# IEEE Draft Std P1355
# Standard for Heterogeneous InterConnect (HIC)
# (Low Cost Low Latency Scalable Serial Interconnect for Parallel System Construction)

Date printed 7 October 1993
Version D0.0

Sponsored by the IEEE Computer Society Standards Activities Board
Prepared by the P1355 Working Group

This is an unapproved draft of a proposed IEEE Standard, subject to change. Permission is hereby granted for IEEE Standards Committee participants to reproduce this document for purposes of IEEE standardisation activities. Permission is also granted for member bodies and technical committees of ISO and IEC to reproduce this document for purposes of developing a national position. Other entities seeking permission to reproduce this document for standardisation or other activities, or to reproduce portions of this document for these or other uses, must contact the IEEE Standards Department for the appropriate licence. Use of information contained in this unapproved draft is at your own risk.

IEEE Standards Department
Copyright and Permissions
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

# 1. Introduction

(This introduction is not part of IEEE Std 1355-19XX, Standard for Heterogeneous InterConnect (HIC) (Low Cost Low Latency Scalable Serial Interconnect for Parallel System Construction))

## 1.1. Historical Perspective

The construction of high-performance systems with parallel processing and/or parallel i/o demands a fast, low-cost, low-latency interconnect. It must be fast and low-latency, otherwise it will be the limiting factor in system performance, and it must be low-cost, or else it will dominate the system cost. It must also scale well in both performance and cost relative to the system size, otherwise highly parallel systems will be limited in performance or too expensive. Existing standards do not meet these criteria, either because they are designed for communication over long distances (which incurs high costs), or because they aim at the extreme of currently achievable performance (which again increases costs), or because they are based on a restricted model such as a bus, which limits overall performance and scalability.

This Standard is based on technical developments concentrating on highly integrated, low power interconnect technology implemented in high volume commodity VLSI processes. Aspects of the baseline for this standard have their origins in work on parallel transputer-based systems which has taken place in a number of ESPRIT projects. In particular the routing strategy was established in the PUMA project and DS-Links were developed partially in the GP MIMD project. Work at interconnect for high performance mainframe computers at Bull led to the development of the gigabit link technology implemented in Bi-CMOS and CMOS processes. More recently, these developments, together with corresponding optical technology, have been brought together in the OMI/HIC Project (Open Microprocessor systems Initiative - High Performance Heterogeneous Interconnect)

## 1.2. Acknowledgements

This Standard has been developed with the efforts of many volunteers. The following is a list of those who attended Working Group meetings while the draft and final standard documents were compiled:

Colin Whitby-Strevens, Chair
Roland Marbot, Co-Chair
Ralf Bokämper
Andrew Cofler, Editor
Stein Gjessing
Stefan Haas
Bjørn Johnsbråten
Ernst Kristiansen
Calogero Mantellina
Brian Martin
Paolo Melloni
Mats Olstedt
Joar Martin Østby
Eivind Rongved
Pino Rosario
Peter Thompson

Contributions have also been received from:

Dave Cormie
Alain Greiner
Howard Gurney
Jean-Claude Le Bihan
Reza Nezamzadeh
Sylvain Paineau
Eugene Schramm
Paul Walker
James Wolffe
Belkacem Zerrouk

1

2

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

# Table of Contents

1. Introduction ....2
1.1. Historical Perspective....2
1.2. Acknowledgements....2
2. Overview....6
2.1. Scope....6
2.2. Purpose....6
2.3. References....6
3. Glossary....7
4. Physical Layer, Logical Layer and the Protocol Stack....9
4.1. Physical Layer....9
4.2. Logical Layer - the Protocol Stack....9
4.2.1. Level 0: SIGNAL LEVEL (SL)....9
4.2.1.1. Definition of a Signal....9
4.2.1.2. General Description....9
4.2.2. Level 1: CHARACTER LEVEL (CL)....10
4.2.2.1. Definition of a Character....10
4.2.2.2. General Description....10
4.2.3. Level 2: EXCHANGE LEVEL (EL)....10
4.2.3.1. Definition of the EL....10
4.2.3.2. General Description....10
4.2.4. Level 3: PACKET LEVEL (PL)....11
4.2.4.1. Definition of a Packet....11
4.2.4.2. General Description....11
4.2.4.3. Destination....11
4.2.4.4. Payload....11
4.2.4.5. End of Packet Marker....12
4.2.5. Level 4: TRANSACTION LEVEL (TL)....12
4.2.5.1. Formal Definition....12
4.3. Interaction of Levels....13
4.3.1. 'Downward Going'....13
4.3.1.1. TL to PL....13
4.3.1.2. PL to EL....13
4.3.1.3. EL to CL....13
4.3.1.4. CL to SL....13
4.3.1.5. SL....13
4.3.2. 'Upward Going'....13
4.3.2.1. SL to CL....13
4.3.2.2. CL to EL....13
4.3.2.3. EL to PL....13
4.3.2.4. PL to TL....14
4.3.2.5. Link Status Information....14
4.4. Implementations Defined in the Standard....14
5. DS-Links (DS-SE and DS-DE)....16
5.1. General....16
5.2. DS-SE: Physical Medium....16
5.2.1. Transmission line requirements....16
5.3. DS-DE: Physical Medium....16
5.3.1. Connector....16
5.3.2. Link Cable....24
5.4. DS-SE Signal Level....25
5.4.1. General....25
5.4.2. DS-SE link driver....26
5.4.3. DS-SE link receiver....26
5.4.4. DS-SE (and DS-DE) link timing....27
5.4.5. DS-SE Link Signals....28

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

5.5. DS-DE Signal Level ....29
5.5.1. General....29
5.5.2. DS-DE Signals ....29
5.6. DS-SE and DS-DE Character Level ....31
5.7. DS-SE and DS-DE Exchange Level ....33
5.7.1. General....33
5.7.2. DS-SE and DS-DE Initialisation....33
5.7.3. DS-SE and DS-DE Error Detection ....34
5.7.3.1. General ....34
5.7.3.2. Disconnection errors ....34
5.7.3.3. Parity errors ....34
6. TS-FO Link Optical Interconnect ....35
6.1. TS-FO: Physical Medium....35
6.1.1. TS-FO Link optical connector....35
6.1.2. TS-FO Link Optical Cable....35
6.2. TS-FO Link Optical Signal Level....35
6.3. TS-FO Link Optical Character and Exchange Levels ....37
6.3.1. Symbols....37
6.3.2. Data characters....37
6.3.3. Control characters ....38
6.3.4. Initialisation ....38
6.3.5. Initialisation sequence ....38
6.3.6. Error detection....39
6.3.6.1. Disconnection ....39
6.3.6.2. Packets checked with longitudinal error check code....39
6.3.6.3. Character level checks....39
6.3.6.4. Symbol errors which do not cause loss of synchronisation ....39
6.3.6.5. Bit-level checks and synchronisation....39
6.3.6.6. Recovery of character synchronisation....40
7. HS-SE-10....41
7.1. Copper Link: Physical Medium....41
7.1.1. General Electrical Characteristics....41
7.1.2. Printed Circuit Board (PCB)....41
7.1.3. Single Minicoaxial Cable....41
7.1.3.1. Physical....41
7.1.3.2. Electrical....41
7.1.4. Link Cable....41
7.1.4.1. General ....41
7.1.4.2. Shielding....42
7.1.4.3. Mechanical Performance ....42
7.1.5. Link Connectors ....42
7.1.5.1. General ....42
7.1.5.2. Shielding....43
7.1.6. Coaxial Contacts ....43
7.1.6.1. General ....43
7.1.6.2. Characteristics ....43
7.2. Copper Link: Signal Level ....44
7.2.1. General....44
7.2.2. Line Signal Levels - Driver Side....45
7.2.3. AC Coupling of the link ....46
7.2.4. Receiver Electrical and BER....46
7.3. Character Level (8B/12B Code) ....47
7.3.1. General....47
7.3.2. Transmission Characters ....47
7.3.3. DC balance....47
7.3.4. Control Characters ....48
7.3.5. The Code....48

