R_BHPJ (p-p UI)	TJ (p-p UI)
0.35	0.65	0.055	0. 525	0.05	5	0.45	0.70

NOTE 1 R_SJ-hf = Receive Sinusoidal Jitter, High Frequency
NOTE 2 R_SJ-max = Receive Sinusoidal Jitter, Maximum
NOTE 3 R_BHPJ = Receive Bounded High Probability Jitter – Breakdown is 0.25 UIpp Uncorrelated, 0.20 UIpp Correlated
NOTE 4 R_X1 = TJ / 2.
NOTE 5 Unit Interval (UI) is specified in
Table 6. However, for baud-rates greater than 11.1 Gsym/sec, UI shall be 90.09ps.
NOTE 6 Total Jitter (TJ) includes high-frequency sinusoidal jitter (R_SJ-hf)
NOTE 7 The Gaussian Jitter (GJ) portion of the Total Jitter (TJ) is defined with respect to a BER of 1e-15 (Q=7.94).

Figure 10 — Receive Eye Mask for LV-OIF-11G-SR – based operation

4.7 Device clock

The device clock is the timing reference of each element in the JESD204B system. Each transmitter and receiver device must receive their device clock from a clock generator circuit which is responsible for generating all device clocks from a common source, which is known as the source clock. The device clock signal may have a different period than the frame or multiframe period and, if so, the device is responsible for generating the frame clock period and/or multiframe clock period from the device clock period.

The permitted frequency relationships between device clock, frame clock and multiframe clock frequencies depend on the JESD204B subclass as follows:

- Subclass 0: To be specified by the device implementer.
- Subclass 1: The multiframe period shall be a whole number of device clock periods.
- Subclass 2: The multiframe period shall be a whole number of device clock periods. Additionally, the TX device clock period shall be a whole number of RX device clock periods, or the RX device clock period shall be a whole number of TX device clocks periods.

The electrical characteristics of the device clock signal are not specified in this document, because other considerations may determine these characteristics. It is expected that Low Voltage CMOS type electrical characteristics, as specified by such JEDEC standards as JESD8-5 (2.5 V) or JESD8-7 (1.8 V) would be used for this signal. Higher clock rates might demand the use of a differential signal format, such as LVPECL, LVDS or CML.

The device clock input's frequency and phase stability (jitter), however, have a direct bearing on the operation of this interface. For this reason, the following informative guideline is given:

The device clock total peak-to-peak jitter (random and deterministic) at the device inputs should not exceed 0.125 UI when measured across a jitter frequency range from 12 kHz to 20 MHz.

Other circuitry in the Transmitter or Receiver device may require more restrictive jitter characteristics and may also have additional requirements such as duty cycle. The requirement stated above is deemed the minimum necessary for compliant operation of the high-speed, differential data interface specified in this document.

4.8 Frame Clock, and Local Multiframe Clock

The frame clock domain forms the interface between the application layer and the JESD204 link layer.

For links on which the data is arranged in multiframes (mandatory for links supporting deterministic latency and/or those having multiple lanes), the multiframes are aligned to the edges of the (local) multiframe clocks (LMFC) in those devices.

Each transmitter and receiver device must receive a separate device clock signal from which the frame and multiframe periods can be derived. The device clock is specified in 4.7. Optionally, the device clock may be supplied at the frequency of the frame or multiframe clock. A frame or multiframe clock that is not supplied directly to the device clock input, and is instead derived within the device is referred to as a “local” clock.

4.8 Frame Clock, and Local Multiframe Clock (cont'd)

If the multiframe clock is derived within a device, the phase of the LMFC is dictated by the sampled SYSREF input for Subclass 1 devices, and by the SYNC~ rising edge for Subclass 2 devices. For a definition of device subclass types, please refer to 9.1.

Frame and multiframe clocks shall comply with the following requirements:

- The frame period in all transmitter and receiver devices must be identical
- The multiframe period in all transmitter and receiver devices must be identical
- All frame clocks and multiframe clocks in the JESD204 system must be derived from a common clock source.
- In each device, the frame clock and LMFC must be phase-aligned to each other.
- The phase of the frame clock and LMFC shall be dictated by the device clock edge upon which the SYSREF signal is detected as being active (for Subclass 1 devices).
- The phase of the frame clock shall be dictated by the "adjustment clock" (6.4.1.2) edge following the detection of the SYNC~ de-assertion. (for Subclass 2 devices)
- Devices may optionally allow the LMFC (and Frame Clock) phase alignment to be adjusted in fine-grained increments. This is to provide flexibility in perfectly aligning LMFCs within all devices in a system.

If a device supports multiple JESD204 links, the above requirements are applied separately to each link.

4.9 SYNC interface

The SYNC interface is used as a time-critical return path from receiver(s) to transmitter(s). It shall be synchronous with the internal frame clock of the RX device. It is strongly recommended that synchronicity be maintained with the TX frame clock as well if specific clauses requiring informational passage over the SYNC interface (required for Subclass 0 and Subclass 2 operation) are to be supported. It is also strongly recommended to use a similar interface for the SYNC interface and the device clock, in order to maintain an accurate timing relationship (with the exception that SYNC~ should never be AC-coupled). The SYNC interface contains exactly one signal, which is denoted by SYNC~. The tilde indicates that the signal is active low. In case of a differential interface, the true part of the signal is active low.

Figure 11 shows the critical timing specifications relating to the SYNC~ signal that are necessary for both Subclass 0, due to backward compatibility to JESD204A, and Subclass 2 deterministic latency devices. The values for these parameters are not specified here but the Transmitter and Receiver device specifications shall specify these values.

$\mathbf{t}_{DS\_R}$ (min/max): Device-Clock-to-SYNC~ Delay at Receiver device pins. Subclass 0 and Subclass 2 Receiver devices must specify this parameter.

$\mathbf{t}_{SU\_T}$ (min) and $\mathbf{t}_{H\_T}$ (min): Setup and Hold times of SYNC~ with respect to Device Clock at the Transmitter device pins. Subclass 0 and Subclass 2 Transmitter devices must specify these parameters.

4.9 SYNC interface (cont'd)

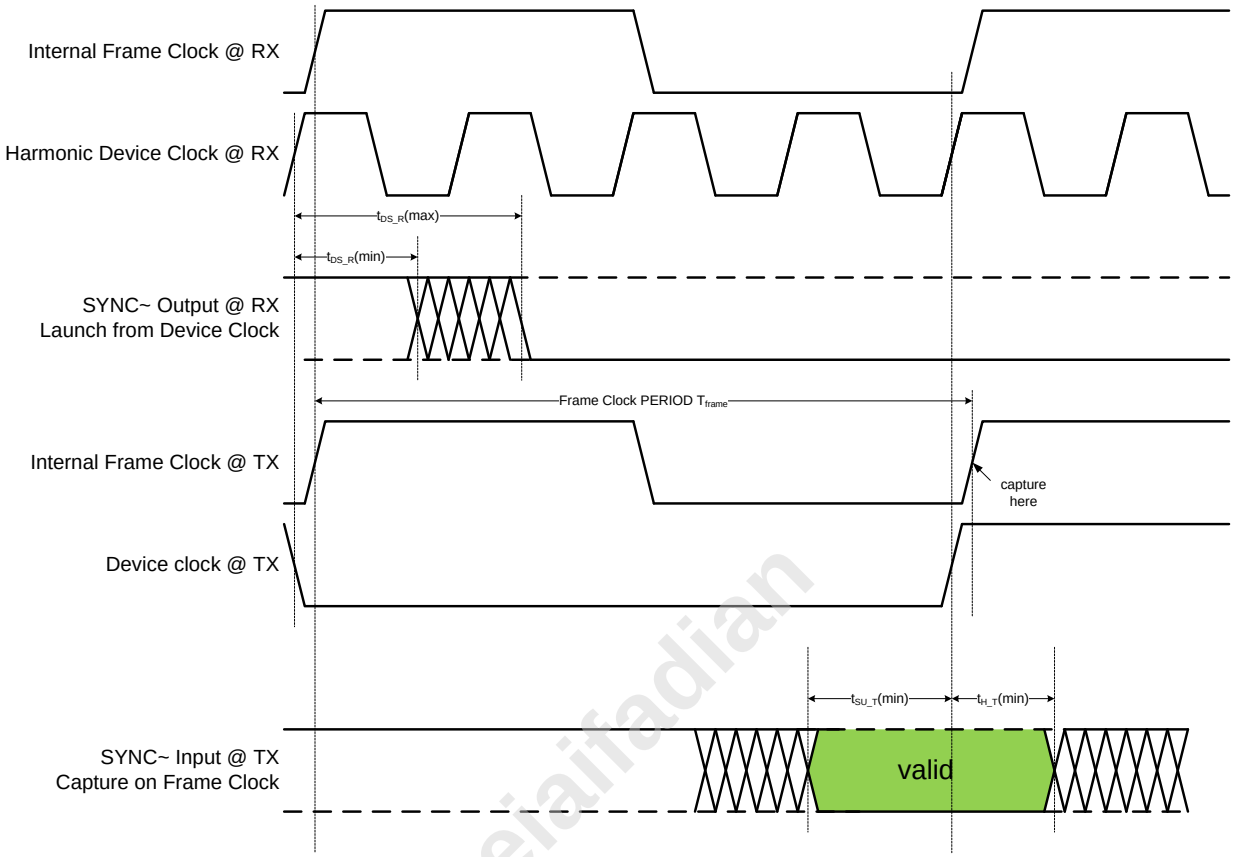


Figure 11 — SYNC~ signal timing for Subclass 0 and Subclass 2 Devices

In devices where the device clock is faster than or equal to the frame clock, then timing can be related to the device clock. In a device where the frame clock is faster than the device clock, SYNC~ may be launched or captured by the frame clock and the timing can only be measured relative to an associated edge. The frame clock is aligned to the device clock for at least one edge per multi-frame. This edge of the device clock is associated to an edge of the frame clock, and measurements can be taken relative to it. All other timing is relative to this basis and will differ by some number of frame periods (T_f).

Figure 12 shows edge associations for two RX SYNC~ launch cases. In the first case, the frame clock is slower than the device clock. In the second, the device clock is slower than the frame clock.

4.9 SYNC interface (cont'd)

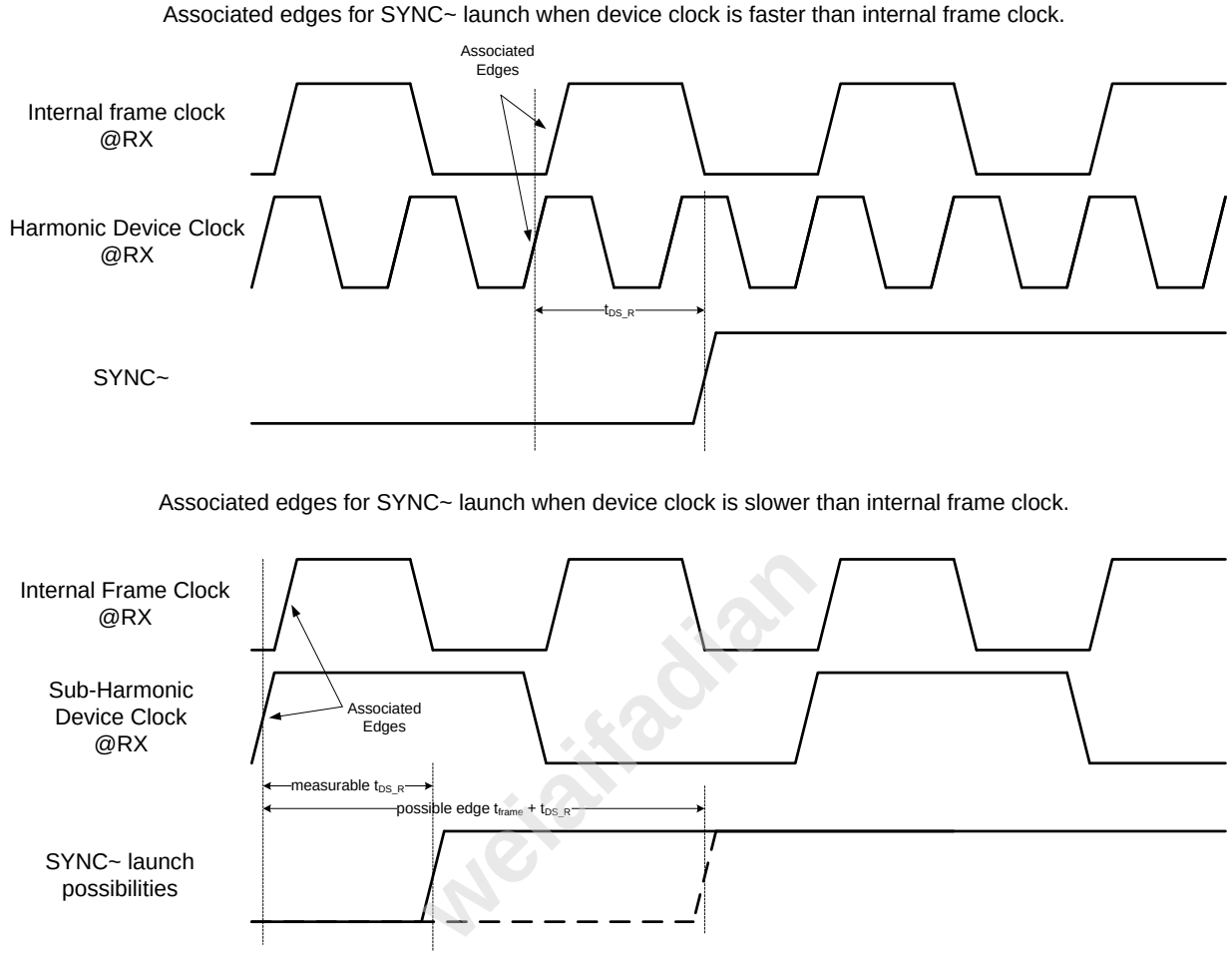


Figure 12 — Associated device clock edges for Subclass 0 and Subclass 2 devices

4.10 Lane-to-lane inter-device synchronization interface

For devices not supporting deterministic latency (i.e. Subclass 0), synchronization between different receiver devices (DACs) will require a separate interface for synchronizing the local timing references for lane alignment. It is expected that this interface will be synchronous to the frame clock. It is recommended that this interface use the same electrical characteristics as the device clock interface. The device user shall not be required to generate a separate clock signal for the lane-to-lane inter-device synchronization interface.

For devices that do support deterministic latency, there is no requirement for this separate inter-device synchronization interface.

4.11 SYSREF signal (Device Subclass 1)

In Subclass 1 deterministic latency systems, a signal named SYSREF is distributed to all devices in the system. The purpose of SYSREF is to identify the device clock edge that should be used to align the phase of the internal LMFC and frame clock. Because the LMFC and frame clock are usually aligned to the character clock, it may be necessary to adjust the phase of the character clock simultaneously when LMFC and frame clock are being adjusted.

SYSREF can be either a periodic, one-shot (strobe-type), or "gapped" periodic signal. It is an active high signal which is sampled by the device clock. Devices must be able to support sampling SYSREF on the device clock rising edge, but may optionally allow SYSREF to be sampled on the device clock falling edge.

For periodic or "gapped" periodic SYSREF signals, the period should be an integer multiple of the LMFC period. The LMFC and frame clock within a device shall be phase aligned to the device clock sampling edge upon which the sampled SYSREF value has transitioned from 0→1.

It is not mandatory for the same SYSREF signal to be generated to all devices in the system. However, it is required that SYSREF signals be generated to all devices in a manner that ensures a deterministic relationship between when SYSREF is sampled active by all devices in the system. A timing diagram illustrating this concept (using device clock rising edge to sample SYSREF) and showing simultaneous sampling of the active SYSREF signal at both devices is shown below in Figure 13.

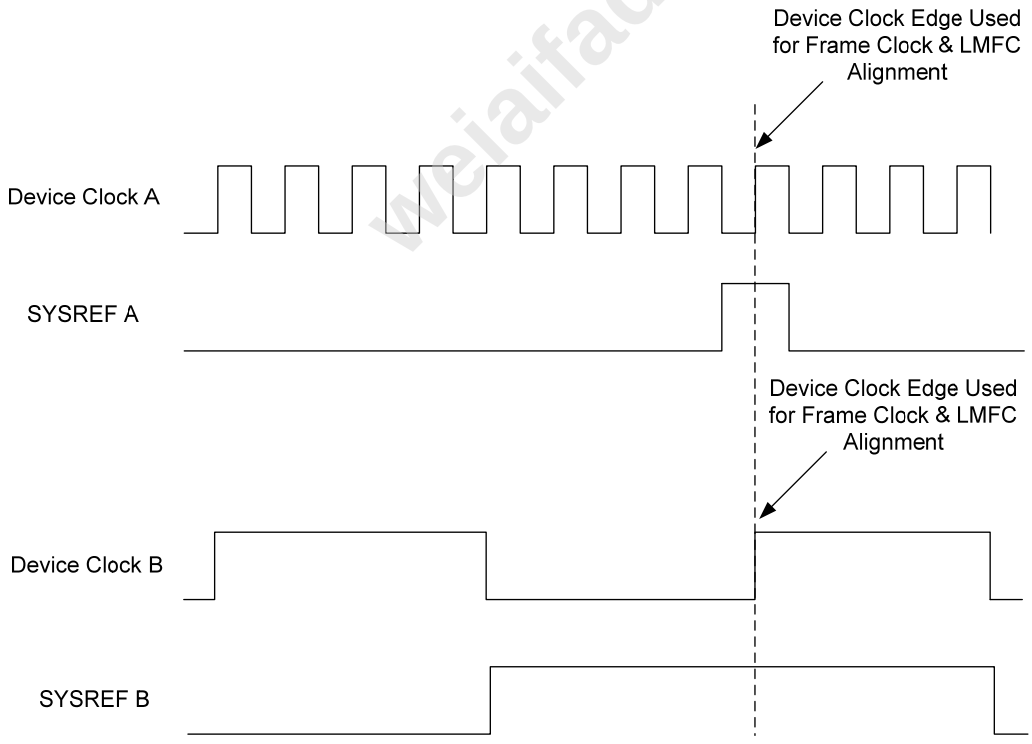


Figure 13 — SYSREF Sampling Illustration

4.11 SYSREF signal (Device Subclass 1) (cont'd)

Delay uncertainty across the JESD204B link is caused by the uncertainty in phase alignment between the LMFCs in the TX and RX devices within the system. Minimizing delay uncertainty relies on the generation of the frame clock and LMFC in both the TX and RX to be governed by events that happen at the same time instant. For Subclass 1 devices, this time instant corresponds to the device clock cycle upon which a 0→1 transition on SYSREF is detected. Therefore, it is recommended that the system implementer follows these guidelines for minimizing phase alignment offset between TX and RX LMFCs.

1. Skew between device clock inputs in the system must be minimized.
2. SYSREF (source synchronous to device clock) must be distributed within the system for maximum setup/hold compliance at the device receiver.
3. SYSREF should ideally have the timing relationship relative to the device clock as shown in Figure 14. However, with very high speed device clocks, it may not be possible to meet SYSREF setup/hold time requirements. The resulting amount of delay uncertainty across the JESD204B link can be minimized by controlling the phase of the SYSREF signal relative to the setup/hold time requirements of the device which is sampling it.

It is strongly recommended to use the same type of signal type for SYSREF and the device clock, in order to maintain an accurate timing relationship.

Figure 14 shows the critical timing specifications relating to the SYSREF~ signal. The values for these parameters are not specified here but the Transmitter and Receiver device specifications shall specify these values.

$\mathbf{t}_{SU\_S}$ (min) and $\mathbf{t}_{H\_S}$ (min): Setup and Hold times of SYSREF with respect to Device Clock at the device SYSREF pin. Sublcass 1 Transmitter and Receiver devices must specify these parameters.

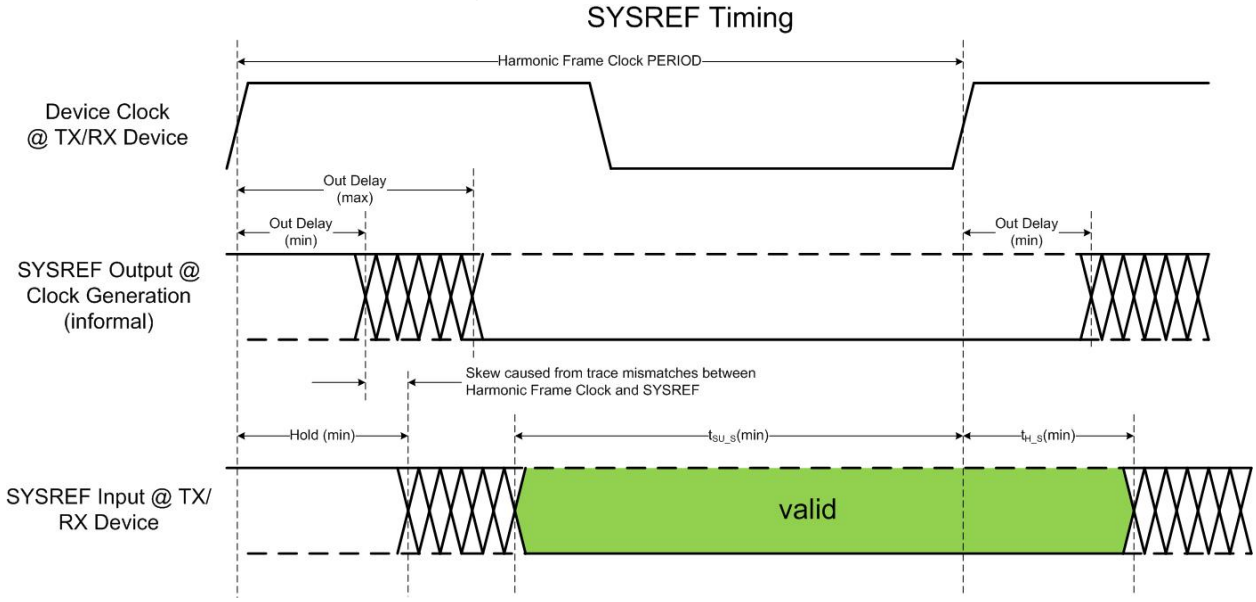


Figure 14 — SYSREF signal timing

4.12 Skew and misalignment budget

Skew is the difference in arrival time between signals that should nominally arrive simultaneously. Skew is always comparing similar signals and at the same side of the (multipoint) link, either the transmitting or the receiving side. Skew is introduced at various places in the link(s) and within the clock distribution circuit. The implementation of the system shall be such that certain amounts of skew can be tolerated without affecting system performance. In JESD204A this means that all data generated simultaneously at the TX side will be aligned to the same frame period at the RX side. In JESD204B subclasses 1 and 2 this means alignment to the same and predictable frame period.

The total skew budget can be broken down into the following parts:

- **Interconnect skew**: the maximum difference in propagation time between any two lanes of a link or a multipoint link. Interconnect skew is measured between device pins. Interconnect skew depends on the physical properties of the lanes, such as length and dielectric constant.
- **Intra-device skew**: the maximum difference in arrival time at a given output reference plane of signals that have been provided simultaneously to an input reference plane or have been generated simultaneously inside the device. Intra-device skew is measured between pins of the same device, or between signals inside the same device. In most JESD204 devices the SERDES and the application layer will operate in different clock domains. Data must cross from a source clock domain to a destination clock domain that have no known phase relationship to each other. If the data from the source clock domain does not arrive in time to be sampled at a certain edge of the destination clock, it must wait until the next edge to be sampled. Therefore a small variation in clock phase or propagation delay can cause a delay change of one destination clock period at a clock domain crossing. When the SERDES clock domains of different lanes are not synchronized to each other, clock domain crossings can thus cause a substantial delay variation between lanes. For signals where clock domain crossings add no delay uncertainty, intra-device skew is only caused by the differences in propagation delays between different paths.
- **Inter-device skew**: the maximum difference between the propagation time from a given input to a given output in one device and the propagation time between a corresponding input and output in another device. Inter-device skew is measured between pins of different devices, or between signals inside different devices Similar to intra-device skew, inter-device skew is caused by clock domain crossings and by physical delays. Because there will be more variation in clock phases and physical delays between devices than inside devices, inter-device skew will generally be larger than intra-device skew.
- **Clock distribution skew**: the maximum difference in arrival time of clock signals that have been generated from a common source. Clock distribution skew is measured between device pins. Clock distribution skew depends on the matching in propagation delays between buffers and other components used in the clock distribution circuit, and on the matching of the physical lengths of the clock lines to different devices. Also, the finite rise times of clock signals in combination with tolerances on the threshold voltages of the clock inputs cause clock distribution skew. Therefore, fast clocks will typically exhibit less skew than slow clocks.
- **SYSREF distribution skew**: the maximum difference in arrival time of SYSREF signals that have been generated from a common source. SYSREF distribution skew is measured between device pins. The same factors play a role as in clock distribution skew.

4.12 Skew and misalignment budget (cont'd)

- **SYNC~ distribution skew:** the maximum difference in arrival time of SYNC~ signals that have been generated from a common source (RX logic device) or simultaneously from multiple sources (DACs). SYNC~ distribution skew is measured between device pins. The same factors play a role as in clock distribution skew.
- **Inter-device SYNC~ generation skew**: The maximum difference in delays from the rising edge of the LMFC to the rising edge of the SYNC~ signal in different receiver devices, given a common trigger event.

There is no direct relation between interconnect skew and serial bit rate. In many cases the insertion loss requirement for the higher baud rates would lead to a smaller interconnect length and consequently lower skew. However, there are means to increase the link length, such as the use of a low loss medium, pre-emphasis at the TX and equalization at the RX. Therefore the maximum allowable interconnect skew in absolute time units shall be independent of the baud rate. When expressed in units of UI, the allowable interconnect skew shall be proportional to the baud rate.

Because clock domain crossings are the main contributors to intra- and inter-device skews, these skews are generally specified as functions of the clock periods. Typical SERDES devices have an internal parallel clock period corresponding to two octets, or 20 serial bits. Therefore XAUI (reference 6) assumes a maximum skew of 20 UI between receivers of different lanes. If a clock crossing to the frame clock domain causes timing uncertainty, the maximum skew will be one frame period, or $10 \cdot F$ UI.

Clock distribution skew only plays a role on multipoint links. It will provide the different converter devices with different timing references. On a single link, all lanes will share a common timing reference and clock distribution skew plays no role.

In JESD204B subclass 1 devices, the detection of the rising edge of SYSREF defines the phase of the local multiframe clock LMFC. In the TX, the detected phase of the LMFC determines the moments when alignment characters can be sent. In the RX, the detected phase of the LMFC determines the moments that the alignment characters are read out from the FIFO. SYSREF distribution skew can thus cause different delays on the different links that constitute a multipoint link. However, this will only happen if the skew causes detection of a rising SYSREF edge on a non-intended edge of the device clock. If the device clock is late in one device, but SYSREF is equally late, the skew of SYSREF will cause no additional timing errors on the link. SYSREF distribution skew as such is thus not harmful, as long as SYSREF stays properly aligned to the device clock in each device.

In ADCs conforming to JESD204A or JESD204B subclasses 0 or 2, the rising edge of SYNC~ determines when the TX can send alignment signals over the link. In these variants SYNC~ distribution skew can cause misalignment between ADCs on a multipoint ADC link in the same way as SYSREF distribution skew in subclass 1.

In JESD204B Subclass 0 and JESD204A devices, multipoint DAC links must use SYNC~ combining at the logic device TX to ensure that alignment characters will be sent simultaneously to all DACs. Therefore SYNC~ generation and distribution skew are not critical for lane alignment in Subclass 0 and JESD204A devices. However, the skew could result in violation of setup and hold times for the SYNC~ signal at the TX, such that a JESD204B TX could not always detect an error report from a JESD204A RX or a JESD204A TX could misinterpret an error report from an RX as a synchronization request.

4.12 Skew and misalignment budget (cont'd)

When JESD204B devices are communicating together, error reports and synchronization requests can be detected reliably even when SYNC~ setup and hold times are violated.

Subclass 1 is insensitive to SYNC~ skew, because it uses SYSREF as its timing reference instead of SYNC~. However, in subclass 2 the logic device TX estimates the phase of the LMFCs in the DACs from the arrival time of the SYNC~ pulses and based on that gives correction commands to the DACs to adjust their LMFCs. In subclass 2, SYNC~ generation and distribution skew will thus directly affect the alignment between multiple DAC links if the skew exceeds the SYNC~detection resolution in the TX. Skew is defined between points located at the same side of the link. In addition, the latency on each link depends on the timings of clock and reference signals at the RX side relative to the TX side. These timings are part of the system implementation and not of JESD204. JESD204 cares only about the alignment between signals at the same side of the link.

The items in the skew budget are illustrated in

Figure 15 and

Figure 16.

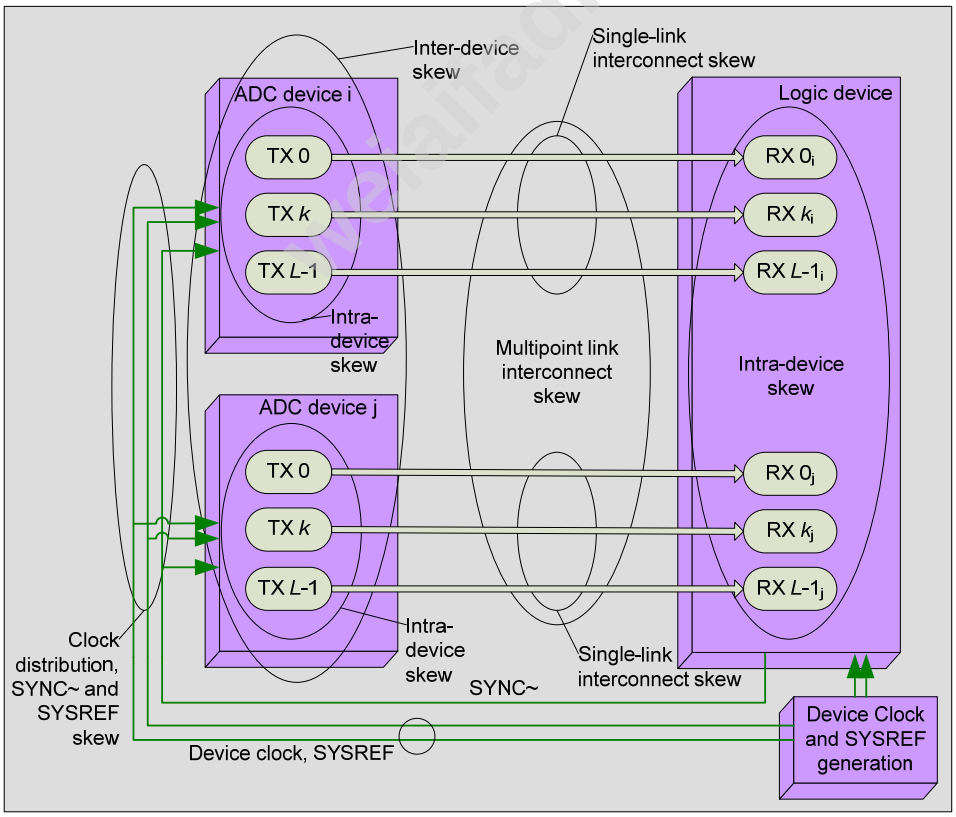


Figure 15 — Skew items in configuration with multiple ADC devices
(SYNC~ combining enabled in this example)

4.12 Skew and misalignment budget (cont'd)

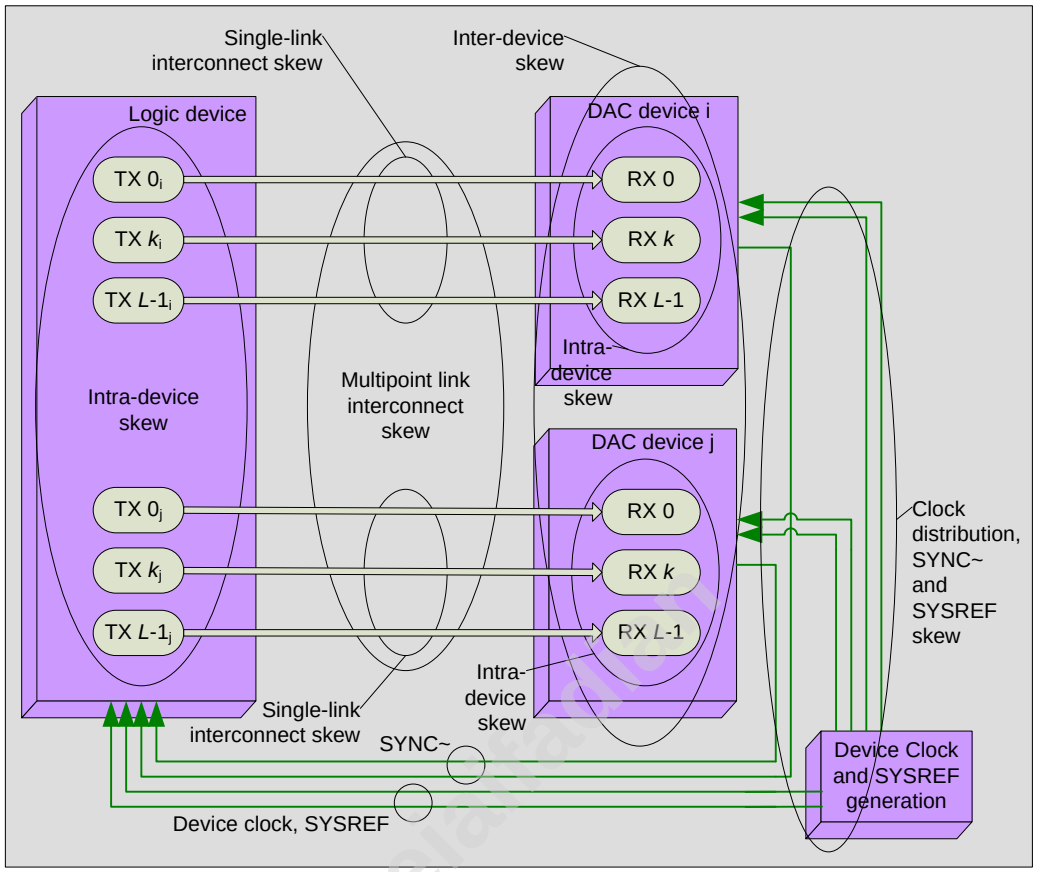


Figure 16 — Skew items in configuration with multiple DAC devices

The following tables specify the allowable skew for all link elements. Skew is most easily measured during the initial lane alignment sequence, when all transmitters transmit the same symbol sequence, apart from possibly small differences in link configuration octets. Skew is interpreted according to the definition in references 3 and 4, not including wander.

The specifications per link element may depend on frame length F and the number of used lanes L or L_{MP}, but do not depend otherwise on link configuration. However, not all skew items are relevant in each configuration. As a result, the required skew correction capabilities per receiver depend on the configuration. The allowed skew values scale with the unit interval UI when UI is more than 320 ps (lane speed less than 3.125 Gbps). When UI is less than 320 ps, the allowed skew values do not depend on the lane speed. A skew of 320 ps corresponds to about 5 cm length difference between cables or traces of a printed circuit board. The receiver skew includes the deserialization function, physical deserializer skew and clock boundary transitions.

4.12 Skew and misalignment budget (cont'd)

Table 8 — Skew budget for link from single ADC device

Skew Source	Skew [SU*]	Notes
Intra-device skew (ADC)	min(5+5·*F*,20)	
Single-link interconnect skew	3	Supports e.g. 1 SU skew on a TX PCB, 1 SU on an RX PCB and 1 SU in a connecting cable.
Intra-device skew (RX)	20	Informative, value from XAUI (reference 6).
Deskewing capability at RX input	13 ... 23	Depending on frame length. This is the maximum skew between the incoming serial bits.
Deskewing capability at RX output	33 ... 43	Informative, assuming deskewing in character or frame clock domain.
* Skew unit SU= max(UI, 320 ps)		

Table 9 — Skew budget for link to single DAC device

Skew Source	Skew [SU*]	Notes
Intra-device skew (Logic Device TX)	$\sqrt{L}$	Up to *L*= 32 lanes supported by JESD204A.
Single-link interconnect skew	3	Supports e.g. 1 SU skew on a TX PCB, 1 SU on an RX PCB and 1 SU in a connecting cable.
Intra-device skew (DAC)	20	Informative, value from XAUI (reference 6).
Deskewing capability at DAC input	5 ... 9	Depending on number of lanes *L*. This is the maximum skew between the incoming serial bits.
Deskewing capability at DAC output	25 ... 29	Informative, assuming deskewing in character or frame clock domain.
* Skew unit SU= max(UI, 320 ps)		

4.12 Skew and misalignment budget (cont'd)

Table 10 — Skew budget for configuration with multiple ADC devices

Skew Source	Skew [SU*]	Notes
Intra-device skew (ADC)	$\min(5+5\cdot F,20)$	If single lane per ADC device, this item may be interpreted as extra tolerance in device delay.
Inter-device skew (ADC)	$\min(5+5\cdot F,20)$	
Clock distribution skew	$\min(4+2\cdot\frac{BaudRate}{10 * DeviceClockRate},20)$ **	Includes effect of finite rise times combined with mismatches in threshold voltages.
Single-link interconnect skew	3	
Multipoint link interconnect skew	8**	
Intra-device skew (RX)	20	Informative, value from XAUI (reference 7).
SYNC~ and SYSREF distribution skew	see notes →	For lowest uncertainty in latency, the system implementer shall ensure that in each ADC the SYNC~ or SYSREF signal*** will be decoded at the same rising edge of the device clock.
Deskewing capability at RX input	34 ... 68	Depending on frame length. This is the maximum skew between the incoming serial bits.
Deskewing capability at RX output	54 ... 88	Informative, assuming deskewing in character or frame clock domain.

* Skew unit SU= max(UI, 320 ps)

** For system operation it is important that the sum of the clock distribution skew and multipoint link interconnect skew does not exceed the total specified value. One of the two items may be larger than specified, if this is compensated in the other item.

*** SYSREF is relevant for JESD204B subclass 1, SYNC~ for all other variants.

4.12 Skew and misalignment budget (cont'd)

Table 11 — Skew budget for configuration with multiple DAC devices

Skew Source	Skew [SU*]	Notes
Intra-device single-link skew (logic device TX)	$\sqrt{L}$	Up to L= 32 lanes supported by JESD204A.
Total intra-device skew (logic device TX)	$\sqrt{L_{MP}}$	Total number of lanes L_{MP} connected to multipoint link expected to be ≤ 256.
Single-link interconnect skew	3	
Multipoint link interconnect skew	8**	
Clock distribution skew	$\min(4 + 2 \cdot \frac{BaudRate}{10 * DeviceClockRate}, 20)$ **	Includes effect of finite rise times combined with mismatches in threshold voltages.
Intra-device skew (DAC)	20	Informative, value from XAUI (reference 7). If single lane per DAC device, this item may be interpreted as extra tolerance in device delay.
Inter-device skew (DAC)	$\min(5+5 \cdot F, 20) + 10 \cdot F$	Informative, value from ADC inter-device tolerance plus one added frame cycle due to clock boundary transition.
SYNC~ generation and distribution skew	see notes →	In JESD204B subclass 2, for lowest uncertainty in latency, the skew shall not exceed the SYNC~ detection resolution in the logic device.
SYSREF distribution skew	see notes →	In JESD204B subclass 1, for lowest uncertainty in latency, the system implementer shall ensure that in each DAC the SYSREF signal will be decoded at the same rising edge of the device clock.
Deskewing capability at DAC input	5 ... 9	Depending on number of lanes L. This is the maximum skew between the incoming serial bits on the same link.
Deskewing capability at DAC output	(46 ... 84) + $10 \cdot F$	Informative, assuming deskewing of all lanes of the multipoint link in the frame clock domain

* Skew unit SU= max(UI, 320 ps)

** For system operation it is important that the sum of the clock distribution skew and multipoint link interconnect skew does not exceed the total specified value. One of the two items may be larger than specified, if this is compensated in the other item.

