INTERNATIONAL STANDARD

ISO/IEC 11518-10

First edition
2001-03

Information technology – High-performance parallel interface

Part 10: 6 400 Mbit/s Physical Layer (HIPPI-6400-PH)

Reference number
ISO/IEC 11518-10:2001(E)

INTERNATIONAL STANDARD

ISO/IEC 11518-10

First edition
2001-03

Information technology – High-performance parallel interface

Part 10: 6 400 Mbit/s Physical Layer (HIPPI-6400-PH)

PRICE CODE V

For price, see current catalogue

CONTENTS

INFORMATION TECHNOLOGY – HIGH-PERFORMANCE PARALLEL INTERFACE –

Part 10: 6 400 Mbit/s Physical Layer (HIPPI-6400-PH)

FOREWORD

1) ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

2) In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

3) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 11518-10 was prepared by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 3.

ISO/IEC 11518 consists of the following parts, under the general title *Information technology – High-Performance Parallel Interface:*

– *Part 1: Mechanical, electrical, and signalling protocol specification (HIPPI-PH)*
– *Part 2: Framing Protocol (HIPPI-FP)*
– *Part 3: Encapsulation of ISO/IEC 8802-2 (IEEE Std 802.2) Logical Link Control Protocol Data Units (HIPPI-LE)*
– *Part 4: Mapping of HIPPI to IPI device generic command sets (HIPPI-IPI)* [1)]
– *Part 5: Memory Interface (HIPPI-MI)* [1)]
– *Part 6: Physical Switch Control (HIPPI-SC)*
– *Part 8: Mapping to Asynchronous Transfer Mode (HIPPI-ATM)*
– *Part 9: Serial Specification (HIPPI-Serial)*
– *Part 10: 6 400 Mbit/s Physical Layer (HIPPI-6400-PH)*
– *Part 11: 6 400 Mbit/s Physical Switch Control (HIPPI-6400-SC)* [1)]
– *Part 12: 6 400 Mbit/s Optical Specification (HIPPI-6400-OPT)* [1)]

Annex A is for information only.

[1)] Under consideration.

INTRODUCTION

Characteristics of a HIPPI-6400-PH physical-layer interface include:

– user data transfer bandwidth of 6 400 Mbit/s (800 MByte/s);

– a full-duplex link capable of independent full-bandwidth transfers in both directions simultaneously;

– four virtual circuits providing a limited multiplexing capability;

– a fixed-size transfer unit, i.e., a 32-byte micropacket, for hardware efficiency;

– a small transfer unit resulting in low latency for short Messages, and a component for large transfers;

– credit-based flow control that prevents buffer overflow;

– end-to-end, as well as link-to-link, checksums;

– automatic retransmission to correct flawed data providing guaranteed, in-order, reliable, data delivery;

– an a.c. coupled parallel electrical interface for driving parallel copper cable over limited distances;

– a parallel electrical interface for driving a local optical interface for longer distances.

INFORMATION TECHNOLOGY – HIGH-PERFORMANCE PARALLEL INTERFACE –

Part 10: 6 400 Mbit/s Physical Layer (HIPPI-6400-PH)

1 Scope

This part of ISO/IEC 11518 specifies a physical-level, point-to-point, full-duplex, link interface for reliable, flow-controlled transmission of user data at 6 400 Mbit/s per direction, across distances of up to 1 km. A parallel copper cable interface for distances of up to 40 m is specified. Connections to a separate longer-distance optical interface are provided. Small fixed-size micropackets provide an efficient, low-latency structure for small transfers, and a component for large transfers.

Specifications are included for:

– automatic retransmission to correct flawed data;
– the format of a small data transfer unit called a micropacket;
– a message structure that includes routing information for network applications;
– end-to-end, as well as link-to-link, checksums;
– the timing requirements of the parallel signals;
– a parallel interface using copper coaxial cable;
– connections to a separate local optical interface;
– a link-level protocol tuned for a maximum distance of 1 km.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 11518. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC 11518 are encouraged to investigate the possibility of applying the most recent edition of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC TR 8802 (all parts), *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements*

ISO/IEC TR 8802-1:1997, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 1: Overview of Local Area Network Standards*

ISO/IEC 8802-2:1998, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 2: Logical link control*

ISO/IEC 15802-3:1998, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Common specifications – Part 3: Media Access Control (MAC) Bridges*

3 Definitions and conventions

3.1 Definitions

For the purposes of this part of ISO/IEC 11518, the following definitions apply.

3.1.1
acknowledge (ACK)
confirmation that the Destination has received the micropacket without errors

3.1.2
administrator
station management entity providing external management control

3.1.3
credit
credit corresponds to one micropacket's worth of buffer space available in the Destination's VC buffer

3.1.4
destination
receiving end of a physical link

3.1.5
element
component of a HIPPI-6400 system that is able to receive, process, and send Admin micropackets

3.1.6
final destination
end device that receives, and operates on, the data payload portion of the micropackets

This is typically a host computer system, but may also be a non-transparent translator, bridge, or router.

3.1.7
link
full-duplex connection between HIPPI-6400-PH devices

3.1.8
log
act of making a record of an event for later use

3.1.9
message
ordered sequence of one or more micropackets that have the same VC, Originating Source, and Final Destination

Messages are the basic transfer unit between an Originating Source and a Final Destination. The first micropacket of a Message is a Header micropacket. The last micropacket, which may also be the first micropacket, has the TAIL bit set (see 4.5).

3.1.10
micropacket
basic transfer unit, between a Source and Destination, consisting of 32 data bytes and 64 bits of control information (see 4.4)

3.1.11
next-layer
protocols above the service interface

These could be implemented in hardware or software, or they could be distributed between the two.

3.1.12
optional
characteristics that are not required by HIPPI-6400-PH

However, if any optional characteristic is implemented, it shall be implemented as defined in HIPPI-6400-PH.

3.1.13
originating source
end device that generates the data payload portion of the micropackets

This is typically a host computer system, but may also be a non-transparent translator, bridge, or router.

3.1.14
source
sending end of a physical link

3.1.15
station management (SMT)
supervisory entity that monitors and controls the HIPPI-6400-PH entity

3.1.16
syndrome
value (should be zero if no error) obtained by exclusive ORing the calculated CRC value with the CRC value received with the micropacket

3.1.17
Universal LAN MAC Address (ULA)
logical address stored in a Source or Destination field that uniquely identifies an Originating Source or Final Destination

The ULA conforms to the 48-bit MAC address specified by ISO/IEC 8802-1.

3.1.18
Virtual Channel (VC)
one of four logical paths within each direction of a single link

3.2 Editorial conventions

In this part of ISO/IEC 11518, certain terms that are proper names of signals or similar terms are printed in upper case to avoid possible confusion with other uses of the same words (e.g., FRAME). Any lower case uses of these words have the normal technical English meaning.

A number of conditions, sequence parameters, events, states, or similar terms are printed with the first letter of each word in upper case and the rest lower case (e.g., Block, Source). Any lower case uses of these words have the normal technical English meaning.

The word *shall,* when used in this part of ISO/IEC 11518, states a mandatory rule or requirement. The word *should,* when used in this part of ISO/IEC 11518, states a recommendation.

3.2.1 Binary notation

Binary notation is used to represent relatively short fields. For example a two-bit field containing the binary value of 10 is shown in binary format as b'10'.

3.2.2 Hexadecimal notation

Hexadecimal notation is used to represent some fields. For example a two-byte field containing a binary value of b'1100010000000011' is shown in hexadecimal format as x'C403'.

3.3 Acronyms and abbreviations

ACK	acknowledge indication
CR	credit amount parameter
CRC	cyclic redundancy check
DSAP	Destination Service Access Protocol
ECRC	end-to-end CRC
HIPPI	High-Performance Parallel Interface
K	kilo (2^{10} or 1024)
LCRC	link CRC
LLC	Logical Link Control
lsb	least significant bit
M	mega (10^6)
MAC	Media Access Control
ms	milliseconds
msb	most significant bit
ns	nanoseconds
ps	picoseconds
RSEQ	receive sequence number
SMT	station management
SNAP	SubNetwork Access Protocol
SSAP	Source Service Access Protocol
TSEQ	transmit sequence number
ULA	Universal LAN Address
VC	Virtual Channel
VCR	Virtual Channel Credit selector
µs	microseconds
Ω	ohms

4 System overview

4.1 Overview

This clause provides an overview of the structure, concepts, and mechanisms in HIPPI-6400-PH. Figure 1 shows an example HIPPI-6400 system.

4.2 Links

HIPPI-6400-PH defines a point-to-point physical link for transferring micropackets. The physical links, as shown in figure 2 between a local end and a remote end, are bi-directional. The logical links are simplex, i.e., the data inbound and outbound are completely separate. Some control information, e.g., credit, flows in the reverse direction, and it is included in the micropackets flowing in the reverse direction. This is why the physical links must be bi-directional with information flowing in both directions simultaneously.

A link is composed of two Sources that transmit information, and two Destinations that receive information. Each end of a link has a Source and a Destination.

The data path is 16 bits wide for the copper implementation, and is 8 bits wide for a fibre implementation. The control path is one-fourth the width of the data path, e.g., the control path for the copper implementation is 4 bits wide. CLOCK, CLOCK_2, and FRAME are individual signals carried on separate conductors. The CLOCK_2 signal is only used in 16-bit systems.

4.3 Virtual Channels

Four Virtual Channels, VC0, VC1, VC2, and VC3 are available in each direction on each link. The VCs are assigned to specific Message sizes and transfer methods.

All of the micropackets of a Message are transmitted on a single VC, i.e., the VC number does not change as the micropackets travel from the Originating Source to the Final Destination over one or more links. Messages to a Final Destination are delivered in order on a single VC. Multiple messages may be out of order if sent over different VCs-even if the VCs are in the same physical link. The VCs provide a multiplexing mechanism that can be used to prevent a large Message from Blocking a small Message until the large Message has completed.

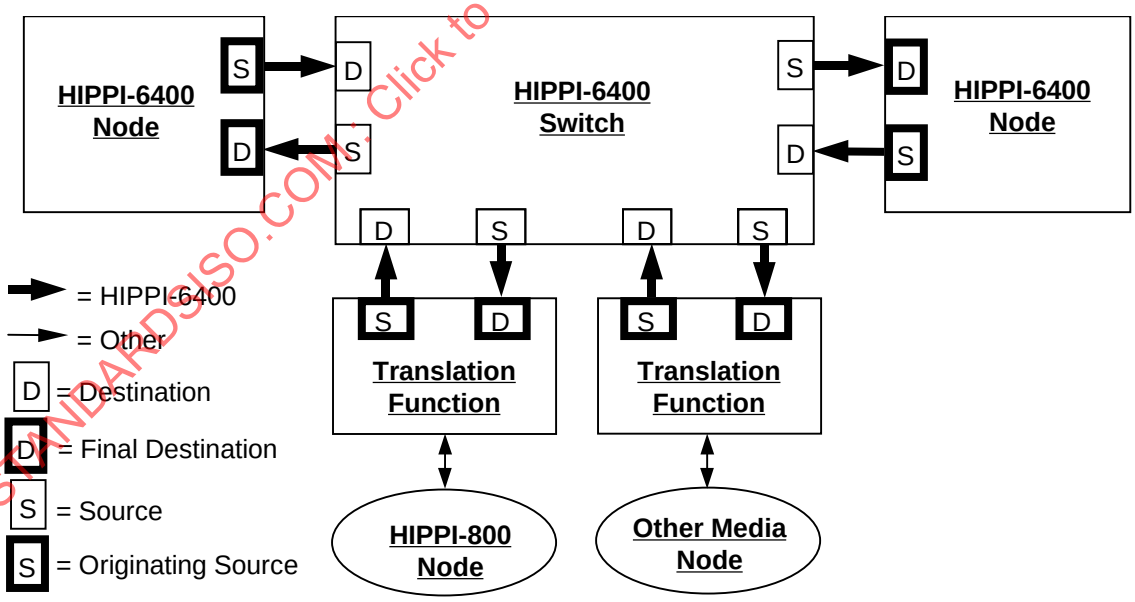
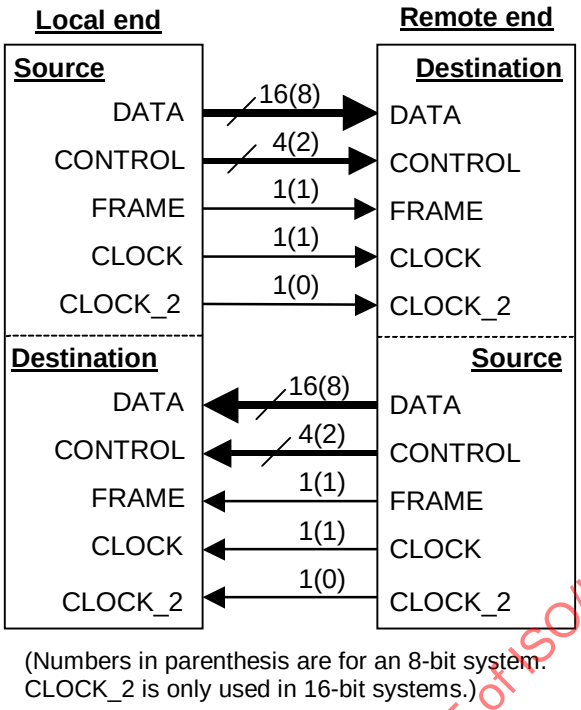


Figure 1 – System overview



(Numbers in parenthesis are for an 8-bit system.
CLOCK_2 is only used in 16-bit systems.)

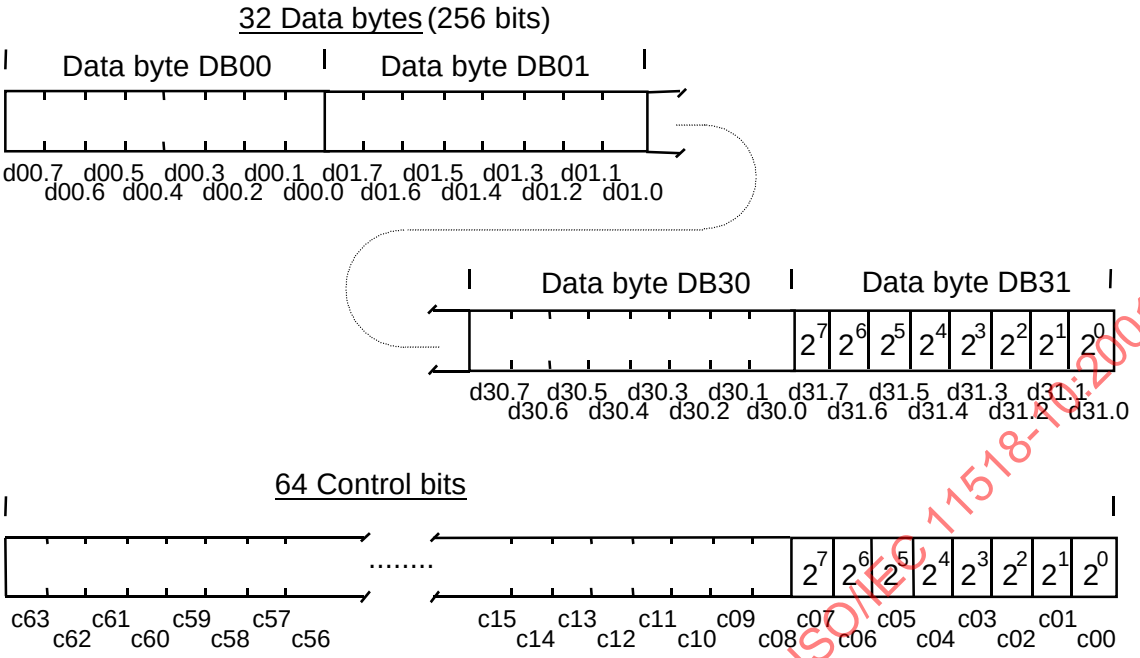
Figure 2 – HIPPI-6400-PH link showing signal lines

4.4 Micropacket

Micropackets are the basic transfer unit from Source to Destination on a link. As shown in figure 3, a micropacket is composed of 32 data bytes and 64 bits of control information. At 6 400 Mbit/s a micropacket is transmitted every 40 ns, with Null micropackets transmitted when other micropackets are not available. Credit and retransmit operations are performed on a micropacket basis.

The 64 bits of control information in each micropacket includes parameters for:

– selecting a VC;
– detecting missing micropackets;
– denoting the types of information in the micropacket;
– marking the last micropacket of a Message;
– signaling that the Message was truncated at its originator, or damaged en-route;
– passing credit information from the Destination to the Source;
– Link-level and end-to-end checksums.



Naming conventions:

Data bytes are labelled capital DB and a two-digit number, e.g., DB00.

In a parameter that uses multiple Data bytes, the most-significant byte is the lowest-numbered byte.

Data bits are labelled lower case d, a two-digit byte number, and a one-digit bit number, e.g., d31.7.

Within each Data byte, the most-significant bit is the highest-numbered bit, i.e., d*nn*.7.