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Table 7: Data Characters....49
Table 8: Control Characters....51
7.4. Exchange Level....52
7.4.1. General....52
7.4.2. Start-up....53
7.4.2.1. Unidirectional Start-up....53
7.4.2.2. Bidirectional Start-up....53
7.4.3. Errors on Start-up....54
7.4.3.1. Loss of START_REQ....54
7.4.3.2. Destination node not connected or not under power....54
7.4.3.3. Other Errors....54
7.4.4. Functional....54
7.4.4.1. General....54
7.4.4.2. Flow Control....55
7.4.4.3. Maximum transmission line length and Latency....55
7.4.5. Errors when Functional....56
7.4.5.1. Loss of Receiver Calibration....56
7.4.5.2. Parity Error Detected....56
7.4.5.3. Request Reset....56
7.4.6. Shutdown....56
7.4.7. Built-in Loopback Test....56
8. HS-FE-10 Fiber Optic Link....62
8.1. Physical Medium....62
8.2. Signal Level....62
8.2.1. General....62
8.3. Character Level and Exchange Level....62
9. Common Packet Level....63
9.1. General Discussion....63
9.2. Packet Format....63
9.2.1. Destination....63
9.2.2. Payload....63
9.2.3. End_Of_Packet Marker....63
9.3. Routing....64
9.3.1. Routing Function....64
9.3.2. Routing Functionalities....64
9.3.3. Use of simple and complex destinations....64
9.3.3.1. Multiheader....64
9.3.3.1.1. Simple....65
9.3.3.1.2. Complex....65
9.3.3.2. Multicast....65
9.3.3.3. Multicast in a flat network....65
9.3.3.4. Multicast in a hierarchical network....65
9.3.3.5. Broadcast....65
9.4. Error Detection, Recovery and Reporting....66
9.4.1. General....66
9.4.2. Packet Level Check: General Discussion....66
10. Conformance criteria....67
10.1. Definition of subsets....67
10.2. Conformance testing....68
Normative Annexes....69
Informative Annexes....70
Annex A. Rationale....70
Annex B. Switch chips, switches and fabrics....75
Annex C. Use of the Transaction Layer....76
Annex D. DS-Link error handling....77
Annex E. DS-DE Connector PCB recommendation....78
Annex F. DS-DE Cable Recommendation....79

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

# 2. Overview

## 2.1. Scope

The scope of this standard is the physical connectors and cables, electrical properties, and logical protocols for point-to-point serial scalable interconnect, envisaged speeds of 10 - 200 Mbaud, 1 Gbaud and 3 Gbaud, copper and optic technologies (as developed in OMI/HIC).

## 2.2. Purpose

The purpose of this standard is to enable high-performance, scalable, modular, parallel systems to be constructed with low system integration cost; to support communications systems fabric; to provide a transparent implementation of a range of high level protocols (communications (eg ATM), message passing, shared memory transactions, etc), to support links between heterogeneous systems.

Further details and a rationale are given in the Informative Annex A. The Informative Annex B provides a brief overview of how packet routing switch chips may be put together to form switches, and these combined into routing fabrics.

## 2.3. References

ATM
ATM Platform (IEEE P1356, IEEE P1357)
SCI (IEEE 1596-1992)
Draft IEC 1076-4-101

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

# 3. Glossary

[Editor's note: this section is to be completed]

Bidirectional link
Bit
Block
Box
Cable
Character
Character Level (CL)
Connector
Control character
Data character
Destination Node
Destination_identifier
Disparity
End_of_Packet
Exchange Level (EL)

**Fabric:** A general routing capability, constructed from one or more switches using links. *See also:* **link, switch**.

Flat network
Flow_control (FCC)
Functional
Header deletion
Hierarchical network
Level 0
Level 1
Level 2
Level 3
Level 4

**Link:** A means of communicating digital information bi-directionally in serial format between two (VLSI) devices. A link comprises two link interfaces connected by an appropriate medium (or media, for connections between boards or cabinets), such that the link output of each interface is connected to the link input of the other.

Link cable
Link_character (l_char)

**Link input:** A connection point for receiving signals. *See also:* **link interface**.

**Link interface (or Port)**. A connection point comprising a link input and a link output and supporting the protocols defined in this standard. *See also:* **link**.

**Link output:** A connection point for transmitting signals. *See also:***link interface.**

Multi-way
Node

**Node interface:** A link interface on a switch. *See also:* **link interface, switch**.

Normal_character (n_char)
Packet Level (PL)

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Packet switched
Payload
Physical Medium

**Port**. As for Link Interface.

Running Disparity
Signal
Signal Level (SL)
Single-way
Sink
Source
Source Node
Start_request (START_REQ)
Start_acknowledge (START_ACK)
Stop_request (STOP_REQ)
Stop_acknowledge (STOP_ACK)
Stop_negative_acknowledge (STOP_NACK)

**Switch:** A routing capability (for example a box or board) providing a set of numbered node interfaces, constructed from one or more switch chips (or by other methods). *See also:* **fabric, node interface, switch chip**.

**Switch chip:** a VLSI integrated circuit with two or more link interfaces, between which it provides packet routing. *See also:* **link, switch.**

Terminal node
Transaction Level (TL)

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

# 4. Physical Layer, Logical Layer and the Protocol Stack

This standard defines a packet protocol together with the way it shall be implemented on a variety of physical media. It thus comprises a physical layer and a logical layer. The logical layer is defined as a protocol stack. This standard defines representations (information formats) at each level, which are specified in terms of the immediately lower level, together with rules for the exchange of these representations (protocols) at each level.

## 4.1. Physical Layer

Physical characteristics include the specification of [Editor's note: non-exhaustive list]:

a. Electrical links:

1. Circuit board characteristics
2. Connectors and contacts: single and multi-way
3. Cables: twisted pair, coaxial, mini-coaxial
4. Shielding requirements
5. Line impedance
6. Crosstalk
7. Electromagnetic emission
8. ESD susceptibility

b. Optical Links

1. Optical tranceivers
2. Optical connectors: single and multi-way
3. Optical fibre

## 4.2. Logical Layer - the Protocol Stack

### 4.2.1. Level 0: SIGNAL LEVEL (SL)

#### 4.2.1.1. Definition of a Signal

A signal is a physical quantity (e.g. a voltage) which varies in time in order to transmit information. Signals propagate along a wire(s), optic fibres or free space. They are interpreted as sequences of bits. Signals are generated by a *link output* and are absorbed by a *link input*.

#### 4.2.1.2. General Description

The SL includes the specification of [Editor's note: non-exhaustive list]:

a. Electrical links:

1. Line signal levels and rates
2. Power budget: cable length versus transmission distance
3. Noise margins

b. Optical Links

1. Interface electrical - optical - electrical

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

2. Optical power budget

It is important to note that there is a difference between signals and bits. Signals are the values of a physical quantity (e.g. voltage levels) that are on the transmission line, bits are the logical '1's and '0's that the signals represent. There are different voltage levels depending on the implementation (CMOS, ECL, CML, light pulses on a fibre optic cable...).

The signals may have various properties, such as limited running disparity, limited run length, DC-balance over a period of time.

### 4.2.2. Level 1: CHARACTER LEVEL (CL)

#### 4.2.2.1. Definition of a Character

Bits are not transmitted singly, but in groups called *characters*. Characters are the smallest usable (by the upper levels of the protocol) units of information.

#### 4.2.2.2. General Description

The regroupment of the bits is performed by a specific process dependent on the implementation (i.e. serial-to-parallel conversion).

There are various implementations of the point-to-point link that could be implemented, each of which gives different types and a different number (minimum 256) of characters.

### 4.2.3. Level 2: EXCHANGE LEVEL (EL)

#### 4.2.3.1. Definition of the EL

A *link* is the means of communicating digital information bidirectionally in serial format between two (VLSI) devices. A link comprises two *link interfaces* (or *ports*) connected by an appropriate medium (or media, for connections between boards or cabinets), such that the link output of each interface is connected to the link input of the other.

The exchange level describes the procedure of the node-to-node exchange of characters to ensure the proper functioning of the link.

#### 4.2.3.2. General Description

Two types of character are defined: normal character (n_char) and link character (l_char).

a. N_chars:
N_chars can represent, at minimum, the 256 values of a byte plus an end_of_packet marker a*nd cannot be 'consumed' due to limitations of resources*. Limitation of resources refers in general to buffer overwrite. Note that n_chars may however be deleted, added or modified by the PL and/or TL in order to perform the function relative to these levels.

b. L_chars:
L_chars are used locally to the node-to-node link in order to ensure flow control and the proper functioning of the link.

L_chars can be arbitrarily interleaved with n_chars on the link.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Note that in the text of this standard the data values of a byte (0 to 255) will be referred to as *data characters*, and all the other used characters (i.e. all l_chars and the remainder of the n_chars) as *control characters*.

### 4.2.4. Level 3: PACKET LEVEL (PL)

#### 4.2.4.1. Definition of a Packet

A packet is a sequence of characters (n_chars only) with a specific order and format. Constituent characters of different packets cannot be interleaved on a link. A packet consists of a *Destination* followed by a *Payload*. A packet is delimited by an end_of_packet marker.

#### 4.2.4.2. General Description

*Terminal nodes* generate (*source terminal node*) and consume (*destination terminal node*) packets. A *network* is any sequence of devices connected by links joining a source terminal node to a destination terminal node.

The HIC protocol assumes the use of packet-switched networks in which the routing information necessary to correctly transmit the packet across the network is contained in the first K n_chars of the packet (where K is a network-wide parameter). It is at the PL that the routing decisions are taken. The protocol does not define a specific (or maximum) size for a packet. Successive packets transmitted on a link may have different destinations. Each packet is transmitted in its entirety, ie the transmission of a packet on a link shall be completed before the transmission of the successive packet may commence. Packets may arrive at their destination out of their sending order.