4.12 Skew and misalignment budget (cont'd)

A logic device with multiple lane capability shall always support the budgets with multiple converter devices in Table 10 and Table 11. The following table specifies the skew budgets to be supported by different converter device classes.

Table 12 — Required skew budget compliance per converter device type

Device class (clause 6)	Converter type	Skew budget
NMCDA-SL	any	none
NMCDA-ML	ADC	4.12Skew and misalignment budget (cont'd) Table 8
	DAC	Table 9
MCDA-SL or MCDA-ML	ADC	Table 10
	DAC	Table 11

4.13 Control Interfaces (informative)

The application may require one or more control interfaces to pass information (status, control, etc.) between the devices mutually and/or between the devices and a higher layer application level. The specification of these interfaces is outside the scope of this standard. It is expected that such status and control interfaces will be based on dedicated standards for this purpose, e.g., SPI or JTAG. The control interfaces will generally be sequenced by dedicated clocks, which are generally not the same as the device clock.

5 Data stream

5.1 Transport layer

5.1.1 Overview

The transport layer maps the conversion samples to non-scrambled octets. JESD204 provides several options for this mapping:

- A single converter to a single-lane link
- Multiple converters in the same device to a single-lane link
- A single converter to a multi-lane link
- Multiple converters in the same device to a multi-lane link

In addition, it is possible to combine the interfaces of multiple converter devices on a multipoint link. However, this is not a special mapping case. The transport layer mapping applies only to the samples of a single converter device, i.e. transmitted over a single link.

A set of samples and/or partial samples is grouped into a frame of *F* octets. In many applications the frame clock will have the same frequency as the sample clock. However, JESD204 allows more than one sample per converter to be transmitted per frame cycle. The number *S* of samples per converter per frame cycle must always be an integer. This is necessary in order to minimize cross-talk between the SERDES circuits and sensitive analog parts. Each sample is transmitted as a group of *N'* bits, consisting of *N* data bits, optional control bits and optional tail bits. Additional tail bits at the end of the frame may be necessary to fill a whole number of octets per lane per frame cycle.

The user data mapping is parameterized according to the descriptions in the next subclauses. An overview of the mapping parameters and their supported ranges is presented in Table 20 in 8.3.

5.1.2 User data format for an independent lane

This subclause specifies the mapping of samples from one or more converters in the same device to octets for transmission over a single lane. This specification is backward-compatible with JESD204A. However, the position of tail bits may differ from JESD204 version April 2006.

5.1.2.1 User data mapping without oversampling

The mapping without oversampling is specified with reference to Figure 17. One device contains *M* converters, each producing *N* data bits per sample. The numbering scheme for all items in the picture starts from 0. Within a sample, the leftmost bit is the most significant bit (MSB) and the rightmost bit is the least significant bit (LSB). The process of mapping the samples to octets is described in the following steps:

1. The samples are mapped to a linear axis, starting with converter 0, followed by converter 1, etc. until all samples have been mapped.
2. The samples are mapped to words. When the samples contain no control bits (e.g., overrange indication), the words are identical to the samples. When sample-specific control bits are available, there are two options:
   a. A conversion word is formed by appending the relevant control bits after the LSB of each conversion sample.
   b. A conversion word is identical to the corresponding sample. The control bits are grouped into a separate control word that is appended after the words containing the samples. The first bit(s) of the control word correspond(s) to the control bit(s) of converter 0, the next bit(s) in the control word correspond(s) to the control bits of converter 1, etc.

      If *CF* is the number of control words in the frame, with *CF*=0 or *CF*=1 for an independent lane, the total number of words transmitted per frame cycle is thus *M*+*CF*.
3. Words not containing a whole multiple of 4 bits are extended to the smallest possible nibble group (group of half octets) using tail bits. The extended words are indicated by "NG" in Figure 17. This step is optional and may be disregarded in cases where highest line efficiency is prioritized against easier reconfigurability in mapping. A conversion word may thus be extended by control and/or tail bits to a length of $N' \geq N$ bits, where N' is a whole multiple of 4. Note that for *CF*=0, control bits are considered part of the data word and there will be no tail bits between data and control bits, but one or more tail bits could be necessary after the control bit(s). For *CF*=1, data and control bits are in different words and one or more tail bits could be necessary after the data bits of each sample
4. If necessary, tail bits are appended to make the total number of bits after the last step a whole multiple of 8.
5. The sequence obtained in the previous step is regrouped into *F* octets.

5.1.2.1 User data mapping without oversampling (cont'd)

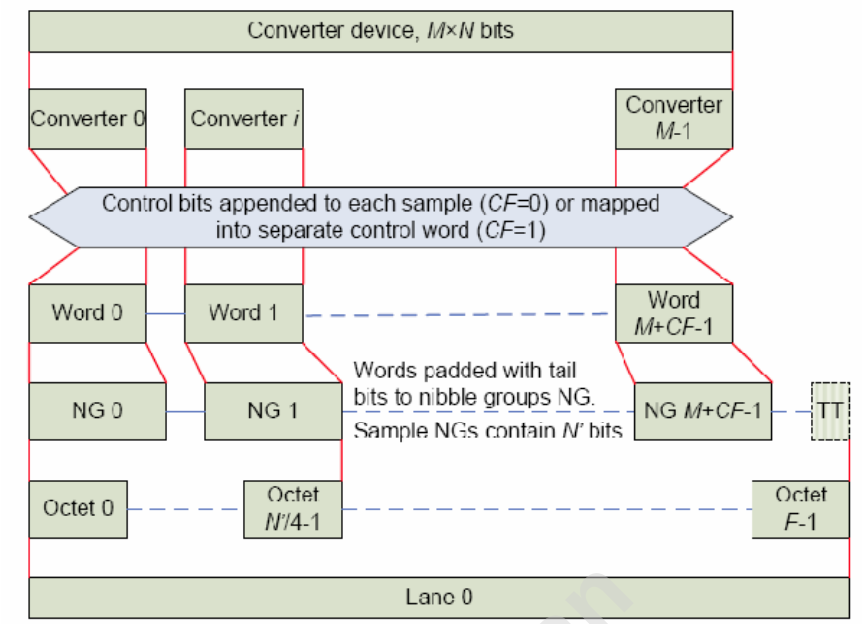


Figure 17 — User data format for an independent lane without oversampling.

EXAMPLE Difference in tail bit position between JESD204 version 2006 and newer versions

Figure 18 illustrates the difference in the recommended location of tail bits in version 2006 of JESD204 and later versions. In version 2006, tail bits were only inserted at the end of a frame. The newer versions recommend that tail bits be inserted at the end of each sample, if needed to fill a nibble group. In this example the new mapping method needs one extra octet per frame.

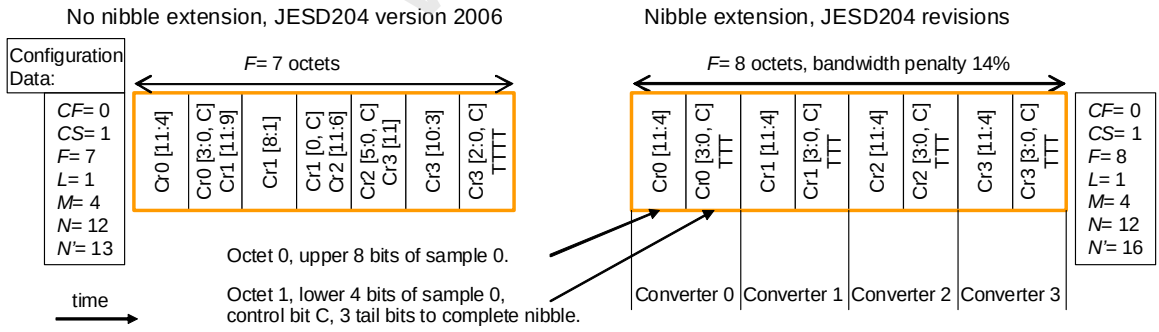


Figure 18 — JESD204 version 2006 and newer versions compared for a 4×12-bit converter with control bit over a single lane. No control words.

5.1.2.1 User data mapping without oversampling (cont'd)

EXAMPLE Compacting the frame structure by using a control word

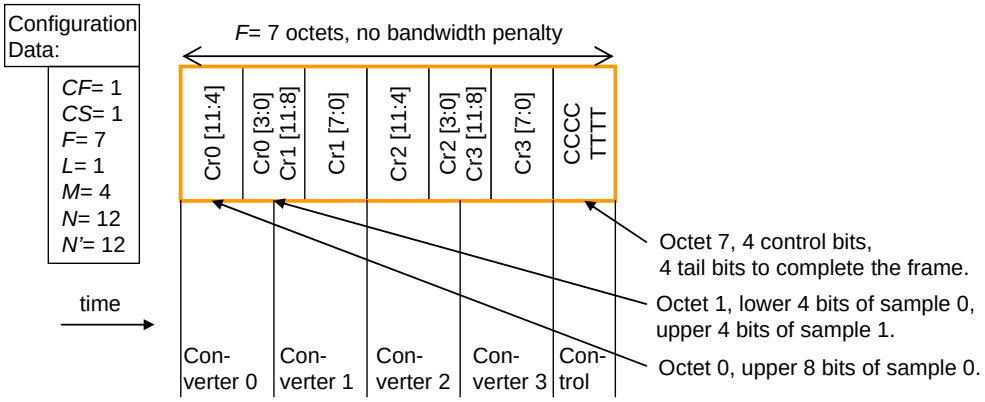


Figure 19 illustrates how grouping the control bits into a control word can reduce the amount of tail bits and enable a shorter frame.

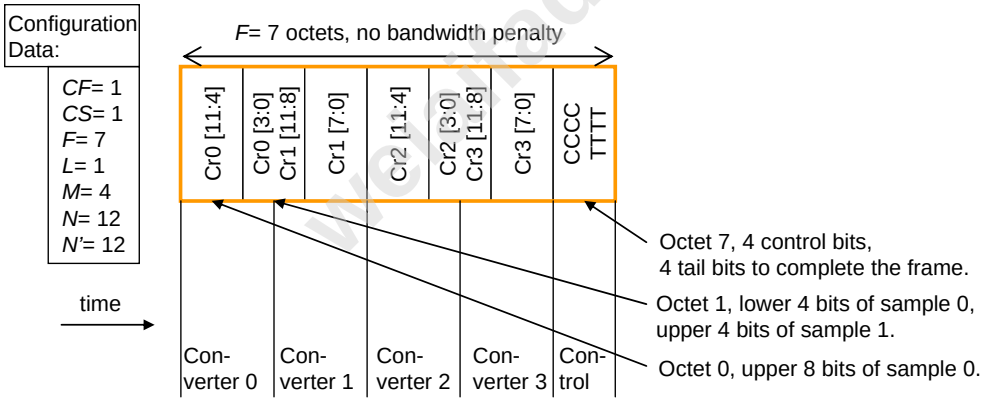


Figure 19 — User data format with control word. Example for 4×12-bit converter.

5.1.2.2 User data mapping with oversampling

The mapping used with oversampling is similar to the mapping without oversampling. The general principle is illustrated in Figure 20. Instead of one sample per converter, S samples are cascaded before mapping the data of the next converter.

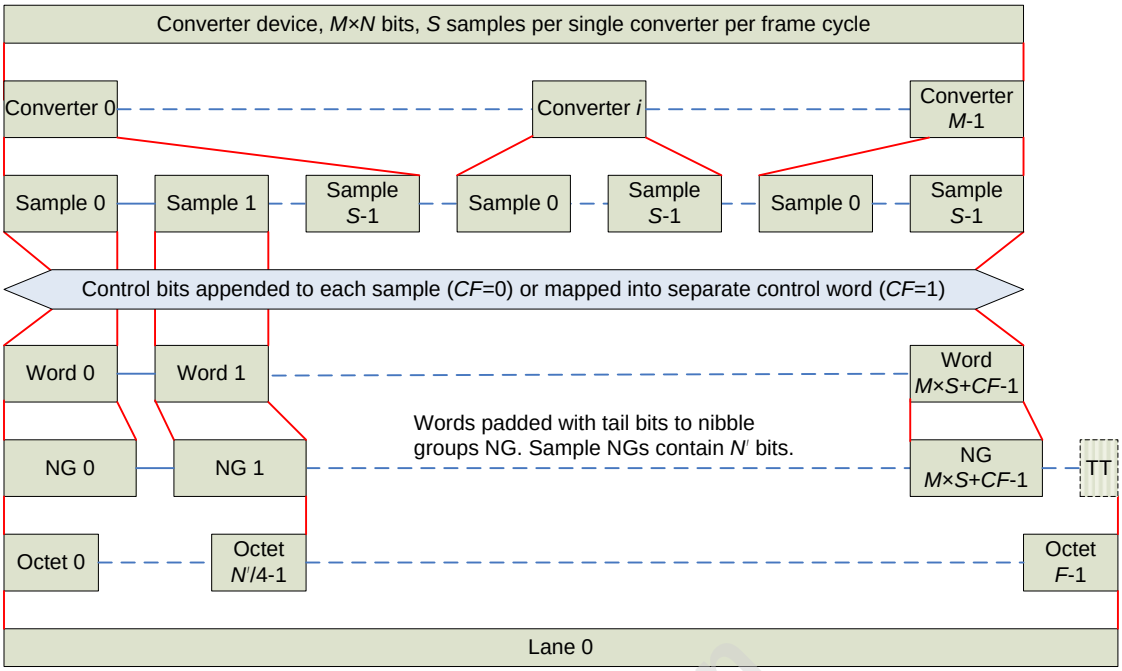


Figure 20 — User data format for an independent lane with oversampling.

5.1.2.2 User data mapping with oversampling (cont'd)

EXAMPLE Mapping for a 4×12-bit converter with two times oversampling and control word

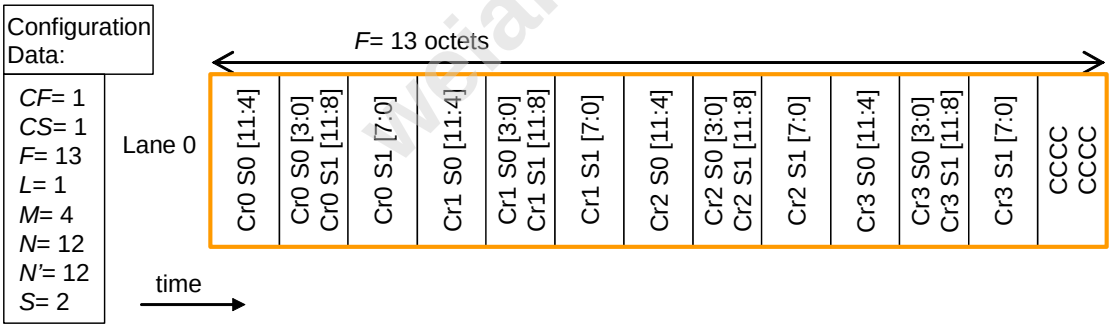


Figure 21 — User data format with oversampling. Example for 4×12-bit converter, 2× oversampling and control word.

5.1.3 User data format for multiple lanes

For a link consisting of L lanes, the same mapping method is used as for a single lane. However, in the last step a row of $L \cdot F$ octets is obtained. The first F octets are transmitted over lane 0, the next F octets over lane 1, etc. The last F octets are transmitted over lane L-1. This is illustrated in

Figure 22.

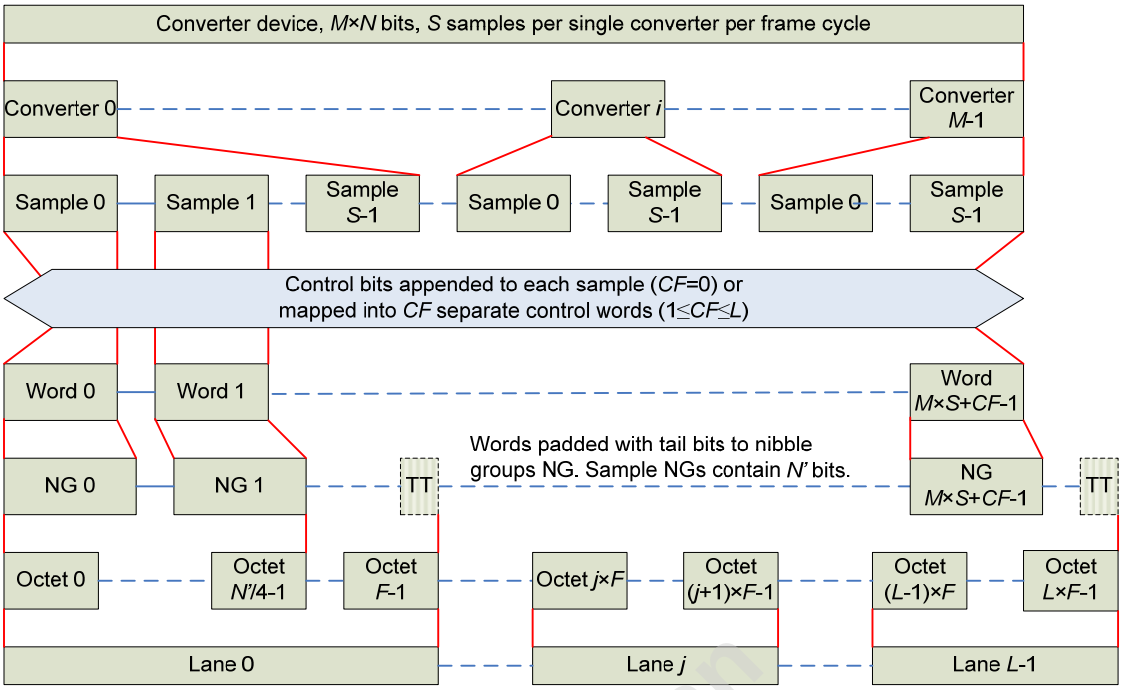


Figure 22 — User data format for multiple lanes.

5.1.3 User data format for multiple lanes

Compared to the independent lane format, the following additional specifications are needed.

1. Parameter *HD* controls whether a sample may be divided over more lanes. In the Low Density mode (*HD*=0), partial conversion words at the end of a group of *F* octets are avoided by adding more tail bits (TT) after the last full nibble group (NG) in the group if necessary. In the High Density mode (*HD*=1), the conversion words may break at the lane boundary.
2. Parameter *CF*, the total number of control words per frame period per link, controls which lanes will carry control words. *CF*=0 means that no control words are used. Other allowed *CF* values are common subdivisors of *L* and *M*. The *L* lanes are divided into *CF* groups of *L*/*CF* lanes. Each group of lanes transmits the samples of *M*/*CF* converters. After these samples a control word is inserted, containing in successive order the control bits belonging to these samples. If the control word fits on a single lane, it is not allowed to break it over a lane boundary.

EXAMPLE Mapping of a 16×(11+2)-bit converter with and without control words

Figure 23 shows the mapping of a 16×11-bit converter with two control bits per sample without using control words. Note that the control bits extend each 11-bit sample to a 13-bit word, that is further extended to four nibbles by means of tail bits. In total 32 octets must be transmitted during one frame period, e.g., using 8 lanes of 4 octets per frame.

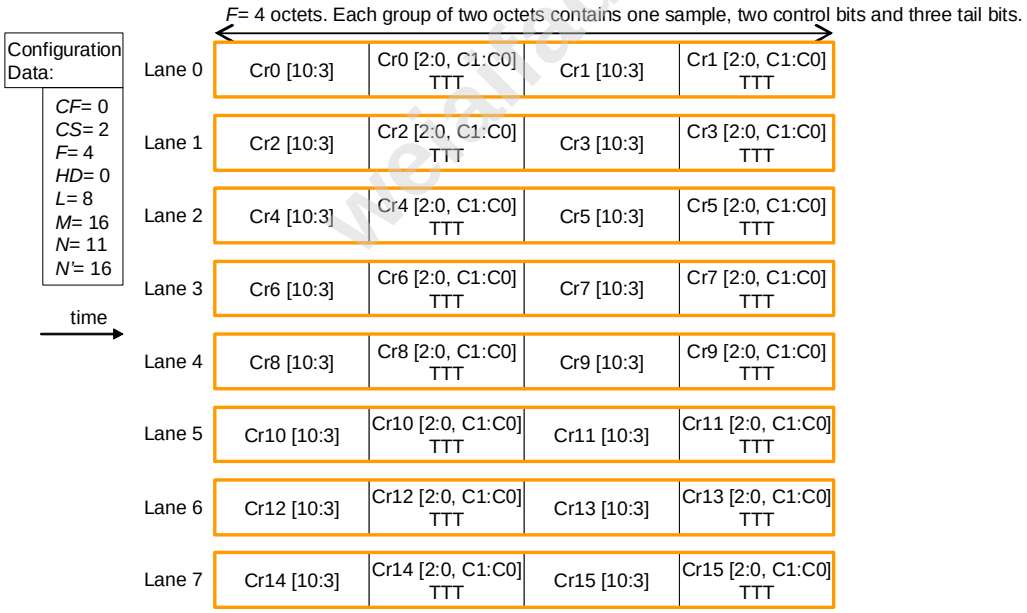


Figure 23 — Multilane user data format for 16×(11+2)-bit converter without control word.

Figure 24 shows how the total amount of octets to transmit per frame period can be reduced to 28, by grouping all control bits into one control word, which is transmitted on the last lane. This way one lane can be saved. However, if only a fraction of the converters is active, there may be no saving in the amount of lanes that need to be powered on. For instance, to transmit only the data of converter 2, it is enough to activate lane 1 in the mapping of Figure 23, while lanes 0, 1 and 6 must be active in the mapping of Figure 24.

5.1.3 User data format for multiple lanes (cont'd)

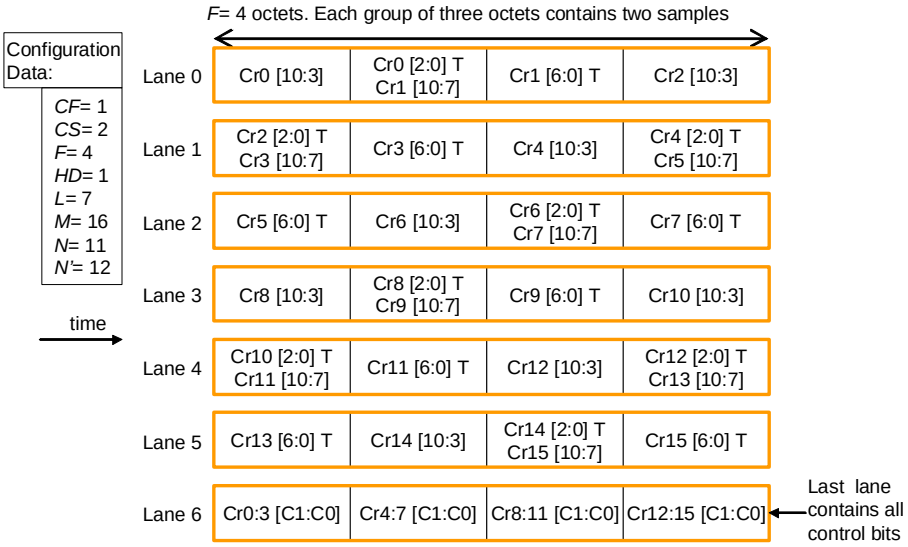


Figure 24 — Multilane user data format for 16×(11+2)-bit converter with single control word.

Figure 25 shows the mapping of the same converter when using two control words. No octets are saved in this configuration. There would be room for an extra sample to transmit on lane 3, but that is prohibited by the requirement to transmit the data of an equal amount of converters in each group.

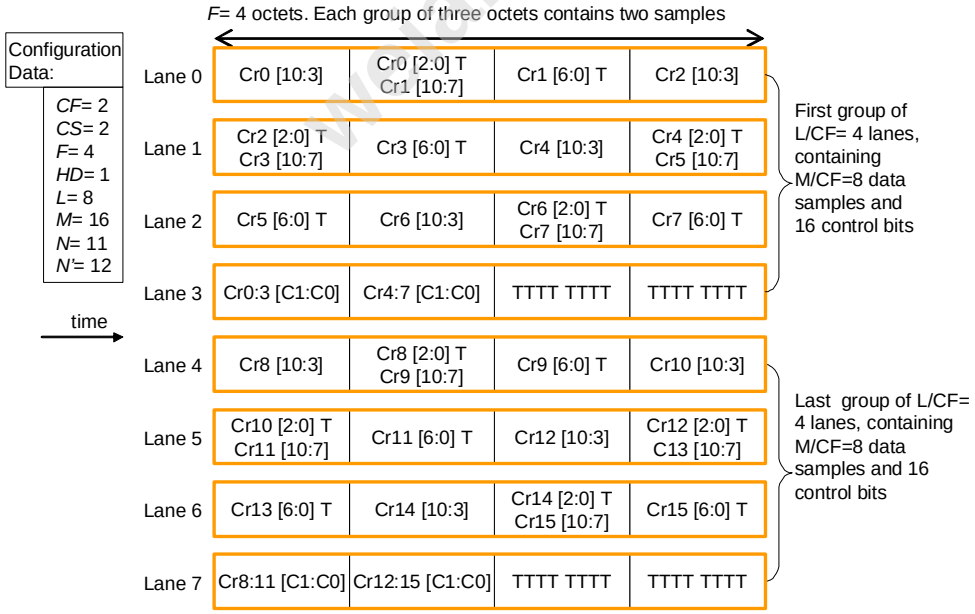


Figure 25— Multilane user data format for 16×(11+2)-bit converter with two control words.

5.1.4 Tail bits

The tail bits (T) are specified for the unscrambled frame and are fed through the scrambler with the data bits if the scrambler is enabled (see subclause 5.2). To avoid situations in which the tail bits prevent or significantly reduce the generation of frame synchronization symbols, they shall be compliant with one of the following requirements:

- either the sequence of tail bits is the same from frame to frame, or
- the sequence is generated by a pseudo-random generator based on a polynomial of a degree of at least 9.

It should be noted that constant tail bits can potentially cause spurious spectral lines if scrambling is not used.

5.1.5 Idle mode

5.1.5.1 General

An idle mode is a state in which one or more converters that are connected to the same link are inactive, but the interface is kept alive and the frame structure is not changed.

In systems having multiple converters per link, it is possible that a converter is sharing parts of all its octets with other converters. Therefore an inactive converter cannot be marked on the data link layer, e.g., by a control symbol in the 8B10B code. Instead, a sample specific control bit could be used for this purpose. It is also possible to pass information on inactive converters via a control interface (see 4.13).

5.1.5.2 Dummy samples

The samples of an inactive converter are replaced by dummy samples. There is no other requirement for the dummy samples than that they shall not prevent the generation of alignment characters (see 5.3.3.4) irrespective of whether scrambling (see 5.2) is enabled or not.

Dummy samples might be generated in the application layer, where it may not be known which bits in a dummy sample will be mapped to the last octet of a frame. Therefore, to avoid the possibility of interference with alignment character generation, it is recommended that all dummy bits comply with the same requirement as the tail bits (see 5.1.4). Pseudo random bits are a good choice because they will avoid peaks in the transmitted spectrum when scrambling is disabled. Another option is to replace the samples of an inactive converter by a transport layer test sequence (see 5.1.6). However, if control bits are toggled as part of the test sequence a possible control bit for marking the inactive converter shall remain constant throughout the test sequence.

5.1.6 Test modes (transport layer)

5.1.6.1 General

A *transport layer* test mode is a state where the *data samples* from or to all converters connected to the same link are replaced by predetermined *test samples*. If the user data contains control bits, the control bits are replaced by test control bits. The test samples and control bits are mapped according to the current user data format and scrambled if scrambling is enabled. A JESD204 device is put into a test mode via a control interface, see 4.13.

NOTE Generation and detection of test samples is carried out in the application layer. The JESD204 link itself does not need a special mode for transport layer testing.

JESD204 specifies a long transport layer test pattern with a periodicity of multiple frames as well as a short transport layer test pattern with a periodicity of a single frame period. The use of other test patterns is optional. In general, test samples for transport layer testing shall comply with the following requirements:

- The pattern of test samples and possible control bits shall be repetitive. The periodicity shall be a whole number of frame periods, at the shortest one frame period.
- The pattern shall be such that the receiver is able to find the boundaries between successive periods of the pattern.

A transport layer test sequence will allow users to verify that the mappings between samples and octets and lanes in transmitter and receiver transport layers match. Although at the converter side the samples are also available in the analog domain, JESD204 devices shall have the possibility for transport layer testing entirely in the digital domain. This allows for testing the functionalities of the link and the data conversion independently from each other. A JESD204 TX shall support a test mode in which it replaces conversion samples and possible control bits by predetermined test samples and control bits. A JESD204 receiver shall support a test mode in which it compares the received test samples and control bits to the expected ones. The receiver shall have a means to communicate the test result to an upper layer, e.g. via a control interface.

5.1.6.2 Short transport layer test pattern

Support for the short test pattern is mandatory for all logic devices and for those converter devices that do not support the long test pattern. The short test pattern has a duration of one frame period and is repeated continuously for the duration of the test. Each sample shall have a unique value that can be identified with the position of the sample in the user data format, see subclauses 5.1.2 and 5.1.3. The sample values shall be such that correct sample values will never be decoded at the receiver if there is a mismatch between the mapping formats being used at the transmitter and receiver devices. This can generally be accomplished by ensuring there are no repeating subpatterns within the stream of samples being transmitted. The converter manufacturer shall specify the test samples transmitted by an ADC device or expected by a DAC device, or shall give the device user the possibility to program these sample values via a control interface.

5.1.6.3 Long transport layer test pattern

Support for the long test pattern is mandatory for all logic devices and for those converter devices that support transmission of control bits over the JESD204 interface. Also the long test pattern is repeated continuously for the duration of the test. For a link connected to *M* converters, with *S* samples transmitted per converter per frame period, the test cycle has a length of max(*M*·*S*+2, 4) frames, rounded up to the lowest number of full multiframes. For converters not supporting multi-lane alignment (i.e. converters belonging to the NMCDA-SL class, see clause 9), the length of the test cycle is rounded up to the lowest whole multiple of 4 frames. With frame number *i* in the test cycle starting from 0, the data in the sequence is defined as follows:

- Frame 0: Each sample is the binary representation of the converter identification number *CID*+1 mod 2^N, with *CID* running from 0 to *M-1* (see Figure 20)
- Frame 1: Each sample is the binary representation of the sample identification number *SID*+1 mod 2^N, with *SID* running from 0 to *S-1* (see Figure 20)
- Frame $i \geq 2$: Each sample has the highest bit set to "1" and all other bits are set to "0".
- Frame $0 \leq i \leq M \cdot S-1$: If the converter device supports control bits, the highest control bit belonging to sample mod(i,S) of converter floor(i/S) is set to "1", while the remaining control bits are set to "0".
- Frame $i \geq \max(M \cdot S,2)$: Data bits like $i \geq 2$, all control bits set to "0"
- If tail bits are used, they shall be the same from frame to frame.

An example for a 2× oversampling 2×14-bit converter with one control bit is given in Figure 26.

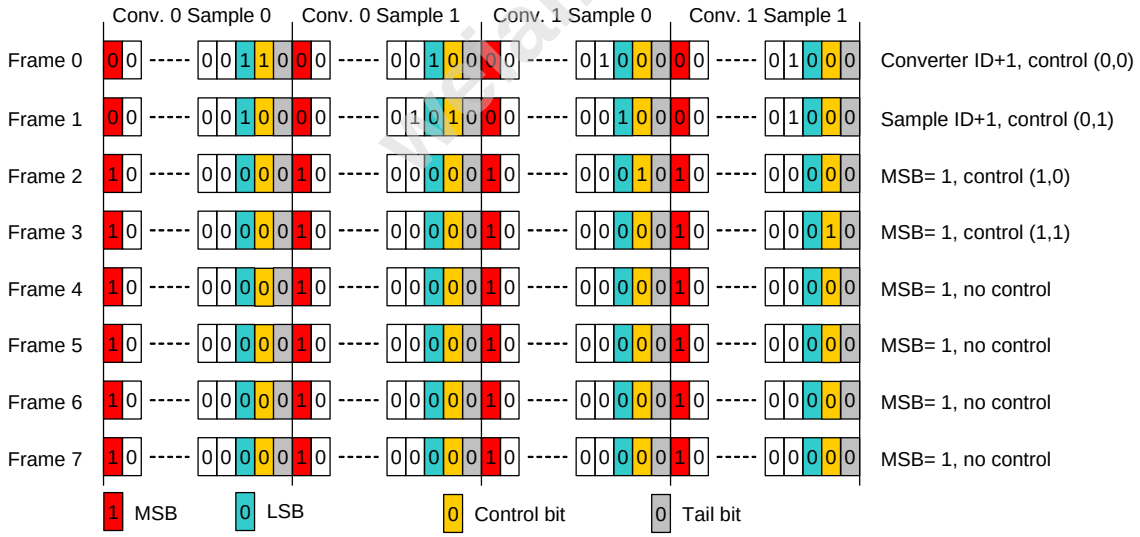


Figure 26 — Example of transport layer test sequence for a two times oversampling dual 14-bits converter with one control bit.

5.2 Scrambling

Although it is not mandatory to enable scrambling, JESD204 TX and RX devices shall support scrambled data octets, as shown below. The scramblers and descramblers are 1 per lane. Functionally they are located between the transport layer and the link layer, as illustrated in Figure 27. Enabling scrambling/descrambling for a link involves activating the individual scramblers/descramblers on each lane belonging to the link. Mixed mode operation where only some lanes in a link contain scrambled data is not permitted.

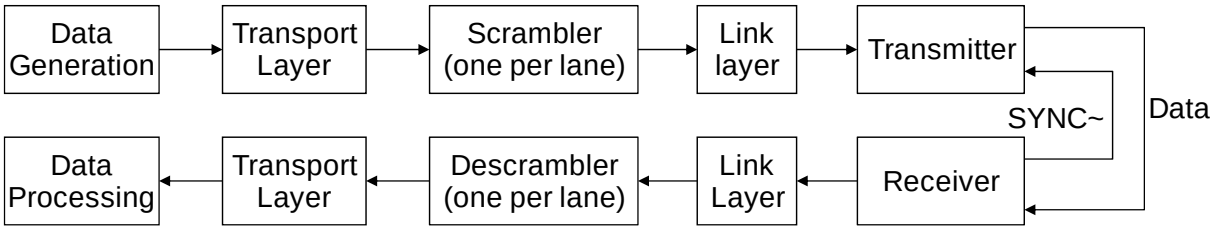


Figure 27 — Functional location of scrambler and descrambler

The main purpose of scrambling is to avoid the spectral peaks that would be produced when the same data octet repeats from frame to frame. Spectral peaks can cause electromagnetic compatibility or interference problems in sensitive applications. Via aliasing, they also cause code-dependent DC offsets in the data converters. Another advantage of scrambling is that it makes the spectrum data-independent, so that possible frequency-selective effects on the electrical interface will not cause data-dependent errors. However, all digital operations in converters (including scrambling) cause some amount of switching noise, so there may be applications where it is of advantage to disable the scrambling.

5.2.1 Scrambler polynomial

The scrambler polynomial shall be

$$1 + x^{14} + x^{15}$$

The period of this polynomial is long enough (32,767 bits) to meet the spectral requirements of sensitive radio applications while allowing the descrambler to self-synchronize in two octets.

5.2.2 Scrambler bit order

The scrambler and descrambler are defined via their serial implementations, processing the transmitted / received data frame by frame. The leftmost bit of the frame is shifted in first, as illustrated in Figure 28. The actual implementation shall produce the same result as the serial definition.

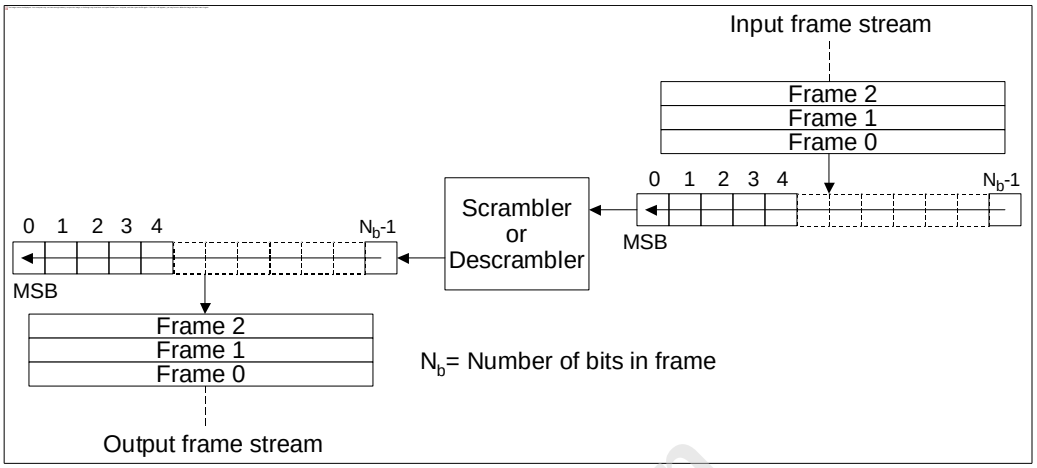
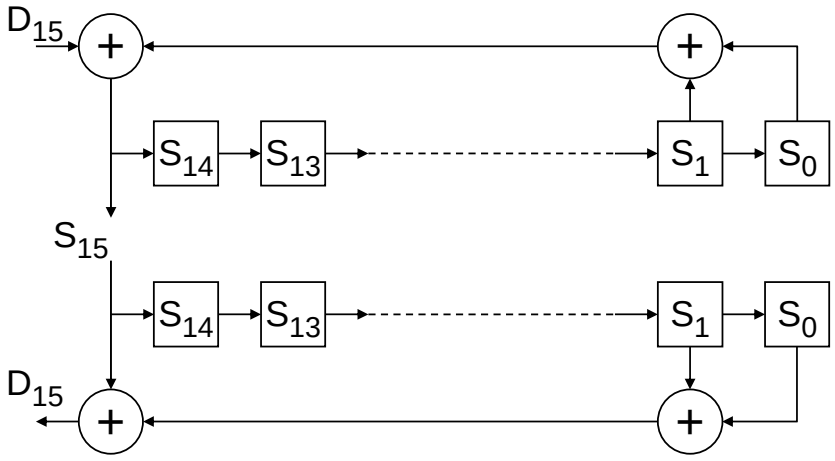


Figure 28 — Serial scrambling

5.2.3 Scrambler type

The scrambler shall be of the self-synchronous type. The serial implementation and the equations for parallel implementation are shown in Figure 29. An alternative scrambler is defined in 5.2.4 and Figure 30. Parallel implementation examples for scramblers and descramblers are shown in annex D.

5.2.3 Scrambler type (cont'd)



$D = D_0, D_1, D_2, \cdots = unscrambled\ data\ in \quad (MSB\ first)$

$S = S_0, S_1, S_2, \cdots = scrambler\ state = scrambled\ data\ out$

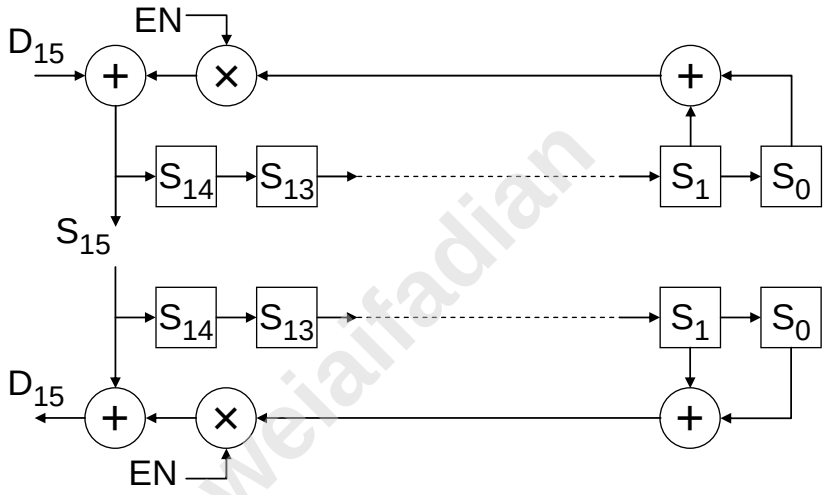
$serial\ update: S_{15} = D_{15} + S_1 + S_0\ ;\ D_{15} = S_{15} + S_1 + S_0$

$$parallel\ update: \begin{cases} S_{31} = D_{31} + S_{17} + S_{16}\ ;\ D_{31} = S_{31} + S_{17} + S_{16} & (LSB \cdot 16b) \\ S_{30} = D_{30} + S_{16} + S_{15}\ ;\ D_{30} = S_{30} + S_{16} + S_{15} & \\ \ldots & \\ S_{23} = D_{23} + S_9 + S_8\ ;\ D_{23} = S_{23} + S_9 + S_8 & (LSB \cdot 8b) \\ S_{22} = D_{22} + S_8 + S_7\ ;\ D_{22} = S_{22} + S_8 + S_7 & \\ \ldots & \\ S_{17} = D_{17} + S_3 + S_2\ ;\ D_{17} = S_{17} + S_3 + S_2 & \\ S_{16} = D_{16} + S_2 + S_1\ ;\ D_{16} = S_{16} + S_2 + S_1 & (MSB) \end{cases}$$

Figure 29 — Serial scrambler and descrambler implementation and equations for parallel implementation.