Control bits are labelled lower case c and a two-digit number, e.g., c00.

In a parameter that uses multiple Control bits, the most-significant bit is the highest-numbered bit.

Figure 3 – Logical micropacket format and naming conventions

4.5 Message

As shown in figure 4, a Message is an ordered sequence of one or more micropackets that have the same VC, Originating Source, and Final Destination. The first micropacket of a Message, i.e., the Header micropacket, contains information used to route through a HIPPI-6400 fabric. The last micropacket of the Message is marked with the TAIL bit.

1	Header information, Bytes 0 - 7	c63–c00
2	Bytes 8 - 39 of Message data	c63–c00
3	Bytes 40 - 71 of Message data	c63–c00
⋮	⋮	
n	Last bytes of Message data	c63–c00

Micropacket transmission order

Figure 4 – Message format

4.6 FRAME and CLOCK signals

The FRAME signal, carried on a separate signal line, marks a micropacket's beginning. Both edges of either the CLOCK or CLOCK_2 signals, also carried on separate signal lines, are used for strobing the data. The data, control, and FRAME signals from a Source are synchronous with that Source's CLOCK and CLOCK_2 signals. The CLOCK rate is dependent on the width of the data bus, e.g., a 16-bit data bus utilizing 4b/5b encoding requires the CLOCK line to run at 250 MHz and each data and control line may transition every 2 ns.

4.7 Flow control

Link-level credit-based flow control is used between a Source and Destination. As shown in figure 5, the credits are assigned on a VC basis, i.e., VC0's credits are separate from VC1's credits. The Destination end of a link grants credits to match the number of free receive buffers for a particular VC. The Source end of the link consumes credits as it moves micropackets from the VC Buffers to the Output Buffer. Note that flow control is on a link basis, i.e., hop-by-hop.

4.8 Retransmission

Retransmission is performed to correct flawed micropackets (see 8.4). Go-back-N retransmission is used, i.e., if an error is detected, then the flawed micropacket, and all micropackets transmitted after it, are retransmitted. The CRCs in each micropacket are checked at the Destination side of a link; at the Input Buffer in figure 5. Correct micropackets are acknowledged, flawed micropackets are discarded. Note that retransmission is independent of the VC used, and also independent of the credit information, i.e., retransmission occurs between the Output and Input Buffers in figure 5 while VC and credit information pertains only to the VC Buffers. Retransmission is on a link basis, i.e., hop-by-hop.

4.9 Check functions

As shown in table 1, two 16-bit cyclic redundancy checks (CRCs) are used, and they use different polynomials. The end-to-end CRC (ECRC) covers the data bytes of all of the micropackets in a Message, i.e., the Header micropacket and all of the Data micropackets (if any) up to this point in a Message. The ECRC does not cover the control bits. The ECRC is unchanged from the Originating Source to the Final Destination. The ECRC is accumulated over an entire Message, i.e., it is not reinitialized for intermediate Data micropackets (see 6.6.3). Note that in table 1, the second micropacket's ECRC covers the information in the first and second micropacket; the third micropacket's ECRC covers the information in the first, second, and third micropacket, etc.

The link CRC (LCRC) covers all of the data and control bits of a micropacket, with the exception of itself (see 6.6.2). The LCRC is initialized for each micropacket, and must be calculated fresh for each link since other control fields change.

The combination of two 16-bit CRCs provides a stronger check than a single 16-bit CRC for link-level checking of individual micropackets. In addition, the 16-bit ECRC provides checking over a whole Message.

4.10 Local electrical interface (optional)

The optional local on-board electrical interface (see clause 15) provides a connection to a separately specified optical interface (see ISO/IEC 11518-12, HIPPI-6400-OPT)[1)] for longer distances. Note that the TSEQ and RSEQ parameter sizes in this part of ISO/IEC 11518 support full speed operation at distances up to 1 km.

---

1) ISO/IEC 11518-12, *Information technology – High Performance Parallel Interface – 6 400 Mbit/s Optical Interface (HIPPI-6400-OPT)* (under consideration).

4.11 Copper cable physical layer (optional)

The optional HIPPI-6400-PH copper cable variant (see clause 16) uses a cable with 46 conductor pairs, 23 in each direction, and an overall shield. The maximum length is dependent upon the quality of the cable. The signals are a.c. coupled to the cable to accommodate some difference in the ground potential between the equipment.

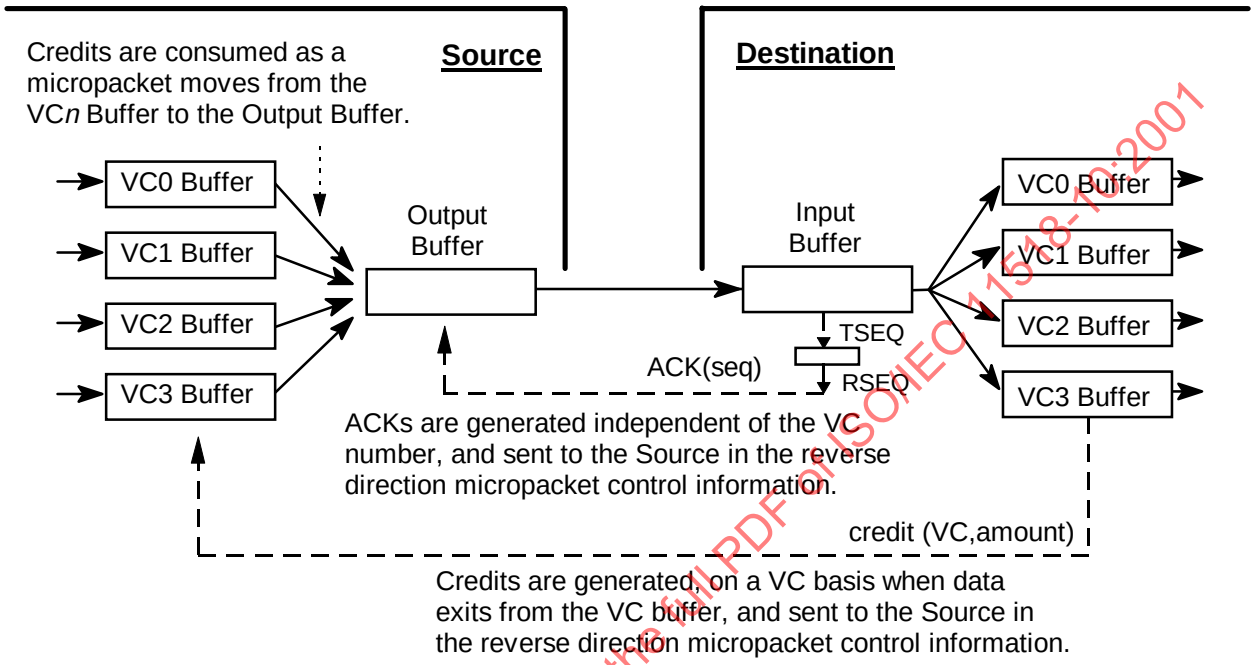


Figure 5 – Reverse direction control information

Table 1 – CRC coverages in a 128-byte Message

Micropacket number	Data Bytes DB00 – DB31 contents	ECRC coverage	LCRC coverage
1	Header, Bytes – 7	Header, Bytes 0 – 7	Header, Bytes 0 – 7, c00 – c47
2	Bytes 8 – 39	Header, Bytes 0 – 39	Bytes 8 – 39, c00 – c47
3	Bytes 40 – 71	Header, Bytes 0 – 71	Bytes 40 – 71, c00 – c47
4	Bytes 7 – 103	Header, Bytes 0 – 103	Bytes 72 – 103, c00 – c47
5	Bytes 10 – 135	Header, Bytes 0 – 135	Bytes 104 – 135, c00 – c47
NOTE For a 128-byte Message, bytes 128-135 would be pad bytes containing x'00'.			

5 Service interface

5.1 Overview

This clause specifies the services provided by HIPPI-6400-PH. The intent is to allow next-layers to operate correctly with this HIPPI-6400-PH. How many of the services described herein are chosen for a given implementation is up to that implementor; however, a set of HIPPI-6400-PH services shall be supplied sufficient to satisfy the next-layer(s) being used. The services as defined herein do not imply any particular implementation, or any interface.

Figure 6 shows the relationship of the HIPPI-6400-PH interfaces.

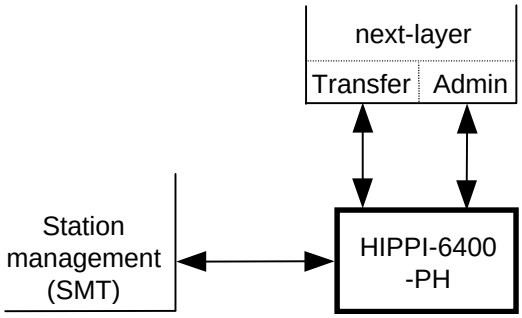


Figure 6 – HIPPI-6400-PH service interface

5.2 Service primitives

HIPPI-6400-PH service primitives are of four types.

– *Request primitives* are issued by a service user to initiate a service provided by the HIPPI-6400-PH.

– *Confirm primitives* are issued by the HIPPI-6400-PH to acknowledge a Request.

– *Indicate primitives* are issued by the HIPPI-6400-PH to notify the service user of a local event. This primitive is similar in nature to an unsolicited interrupt. Note that the local event may have been caused by a service Request.

– *Response primitives* are issued by a service user to acknowledge an Indicate.

5.3 Sequences of primitives

The order of execution of service primitives is not arbitrary. Logical and time sequence relationships exist for all described service primitives. Time sequence diagrams are used to illustrate a valid sequence. Other valid sequences may exist. The sequence of events between peer users across the user/provider interface is illustrated. In the time sequence diagrams, the HIPPI-6400-PH users are depicted on either side of the vertical bars while the HIPPI-6400-PH acts as the service provider.

In this part of ISO/IEC 11518, a second Request primitive of the same name for the same VC shall not be issued until the Confirm for the first request is received. Likewise, a second Indicate primitive of the same name for the same VC shall not be issued until the Response for the first Indicate is received. In addition, since TRANSFER and ADMIN operations of the same type share VC resources, only one at a time is allowed on the same VC.

5.4 Data transfer service primitives

These primitives, as shown in figure 7, shall be used to transfer next-layer data from an Originating Source next-layer to a Final Destination next-layer. The next-layer data shall be carried in a Message, with HIPPI-6400-PH MAC and IEEE 802.2 LLC/SNAP headers preceding the next-layer payload data (see figures 4 and 14, and clause 7). The next-layer data shall immediately follow the LLC/SNAP header.

While figure 7 shows the data being transferred after the 64_TRANSFER.Confirm is issued, this ordering is not mandatory.

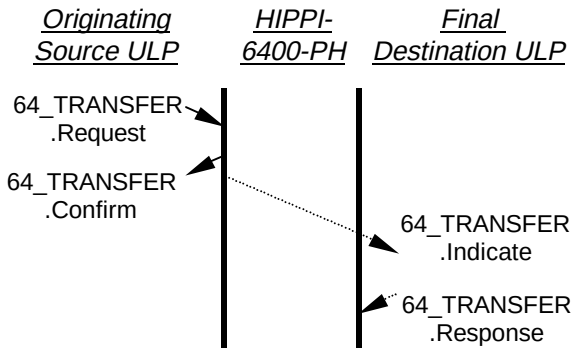


Figure 7 – Data transfer service primitives

5.4.1 64_TRANSFER.Request

This primitive is issued by the Originating Source's next-layer to request a data transfer.

Semantics – 64_TRANSFER.Request (
D_ULA,
S_ULA,
VCn,
LLC/SNAP,
Length,
Data)

The D_ULA (Destination address) shall be placed directly in the MAC header (see 7.2).

The S_ULA (Source address), if allowed by S_ULA_Allowed = true (see 5.6.1), shall be placed directly in the MAC header (see 7.2). If S_ULA_Allowed = false, then the HIPPI-6400-PH entity shall use its native S_ULA address. Note that by allowing the next-layer to specify the Source address, a server can use a "spoof" address, e.g., to provide a broadcast service. Whether the next-layer is allowed to set the S_ULA or not is controlled by the local station management entity through the S_ULA_Allowed flag (see 5.6.1).

VCn shall be the Virtual Channel (see 6.2) that the message shall be sent on.

LLC/SNAP shall be the LLC/SNAP header, including the appropriate EtherType specifying the data type (see 7.3).

Length shall specify the number of bytes of next-layer payload data. Note that the length parameter in the MAC header (see clause 7) M_len = Length + 8 to account for the LLC/SNAP header.

Data shall be the next-layer payload data.

Issued – The Originating Source next-layer issues this primitive to the HIPPI-6400-PH entity to request the transfer of the next-layer payload data to the Final Destination. Note that one 64_TRANSFER.Request or one 64_ADMIN.Request may be issued for each of the four VCs before receiving a Confirm for any of them, i.e., the .Confirm / .Request exchange is on a per-VC basis. For example, a 64_TRANSFER.Request for VC1 shall not be issued if a 64_ADMIN.Request is in progress on VC1.

Effect – The HIPPI-6400-PH entity shall accept the data for transmission and build the Message with the appropriate MAC and LLC/SNAP headers. If the Message size violates the Virtual Channel limitations (see 6.2), then this transfer request shall be rejected (see 5.4.2); otherwise, the Message shall be sent. If the Message does not end on a micropacket boundary, then padding shall be provided (see clause 7).

5.4.2 64_TRANSFER.Confirm

This primitive acknowledges the 64_TRANSFER.Request from the Originating Source next-layer.

Semantics – 64_TRANSFER.Confirm (
VCn,
Status)

VCn shall be this Message's Virtual Channel (see 6.2).

Status shall be

– accept – The Message has been accepted for transmission,
– reject – The Message was unable to be transmitted for any reason, e.g., invalid size.

Issued – The HIPPI-6400-PH shall issue this primitive to the Originating Source next-layer to acknowledge the 64_TRANSFER.Request on this VC.

Effect – Another 64_TRANSFER.Request, or a 64_ADMIN.Request, is enabled on this VC.

5.4.3 64_TRANSFER.Indicate

This primitive indicates to the Final Destination next-layer that a Message has been received from the Originating Source.

Semantics – 64_TRANSFER.Indicate (
D_ULA,
S_ULA,
LLC/SNAP,
Status,
Length,
Data)

The D_ULA shall be the value received in the MAC header (see 7.2).

The S_ULA shall be the value received in the MAC header (see 7.2).

LLC/SNAP shall be the received Message's LLC/SNAP header (see 7.3).

Status denotes whether the data payload being delivered was received with errors. Status includes but is not limited to:

– ECRC errors (see 9.1.3);
– missing end of Message (see 9.2.3);
– data overrun (see 9.2.5);
– data underrun (see 9.2.5).

Data overrun and underrun indications to the next-layer are optional, but logging is required (see 9.2.5).

Length shall be the payload length as specified in the MAC header, i.e., Length = M_len – 8.

Data shall be the next-layer payload data, with any pad or overrun (i.e., beyond Length), removed.

Issued – The Final Destination HIPPI-6400-PH shall issue this primitive to the Final Destination next-layer when a Message has been received.

Effect – Unspecified

5.4.4 64_TRANSFER.Response

This primitive acknowledges a 64_TRANSFER.Indicate.

Semantics – 64_TRANSFER.Response ( )

Issued – The Final Destination next-layer issues this primitive to acknowledge the receipt of the 64_TRANSFER.Indicate.

Effect – The HIPPI-6400-PH Final Destination is enabled to issue another 64_TRANSFER.Indicate.

5.5 Admin service primitives

These primitives, as shown in figure 8, shall be used to transfer Admin micropackets (see 6.3.6). Admin micropackets, as defined in ISO/IEC 11518-11[2], are used for support and initialization of HIPPI-6400 links, Elements, and systems. Control service primitives (see 5.6) used to affect the local interface may use Admin micropackets or other unspecified means.

While figure 8 shows the Admin micropacket being transferred after the 64_ADMIN.Confirm is issued, this ordering is not mandatory.

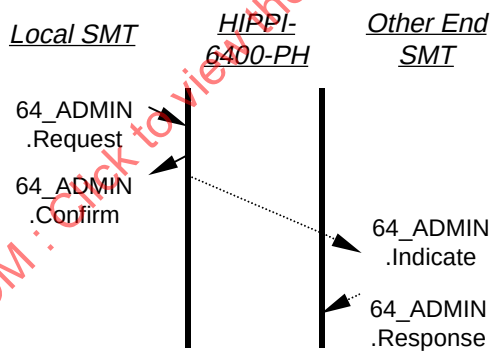


Figure 8 – Admin service primitives

5.5.1 64_ADMIN.Request

This primitive is issued to request the transfer of an Admin micropacket.

Semantics – 64_ADMIN.Request (
VCn
Admin micropacket)

VCn shall be the Virtual Channel that the Admin micropacket shall be sent on (see 6.2).

The Admin micropacket contents are specified in ISO/IEC 11518-11[2].