#### 4.2.4.3. Destination

The destination contains one or more destination identifiers (dest_id) which are used as the input to the routing function, enabling the routing network to transit the packet from its source-terminal-node to its destination-terminal-node. The destination could be null for specific case of a point-to-point link (i.e. the network is just one link).

Two types of destination are defined:

a. Simple
   The destination is zero or more dest_id(s) in a list. A dest_id is a fixed size field (from 1 to n bytes long), its size being known to the network.

b. Complex
   The destination contains a list of dest_ids, the list having a start_list and a stop_list marker. In practice, these two markers are the same control character (a n_char). Since different types of list are envisioned, the list marker is parametrised: complex_identifier_n (cid_n)

#### 4.2.4.4. Payload

The payload is the data (a message, a memory access request, an acknowledgment etc.) that has to be transferred from the source-terminal-node to the destination-terminal-node. It has a specific format, defined in the Transaction level. Note that a payload may be null.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

#### 4.2.4.5. End of Packet Marker

A packet is delimited by an end_of_packet marker.

### 4.2.5. Level 4: TRANSACTION LEVEL (TL)

#### 4.2.5.1. Formal Definition

A transaction is a sequence of packets sent between two or more terminal nodes to perform some function. At this level there are two concepts:

a. Definition of the specific payload formats necessary to perform the function.

b. Definition of the sequence of exchanges of packets necessary to perform required function.

The definition of any form of transaction is outside the scope of this standard.

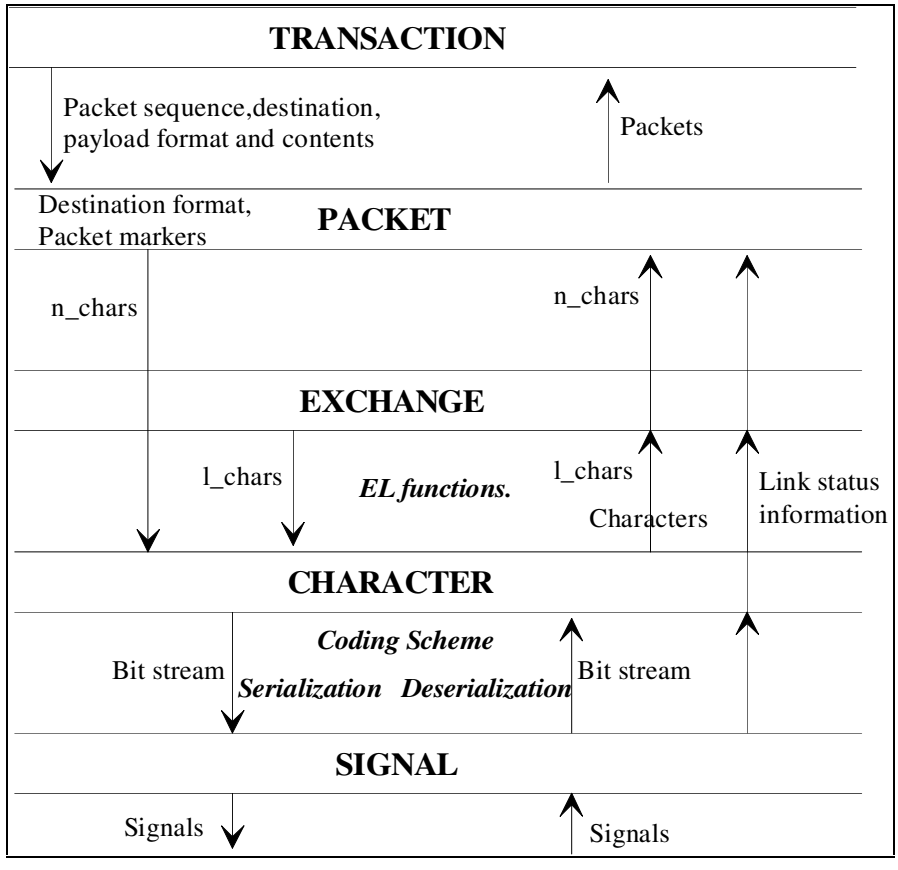


Figure 1: Protocol stack diagram showing interaction of levels

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

## 4.3. Interaction of Levels

### 4.3.1. 'Downward Going'

#### 4.3.1.1. TL to PL

The TL provides packets to the PL. The packet sequence, payload contents and destination address(es) are all determined by the TL. The PL forms the packet destination from the destination address(es) and appends the packet with the appropriate end_of_packet marker.

#### 4.3.1.2. PL to EL

The PL provides the sequence of n_chars that forms each packet to the EL.

#### 4.3.1.3. EL to CL

The EL passes the n_chars that have been provided to it by the PL onto the CL. In order to perform the EL functions, it interleaves l_chars arbitarily into the sequence of n_chars.

#### 4.3.1.4. CL to SL

The CL translates the characters (n_chars and l_chars) it receives into a bit stream that it passes to the SL. The translation is formed of two parts: *coding* and *serialization*. Coding is the translation from the original set of bits to a new set of bits suitable for serial transmission. Serialization is the process of transmitting coded characters one bit at a time.

#### 4.3.1.5. SL

The SL takes the bit stream and outputs signals onto the line. At the receiver end, it receives signals from the line and generates the corresponding bit stream.

### 4.3.2. 'Upward Going'

#### 4.3.2.1. SL to CL

The bit stream is *deserialized* and *decoded* to produce a sequence of characters. Deserialization is the assembly of a coded character from the sequence of serial bits. Decoding is the translation from the coded character to the original character.

#### 4.3.2.2. CL to EL

The characters are passed to the EL. The EL filters all the l_chars for use in the EL functions.

#### 4.3.2.3. EL to PL

The EL passes the n_chars to the PL.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

#### 4.3.2.4. PL to TL

The PL reconstitutes the packets from the received n_chars. Routing functions are carried out, based on the information in the packet destination. When necessary (e.g. when a packet has arrived at its destination node), the packets are delivered to the TL.

#### 4.3.2.5. Link Status Information

Link status information is passed up between the levels from the SL to the PL. The link status information necessary is dependent on the implementation of the link and includes, for example, information on the calibration of the receiver, parity errors etc. It is for each level to filter the information that is relevant to its functioning and to pass-up the remaining information.

## 4.4. Implementations Defined in the Standard

The physical and logical layers defined in this standard are illustrated in figure 2.

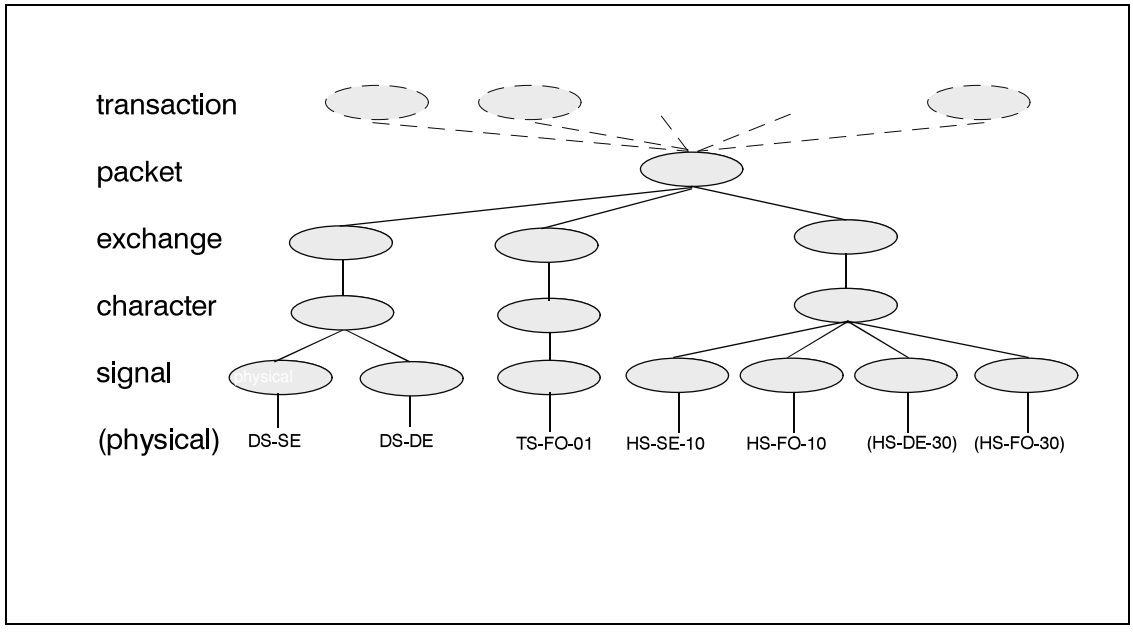


Figure 2: Defined Implementations of Physical and Logical layers

The physical layer may be implemented in a number of different technologies, identified according to signalling convention, transmission medium, and operating speed.