5.2.4 Early synchronization option

If used, scrambling is enabled at the start of user data. A scrambler implemented according to Figure 29 would be bypassed during code group synchronization and transmission of an initial lane alignment sequence. After enabling the scrambler, two octets must be received before the state registers in the scrambler and descrambler have synchronized and the descrambler starts producing correct data. In order to avoid loss of user data at start up, an alternative scrambler may be implemented, in which unscrambled octets are also flowing through the state registers. The selection between scrambled and unscrambled operation is made using an enable signal to the scrambler logic. At the receiver, the descrambler input can always be connected to the 8B/10B decoder output, while the selection between original and descrambled data is made at the descrambler output. The serial implementation and the equations for parallel implementation for the modified scrambler are shown in Figure 30. A parallel implementation example for the alternative scrambler is shown in Figure D.3.



$D = D_0, D_1, D_2, \cdots = unscrambled\ data\ in \quad (MSB\ first)$

$S = S_0, S_1, S_2, \cdots = scrambler\ state = scrambled\ data\ out$

$EN = Enable\ scrambling\ /\ descrambling$

$$serial\ update: S_{15} = D_{15} + EN \cdot (S_1 + S_0);\ D_{15} = S_{15} + EN \cdot (S_1 + S_0)$$

$$parallel\ update: \begin{cases} S_{31} = D_{31} + EN \cdot (S_{17} + S_{16});\ D_{31} = S_{31} + EN \cdot (S_{17} + S_{16}) \quad (LSB \cdot 16b) \\ S_{30} = D_{30} + EN \cdot (S_{16} + S_{15});\ D_{30} = S_{30} + EN \cdot (S_{16} + S_{15}) \\ \cdots \\ S_{23} = D_{23} + EN \cdot (S_9 + S_8);\ D_{23} = S_{23} + EN \cdot (S_9 + S_8) \quad (LSB \cdot 8b) \\ S_{22} = D_{22} + EN \cdot (S_8 + S_7);\ D_{22} = S_{22} + EN \cdot (S_8 + S_7) \\ \cdots \\ S_{17} = D_{17} + EN \cdot (S_3 + S_2);\ D_{17} = S_{17} + EN \cdot (S_3 + S_2) \\ S_{16} = D_{16} + EN \cdot (S_2 + S_1);\ D_{16} = S_{16} + EN \cdot (S_2 + S_1) \quad (MSB) \end{cases}$$

Figure 30 — Alternative serial scrambler and descrambler implementation and equations for parallel implementation.

5.2.5 Initial state

The scramblers described in 5.2.3 and 5.2.4, as well as many other commonly used self-synchronous scramblers, will produce repetitive output when the input data is a repeating copy of the initial state. Such repetitive output causes peaks in the spectral domain which could lead to electromagnetic interference (EMI). In order to minimize the occurrence of repetitive output, the scrambler must be initialized to a state that is unlikely to repeat continuously in the octet data produced by the transport layer. The recommended initial state is "1" for the eight storage elements with the highest indices and "0" for the seven remaining ones. No preset is needed in the descrambler, because it is self-synchronized. In the alternative scrambler defined in 5.2.4 there is also no need for preset, because the initial state at the start of scrambling will be determined by the last two unscrambled octets.

5.2.6 Scrambling disable

In some applications the disadvantages of scrambling outweigh the benefits. In such cases, the converter manufacturer may at his discretion provide a means to disable the (de)scrambler. Consequently the logic device shall have the option of disabling the (de)scrambling.

5.3 Data Link Layer

5.3.1 8B/10B encoding

8B/10B coding is used to encode data before being transmitted. The 8B/10B coding is widely used in industry and has proven robustness. The 8B/10B codes have the following properties:

- Sufficient bit transition density (3 to 8 transitions per 10-bit symbol) to allow clock recovery by the receiver.
- Control symbols that are used:
- to establish receiver synchronization to the 10-bit symbol boundaries,
- to mark the start and end of frames or other sequences of data, and
- to enable alignment between serial lanes.
- Balance (can be AC-coupled).
- Detection of single bit errors.

For a complete description of 8B/10B coding and decoding refer to reference 1, clause 36.2.

5.3.2 Transmission order

The frame contents are processed from left to right, i.e., from MSB to LSB. After serialization the leftmost bit of the 8B/10B code group, i.e., bit "a" (reference 1, clause 36.2.4) is transmitted first.

5.3.3 Link operation

5.3.3.1 Code group synchronization

Code group synchronization is achieved by the following process. Although described for multiple receivers and transmitters, the same process is applicable to a single receiver and transmitter:

- The receivers issue a synchronization request via the synchronization interface.
- The transmitters emit a stream of /K/= /K28.5/ symbols.
- The receivers synchronize, and then wait for the correct reception of at least four consecutive /K/symbols.
- The receivers deactivate the synchronization request in accordance with the guidelines outlined in 7.1.

The next steps in the process are dependent on the Deterministic Latency Subclass of the transmitter device.

Subclass 0 transmitters shall implement the following behaviour:

- Upon detecting that all receivers have deactivated their synchronization requests, the transmitters continue emitting /K/ symbols until the start of the next frame.
- From the start of the next frame, the transmitters emit either an initial lane alignment sequence or encoded user data.

For Subclass 1 or 2 transmitters:

- Upon detecting that all receivers have deactivated their synchronization requests, the transmitters continue emitting /K/ symbols until the next LMFC boundary. (Default operation should be to use the next LMFC boundary, but devices may optionally allow a programmable option of using a later LMFC boundary)
- On the first frame following the chosen LMFC boundary, the transmitters emit an initial lane alignment sequence.

The synchronization process is shown in the flow diagrams of the following two figures. The ‘SYNC’ transitions indicate a change of state in the SYNC~ signal generated by the RX. The ‘DATA’ transitions indicate a change of state in the data generated by the TX.

5.3.3.1 Code group synchronization (cont'd)

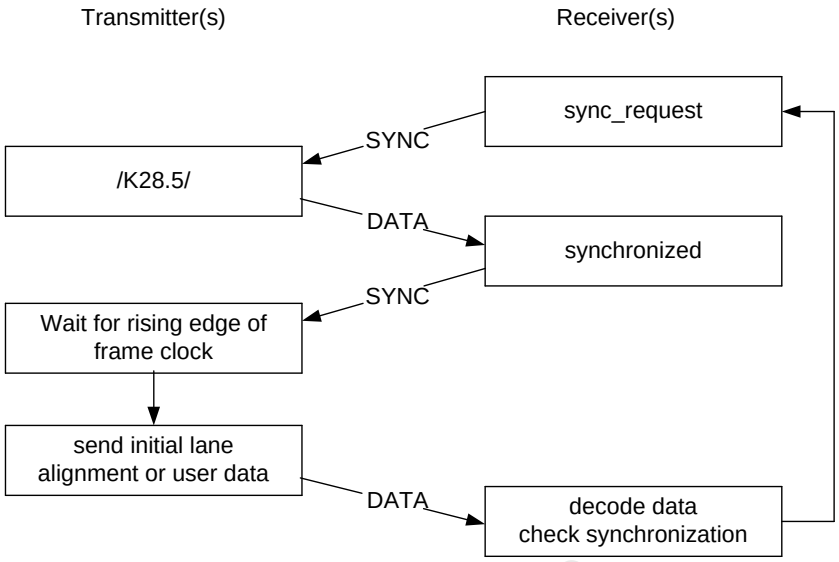


Figure 31 — Synchronization process for Subclass 0

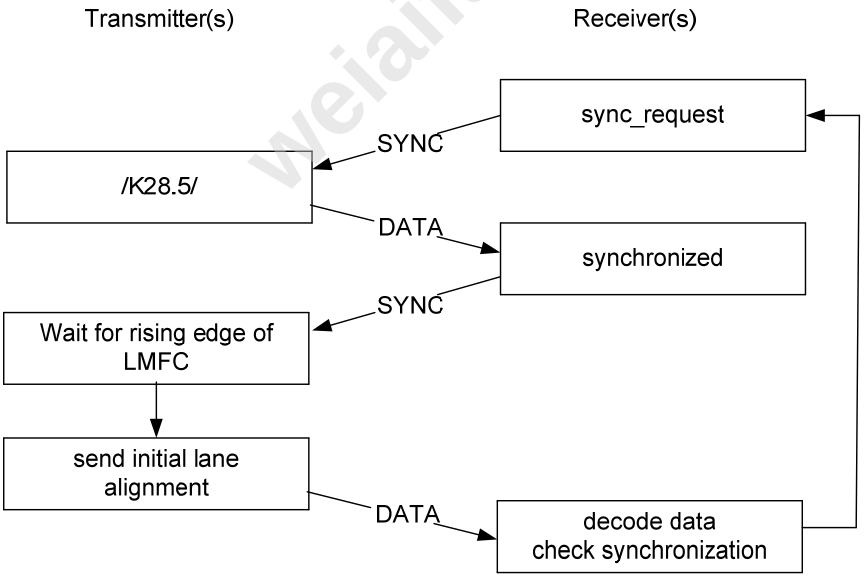


Figure 32 — Synchronization process for Subclass 1 or 2

5.3.3.2 SYNC~ signal combining

With multiple lanes, the synchronization requests of all receivers belonging to the same link are combined into one signal and presented simultaneously to the transmitter device. On multipoint links it is permitted, but not mandatory to combine the synchronization requests of the individual links. The options for SYNC~ signal combining in multipoint links are outlined below:

- Inside a receiver logic device, the SYNC~ signals from each link in the logic device shall either be combined and distributed to all ADCs, or distributed to each ADC as individual per-link SYNC~ signals.
- Inside a transmit logic device, the SYNC~ signals from all DAC devices may either be first decoded and then combined in the transmit logic device, or treated as individual per-link SYNC~ signals.

Figure 33 provides an example of SYNC~ signal combination. Figure 34 provides an example of non-combined SYNC~ signaling.

When using SYNC~ signal combining, as long as a single receiver requests code group synchronization, all transmitters connected to the multipoint link will send /K28.5/ symbols.

When not using SYNC~ signal combining, only the specific link requesting code group synchronization will be affected.

For multipoint links, if deterministic latency is not implemented (namely, JESD204A and subclass 0 applications), SYNC~ signal combining must be used to ensure that transmitter ILA generation is properly aligned across all links.

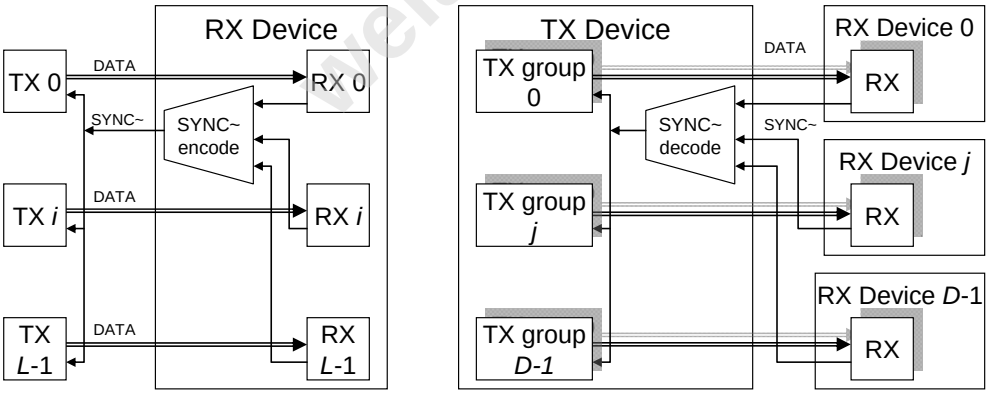


Figure 33 — Examples of SYNC~ signal combination

5.3.3.2 SYNC~ signal combining (cont'd)

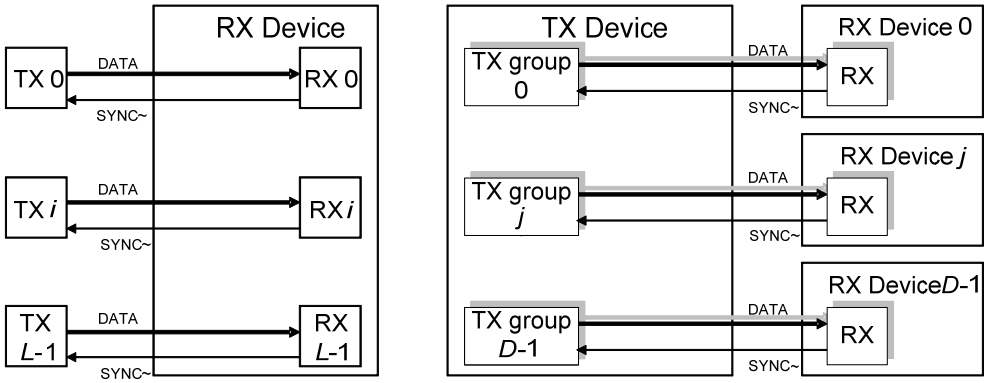


Figure 34 — Examples of non-combined SYNC~ signaling

5.3.3.3 Initial frame synchronization

At link startup, frame synchronization is achieved in the following way:

- During code group synchronization, the transmitter always sends full frames of /K28.5/ comma symbols.
- After code group synchronization, the receiver assumes that the first non-/K28.5/ symbol marks the start of a frame. If the transmitter emits an initial lane alignment sequence (see 5.3.3.5), the first non-/K28.5/ symbol will always be /K28.0/.
- The receiver assumes that a new frame starts every *F* octets.

5.3.3.4 Frame alignment monitoring and correction

5.3.3.4.1Alignment characters

Frame alignment is monitored via alignment characters, which are inserted by the transmitter under certain conditions at the end of a frame. The receiver resynchronizes its frame to the alignment characters after checking that their reception is not likely to have been caused by a bit error on the lane. In general, resynchronization will require repeated reception of a valid alignment character at the same unexpected position. However, if loss of frame alignment is the likely result of a recent lane realignment (this could occur in some receiver implementations e.g., during initial lane alignment, see 5.3.3.4.4), there is no need to wait for repetition of an alignment character at the same position.

The alignment character shall be a frame alignment character /F/= /K28.7/. However, if both sides of the lane support lane synchronization, the, lane alignment character /A/= /K28.3/ shall be used in the last frame of a multiframe. Multiframes are defined in 5.3.3.5. Note that /F/ is encoded from octet value 0xFC and /A/ from octet value 0x7C.

The character replacement depends on whether scrambling has been enabled or disabled and on whether lane synchronization is supported. Lane synchronization is required for all device classes except NMCDA-SL (see clause 9).

5.3.3.4.2Character replacement without scrambling

If both sides of the lane support lane synchronization, character replacement in the transmitter and receiver during transmission of data from the transport layer (see 5.1) shall be as follows:

- When the last octet in the current frame, not coinciding with the end of a multiframe, equals the last octet in the previous frame, the transmitter shall replace the current last octet and encode it as control character /F/= /K28.7/. However, if an alignment character was already transmitted in the previous frame, the original octet shall be encoded.
- When the last octet in the current frame at the end of a multiframe equals the last octet in the previous frame, the transmitter shall replace the current last octet and encode it as control character /A/= /K28.3/, even if a control character was already transmitted in the previous frame.
- Upon receiving an /F/ or /A/ symbol, the receiver shall replace it with the value of the octet decoded or used at the same position in the previous frame.

If at least one side of the lane does not support lane synchronization (i.e., for NMCDA-SL class devices, see clause 9), character replacement in the transmitter and receiver during transmission of data from the transport layer (see 5.1) shall be as follows:

- When the last octet in the current frame equals the last octet in the previous frame, the transmitter shall replace the current last octet with /K28.7/. However, if a /K28.7/ symbol was already transmitted in the previous frame, the actual octet shall be transmitted
- Upon receiving a /K28.7/ symbol, the receiver shall replace it with the value of the data octet decoded at the same position in the previous frame

NOTE The "last octet in a frame or multiframe" means the last octet in the frame or multiframe transmitted on a given lane, hence the character replacement functions in each lane are independent.

5.3.3.4.3Character replacement with scrambling

If both sides of the lane support lane synchronization, character replacement in the transmitter and receiver during transmission of data from the transport layer (see 5.1) shall be as follows:

- When the last scrambled octet in a frame, but not at the end of a multiframe, equals 0xFC, the transmitter shall encode it as a control character /F/.
- When the last scrambled octet in a multiframe equals 0x7C, the transmitter shall encode it as a control character /A/.
- Upon receiving an /F/ or /A/ symbol, the receiver shall input the corresponding data octet 0xFC or 0x7C to the descrambler.

If at least one side of the lane does not support lane synchronization (i.e., for NMCDA-SL class devices, see clause 9), character replacement in the transmitter and receiver during transmission of data from the transport layer (see 5.1) shall be as follows:

- When the last scrambled octet in the current frame equals D28.7, the transmitter shall replace it with /K28.7/
- Upon receiving a /K28.7/ symbol, the receiver shall input D28.7 to the descrambler

NOTE The "last octet in a frame or multiframe" means the last octet in the frame multiframe transmitted on a given lane, hence the character replacement functions in each lane are independent.

5.3.3.4.4Frame alignment correction in the RX

When enabled, the alignment correction shall be carried out in the following way:

- If two successive valid alignment characters are detected at the same position, other than the assumed end of the frame, without receiving a (valid or invalid) alignment character at the expected position between the two alignment characters, the receiver will realign its frame to the position of the received alignment characters.
- However, after a lane realignment that could have resulted in a frame alignment error, the receiver will realign its frame to the first received valid alignment character. See NOTE 1.
- The receiver shall have an option to disable the resynchronization described in the previous items in this list, because without scrambling, certain types of periodic data may not produce enough alignment characters for reliable detection of frame misalignment. Another reason for disabling the resynchronization process could be an implementation where frame re-alignment could cause a lane alignment or latency error. See NOTE 3.

NOTE 1 If frame alignment is carried out in a separate buffer after lane alignment, a re-alignment by a non-integer number of frames in the lane buffer will need an additional re-alignment in the frame buffer to avoid a shift of the frame boundary.

NOTE 2 If frame alignment is carried out in a separate buffer after lane alignment, lanes and frames are mutually aligned at the output of the lane alignment buffer. As such, the adjustment range of a separate frame alignment buffer should be small enough to ensure that lane alignment is maintained at the output of the frame alignment buffers.

NOTE 3 If the same flexible buffer is used for lane and frame alignment, lane re-alignment will guarantee correct frame alignment, because the lane alignment character doubles as a frame alignment character. However, a frame re-alignment could cause an incorrect lane alignment or link latency, since there could be multiple positions of frame re-alignment possible within the adjustment range of the buffer.

NOTE4 When scrambling is enabled, on the average one out of 256 frames will end in an alignment character. Without scrambling, the frequency of alignment characters depends on the sampled data and the mapping of the samples to the frame. Most practical signals will map to random or quasi random octets at the end of the frame and generate alignment characters with a frequency of about one per 256 frames. However, noise-free periodic signals with a harmonic frequency relation to the sample clock may not generate alignment characters at all for certain values of initial phase. Usually this problem is corrected with a small amount of random noise on the signal. There may however be singular cases in which alignment monitoring cannot be performed reliably in the non-scrambled mode.

5.3.3.5 Initial lane synchronization

The initial lane synchronization is carried out before the start of user payload data. The initial lane synchronization procedure follows the principles of other standards such as the XAUI standard, Ref. 6, clause 48.2. At a well-defined point in time, all the transmitters issue a dedicated lane alignment character /A/= /K28.3/. Due to different lane delays, these alignment characters may be received at different times by the receivers. With the reception of an /A/, each receiver stores subsequent data in a buffer memory and indicates a flag (‘ready’) to the other receivers, indicating that the buffer contains a valid alignment start point. When all receivers have raised their “alignment received” flags, they start propagating the received data to subsequent data processing logic/function at the same point in time, where synchronization is based on a common signal (‘start’).

5.3.3.5 Initial lane synchronization (cont'd)

For JESD204B devices supporting deterministic latency, this 'start' signal is defined in 6.1. The specification also permits this alignment procedure at a specified time after the reception of the alignment character in a master receiver, with the other receivers issuing an error signal if they do not find the alignment character in their own buffer at the same time.

Initial lane synchronization is achieved by means of an initial lane alignment sequence, starting immediately after code group synchronization. An initial lane alignment sequence shall never be scrambled. An initial lane alignment sequence transmitted by an ADC consists of exactly 4 multiframes. An initial lane alignment sequence required by subclass 1 and 2 DACs consists of exactly 4 multiframes as well. Configurations with multiple subclass 0 DAC devices may require additional multiframes to achieve lane alignment. Therefore, in logic devices the length of the initial lane alignment sequence shall be programmable from 4 up to at least 256 multiframes. A multiframe is defined as a group of K successive frames, where K is between 1 and 32 and such that the number of octets per multiframe is between 17 and 1024:

$\text{ceil}(17/F) \leq K \leq \min(32, \text{floor}(1024/F))$

In JESD204 transmitter devices, the value of K shall be programmable. In JESD204 receiver devices, the value of K is recommended to be programmable. JESD204 receiver devices shall clearly specify their requirements or recommendations for the setting of the factor K in the transmitter device(s).

The structure of the initial lane alignment sequence is illustrated in

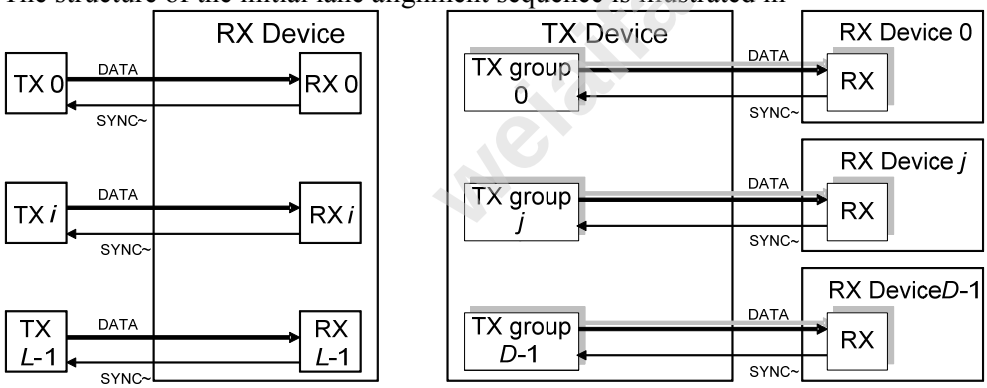


Figure 35. Each multiframe starts with an /R/= /K28.0/ and ends with an /A/. The /R/ is an indication to the receiver that the multiframe is part of an initial lane alignment sequence. The /A/ marks the end of the multiframe and is used for both lane and frame synchronization. The second multiframe shall contain configuration information about the JESD204 link from the transmitter to the receiver, starting from the third symbol. The fixed transmission code of /K28.4/ in the second symbol is an extra confirmation to the receiver that configuration data is going to start. The contents of the initial lane alignment sequence are further specified in 8.2 and 8.3.

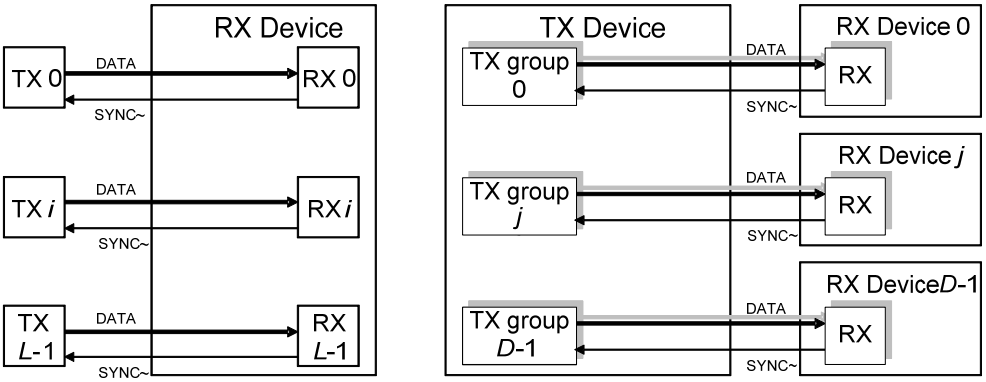


Figure 35 — Initial lane alignment sequence with 4 multiframes

5.3.3.6 Lane alignment monitoring and correction

After initial frame and lane alignment, the channel switches to the alignment monitoring mode. Lane alignment is monitored via /A/= /K28.3/ characters, which are inserted by the transmitter under certain conditions at the end of a multiframe. The insertion is further specified under the frame alignment monitoring, see 5.3.3.4. In general not all lanes will transmit an /A/ simultaneously. However, each receiver separately can check the arrival of an /A/ against a local timing reference and make necessary corrections in alignment.

Each receiver in a multi-lane environment can be authorized from a higher level application layer to perform a dynamic (on-the-fly) re-alignment when the following three conditions are met:

- misalignment has been reliably detected, AND
- a reliable new alignment position has been found, AND
- re-alignment is possible based on the conditions of the data that is maintained in the alignment data buffer.

The authorization of realignment is given to the receiver via the control interface defined in 4.13. The realignment rules for lanes are similar to the rules for frames (see 5.3.3.4.4):

- If two successive, valid /A/ symbols are detected at the same position, other than at the assumed end of the multiframe, without receiving a (valid or invalid) /A/ symbol at the expected position between the two /A/ symbols, the receiver shall realign the lane to the position of the newly received /A/ symbols.
- If a recent frame realignment is the most likely cause of loss of lane alignment, the receiver will realign its lane frame already to the position of the first received /A/ symbol at an unexpected position. See 5.3.3.4.4.

NOTE 1 Dynamic re-alignment will correctly restore the lane delay only if the local timing reference is correct. If lane misalignment has been detected as a result of a phase change in the local timing reference, dynamic re-alignment will ultimately align all lanes to the same reference, but the latency of the link will change. To avoid the possibility of a latency change, subclass 1 and 2 receivers can initiate an LMFC re-alignment via SYSREF or SYNC~ in addition to dynamic re-alignment. Receivers of all subclasses may also request link resynchronization as an alternative to dynamic re-alignment.

NOTE 2 If separate buffers are used for lane and frame alignment, a lane re-alignment can cause the need for an additional frame re-alignment, see 5.3.3.4.4.

5.3.3.7 Link re-initialization

Under certain error conditions, see 7.6.3, a receiver can request re-initialization of the link by asserting a synchronization request. If all receivers are within the same device, all synchronization requests can be made visible to all receivers in the device. This way device-internal synchronization requests can be used to re-initialize the frame and lane synchronization in all receivers in the device.

An alternative method resets the receivers via the transmitters. Because synchronization requests from all receivers on a link and optionally on a multipoint link are combined into a single SYNC~ signal that is presented simultaneously to all transmitters (see 5.3.3.2), all transmitters will start sending /K/= /K28.5/ symbols based on any synchronization request from any receiver.

5.3.3.7 Link re-initialization (cont'd)

Additionally, a transmitter can request re-initialization of the link by moving its state machine to the SYNC state and issuing a stream of /K/ symbols. A receiver belonging to an MDCA device class (see clause 9) shall, upon reception of a stream of /K/ symbols, return to the initial frame and lane alignment states as shown in Figure 45 (in 7.2). In order to guarantee a sufficiently long stream of /K/ symbols, a minimum duration of the synchronization request is defined in 7.6.4.

5.3.3.8 Test modes

5.3.3.8.1 General

A *link layer* test mode is a state where predetermined *8B/10B characters* are transmitted in all frames and on all lanes on the multipoint link. A JESD204 system is put into a test mode via an external control interface, see 4.9.

NOTE Link layer test characters are specified as if injected directly to the 8B/10B encoder. They are never scrambled.

5.3.3.8.2 Test sequences

All TX devices shall be able to transmit the following test sequences.

- A continuous sequence of /D21.5/ characters (high frequency pattern, standard test for random jitter RJ, can also be used for verification of receiver eye mask).
- A continuous sequence of /K28.5/ characters (mixed frequency pattern or code group synchronization, standard test for deterministic jitter DJ).
- Repeated transmission of a lane alignment sequence, with sensitivity to synchronization requests from the receiver. If this test mode is enabled in the transmitter, whenever the receiver issues a synchronization request, the transmitter shall initiate code group synchronization. Upon completion of code group synchronization, the transmitter shall repeatedly transmit the ILA sequence. If there is no active code group synchronization request at the moment the transmitter enters the test mode, the transmitter shall behave as if it received one. When entering the test mode, the transmitter shall transmit at least four /K28.5/ symbols in succession as part of the code group synchronization sequence. This test sequence is not required in devices belonging to the NMCDA-SL class (see clause 9).
- At least one of the following two patterns:
  - A continuous sequence of a modified random pattern (modified RPAT)
  - A continuous sequence of a scrambled jitter pattern (JSPAT)

The 12-octet modified RPAT pattern is defined for instance in INCITS TR-35-2004 (Ref. 9) and the 50-octet JSPAT pattern in INCITS 450-2009 (Ref. 8). Optionally, JESD204 transmitter devices may support other standardized link layer test patterns to be injected before the 8B/10B encoder, such as a the low frequency pattern /K28.7/ or a pseudo-random bit sequence (PRBS).

NOTE 1 RPAT and JSPAT rely on specific running disparity to create the desired test bit pattern. The transmitter will need a means to control the starting running disparity of these patterns.
NOTE 2 A PRBS15-like sequence will be generated in the transport layer tests defined in 5.1.6.2 and 5.1.6.3, when scrambling is enabled.

5.3.3.8.2 Test sequences (cont'd)

All RX devices shall be able to verify the first of the following sequences. The second sequence shall be supported by all devices not belonging to the NMCDA-SL class (see clause 9).

- A continuous sequence of /K28.5/ characters for code group synchronization.

- A code group synchronization sequence, followed by repeated transmission of a lane alignment sequence.

In addition, RX devices shall have the capability to suppress error reports due to a missing ILA. This makes it possible to perform BER measurements using 8B/10B encoded test patterns from a standard pattern generator, after initial synchronization with a /K28.5/ sequence. Most of the bit errors will lead to a running disparity or not-in-table error, which will be indicated by the RX with an error report on the SYNC~ interface. Using the SYNC~ as a bit error indicator will be sufficiently accurate for BER testing. BER compliance testing shall be performed with a repetitive JTSPAT sequence, which is defined for instance in INCITS 450-2009 (Ref. 8).

NOTE The RX is not required to verify other sequences than /K28.5/ and ILA. Errors in other sequences can be detected via the regular error detection functionality in the 8B/10B decoder.

6 Deterministic Latency

6.1 Introduction

Many JESD204 systems contain various data processing elements that are distributed across different clock domains and lead to ambiguous delays through the interface. These ambiguities lead to non-repeatable latencies across the link from power-up to power-up or over link-reestablishments. JESD204A did not provide for a mechanism to make the interface latency deterministic. JESD204B provides two possible mechanisms for this purpose, defined as Subclass 1 and Subclass 2 operation.

The deterministic latency across the link is defined from the parallel frame-based data input on the TX device to the parallel frame-based data output on the RX device, all measured within the frame clock domain (Refer to Figure C.4 and Figure C.7 for reference). This latency across the link shall be programmable in units at least as small as the frame clock period, and shall be repeatable from power-up cycle to power-up cycle and across link re-synchronization events, provided that the auxiliary timing signals meet the required specifications at the device inputs.

The achievement of deterministic delay across a link involves two requirements:

1. In the TX device, ILA generation must be initiated simultaneously across all lanes at a well-defined moment in time. (This also ensures that user data following the ILA is initiated simultaneously across all lanes at a well-defined moment in time).
    a. The ‘well-defined moment in time’ for ILA generation (and hence user data generation) in the TX device is the first LMFC boundary after the detection of the SYNC~ rising edge. While TX devices must be able to generate the ILA on the first LMFC boundary after the detection of the SYNC~ rising edge, devices may also support a programmable number of additional LMFC boundaries to wait before starting the ILA sequence.
2. In the RX device, the incoming data on each lane must be buffered to account for skews across TX SERDES lanes, physical channels, and RX SERDES lanes. The RX buffers must be released (i.e., data allowed to propagate) simultaneously across all lanes at a well-defined moment in time.
    a. The ‘well-defined moment in time’ for RX buffer release is a programmable number of frame cycles after an LMFC boundary. This programmable number of frame cycles is referred to as the Rx Buffer Delay (RBD). For details on which LMFC boundary to use, see the Example given in this clause.

The ILA generation and Rx Buffer Release alignments mentioned above are related to the LMFCs in the TX and RX devices. Thus, the achievement of deterministic latency with minimum uncertainty relies on aligning the LMFCs within TX and RX devices as closely as possible.

6.1 Introduction (cont'd)

In order to achieve proper performance of the deterministic latency protocol, the following requirements must be observed by system implementers:

- The length of a multiframe must be larger than the maximum possible delay across any link. (Link delay is defined in the example below)
- The value of RBD x T_f (frame period) must be larger than the maximum possible delay across any link. (Link delay is defined in the example below)
- The value of RBD, in terms of frame cycles, must be between 1 and K.

The purpose of the above 3 requirements is to ensure that the RBD is large enough to guarantee that the Tx data will have reached the Rx Buffers for all lanes before the Rx Elastic Buffers are released.

The resulting latency across the JESD204B link is equal to RBD x T_f.

EXAMPLE Implementation of deterministic latency in RX devices

Deterministic latency for JESD204B links requires the RX device to be able to buffer incoming ILA or user data on all lanes until the Rx elastic buffers can be released. The buffers must be released RBD frame cycles after a LMFC boundary. In order to release the elastic buffers, the following condition must be true:

- Valid data must be present in the Rx elastic buffers for all active lanes
    - If the ILA sequence is passed through the Rx elastic buffers, the 'valid data' mentioned above will be the start of the ILA sequence.
    - If the ILA sequence is not passed through the Rx elastic buffers, the 'valid data' mentioned above will be sample data appearing after the completion of ILA. In this case, the Rx elastic buffers will be released exactly 4 multiframes later than the scenario described above, (as there are 4 multiframes in the ILA sequence for subclass 1 devices).

The delay across the link can be represented by: $Delay_{LINK} = \Delta T_{LMFC}$ = TX delay + Lane Delay + RX delay, with the following definitions:

TX Delay: Delay from parallel TX ILA generation (which is aligned to an LMFC boundary) to appearance of ILA at TX SerDes output.

Lane Delay: Delay across the external physical channel.

RX Delay: Delay from RX SerDes input to elastic buffer output. The start of ILA and/or start of user data will appear at the elastic buffer output with an alignment equal to LMFC boundary + RBD frame cycles.

ΔT_{LMFC} : The total delay across the link, which can be expressed as delay between the TX LMFC rising edge on which start of ILA/user data is written into the link, and the RX LMFC + T_f* RBD edge on which the start of ILA/user data appears at the output of the RX elastic buffers.

The minimum size required for the RX elastic buffers is equal to the difference between the earliest possible arrival of data to the RX elastic buffer input, and the next RX elastic buffer 'release opportunity', which occurs RBD frame periods after each LMFC boundary.

6.2 No Support for Determisitic Latency (Device Subclass 0) (Informative)

Subclass 0 is defined for devices that wish to take advantage of JESD204B features while maintaining backwards compatibility with JESD204A, without requiring support for deterministic latency. A subclass 0 device can be summarized as follows:

- No support for deterministic latency (i.e. no method for synchronizing LMFCs across transmitter and receiver devices)
- Optionally supports a device clock input without requiring an externally supplied frame clock.
- Optionally allows for lane operation up to 12.5 Gbps.
- Full backward compatibility with JESD204A provided that the OIF-Sx5 physical layer variant is used.

6.3 Deterministic Latency Using SYSREF (Device Subclass 1)

For Subclass 1 devices, the creation of properly aligned LMFC signals within TX & RX devices is achieved using a signal designated as 'SYSREF', which must be distributed to all converter and logic devices.

Delay uncertainty in the system is minimized by using highly accurate SYSREF / device clock signal pairs. While it is not mandatory, it is recommended that SYSREF be generated from the same device producing the TX & RX device clocks.

The requirements for the SYSREF signals and the adjustment capabilities of the local clocks have been provided in 4.8 and 4.11.

Due to the variety of allowable SYSREF signal types (periodic, one-shot (strobe-type), or "gapped" periodic), the variety of clock generator devices that may not support all the options for SYSREF generation, and to accommodate systems where SYSREF will be 'turned off' during normal operation, the following requirements for Subclass I devices apply:

- RX Logic Devices shall have the ability to issue a 'generate SYSREF' request, which enables the clock generator (or other SYSREF generating device) to generate one or more SYSREF pulses to all devices in the system. If enabled, this 'generate SYSREF' request shall be issued any time a link is issuing a re-synchronization request on the SYNC~ interface.
- TX Logic Devices shall have the ability to issue a 'generate SYSREF' request, which enables the clock generator (or other SYSREF generating device) to generate one or more SYSREF pulses to all devices in the system. If enabled, this 'generate SYSREF' request shall be issued any time a link is detecting a re-synchronization request on the SYNC~ interface.
- TX & RX devices shall have the ability to determine whether or not to adjust the phase alignment of local frame & multiframe clocks based on the next detected SYSREF pulse. Implementation details for this function are left to the device implementer but three possible options are listed below:
    - Every SYSREF pulse may be examined by the device to determine if the existing phase alignment of LMFC and frame clocks requires adjustment.
    - A device may be instructed via device input pin or control interface command to use the next detected SYSREF pulse to force a LMFC and Local Frame phase alignment.
    - A device may be instructed via device input pin or control interface command to ignore all future SYSREF pulses.

6.3 Deterministic Latency Using SYSREF (Device Subclass 1) (cont'd)

It should be noted that for Subclass 1 devices, LMFC and Frame Clock phase re-alignment based on SYSREF will only be necessary in cases where a device is being initialized or where a link has exhibited a failure and is requesting a re-synchronization request.

In addition, subclass 1 devices must comply with the following requirement related to SYSREF timing:

- TX & RX devices shall specify the functional delay from device clock edge upon which SYSREF is sampled, to LMFC rising edge.

EXAMPLE LMFC adjustment for deterministic latency equal to multiframe period

For applications where deterministic latency equal to a full multiframe period is desired, the value of RBD should be set to 'K'. This forces the Rx elastic buffers to be released exactly on a multiframe boundary. Figure 36 provides a timing diagram illustrating this scenario.