[2] ISO/IEC 11518-11, *Information technology – High-Performance Parallel Interface – 6 400 Mbit/s Switch Control (HIPPI-6400-SC)* (under consideration).

Issued – The next-layer issues this primitive to the HIPPI-6400-PH entity to request the transfer of an Admin micropacket to the desired Element. Note that one 64_ADMIN.Request or one 64_TRANSFER.Request may be issued for both VC1 and VC2 before receiving a Confirm for either of them, i.e., the .Confirm / .Request exchange is on a per-VC basis. For example, a 64_ADMIN.Request for VC1 shall not be issued if a 64_TRANSFER.Request is in progress on VC1.

Effect – The HIPPI-6400-PH entity shall accept the Admin micropacket for transmission.

5.5.2 64_ADMIN.Confirm

This primitive acknowledges the 64_ADMIN .Request from the next-layer.

Semantics – 64_ADMIN.Confirm (
VCn,
Status)

VCn shall be the Admin micropacket's Virtual Channel (see 6.2).

Status shall be:

– Accept – The Admin micropacket has been accepted for transmission.
– Reject – The Admin micropacket was unable to be transmitted.

Issued – The HIPPI-6400-PH entity shall issue this primitive to the next-layer to acknowledge the 64_ADMIN.Request on this VC.

Effect – Another 64_ADMIN.Request, or a 64_TRANSFER.Request, is enabled on this VC.

5.5.3 64_ADMIN.Indicate

This primitive indicates that an Admin micropacket has been received.

Semantics – 64_ADMIN.Indicate (
VCn,
Status,
Admin micropacket)

VCn shall be the Admin micropacket's Virtual Channel (see 6.2).

Status denotes whether the Admin micropacket being delivered was received with errors. Status includes but is not limited to:

– ECRC errors (see 9.1.3);
– missing TAIL bit (see 9.2.1);
– missing end of Message (see 9.2.3).

Admin micropacket is the contents.

Issued – The HIPPI-6400-PH entity shall issue this primitive to the next-layer when it receives an Admin micropacket.

Effect – Unspecified

5.5.4 64_ADMIN.Response

This primitive acknowledges a 64_ADMIN.Indicate.

Semantics – 64_ADMIN.Response (VCn)

VCn shall be the Admin micropacket's Virtual Channel (see 6.2).

Issued – The next-layer issues this primitive to acknowledge the receipt of the 64_ADMIN.Indicate.

Effect – The HIPPI-6400-PH is enabled to issue another 64_ADMIN.Indicate on this Virtual Channel.

5.6 Control service primitives

These primitives, as shown in figure 9, may be used by the local station management (SMT) entity to set parameters and control the local HIPPI-6400-PH entity. 64_ADMIN primitives (see 5.5) may also be used to control the local HIPPI-6400-PH entity, and are not shown in figure 9.

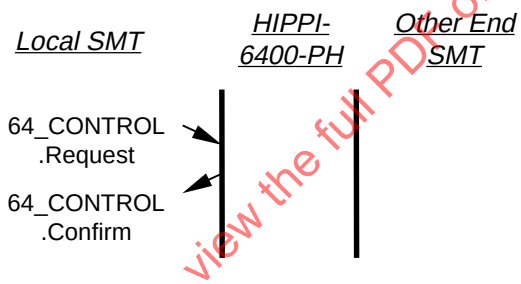


Figure 9 – Control service primitives

5.6.1 64_CONTROL.Request

This primitive is issued by the local SMT to set parameters, or otherwise control the local HIPPI-6400-PH entity. Several functions are specified and others are left to specific implementations.

Semantics – 64_CONTROL.Request (
Command,
Command_Parameters)

Command specifies the function to be performed.

Command_Parameters are specific to each command.

The commands and parameters include but are not limited to:

– Set/reset S_ULA_Allowed flag (see 5.4.1);
– Set native S_ULA value (see 7.2);
– Set timeout values (see table 6);
– Link Reset (see 12.2);
– Initialize (see 12.3);
– Initialize logged events counters (see table 7).

Issued – The SMT issues this primitive to perform some control function.

Effect – The HIPPI-6400-PH shall perform the function specified.

5.6.2 64_CONTROL.Confirm

This primitive acknowledges the 64_CONTROL.Request from the SMT.

Semantics – 64_CONTROL.Confirm (Status)

Status reports the success or failure of the 64_CONTROL.Request commands.

Issued – The HIPPI-6400-PH entity shall issue this primitive to the SMT in response to a 64_CONTROL.Request.

Effect – Enables another 64_CONTROL.Request.

5.7 Status service primitives

These primitives, as shown in figure 10, may be used to obtain status information from the local HIPPI-6400-PH entity.

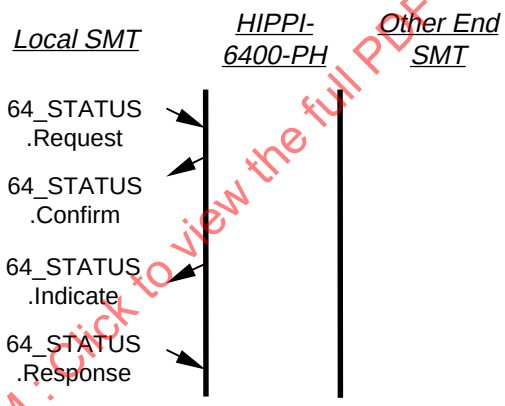


Figure 10 – Status service primitives

5.7.1 64_STATUS.Request

This primitive is issued by the local SMT to request a status report.

Semantics – 64_STATUS.Request (
Requested_item(s))

Requested_item(s) may include, but are not limited to:

– S_ULA_Allowed state (see 5.4.1);
– Native S_ULA value (see 7.2);
– Timeout values (see table 6);
– Logged events (see table 7);
– Activity monitor state (see 13.1);
– Link state, i.e., Normal, Resetting, Initializing, or Shutdown (see clause 12).

Issued – The local SMT issues this primitive when it wishes to obtain the status of the HIPPI-6400-PH entity. Note that an implementation, in a vendor specific fashion, may issue a blanket request for all of the requested items, or for specific items.

Effect – The HIPPI-6400-PH entity shall respond with a 64_STATUS.Confirm.

5.7.2 64_STATUS.Confirm

This primitive replies to the previous 64_STATUS.Request with status information.

Semantics – 64_STATUS.Confirm (
Requested_item(s))

Requested_item(s) shall contain, but not be limited to the items requested in the 64_STATUS.Request.

Issued – The HIPPI-6400-PH entity shall issue this primitive to the SMT in response to a 64_STATUS.Request.

Effect – Enables another 64_STATUS.Request.

5.7.3 64_STATUS.Indicate

This primitive informs the SMT entity that a monitored event has occurred that affects the operation of the HIPPI-6400-PH entity.

Semantics – 64_STATUS.Indicate

Issued – The HIPPI-6400-PH shall issue this primitive whenever a monitored event is detected. Monitored events shall include, but are not limited to:

– activity monitor indication going false (see 13.1);
– link going into Shutdown (see 13.2).

Effect – Unspecified. A normal response would be for the SMT entity to read the status and determine which event occurred.

5.7.4 64_STATUS.Response

This primitive acknowledges the 64_STATUS.Indicate.

Semantics – 64_STATUS.Response

Issued – The SMT entity issues this primitive to acknowledge receipt of the 64_STATUS.Indicate.

Effect – The HIPPI-6400-PH is allowed to issue another 64_STATUS.Indicate.

6 Micropacket contents

6.1 Bit and byte assignments

Table 2 specifies the contents that shall be in each micropacket. As shown in figure 3, each micropacket shall consist of 32 data bytes and 64 bits of control information. The data bytes shall be numbered DB00 – DB31. DB00 shall be transmitted first. The data bits in the micropacket shall be numbered d*xx.y* where *xx* is the byte number and *y* is the bit number in the byte.

The 64 bits of control information shall be numbered as bits c63 – c00. Control bit c00 shall be transmitted first. As shown in figure 3, a field with a numerical value shall have its most-significant bit in the highest numbered bit position.

The control information shall contain the following parameters located in the bits specified (see figure 11). The Source side of a link supplies all of the parameters, except for RSEQ, VCR and CR which come to the Source from its local Destination side. The VC parameter comes from the Originating Source. The TAIL, TYPE, and ECRC parameters normally come from the Originating Source, but may under error conditions come from an intermediate device (see 9.2.3 and 9.2.4).

VC (2 bits, c01–c00) – The Virtual Channel selector. (See 6.2.)

TYPE (4 bits, c05–c02) – Identifies the type of information within the micropacket. (See 6.3.)

TAIL (1 bit, c06) – TAIL = 1 identifies the last micropacket of a Message. TAIL = 0 means that more micropackets for this Message follow.

ERROR (1 bit, c07) – ERROR = 1 means that an unrecoverable error has been detected in the Message, do not check the ECRC. ERROR = 0 means that the Message is OK so far. (See 6.6.3 and 9.1.3.)

NOTE Unrecoverable errors may include other upstream media errors, e.g., an LLRC error on a HIPPI-800 link.

VCR (2 bits, c09–c08) – Virtual Channel number associated with credit addition. (See 6.5.)

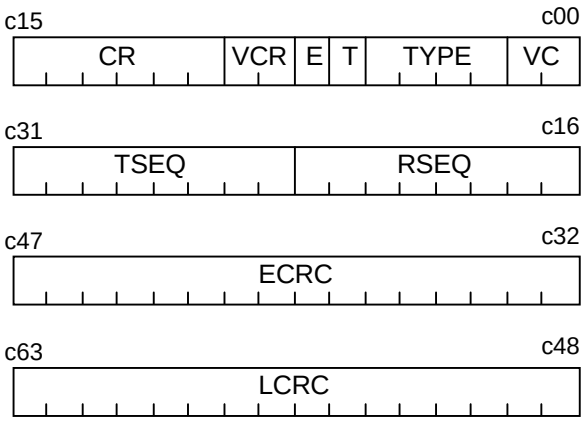
CR (6 bits, c15–c10) – Amount of credit to add to the Virtual Channel specified in VCR. (See 6.5.)

RSEQ (8 bits, c23–c16) – Sequence number associated with micropacket ACK indication. (See 6.4.)

TSEQ (8 bits, c31–c24) – Sequence number of transmitted micropacket. (See 6.4.)

ECRC (16 bits, c47–c32) – End-to-end checksum covering all of the data bytes up to this point in a Message, including those in the Header micropacket. (See 6.6.)

LCRC (16 bits, c63–c48) – Link level checksum covering the 32 data bytes, and the c00 through c47 control bits, in this micropacket. (See 6.6.)



NOTE Transmission order is top to bottom, and right to left, in 4-bit groups, as shown in tables 3 and 4. The most-significant-bit of a parameter is at the left end of its field.

Figure 11 – Control bits summary

6.2 Virtual Channel (VC) selector

Four Virtual Channels shall be available in each direction on a link. Messages on the Virtual Channels shall be assigned as follows:

– VC0 = Messages with a maximum size of 68 data micropackets (2 176 bytes) plus a Header micropacket.

– VC1 = Messages with a maximum size of 4100 data micropackets (~128 KBytes) plus a Header micropacket. VC1 also carries Admin Command micropackets.

– VC2 = Messages with a maximum size of 4100 data micropackets (~128 KBytes) plus a Header micropacket. VC2 also carries Admin Response micropackets.

– VC3 = Messages with a maximum size of 134 217 728 data micropackets (~4 GBytes) plus a Header micropacket. To avoid congestion, Originating Sources shall only initiate VC3 transfers to Final Destinations that have agreed to accept them, e.g., by using the Scheduled Transfer Protocol or by other unspecified means.

NOTE The maximum Message size for VC0, VC1, and VC2 was picked to be an integral power of 2, plus up to 128 bytes for next-layer header(s). For example, VC0's maximum Message size is 69 micropackets: one Header micropacket, four micropackets carrying 128 bytes of next-layer header(s), and 64 micropackets carrying 2 048 bytes of user payload.

6.3 Micropacket TYPEs

The 4-bit TYPE parameter shall indicate the contents of the micropacket.

Micropackets whose TYPE < x'8', or whose TYPE = x'A', are provided for control at the link level or for credit update. These micropackets are not loaded into any VC Buffer (see figure 5) at the Destination despite the VC field being transmitted as x'0'. As such, the Source need not have credit available for VC0 prior to sending these micropackets, and the Destination shall not generate additional VC0 credit as a result of having received these micropackets.

Only micropackets whose TYPE ≥ x'8' consume sequence numbers (see 6.4). Therefore, only these micropackets shall be retransmitted.

Undefined TYPE values are reserved for future use. Actions to be taken as a result of receiving an undefined TYPE are detailed in 9.1.4.

6.3.1 TYPE = link control micropackets

Control micropackets operate at the link level, do not carry any user data, acknowledgements, or credit update information (see clause 12). Control micropackets may occur during the transmission of a Message (see clause 7). Control micropackets include:

– Reset (TYPE = x'2') – Sent to initiate a Link Reset operation. (See 12.2.)

– Reset_ACK (TYPE = x'3') – The receiving device has completed the Link Reset operation.

– Initialize (TYPE = x'4') – Sent to initiate an Initialization operation. (See 12.3.)

– Initialize_ACK (TYPE = x'5') – The receiving device has completed the Initialization operation.

NOTE Reset_ACK and Initialize_ACK micropackets should be discarded if received during normal operation.

6.3.2 TYPE = Null micropackets

Null micropackets (TYPE = x'7') are gap-fillers, and shall be used to keep the link active when there are no other micropackets to transmit. Null micropackets may carry ACK indications. Null micropackets may occur during the transmission of a Message (see clause 7).

6.3.3 TYPE = Data micropackets

Data micropackets (TYPE = x'8') carry payload.

6.3.4 TYPE = Header micropackets

Header micropackets (TYPE = x'9') carry routing and control information.

6.3.5 TYPE = Credit-only micropackets

When credits are available, and there are no Data micropackets to send, then Credit-only micropackets (TYPE = x'A') are used to carry credit update information, and acknowledgements. Credit-only micropackets may occur during the transmission of a Message (see clause 7).

6.3.6 TYPE = Admin micropackets

Admin micropackets (TYPE = x'F') are used for support and initialization of HIPPI-6400 links, Elements, and systems. Admin micropacket contents and uses are specified in ISO/IEC 11518-11[3)], HIPPI-6400-SC. Admin micropackets shall not be sent on a VC currently transmitting a Message (see clause 7).

6.4 Sequence number parameters

The transmit sequence number (TSEQ) shall increment by one for each micropacket transmitted whose TYPE ≥ x'8'. TSEQ shall wrap from x'FE' to x'00'. The receive sequence number (RSEQ) shall be used to acknowledge (ACK) these micropackets. RSEQ shall equal the TSEQ of the most recent micropacket being acknowledged, or the latest TSEQ of a contiguous group of micropackets being acknowledged (see 9.3 and 8.2). TSEQ shall begin with the value = x'00' after a Link Reset. RSEQ = x'FF' indicates that no ACK indication is being transmitted (used while the link fills with micropackets after a Link Reset). TSEQ shall not equal or surpass RSEQ, i.e., there shall be no more than 254 unacknowledged micropackets.

NOTE 1 The TSEQ and RSEQ parameters are independent of the Virtual Channel used to transmit the micropacket.

NOTE 2 The TSEQ and RSEQ parameters are local to a specific link. For example, a micropacket that transverses more than one link will most likely have different TSEQ numbers on the different links.

NOTE 3 The first micropacket with TYPE ≥ x'8' following a stomped micropacket (see 6.6.2.1) uses the same TSEQ value as in the stomped micropacket since that TSEQ value was not consumed.

The wrap at x'FE' shall be taken into account when processing ACK indications. For example, if the previous ACK indication had RSEQ = x'F7', and an ACK indication with RSEQ = x'03' is received, then the micropackets whose TSEQ value = x'F8' through x'FE', and from x'00' through x'03', are acknowledged and their memory may be reused by the Source.

---

3) ISO/IEC 11518-11, *Information technology – High-Performance Parallel Interface – 6 400 Mbit/s Switch Control (HIPPI-6400-SC)* (under consideration).

Table 2 – Micropacket contents summary

	Reset/ Initialize	Null	Credit-only	Header	Data	Admin
Data Bytes contents	0*	0*	0*	32 bytes of header information (see 7.2)	32 bytes of payload	Administrative information
VC	0*	0*	0*	any	any	Requests on VC1 Responses on VC2
TYPE (hex)	2,3,4,5	7	A	9	8	F
TAIL	1*	0*	0*	= 1 on last micropacket of Message	= 1 on last micropacket of Message	1
ERROR	0*	0*	0	= 1 if error	= 1 if error	= 1 if error
TSEQ	x'FF'	x'FF'	increments	increments	increments	increments
RSEQ	1*	ACK	ACK	ACK	ACK	ACK
VCR	0*	0*	any	any	any	any
CR	0*	0*	any	any	any	any
LCRC	single	single	single	single	single	single
ECRC	single	single	single	accumulating	accumulating	single

0* = transmit all bits of this field as 0's, a receiver must permit any value

1* = transmit all bits of this field as 1's, a receiver must permit any value

any = any data value as appropriate

single = this CRC is calculated and checked for this single micropacket

accumulating = ECRC as defined in 6.6.3

6.5 Credit update parameters

The Destination shall insert the VCR and CR parameters in micropackets to inform the Source that CR number of micropacket buffers have been freed up for the VC indicated by VCR. The Source shall increase its Credit Counter for this Virtual Channel by the value in CR. The Source Credit Counter range shall be 255, and the number of outstanding credits shall be ≤255.