The format for this identification is:

SS-TT-DD

given by the following table:

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

| | | |
|---|---|---|
| SS | DS | Data/Strobe encoding |
| | TS | Three-of-six encoding |
| | HS | 'High Speed' encoding |
| TT | SE | Single ended Electrical transmission |
| | DE | Differential Electrical transmission |
| | FO | Fibre Optic transmission |
| DD | two digits, representing operating speed in units of 100 Mbaud (not applicable to DS links) | |

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

# 5. DS-Links (DS-SE and DS-DE)

## 5.1. General

The DS-Link is designed for point-to-point communication which may be on a single PCB, board to board (DS-SE links) or box to box (DS-DE links). Since this implies that transmission line problems might be present, the electrical level is specified as a transmission line system. In order to reduce the power required for each link (enabling the use of many links) source only termination is used.

A DS-Link input is self clocking, which means that in any particular system the actual operating speed may be chosen to suit the system, rather than the more normal case of having to engineer a system to suit a pre-determined transmission speed. The maximum speed at which it may run is a function of the maximum speed of the circuitry at each end, plus the properties of the medium connecting the two ends. A link may be operated at lower speeds, for example to provide power saving, down to a level at which the disconnection detection mechanism is triggered.

A link shall operate at a speed of 10 Mbaud unless and until the system in which it operates determines that it may operate at a faster speed. The electrical specification of the link has been designed to allow operating speeds of up to 200 Mbaud, but the actual operating speed is left open to allow the design of any particular system to trade off engineering costs against operating speed. The means of setting the speed of a link is not defined within this standard.

## 5.2. DS-SE: Physical Medium

### 5.2.1. Transmission line requirements

For distances of greater than 20cm (at 100 Mbaud), TBS cm (at 200 Mbaud) the link line must be considered as a transmission line. Discontinuities or variations in characteristic impedance should be kept to a minimum. The transmission line may be made on pcbs but care must be taken to provide a good ground or power plane beneath the link track and crosstalk should be minimized with other tracks (including between data and strobe lines of the same link). This can be helped by placing grounded tracks either side of the link track. The longest length of line achievable will depend on the materials used for interconnect and the grounding arrangements.

## 5.3. DS-DE: Physical Medium

### 5.3.1. Connector

The connector system consists of a cable-mounted male plug; and a PCB-mounted female socket.

It provides the following features:

- A single link connection per connector.
- Option for multiple connections on the cable connector
- Complete screening.
- Robust.
- Latching.
- Small size.

The connectors shall be implemented according to the Table 1, and be of dimensions specified in figures .

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Table 1: DS-DE Connector Requirements

| Characteristic | Specification | Test specification reference |
|---|---|---|
| | | |
| Contact grid | 2mm x 2mm | |
| Connector width | 6mm max | |
| Contacts | 10 per connector, arranged as two rows of 5 contacts | |
| Shielding | completely shielded | |
| Modularity | Adjacent connector modules shall be placed on a 6mm pitch. | |
| Polarisation | chamfer design prevents 180° incorrect mating | |
| Male-female contact overlap | 0.5mm min when latched | Draft IEC 1076-4-101 |
| misalignment in transverse and longitudinal axes | guiding device must accept a misalignment ± 0.2mm | Draft IEC 1076-4-101 |
| Inclination of transverse A. longitudinal axes | guiding device must accept an initial angular misalignment of ± 7° max | |
| Rated voltage | 500V contact/contact | Draft IEC 1076-4-101 |
| Voltage proof V R.M.S. | 750V contact/contact<br>750V contact/shielding | Draft IEC 1076-4-101<br>Draft IEC 1076-4-101 |
| Current carrying capacity | 1A at 70°C | Draft IEC 1076-4-101 |
| Contact resistance | 30 mOhms max | |
| Attenuation | | |
| Contact/contact isolation resistance | $10^4$ MOhms min | Draft IEC 1076-4-101 |
| Contact/shielding isolation resistance | $10^4$ MOhms min | Draft IEC 1076-4-101 |
| Crosstalk | | |
| Contact/contact impedance | | |
| Contact/shielding impedance | | |
| Impedance of environment | 90 ± 9 Ohms (differential impedance) | |
| Electromagnetic Radiation Compatibility | TBS | |
| Category temperature | -55°C to +105°C | |
| Contact/contact creepage distance | Male connector 0.7 mm min<br>Female connector 1.1 mm min | |
| Contact/shielding creepage distance | Male connector 0.65 mm min<br>Female connector 1.2 mm min | |
| Contact/contact clearance distance | Male connector 0.7 mm min<br>Female connector 1.1 mm min | |
| Contact/shielding clearance distance | Male connector 0.65 mm min<br>Female connector 1.0 mm min | |
| Mechanical operation | 250 operations | Draft IEC 1076-4-101 |
| Total engaging force | Connector 10N max per connector module | |

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Table (contd).

| | | |
|---|---|---|
| Withdrawal force by pulling on latch | 2N min per connector module | Draft IEC 1076-4-101 |
| | (TBS) N max per connector module | Draft IEC 1076-4-101 |
| Withdrawal force with closed latch | (TBS) N min per connector module | Draft IEC 1076-4-101 |
| | (TBS) N max per connector module | Draft IEC 1076-4-101 |
| Tensile strenght of cable/connector clamp | Greater than the maximum latched withdrawal force | |
| Vibration | | |
| Shock | | |

The relevant buffer shall not be placed more than 3 cm from the corresponding female (PC board mounted) connector unless the tracks between the buffer and the connector are matched to 100 ohms.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