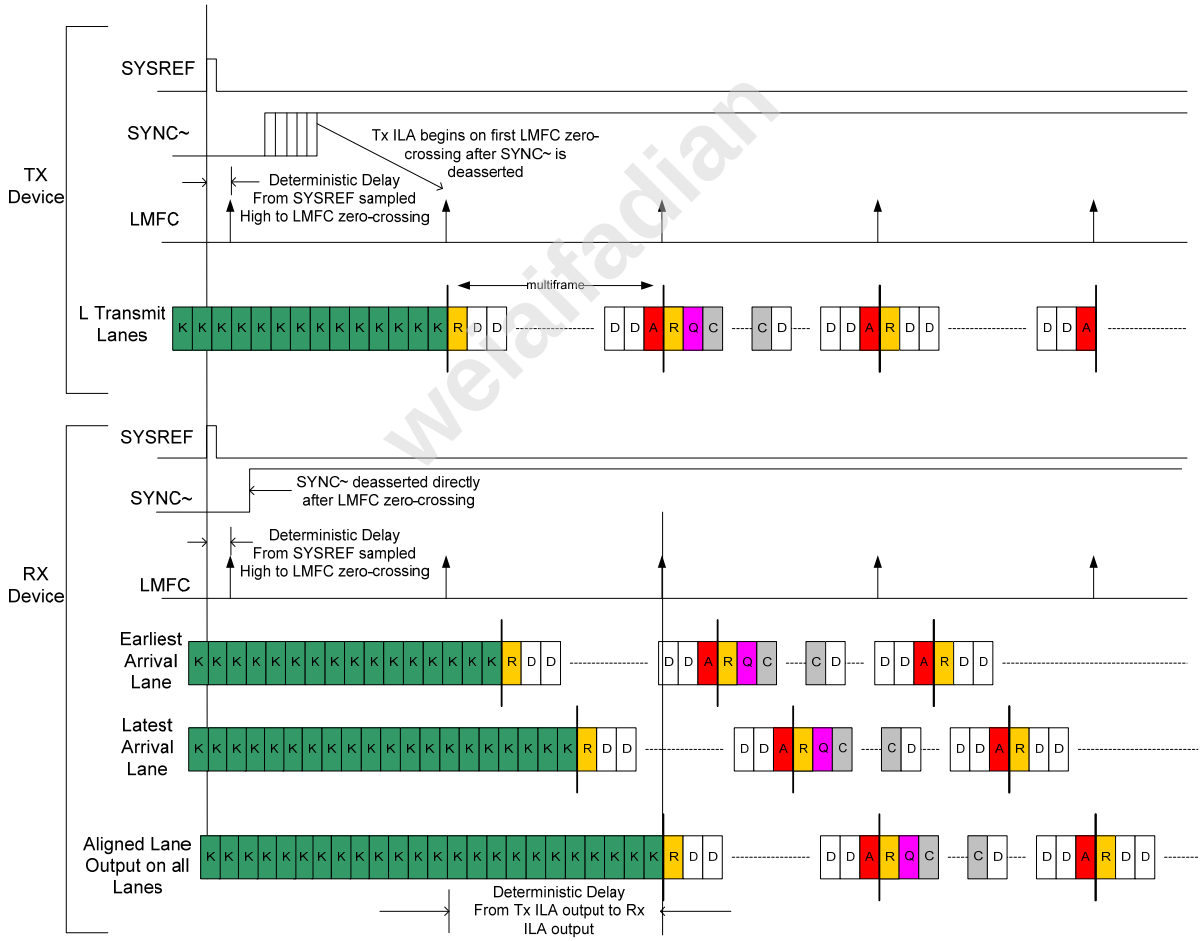


Figure 36 — Timing Diagram Illustration for deterministic latency equal to multiple of multiframe period

In Figure 36, the TX and RX devices have the same device clock cycle delay from SYSREF sampled high to LMFC rising edge. This results in identical LMFCs in the TX and RX devices.

6.3 Deterministic Latency Using SYSREF (Device Subclass 1) (cont'd)

When the RX device has achieved code group synchronization on all lanes, it deasserts the SYNC~ output at any subsequent LMFC rising edge. Shortly after, the TX device samples this deactivated SYNC~ signal and begins to transmit the ILA sequence at a subsequent LMFC rising edge (first available LMFC rising edge is used in Figure 36). The RX device will then detect the start of ILA sequences on all lanes, and will feed this data into per-lane elastic store buffers. On the next LMFC rising edge, the RX device will have detected the presence of valid ILA data on all lanes, and will release all the elastic store buffers. The resulting data output from the RX device is aligned with a fixed latency of 1 multiframe across the JESD204B link.

EXAMPLE LMFC adjustment for minimizing deterministic latency

For applications where a different deterministic latency is desired (namely, when trying to minimize the delay across the link, or the required RX elastic store buffer size) RBD should be less than 'K'. Figure 37 provides a timing diagram illustrating this scenario.

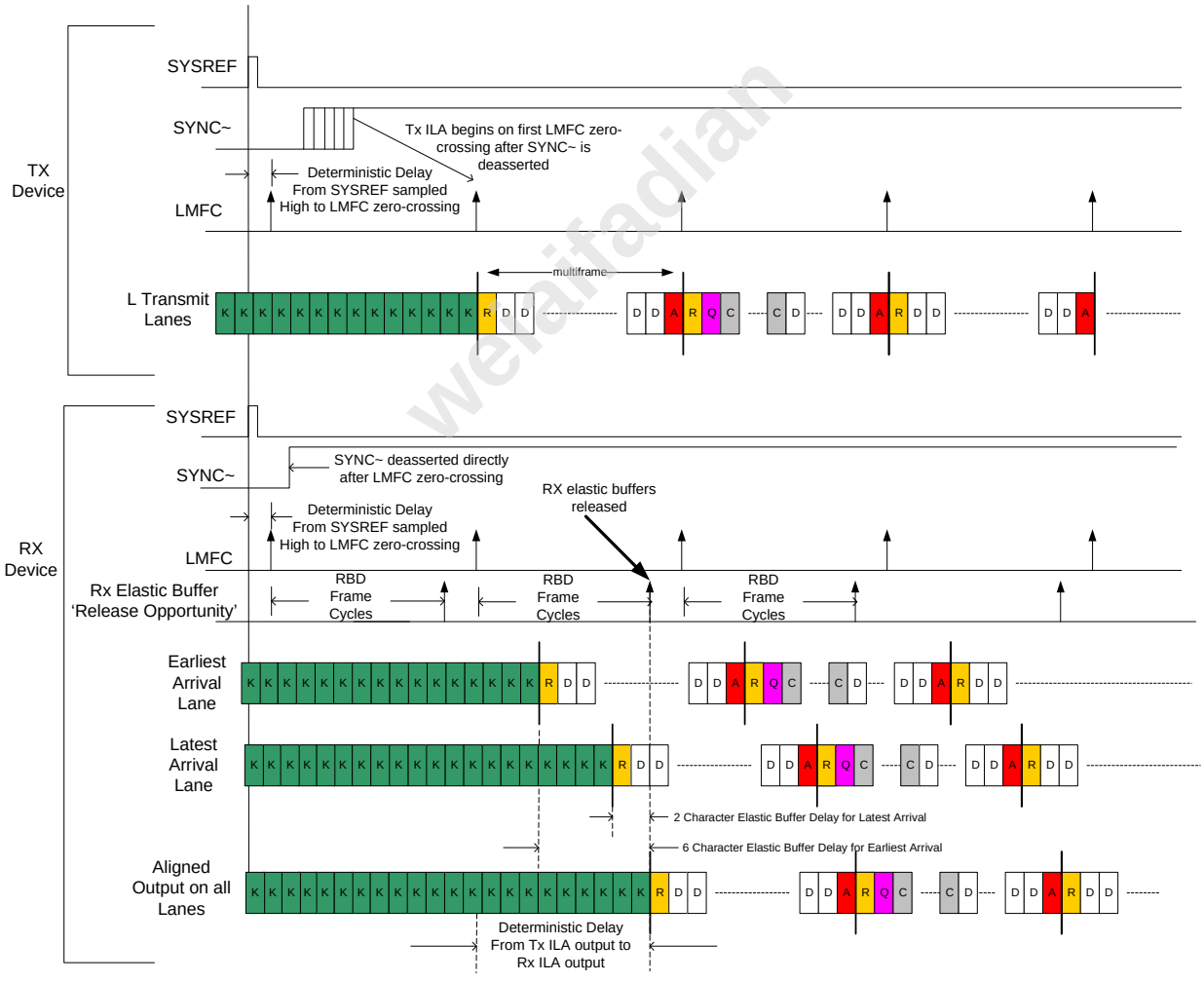


Figure 37 — Timing Diagram Illustration for achieving minimum deterministic latency

6.3 Deterministic Latency Using SYSREF (Device Subclass 1) (cont'd)

In Figure 37, both The TX and RX devices have the same frame cycle delay from SYSREF sampled high to LMFC rising edge, resulting in identical LMFCs in the TX and RX devices. However, in this example, the RX device uses RBD < K to provide a RX elastic buffer 'release opportunity' that is not aligned to a LMFC boundary.

When the RX device has achieved code group synchronization on all lanes, it deasserts the SYNC~ output at any subsequent LMFC rising edge. Shortly after, the TX device samples this deactivated SYNC~ signal and begins to transmit the ILA sequence at a subsequent LMFC rising edge (first available LMFC rising edge is used in Figure 37). The RX device will then detect the start of ILA sequences on all lanes, and will feed this data into per-lane elastic store buffers. On the next 'release opportunity' (i.e., RBD frame clock cycles after an LMFC rising edge), the RX device will have detected the presence of valid ILA data on all lanes, and will release all the elastic store buffers. The resulting data output from the RX device is aligned with a fixed latency of RBD frame cycles across the JESD204B link.

6.4 Deterministic Latency Using SYNC~ Detection (Device Subclass 2)

Deterministic latency in Subclass 2 devices also relies on correct alignment of the TX and RX LMFCs, but alignment must be achieved without the use of an additional SYSREF signal. In general all information and examples outlined in 6.1 and 6.3 with exception of SYSREF related clauses is directly applicable in Subclass 2 operation. Instead of alignment based on SYSREF (subclass 1), the system synchronous SYNC~ signal must be accurately captured by the TX device to determine the relative alignment of TX and RX multi-frame periods. With knowledge of the relative phase difference between RX and TX multi-frame periods, correct alignment must be achieved by

a) aligning a TX converter device to the RX logic device, or

b) aligning a RX converter device to the TX logic device.

The accuracy of the deterministic latency in subclass 2 is based on the accuracy of the transfer and the sampling of the SYNC~ de-assertion on the SYNC~ interface. At both the RX and TX ends of the link, the devices will introduce a variable amount of delay to the launch and capture phase of SYNC~ respectively. The extent to which the variability of the delay can be controlled will limit the maximum speed at which deterministic latency can be achieved in subclass 2. Figure 38 gives an idea of the variable delays in the system and the effects they have on the available timing margin. The figure is shown for the case of TX device clock = frame clock and shows representative timing effects when SYNC~ is detected with an internal clock. As can be seen, if SYNC~ is either launched from or detected by the device clock, less PVT variation is incurred than if the launch or detection comes directly from an internal adjustment or detection clock. All means necessary to minimize variation of delay due to PVT in SYNC~ launch and detect must be used at both ends of the link to maximize speed of operation. The variable delays are specified by the manufacturer through the parameters outlined in subclause 4.9. The maximum speed of operation is achieved when the excess timing margins labeled "PVT margin" in the figure reach zero.

6.4 Deterministic Latency Using SYNC~ Detection (Device Subclass 2)

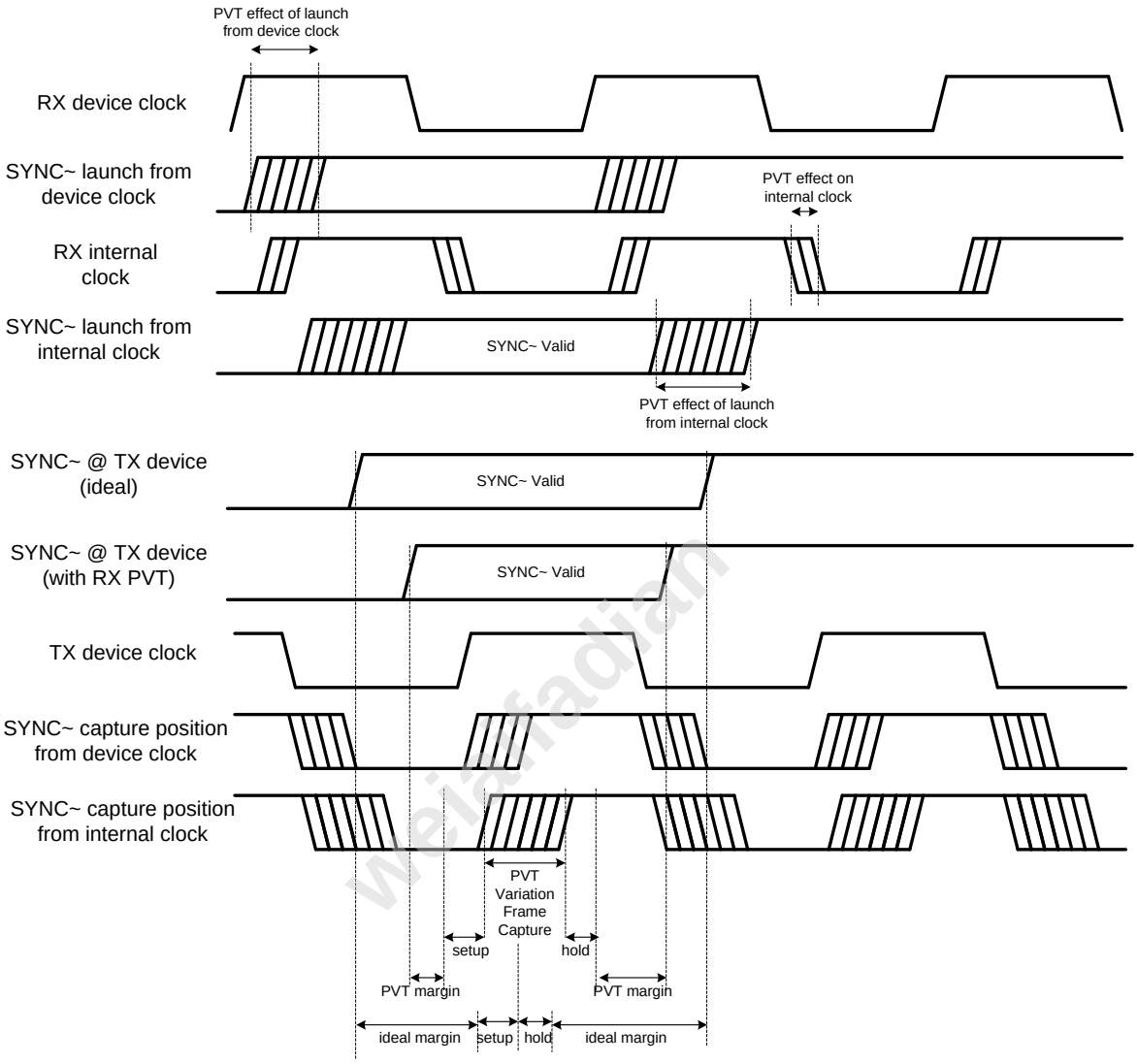


Figure 38 — PVT effects on SYNC~ detection and generation

6.4.1 Principles of SYNC~ sampling

The following section outlines concepts that are important to understand when developing a subclass 2 deterministic latency system.

6.4.1.1 SYNC~ generation at the RX device

The SYNC~ signal from the RX shall always be generated from the internal frame clock of the device and its de-assertion is aligned to the RX LMFC boundary. The de-assertion carries the phase information of the RX LMFC to the TX device via the SYNC~ interface. Generally the frame clock will be used to directly launch SYNC~ to the TX device. When the ratios between the clock frequencies of a device allow it, the SYNC~ should be re-aligned to the device clock before launch.

6.4.1.2 Adjustment resolution and adjustment clock

In converter devices, the phase of the LMFC is adjusted in increments of the adjustment resolution. The adjustment resolution is defined as the minimum time-step in which the LMFC can be adjusted. It is set by the highest frequency device clock in the link; and as such may be finer than the converter device clock period. It is useful to define an adjustment clock with the following properties:

- The adjustment clock period is equal to the adjustment resolution of the link and shall be programmable through a control interface.
- The converter local frame clock and LMFC will be aligned to a rising edge of the adjustment clock.
- The adjustment clock is phase aligned to the device clock for purposes of timing measurement.

If the adjustment resolution is both finer than the converter device clock period and the frame clock period, it may be necessary for the converter to develop a high speed adjustment clock from its device clock to meet the adjustment resolution of the link. Figure 39 shows two possibilities which may arise in a subclass 2 link with regards to adjustment clock:

1. When the converter device clock is faster than the logic device clock, then the adjustment resolution will be the period of the converter device clock.
2. When the converter device clock is the slower of the logic device clock, then the adjustment resolution will be set to the period of the logic device clock.

   NOTE Only in cases where the adjustment resolution is finer than the converter device clock period, the converter LMFC need not be aligned to the device clock.

6.4.1.2 Adjustment resolution and adjustment clock (cont'd)

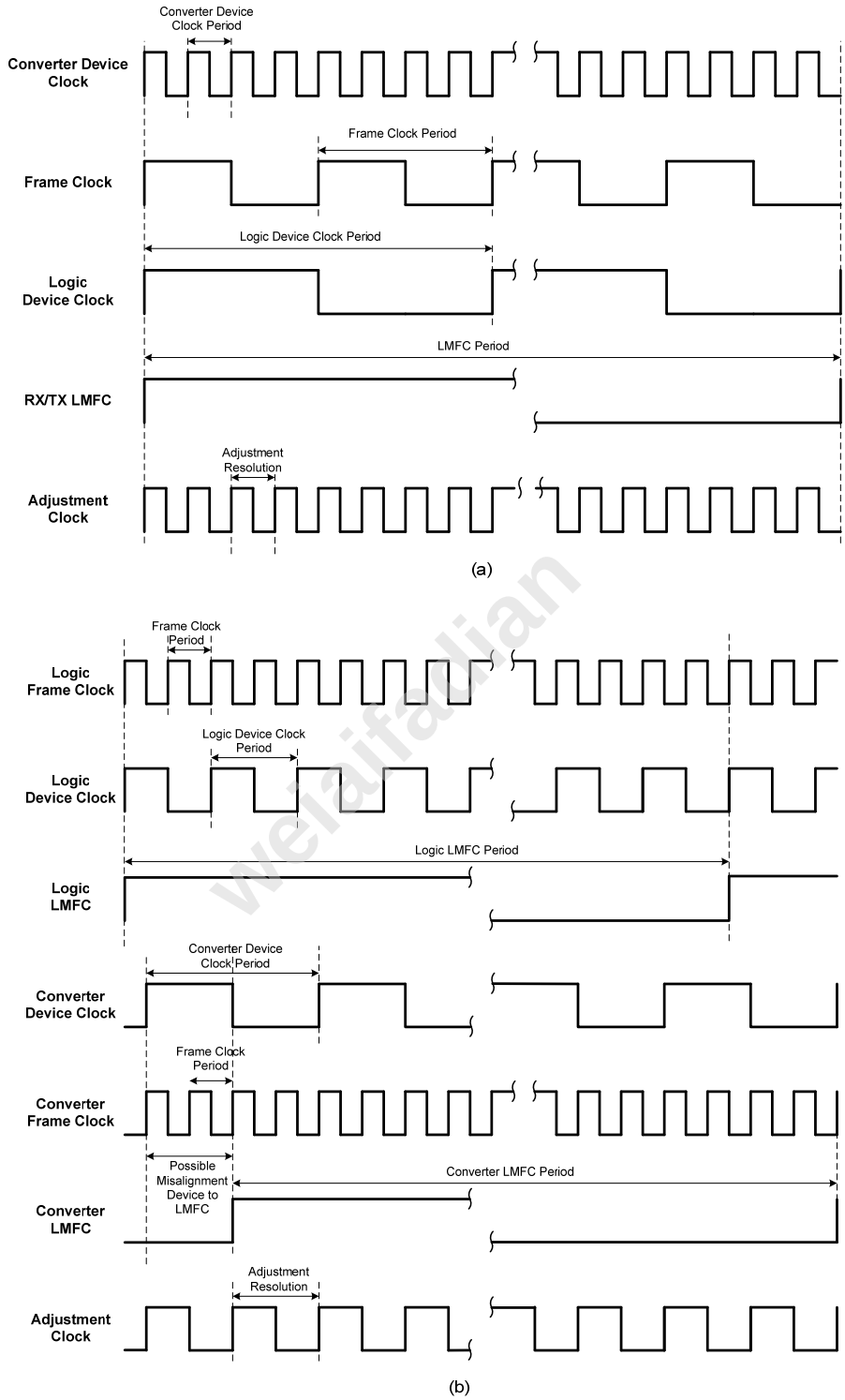


Figure 39 — Relationships between device clocks and adjustment clock, (a) converter device clock faster than logic device clock, (b) logic device clock faster than converter device clock

6.4.1.3 Detection resolution at the TX device

In TX devices, the de-assertion of SYNC~ is detected at the SYNC~ decoder. As the de-assertion of SYNC~ carries the phase of the RX LMFC, the phase is detected and acted upon. The detection of the de-assertion is performed based on a detection clock with period equal to the detection resolution of the link. This clock period may not be longer than the period of the TX local frame clock as error reports must also be considered in SYNC~ detection. The accuracy of this subclass's latency relies on the relationship that the detection resolution is finer than or equal to the adjustment resolution. There is a possibility that the detection resolution of a TX logic device may differ from the adjustment resolution of its links. In order to develop fine detection resolution from a slower device clock, it may be necessary to develop a higher speed detection clock inside the TX device.

6.4.1.4 SYNC~ de-assertion detection and the detection interval

As the RX and TX device clocks may not always be phase aligned to within one frame clock across any one link, SYNC~ de-assertion must be detected using the TX local frame clock in order to accurately process SYNC~ de-assertion from error reports. The possibility exists that the adjustment resolution of the link will be coarser than the detection resolution. Due to this, there may be multiple detection points within one adjustment step. If this is the case, an interval in detection resolution steps can be defined to equal the adjustment resolution. The detection interval now will allow the TX device to detect based on the converter's ability to adjust itself. A detection interval has the following properties:

- The detection interval exists only when the adjustment resolution of the link is coarser than the detection resolution.
- The detection interval is set equal to the adjustment resolution of the system.

Figure 40 shows how a detection interval would function for a DAC link. In this example, the logic device samples SYNC~ de-assertion with its frame clock. The vertical arrows indicate valid detection positions for interval "1". When SYNC~ is sampled at either position, it will be considered detected within interval "1".

6.4.1.4 SYNC~ de-assertion detection and the detection interval (cont'd)

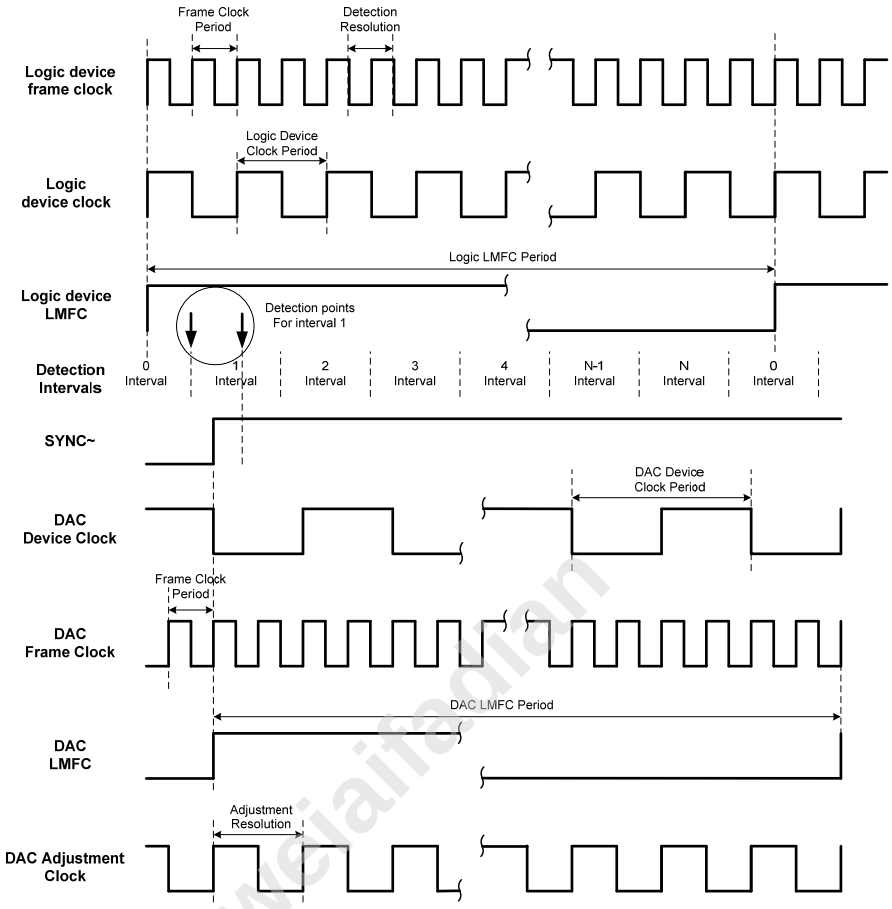


Figure 40 — Relationship of detection interval to system signals (DAC link).

6.4.2 Master and slave configurations

Subclass 2 is a master and slave configuration between two devices. In multiple converter configurations, the logic device is the only device in the system that can share a common LMFC across all links. For this reason, the LMFC of the logic device shall be used as the master timing reference of the link and the converter will always slave from the master reference. The operation of the master and slave configurations for ADC devices and DAC devices is outlined in 6.4.2.1.

6.4.2.1 ADC Master and slave configuration

In a Subclass 2 ADC device, the de-assertion of SYNC~ input is detected based on the ADC detection resolution. If the adjustment resolution of the ADC is finer than the frame period, then the ADC will reset both its local frame clock and LMFC phase relative to the detected de-assertion phase. Alternatively, if the adjustment resolution is not finer than the frame clock period, then the adjustment clock will be aligned to the local frame clock and no frame clock reset will be necessary. The local frame clock and LMFC phase reset must occur a deterministic number of adjustment clock periods following the detection of SYNC~ de-assertion. This delay shall be defined by the device manufacturer. The ADC can perform LMFC phase alignment operations based on one-shot or repetitive SYNC~ de-assertions. The usage of SYNC~ for latency monitoring is left to the implementer. Examples 1 and 2 give a few implementation possibilities.

EXAMPLE 1: Synchronization request based SYNC~ usage

In this usage model, the ADC will assume that each synchronization request from the RX logic device also be considered a request for phase reset. Upon initial ADC link startup, the RX logic device will de-assert the SYNC~ after code group synchronization. The ADC will reset its LMFC and frame clock to that SYNC~ de-assertion location. The ADC will issue at least a minimum duration of /K28.5/ characters on its data lanes to reinitialize the link (5.3.3.7) if a frame or LMFC phase adjustment is made. It will continue to send /K28.5/ characters until internal timing adjustment has settled. After this point, on the next ADC LMFC boundary, a new ILA sequence will be sent along with a resumption of data transmission. This mode can may also be used if deterministic latency accuracy is not particularly critical and SYNC~ detection can violate setup/hold. **Figure 41** shows a timing diagram of the procedure outlined above.

EXAMPLE 2: All event based SYNC~ usage

In this usage model, the SYNC~ de-assertion may be used at each synchronization request or error report. Both events will be considered equally. Upon the first de-assertion seen by the ADC, it will reset its phases as outlined above in Example 1 Subsequent SYNC~ de-assertions can be used to either reset the phases or to monitor the latency of the link. If constant reset of the link is required, it is important that SYNC~ de-assertion detection meet proper setup/hold requirements to prevent unnecessary phase adjustment during each reset operation. Each detection of a SYNC~ de-assertion will reset with behavior similar to the “One shot SYNC~ usage” mode. Subsequent de-assertions can alternatively be used to monitor whether the alignment of the TX and RX LMFCs is correct. This can be done by simply detecting each de-assertion and comparing its phase to the current expected phase for SYNC~ de-assertion. If the phases are different, then latency has been lost. In the event that the detection resolution is finer than the time period required to meet the total timing margin the SYNC interface, a programmable “window” of expected phases can be used for this comparison. Finally, if the link latency has been lost, the ADC device can always reset its LMFC from a future SYNC~ de-assertion by either waiting for an error report or forcing one via a resynchronization over the data interface (5.3.3.7).

6.4.2.1 ADC Master and slave configuration (cont'd)

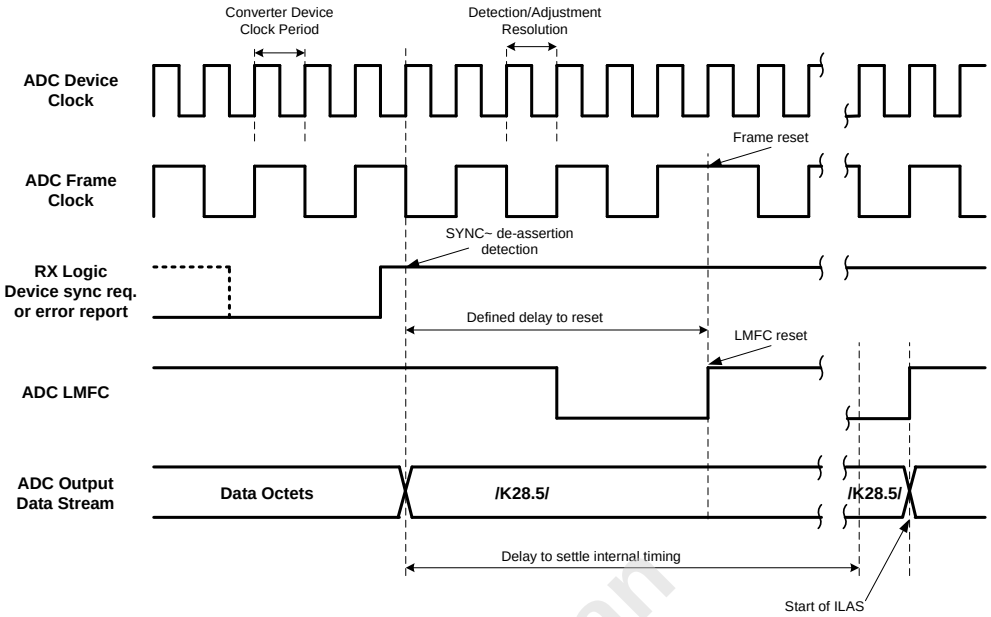


Figure 41 — ADC reset as a result of SYNC~ de-assertion detection

6.4.2.2 DAC master and slave configurations

A Subclass 2 DAC device must support adjustment of its internal frame clock and LMFC phases in response to requests by a logic device. At the DAC, de-assertion of the SYNC~ signal will always be launched at the LMFC boundary. As mentioned above, it can be launched relative to either the device clock or the internal frame clock. The SYNC~ de-assertion for each individual DAC will be detected, to the accuracy of the detection resolution, at the TX logic device. The detection phase will be compared to the TX logic device LMFC phase. If adjustment of the DAC LMFC phase is required, the logic device will signal the adjustment to the DAC by embedding the request, PHADJ, an adjustment step count, ADJCNT, and an adjustment direction, ADJDIR, in configuration bytes 1 and 2 during an ILA sequence. Table 21 defines the locations of PHADJ, ADJCNT, and ADJDIR in the configuration data for a DAC link.

If the need to adjust the LMFC was detected from a synchronization request, the ILA will occur as part of initial lane synchronization, see 7.4. Otherwise the logic device logic device will perform a re-initialization over the data interface (5.3.3.7) in order to provide the ILA sequence. If SYNC~ combining is used in the system, SYNC~ de-assertion detection must take place prior to SYNC~ combination, allowing for individual phase adjustment of each DAC device.

6.4.2.2 DAC master and slave configurations (cont'd)

One bit, PHADJ, is defined for the purposes of adjustment request. ADJCNT<3:0> is defined as the adjustment resolution step count. ADJDIR is defined as the direction of adjustment. ADJDIR equal to 1 is a delay while ADJDIR equal to 0 is an advance. When the DAC sees PHADJ set while processing an ILA sequence, it will respond by adjusting its LMFC by the number of adjustment resolution steps sent in ADJCNT<3:0> and in a direction defined by ADJDIR (see NOTE 1). After adjustment is done, the DAC will issue an error report to the TX logic device with the new LMFC phase information as an acknowledgement of adjustment request (see NOTE 2). The use of PHADJ, ADJCNT, and ADJDIR enables an iterative adjustment of the DAC LMFC until which point it is phase aligned to the TX logic device LMFC.

NOTE 1 Not all DACs will provide the full ability to adjust up to 15 steps. The number of steps which can be adjusted in any DAC must be defined and, if less than 15, will be LSB justified in ADJCNT<3:0>.

NOTE 2 In order to prevent invalid LMFC phase information passing from DAC to TX logic device the reporting of reception errors must be suspended during loop operation.

Figure 42 shows an example state diagram used by the TX logic device to handle alignment of a subclass 2 DAC LMFC. During the detect SYNC~ de-assertion state, the TX logic device samples SYNC~ continually using its detection clock. When a de-assertion is detected, the TX logic device determines whether an adjustment is needed relative to its LMFC. An adjustment is needed if a discrepancy exists between the LMFC boundary and the SYNC~ detection based on either direct detection resolution or detection interval. If an adjustment is not required and the SYNC~ de-assertion did not happen as a result of a synchronization request from the DAC, an adjustment need not be requested, and the TX logic device goes back to sampling the SYNC~. If either an adjustment is required or the SYNC~ de-assertion happened as a result of a synchronization request by the DAC, then a link initialization will be performed as follows. If an adjustment is required, it will be requested by setting PHADJ high, setting ADJCNT<3:0> to the number of adjustment steps requested, and ADJDIR to the direction requested.

If no adjustment is required and the SYNC~ de-assertion is the result of a synchronization request from the DAC, PHADJ will be set to zero. After these parameters are set, the TX logic device will send a stream of /K/ characters and an ILA sequence (see 5.3.3.7 or 7.4), in which the three parameters above will be embedded. I. The checksum for the configuration data will need to be recalculated to take into account the new values for configuration bytes 1 and 2. Once ILA is transmitted, the TX will return to sampling SYNC~ until a new de-assertion is detected. The DAC will acknowledge the phase change with an error report containing the new DAC LMFC phase information. This error report will allow the loop to iterate.

6.4.2.2 DAC master and slave configurations (cont'd)

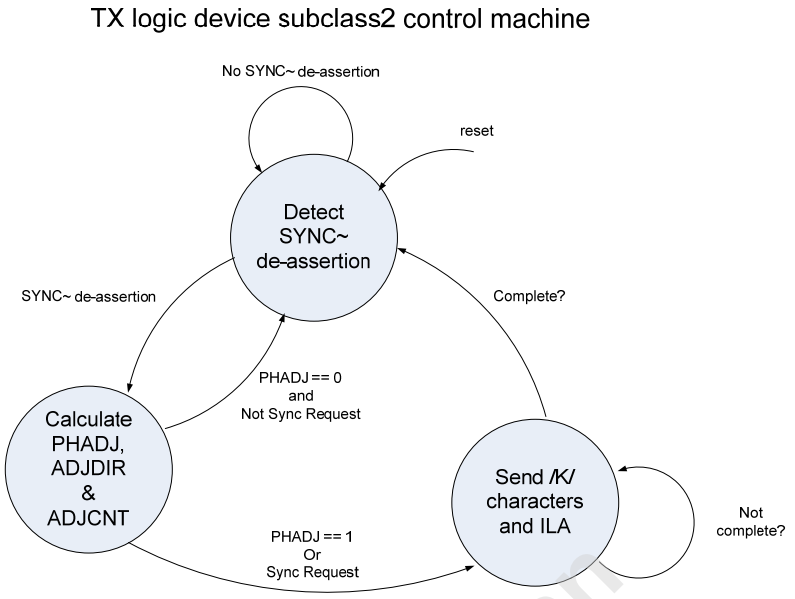


Figure 42 — TX logic device supporting subclass 2 deterministic latency

The DAC will respond to the requests from the logic device. Figure 43 shows a adjustment machine state diagram used by the DAC device to perform its end of the loop. After reset, the adjustment machine will monitor the interface for a link initialization condition. This condition will be the reception of multiple /K/ characters followed by and ILA sequence. During the ILA sequence, the machine will evaluate what is transmitted in PHADJ, ADJDIR, and ADJCNT. If PHADJ is not set, then no adjustment is needed and the DAC will proceed to propagate data. The machine will go back to monitoring the interface for the next link initialization condition. If PHADJ is set, then the machine will make a timing adjustment to the DAC frame clock phase and LMFC phase. The adjustment is equal to the number of adjustment resolution steps in ADJCNT<3:0> and in the direction set by ADJDIR. In order that the TX logic device can accurately calculate ADJCNT, the DAC manufacturer must define the DAC adjustment resolution in units of DAC device clock periods.

If ADJCNT equals zero, a zero adjustment is performed. Once the LMFC adjustment is complete, the DAC will issue an error report to the TX logic device through the SYNC~ interface. Once the error report is issued, the machine will go back to waiting for the next link initialization event.

6.4.2.2 DAC master and slave configurations (cont'd)

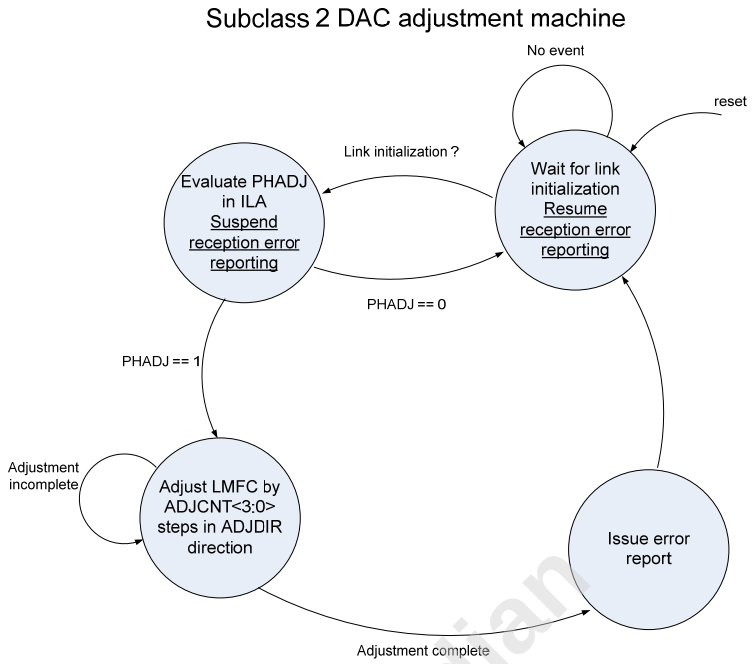


Figure 43 — RX converter device supporting deterministic latency

Loop initialization

In order for the loop above to be set in motion, a synchronization event must be developed. Upon startup, the DAC will assert the SYNC~ signal to request synchronization, and this event will get the loop started. At a future point, the user may want to start the loop again. This can be done in one of two ways. The first way involves the TX logic device performing a re-initialization over the data interface and set PHADJ while setting ADJCNT to zero. By doing so, the TX logic device requests an adjustment of zero counts, and the DAC will respond with an error report. The second way is to program the DAC, via the control interface if so equipped, to issue a synchronization request to the transmitter.

Latency monitoring:

Similarly to the ADC case, SYNC~ detection can be used to monitor the alignment of the RX and TX LMFCs. During an alignment, the TX logic device will adjust the RX LMFC to the proper position.. After initial alignment, a monitoring phase may be entered, and an expected detection phase (based on the detection resolution) can be defined. If future detected SYNC~ de-assertions differ from the expected detection phase, then latency has been lost. It is up to the implementer, as with the ADC, whether a re-alignment be requested or not.

In the case that the detection resolution is finer than the time period required to meet the total timing margin the SYNC interface, a programmable window can be applied to the SYNC~ detection during the monitoring phase. Its width can be set to a programmable number of detection resolution steps. If SYNC~ de-assertion falls outside this detection window, the logic device can treat this as a loss of latency and optionally treat it as above.

6.4.3 Summary of requirements for subclass 2 deterministic latency

General requirements for support of subclass 2 protocol.