NOTE 1 The CR value is an incremental update value, not the number of buffers currently available in the Destination.

NOTE 2 At 40 ns per micropacket, and 5 ns per meter of cable, each credit is equivalent to about 8 m of cable. Hence, a Credit Count, and Destination buffer capacity per Virtual Channel, of 255 will support full bandwidth on a 1 km link when round trip time is taken into account and Destination latency is low.

NOTE 3 If the Destination does not send adequate credits then the Source may not be able to send on some VCs.

6.6 Check functions

6.6.1 Intended use of CRCs

Two 16-bit cyclic redundancy checks (CRCs) shall be used. The link CRC (LCRC) checks all of the data bytes (including any pad), and control bits in a single micropacket. The LCRC shall be generated by a link's Source, and checked by the same link's Destination, i.e., it is local to a link.

The end-to-end CRC (ECRC) checks all of the data bytes (including any pad), of a Message up to the end of the current micropacket of the Message, i.e., it may cover multiple micropackets. The ECRC shall be generated by the Originating Source, and should be passed unchanged through intermediate link-level devices. The ECRC shall be checked at each Destination in the path. See 9.1.3 for ECRC error operations.

The LCRC and ECRC shall not be generated or checked for a training sequence (see 11.1).

While this part of ISO/IEC 11518 covers the link level and host interface, other documents may require intermediate link-level devices to carry the ECRC across them, for example, across switches. Link-level devices, as described here, are devices that do not operate on the payload portion of data micropackets.

6.6.2 Link-level CRC (LCRC)

The link CRC (LCRC) shall cover all of the data bytes, and the control bits except for itself. The LCRC generator and checker shall be initialized to all ones (x'FFFF') for each micropacket.

The LCRC polynomial shall be:

$$x^{16} + x^{12} + x^5 + 1$$

Figure 12 shows an example of a serial implementation. The LCRC may be implemented in a parallel fashion rather than serial, but must produce the same results as the serial example. The c63 through c48 bits are the LCRC bits in the control word. The incoming data and control bits are exclusive OR'd with c48 to generate a *sum* value. The *sum* value is exclusive OR'd with selected control bits as they are shifted right once each bit period. The data and control bits shall be input to the generator in transmission sequence, i.e., 64 data bits, 16 control bits, 64 data bits, 16 control bits, etc. The sequence is d00.0, d00.1, d00.2, ...d00.7, d01.0...d01.7, ...d07.7, c00, c01, c02...c15, d08.0...d15.7, c16...c31, d16.0...d23.7, c32...c47, d24.0...d31.7. Refer to tables 3 and 4 for the transmission sequence. After passing all 304 input bits, c63-through-c48 contain the most-significant through least-significant bits of the LCRC.

At the destination, the LCRC check may be implemented by clocking the entire micropacket, including the LCRC parameter (c63..c48), into either a serial or parallel checker. In this case, a residue is available in the checker register after the last clock rather than a syndrome. If this check method is used, a residue of x'0000' indicates no errors, and x'06A9' indicates that a "stomp" code was received.

See 9.1.1 for details of a Destination's actions when checking the LCRC. See clause A.3 for the equations to generate the LCRC in a parallel fashion.

6.6.2.1 Stomp code at Source

A Source may decide during the course of transmitting a micropacket that it wishes to "nullify" that transmission. This shall be done by XORing a "stomp" code of x'874D' with the LCRC that it has calculated for the micropacket. The Source shall treat a "stomped" micropacket as if it never occurred, i.e., not save the "stomped" micropacket in the retransmit buffer, and not increment the TSEQ number since the TSEQ number was not consumed.

6.6.2.2 Stomp code at Destination

If the Destination detects a "stomp" code (see 6.6.2), then an LCRC error shall not be logged (see 9.1.1).

6.6.3 End-to-end CRC (ECRC)

The end-to-end CRC (ECRC) shall include only the micropacket's data bytes, not the control bits, in its calculation. The ECRC shall include all of a Message's data bytes up to this point in the Message, i.e., the data bytes in the Header micropacket and in all of the Data micropackets up to this point in the Message.

If ERROR ≠ 1, then all Sources not generating the original ECRC shall check the ECRC prior to transmission, and if the ECRC is in error then set ERROR = 1 in this micropacket's control bits. An ECRC_Source_Error shall be logged for only the first occurrence of this error in a Message (see 14.1). This aids in error isolation and prevents endless retransmission loops.

The ECRC generator polynomial shall be:

$$x^{16} + x^{12} + x^3 + x + 1$$

The ECRC is calculated and maintained independently for each VC. The ECRC checker and generator for a VC shall be initialized to all ones (x'FFFF') for single coverage micropacket TYPEs (see table 2) and for a particular VC at the beginning of a Message on that VC (i.e., when the previous micropacket had TAIL = 1 or the current micropacket has TYPE = Header).

Figure 13 shows an example of an ECRC serial implementation. The ECRC may be implemented in a parallel fashion rather than serial, but must produce the same results as the serial example. The c47 through c32 bits are the ECRC bits in the control word. The incoming data bits are exclusive OR'd with c32 to generate a *sum* value; the *sum* value is exclusive OR'd with selected control bits as they are shifted right once each bit period. The data bits shall be input to the generator in transmission sequence, i.e., d00.0, d00.1, d00.2, ...d00.7, d01.0...d01.7, ...d31.7. Refer to tables 3 and 4 for the transmission sequence. After passing all 256 of the micropacket's data bits, c47-through-c32 contain the most-significant through least-significant bits of the ECRC for this micropacket. The ECRC value will normally be different for each micropacket of a Message since the ECRC accumulates as the Message progresses (see table 1).

See 9.1.3 for details of a Destination's actions when checking the ECRC. See clause A.4 for the equations to generate the ECRC in a 64-bit-wide fashion.

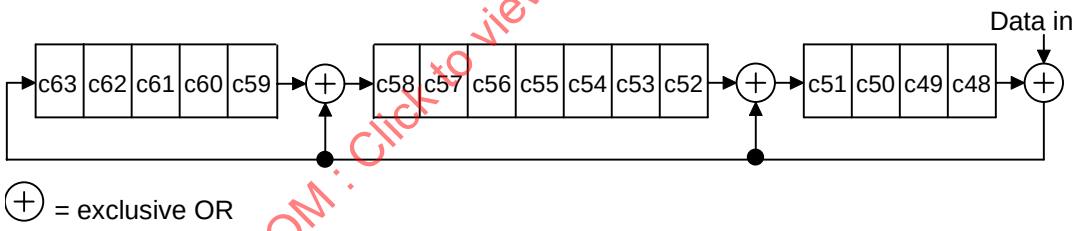


Figure 12 – LCRC implementation example

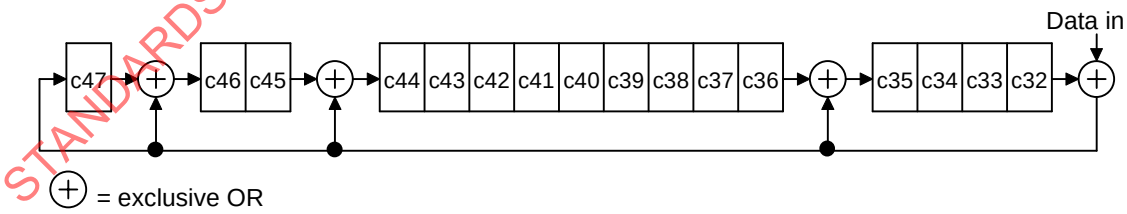


Figure 13 – ECRC implementation example

7 Message structure

7.1 Overview

As defined in 4.5, a Message is an ordered sequence of one or more micropackets that have the same VC, start with a Header micropacket (TYPE = Header), and have TAIL = 1 in the last micropacket. Each VC may only have a single Message in progress at any time. Admin micropackets shall only be sent on VCs not transmitting a Message. Reset, Reset_ACK, Initialize, Initialize_ACK, Null, and Credit-only micropackets may occur during the transmission of a Message.

Since only complete micropackets are transmitted, a Message that is not an integral multiple of 32 bytes in length shall be padded with zeros in the last micropacket.

The Message header format is shown in figure 14 as a group of 32-bit words. The Media Access Control (MAC) header, and LLC/SNAP header, shall reside in the first 24 bytes of all Header micropackets. If a parameter uses more than one byte, the lowest numbered byte is the most-significant byte. The last eight bytes of the Header micropacket may be used by other protocols, and are not defined in this part of ISO/IEC 11518.

7.2 MAC header

The MAC header shall be included in all HIPPI-6400 Messages. The MAC header shall be in the first micropacket (TYPE = Header) of a Message, and shall contain:

D_ULA (48 bits, DB00-DB05) – The IEEE 48-bit ULA network address, as defined in ISO/IEC 8802, identifying the payload's Final Destination. Figure 15 (following ISO/IEC 8802. A canonical bit order, and HIPPI byte order) details the placement of the D_ULA.

S_ULA (48 bits, DB06-DB11) – The IEEE 48-bit ULA network address, as defined in ISO/IEC 8802, identifying the payload's Originating Source. Figure 15 details the placement of the S_ULA.

M_len (32 bits, DB12-DB15) – The Message length, in bytes, following the M_len field, exclusive of any padding in the last micropacket.

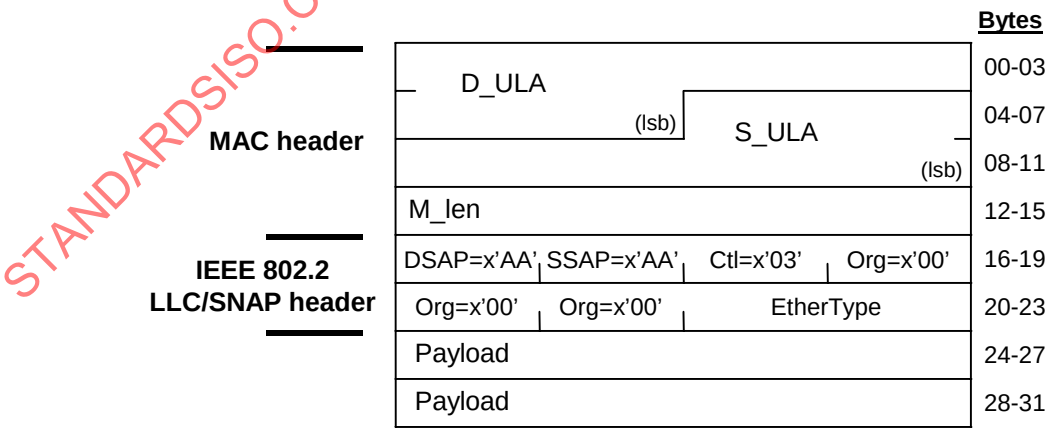


Figure 14 – Header micropacket contents

D_ULA Octet 0 U/L I/G	D_ULA Octet 1	D_ULA Octet 2	D_ULA Octet 3
D_ULA Octet 4	D_ULA Octet 5	S_ULA Octet 0 U/L I/G	S_ULA Octet 1
S_ULA Octet 2	S_ULA Octet 3	S_ULA Octet 4	S_ULA Octet 5

NOTE U/L = 0 for Universal address, 1 for Locally administered; I/G = 0 for Individual address, 1 for Group.

Figure 15 – Detailed ULA layout

7.3 LLC/SNAP header

The LLC/SNAP header, as defined in ISO/IEC 8802-2, shall be included in all Messages. The LLC/SNAP header shall be 64 bits (DB16-DB23) and shall immediately follow the MAC header in the first micropacket (Header micropacket). The values of the LLC/SNAP header subfields shall be: DSAP = x'AA' (i.e., SNAP), SSAP = x'AA' (i.e., SNAP), Ctl = x'03' (i.e., unnumbered packets), and the three Org = x'00' (i.e., generic packets). Codings of the EtherType field shall be as assigned in the "Assigned Numbers" RFC, e.g., RFC 1700 (see http://www.iana.org/iana/). For the convenience of the reader, HIPPI-6400-specific EtherTypes are listed below:

– x'8180' = Carrying HIPPI Framing Protocol (HIPPI-FP) (specified in ISO/IEC 11518-2).

– x'8181' = Carrying Scheduled Transfer Protocol.

– x'8182' = Carrying Bridge Protocol Data Units for constructing a spanning tree (specified in ISO/IEC 15802-3 (1998)).

– x'8183' = Reserved.

7.4 Payload

The eight bytes following the LLC/SNAP header belong to the next-layer using this Message. The payload bytes may be used to carry additional headers, parameters, or data.

8 Source specific operations

8.1 Credit update indications on Source side

Credit update indications from the remote end are received on the local Destination side, and passed to the local Source side, as shown in figure 5. A credit update shall increase the available credit, by the amount in the CR parameter, on the Virtual Channel whose number is the value in the VCR parameter.

If data is ready to be sent on a given VC, but credits are exhausted for this VC (i.e., credit = 0) for the duration of a timeout period, then the link is shut down (see 13.2), and a VC[0-3]_Credit_Timeout_Error logged. The default timeout value shall be 2 seconds (see 14.1).

If a credit update results in credit >255, then the link shall be reset (see 12.2) and a VC[0-3]_Credit_Overflow_Error logged.

8.2 ACK indications on Source side

ACK indications (see 6.4 and 9.3) from the remote end are received on the local Destination side, and passed to the local Source side, as shown in figure 5. An ACK indication acknowledges all of the transmitted micropackets whose TSEQ ≤ RSEQ, i.e., the memory allocated to these micropackets may be reused. RSEQ = x'FF', which may occur immediately after a Reset operation (see 9.3), shall be ignored.

The ACK indication timeout indicates that a TSEQ was transmitted, but not acknowledged for the length of time longer than the worst-case round trip time possible for an acknowledgement to occur. If the ACK indication timeout expires, the Source shall retransmit all micropackets, (see 8.4), that have not been acknowledged, and shall log an RSEQ_Missing_Error (see 14.1). The ACK indication timeout default value shall be 12 µs (see 14.1).

NOTE The ACK indication timeout provides a recovery mechanism even in the event of lost RSEQ values due to link errors. Faster recovery may be possible with other schemes, e.g., NAKs, but the complexity required for the performance gain did not seem worth it, especially since errors should be infrequent.

If an illegal RSEQ value is received, the Source shall retransmit all micropackets, (see 8.4), that have not been acknowledged, and log a RSEQ_Out_Of_Range_Error (see 14.1). An illegal RSEQ is one that does not equal or fall between the last successfully received RSEQ and the highest transmitted but not acknowledged TSEQ.

8.3 ACKs and credit updates to remote end

The local Destination side sends ACK indications and credit update information to the remote end by first queuing them to the local Source side, as shown in figure 5. The Source side shall transmit this information in micropackets using the appropriate control bits. Since the ACK indications and credit update information do not share their fields with any other parameters they can be sent with every micropacket.

The local Destination may queue multiple ACK indication RSEQ parameters before one is transmitted by the local Source end. The RSEQ parameter should be overwritten so that the ACK indication Message transmitted uses the latest value of RSEQ.

8.4 Micropacket retransmission

A retransmission sequence, as triggered by the error conditions defined in 8.2, shall consist of two consecutive training sequences (see clause 11) followed by retransmission of all of the unacknowledged micropackets in the Output Buffer (see figure 5). Each retransmission sequence shall log a Retry_Count (see 14.2).

Multiple retransmissions may be required in the event of poor link quality. If successful operation is not achieved after a number of successive retransmissions of the same data, then the link shall be shut down (see 13.2), and a Retry_Failure_Error logged (see 14.2). The default number of consecutive retransmissions is two, and it shall be programmable to other values, including 1 and 4. The mechanisms and procedures used to set values, different from the default value, are outside the scope of this part of ISO/IEC 11518.

NOTE This number of allowed consecutive retransmissions may need to be larger to accommodate lengthy noise hits and/or to provide equivalent noise immunity when smaller ACK indication timeout values (see 8.2) are chosen for short cables.

Upon retransmission, the following parameters, from the original micropacket, shall have the same value in the retransmitted micropacket.

– VC
– TYPE
– TAIL
– ERROR
– TSEQ
– VCR
– CR
– ECRC

The following parameters may change as a micropacket is retransmitted.

– RSEQ

– LCRC

9 Destination specific operations

9.1 Link level processing

The Destination shall process received micropackets in the order of the following subclauses. The unnumbered items within each subclause may be checked in any order. Note that no acknowledgement (i.e., with RSEQ) shall be given for a micropacket that is discarded.