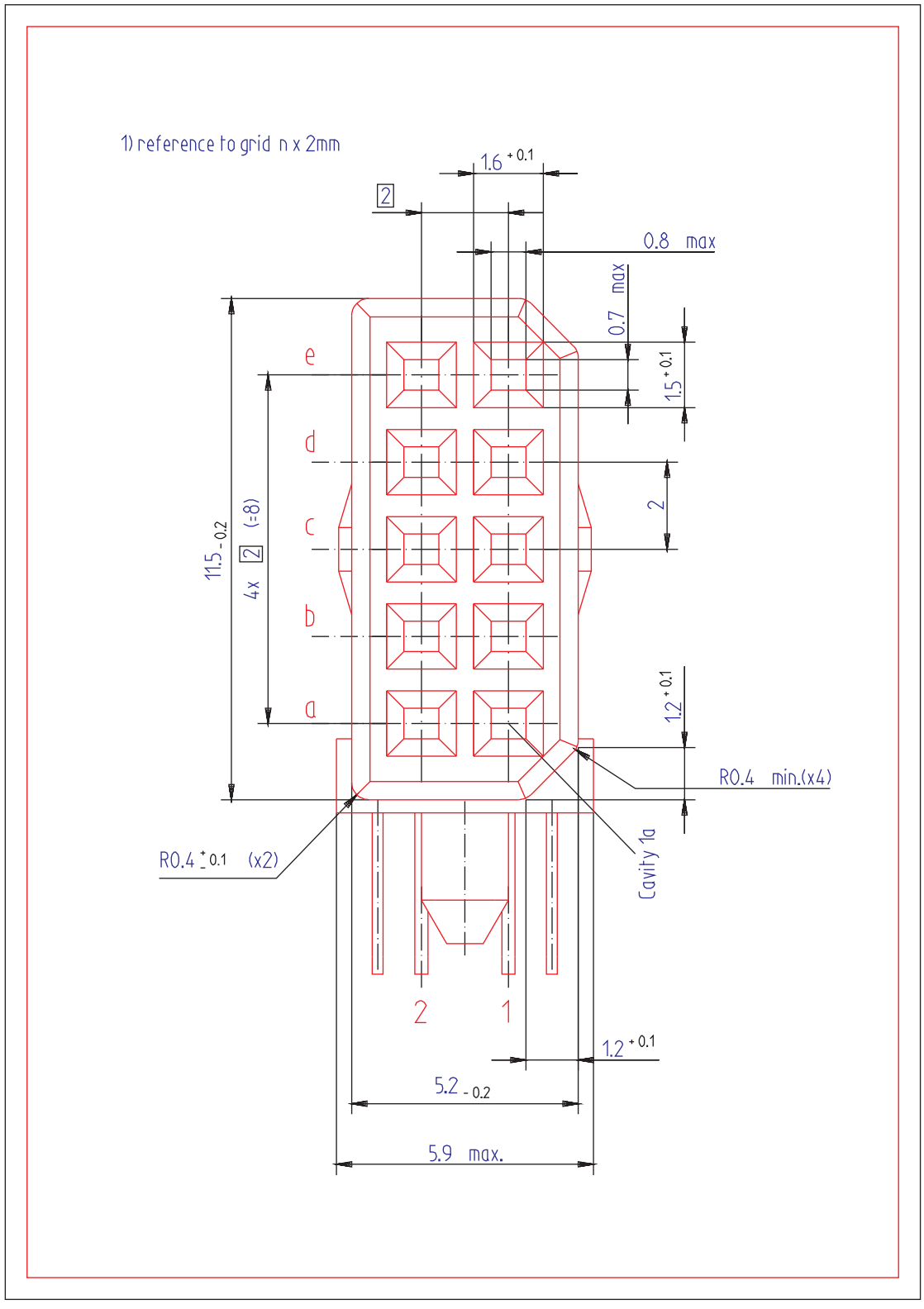


Figure 3: DS Link female connector front view

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

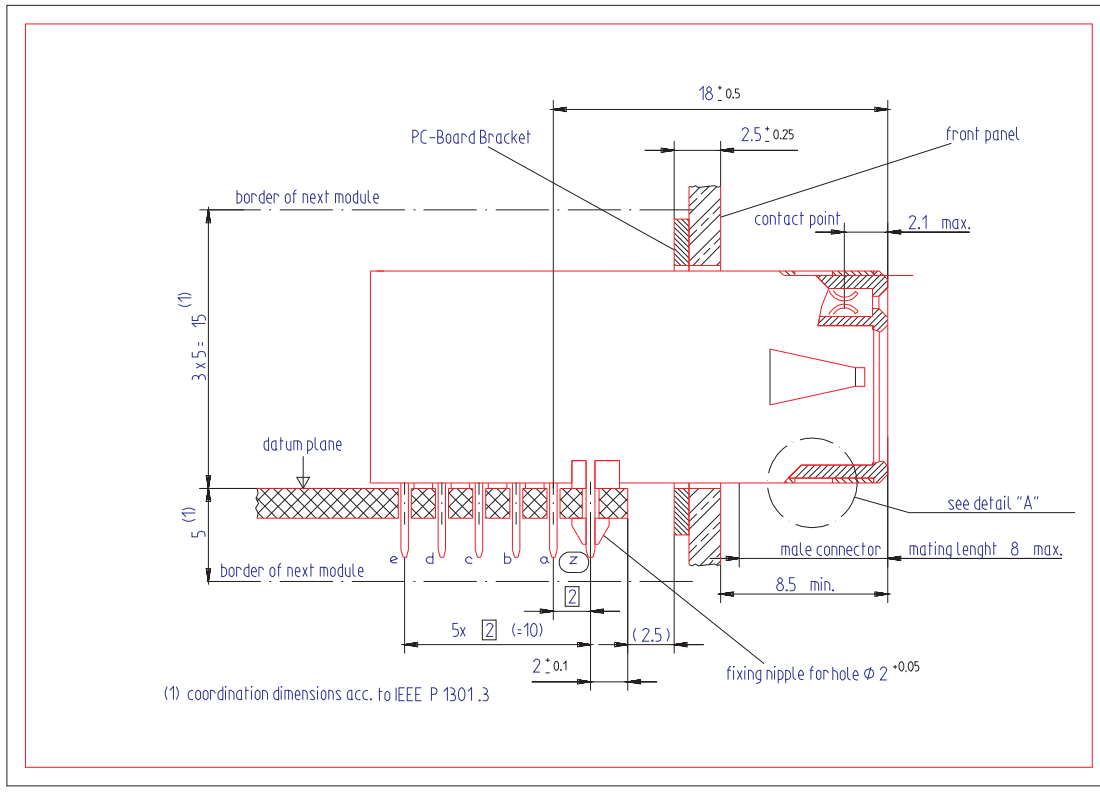


Figure 4: DS Link female connector side view

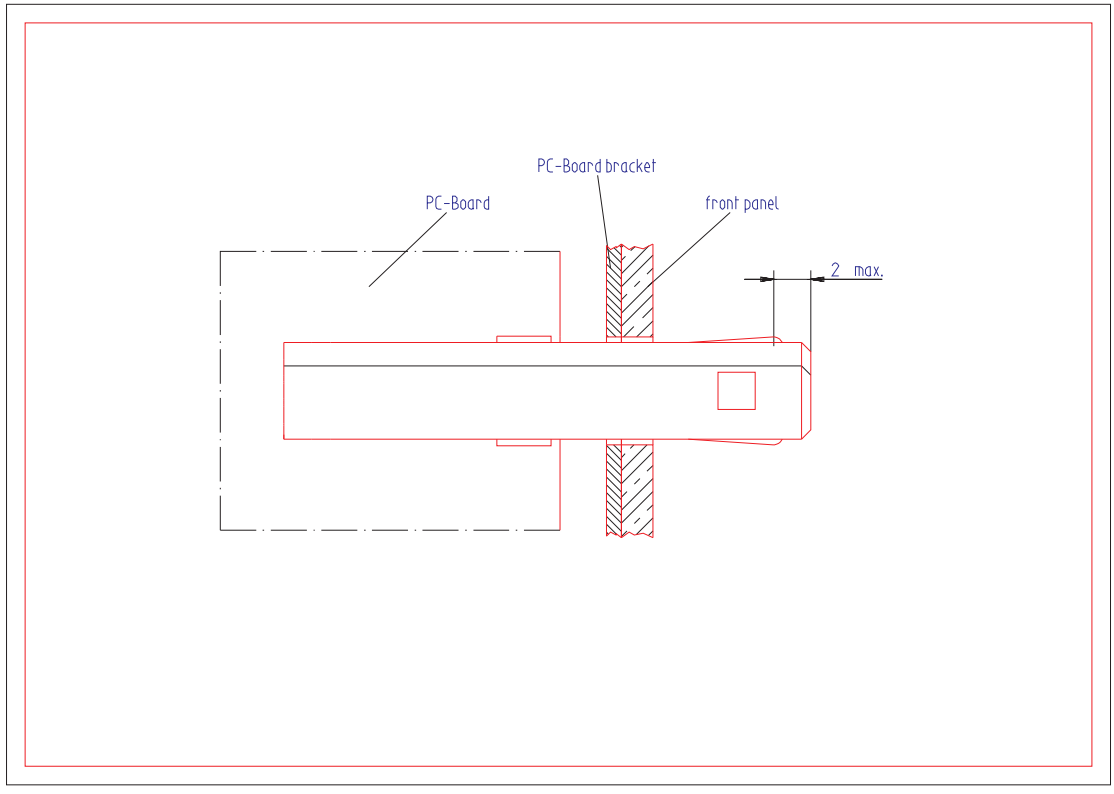


Figure 5: DS Link female connector top view

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

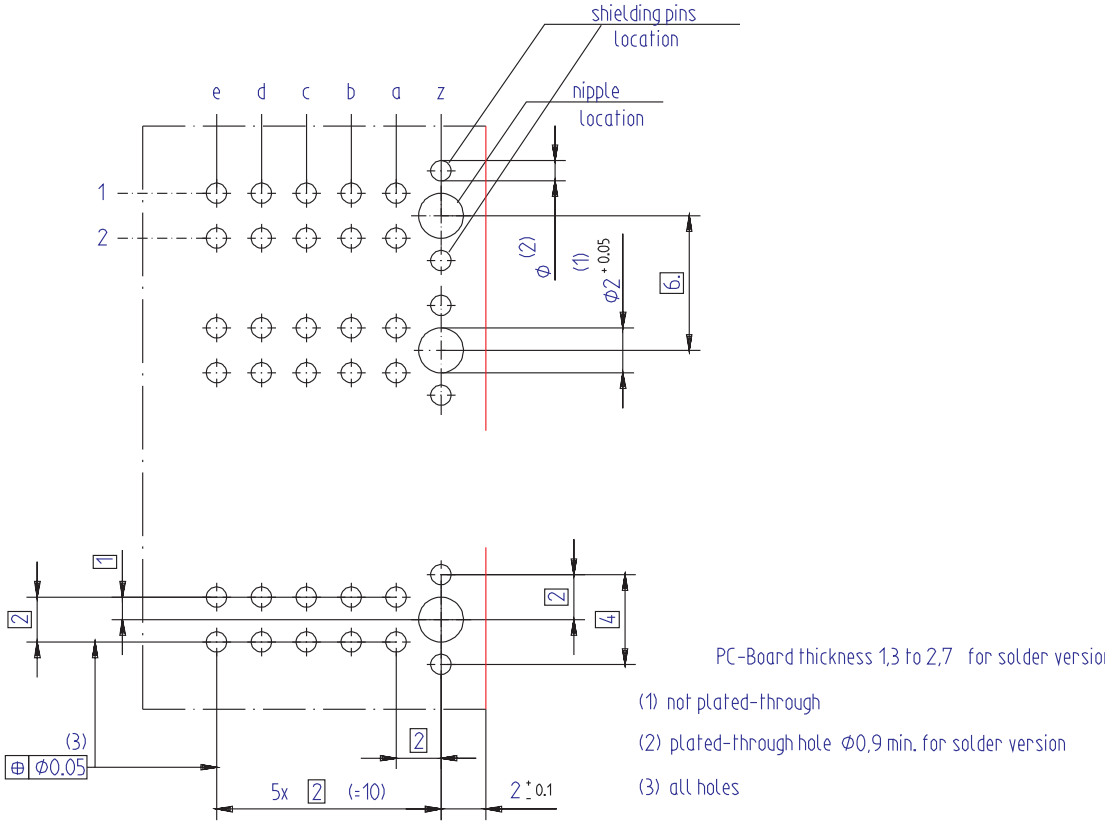


Figure 6: DS Link female connector PCB layout

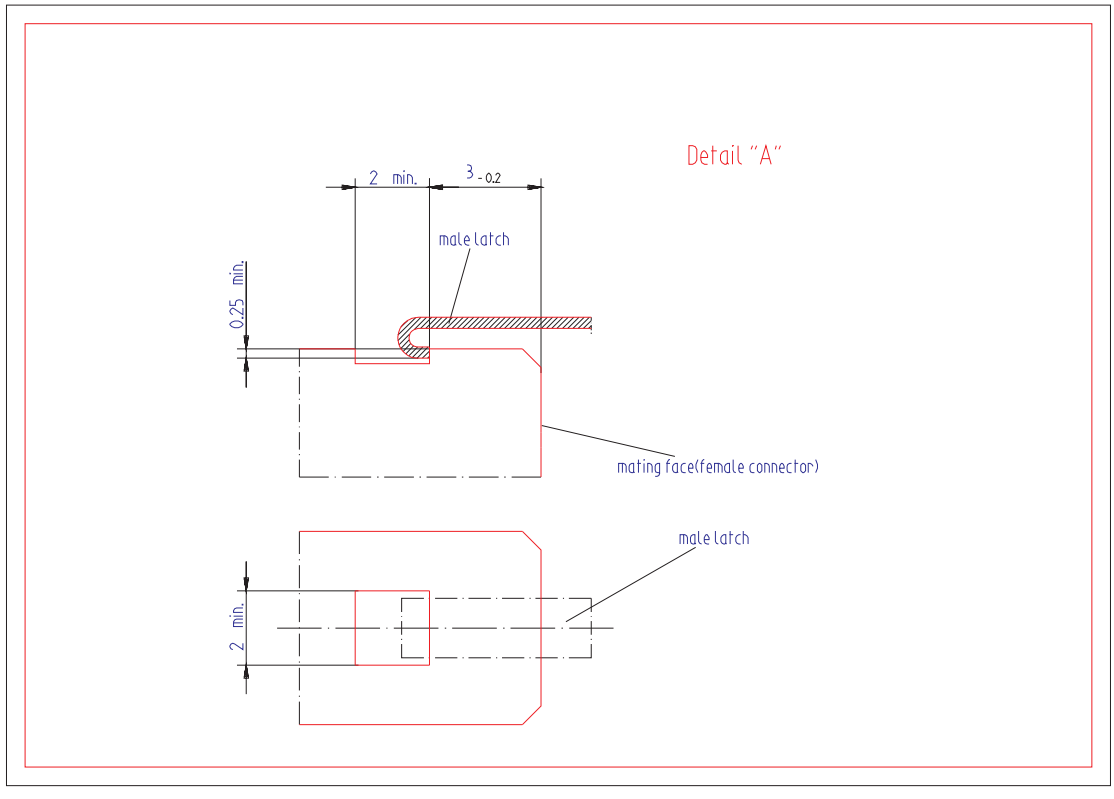


Figure 7: DS Link connector latch

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

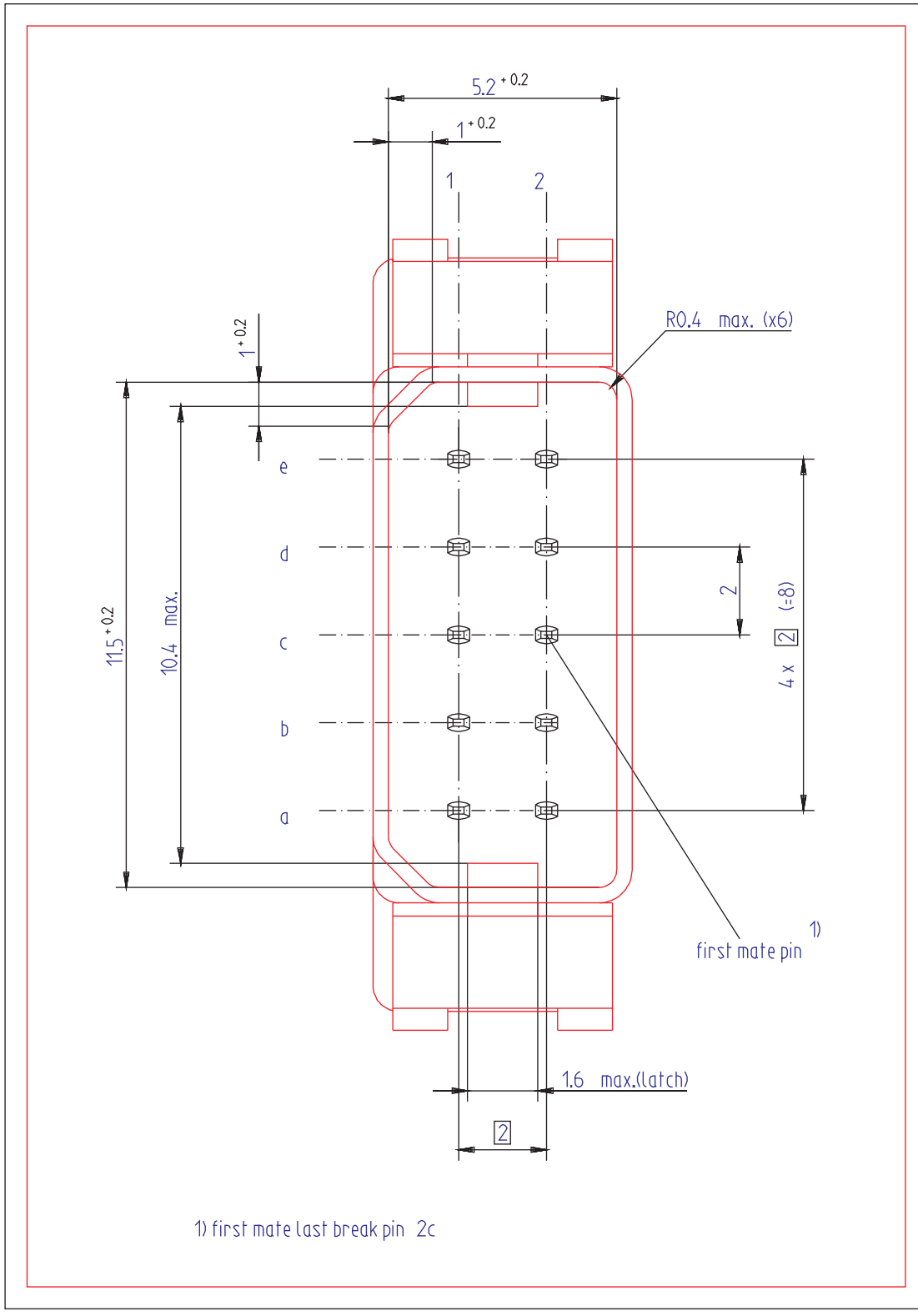


Figure 8: DS Link male connector front view

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

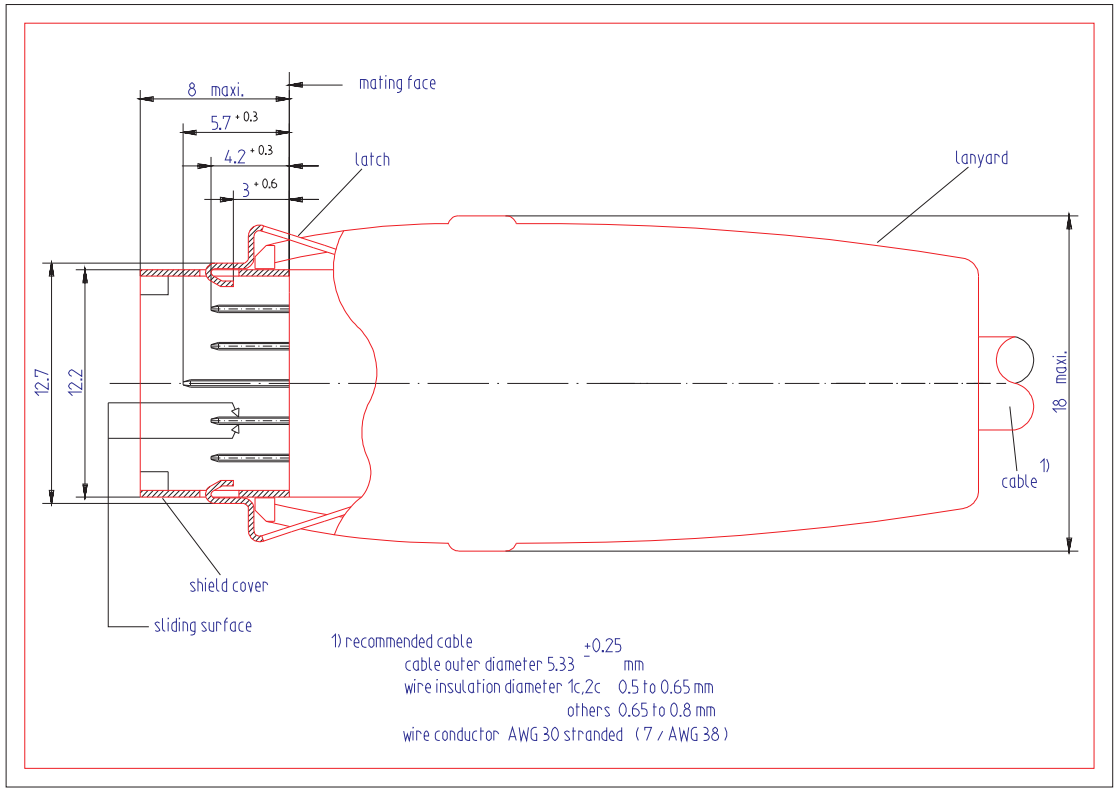


Figure 9: DS Link male connector side view

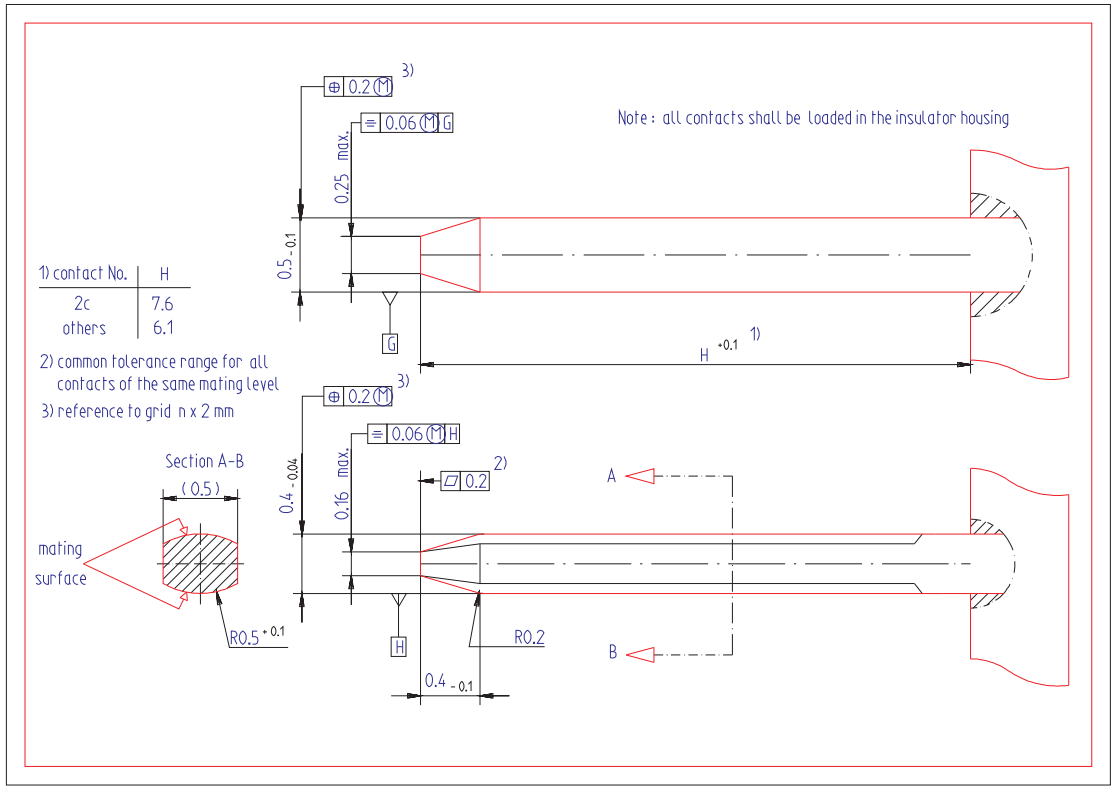


Figure 10: DS Link male connector contact

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

The allocation of signals to the connector pins is given in the following table (front view of the female connector, ie as seen looking at the edge connector mounted on a horizontal pcb).

| | 2 | 1 |
|---|---|---|
| | | |
| **e** | DS_DE_data_in_plus | DS_DE_data_in_minus |
| **d** | DS_DE_strobe_in_plus | DS_DE_strobe_in_minus |
| **c** | *No connect* | *No connect* |
| **b** | DS_DE_strobe_out_minus | DS_DE_strobe_out_plus |