- Per 4.9, 7.1, and 7.6.4 a subclass 2 RX device shall de-assert SYNC~ synchronous to its internal frame clock at its LMFC boundary; therefore transmitting its LMFC phase information to a TX device through the SYNC~ interface.
- Per 4.7, the device clocks of the RX and TX shall have a defined relationship.

Requirement for subclass 2 TX converter devices.

- A TX converter device shall have the ability to adjust the phase of its internal frame clock and LMFC signal, in increments of the adjustment resolution, relative to the detection of the SYNC~ de-assertion.
- A TX converter device shall specify the functional delay, in units of adjustment clock periods, from the detection of SYNC~ de-assertion to the start of the TX LMFC.
    - When detection and adjustment are done by the device clock period, this value is referenced to the device clock coincident to the detection of de-assertion.
    - When the detection is done by the frame clock, this value is referenced to the adjustment clock rising edge coincident or following the frame clock when the detection was done.

Requirement for subclass 2 RX converter devices.

- A RX converter device shall have the ability to adjust the phase of its internal LMFC signal in increments of its adjustment resolution.
- An RX converter device shall be able to adjust its LMFC phase when PHADJ is equal to 1 and based on information resident in the ADJDIR, and ADJCNT<3:0>. See Table 20 and Table 21.
- An RX converter device shall issue an error report each time an adjustment has been performed (PHADJ = 1).
- The adjustment resolution of an RX converter device shall be specified in its datasheet for use in the calculation of ADJCNT by the TX logic device. It shall be defined in units of RX device clock periods.

Requirement for a TX logic device for subclass 2 support.

- The TX logic device shall have the ability to detect the phase of the captured SYNC~ de-assertion relative to its LMFC based on its detection resolution.
- The TX logic device shall accept a programmed DAC adjustment resolution, and have the ability to use this adjustment resolution to calculate ADJCNT based on the relative detection required above.
- The TX logic device shall provide correction information to the RX converter device through an ILA sequence as a result of a synchronization request (7.4) or “re-initialization over the data interface” (5.3.3.7). The correction information shall be contained in PHADJ, ADJDIR, and ADJCNT. See also Table 20 and Table 21 as well as 6.4.2.2.

6.5 Interoperability Between JESD204A and JESD204B Devices

JESD204B is not backwards compatible with JESD204A in several areas related to the timing of SYNC~ error reporting. Therefore, a JESD204B receiver device interfacing to a JESD204A transmitter device must comply with the JESD204A requirements in order to achieve full interoperability of SYNC~ error reporting. Without providing an external frame clock to the JESD204B RX device, it may not be possible to meet the setup and hold requirements for the SYNC~ signal at the TX device, as outlined in JESD204A. Consequently it must be possible to disable the error reporting functions on the SYNC interface for the JESD204B RX device, if interoperability with JESD204A devices is to be supported.

Table 13 outlines the various options for SYNC~ error reporting when interfacing various classes of JESD204 devices.

Table 13 — SYNC~ Requirements for Interoperability

TX Device	RX Device	SYNC~ Error Reporting Conditions
JESD204A	JESD204B	RX must have error report generation disabled, or must use error reports of 1 frame width (as per JESD204A).
JESD204B	JESD204A	TX will not always detect error reports of 1 frame width
JESD204B	JESD204B Subclass 0	RX Error report generation shall be as per JESD204B Subclass 1-2

7 Receiver Operation

7.1 Code group synchronization

Code group synchronization in receiver devices is obtained and maintained in the following way:

- On link start-up, the receiver issues a synchronization request and the transmitter emits comma characters /K/= /K28.5/
- The receiver de-asserts the synchronization request as follows:
  - Subclass 0 devices: On any frame boundary after correct reception of four successive /K/ characters.
  - Subclass 1 and 2 devices: On any LMFC boundary after correct reception of four successive /K/ characters.
- After correct reception of another four 8B/10B characters, the receiver assumes full code group synchronization.
- Upon receiving an invalid code, the receiver enters a check state.
- If three additional invalid codes are received while the receiver is in the check state, loss of synchronization is declared.
- The receiver exits the check state and returns to normal operation after receiving four valid codes in succession.

The synchronization request is transferred to the transmitter via the SYNC interface, further specified in 4.9. The synchronization request is encoded as an active low SYNC~ signal and may change state only on the rising edges of the frame clock in the RX device. Furthermore, as stated above, the frame clock edge on which a synchronization request may be de-asserted is as follows:

- Subclass 0 devices: Any frame clock rising edge.
- Subclass 1 or 2: Any frame clock rising edge corresponding to a LMFC boundary.

The minimum duration for a synchronization request on the SYNC~ signal is 5 frames plus nine octets. Further timing requirements are specified in 4.9.

The code group synchronization is illustrated in the receiver state machine of Figure 44. In each of the states, the decoder processes one code group. The meaning of the variables is explained in Table 14.

7.1 Code group synchronization (cont'd)

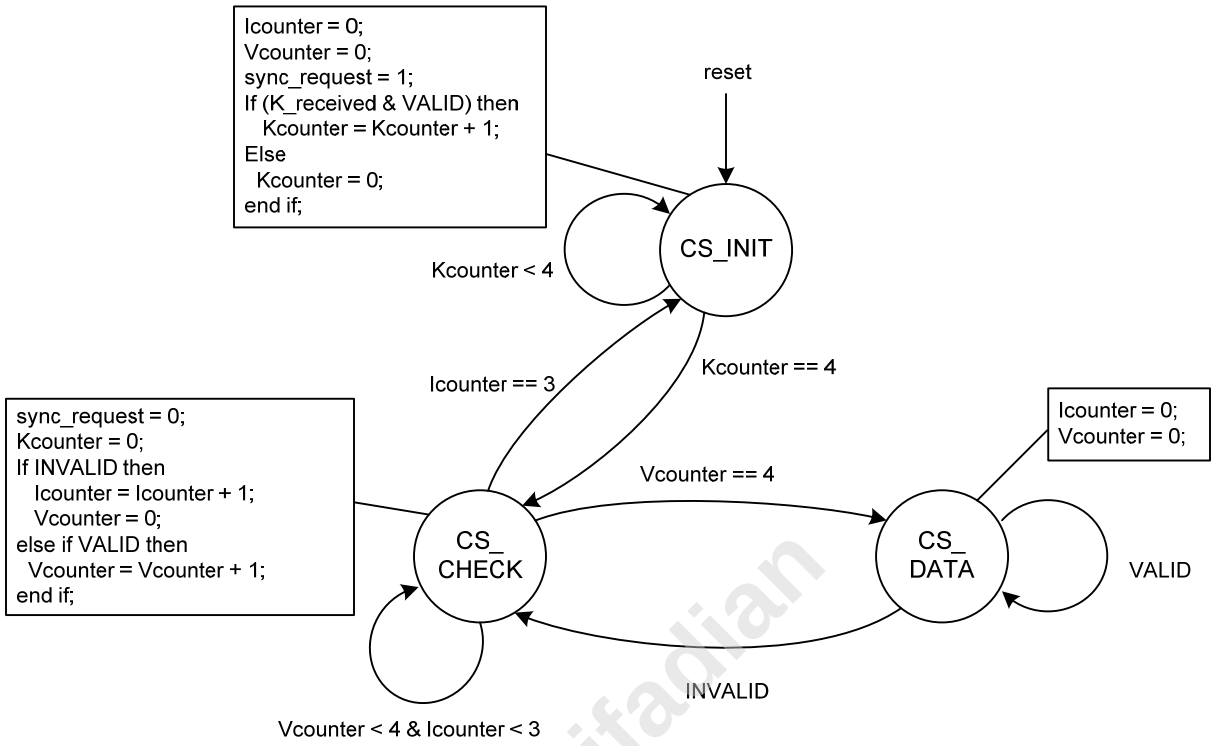


Figure 44 — Receiver state machine for code group synchronization

Table 14 — Variables used in receiver state machine for code group synchronization

Variable	Meaning
Icounter	Counter used in the CS_CHECK phase to count the number of invalid symbols
INVALID	Asserted by receiver to indicate that the current symbol is an invalid symbol given the current running disparity.
K_received	Asserted when the current symbol corresponds to control character K28.5
Kcounter	Counter used in the CS_INIT phase to count the number of valid K28.5 symbols
sync_request	Asserted by receiver when loss of code group synchronization has been detected. Note that sync_request does not drive SYNC~ directly, as SYNC~ assertion/de-assertion is based on more than just the sync_request signal described here.
VALID	Asserted by receiver to indicate that the current symbol is a valid symbol given the current running disparity.
Vcounter	Counter used in the CS_CHECK phase to count the number of successive valid symbols

The receiver is allowed and required to align the code group boundary to received comma characters only during an active synchronization request. During data transmission, commas can be detected across the border of two code groups as the result of bit errors. Spurious commas can also be generated across the border of a frame alignment symbol /K28.7/ and certain data symbols.

7.2 Initial frame synchronization

The initial frame synchronization process is illustrated in the state machine shown in Figure 45 and Figure 46. The meaning of the variables used in these figures is explained in Table 15. At reset, the state machine enters its initial state and the octet counter is cleared (zeroed). At reset, the code group synchronization machine activates a synchronization request, which will keep the frame synchronization in its initial state for the duration of code group synchronization.

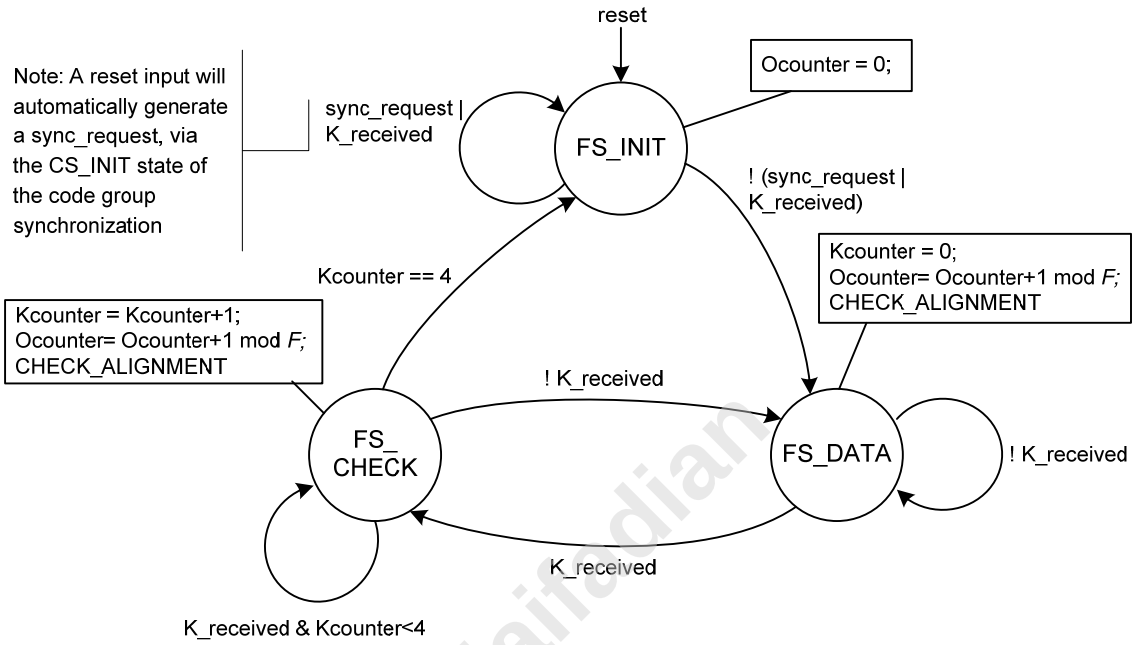


Figure 45 — State machine for frame synchronization in receivers supporting re-initialization over the data interface.

The state machine moves to the FS_DATA state when the receiver has deasserted its synchronization request and the transmitter has stopped sending /K/ symbols. In the FS_DATA state, the octet counter counts the position of the received octet in the frame, between 0 and F-1. A receiver shall return to the FS_INIT state if any receiver on the link issues a synchronization request. In a configuration having multiple receiver devices, this is only possible by monitoring the reception of /K/= /K28.5/ symbols, as shown in Figure 45.

If a K28.5 symbol is received, the state machine moves to the FS_CHECK state. In this state, the octet counter keeps running. However, if four K28.5 symbols are received in succession, the frame synchronization returns to its initial state. Alternately, receivers intended only for configurations with single receiver devices can monitor the synchronization requests of all receivers in the device that are connected to the link. This alternative is illustrated in Figure 46.

7.2 Initial frame synchronization (cont'd)

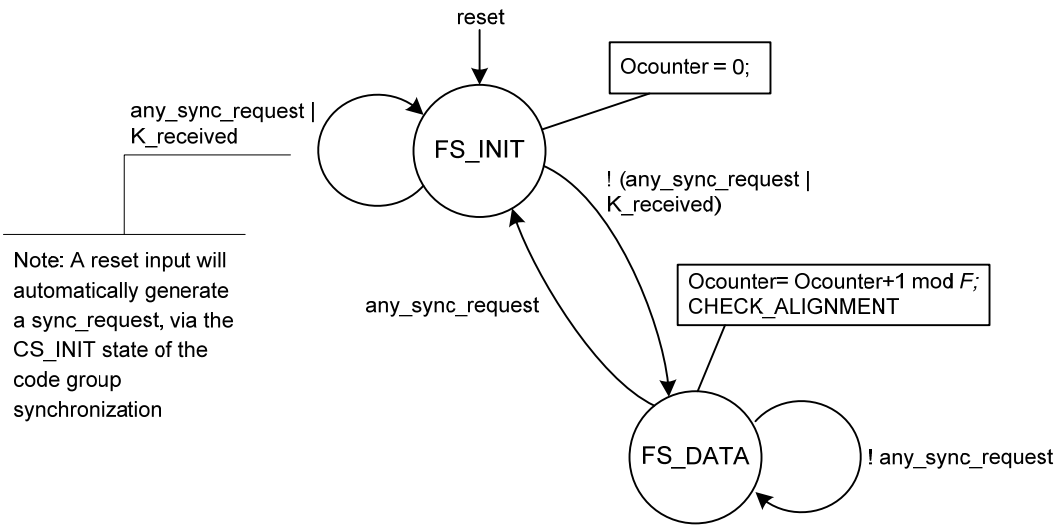


Figure 46— Alternative state machine for frame synchronization in receivers *not* supporting re-initialization over the data interface

Table 15 — Variables used in initial frame synchronization

Variable	Meaning
any_sync_request	A sync request asserted by any receiver connected to the link.
CHECK_ALIGNMENT	Perform frame alignment monitoring, see subclause 7.3
F	Number of octets per frame
Kcounter	Counter used in the FS_CHECK phase to count the number of K28.5 symbols
K_received	Asserted when the current symbol corresponds to control character K28.5 (valid or invalid)
Ocounter	Counter used to mark the position of the current octet in the frame.
sync_request	Asserted by receiver when loss of code group synchronization has been detected or if another error requires re-initialization.

7.3 Frame alignment monitoring and correction

The detection and correction process is illustrated in the pseudo-code of Figure 47. The meaning of the variables and functions is explained in Table 16. It shall be possible to disable the frame alignment correction (RESET_OCTET_COUNTER), e.g. via the control interface, when the user data does not produce sufficient alignment characters for reliable detection of frame alignment errors or when the possible impact of frame re-alignment on lane alignment or link latency needs to be avoided, see 5.3.3.4.4.

```
if (A_received | F_received)
  REPLACE_ALIGNMENT_CHARACTER; /* mark if necessary */
  if (((Ocounter == previous_AF_position) | CROSS_COUPLING) & VALID)
    RESET_OCTET_COUNTER;
  end if;
  if (VALID | (Ocounter==F-1))
    previous_AF_position = Ocounter;
  end if;
end if;
```

Figure 47 — Pseudo code for frame alignment monitoring and correction in receiver

7.3 Frame alignment monitoring and correction (cont'd)

Table 16 — Variables and functions in frame alignment monitoring and correction

Variable	Meaning
A_received	Asserted when the current symbol, before possible substitution in lane alignment monitoring, corresponds to control character K28.3 Note: detection of K28.3 is not required in NMCDA-SL DACs.
CROSS_COUPLING	Frame misalignment expected because of cross coupling between lane and frame alignment
F	Number of octets per frame
F_received	Asserted when the current symbol corresponds to control character K28.7
Ocounter	Counter used to mark the position of the current octet in the frame. Octet indexing starts from 0.
previous_AF_position	Variable into which to store the position in the frame of a K28.3 or K28.7 symbol
REPLACE_ALIGNMENT_CHARACTER	Replace the alignment character at the decoder output by: • The data character decoded or used at the same position in the previous frame when scrambling is disabled • The data character with the same value when scrambling is enabled Mark the possition of a K28.3 character if needed in subsequent lane synchronization or lane alignment monitoring.
RESET_OCTET_COUNTER	Reset octet counter to zero at reception of next octet
VALID	Asserted by receiver to indicate that the current symbol is a valid symbol given the current running disparity.

7.4 Initial lane synchronization

Receivers inside a single device shall be able to achieve mutual lane alignment within four multiframes of the initial lane alignment sequence. Subclass 1 and 2 devices will further align incoming alignment characters to their internal LMFCs within four multiframes. When receivers are divided over several subclass 0 devices, the implementer shall specify how many multiframes are needed in the initial lane alignment sequence, but not more than 256.

Each receiver in a multi-lane configuration shall prepare for a new initial lane synchronization if any receiver on the link issues a synchronization request. Receivers in an MCDA device class (i.e. supporting alignment between multiple receiver devices, see clause 9) shall detect such a request from the reception of 4 successive K28.5 characters. Alternately, receivers intended only for configurations with single receiver devices shall monitor the synchronization requests of all receivers in the device that are connected to the link.

7.5 Lane alignment monitoring and correction

The detection and correction process is illustrated in the pseudo-code of Figure 48. The meaning of the variables and functions is explained in Table 17. The process is very similar to the one used for frame alignment monitoring and correction. However, it contains one extra action: when the alignment characters do not arrive at the expected place, it is recommended to initiate a synchronization check between the LMFCs via one of the methods supported by the device class and subclass.

It shall be possible to disable the lane alignment correction (RESET_FRAME_COUNTER) in case the user data is such that not sufficient alignment characters are produced for reliable detection of alignment errors or when lane re-alignment via a link synchronization request is preferred, see 5.3.3.6.

```
if A_received
  REPLACE_A; /* see Table 17*/
  if (((Fcounter == previous_A_position) | CROSS_COUPLING) & VALID)
    RESET_FRAME_COUNTER;
    if (Fcounter!=K-1)
      INITIATE_SYNC_CHECK;
    end if;
  end if;
  if (VALID | (Fcounter==K-1))
    previous_A_position = Fcounter;
  end if;
end if;
```

Figure 48 — Pseudo code for lane alignment monitoring and correction in receiver

7.5 Lane alignment monitoring and correction (cont'd)

Table 17 — Variables and functions in lane alignment monitoring and correction

Variable	Meaning
A_received	Asserted when the current symbol, before possible substitution in frame alignment monitoring, corresponds to control character K28.3
CROSS_COUPLING	Lane misalignment expected because of cross coupling between frame and lane alignment
Fcounter	Counter used to mark the position of the current frame in the multiframe. Frame indexing starts from 0.
K	Number of frames in multiframe.
previous_A_position	Variable into which to store the position in the multiframe of a K28.3 symbol
REPLACE_A	Replace the K28.3 at the decoder output by: • The data character decoded or used at the same position in the previous frame when scrambling is disabled • D28.3 when scrambling is enabled However, if the position of the K28.3 is required in subsequent frame alignment monitoring, the K28.3 shall not be replaced or it shall be marked.
RESET_FRAME_COUNTER	Reset frame counter to zero at reception of next frame
INITIATE_SYNC_CHECK	In receivers belonging to a MCDA device class (see clause 9), if authorized via the control interface, initiate a synchronization check between the LMFCs via one of the methods supported by the device class and subclass.
VALID	Asserted by receiver to indicate that the current symbol is a valid symbol given the current running disparity.

7.6 Error handling

7.6.1 Error kinds

Table 18 shows the minimum set of errors to be detected in each receiver:

Table 18 — Minimum set of errors to detect per receiver

Error	Description
Disparity error	The received code group exists in the 8B/10B decoding table, but is not found in the proper column according to the current running disparity.
Not-in-table error	The received code group is not found in the 8B/10B decoding table for either disparity
Unexpected control character	A control character is received that is not expected at the given character position
Code group synchronization error	The state machine for code group synchronization has returned to the CS_INIT state

In addition, many other kinds of errors may occur, which may not always require detection or action in each application, e.g.,

- Frame realigned (previous conversion samples may be in error)
- Lane realigned (previous conversion samples may be in error)
- Uncorrectable frame alignment error
- Initial lane alignment failure
- Uncorrectable lane alignment error
- Link configuration data error (parameters in TX and RX do not match)
- Lane alignment sequence decoding error (wrong octets decoded)

7.6.2 Data output on error

This subclause applies only to the reception of data to the transport layer (see subclause 5.1). Figure 44 specifies how to deal with errors occurring during code group synchronization. During initial lane alignment, the reaction on errors will depend on the implementation of lane alignment.

In response to a not-in-table error, the decoder shall repeat the previously received non-error frame. A not-in-table error with scrambling enabled can corrupt the following two octets, so in cases where the corrupted octets span a frame boundary both the frame containing the not-in-table error and the following frame shall be replaced by the frame preceding the not-in-table error.

In case of a disparity error, the output shall be the decoded symbol according to the running disparity indicated by the received code group.

If an unexpected control character is received, the action depends on its value. In case of an /A/ or /F/, octet replacement will take place according to the rules for frame and lane alignment monitoring. Other unexpected control characters are treated like not-in-table characters.

7.6.3 Errors requiring re-initialization

Certain errors cannot be corrected other than by re-initialization of the whole JESD204 channel. Because some of these errors could be tolerated in certain applications, it is recommended that the receiver implementer provide the option to specify via a control interface which errors will lead to re-initialization.

7.6.4 Error reporting via SYNC interface

On detection of an error requiring re-initialization, the receiver shall activate the SYNC~ signal for the duration of a whole number of frames. The minimum duration is five frames plus nine octets.

On detection of an error not requiring re-initialization, the required behavior is dependent on the deterministic latency subclass of the device.

- Subclass 1 or Subclass 2 receiver devices shall indicate the detection of such an error by activating the SYNC~ signal for exactly 2 frame periods. For one or more errors occurring during a particular LMFC period, the SYNC~ signal shall be activated two frame periods prior to the end of the next LMFC period and deactivated at the LMFC boundary associated with the end of this period. Other errors scheduled for signaling during this multiframe period may be ignored, unless they would require link re-initialization.

    - For backwards compatibility with JESD204A, Subclass 1 or 2 receiver devices may optionally support SYNC~ error reports of 1 frame length, and/or the ability to disable error reporting completely.

- A Subclass 2 DAC device must also be able to suspend reporting of errors not requiring re-initialization. As the DAC uses the SYNC~ interface to transmit its LMFC phase, it must control when error reports occur during its master slave alignment operation. Please see subclause 6.4.2.2 for further information.

- Subclass 0 receiver devices shall be programmable to support error report generation in the manner described in JESD204A, as well as the manner described above for Subclass 1 & 2.

- Subclass 0 receiver devices shall also be programmable to suppress reporting errors not requiring re-initialization.

7.6.5 Error reporting via control interface

A receiver in a logic device shall have the ability to report decoding errors via a control interface to higher application layers. The details of this reporting are application- and implementation-dependent. A receiver in a DAC may report errors via a control interface to higher application layers, either directly or via the logic device.

8 Transmitter Operation

8.1 Synchronization

The transmitter state diagram is shown in Figure 49. The variables and functions used are explained in Table 19. The INIT_LANE state may be skipped if both sides of the lane do not support lane synchronization.

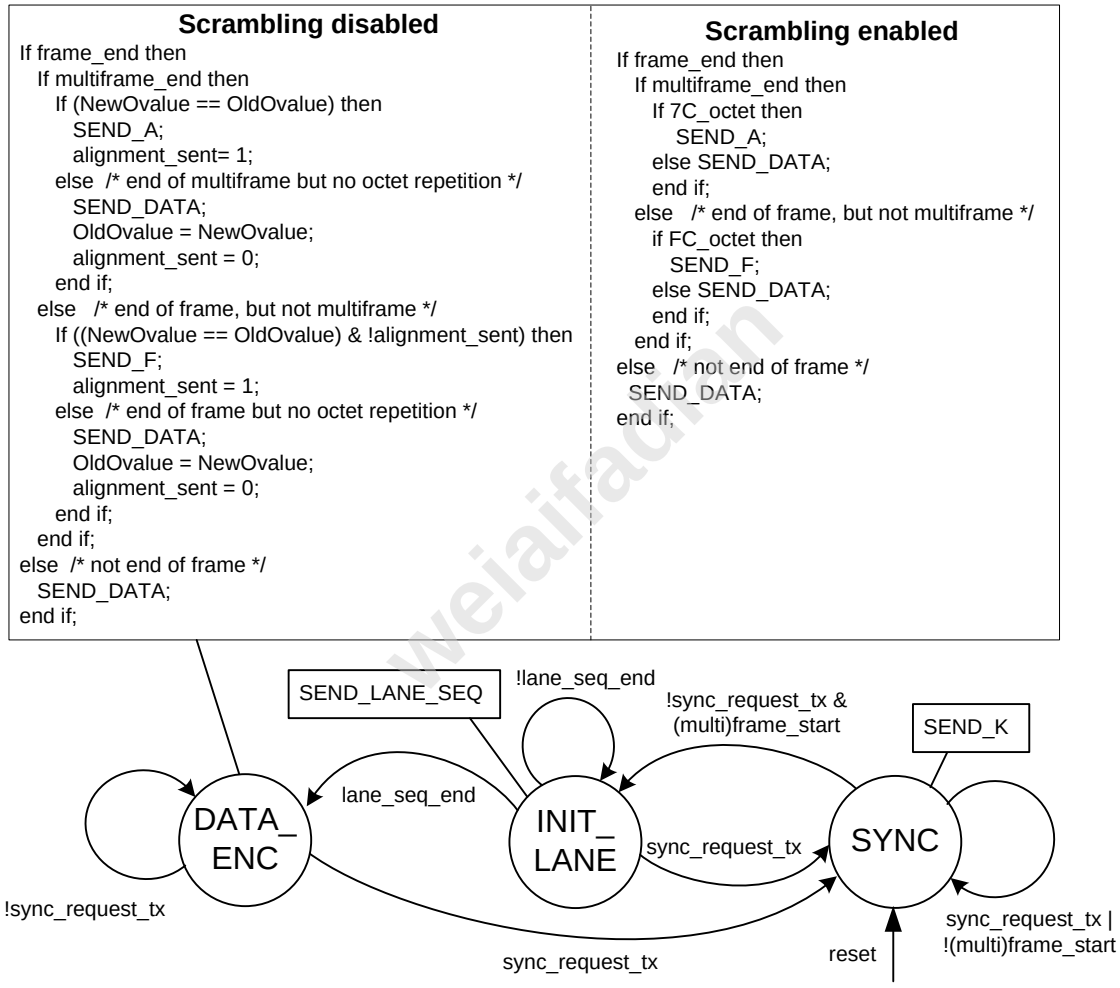


Figure 49 — Transmitter state machine

8.1 Synchronization (cont'd)

Table 19 — Variables and functions in transmitter state machine

Variable	Meaning
7C_octet	Asserted when the scrambler outputs a 0x7C (28.3) octet
alignment_sent	Used to indicate a /K28.3/ or /K28.7/ has been sent in the previous frame.
FC_octet	Asserted when the scrambler outputs a 0xFC (28.7) octet
frame_end	Asserted by transmitter to indicate end of frame.
(multi)frame_start	Asserted by transmitter to indicate start of frame for devices belonging to the NMCDA-SL device class, or start of multiframe for devices belonging to other device classes. (See clause 9).
lane_seq_end	Asserted by transmitter to indicate end of initial lane alignment sequence
multiframe_end	Asserted by transmitter to indicate end of multiframe. Only to be asserted if both sides of the lane support lane synchronization.
NewOvalue	Value of last octet in current frame
OldOvalue	Used for storage of last octet in frame
SEND_A	Send /K28.3/ symbol
SEND_DATA	Send code group belonging to current data octet
SEND_F	Send /K28.7/ symbol
SEND_K	Send /K28.5/ symbol
SEND_LANE_SEQ	Send initial lane alignment sequence
sync_request_tx	asserted when transmitter detects a synchronization request

8.2 Initial lane alignment sequence

The contents of the initial lane alignment sequence are specified by means of the flow diagram of Figure 50. Note that this diagram defines the bits at the input of the 8B/10B encoder. In ADCs the sequence consists of exactly 4 multiframes. In logic devices, the length of the sequence shall be programmable from 4 up to at least 256 multiframes.

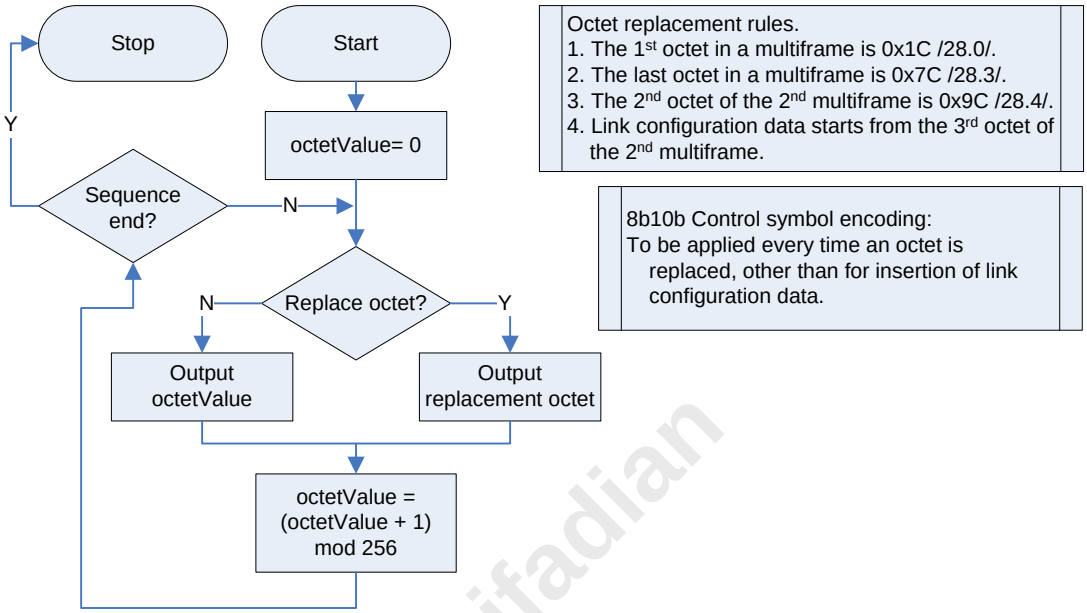


Figure 50 — Character specification for initial lane alignment sequence

8.3 Link configuration data and encoding

The link configuration parameters and their encoding are summarized in Table 20. The link configuration data shown in Figure 35 is meant to specify the parameter values defining the user data formats used by the transmitter device, as described in 5.1, and to specify information on the transmitter configuration and capabilities. In Subclass 2 logic devices (see 9.1) the link configuration data can also contain commands for re-alignment of the receiver LMFC. Table 21 shows how the link configuration fields shall be mapped to octets. It is impossible to foresee all future applications and configurations of the JESD204 interface. Therefore, bits marked by “X” in Table 21 may be used to pass information not specified here or to extend existing fields if they are turn out to be too narrow. When device vendors reach consensus about use of currently unspecified bits, this may be added to the standard in a later phase.

8.3 Link configuration data and encoding

Table 20 — Link configuration parameters

Parameter	Description	Parameter Range	Field	Encoding
ADJCNT	Number of adjustment resolution steps to adjust DAC LMFC. Applies to Subclass 2 operation only.	0 … 15	ADJCNT<3:0>	Binary value
ADJDIR	Direction to adjust DAC LMFC 0 – Advance 1 – Delay Applies to Subclass 2 operation only	0 … 1	ADJDIR<0>	Binary value
BID	Bank ID – Extension to DID	0 ... 15	BID<3:0>	Binary value
CF	No. of control words per frame clock period per link	0 ... 32	CF<4:0>	Binary value*
CS	No. of control bits per sample	0 ... 3	CS<1:0>	Binary value
DID	Device (= link) identification no.	0 ... 255	DID<7:0>	Binary value
F	No. of octets per frame	1 ... 256	F<7:0>	Binary value minus 1
HD	High Density format	0 ... 1	HD<0>	Binary value
JESDV	JESD204 version 000 – JESD204A 001 – JESD204B	0 … 7	JESDV<2:0>	Binary Value
K	No. of frames per multiframe	1 ... 32	K<4:0>	Binary value minus 1
L	No. of lanes per converter device (link)	1 ... 32	L<4:0>	Binary value minus 1
LID	Lane identification no. (within link)	0 ... 31	LID<4:0>	Binary value
M	No. of converters per device	1 ... 256	M<7:0>	Binary value minus 1
N	Converter resolution	1 ... 32	N<4:0>	Binary value minus 1
N'	Total no. of bits per sample	1 ... 32	N'<4:0>	Binary value minus 1
PHADJ	Phase adjustment request to DAC Subclass 2 only.	0 … 1	PHADJ<0>	Binary value
S	No. of samples per converter per frame cycle	1 ... 32	S<4:0>	Binary value minus 1
SCR	Scrambling enabled	0 ... 1	SCR<0>	Binary value
SUBCLASSV	Device Subclass Version 000 – Subclass 0 001 – Subclass 1 010 – Subclass 2	0 … 7	SUBCLASSV <2:0>	Binary Value
RES1	Reserved field 1	0 ... 255	RES1<7:0>	Binary value
RES2	Reserved field 2	0 ... 255	RES2<7:0>	Binary value
CHKSUM	Checksum Σ(all above fields)mod 256	0 ... 255	FCHK<7:0>	Binary value

* *CF*==*L* shall always be encoded as 31: control words on all lanes. *CF*==31 can only occur when *L*==31, see 5.1.3.

8.3 Link configuration data and encoding (cont'd)

Table 21 — Mapping of link configuration fields to octets

Configuration octet no.	Bits							
	MSB	**6**	**5**	**4**	**3**	**2**	**1**	**LSB**
0	DID<7:0>							
1	ADJCNT<3:0>				BID<3:0>			
2	X	ADJDIR<0>	PHADJ<0>	LID<4:0>				
3	SCR<0>	X	X	L<4:0>				
4	F<7:0>							
5	X	X	X	K<4:0>				
6	M<7:0>							
7	CS<1:0>		X	N<4:0>				
8	SUBCLASSV<2:0>			N'<4:0>				
9	JESDV<2:0>			S<4:0>				
10	HD<0>	X	X	CF<4:0>				
11	RES1<7:0> - Set to all X							
12	RES2<7:0> - Set to all X							
13	FCHK<7:0>							

8.4 SYNC signal decoding

A high to low transition of the SYNC~ signal shall be interpreted as a reported error from the receiver (see 7.6.4). If the TX device detects a low SYNC~ signal for at least 4 consecutive local frame clock periods, it shall additionally interpret the SYNC~ pulse as a synchronization request.

Once the SYNC~ signal has been interpreted as a synchronization request, the required behavior of the TX device depends on the deterministic latency subclass of the device:

- Device Subclass 0: A subclass 0 transmitter shall send exactly one frame of /K28.5/ symbols for each additional frame period that the SYNC~ signal remains low. Once the SYNC~ de-activates (i.e., goes high), subclass 0 transmitters shall cease sending /K28.5/ symbols at the next frame boundary after that point.
- Device Subclass 1 & 2: A subclass 1 or 2 transmitter shall send continuous /K28.5/ symbols until the SYNC~ signal de-activates (i.e., goes high). At this point, the transmitter shall continue to send continuous /K28.5/ symbols until both of the following conditions are met:
  - The /K28.5/ symbol generation has lasted for at least 1 frame + 9 octets
  - The end of a local multiframe period is reached. (The transition from /K28.5/ generation to ILA generation must occur at a multiframe boundary, but may be a programmable number of multiframes after SYNC~ signal de-activation is detected. For devices supporting a programmable number of multiframes, the device must be able to support generation of ILA on the 1st multiframe boundary after SYNC~ de-activation).

8.4 SYNC signal decoding (cont'd)

A high SYNC~ signal shall always be interpreted as a deactivated resynchronization request. In subclass 0 and 1 TX devices, the sampling of SYNC~ shall occur on the rising edge of the TX local frame clock. In subclass 2 TX devices, SYNC~ may alternatively be sampled by a higher speed detection clock (see 8.5 and 6.4.1.3), if so required, before being passed to the frame clock for processing.

Depending on the Deterministic Latency device subclass, the following delay requirements apply:

- Device Subclass 0: The delay between the rising edge of the local frame clock cycle upon which the transmitter ceases to send /K28.5/ symbols, and the appearance of the first bit of the first non-/K28.5/ symbol at the transmitter output shall be constant within the margins of the skew budget.
- Device Subclass 1 or 2: The delay between the rising edge of the local multiframe clock cycle upon which the transmitter transitions from /K28.5/ generation to ILA generation, and the appearance of the first bit of the first non-/K28.5/ symbol at the transmitter output shall be constant within the margins of the skew budget.

8.5 SYNC~ detection (device subclass 2)

The SYNC~ signal will be used to adjust the LMFC and frame clock of a Subclass2 deterministic latency device. SYNC~ detection will be done differently on ADC devices and TX logic devices.

A SYNC~ de-assertion is detected by using a detection clock to detect the high side of a low to high transition of the SYNC~ signal at the TX device. The accuracy of the detection is governed by the detection resolution of the TX device. In an ADC device, the frame clock and LMFC boundaries shall be adjusted such that they occur a deterministic number of adjustment clock cycles after the detection event. See 6.4 for clock and timing definitions, as well as timing diagrams for the resetting function.

In a TX logic device, SYNC~ de-assertion is detected and the calculated phase discrepancy between the detection phase and the TX logic device LMFC phase is used to adjust the DAC frame and LMFC boundaries. The required adjustment in the DAC is embedded in parameters PHADJ, ADJDIR, and ADJCNT found in ILA configuration registers 1 and 2, (see Table 21) and passed from logic device to DAC through the ILA sequence. This ILA is a result if either re-initialization over the data interface (see 5.3.3.7) or initial lane synchronization (see 7.4). Please see 6.4 for functional usage of measured information.

9 Device classification

As indicated in the Scope, there is a wide application range for this standard, varying from a single converter connected via a single lane, to multiple converter devices connected via multiple lanes per device. Converter manufacturers may decide not to support all configuration options in a certain device. The device classification defines which configurations are supported by a device.

The converter manufacturer shall classify each device for a JESD204 link according to the scheme of Table 22.

Table 22 — Device classification

Device class	Description	Support for lane alignment across different converter devices	Supported number of aligned lanes per converter device
NMCDA-SL	No Multiple-Converter Device Alignment, Single-Lane	No	1*
NMCDA-ML	No Multiple-Converter Device Alignment, Multiple-Lanes	No	≥2
MCDA-SL**	Multiple-Converter Device Alignment, Single-Lane	Yes	1
MCDA-ML**	Multiple-Converter Device Alignment, Multiple-Lanes	Yes	≥2

* A device classified as "NMCDA-SL" could in principle support multiple non-aligned lanes. In that case each lane is considered to belong to a separate instance of the JESD204 standard.

** Subclass 1 and Subclass 2 devices (see Table 25) shall always belong to one of the MCDA device classes.

A logic device shall support the requirements for all above classes within the limitations set by the number of available physical interfaces for the JESD204 application. It shall be possible to disable features in the logic device that are not supported by the device class of the connected converter device(s) in the application.

The features that are optionally supported in converters for certain device classes are summarized in Table **23** and Table 24. Note that it may be necessary to disable certain supported features in transmitters if the corresponding feature is not supported in the receiver and vice versa.