9.1.1 Check received LCRC

– If LCRC syndrome = x'06A9' (stomp code) then the Destination shall discard the micropacket, and not log an error.

– If LCRC syndrome ≠ x'0000', and ≠ x'06A9' (stomp code), then the Destination shall discard the micropacket and log an LCRC_Error.

9.1.2 Check received TSEQ

If no errors were detected in 9.1.1 then the following checks shall be made. TSEQ = x'FF' and TYPE ≥ x'8' is an error. TSEQ ≠ x'FF' and TYPE < x'8' is an error. TYPE ≥ x'8', and TSEQ not one greater than the last non-x'FF', non-stomped, TSEQ received, is also an error. In the above cases, the micropacket shall be discarded. Additionally, a TSEQ_Error shall be logged unless no micropackets with TYPE ≥ x'8' have been accepted since the last TSEQ_Error was logged.

9.1.3 Check received ECRC

If no errors were detected in 9.1.1 or 9.1.2, then the following checks shall be made.

– If ERROR = 0 and the ECRC syndrome ≠ x'0000', then the Destination shall discard the micropacket and log an ECRC_Error.

– If ERROR = 1, then the Destination shall process the micropacket as if the ECRC were correct (unless this is the Final Destination in which case an error shall be signaled to the next-layer).

9.1.4 Undefined TYPE

If TYPE = undefined and is in the range of x'0' - x'7', then the Destination shall treat the micropacket as a Null micropacket. If TYPE = undefined and in the range of x'8' to x'F' then intermediate Destinations shall treat the micropacket as a Data micropacket. Treatment by a Final Destination is not specified by this part of ISO/IEC 11518. For any undefined TYPE value, a VC[0-3]_Undefined_TYPE_Error shall be logged and the most recent offending TYPE value stored in Undefined_TYPE_Value.

NOTE The actions applied to Undefined TYPEs are intended to allow for future use of the Undefined TYPE values.

9.2 Check for Message protocol errors

Message protocol error checking (at the Destination) shall be done on micropackets that have not been discarded in 9.1 and its subclauses. Since a Message is restricted to a single Virtual Channel, all Message protocol checking shall be applied to each Virtual Channel independently. Credit-only (TYPE = x'A') micropackets shall be ignored for the purposes of Message protocol checking. Otherwise, micropackets shall be checked in the order received on each Virtual Circuit.

9.2.1 Admin missing TAIL bit

If TYPE = Admin, and Tail = 0, then the Destination shall forward the Admin micropacket with ERROR = 1, and TAIL = 1. A VC[1-2]_Admin_Tail_Error shall be logged.

9.2.2 Missing start of Message

If a Message is missing the Header micropacket (i.e., a micropacket with TYPE = Data or undefined is received following a micropacket with TAIL = 1, or a Link Reset operation) then the Destination shall process the micropackets on this VC until a micropacket with TYPE = Header or Admin is received. This processing for the micropackets shall consist of discarding the data bytes; their control information shall be treated normally and RSEQs shall be generated. Subsequent Header or Admin micropackets shall be treated normally. The Destination shall log a VC[0-3]_Missing_Start_of_Message_Error for each discarded Message; not log an error for each discarded micropacket.

9.2.3 Missing end of Message

If the end of a Message is missing (i.e., TYPE = Header or Admin following a Data, Header, or undefined TYPE ≥ x'8' micropacket with TAIL = 0), then the Destination shall fabricate an end of Message micropacket (Data Bytes = x'00', VC = as received, TYPE = Data, TAIL = 1, ERROR = 1, other parameters as appropriate). The Destination shall insert the fabricated micropacket into the VC stream, and shall log a VC[0-3]_Missing_End_of_Message_Error. The Header or Admin micropacket shall be treated normally.

9.2.4 Stall timeout

If a Message is in progress on a VC, that VC's buffer is empty, and no Data micropackets have been received within the Stall timeout period, then the Destination shall fabricate an end of Message micropacket (Data Bytes = x'00', VC = as received, TYPE = Data, TAIL = 1, ERROR = 1, other parameters as appropriate). The Destination shall insert the fabricated micropacket into the VC stream, and shall log a VC[0-3]_Stall_Timeout_Error. This action flushes the Message in progress. The default value of the Stall timeout shall be 2 ms (see 14.1).

NOTE Implementors are cautioned that the Stall timeout may be triggered by a slow Source host. If slow hosts are expected, then the Stall timeout value may be set to a larger value to avoid inadvertent actions.

9.2.5 Message length errors

Final Destinations shall limit Message payloads delivered to a next-layer to ≤ M_len – 8 bytes (see 7.2). Any pad or overrun bytes beyond this length shall be discarded (see 5.4.3). A Message's last legitimate payload byte, i.e., the (M_len – 8)th byte, shall be in the Message's last micropacket, (i.e., the micropacket with Tail = 1). An Overrun_Error shall be logged if the micropacket containing the last legitimate payload byte has Tail = 0. An Underrun_Error shall be logged if a micropacket with Tail = 1 occurs before the last legitimate payload byte has been received.

9.2.6 No errors detected

If no errors are detected, and TYPE = Header or Data, the micropacket shall be acknowledged and delivered to the Virtual Channel buffer designated by the VC parameter.

If no errors are detected, and TYPE ≠ Header or Data, the micropacket shall be processed by the Destination.

9.3 Generating ACKs

The Destination acknowledges correctly received micropackets by using the RSEQ parameter of micropackets flowing in the reverse direction. Multiple micropackets may be acknowledged with a single RSEQ (e.g., if micropackets with TSEQ = 0,1...7 are received, transmitting RSEQ = 5 acknowledges micropackets 0,1...5, but not 6 and 7). Only micropackets that are not discarded due to errors (see 9.1) and whose TYPE value is in the range x'8' to x'F' shall be acknowledged. If the Destination does not have a new value of RSEQ to send, it shall repeat the last RSEQ value.

Once an error is detected that causes a micropacket not to be acknowledged, the Destination shall not change the RSEQ value until correctly receiving a micropacket with TSEQ = RSEQ + 1 (the retransmission of the micropacket that was in error). Hence, an error will result in a given RSEQ value being continually sent, and the Source timing out waiting for the expected RSEQ value (i.e., RSEQ > last RSEQ).

The Destination shall use RSEQ = x'FF' after a Link Reset or Initialize operation until it has received the micropacket with TSEQ = x'00'.

10 Signal line encoding

10.1 Signal line bit assignments

The data bytes and control bits shall be transmitted on the signal lines specified in table 3 for a 16-bit wide interface, and as specified in table 4 for an 8-bit wide interface. Nomenclature for the data and control bits is detailed in figure 3. Data signal lines are labelled capital D and a two-digit number, e.g., D00. Control signal lines are labelled capital C and a one-digit number, e.g., C0. The horizontal rows correspond to logical clock ticks. They are grouped in fours, corresponding to the 4b/5b coding.

10.2 CLOCK and CLOCK_2 signals

The CLOCK signal shall be a constant square wave except during a training sequence (see figures 18 and 19). The CLOCK_2 signal shall always be a constant square wave. In an 8-bit system, the square wave shall have a period of 2 ns ± 0,4 ps (i.e., ±0,02 % or 200×10^{-6}). In a 16-bit system, the square wave shall have a period of 4 ns ± 0,8 ps (i.e., ± 0,02 % or 200×10^{-6}).

Table 3 – Signal line bit assignments in a 16-bit system

	Signal lines																		
bit	C3	C2	C1	C0	D 15	D 14	D 13	D 12	D 11	D 10	D 09	D 08	D 07	D 06	D 05	D 04	D 03	D 02	D 01	D 00
a	12	08	04	00	07.4	07.0	06.4	06.0	05.4	05.0	04.4	04.0	03.4	03.0	02.4	02.0	01.4	01.0	00.4	00.0
b	13	09	05	01	07.5	07.1	06.5	06.1	05.5	05.1	04.5	04.1	03.5	03.1	02.5	02.1	01.5	01.1	00.5	00.1
c	14	10	06	02	07.6	07.2	06.6	06.2	05.6	05.2	04.6	04.2	03.6	03.2	02.6	02.2	01.6	01.2	00.6	00.2
d	15	11	07	03	07.7	07.3	06.7	06.3	05.7	05.3	04.7	04.3	03.7	03.3	02.7	02.3	01.7	01.3	00.7	00.3
a	28	24	20	16	15.4	15.0	14.4	14.0	13.4	13.0	12.4	12.0	11.4	11.0	10.4	10.0	09.4	09.0	08.4	08.0
b	29	25	21	17	15.5	15.1	14.5	14.1	13.5	13.1	12.5	12.1	11.5	11.1	10.5	10.1	09.5	09.1	08.5	08.1
c	30	26	22	18	15.6	15.2	14.6	14.2	13.6	13.2	12.6	12.2	11.6	11.2	10.6	10.2	09.6	09.2	08.6	08.2
d	31	27	23	19	15.7	15.3	14.7	14.3	13.7	13.3	12.7	12.3	11.7	11.3	10.7	10.3	09.7	09.3	08.7	08.3
a	44	40	36	32	23.4	23.0	22.4	22.0	21.4	21.0	20.4	20.0	19.4	19.0	18.4	18.0	17.4	17.0	16.4	16.0
b	45	41	37	33	23.5	23.1	22.5	22.1	21.5	21.1	20.5	20.1	19.5	19.1	18.5	18.1	17.5	17.1	16.5	16.1
c	46	42	38	34	23.6	23.2	22.6	22.2	21.6	21.2	20.6	20.2	19.6	19.2	18.6	18.2	17.6	17.2	16.6	16.2
d	47	43	39	35	23.7	23.3	22.7	22.3	21.7	21.3	20.7	20.3	19.7	19.3	18.7	18.3	17.7	17.3	16.7	16.3
a	60	56	52	48	31.4	31.0	30.4	30.0	29.4	29.0	28.4	28.0	27.4	27.0	26.4	26.0	25.4	25.0	24.4	24.0
b	61	57	53	49	31.5	31.1	30.5	30.1	29.5	29.1	28.5	28.1	27.5	27.1	26.5	26.1	25.5	25.1	24.5	24.1
c	62	58	54	50	31.6	31.2	30.6	30.2	29.6	29.2	28.6	28.2	27.6	27.2	26.6	26.2	25.6	25.2	24.6	24.2
d	63	59	55	51	31.7	31.3	30.7	30.3	29.7	29.3	28.7	28.3	27.7	27.3	26.7	26.3	25.7	25.3	24.7	24.3

NOTE 1 The two-digit numbers in the C*n* columns are the control bits, c*nn*.

NOTE 2 The three-digit numbers in the D*nn* columns are the data bits, d*xx.y*, where *xx* is the byte number and *y* is the bit number in the byte.

NOTE 3 The 4-bit groups in a column are transmitted on the associated signal line, top group first, bottom group last.

NOTE 4 The four-bit groups in a column denote 4-bit code groups (dcba) for encoding/decoding to/from the 5-bit transmission codes (zyTxw) specified in table 5. A 5-bit group code (wxTyz) is transmitted over one signal line, e.g., D00.

Table 4 – Signal line bit assignments in an 8-bit system

	Signal lines									
bit	C1	C0	D 07	D 06	D 05	D 04	D 03	D 02	D 01	D 00
a	08	00	07.0	06.0	05.0	04.0	03.0	02.0	01.0	00.0
b	09	01	07.1	06.1	05.1	04.1	03.1	02.1	01.1	00.1
c	10	02	07.2	06.2	05.2	04.2	03.2	02.2	01.2	00.2
d	11	03	07.3	06.3	05.3	04.3	03.3	02.3	01.3	00.3
a	12	04	07.4	06.4	05.4	04.4	03.4	02.4	01.4	00.4
b	13	05	07.5	06.5	05.5	04.5	03.5	02.5	01.5	00.5
c	14	06	07.6	06.6	05.6	04.6	03.6	02.6	01.6	00.6
d	15	07	07.7	06.7	05.7	04.7	03.7	02.7	01.7	00.7
a	24	16	15.0	14.0	13.0	12.0	11.0	10.0	09.0	08.0
b	25	17	15.1	14.1	13.1	12.1	11.1	10.1	09.1	08.1
c	26	18	15.2	14.2	13.2	12.2	11.2	10.2	09.2	08.2
d	27	19	15.3	14.3	13.3	12.3	11.3	10.3	09.3	08.3
a	28	20	15.4	14.4	13.4	12.4	11.4	10.4	09.4	08.4
b	29	21	15.5	14.5	13.5	12.5	11.5	10.5	09.5	08.5
c	30	22	15.6	14.6	13.6	12.6	11.6	10.6	09.6	08.6
d	31	23	15.7	14.7	13.7	12.7	11.7	10.7	09.7	08.7
a	40	32	23.0	22.0	21.0	20.0	19.0	18.0	17.0	16.0
b	41	33	23.1	22.1	21.1	20.1	19.1	18.1	17.1	16.1
c	42	34	23.2	22.2	21.2	20.2	19.2	18.2	17.2	16.2
d	43	35	23.3	22.3	21.3	20.3	19.3	18.3	17.3	16.3
a	44	36	23.4	22.4	21.4	20.4	19.4	18.4	17.4	16.4
b	45	37	23.5	22.5	21.5	20.5	19.5	18.5	17.5	16.5
c	46	38	23.6	22.6	21.6	20.6	19.6	18.6	17.6	16.6
d	47	39	23.7	22.7	21.7	20.7	19.7	18.7	17.7	16.7
a	56	48	31.0	30.0	29.0	28.0	27.0	26.0	25.0	24.0
b	57	49	31.1	30.1	29.1	28.1	27.1	26.1	25.1	24.1
c	58	50	31.2	30.2	29.2	28.2	27.2	26.2	25.2	24.2
d	59	51	31.3	30.3	29.3	28.3	27.3	26.3	25.3	24.3
a	60	52	31.4	30.4	29.4	28.4	27.4	26.4	25.4	24.4
b	61	53	31.5	30.5	29.5	28.5	27.5	26.5	25.5	24.5
c	62	54	31.6	30.6	29.6	28.6	27.6	26.6	25.6	24.6
d	63	55	31.7	30.7	29.7	28.7	27.7	26.7	25.7	24.7

NOTE 1 The two-digit numbers in the C*n* columns are the control bits, c*nn*.

NOTE 2 The three-digit numbers in the D*nn* columns are the data bits, d*xx.y*, where *xx* is the byte number and *y* is the bit number in the byte.

NOTE 3 The 4-bit groups in a column are transmitted on the associated signal line, top group first, bottom group last.

NOTE 4 The four-bit groups in a column denote 4-bit code groups (d.c.ba) for encoding/decoding to/from the 5-bit transmission codes (zyTxw) specified in table 5. A 5-bit code group (wxTyz) is transmitted over one signal line, e.g., D00.

10.3 FRAME signal

The FRAME signal transitions shall be as shown in figures 16 through 19. As shown in figures 18 and 19, the start of a training sequence (40 ns long) shall be signaled by a 10101 FRAME signal pattern in a 16-bit system, and by a 1100110011 FRAME signal pattern in an 8-bit system.

As shown in figures 16 and 17, a 0 to 1 transition on the FRAME signal shall signal the beginning of a micropacket, unless the transition is part of a training sequence. In a micropacket, the FRAME signal shall = 1 for the first half (20 ns), and shall = 0 for the last half (20 ns), of the micropacket.

10.4 Source-side encoding for d.c. balance

The transmitted signals shall be encoded to achieve d.c. balance on each signal line. Table 5 specifies the 5-bit signal line codes (zyTxw) corresponding to the 4-bit input codes (dcba) from tables 3 and 4. For example, on signal line D00, the first dcba 4-bit code consists of bits d00.0, d00.1, d00.2, and d00.3. See clause A.1 for an example of a circuit.

For each signal line, a running count, called the Disparity Count, shall be kept of all the ones and zeros transmitted on that line since the link was reset. The Disparity Count shall be incremented for each one transmitted, and decremented for each zero transmitted.

The 5-bit code to be transmitted shall be based on the current value of the Disparity Count and the input data 4-bit code. The 5-bit code shall be transmitted in the sequence, w,x,T,y,z where T is the true/complement bit. For example, with input data from the right column of tables 3 or 4, if:

a = d00.0 = 1 (least-significant bit)

b = d00.1 = 0

c = d00.2 = 0

d = d00.3 = 0

and Disparity Count = +1 before encoding,

then, based on the third column second row in table 5, transmit on D00:

w = 1 (transmitted first)

x = 0

T = 1

y = 0

z = 0

Disparity Count = 0 after encoding.

NOTE 1 The range for the Disparity Count at the 5-bit boundaries is from +4 to –5. The range for the Disparity Count is from +6 to –7.

NOTE 2 The Disparity Count may also be updated by adding or subtracting the value of Delta Disparity shown in table 5. Add Delta Disparity if Disparity Count <0; subtract if ≥0.

NOTE 3 The 5-bit code is derived by inserting a 1 in the middle of the 4-bit code, and then transmitting either the true or complement value of the resultant 5-bit quantity.