| **a** | DS_DE_data_out_minus | DS_DE_data_out_plus |

### 5.3.2. Link Cable

The cable to be used to connect two DS-DE connectors shall have the following characteristics:-

*Electrical characteristics*

| | |
|---|---|
| Differential impedance | 95 +/- 10 Ohms |
| Time delay skew | 0.15ns/m maximum |
| Conductor DC resistance | 0.35 Ohms/m nominal |
| Attenuation (differential) | 0.45 dB/m maximum @ 100 MHz |
| Electromagnetic Radiation (C?) | |
| Near end crosstalk (backward) | 2% maximum @ 100 MHz measured on 10 metre sample in differential mode |
| Far end crosstalk (forward) | 4% maximum @ 100 MHz measured on 10 metre sample in differential mode |

*Mechanical performance*

| | |
|---|---|
| Flexibility | A 60mm maximum radius shall result when 610mm of cable is loaded with a suspended weight of 100 grams. See figure nnn for details |
| Flex life | 1000 cycles through 180 degrees @ 60mm bend radius and 100 gram load. Following this test, the insulation and jacket should be free of cracks and the conductors free of opens and shorts |
| Minimum bending radius (static operation) | 40mm |

*Safety certification*

UL listing: AWM style 20276, 80C, 30 volts, VW1

The cable shall connect each pin of the connector at one end to a pins in the connector at the other end as defined in the following table:

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

| Pin | Pin |