9 Device classification (cont'd)

Table 23 — Supported features in ADCs per device class

Feature	NMCDA-SL	NMCDA-ML	MCDA-SL	MCDA-ML
Transmission of initial lane alignment sequence	O	M	M	M
Transmission of lane alignment characters	O	M	M	M
Conformance to skew budget	O	M	M	M

Table 24 — Supported features in DACS per device class

Feature	NMCDA-SL	NMCDA-ML	MCDA-SL	MCDA-ML
Initial lane alignment	O	M	M	M
Lane alignment monitoring and correction	O	M	M	M
Lane alignment character replacement	O	M	M	M
Processing of initial lane alignment sequence	O	M	M	M
Inter-RX synchronization interface*	O	O	M	M
Re-initialization via data interface	O	O	M	M
* Applies only to Subclass 0 devices (see Table 25)				

9.1 Device Subclassification

In addition to the device classes defined in the previous section, a device subclass is defined in order to reference the device's ability to support deterministic latency across the JESD204 link. The converter and logic device manufacturer shall classify each device for a JESD204 link according to the scheme of Table 25.

Table 25 — Device Subclassification

Device Subclass	Description
Subclass 0	No Support for Deterministic Latency – Backwards compatibility with JESD204A[1].
Subclass 1	Deterministic Latency supported using SYSREF signaling.
Subclass 2	Deterministic Latency supported using SYNC~ sampling.

[1] True backwards compatibility is only achieved when the Subclass 0 device supports the LV-OIF-SxI5 Physical Layer variant.

Annex A (informative) Differences between JESD204B and JESD204A

This annex briefly describes most of the changes made to entries that appear in this standard, JESD204B, compared to its predecessor, JESD204A (2008).

1 Scope

- Figure1 updated, and text updated to outline differences in features between JESD204A and JESD204B (higher lane speed, support for deterministic latency, support for device clock input instead of frame clock input).

2 References

- OIF-TFI-5 reference removed.
- Reference 1 updated to 2008 version
- Reference 4 added.
- Reference 6 updated to 2008 version.
- References 8 & 9 added.

3 Terminology

3.1 Terms and Definitions

- Added definitions for clock generator, conversion clock, device clock, frame period, local clock, source clock, and SYSREF.
- Updated definitions for ‘frame’ and ‘sample clock’.
- Several clarifying notes added

3.2 Meaning of symbols and abbreviations

- Added definitions for CGS, ILA, PVT, RBD, and T_f.

4 Electrical specification

4.1 Electrical specification overview

- Section updated to include 2 new PHY variants: LV-OIF-6G-SR and LV-OIF-11G-SR based for operation up to 6.375 and 12.5 Gbps, respectively.
- Text added to clarify expected BER performance.

4.2 Compliance types

- Text updated to reflect 3 possible PHY variants instead of 1.

4.3 Interconnect

- Removed explicit reference to FR-4 PCB and removed insertion loss requirement, as this now appears in 4.3.1.

Annex A (informative)Differences between JESD204B and JESD204A (cont'd)

4.3.1 Interconnect Insertion Loss

- New. JESD204B now specifies an insertion loss mask instead of a single 6dB @ -.75 baud data point as was in 204A.

4.4 LV-OIF-SxI5 Data interface signals (Differential)
4.4.1 Compliance verification

- Compliance verification updated to include required test patterns.

4.4.2 LV-OIF-SxI5 Transmitter Electrical Specifications

- Text updated for clarification now that there are 3 PHY variants in use. NOTE 4 added to figure at the end of 4.4.2.

4.4.3 LV-OIF-SxI5 Receiver Electrical Specifications

- Text updated for clarification now that there are 3 PHY variants in use. NOTE4 added to figure at the end of 4.4.3.

4.5 LV-OIF-6G-SR Data interface signals (Differential)

- New. Performance requirements based on OIF CEI-2.0 6G-SR specification

4.6 LV-OIF-11G-SR Data interface signals (Differential)

- New. Performance requirements based on OIF CEI-2.0 11G-SR specification.

4.7 Device clock

- Frame Clock replaced by Device clock as the master clock input. Preliminary description of frame clock & multiframe clock added, as well as permitted relationships for the 3 subclasses.

4.8 Frame Clock, and Local Multiframe Clock.

- New.

4.9 SYNC interface

- Updated to specify that SYNC interface is now synchronous with the internal frame clock. Performance requirements and timing diagrams for SYNC based timing are now largely focused on Subclass 0 and Subclass 2 devices.

4.10 Lane-to-lane inter-device synchronization interface

- Updated to indicate 4.10 only applies for lane alignment across Subclass 0 devices.

4.11 SYSREF signal (Device Subclass 1)

- New.

Annex A (informative)Differences between JESD204B and JESD204A (cont'd)

4.12 Skew and misalignment budget

- Updated to change title from 'Skew budget' to 'Skew and misalignment budget'.
- Various new skew definitions added.
- Text rewritten to address subclass 0/1/2 performance as it relates to skew and misalignment.
- Skew unit changed from UI to SU, which is the largest of UI and 320 ps.
- Clock distribution skew limit has been updated. No change from JESD204A if device clock rate is equal to frame rate.

4.13 Control Interfaces (informative)

5.1.2 User data format for an independent lane

- Clarifications added to alleviate confusion from 204A surrounding the potential location of control words within the frame structure.

5.1.3 User data format for multiple lanes

- Updated to specify allowable values of CF. New examples and diagrams provided to clarify mapping options for control words on links with multiple lanes.

5.1.6 Test modes (transport layer)
5.1.6.1 General

- New.

5.1.6.2 Short transport layer test pattern

- New.

5.1.6.3 Long transport layer test pattern

- Updated to reflect that pattern length is max(M*S+2,4) frames.

5.2 Scrambling

- Clarifications added and diagram updated to indicate that scrambling function must be present on a per-lane basis, but that scrambling settings for all active lanes must match.
- The ability to (de)scramble is now mandatory for all devices

5.2.6 Scrambling disable

- Updated to remove the option of not providing (de)scrambling. The ability to (de)scramble is now mandatory for all devices.

5.3.3.1 Code group synchronization

- Updated to specify different operation for Subclass 0 vs Subclass 1 & 2 devices.

Annex A (informative)Differences between JESD204B and JESD204A (cont'd)

5.3.3.4.2 Character replacement without scrambling

- Added note to clarify that character replacement functions must be implemented for each lane.

5.3.3.4.3 Character replacement with scrambling

- Added note to clarify that character replacement functions must be implemented for each lane.

5.3.3.4.4 Frame alignment correction in the RX

- Several clarifying notes added

5.3.3.5 Initial lane synchronization

- Updated to require Subclass 1 & 2 DACs to use ILA of exactly 4 multiframes.

5.3.3.6 Lane alignment monitoring and correction

- Several clarifying notes added

5.3.3.7 Link re-initialization

- Clarifications made to 5.3.3.7 to consistently indicate that re-initialization over the data interface is mandatory for MCDA class devices, and optional for NMCDA class devices.

5.3.3.8 Test modes

- Note added to clarify that link layer test characters are never scrambled.

5.3.3.8.2 Test sequences

- Previous version of 5.3.3.8.2 deleted and replaced with updated text.

6 Deterministic Latency

- New clause.

7 Receiver Operation

- Receiver operation now defined in clause 7, instead of 5.3.4 as in JESD204A.

7.1 Code group synchronization

- Updated to specify different operation for Subclass 0 vs Subclass 1 & 2 devices.
- Added minimum resync request duration of 5 frames plus 9 octets.

7.4 Initial lane synchronization

- Updated to specify Subclass 1 & 2 devices will align incoming alignment characters to the LMFC within 4 multiframes.

7.5 Lane alignment monitoring and correction

- Updated description for function INITIATE_SYNC_CHECK.

Annex A (informative)Differences between JESD204B and JESD204A (cont'd)

7.6.4 Error reporting via SYNC interface

- SYNC~ error reports updated to be 2 frames in length for Subclass 1 & 2 devices, and must occur at the end of a multiframe.
- SYNC~ resync requests must last at least 5 frames + 9 octets.

8 Transmitter Operation

- Transmitter operation now defined in clause 8, instead of 5.3.5 as in JESD204A.

8.1 Synchronization

- Diagram updated to fix missing state transition requirement on right side of diagram. (Bug fix from JESD204A).
- Diagram and table updated to specify different operation for Subclass 0 and Subclass 1 & 2 devices.

8.2 Initial lane alignment sequence

- Diagram updated to provide simplification and clarity. No functional changes.

8.3 Link configuration data and encoding

- New fields ADJCNT, ADJDIR, JESDV, PHADJ, and SUBCLASSV added.

8.4 SYNC signal decoding

- Updated to specify different operation for Subclass 0 vs Subclass 1 & 2 devices and to outline the required width of SYNC~ pulses for resynchronization events.

8.5 SYNC~ detection (device subclass 2)

- New.

9 Device classification

- Table 22 updated to show that Subclass 1 & 2 devices always belong to MCDA device classes.
- Table 24 updated to indicate that the inter-RX synchronization interface applies only to Subclass 0 devices.

9.1 Device Subclassification

- Newe.

Annex B (informative) Application overview

- Previous Annex B deleted and replaced with new text. Text has major updates throughout to accommodate new clocking requirements for JESD204B and deterministic latency applications.

Annex A (informative)Differences between JESD204B and JESD204A (cont'd)

Annex C (informative) Device level implementation

- Previous Annex C deleted and replaced with new text. Text has major updates throughout to accommodate new clocking requirements for JESD204B and deterministic latency applications.

Annex E (informative) Instructions for Determining Linear Fitted Insertion Loss

- New Annex

Annex F (informative) Example of device clock and SYSREF generation

- New Annex

Annex G (informative) Clock Terminology

- New Annex

Annex B (informative) Application overview

B.1 Background

In many data converter applications, a side effect of advances in converter technology is that device packages, power consumption and circuit board area are now driven by interconnect and data transfer requirements, instead of the intrinsic needs of the application. High resolution data converters typically need more board space for the interconnection (board traces and vias) than for the chip package itself. Single-ended digital interfaces cause switching noise, which affects the performance of the analog part, and parallel differential-signaling interfaces (such as LVDS) exacerbate the board area cost of the interconnect.

While technological advances have led to significantly lower power consumption in the intrinsic part of converters, similar reductions in the I/O power consumption and the size of the chip package have not taken place because of the large number of I/O lines needed. These problems become even more evident when multiple converters are integrated in the same package, e.g. for direct conversion radio transceivers. This I/O pin count limit also affects multi-channel data acquisition applications, such as beam forming, in which signal processors synthesize the output of an array of A/D converters into a single data stream.

Various approaches, such as integrating converters with signal processing logic or multi-chip modules, have addressed this problem but have drawbacks where converter performance is critical. The optimal processes for the converter function and digital VLSI are different. Integrating the two can impose performance limits on mixed-signal design, increases the risk of coupling digital switching noise into sensitive analog circuitry and can add cost to the digital design. Over multiple generations of a system design, requirements for the two areas evolve independently. From a practical standpoint, the two functions are typically supplied by different manufacturers. For many applications, then, the preferred configuration remains separate ADCs, DACs and logic devices for driving the DAC and receiving data from the ADC.

Serialized differential links offer a solution in this case. Although such links have been widely adopted in data transfer applications, existing industry standards such as PCI Express provide services better suited to processors or dedicated interface devices. Such PC-centric interfaces impose unacceptable additional complexity and overhead on the simple, continuous, constant-bandwidth data stream considered here. What is needed is a definition of signals and data formats that enables front-end data acquisition systems to take advantage of high-speed serial links with minimal overhead.

The purpose of this specification is to allow manufacturers of converters (ADCs and DACs) and logic devices (ASICs and FPGAs) to guarantee interoperability between devices, which will promote reuse of components across applications and enable system integrators to independently select the best device for each function. Implementation of this standard only requires logic blocks that are already commonly available to both FPGA and ASIC designers. Standard logic libraries, PLLs and SerDes macros are sufficient to implement this standard.

The intended application is one with short, point-to-point links between devices, either on the same printed circuit board, across one or more impedance-controlled connectors, or across short-reach cables. The electrical layer signal generally represents what is commercially referred to as Current Mode Logic or "CML", terminated at 1.2 V. The intended interface is normally DC-coupled, but DC coupling is not mandated by the specification.

In the earlier versions of the JESD204 standards, the target data rate for a single lane was 0.3125 - 3.125 Gbps (raw bit rate), which supports a range of applications with varying requirements of converter resolution, sample rate, and number of channels. In multi-channel applications, several lower bandwidth channels are aggregated on a single link. In addition, the specification optionally supports links consisting of multiple lanes. Multiple lanes may be utilized when the converter data rate is higher than any single lane can accommodate or when lower rates are desired for reasons of simplicity or economy. Further, the specification optionally supports multiple data converters connected to the same logic device. Embedded alignment characters enable the receiver(s) to realign associated information across multiple lanes and multiple links.

In this current version - the JESD204B standard - the target data rates for a single lane have been extended up to 12.5 Gbps. As the bandwidths supported by future generation radios increase, the required net data throughput between the data converters and the logic device is expected to rise at a rapid rate. The 4X increase in the max data rate going from the JESD204A to JESD204B is expected to cover for this increased data throughput for the next few years.

The JESD204B addresses another key problem of the system designer – the problem of having to transmit clocks of different frequencies to different devices in the system. Such transmission causes not only complications in the clock generation scheme but is also prone to spurious effects from sub-harmonic energies existing on the board. The ideal scenario for a system designer therefore would be to transmit clocks of identical frequencies to all the devices in the system. If the different devices in turn have provisions to generate their desired internal clocks from this master clock, it would vastly simplify the problem of system clocking. In the context of the JESD204B standard, this would also imply that the frame clock (that earlier existed as a physical clock input to the devices in the prior versions of the standard), may no longer need to be a physical input to the device, and could be generated within the device from the master clock. The JESD204B terms such a master clock as device clock. The device clocks transmitted to all the converter devices are expected to be identical, and may or may not be identical to the device clock transmitted to the logic device.

The process of generating the internal clocks from the device clock may involve operations like clock division that require a mechanism for phase synchronization across chips. As an example, multiple ADC devices interfacing to the same logic device might generate their internal sampling clocks based on a process of division – if the phase synchronization between these divided clocks is not established, it would result in a relative skew between the sampling instants of the different ADCs. In applications like beamforming for instance, such a relative skew would not be tolerable. The JESD204B offers a solution to this problem of phase synchronization by exploiting the existence of a Local Multiframe clock (LMFC) within the JESD204 link layer. The LMFC period spans multiple frame periods, and should be chosen to be long enough so that the LMFC can act as a common timing reference between the various devices in the system. The JESD204B standard attempts to phase synchronize the LMFCs between all the devices through two different means in the two different subclasses, Subclass 1 and Subclass 2. In Subclass 1, the LMFCs are synchronized by a source synchronous signal called SYSREF that is transmitted to all the devices with controlled timing. In Subclass 2, the logic device generates its LMFC through a process of division of its device clock. The logic device then communicates the phase of its LMFC to the converter devices, each of which slave their LMFC phases to the common logic device master.

Annex B (informative) Application overview (cont'd)

Yet another key system issue addressed in the JESD204B standard is the problem of deterministic latency. In the context of multiple ADCs interfacing to a common logic device, deterministic latency is simply the following – at a particular instant, the frames appearing at the final output of the receiver should be relatable to a specific sampling instant of each ADC. These sampling instants for each ADC should, within margins of aperture error, correspond to the same absolute time instant. This relation between the ADC sampling instants and the frame-based output at the receiver output should be maintained the same over temperature and supply voltage, across devices, and should be repeatable over power-up cycles. For the case of a logic device transmitter interfacing to multiple DAC devices, deterministic latency implies a known deterministic relation between the frames going into the logic device transmitter and the analog output transitions at the output of the multiple DAC devices. Again, the JESD204B standard utilizes the LMFC to achieve deterministic latency.

For the case of multiple Subclass 1 ADCs interfacing to a Subclass 1 logic device, the role of the LMFC can be understood as follows.
The rising edge of SYSREF governs two events:

Event 1: The first sampling instant inside each ADC device that is passed on to the link – let us call this S_0. We can refer to subsequent samples as S_1, S_2, S_3,… S_N,…

Event 2: Determination of the LMFC phase inside each ADC device

Now, the LMFC phase as governed by Event 2 will eventually result in /A/ characters emanating from each lane as part of the ILA. The logic device, will through the mechanism of lane-specific elastic buffer delay control, "stretch" the received /A/ character in each lane to coincide with its LMFC edge, which is common for all the lanes. Since the above mechanism aligns the final positions of the /A/ characters on each lane, it automatically aligns the last sampling instant transmitted in that LMFC period. Thus, deterministic latency on all lanes is achieved provided the ADC device manufacturer can unambiguously map the data samples S_N to their relative positions within the LMFC.

The mechanism for achieving deterministic latency as described above involves controlling timings across multiple clocking domains, including the sample clock and frame clock domain. However, not all of it comes under the purview of JESD204B. For example, the generation of the sample clock and frame clock from the device clock are governed by synchronizing mechanisms specified in the JESD204B. However, operations such as sampling the analog input, conversion from the analog to digital domain and handing off this digital data to the frame clock domain would not be under the purview of the JESD204B, but still need to be done in a manner that is in overall alignment with the mechanism of deterministic latency described above.

The size of the elastic buffers required to align the /A/ characters on all lanes to the LMFC edge of the receiver depends on the value of *K* (the number of frames per multiframe), and may turn out to be unacceptably large especially if a large value of *K* is chosen. Also in some applications, the JESD204B interface might be part of a signal chain (for example, an automatic gain control loop) where the total latency might need to be constrained to some maximum value. In such cases, the JESD204B standard has a provision that enables achieving deterministic latency, but with a minimum possible extra impact on the total link latency. The /A/ characters are, in such a case, stretched until a virtual delayed LMFC edge that is RBD frames delayed from the receiver LMFC edge.

Annex B (informative) Application overview (cont'd)

The value of RBD needs to be chosen based on an estimate of when the latest /A/ character might enter the elastic buffer. Such an estimate would depend on the latency and (maximum) propagation delay in the transmitter, the (maximum) delay across the interconnect and the latency and (maximum) delay in the receiver till the input to the elastic buffer. The depth of the elastic buffer depends on the difference between the maximum and minimum value of this total delay.

The specification may be used for many applications, but its features are driven by the interface requirements between converters (both ADCs and DACs) and digital logic devices (FPGAs and ASICs). The standard provides a full electrical and logical specification for the interface. Where the interface imposes secondary requirements on areas outside the scope, such as primary clock inputs, those parameters are described. Control signals that invoke special device operating modes, such as power-down mode, are not part of this specification. Such control signals are typically communicated via a separate control interface, but some functions of the receiving device may also be controlled via dedicated control bits in the sample stream, at the discretion of the device implementer. Another use of control bits in the sample stream is to indicate the status of the transmitter device.

B.2 Link properties

While in JESD204A, a physical frame clock was required to be supplied as a common timing reference to all devices connected to the link, this requirement is removed in the JESD204B standard. Both the frame clock and the sample clock can be internally generated from the device clock that is provided to each device. Since most converter applications demand a low jitter sample clock to minimize noise, the device clock provided to the converter should have this desired low jitter. Also in low jitter applications, it would be desirable to generate the sample clock from the device clock through a simple process of clock division.

In the logic device, the frame clock (again generated from its device clock) is generally used to sequence the operations in the application-specific logic, as well as to synchronize the receiver SerDes block. In addition, as outlined in Section B.1, the LMFCs of both the converter and the logic device are generated by dividing the device clock. To synchronize the generation of all these internal clocks from the device clock, the timing of the SYSREF (in Subclass 1) and SYNC~(In Subclass 2) needs to be accurately controlled with respect to the device clock in such a manner that the device clock edge sampling these signals is unambiguously fixed by the system designer. Apart from careful considerations to the generation of these signals relative to the device clock, the propagation of these signals to the various devices would also require careful consideration like matching their interconnect lengths and loading with that of the device clock.

Since SYSREF is a source synchronous signal which can be generated in an accurate phase aligned manner with the device clock, it is expected that a system designer aiming to operate at device clock rates higher than 500 MHz would prefer to use a Subclass 1 approach. For this reason, it is recommended that the pin configuration of a Subclass 1 device would have the SYSREF pin in close proximity to the device clock pins. It is also recommended that the electrical characteristics of the SYSREF signal be identical to that of the device clock. While SYSREF is expected to be a much slower clock than the device clock, its rise/fall times are expected to be similar to that of the device clock. As a result, the signal integrity of the SYSREF signal might be as much of a concern as that of the device clock.

Annex B (informative) Application overview (cont'd)

A related concern to be kept in mind while controlling the timing of the SYSREF signal relative to the device clock is the fact that SYSREF, by virtue of being a slow signal would have logic high and low voltage levels that would be close to the saturation levels. In comparison, a high speed device clock is more likely to have logic high and low voltage levels that are further away from the saturation levels. This can lead to a mismatch in the rise/ fall times of the SYSREF signal relative to the device clock, and therefore needs to be considered while analyzing the setup and hold times available for the device clock edge to latch the SYSREF. In a case where SYSREF is provided as a low frequency periodic signal, the data converter vendor as well as the system implementer would do well to analyze the spurious effects from having such a sub-harmonic clock, and minimize its effects on the converter and system performance.

The sampled converter data is sent over a serial data interface as a continuous, 8B/10B-encoded data stream. Control characters are used for synchronization. The data interface consists of one or more lanes, i.e. pairs of differential signals. The collection of lanes and circuitry connecting two devices to each other is called a link. Several links connecting to the same logic device can be combined into a multipoint link on which all lanes are time-aligned. The electrical characteristics of the data interface are compatible with current mode logic (CML).

Since the JESD204B standard specifies data rates much higher than the 3.125 Gbps limit in the JESD204A standard, techniques like pre-emphasis might be required in the transmitter to reduce the effects of Inter Symbol Interference (ISI) at these higher data rates especially for long interconnects. Pre-emphasis boosts up the energy of the high frequency components in the transmitted CML signal, and partially compensates for the increased losses to the high frequency components in the channel. On the receiver side, equalization might need to be implemented at high data rates and/or long interconnect lengths in order to open the eye mask sufficient enough to meet the receive eye mask specification.

The receiver drives a separate SYNC~ signal to the transmitter. The SYNC~ signal is used by receivers in the initial synchronization process, to indicate errors and to request re-initialization of the link if an error condition cannot be corrected. Distinguishing between errors and re-initialization requests imposes requirements on SYNC~ timing with respect to the frame clock,. The SYNC~ is meant to be a time critical signal in JESD204A on links with multiple transmitters (as also in Subclass 0 of JESD204B). In order that the /A/ characters between all transmitters are aligned, all transmitters shall see the rising edge of SYNC~ in the same frame period. In addition, the SYNC~ signal also doubles up as a carrier of timing information between the receiver and the transmitter in Subclass 2 and so careful consideration needs to be given to the timing of generation of the SYNC~ inside the receiver, and its propagation to the transmitter(s).

In JESD204A, the synchronization between multiple DAC devices required a lane-to-lane, inter-device synchronization interface. Because this interface is only used between similar devices from the same converter vendor, its further specification was left open. The inter-device synchronization interface was expected to operate at the frame clock rate, and the device user was not required to generate a separate clock signal for this interface. In JESD204B, the synchronization between multiple DAC devices happens through the synchronization of their LMFCs as determined by SYSREF in Subclass 1, and SYNC~ in Subclass 2. In Subclass 2, the LMFCs of the various DAC devices could, on power up, be misaligned with respect to each other. This would result in a phase misalignment of the SYNC~ signals coming from each of the DAC devices with respect to the LMFC of the common logic device. The logic device can, by a process of sampling each of the SYNC~ signals, register the phase mismatch in each of the DAC device LMFCs, and transmit this information to each DAC through the ILA.

Annex B (informative) Application overview (cont'd)

Each DAC device can then adjust its LMFC phase to match with the common LMFC of the logic device. When the device clock is used for sampling the SYNC~, the timing of each of the SYNC~ signals relative to the device clock is critical to establish synchronization of the DAC device LMFCs. So the transmission of the SYNC~ signal from each of the DAC devices to the logic device should be done with matched delays, and in a manner that each of the SYNC~ signals can be latched by the same edge of the logic device clock.

Although not specified in the standard, individual devices may support an additional control interface to allow configuration and monitoring of the data interface. This additional control interface may be necessary because various link parameters as defined in this standard must be set to allow proper operation of devices on these links.

When one or more converters inside a device are in power-down mode, the link can still be kept active and synchronized. This state is called idle mode. In idle mode, samples of inactive converters are replaced by dummy samples. On the link level, no difference can be seen between an active and an inactive converter. The presence of dummy samples is indicated to the receiver via the control interface or via dedicated control bits in the sample stream.

When all converters in a device are inactive for a long period of time, it may be advantageous for the link to enter a shut-down mode. Operation in shut-down mode is similar to the continuous mode, but with power-down and power-up procedures applied between the periods of activity. These power-down and power-up procedures are not part of the current specification and their implementation is left to the application.

B.3 Variants and modes

The interface supports a wide range of data converter devices over single and multiple serial links. In addition, each link may consist of a single lane or multiple lanes as needed to support the required data throughput, Each lane contains synchronization and alignment data that allows the data to be reassembled in the receiver, effectively rebuilding the structure of the original converter data. Although one or more converter devices are supported by this standard, only one logic device is supported.

The configuration variants can be classified according to the following scheme:

1. Single vs. multiple converters per converter device.
2. Single lane vs. multiple lanes per converter device.
3. Single converter device vs. multiple converter devices connected to one logic device.

The interface can work in one of the following modes:

1. Fundamental mode vs. oversampled mode.
2. Fundamental mode vs. interpolating or decimating mode.
3. Data mode vs. test mode.
4. Active mode vs. idle mode.

Annex B (informative) Application overview (cont'd)

In the fundamental mode, one sample is transmitted per frame clock cycle from each converter over the link. In the oversampled mode, two or more samples per converter are transmitted per frame clock cycle. In the oversampled mode, the sample clock is an integer multiple of the frame clock. In the interpolating or decimating mode, the sample clock is also an integer multiple of the frame clock, but the samples are interpolated or decimated in the converter, so that only one sample is transmitted per frame clock cycle from each converter. The interpolating and decimating modes do not affect the data mapping on the interface, but they do affect the clocking scheme. Test modes can be implemented on the link level, e.g. by transmitting a stream of synchronization characters, or on the transport layer level, by transmitting a stream of test samples. In the idle mode one or more converters in a device may be shut down and their samples are replaced by dummy samples.

In addition to the above defined modes and variants, the JESD204B standard also defines three Subclasses. These Subclasses mainly differ in the mechanism they use to achieve deterministic latency. While Subclass 0 does not provide any such mechanism, Subclass 1 utilizes the source synchronous SYSREF signal and Subclass 2 utilizes the system synchronous SYNC~. A JESD204B device might be configurable through software to operate in one or more of these Subclasses. The Subclass information of a transmitter is conveyed to the receiver through the link configuration data as part of the ILA.

B.4 Configuration examples

B.4.1 General

The following examples illustrate a variety of configuration variants. While specified for data converters, the standard is also suitable for all systems that transmit real-time streaming data between devices. Therefore, the configurations outlined below provide a few examples of the many possibilities.

B.4.2 Single Device ADC Application

Figure B.1 illustrates a typical implementation of the interface as it applies to a single ADC device communicating with a single logic device. With this configuration, a single package contains one or more ADCs, all of which are sampled at the rate of the frame clock (which is generated from the ADC device clock). As described earlier, the converters' output data is aggregated by the JESD204 TX block and is sent across one or more lanes to the receiving logic device.

The logic device recovers the clock and de-serializes the data in the JESD204 RX block. If multiple lanes are involved, the logic device RX block is also responsible for ensuring alignment of the data streams being sent to the application logic.

While all three of Subclass 0,1 and 2 allow clocking the devices with a device clock (rather than needing a physical frame clock), Subclass 1 and 2 additionally have provisions for achieving deterministic latency. In Subclass 1, a 0->1 transition on the signals marked as TX SYSREF and RX SYSREF are sampled by the ADC Device clock and the Logic Device clock respectively, thereby synchronizing the LMFC phase inside each of the devices.

Annex B (informative) Application overview (cont'd)

The SYNC~ signal assists in initial synchronization and can be used to re-initialize the link under certain error conditions – and in Subclass 2 also serves as a carrier of timing reference information from the logic device to the ADC.. In Subclass 2, the logic device acts as the timing master, with its LMFC phase information conveyed to the ADC through the SYNC~ deassertion.

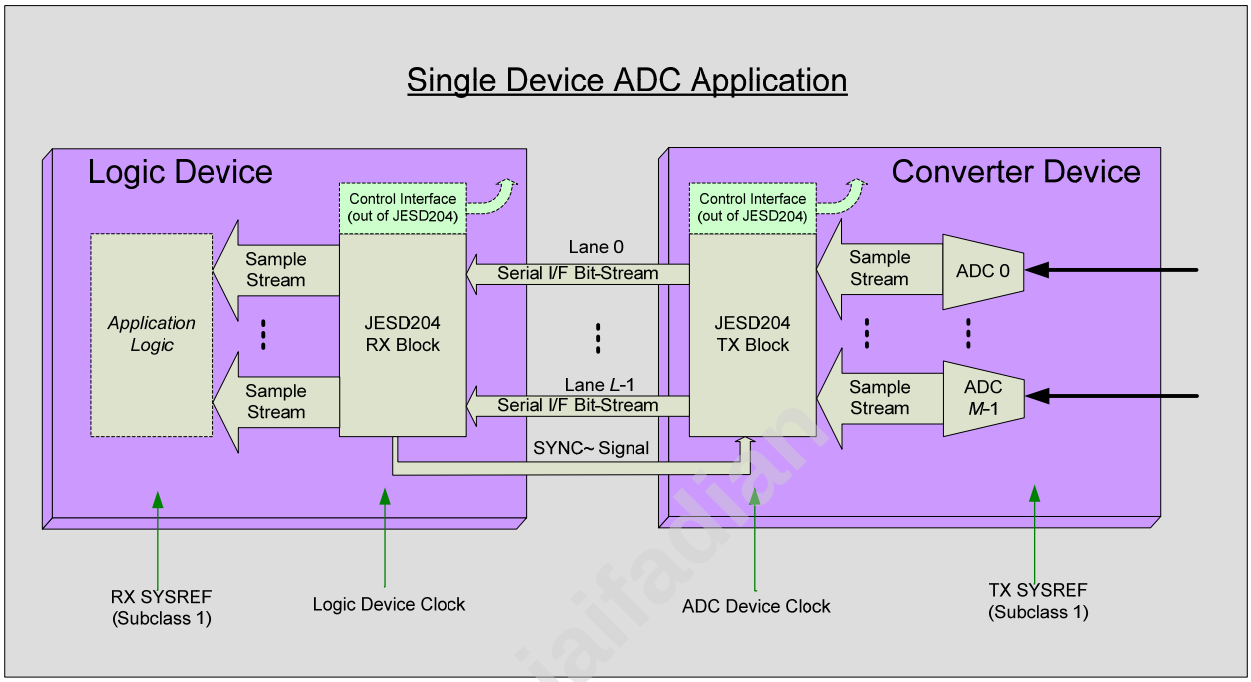


Figure B.1 — Single Device ADC Application

B.4.3 Single Device DAC Application

Figure B.2 shows an application using multiple DACs in a single package. In this application, the logic device provides DAC samples using one or more lanes. In the DAC device's JESD204 RX block, the data is de-serialized and provided to the DACs.

While all three of Subclass 0,1 and 2 allow clocking the devices with a device clock (rather than needing a physical frame clock), Subclass 1 and 2 additionally have provisions for achieving deterministic latency. In Subclass 1, a 0->1 transition on the signals marked as TX SYSREF and RX SYSREF are sampled by the Logic Device clock and the DAC Device clock respectively, thereby synchronizing the LMFC phase inside each of the devices.

The SYNC~ signal assists in initial link synchronization, to report decoding errors, and to restart the synchronization process when necessary. In Subclass 2, the SYNC~ deassertion also carries timing information of the LMFC of the DAC to the logic device. In Subclass 2, the logic device acts as the timing master, and compares the LMFC phase of the DAC device (as interpreted from sampling the SYNC~) with its own LMFC phase. In case of a mismatch in the LMFC phase, the mismatch information is conveyed to the DAC through the ILA.

Annex B (informative) Application overview (cont'd)

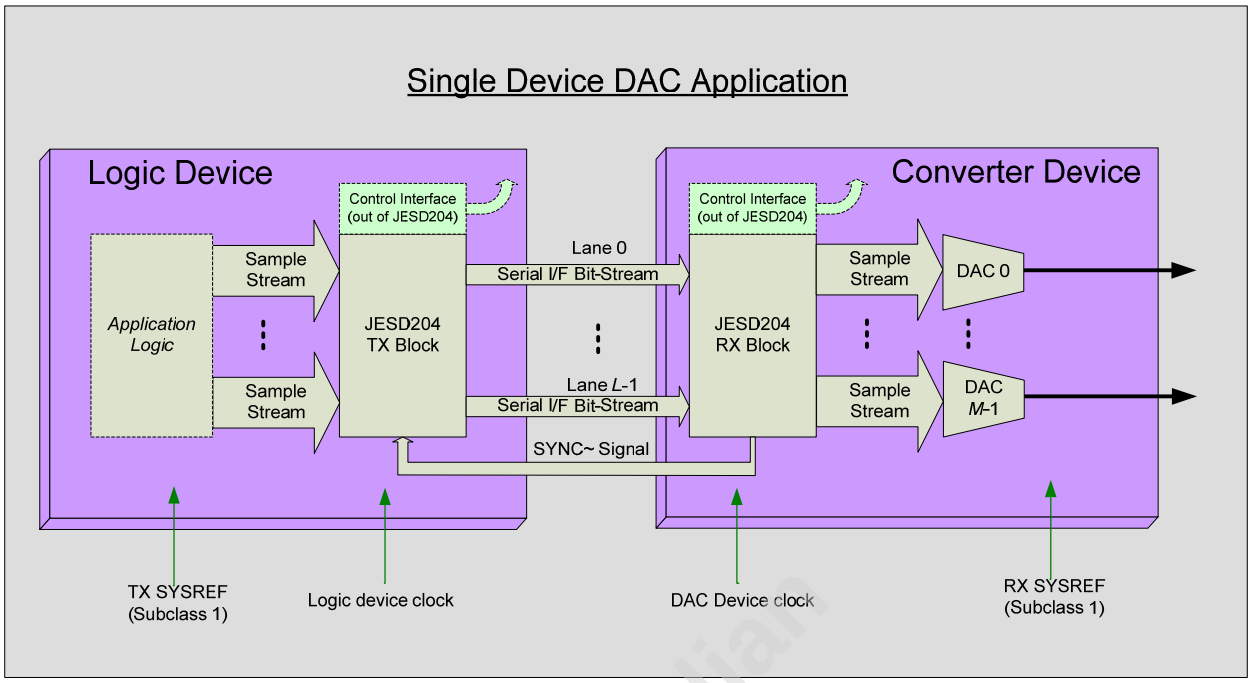


Figure B.2 — Single Device DAC Application

B.4.4 Multiple Device ADC Application

While the standard only addresses single logic devices, it supports systems using multiple converter devices. As shown in Figures B.3a and B.3b, multiple ADC devices may be interfaced to a single logic device. In addition, each ADC device may attach to the logic device using one or more lanes. For this multi-ADC application to operate properly, the SYNC~ signal(s) from the logic device must be connected to each ADC device. The SYNC~ signal(s) allows the logic devices to synchronize all ADC devices at power up and when faults occur. In all other ways, this multi-ADC application functions similarly to the single-ADC application. In JESD204A (and Subclass 0 of JESD204B), it is necessary to send a common SYNC~ signal to all ADCs to ensure simultaneous transmission of the ILA sequence from all the ADC devices. The scheme for such a common combined SYNC~ signal is shown in Figure B.3a. In Subclasses 1 and 2 of the JESD204B, it is possible to send a separate SYNC~ signal to each ADC device. In these two subclasses, it is not necessary that all converter devices send their ILA sequences simultaneously, because the alignment between the links is ensured via the mechanism of synchronizing their LMFCs. A non-combined SYNC~ scheme is shown in Figure B.3b.

The signal marked as ADC Device Clock is to be transmitted in a similar manner to each of the ADC devices. In Subclass 1, the signal marked TX SYSREF is also to be transmitted in a similar manner to all the ADC devices, such that its 0->1 transition can be sampled at the same ADC device clock edge of each of the ADCs.