NOTE 4 The maximum run length, i.e., the longest string of continuous 1s or 0s, is 11. The string of 4-bit code points creating the maximum run length is x'EFC'. Start with Disparity Count = +3 or +4 for a string of 11 zeros. Start with Disparity Count = –4 or –5 for a string of 11 ones.

Table 5 – 4b/5b line coding

4-bit code dcba	5-bit code when Disparity <0 zyTxw	5-bit code when Disparity ≥0 zyTxw	Delta Disparity
0000	11011	00100	3
0001	11010	00101	1
0010	11001	00110	1
0011	00111	11000	1
0100	10011	01100	1
0101	01101	10010	1
0110	01110	10001	1
0111	01111	10000	3
1000	01011	10100	1
1001	10101	01010	1
1010	10110	01001	1
1011	10111	01000	3
1100	11100	00011	1
1101	11101	00010	3
1110	11110	00001	3
1111	11111	00000	5

The data and control signal lines shall be synchronized with the CLOCK, CLOCK_2, and FRAME signals as shown in figures 16 and 18. Figures 15 through 18 are read left to right, i.e., events on the left occur before those on the right. In figures 16 and 18, the CLOCK_2 signal is deliberately shown skewed in relation to the CLOCK signal, although in actual implementation it will not necessarily be skewed (see 16.4).

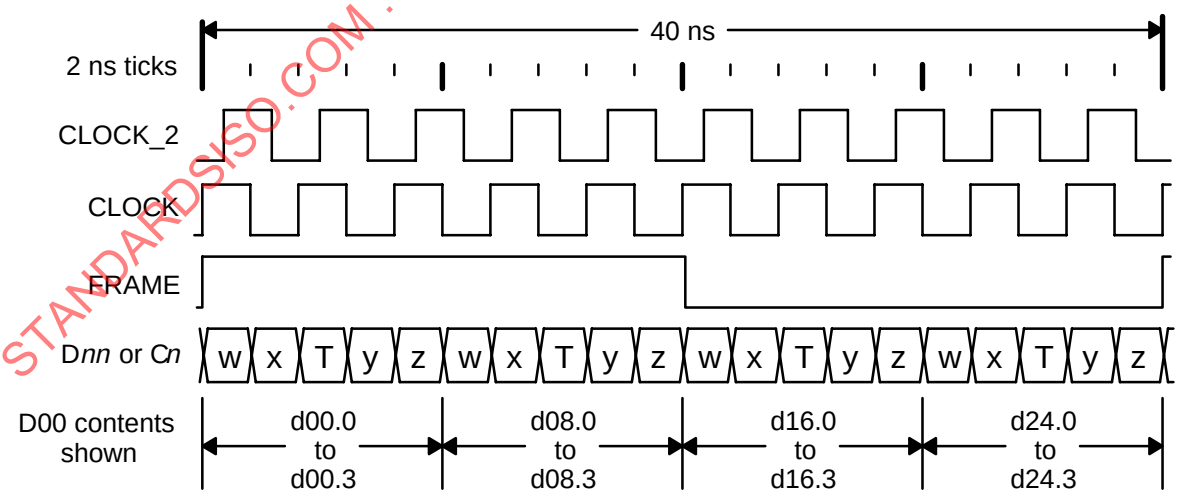


Figure 16 – 16-bit system micropacket

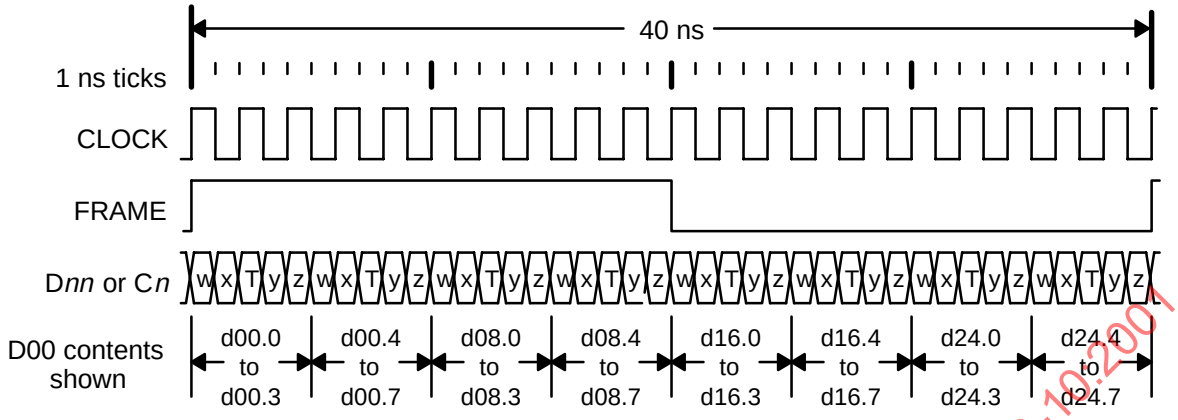


Figure 17 – 8-bit system micropacket

10.5 Destination-side decoding

The received signals shall each be decoded in groups of five bits according to table 5.

NOTE 1 Decoding can be implemented by examining the middle bit of the 5-bit code; if 1 then use the outer bits uncomplemented, if 0 then complement before use.

NOTE 2 There are no illegal 5-bit codes.

11 Skew compensation

11.1 Training sequences

The Destination shall compensate for up to 10 ns of skew among the signals. Skew is defined as the time between the earliest and latest signal arrival at the Destination. Training sequences (see figures 18 and 19) shall be used to measure the skew, and perform dynamic skew adjustments. A signal pattern on either the CLOCK or FRAME signals, as specified in figures 18 and 19, shall be used to identify a training sequence.

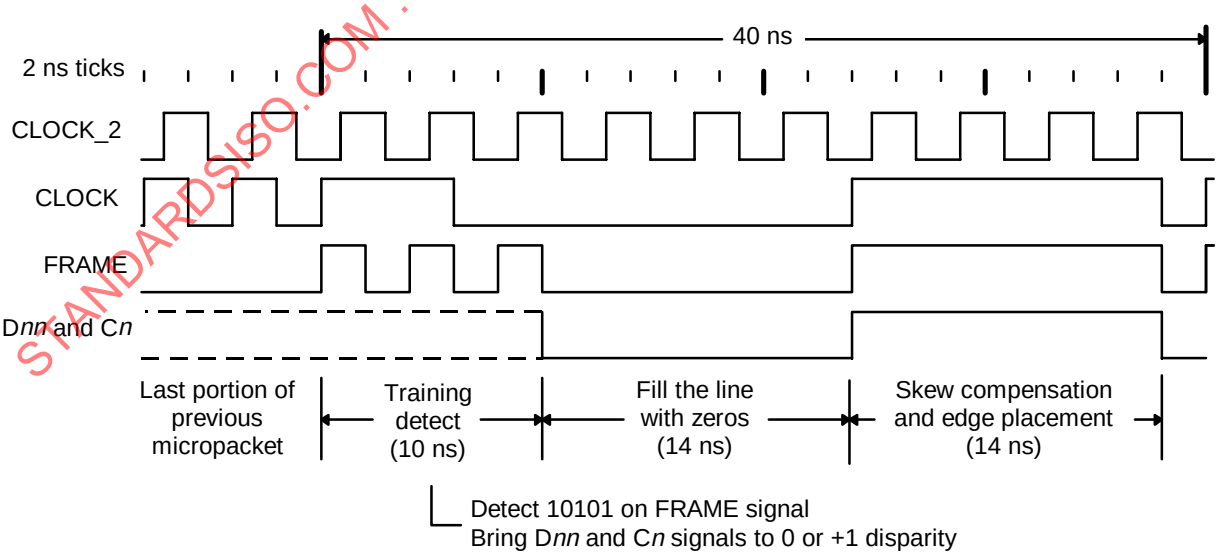


Figure 18 – 16-bit system training sequence

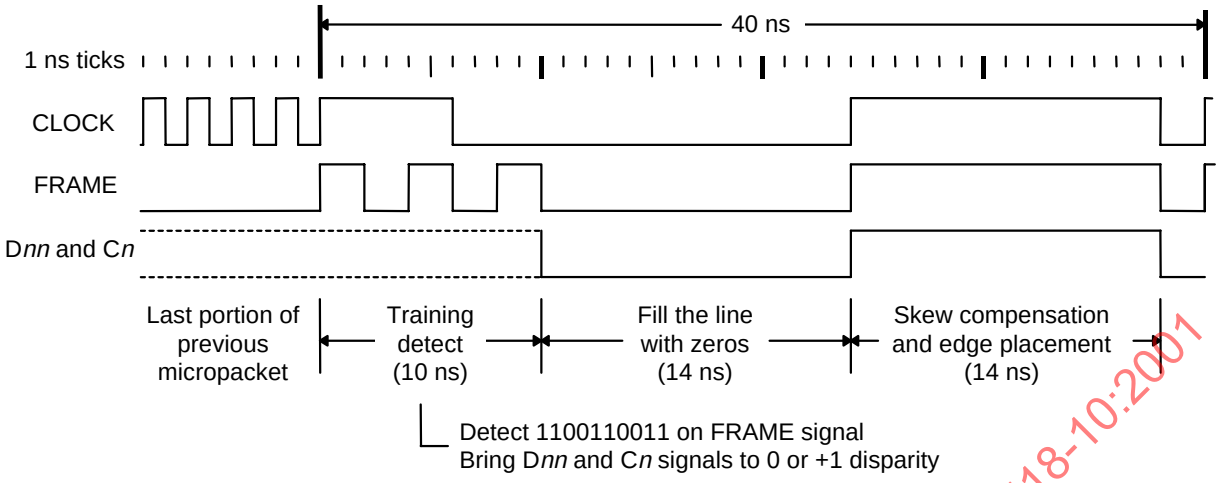


Figure 19 – 8-bit system training sequence

A single training sequence shall be inserted by the Source at least every 10 µs to adjust the dynamic skew, and also to compensate for CLOCK frequency differences between the Source and Destination (see clause A.2). During the first portion of a training sequence, the Source shall insert appropriate Data and Control bits to drive the Disparity Count (see 10.4) on those signal lines to 0 or +1. The Disparity Count shall be set to zero at the end of the training sequence.

11.2 Training sequence errors

If the Destination fails to successfully train and complete its reset/initialize within a Dead-man timeout from any reset/initialize, then the link shall be reset (see 12.2) and a Reset_Initialize_Error shall be logged. The Dead-man default value shall be 100 ms (see 14.1).

If the Destination determines that it has failed to retrain based on the periodic retraining sequences for any contiguous Dead-man period after the link had been healthy, then the link shall be reset (see 12.2) and a Skew_Retraining_Error shall be logged.

12 Link Reset and Initialization

12.1 Overview

Two levels of initialization are specified, Link Reset and Initialize. Link Reset affects the local link only; Initialize may be propagated to other links. Link Reset and Initialize operations are diagrammed in figure 20. A system power-on transition shall trigger, independent of the Hold-off timer (see 12.4), either a Link Reset or Initialize sequence; the choice is implementation and system dependent. Link shutdown (see 13.2) occurs when the link is fatally flawed.

NOTE After commencing normal operation (see figure 20) there can be a short delay until the remote end also begins normal operation. If micropackets other than Null micropackets (i.e., micropackets whose TYPE ≥ x'8') are immediately sent, they may be lost if the remote end is not yet ready to receive them. These lost micropackets will be recovered automatically via retransmission, but this retransmission can typically be avoided if the local end sends a few Null micropackets after commencing normal operation.

12.2 Link Reset

Link Reset affects the local link only, i.e., it is not propagated to other links. When the activity monitor indication = true (see 13.1), Power-on = true, and neither a Link Reset nor an Initialization sequence is currently in progress, then a Link Reset sequence may be triggered by the local administrator, and shall be triggered by:

– the Dead-man timer expiring (see 11.2 and figure 20);

– credit overflow (see 8.1);

– or receiving a micropacket with TYPE = Reset (see 6.3.1).

A Link Reset shall also be triggered by the activity monitor indication going from false to true (see 13.1).

During a Link Reset sequence:

– the receiver shall discard all micropackets except those with TYPE = Reset, Reset_ACK, Initialize, or Initialize_ACK, (i.e., with TYPE = x'2' – x'5');

– the error logging specified in clause 9 shall not occur.

Exit from a Link Reset sequence occurs when a TYPE = Reset_ACK micropacket is received from the other end of the link, indicating that both ends of the link have completed the Reset sequence. At exit

– all of the VC input and output buffers shall be emptied,

– credit for all of the VCs shall be set to zero,

– TSEQ shall be reset to x'00'; RSEQ shall be set to x'FF',

– the dynamic skew compensation circuitry adjusted, and micropackets shall have been received correctly,

– the Disparity Count shall be accurate,

– and the logged events (see table 7) shall have been initialized if the sequence was triggered by a system power-on transition, otherwise the Link Reset shall not modify the logged events.

12.3 Initialize

Initialize sequences may be propagated to other entities. When the activity monitor indication = true (see 13.1) and Power-on = true, an Initialize sequence

– may be triggered by the local administrator,

– shall be triggered by receiving a micropacket with TYPE = Initialize (see 6.3.1), and the Hold-off timer is expired or not running (see 12.4), and not currently doing an Initialize sequence (see figure 20).

During an Initialize sequence:

– the receiver shall discard all micropackets except those with TYPE = Initialize or Initialize_ACK, (i.e., with TYPE = x'4' – x'5');

– the error logging specified in clause 9 shall not occur;

– an Initialize indication shall be passed to the local administrator for possible propagation to other entities.

The normal exit from an Initialize sequence occurs when a TYPE = Initialize_ACK micropacket is received from the other end of the link, indicating that both ends of the link have executed the Initialize sequence. If the link does not complete the Initialize sequence, i.e., receive a TYPE = Initialize_ACK micropacket, within the Dead-man timer period (see 11.2 and table 6), then the Initialize sequence shall be terminated and a Reset sequence started (see figure 20).

NOTE The rationale for terminating an Initialize sequence within a limited time is to prevent stale Initialize sequences from propagating through a HIPPI-6400 network, i.e., Reset affects only the local link while Initialize may be propagated through multiple links.

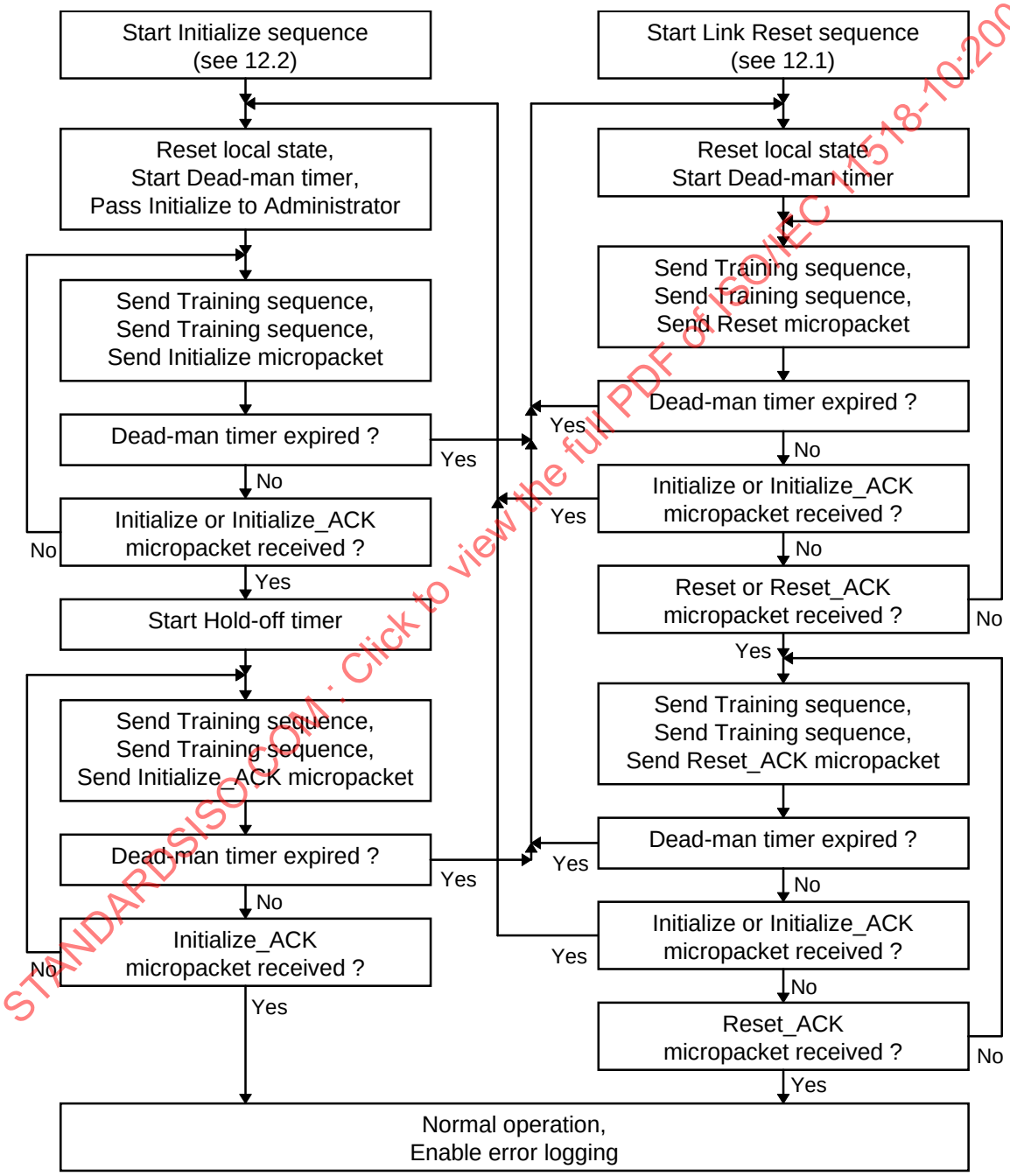


Figure 20 – Initialize and Link Reset sequences

At exit from an Initialize sequence:

– all of the VC input and output buffers shall be emptied;
– credit for all of the VCs shall be set to zero;
– TSEQ shall be reset to x'00'; RSEQ shall be set to x'FF';
– the dynamic skew compensation circuitry adjusted, and micropackets shall have been received correctly;
– the Disparity Count shall be accurate;
– all of the timeout timers (see table 6), except for the Hold-off timer, shall be initialized;
– and the logged events (see table 7) shall have been initialized if the sequence was triggered by a system power-on transition, otherwise the Initialize shall not modify the logged events.