|---|---|
| | |
| 1a | 2e |
| 2a | 1e |
| 1b | 2d |
| 2b | 1d |
| 1c | 1c |
| 2c | 2c |
| 1d | 2b |
| 2d | 1b |
| 1e | 2a |
| 2e | 1a |

## 5.4. DS-SE Signal Level

### 5.4.1. General

Each connection of a DS-SE link shall be implemented as a uni-directional point-to-point transmission line. For specification purposes, each electrical connection is modelled as a driver, a transmission line and a receiver.

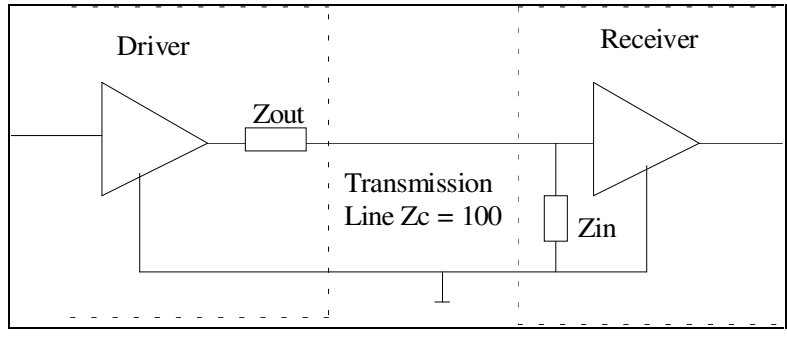


Figure 11: DS-SE Link Electrical Model

The transmission line shall have a characteristic impedance of 100 Ohms, and shall be terminated by high impedance at the receiver. The transmission line may thus be provided by PCB, coax or other suitable controlled impedance interconnect. The receiver may be designed as a standard TTL input with no internal termination. The driver output impedance should be close to 100 Ohms: both when fully on, in either condition, and also while switching. This is to ensure that reflections generated by the receiver are sufficiently damped by the driver that they do not cause spurious transitions.

The range 0.8V - 2.0V at the receiver is defined as the *transition region*. It is very important that the signal at the receiver crosses the transition region cleanly, and that the reflections do not cause spurious transitions. This may be ensured by careful design of the interconnect and the driver output.