Annex B (informative) Application overview (cont'd)

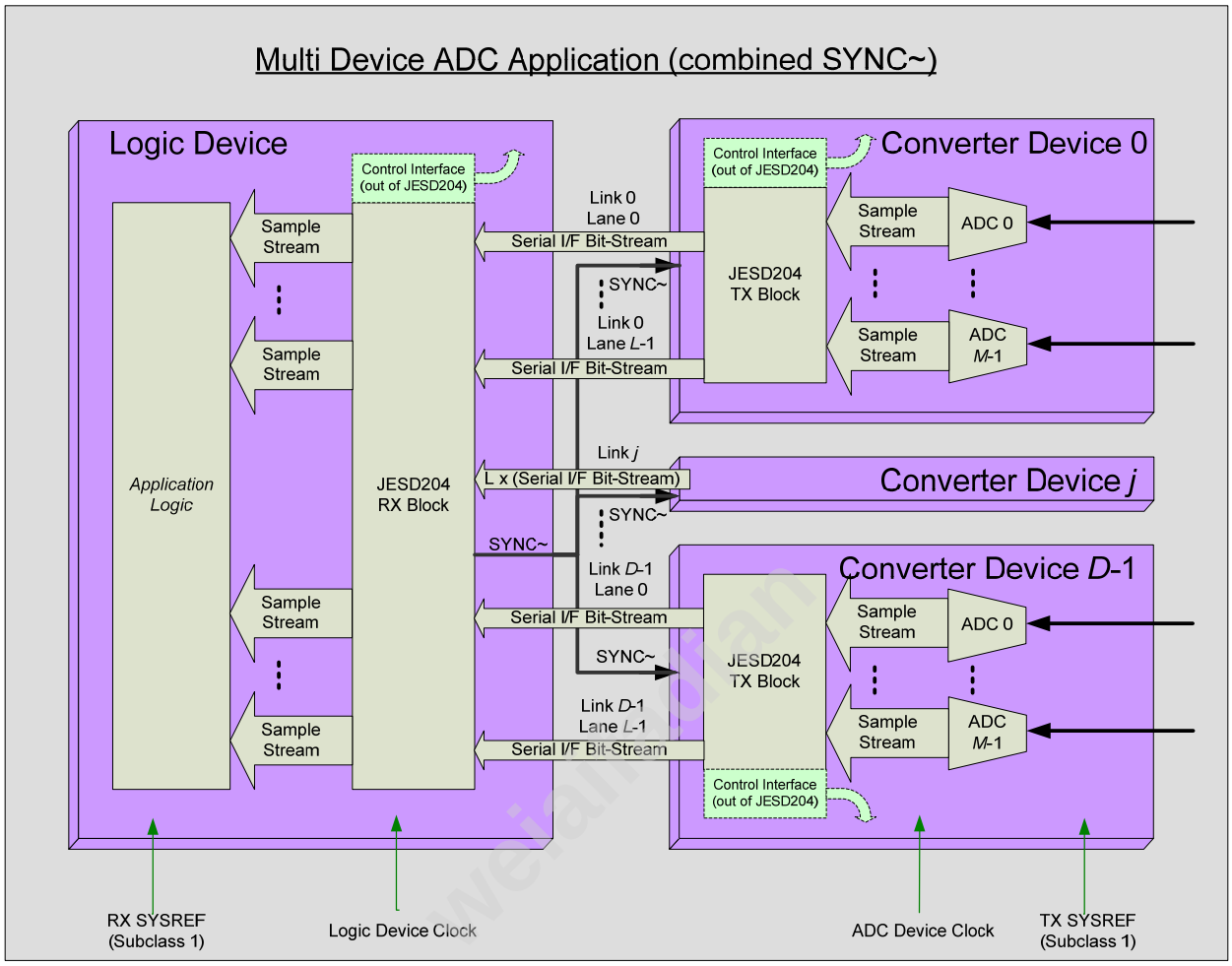


Figure B.3a — Multiple Device ADC Application with a combined SYNC~ scheme

Annex B (informative) Application overview (cont'd)

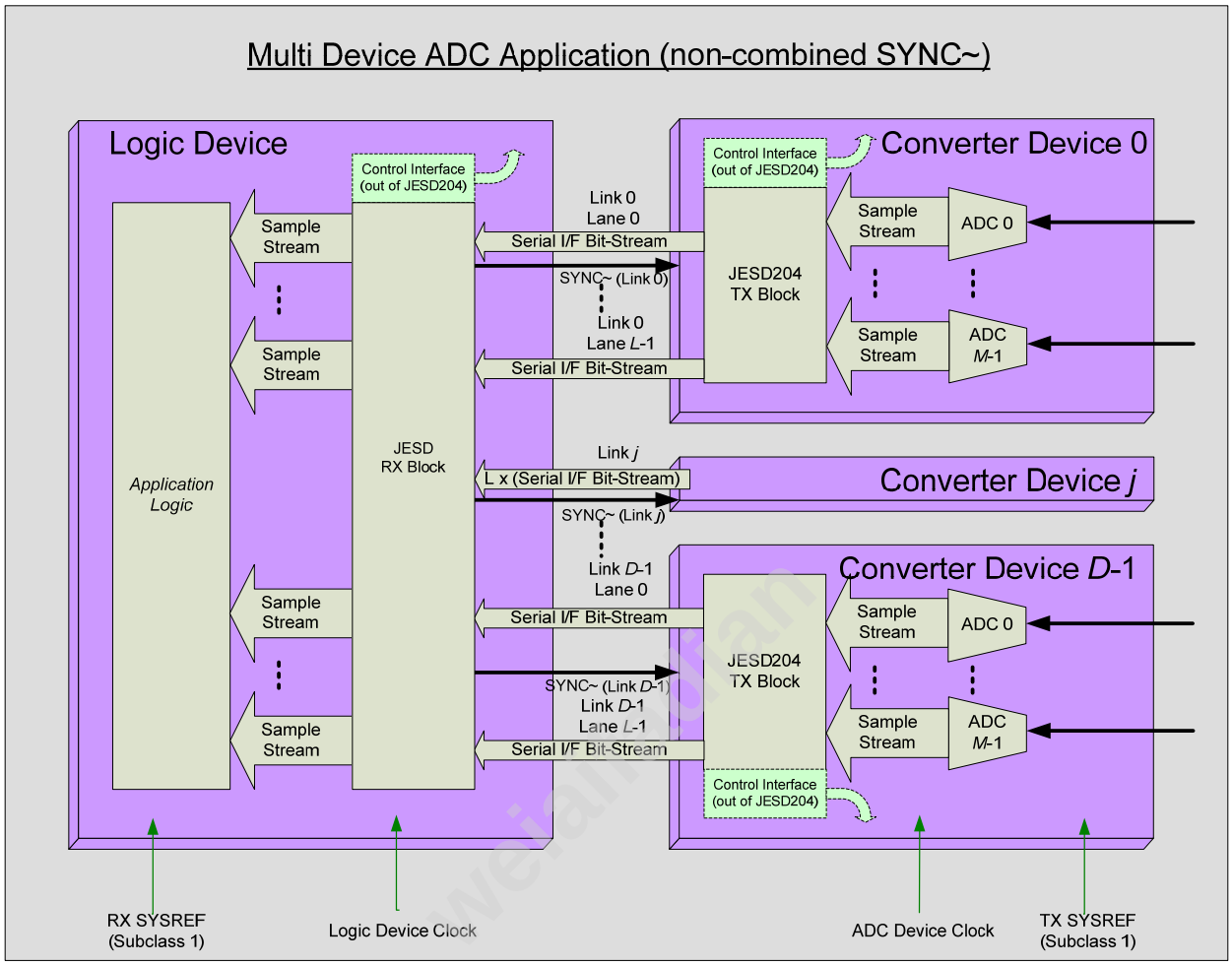


Figure B.3b — Multiple Device ADC Application with a non-combined SYNC~ scheme

B.4.5 Multiple Device DAC Application

A Multiple-Device DAC application is shown in Figure B.4. Inside the logic device, the SYNC~ from the various DAC devices may be optionally combined. Such a combined SYNC~ scheme is required in JESD204A and Subclass 0 of JESD204B. SYNC~ combining is not necessary for Subclass 1 and Subclass 2 devices. For Subclass 2 devices, if SYNC~ combining is used in the system, SYNC~ de-assertion detection must take place prior to SYNC~ combination. This is necessary so that the logic device is able to ascertain the timing of the LMFCs of the individual DAC devices by a process of sampling the incoming SYNC~ of each DAC device using its logic device clock.

As usual, samples to all DACs come from a single logic device. DACs request synchronization and report detected error conditions to the logic device using the SYNC~ signal. Each of the multiple DAC devices has its own SYNC~ signal into the logic device. The data paths leading to the DACs may exhibit various latencies. Many applications require precise phase alignment between the analog outputs of two or more DAC devices. Lanes can be aligned across different DAC devices using similar principles as used for alignment inside a single DAC device.

Annex B (informative) Application overview (cont'd)

While in JESD204A and JESD204B Subclass 0, this will require a dedicated lane-to-lane synchronization interface between the DAC devices, Subclass 1 and 2 avoid the need for such an interface.

In Subclass 1, the signal marked RX SYSREF is to be transmitted in a similar manner to all the DAC devices, such that its 0->1 transition can be sampled at the same DAC Device Clock edge of each of the DAC devices.

In Subclass 2, after power up, the LMFC of each of the DAC devices can come up with random phases relative to the other. The logic device samples the SYNC~ signals from each of the DAC devices, and through the ILA communicates how these phases compare to the phase of its own LMFC. Each DAC device can then adjust its LMFC such that it aligns with the LMFC of the logic device, and thereby aligns with the other DAC devices.

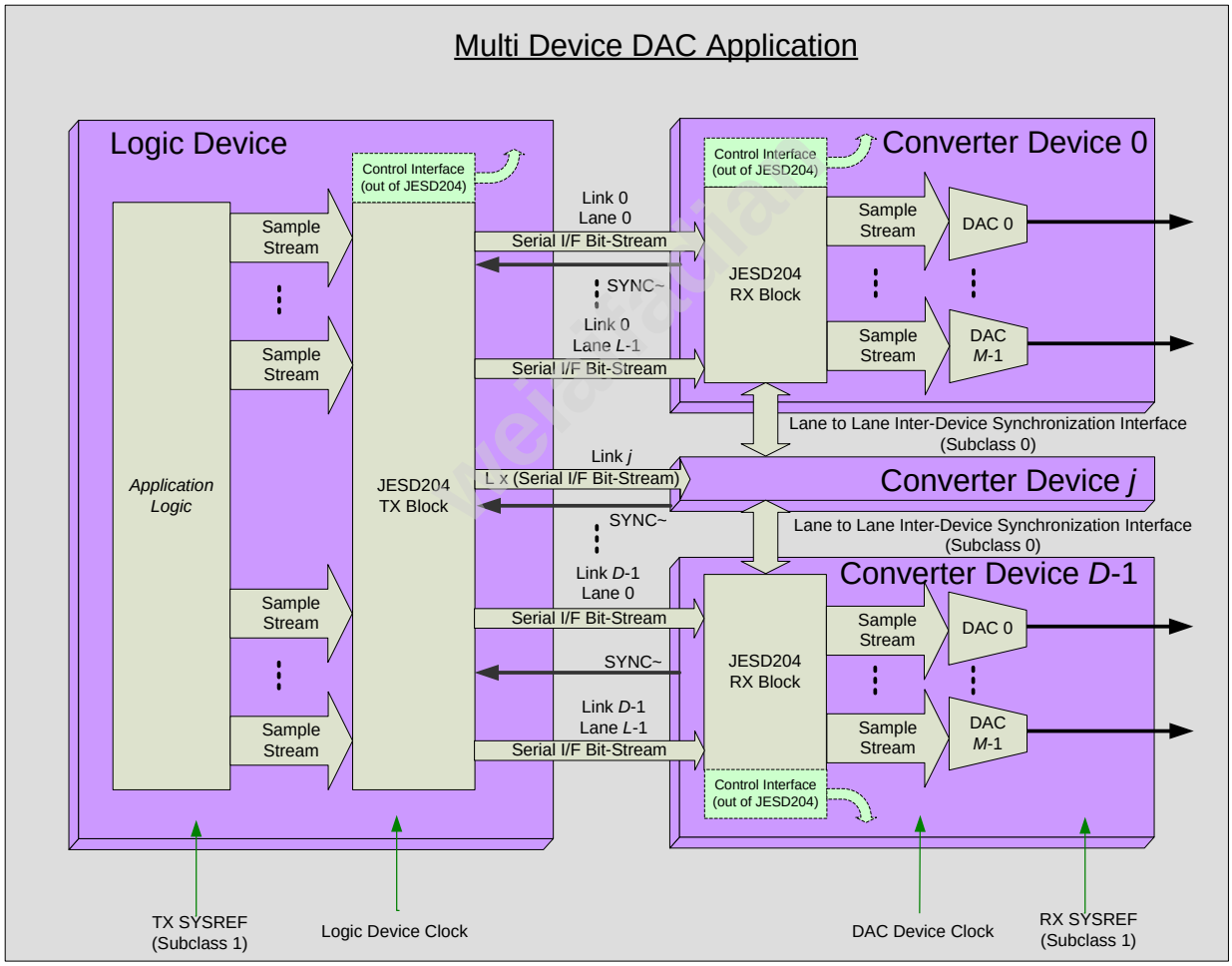


Figure B.4 — Multiple Device DAC Application

Annex B (informative) Application overview (cont'd)

B.4.6 Interface for Decimating or Oversampled ADC

Decimating ADC devices are very similar to standard ADC devices. In an application using the JESD204 standard to transmit ADC samples, the only difference between a non-decimating ADC and a decimating ADC device is the existence of an additional clock domain in the logic. Instead of the sample clock being the frame clock, the decimated data rate is the frame rate. Therefore, the sample rate must be higher than the frame rate by the decimation factor. Both the sample clock and frame clock are derived from a common ADC device clock. If the ratio between the character clock for 8B/10B encoding and the sample clock is a non-integer, spurious tones may be generated in the converter and thus limit its performance. Therefore when specifying and designing these devices, great care must be taken to ensure that full performance can be met within the allowable range of oversampling and decimation factors.

An ADC having an oversampled interface is very similar to a decimating ADC. The only difference is that all samples of an oversampling ADC are actually sent over the link. Both types of ADCs can be described by the same block diagram of Figure B.5.

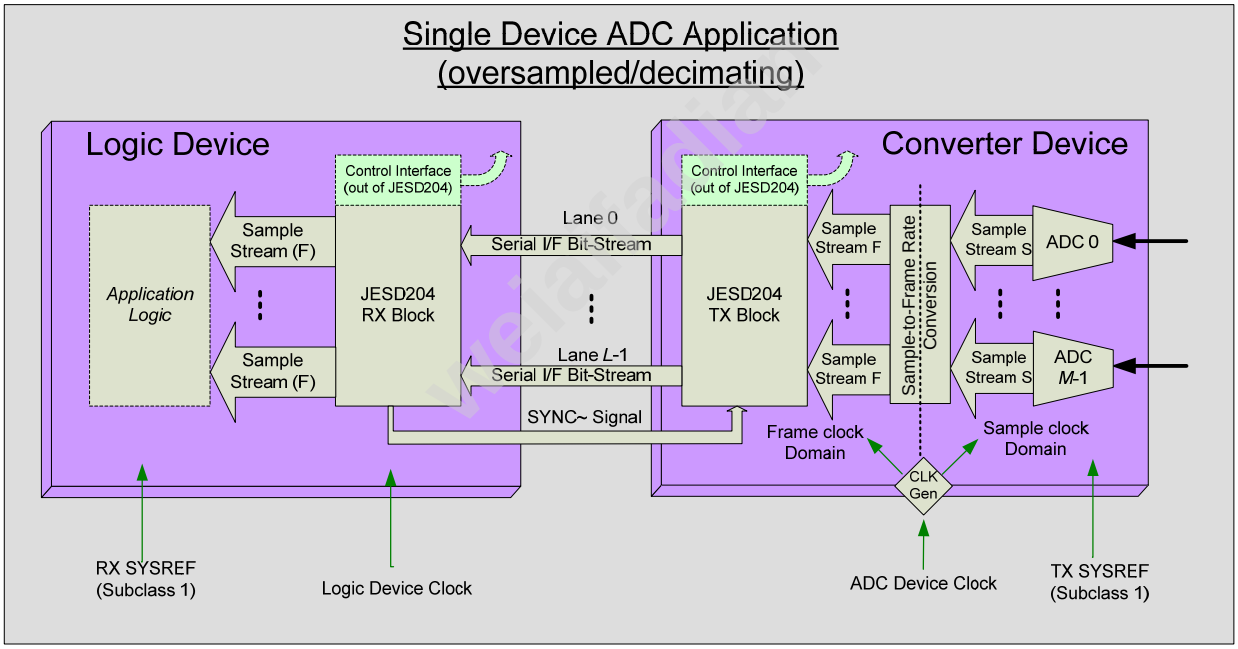


Figure B.5 — Single Device Interface for a Decimating or Oversampled ADC

Annex B (informative) Application overview (cont'd)

B.4.7 Interface for Interpolating or Oversampled DAC

As with decimating ADC devices, interpolating DAC devices include an additional clock domain. The frame rate data leaving such a DAC's JESD204 RX Block is interpolated to the desired rate. As with decimating ADCs, the interpolation rate must be chosen to avoid the generation of unwanted spurs. The data to an interpolating or oversampled DAC must be retimed to ensure proper transfer across the boundary between frame clock and sample clock domain. This transfer is indicated in Figure B.6.

The DAC with oversampled interface is very similar to the interpolating DAC. The only difference is that all samples for an oversampled DAC are actually transmitted over the link. Both types of DACs can be described by the same high-level block diagram of Figure B.6.

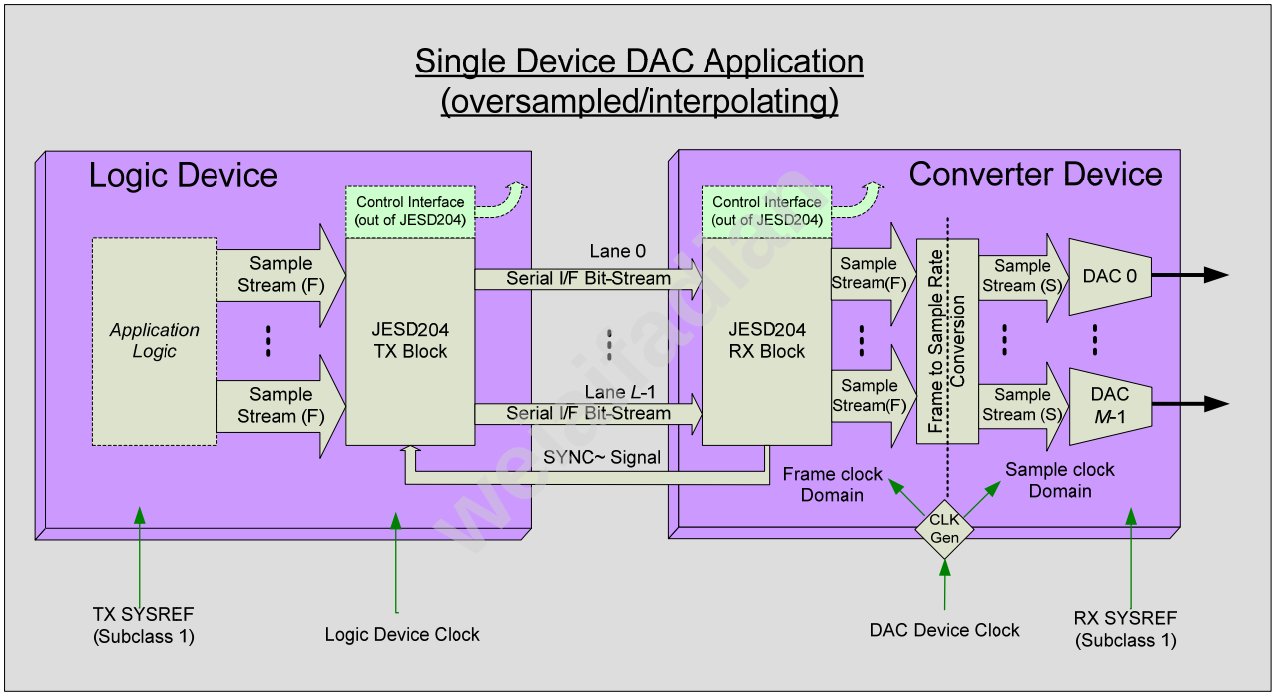


Figure B.6 — Single Device Interface for a Decimating or Oversampled DAC

Annex C (informative) Device level implementation

C.1 Transmitter block

C.1.1 Generic structure

The purpose of the transmitter block is to take one or more digital sample streams and convert them to one or more serial streams. Figure C.1 shows the generic transmitter structure of a single link. The same structure can be used for both an ADC converter device and a logic device.

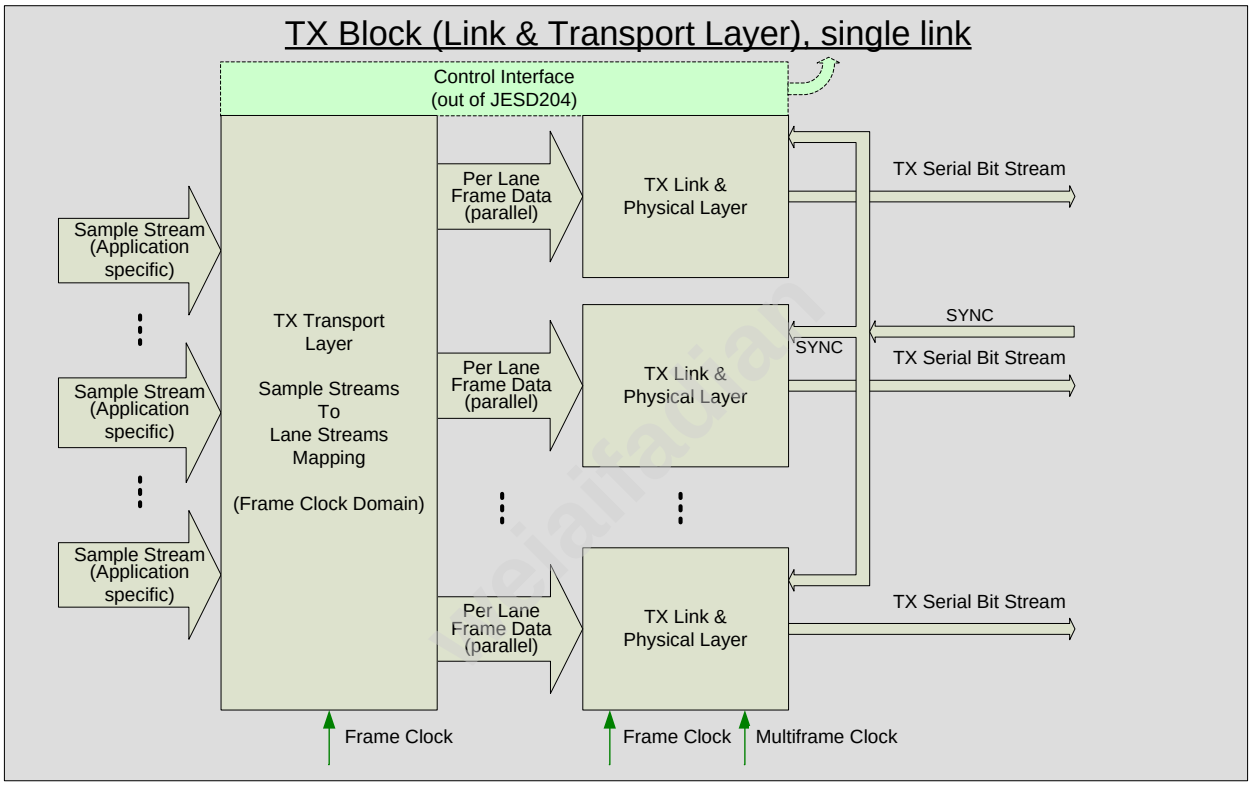


Figure C.1 — Generic structure of a TX block, single link

The functionality of the transmitter device can be divided into a transport layer and one or more link and physical layers. In the transport layer, the incoming stream or streams of samples are mapped to one or more parallel lanes with frame data. Each frame consists of a fixed, application-specific number of octets. The JESD204 transport layer is specified in subclause 5.1. On the link layer, frames are encoded as a stream of 8B/10B symbols, which are transmitted via the physical layer as a serial bit stream across the interconnect. The link layer is specified in subclause 5.3. Each TX link layer monitors the input SYNC~ signal from the RX device. If the SYNC~ signal is active for at least four frame clock cycles, all transmitters in the link enter synchronization mode until the SYNC~ is disabled by the RX device.

On a multipoint link, where one logic device sends aligned data to multiple DAC devices, subclass 0 requires to combine the SYNC~ signals of all DAC devices in the TX device. This is shown in Figure C.2. SYNC~ combining ensures that all TX link layers will send the initial lane alignment sequence simultaneously. In subclasses 1 and 2 it is possible to align the individual links of a multipoint link via the LMFC and SYNC~ combining is not necessary. A non-combined SYNC scheme is shown in Figure C.3.

Annex C (informative)Device level implementation (cont'd)

Optionally SYNC~ combining may also be used in subclasses 1 and 2. However, in subclass 2 special care must be taken that SYNC~ combining does not prevent detection of the phases of individual LMFCs in the DAC devices, see 6.4.2.2.

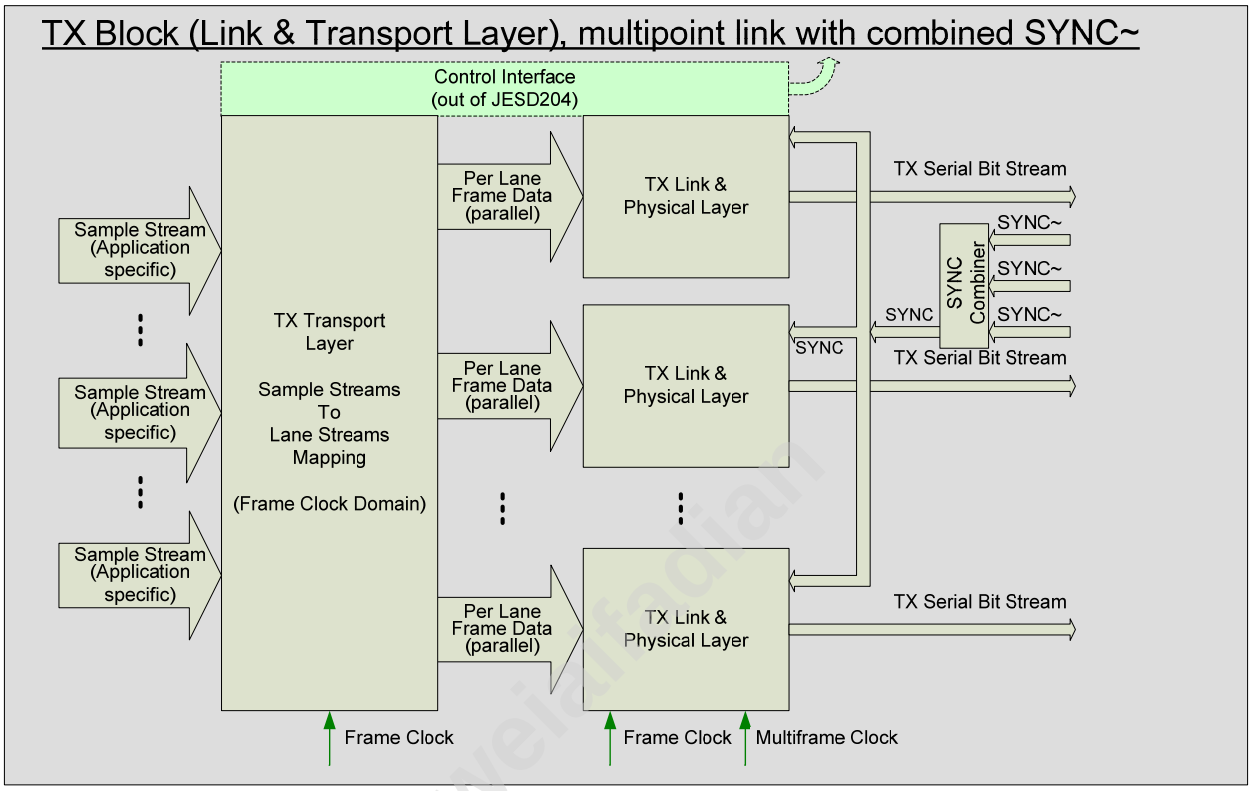


Figure C.2 — Generic structure of a TX block, multipoint link with SYNC~ combining

Annex C (informative)Device level implementation (cont'd)

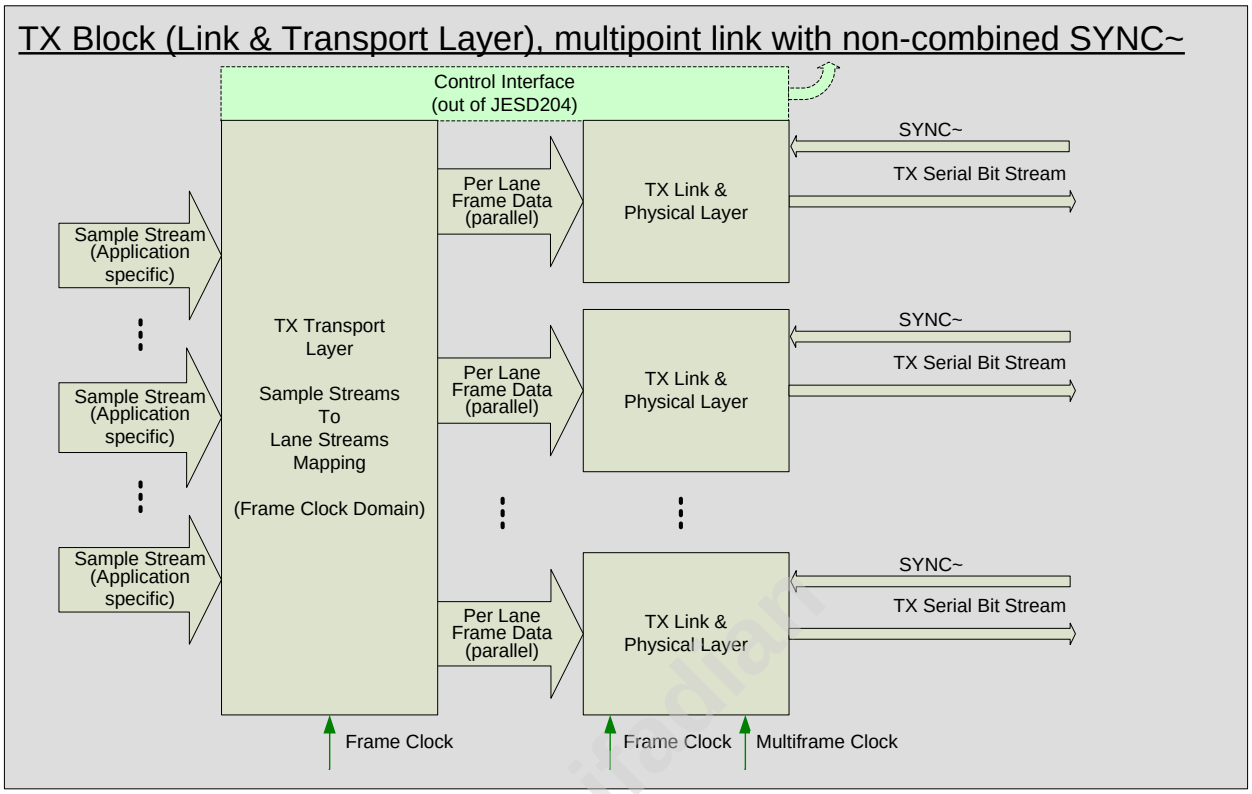


Figure C.3 — Generic structure of a TX block, multipoint link without SYNC~ combining

C.1.2 TX Link Layer

Figure C.4 outlines the TX Link Layer of the JESD204 standard. The frame data is first converted to an octet stream. Optionally, the data can be scrambled. The purpose of the scrambling is to avoid the presence of spectral lines in the signal spectrum. If properly implemented, such scrambling can decrease the analog performance degradation of the overall system. In Figure C.4, the scrambler operates on octets, but the scrambler can alternately be implemented in the frame clock domain. After optional scrambling, alignment characters are substituted for the purpose of monitoring frame and lane alignment in the receiver. The alignment characters replace the original data symbols in such a way that the receiver can still reconstruct the original data symbols. At link initialization, an initial lane alignment sequence is transmitted. All data is encoded as 8B/10B symbols and subsequently passed to the physical layer for serialization.

Annex C (informative)Device level implementation (cont'd)

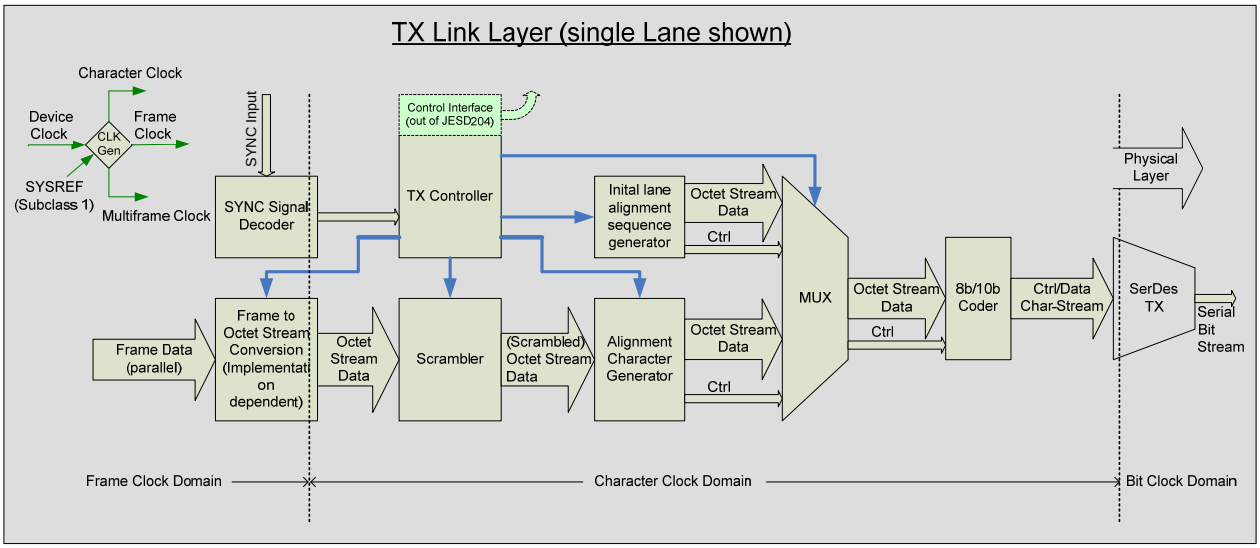


Figure C.4 — TX Link Layer (not supporting early synchronization option in scrambling)

C.1.3 TX Physical Layer

The TX physical layer interfaces the TX link layer to the interconnect. Its operation is illustrated in Figure C.5. Data leaving the parallel to serial converter is passed to an optional pulse shaping/pre-emphasis block. This pre-emphasis network is not needed for the proper implementation of this standard, since the device to device link length is relatively short (typically ≤20 cm) and the interconnection is intentionally kept to a minimum. However, it may be optionally implemented if the application requires it, such as when the standard is implemented across a backplane or a cable. After optional pulse shaping/pre-emphasis, the signal is then sent to the CML driver circuit which connects to the receiver circuit using a suitable transmission line.

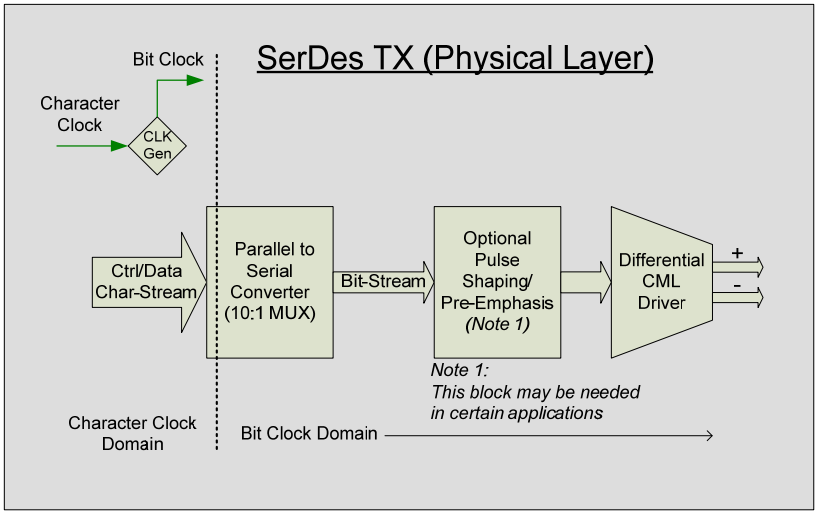


Figure C.5 — TX Physical Layer

Annex C (informative)Device level implementation (cont'd)

C.2 Receiver block

C.2.1 Generic structure

The RX block captures the serial stream(s) from one or more TX blocks and converts the stream(s) into one or more sample streams. The same generic RX structure of Figure C.6 can be used in both a DAC and a logic device. However, the lane-to-lane inter-device synchronization interface is only required in subclass 0 for DACs in the "MCDA" device classes (see clause 9 regarding device classification).

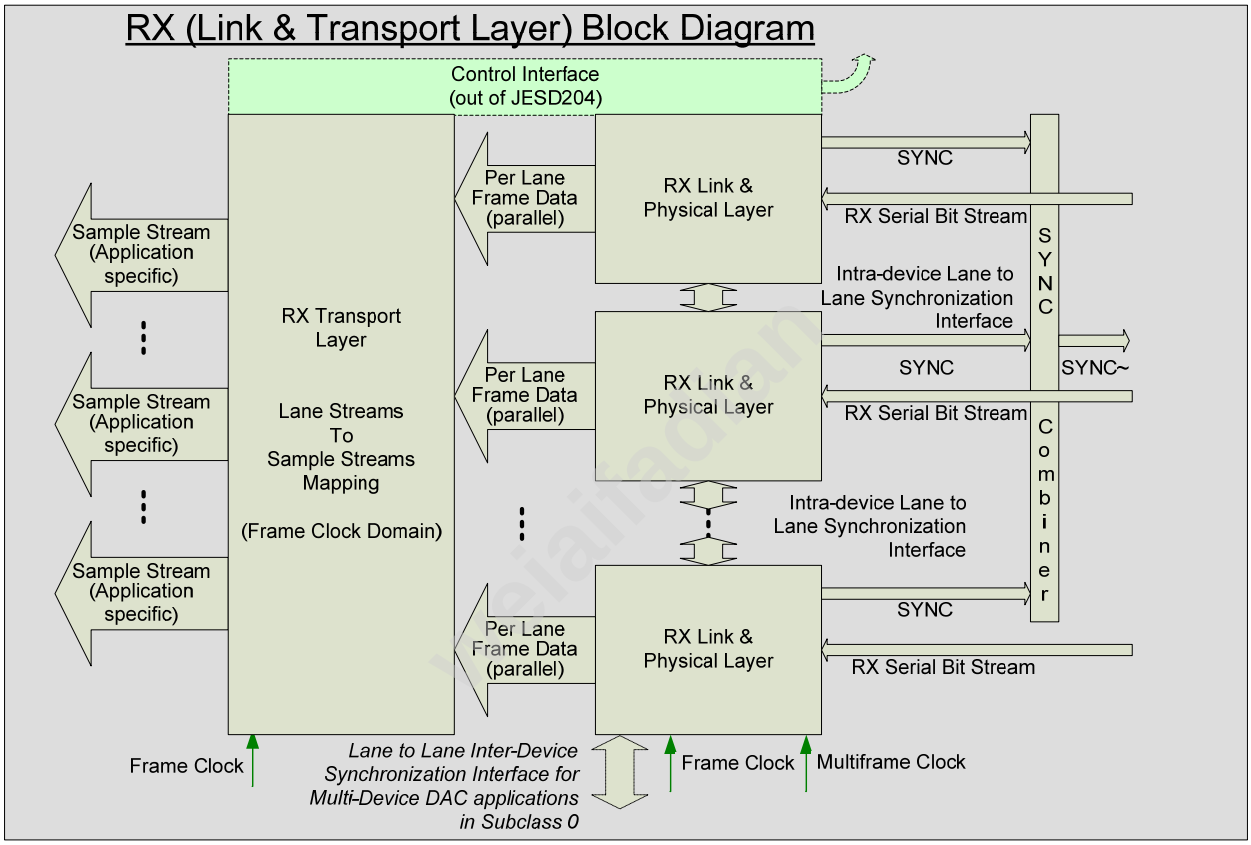


Figure C.6 — Generic structure of an RX block

The functionality of the RX device can be divided into a transport layer and one or more link and physical layers. In the physical layer, the incoming data stream is used to recover the serial clock, which is then used to de-serialize the data. In the link layer, the de-serialized data is then decoded into octets and reorganized into frames. Frames are then processed by the RX transport layer, where they are mapped into one or more sample streams at the frame clock rate. Error reports and synchronization requests of all receivers in the RX block are encoded on a shared SYNC~ signal.

C.2.2 RX Link Layer

Figure C.7 outlines the RX link layer for the standard. Parallel symbols from the physical layer are decoded using an 8B/10B decoder as described in Reference 1. If decoding errors are detected, they are reported to the RX controller where a SYNC~ signal is generated to flag the transmitter when an error occurs or re-initialization is needed.

Annex C (informative) Device level implementation (cont'd)

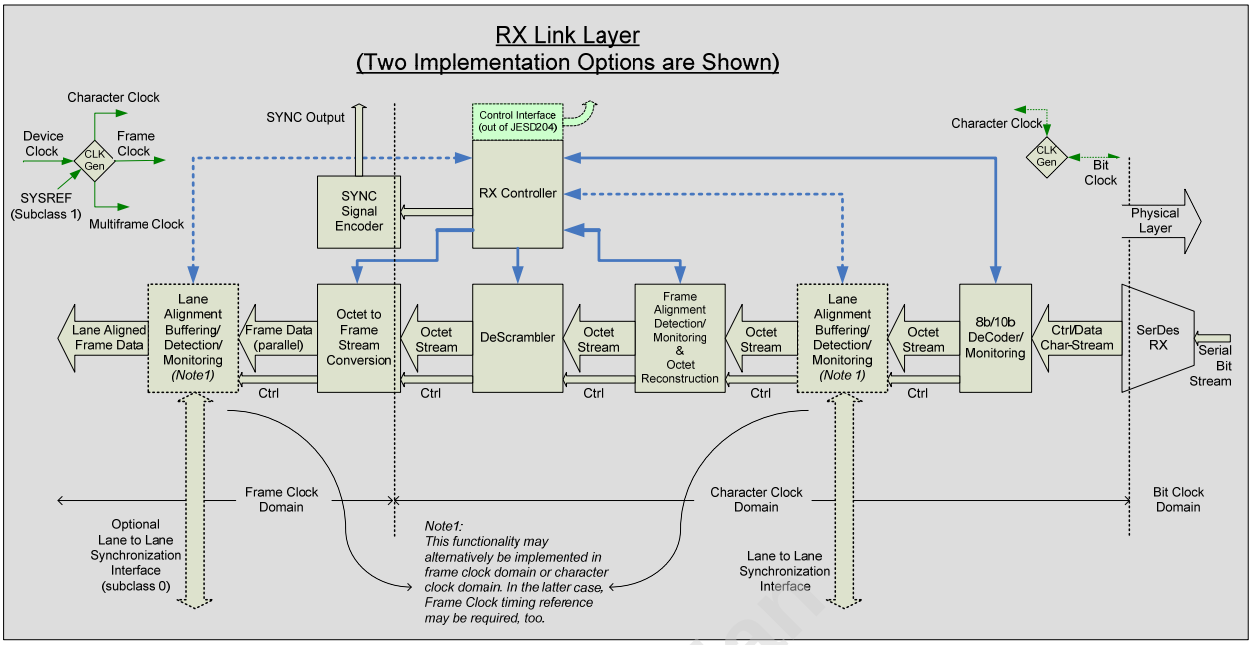


Figure C.7 — RX Link Layer

After 8B/10B decoding, any substitution characters are restored in the octet stream. If data was scrambled at the transmitter, it is also passed through the receiver's optional descrambler. After de-scrambling, data is framed into its original sample format. In applications with multiple lanes or multiple devices, a FIFO and appropriate control mechanism may be required in the final assembly stages to synchronize across multiple lanes.