12.4 Hold-off timer

A Hold-off timer shall be used to prevent infinite Initialize oscillations among connected devices. The Hold-off timer shall be started by the first receipt of a TYPE = Initialize, or Initialize_ACK, micropacket. Until expired, the Hold-off timer shall be used to prohibit incoming TYPE = Initialize micropackets from starting an Initialize sequence. The default value for the Hold-off timer shall be 10 seconds (see 14.1).

13 Link activity monitoring and shutdown

13.1 Activity monitoring

The activity monitor shall be used to verify that the interconnecting media is present, and signals are being passed over the link.

The data (D*nn*), control (C*n*), FRAME, and CLOCK signals are affected by the activity monitor indication. When the activity monitor indication = true,

– the outputs of the transmitting circuit(s) shall be as specified in 15.2 or 16.2;
– the circuit(s) receiving the signals shall be enabled.

When the activity monitor indication = false, the transmitting and receiving circuits may be disabled to prevent damage to the interface or personnel (see 15.4 and 16.4).

The activity monitor shall be tolerant of the received signal, riding through minor signal aberrations during the Activity_Monitor timeout. For example, the activity monitor indication shall change after the detected signal has been stable in its new state (i.e., provide hysteresis) for at least the Activity_Monitor timeout period. The default value of the Activity_Monitor timeout shall be 1 ms (see 14.1).

13.2 Link shutdown

A link shutdown shall be triggered by

– the number of times retransmission occurs exceeds some limit (see 8.4),
– the Source has been unable to transmit due to lack of credit (see 8.1),
– a Destination receives micropackets for a VC whose VC Buffer (see figure 5) is full. In this case, a VC[0-3]_RX_VC_Buffer_Overflow error logged shall be logged.

During a link shutdown:

– the local transmitter shall send continuous Null micropackets (with training sequences at appropriate intervals);
– the local transmitter shall de-assert VC flow control to upstream receiver(s), i.e., micropackets destined to go out a port which is shut down are accepted and discarded by that port;
– all of the local VC input and output buffers shall be emptied;
– the receiver shall discard all micropackets except those with TYPE = Reset, Reset_ACK, Initialize, or Initialize_ACK, (i.e., with TYPE = x'2' – x'5');
– the error logging specified in clause 9 shall not occur;
– administrative actions may clear the error counts accessible by Admin operations (see table 7).

A Link Reset sequence (see 12.2) or Initialize sequence (see 12.3) is the exit from a link shutdown. These may be triggered by a local administrator, or by receipt of a TYPE = Reset or Initialize micropacket.

14 Maintenance and control features

14.1 Timeouts

Table 6 contains a summary of the timeouts, their default value, and the location in this part of ISO/IEC 11518 discussing the timeout. All of the timeouts shall be programmable, at least to values of 2X, ½X, and ¼X. The mechanisms and procedures used to set values, different from the default values, are outside the scope of this part of ISO/IEC 11518.

Table 6 – Summary of timeouts

Name	Default value	Reference
ACK indication timeout	12 µs	8.2
Activity_Monitor timeout	1 ms	13.1
Credit timeout	2 s	8.1
Hold-off timer	10 s	12.4
Dead-man timer	100 ms	11.2
Stall timeout	2 ms	9.2.4

14.2 Logged events

Table 7 contains a summary of the events that shall be logged, the minimum number of bits for the parameter, and the location in this part of ISO/IEC 11518 discussing the event. A counter shall not roll over if its maximum value is reached.

Table 7 – Summary of logged events

Name	Minimum Number of bits	Reference
ECRC_Error	8	9.1.3
ECRC_Source_Error	8	6.6.3
LCRC_Error	8	9.1.1
Overrun_Error	1	9.2.5
Reset_Initialize_Error	1	11.2
Retry_Count	8	8.4
Retry_Failure_Error	1	8.4
RSEQ_Missing_Error	8	8.2
RSEQ_Out_Of_Range_Error	8	8.2
Skew_Retraining_Error	1	11.2
TSEQ_Error	8	9.1.2
Undefined_TYPE_Value	4	9.1.4
Underrun_Error	1	9.2.5
VC[1-2]_Admin_Tail_Error	1/2	9.2.1
VC[0-3]_Credit_Overflow_Error	1/4	8.1
VC[0-3]_Credit_Timeout_Error	1/4	8.1
VC[0-3]_Missing_End_of_Message_Error	1/4	9.2.3
VC[0-3]_Missing_Start_of_Message_Error	1/4	9.2.2
VC[0-3]_RX_VC_Buffer_Overflow	1/4	13.2
VC[0-3]_Stall_Timeout_Error	1/4	9.2.4
VC[0-3]_Undefined_TYPE_Error	1/4	9.1.4
NOTE – The 1/4, and 1/2, entries under the Number of bits column mean that there is one bit for an error; e.g., for VC0_Missing_End_of_Message_Error, and a total of four, or two, errors possible (i.e., one for each VC).		

15 Local electrical interface (optional)

15.1 Overview

The local electrical interface is an 8-bit interface (12 signals wide in each direction), intended to connect to on-board optical drivers and receivers. Figure 21 is a block diagram, and figure 22 shows the components used in a signal path. Appropriate RC coupling networks may be needed to match the component voltages, roll-off frequency, and printed circuit board impedances. The alternative to the local electrical interface is the copper cable interface (see clause 16).

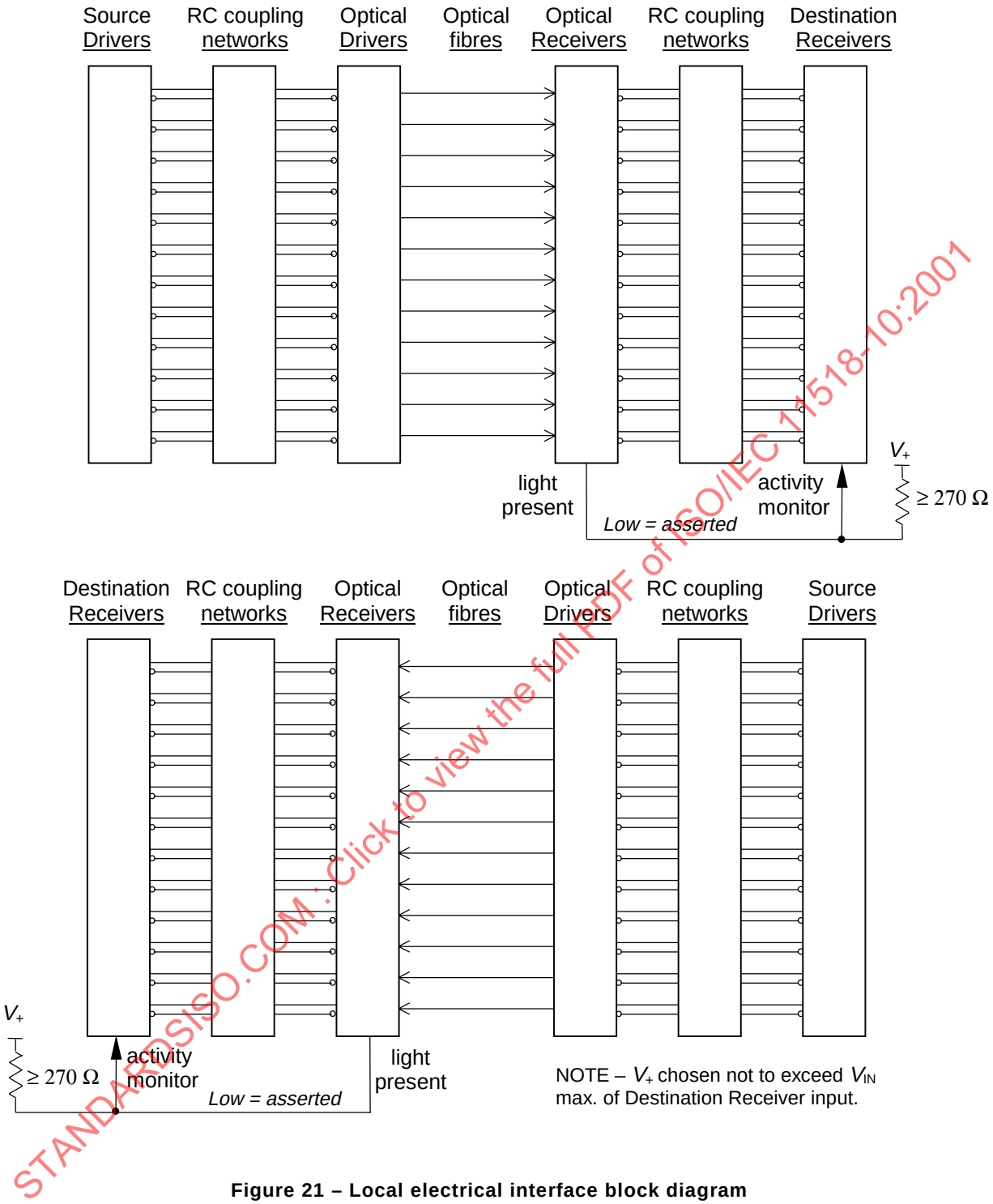


Figure 21 – Local electrical interface block diagram

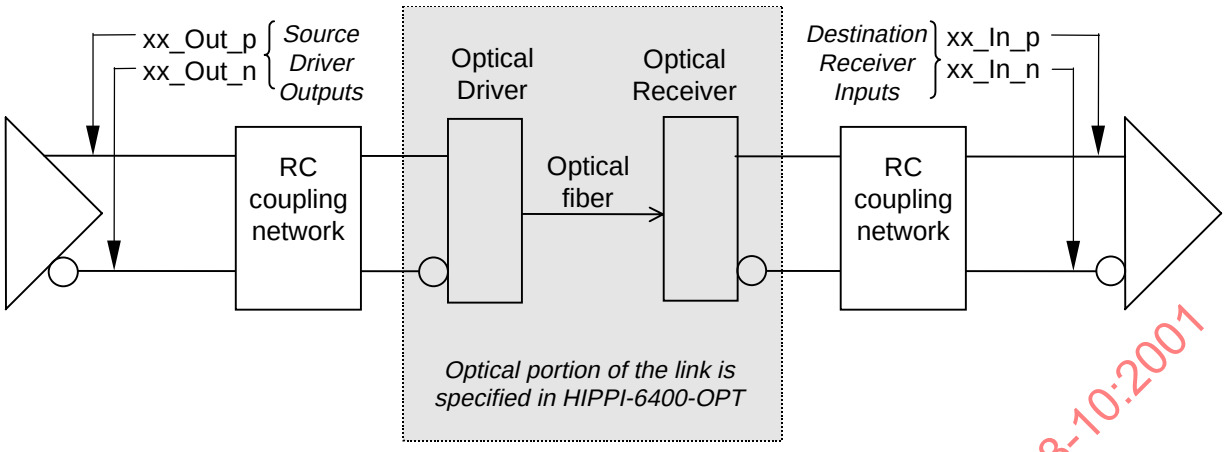


Figure 22 – One signal (of 12 in each direction) of the local electrical interface

15.2 Local electrical interface – Output

The timing for the Source signals shall be as specified in table 8, and shall be measured at the Source driver output pins. During a training sequence, the signals shall be as shown in figure 19.

Differential drivers shall be used on all signal lines, except for the "light-present" signal (see 15.4). Signals in the ‘true’ or ‘1’ state shall have the xx_Out_p pins more positive than the xx_Out_n pins with a peak-to-peak value within the voltage range specified in table 9 for driver output voltage. The corresponding optical signal shall be 'true' or '1' = 'light-on'. Implementation of some differential Source drivers may require d.c. termination for correct operation. Rise and fall times shall be measured at the 20 % and 80 % points of the peak-to-peak signal transition when driving a test load of a 50 Ω transmission line Thevenin terminated to V_{tt}/2 using a 50 Ω equivalent. All parameters shall be measured at the Source driver output pins.

15.3 Local electrical interface – Input

A received 'light-on' optical signal shall indicate 'true' or '1'. The corresponding 'true' or '1' local electrical differential signal shall have the xx_In_p pins more positive than the xx_In_n pins. Implementation of some optical receivers may require d.c. termination for correct operation. Rise and fall times shall be measured at the 20 % and 80 % points of the peak-to-peak signal transition. All parameters are measured at the Destination receiver input pins. The received signals shall be strobed on both edges of the CLOCK signal. See table 10.

15.4 Light present signal

The activity monitor (see 13.1) shall be driven by a 'light-present' signal from the Destination's optical receiver (see figure 21). A no-light condition shall be indicated by a 'high' input in the range of +2,0 V to +2,7 V. The 'light-present' signal is required to sink a maximum current of 10 mA to pull the activity monitor input to a 'low' voltage in the range of +0,8 V to 0,0 V, which shall indicate light present. The resistors shown in figure 21 are used to pull the activity monitor inputs up to the specified 'high' level. A resistance of ≥270 Ω will guarantee that no more than 10 mA are required to pull the activity monitor input from its floating 'high' level to a 'low' level.

Within 1 ms of the activity monitor input going high, the local output signals (with the exception of the CLOCK signal) shall be driven to '0' or 'false', resulting in no light being transmitted on these signals. The local output signals shall remain in this state until the activity monitor input is pulled low. The CLOCK signal shall be continually driven (as specified in 15.2) independent of the activity monitor input level.

NOTE The activity monitor is available for an open fibre control function, shutting off the light for eye-safety reasons unless a complete (nonbroken) optical link is detected. The CLOCK signal, uninterrupted and with a 50 % duty cycle, is intended as the pilot signal. Detection of the CLOCK signal would be an indication that the optical path is complete, and hence the other signals can now be driven.

Table 8 – Local electrical signal timing at Source driver output

Parameter	Value	Units	Comments
CLOCK signal			
Period	2	ns	Nominal, except during a Training sequence
Tolerance	±0,4	ps	±0,02 % or 200 × 10^{-6}
Duty Cycle	50	%	Nominal, except during a Training sequence
Tolerance	±0,8	%	i.e., 49,2 % to 50,8 %
T_{PWD}	32	ps	Peak-to-peak pulse width distortion during a Training sequence
T_{JITTER}	30	ps	Peak-to-peak jitter
DATA and Control signals			
Baud Period	1	ns	
T_{PWD}	32	ps	Peak-to-peak pulse width distortion
T_{JITTER}	32	ps	Peak-to-peak jitter
FRAME signal			
Period	40	ns	Except during a Training sequence
T_{PWD}	32	ps	Peak-to-peak pulse width distortion
T_{JITTER}	32	ps	Peak-to-peak jitter

Table 9 – Local electrical interface, Source driver output

Parameter	Max.	Typical	Min.	Units	Comments
V_O	1 500		500	mVp-p	Driver output voltage swing into test load
T_R and T_F	250	135	90	ps	At 20 % and 80 % points into test load
Signaling rate			1 000	MBd	
Imbalance	40			ps	Within pair skew
Source driver timing					
Channel skew	500			ps	Total pair-to-pair skew
NOTE All measurements, except for T_R and T_F, are single-ended rather than differential.					

Table 10 – Local electrical interface, Destination receiver input

Parameter	Max.	Typical	Min.	Units	Comments
Input signal parameters					
V_{IN}	2 700		200	mVp-p	Input voltage swing
T_R and T_F	400			ps	At 20 % and 80 % points at receiver input
Signaling rate			1 000	MBd	
Channel skew	10			ns	Channel-to-channel skew
Absolute maximum input voltage					
V_{IN}	3 400		–700	mV	Input voltage limits
NOTE – All measurements, except for T_R and T_F, are single-ended rather than differential.					

16 Copper cable interface (optional)

16.1 Overview

The copper cable interface is a 16-bit interface (23 signals wide in each direction), for driving a multi-conductor copper cable for distances up to 40 m. Figure 23 shows the components used in a signal path and table 11 lists the nominal component values. Note that the Z_{0R}, R_0, and R_t values in table 11, and the dimensions in figure 23, are based on tests using FR4 board material; other board material and values may be used with appropriate adjustments determined by simulations and tests. The alternative to the copper cable interface is the local electrical interface (see clause 15).

Table 11 – Copper cable interface

Component	Value	Units	Tolerance %
Z_{0D}	55	Ω	±10
C_{Block}	10 000	pF	±10
R_{0D}	270	Ω	±5
Z_{0R}	75	Ω	±10
R_t	75	Ω	±5

16.2 Copper cable interface – Output

The Source coupling network (i.e., R_0 and C_{Block} in figure 23 and table 11) shall be located on the printed circuit board. The trace from the driver output pin to the receptacle pin shall be ≤100 mm with a characteristic impedance of Z_{0D}. C_{Block} can be located on either side of R_0 and anywhere between the source driver and the connector. For any differential pair, both capacitors shall be on the same side of R_0. Any equalizer components shall be located in the connector backshell and shall be matched to the cable parameters (e.g., length, frequency response, etc.), to obtain a signal meeting the input specifications in 16.3. See table 12.

Differential drivers shall be used on all signal lines. Signals in the 'true' or '1' state shall have the xx_Out_p pin more positive than the xx_Out_n pin with a peak-to-peak value within the voltage range specified in table 13. Rise and fall times shall be measured at the 20 % and 80 % points of the peak-to-peak signal transition when driving a test load of a 75 Ω Thevenin equivalent from each output pin to ground. All Source signal parameters shall be measured at the Source connector, i.e., receptacle, output pins (see figure 23). During a training sequence, the signals shall be as shown in figure 18.

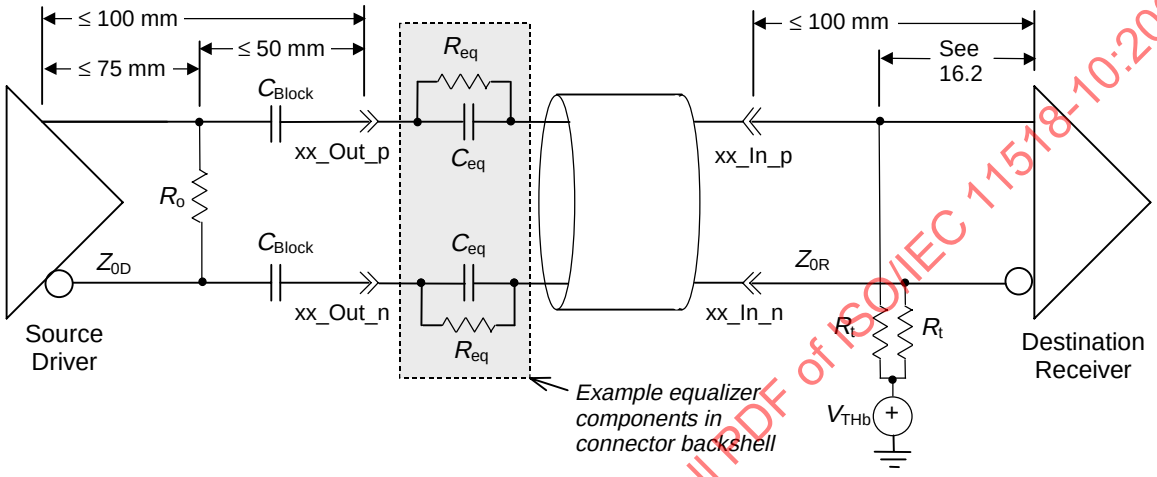


Figure 23 – One signal (of 23 in each direction) of the copper cable interface

16.3 Copper cable interface – Input

The Destination coupling network (i.e., R_t in figure 23 and table 11) shall be located on the printed circuit board and terminated in the appropriate Thevenin equivalent voltage (V_{THb}). The recommended trace from the connector to the Destination receiver is ≤100 mm with a characteristic impedance of Z_{0R}. The maximum propagation time between R_t and the receiver, including the on-chip delay, shall be 350 ps. The 200 ps representative on-chip delay shown in figure 24 shall be used when simulating the Destination Receiver.

Differential receivers shall be used on all signal lines. A received differential signal with the xx_In_p pin more positive than the xx_In_n pin by the minimum signal level specified in table 14 shall indicate a 'true' or '1'. The received signals shall be strobed on both edges of the CLOCK signal. Receivers shall operate correctly when receiving signals meeting the specifications in table 14 and the receiver eye mask in figure 25.

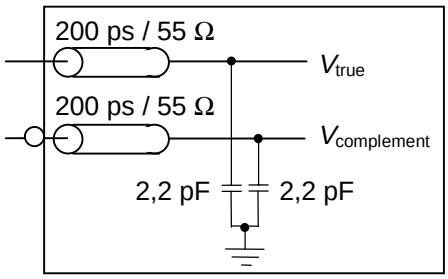


Figure 24 – Destination Receiver equivalent circuit

Rise and fall times shall be measured at the 20 % and 80 % points of the peak-to-peak minimum required signal levels. All parameters are measured at the Destination receiver as simulated by the circuit shown in figure 24.

16.4 CLOCK_2

The phase relationship between the CLOCK_2 and CLOCK signals shall be any constant value. The intended uses of the separate CLOCK and CLOCK_2 signals are:

– to support different skew compensation implementations (e.g., some implementations may prefer to use CLOCK_2, instead of CLOCK, to strobe the signals). The clock signal used to strobe the signals during retraining shall also be used to strobe the signals during normal operation;

– to provide a separate signal that can be monitored for activity (see 13.1) without affecting the signal used for strobing the other signals. If inactivity is detected, the other signals should be ignored to avoid spurious error indications, and an implementation may choose to power down its outputs;

– to provide a free-running clock for systems using phase-locked loops or other implementations that cannot tolerate dropouts of the clock signal.

Table 12 – Copper cable interface signal timing at Source driver output

Parameter	Value	Units	Comments
CLOCK and CLOCK_2 signals			
Period	4	ns	Nominal, except during a Training sequence
Tolerance	±0,8	ps	±0,02 % or 200 × 10^{-6}
Duty Cycle	50	%	Nominal, except during a Training sequence
Tolerance	±0,25	%	i.e., 49,75 % to 50,25 %
CLOCK T_{PWD}	20	ps	Peak-to-peak pulse width distortion during a Training sequence
T_{JITTER}	30	ps	Peak-to-peak jitter
DATA and Control signals			
Baud period	2	ns	
T_{PWD}	50	ps	Peak-to-peak pulse width distortion
T_{JITTER}	40	ps	Peak-to-peak jitter
FRAME signal			
Period	40	ns	
T_{PWD}	50	ps	Peak-to-peak pulse width distortion
T_{JITTER}	40	ps	Peak-to-peak jitter

Table 13 – Copper cable interface, Source driver output

Parameter	Max.	Typical	Min.	Units	Comments
V_0	2 700	2 500	2 200	mVp-p	Driver output voltage into test load
T_R and T_F	250	150	90	ps	At 20 % and 80 % points into test load
Signaling rate			500	MBd	
Imbalance	40			ps	Within pair skew
Source driver timing					
Channel skew	500			ps	Total pair-to-pair skew
NOTE All measurements, except for T_R and T_F, are single-ended rather than differential.					

Table 14 – Copper cable interface, Destination receiver input

Parameter	Max.	Typical	Min.	Units	Comments
Input signal parameters					
V_{IN}	2 700		200	mVp-p	
T_R and T_F	480			ps	At 20 % and 80 % points at receiver input
Signaling rate			500	MBd	
Imbalance	310			ps	Within pair skew
Channel skew	10			ns	Channel-to-channel skew
V_{COM}	±125			mV	Cable induced common mode voltage, not including V_{THb} receiver bias (see figure 23)
Absolute maximum input voltages					
V_{IN}	3 400		–700	mV	Input voltages
NOTE All measurements, except for T_R and T_F, are single-ended rather than differential.					

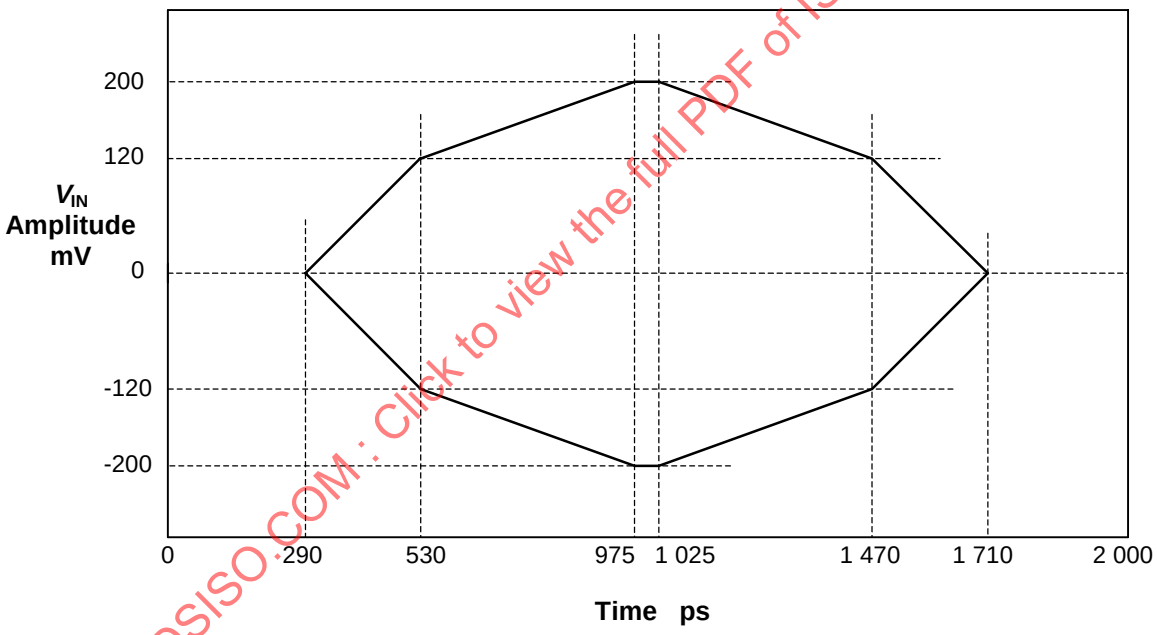


Figure 25 – Receiver eye mask (differential)

NOTE Table 14 specifies the receiver input voltage for a single-ended signal. The receiver eye mask in figure 25 uses differential signals, i.e., twice the magnitude of the single-ended signals.

16.5 Copper cable connectors

The receptacle shall be Berg Micropax 100 position, part number 72546-40x or equivalent (the "x" depends upon the board thickness). A right-angle mount receptacle is shown in figure 28; other mounting methods may be used. The mating cable connector, as shown in figure 29 shall be a Berg Micropax 100 position, part number 72524-001, or equivalent. Figure 29 shows a cable connector with a straight exit; other exit configurations may be used.

NOTE Berg Electronics connectors are examples of suitable products available commercially. This information is given for the convenience of users of this part of ISO/IEC 11518 and does not constitute an endorsement by ISO/IEC, or other publisher of this part of ISO/IEC 11518, of these products.

The receptacle pin assignments shall be as shown in figure 27; pins labelled n.c. shall not be connected. The mating cable connectors shall be wired as shown in table 16.

These connector specifications shall apply for a minimum of 1 000 mating cycles.

Each pin shall have a ≥1 A current capability, with the total current capability for all pins simultaneously shall be ≥5 A.

The connectors shall provide RFI/EMI shielding sufficient to pass all appropriate compliance tests. When mated, the receptacle housing shall provide the ground path for the connector backshell.

Signal attenuation shall be ≤0,1 dB. When multiple pairs are driven differentially with a 100 ps rise time (20 % – 80 %) pulse, near end crosstalk shall be ≤12 %.

Connector thickness shall be ≤19,05 mm (0,75″).

Jackscrews with 4-40 threads shall be used to hold the connectors in the mated position.

16.6 Copper cable specifications

The cable assembly (i.e., cable and connectors) shall provide differential paths for 46 signals, 23 in each direction. All cable assemblies shall meet the specifications in table 15.

The cable assembly length is determined by the cable quality and environmental factors. An equalizer in the cable backshell, as shown in figure 23, may be necessary to achieve the received signal values specified in table 14 and figure 25. The values for the equalizer components are cable dependent.

NOTE Some testing has shown that cables less than 10 m in length work best with no equalizer.

The cable shall have an outside diameter ≤16,9 mm (0,665 in) and a bend radius ≤152 mm (6 in).

The cable shall provide individual shields, or equivalent, for each differential path. These individual shields shall be floating, i.e., isolated from each other, from the overall shield, and from the connector.

There shall be an overall shield. As shown in figure 26, at one end of the cable the overall shield shall be connected to pins 51 and 100 through a total capacitance of 0,4 µF at 50 V. At the other end of the cable, the overall shield shall be directly connected to pins 51 and 100. The overall shield of the cable shall be insulated from the connector backshell at both ends.

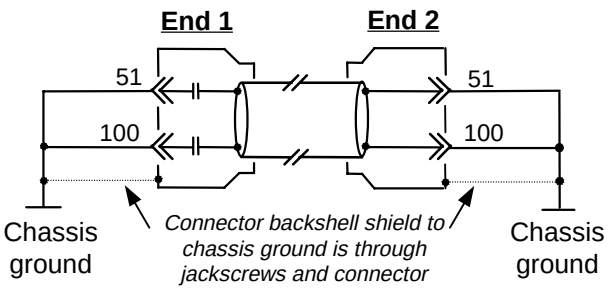


Figure 26 – Connecting the overall shield

Table 15 – Copper cable assembly electrical specifications

Parameter	Max.	Typ	Min.	Units	Comments
Z_0	165	150	135	Ω	Differential impedance
V_{XTALK}	200			MV × ns	Reverse cross talk voltage
V_0			200	mVp-p	Single ended peak-to-peak output voltage
T_{JITTER}	500			ps	Deterministic peak-to-peak jitter
Channel skew	7			ns	Channel-to-channel skew
Imbalance Skew	250			ps	Imbalance skew within a signal pair
NOTE Voltage measurements are single-ended rather than differential.					

Chassis Ground	51	1	CLOCK_2_In_p
D00_In_p	52	2	CLOCK_2_In_n
D00_In_n	53	3	D08_In_p
D01_In_p	54	4	D08_In_n
D01_In_n	55	5	D09_In_p
D02_In_p	56	6	D09_In_n
D02_In_n	57	7	D10_In_p
D03_In_p	58	8	D10_In_n
D03_In_n	59	9	D11_In_p
D04_In_p	60	10	D11_In_n
D04_In_n	61	11	D12_In_p
D05_In_p	62	12	D12_In_n
D05_In_n	63	13	D13_In_p
D06_In_p	64	14	D13_In_n
D06_In_n	65	15	D14_In_p
D07_In_p	66	16	D14_In_n
D07_In_n	67	17	D15_In_p
C0_In_p	68	18	D15_In_n
C0_In_n	69	19	C2_In_p
C1_In_p	70	20	C2_In_n
C1_In_n	71	21	C3_In_p
CLOCK_In_p	72	22	C3_In_n
CLOCK_In_n	73	23	FRAME_In_p
n.c.	74	24	FRAME_In_n
n.c.	75	25	n.c.
n.c.	76	26	n.c.
n.c.	77	27	FRAME_Out_n
CLOCK_Out_n	78	28	FRAME_Out_p
CLOCK_Out_p	79	29	C3_Out_n
C1_Out_n	80	30	C3_Out_p
C1_Out_p	81	31	C2_Out_n
C0_Out_n	82	32	C2_Out_p
C0_Out_p	83	33	D15_Out_n
D07_Out_n	84	34	D15_Out_p
D07_Out_p	85	35	D14_Out_n
D06_Out_n	86	36	D14_Out_p
D06_Out_p	87	37	D13_Out_n
D05_Out_n	88	38	D13_Out_p
D05_Out_p	89	39	D12_Out_n
D04_Out_n	90	40	D12_Out_p
D04_Out_p	91	41	D11_Out_n
D03_Out_n	92	42	D11_Out_p
D03_Out_p	93	43	D10_Out_n
D02_Out_n	94	44	D10_Out_p
D02_Out_p	95	45	D09_Out_n
D01_Out_n	96	46	D09_Out_p
D01_Out_p	97	47	D08_Out_n
D00_Out_n	98	48	D08_Out_p
D00_Out_p	99	49	CLOCK_2_Out_n
Chassis Ground	100	50	CLOCK_2_Out_p

NOTE n.c. = no connection allowed

Figure 27 – Receptacle pin assignments