C.2.3 RX Physical Layer

The RX physical layer interfaces the interconnect to the RX link layer. The operation of the RX physical layer is illustrated in Figure C.8. The incoming signal is passed to an optional equalizer. This equalizer is not required to support this standard, since the link length is relatively short (typically ≤20 cm) and the interconnection is intentionally kept to a minimum. However, it may be optionally implemented if the application requires it to transfer samples across a backplane or a cable. Beyond this, the data stream enters clock recovery and is de-serialized, after which it is passed to the device's link layer for further processing.

Annex C (informative) Device level implementation (cont'd)

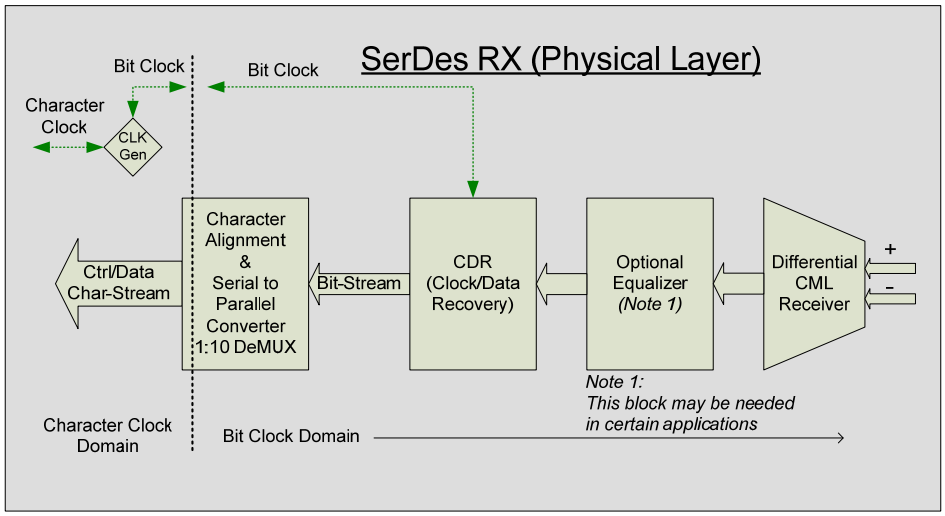


Figure C.8 — RX Physical Layer

Annex D (informative)Parallel scrambler and descrambler implementations

The scrambler and descrambler are defined in their serial implementation form. However, in actual hardware it may be advantageous to use an equivalent parallel implementation. Parallel implementations can be synthesized from the parallel update equations in the scrambler definitions**,** Figure 29and Figure 30. In the 8-bit implementations, the synthesis makes use of the fact that scrambled bit S_i is followed exactly one parallel clock cycle later by scrambled bit S_{i+8}. In this way the parallel update equations of Figure 29 are transformed into the 8-bits scrambler of Figure D.1 and the 8-bits descrambler of

Figure D.2. Similarly, the parallel update equations of the alternative scrambler of Figure 30 are transformed into the alternative 8-bits scrambler of Figure D.3. In the 16-bit implementations, the synthesis is based on the fact that scrambled bit S_i is followed exactly one parallel clock cycle later by scrambled bit S_{i+16}. This way the 16-bit scrambler of Figure D.4 and the 16-bits descrambler of Figure D.5 are derived. Note that the 16-bit implementations have the disadvantage of double propagation time for the two least significant bits. In all these figures the logic symbols are according to IEEE Std. 91a-1991 (Ref. 7).

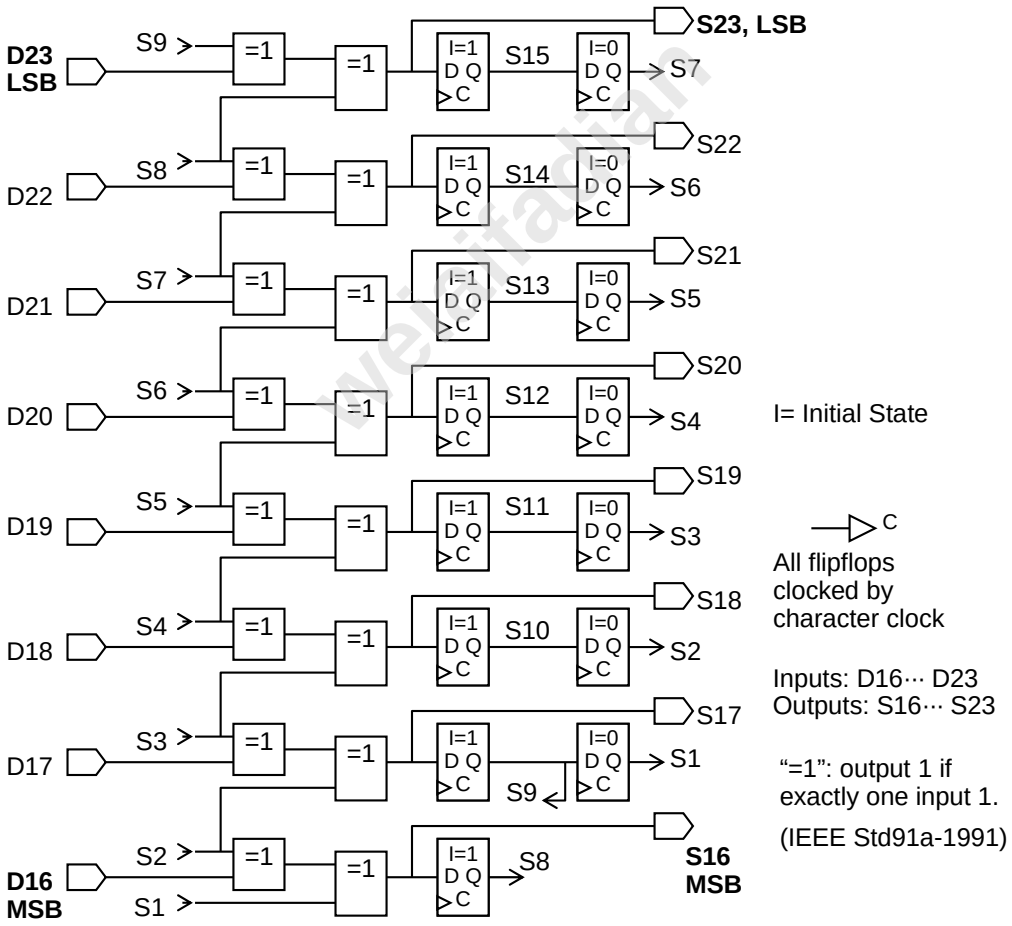


Figure D.1 — 8-bit parallel implementation of self-synchronous scrambler based on $1+x^{14}+x^{15}$

Annex D (informative)Parallel scrambler and descrambler implementations (cont'd)

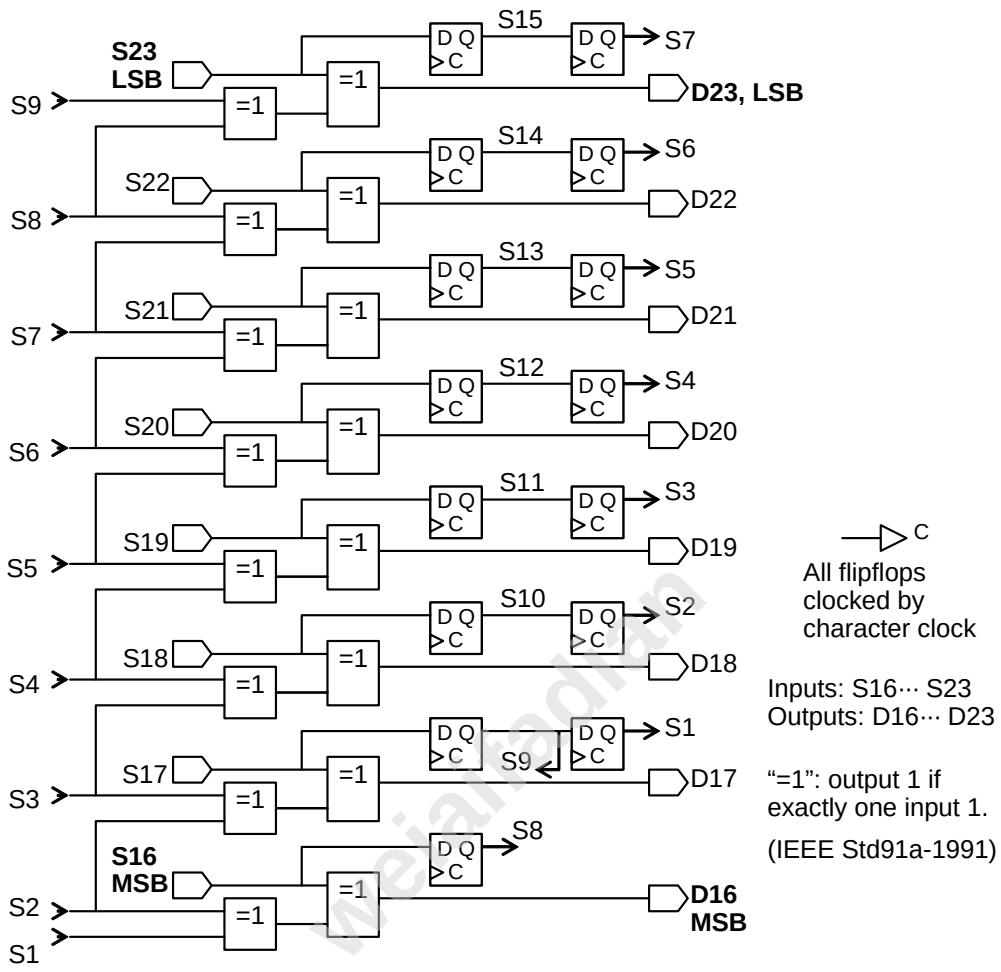


Figure D.2 — 8-bit parallel implementation of self-synchronous descrambler based on $1+x^{14}+x^{15}$

Annex D (informative)Parallel scrambler and descrambler implementations (cont'd)

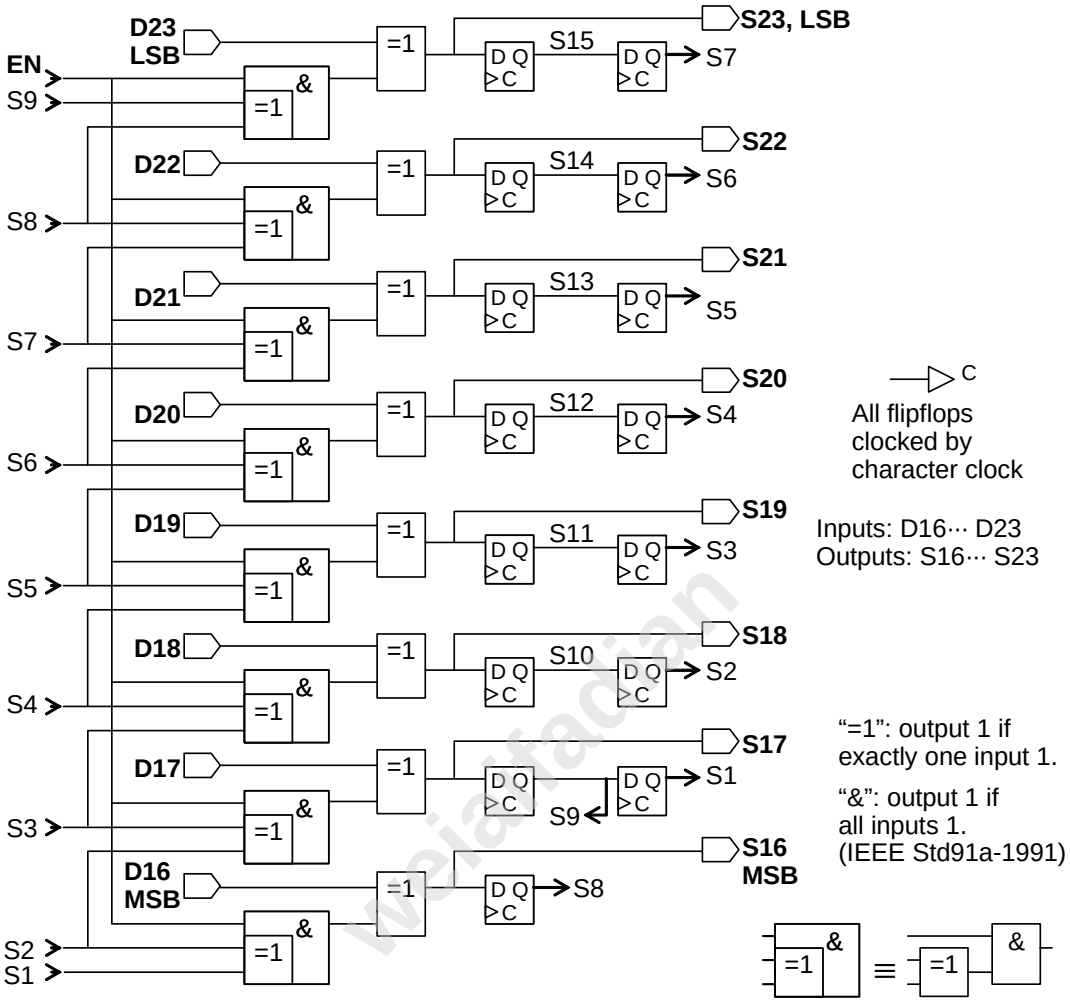


Figure D.3 — 8-bit parallel implementation of alternative early-synchronization scrambler

Annex D (informative)Parallel scrambler and descrambler implementations (cont'd)

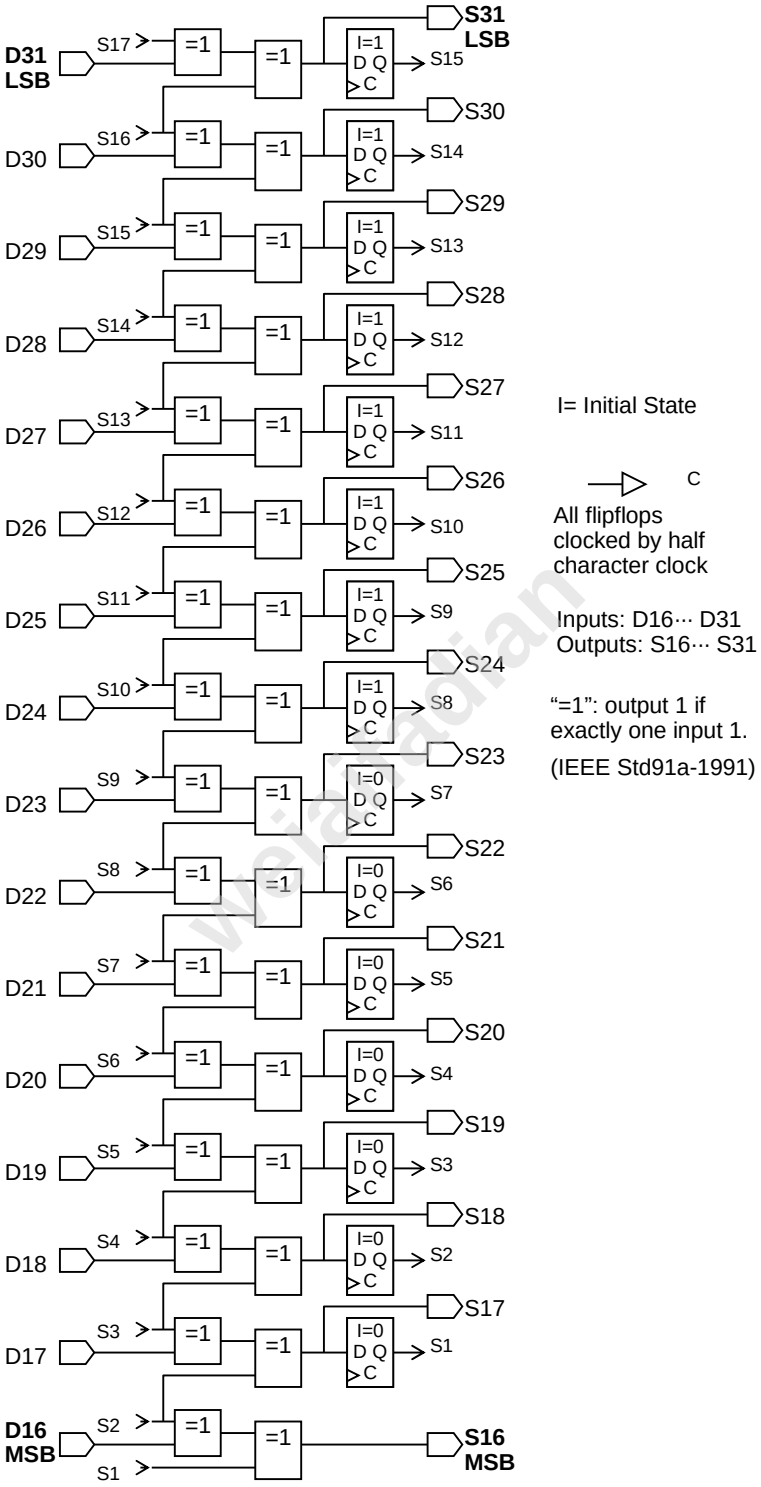


Figure D.4 — 16-bit parallel implementation of self-synchronous scrambler based on $1+x^{14}+x^{15}$

Annex D (informative)Parallel scrambler and descrambler implementations (cont'd)

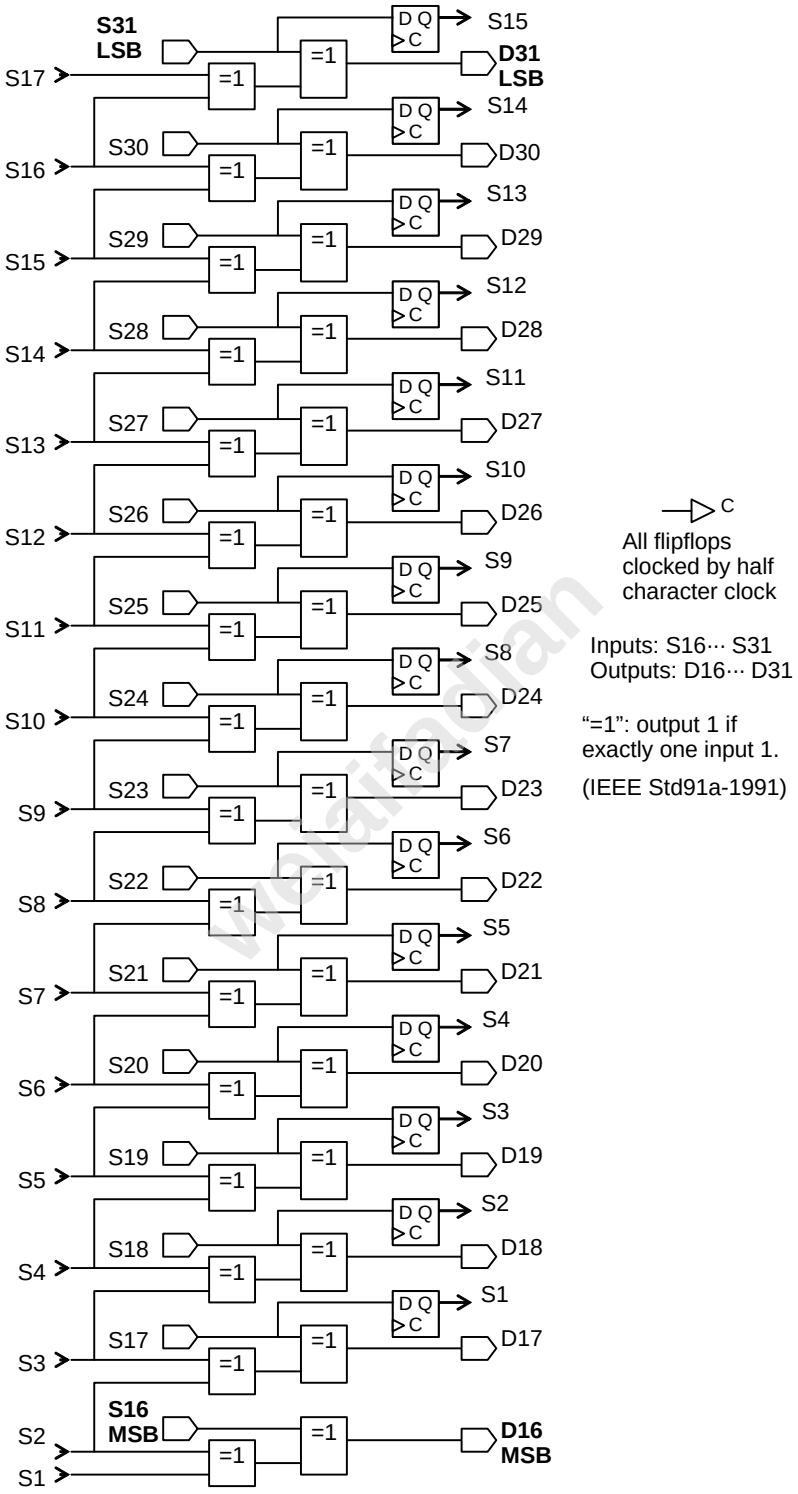


Figure D.5 — 16-bit parallel implementation of self-synchronous descrambler based on $1+x^{14}+x^{15}$

Annex E (informative) Instructions for Determining Linear Fitted Insertion Loss

The linear fitted insertion loss IL_{fit} as a function of frequency f is defined by the equation below:

$$IL_{fit}(f) = a_0 + a_2 f \ (dB)$$

Where f is the frequency in GHz.

Given the channel insertion loss measurement at N uniformly-spaced frequencies f_n spanning f_{min} to f_{max} with a maximum frequency spacing of 10MHz. The coefficients of the fitted insertion loss are computed as follows:

Define the frequency matrix F as shown fellow:

$$\begin{bmatrix} 1 & f_1 \\ \vdots & \vdots \\ 1 & f_N \end{bmatrix}$$

The polynomical coefficients a_0 and a_2 are determined using the equation below:

$$\begin{bmatrix} a_0 \\ a_2 \end{bmatrix} = \langle F^T F \rangle^{-1} F^T IL$$

Where T denoteds the matrix transpose operatior and IL is a column vector of the measured insertion loss values, IL_n at each frequency f_n.

Allowable values for a_0 and a_2 are as follows:

$$0 \le a_0 \le 1$$

$$0 \le a_2 \le 0.8$$

This polynomial fit process is expected to yield values for the coefficients a_0 and a_2 that are greater than zero and less than the maximum coefficients. If any of the coefficients in the equation exceed the maximum value in the specification they are forced to this maximum value. If any coefficients are negative they are forced to zero.

Annex F (informative) Example of device clock and SYSREF generation

In practical systems there may be several JESD204B links connected to the same logic device. All these links could operate at different device clock, frame and multiframe rates. The clock generation and distribution circuit has the task of supplying the system with device clocks and SYSREF signals in such a way that the device clock edges on which the different devices capture SYSREF have predictable timing relations relative to each other. Figure F.1 illustrates several possible implementations for generating a gapped periodic SYSREF. The mechanism for outputting or marking specific SYSREF pulses from the periodic SYSREF source for use in the system needs a certain amount of timing control, which can be implemented at different places in the system and using various auxiliary signals. The figure shows several options for realizing this timing control, which need not all simultaneous implementation. At the moment of writing this standard it is possible to build an implementation according to figure F.1 using commercially available components. Presently much of the timing control for enabling the SYSREF output in the clock generator circuit or enabling SYSREF sampling in the JESD204B devices must be realized in the logic device. It is expected that future clock generator devices will be optimized for JESD204B and reduce the need for most, if not all of the timing control in the logic device.

In a typical application, the timing of the whole system is based on a master clock, which could be for instance a crystal oscillator operating in the 10 MHz range. It is often not convenient to derive the device clocks directly from such a low frequency signal. Instead, a higher frequency source clock may be used from which all device clocks are derived by frequency division. The source clock is phase locked to the master clock and would typically operate somewhere in the range of several hundred MHz to a few GHz. In addition to the device clocks, a timing reference for SYSREF is also derived from the source clock. This reference is designated as the SYSREF clock. In principle the SYSREF clock could be permanently distributed directly to all devices in the system, but this is not always desirable from the perspective of electromagnetic interference. Therefore JESD204B offers the possibility to generate a SYSREF pulse only on request of one of the JESD204B devices. Such a SYSREF request can be used to enable the output of the SYSREF clock from the clock generator circuit.

When the first device clock rising edge after each rising edge of the SYSREF clock can be compared directly to a rising edge of each LMFC in the system, no realignment will occur at repeated applications of a SYSREF pulse. When all LMFCs are at the same frequency, the SYSREF clock rate must be equal to the LMFC rate or a subharmonic of it. When there are multiple LMFC frequencies, the SYSREF clock must be a common subharmonic of all LMFCs. This requirement determines the highest possible SYSREF clock frequency that can be used in the system. The lowest possible frequency is determined by by the maximum supported division ratio in the clock distribution circuit.

Phase alignment between all clocks is ensured by a common reset signal to all frequency dividers in the clock distribution circuit. This reset signal will typically be generated at system start-up only. The reset signal could for instance be derived from the lock detect output of the source clock or inside the logic device. The reset signal is latched inside the clock distribution circuit to the source clock, to guarantee that all dividers are reset and released simultaneously. When the reset signal is de-activated, all dividers start counting from their zero state. This way it is ensured that the transitions from the highest to the zero state are aligned in all dividers. Hence at the moment that the SYSREF clock, which is a subharmonic of all other clocks, has a high-to-low transition, all other clocks would also switch to the low state. However, the preferred implementation of JESD204B is to sample the rising edge of SYSREF on the rising edge of the device clocks. Therefore the inverting outputs of the dividers are used, so that the rising edges of all device clocks will be aligned to the rising edge of the SYSREF clock.

Annex F (informative) Example of device clock and SYSREF generation (cont'd)

Many commercial clock distribution devices contain programmable delays, by means of which the phases of the device clocks and SYSREF can be fine adjusted to match the setup and hold time requirements of the converter devices and logic device. It is also possible to split the enabled SYSREF clock output into multiple SYSREF signals, each with its own delay.

The enabling function must be implemented carefully, so that it does not interfere with the accurate timing of SYSREF. If the enabling occurs when the SYSREF clock is high, the first rising edge of the SYSREF ouput(s) will occur at the moment of enabling and will not be aligned to the rising edge of the SYSREF clock. Therefore it is necessary to either retime the SYSREF generation request to the SYSREF clock domain or otherwise provide a separate indication to the JESD204B devices indicating when they are allowed to sample the SYSREF signal.

For implementations using a SYSREF retiming function, the retiming is most easily done inside the clock distribution device, because it has all required clock signals available. If the clock distribution device does not include this functionality, the retiming can be carried out in the logic device. It requires that the logic device knows the starting phase of the SYSREF clock. The logic device will know that if it has reset the clock distribution device itself, or has received the clock reset signal from the device that generated it. The logic device is also able to derive the SYSREF clock from the master clock. Hence the logic device is able to generate an internal copy of the SYSREF clock. This copy will not necessarily be accurate enough to be used as direct timing reference for SYSREF, but it can be made accurate enough to meet the timing requirements of the SYSREF request.

An alternative to using the SYSREF retiming function is to postpone the sampling of the SYSREF signal in the JESD204B devices until a later period of the SYSREF clock (i.e. avoiding the initial SYSREF rising edge triggered by the enabling of the clock generator output). If the SYSREF generation request is kept active for multiple SYSREF clock periods, the second and all subsequent SYSREF pulses will certainly take their timing from the SYSREF clock. The JESD204B devices could be told when they are allowed to readjust themselves to a new SYSREF pulse for instance via an "Enable SYSREF sampling" signal or possibly a command through the control interface. Such a signal or control command could be generated either in the clocking device or in the logic device, either with or without knowledge of the SYSREF clock timing (i.e. SYSREF Clock Emulator and Re-timing circuits are not required). Without knowledge of the SYSREF clock timing, care should be taken to ensure that at least one SYSREF pulse has been generated before enabling a device to re-sample SYSREF and at least one will SYSREF pulse will be generated after the enabling function is activated.

The timings of the circuit in figure F.1 are illustrated in Figure F.2. For simplicity, no optional delays have been assumed. All dividers start counting at the first rising edge of the source clock after release of the clock reset signal. All divider outputs change state at the rising edge of the source clock. In this example, the device clock of the first converter is divided by one, thus the device clock is an inverted version of the source clock. All other clocks have their low-to-high transitions at the rising edge of the source clock. The SYSREF clock transits from low to high when the internal divider state changes from its highest count to zero count. At this moment all other dividers have also gone through a whole number of counting cycles and their inverted outputs transit also from low to high. In the diagram the SYSREF generation request becomes active when the SYSREF clock is high. Consequently the first rising edge of SYSREF is not aligned to the SYSREF clock.

Annex F (informative) Example of device clock and SYSREF generation (cont'd)

When the SYSREF generation request stays active for multiple periods of the SYSREF clock, the second and later SYSREF rising edges will be properly aligned. SYSREF sampling may be enabled in the JESD204B devices after the first SYSREF pulse has ended. If the implementation can assure that the SYSREF clock output can only be enabled when the SYSREF clock is low, the JESD204B devices may be allowed to sample also the first SYSREF pulse.

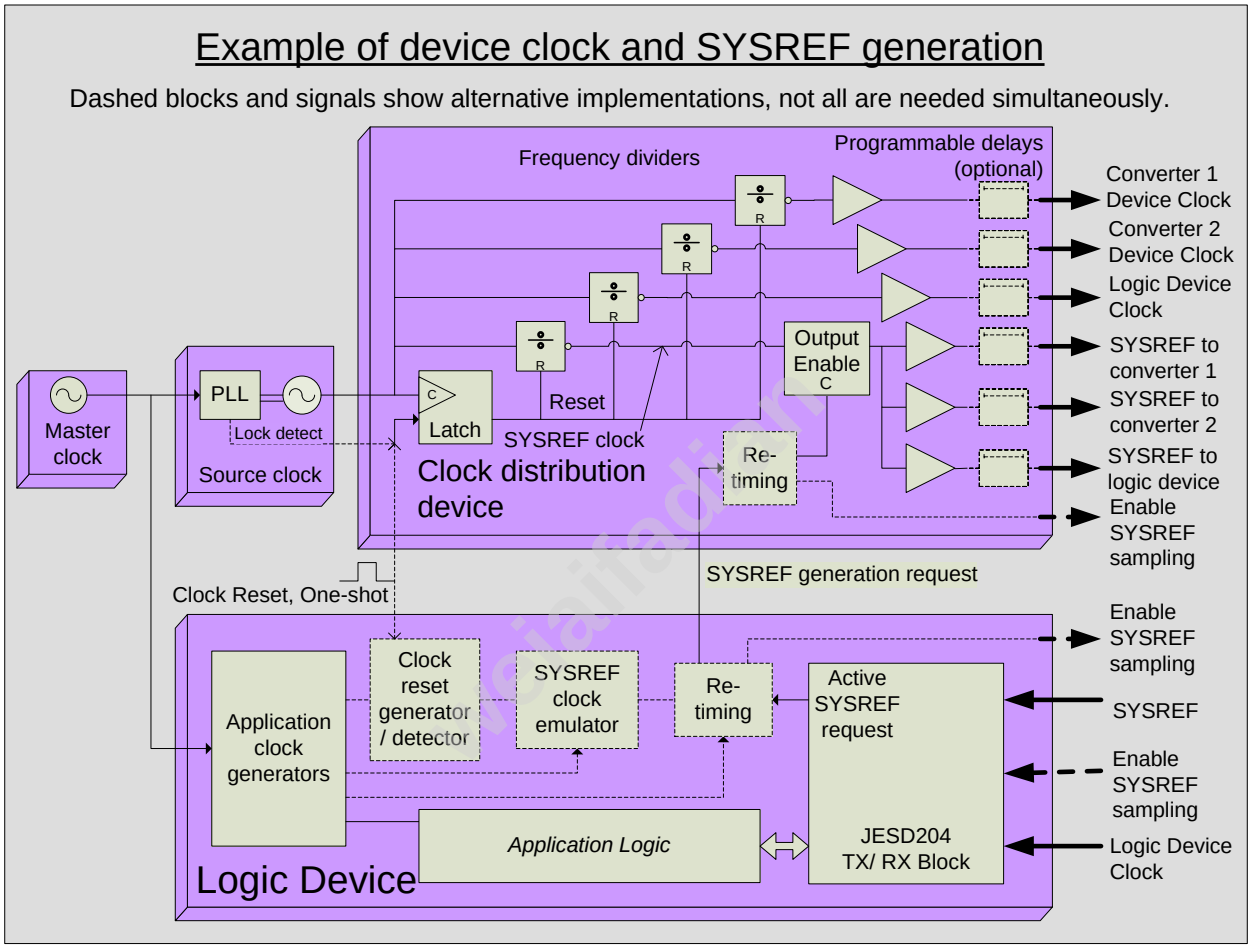


Figure F.1 — Implementation example and options of device clock and SYSREF generation.

Annex F (informative) Example of device clock and SYSREF generation (cont'd)

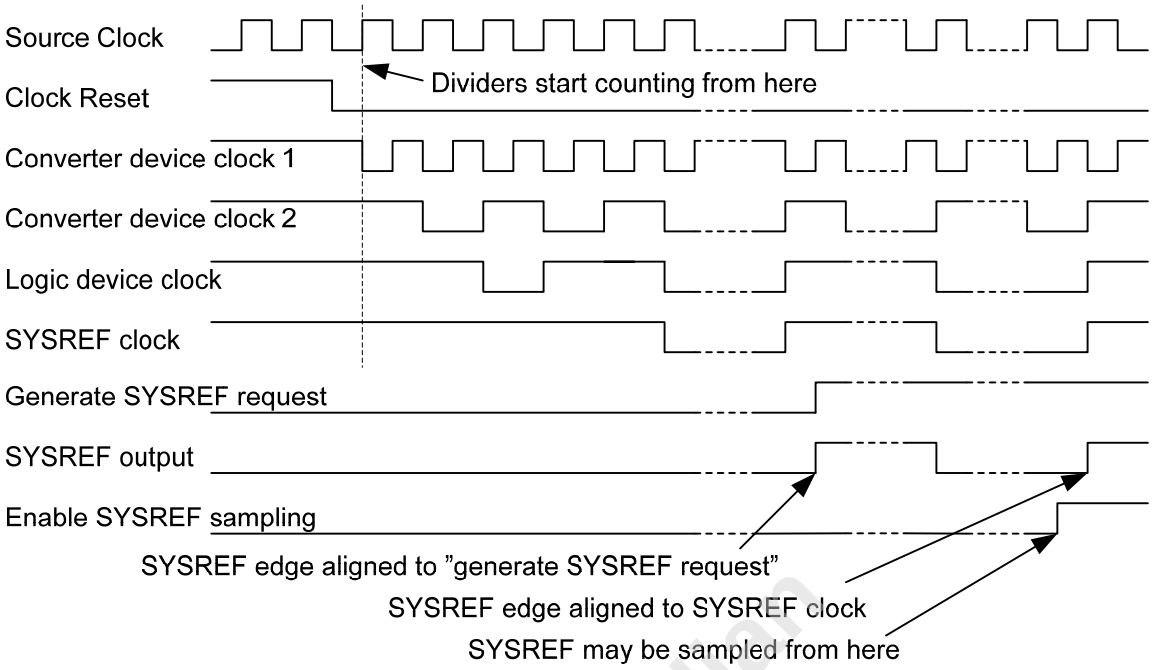


Figure F.2 — Timing diagram.

Annex G (informative)Clock Terminology

Table G.1 — ClockRatios With Respect to MultiFrameClock

Clock name	ratio (w.r.t. multiframeclock)	related JESD204B parameter/range
SYSREF Clock*, **	1 / *R*	*R* = integer
(Local)MultiFrameClock (LMFC)	(1)	
frame clock	x K	ceil (17/F) ≤ K ≤ min(32, floor(1024/F))
character clock	x K x F	F = 1… 256
bitclock	x K x F x 10	8B/10B encoding
sample clock	x K x S	S = 1 … 32
conversion clock*	x K x S x *C*	C = interpolation- or decimation-factor
device clock	x D	D = integer in case of subclass 1, 2
*- Optional Signals ** - SYSREF Clock is defined in Annex F		

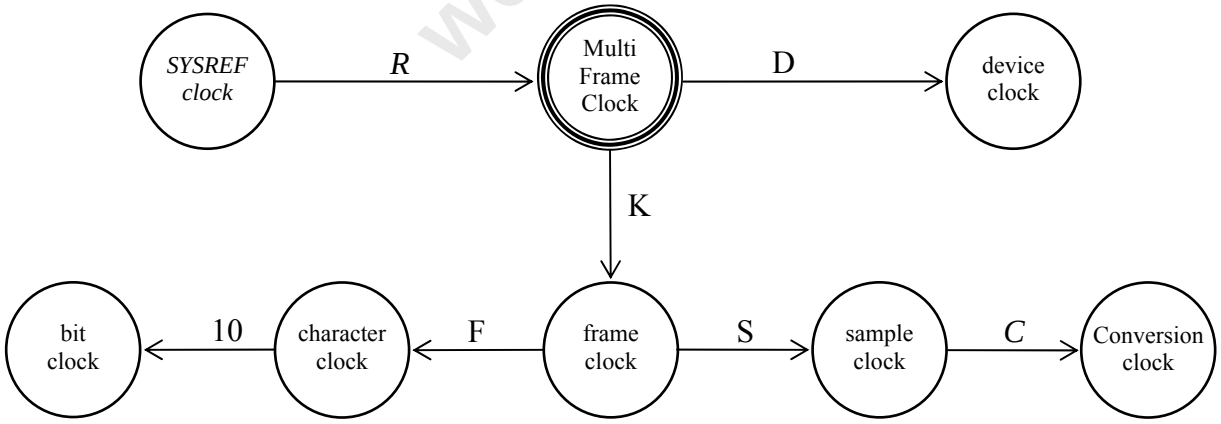


Figure G.1 — Relation diagram for JESD204B clocks

Standard Improvement Form **JEDEC**

The purpose of this form is to provide the Technical Committees of JEDEC with input from the industry regarding usage of the subject standard. Individuals or companies are invited to submit comments to JEDEC. All comments will be collected and dispersed to the appropriate committee(s).

If you can provide input, please complete this form and return to:

JEDEC
Attn: Publications Department
3103 North 10th Street
Suite 240 South
Arlington, VA 22201-2107

Fax: 703.907.7583

1. I recommend changes to the following:

☐ Requirement, clause number _____

☐ Test method number _____ Clause number _____

The referenced clause number has proven to be:

☐ Unclear ☐ Too Rigid ☐ In Error

☐ Other _____

2. Recommendations for correction:

_____

_____

_____

_____

3. Other suggestions for document improvement:

_____

_____

_____

_____

Submitted by

Name: _____ Phone: _____

Company: _____ E-mail: _____

Address: _____

City/State/Zip: _____ Date: _____