The propagation of signals is as follows (see figure 12). Assuming that the line is in a steady-state low condition, a high going transition is output by the driver. Because of the relatively high output impedance of the driver compared to the line, this is attenuated, typically to a level of about 1.4V at the driver output. At the receiver, the high input impedance of the receiver (effectively an open circuit) causes this to appear as a 2.8V step (taking the input cleanly through the transition region), and the reflection travels back down the line to the driver. Depending on how closely the driver output impedance is matched to the line, this may cause further reflections: it is crucial that the driver is

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

matched sufficiently well that these reflections do not cause spurious transitions at the receiver, or interfere with subsequent output transitions. Propagation for a low-going transition is similar.

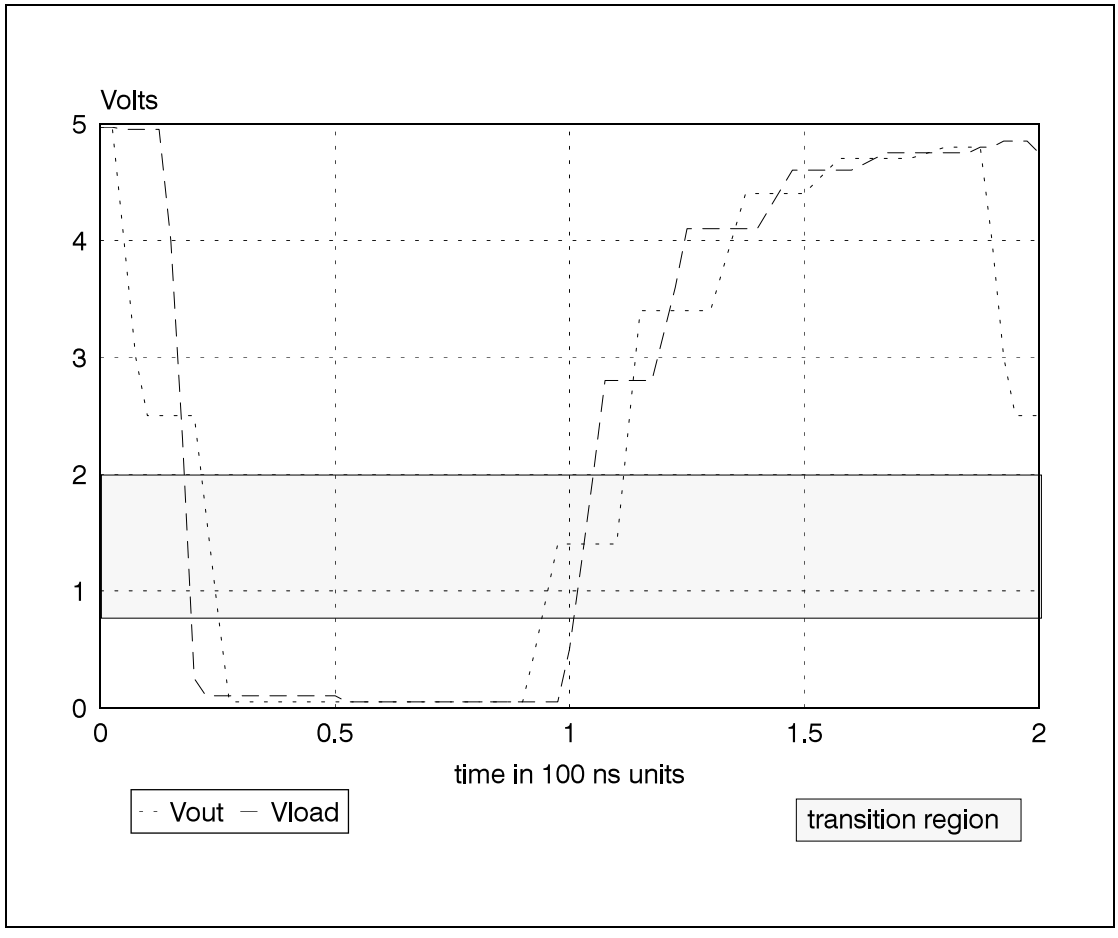


Figure 12: DS-SE Link Signal Propagation

### 5.4.2. DS-SE link driver

Each driver should have a nominal output impedance of 100 Ohms to terminate the reflections from the receiver. The output impedance must be controlled when the output is fully on driving a high or low level, and also while switching.

The output impedance of the driver may vary as follows:

| | min | max |
|---|---|---|
| | | |
| Rol | 60 | 110 |
| Roh | 100 | 250 |

Informative Annex * gives an example DS-SE link driver circuit.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

### 5.4.3. DS-SE link receiver

A DS-SE link input must present a high impedance termination to the transmission line. An impedance of >10K Ohms is recommended. This high input impedance is essential to ensure that the correct signal levels are produced at the receiver input.

The input threshold of the receiver is nominally 1.4V and shall be within the transition region, which is defined at 0.8V - 2.0 V.

The receiver shall be tolerant of transient pulses outside the normal signal levels, as defined by (TBS).

### 5.4.4. DS-SE (and DS-DE) link timing

A link shall operate at a speed of 10 Mbaud, unless and until the system in which it operates shall determine that it may operate at a faster speed. A link may be operated at a speed lower than its nominal speed to suit the transmission medium being used. If a link is being operated at a lower speed, the same timing constraints apply at the receiver inputs.

| Parameter | Min | Max | units | Notes |
|---|---|---|---|---|
| | | | | |
| Nominal bit time | 5 | 100 | ns | 1 |
| tr/tf | 1? | 4? | ns | 2,3 |
| Ttstd/Ttdts | 2 | | ns | 1,4 |
| Ttsts/Ttdtd | ? | | ns | 1,5 |
| V(0) | 0 | 0.8 | V | 6 |
| V(1) | 2.0 | 5.25 | V | 6 |

Note 1. The maximum nominal bit time is determined by disconnect time-out period of the link receiver
Note 2. Rise/fall time of either Data or Strobe signals (see next section)
Note 3. Maximum rise/fall times greater than 2ns may limit the maximum speed of the link to less than 200Mbaud
Note 4. Minimum time between consecutive D and S edges which allow the receiver to discriminate them in correct order
Note 5. Minimum high/low time required to ensure that a transition is received
Note 6. Need definition of allowed transient excursions beyond V(0) min and V(1) max.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

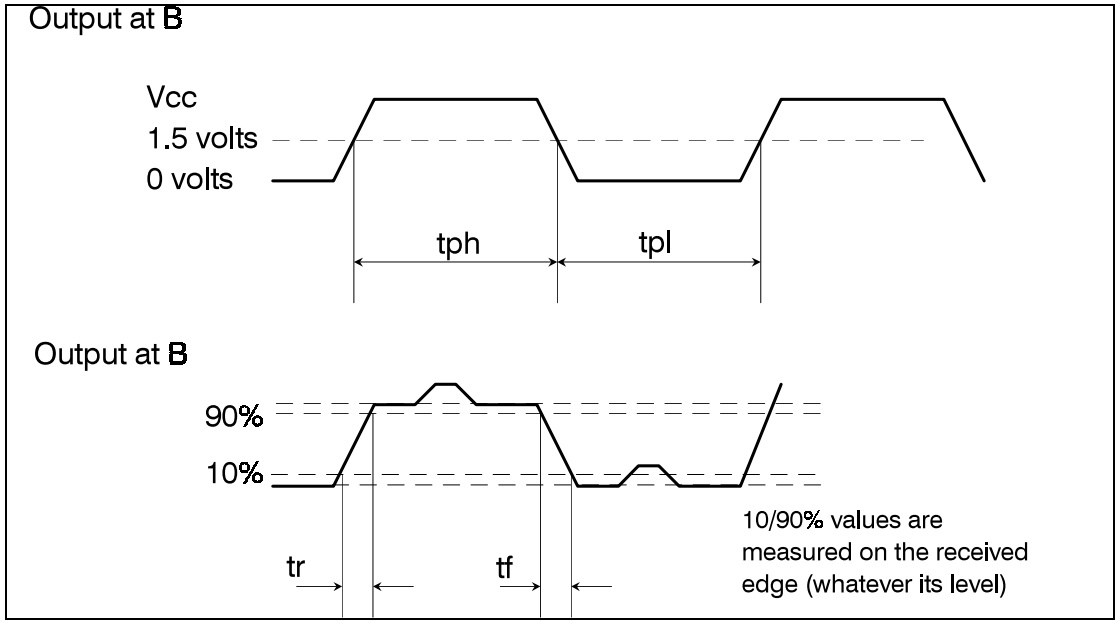


Figure 13: DS-SE Link Timings

**5.4.5. DS-SE Link Signals**

Information on a DS-SE link shall be transmitted in each direction using two lines, each satisfying the electrical constraints indicated in the previous section, referred to as data and strobe respectively. The data line carries binary data values, and the accompanying strobe line changes state each time the next bit has the same value as the previous one 1 (NB: This does not correspond with the usual meaning of `strobe', which would be a signal which indicates every time that another signal is valid). By this means each DS pair carries an encoded clock, in a way which allows up to a full bit-time of skew-tolerance between the two wires. Figure 1.3 shows the form of signals on the data and strobe wires.

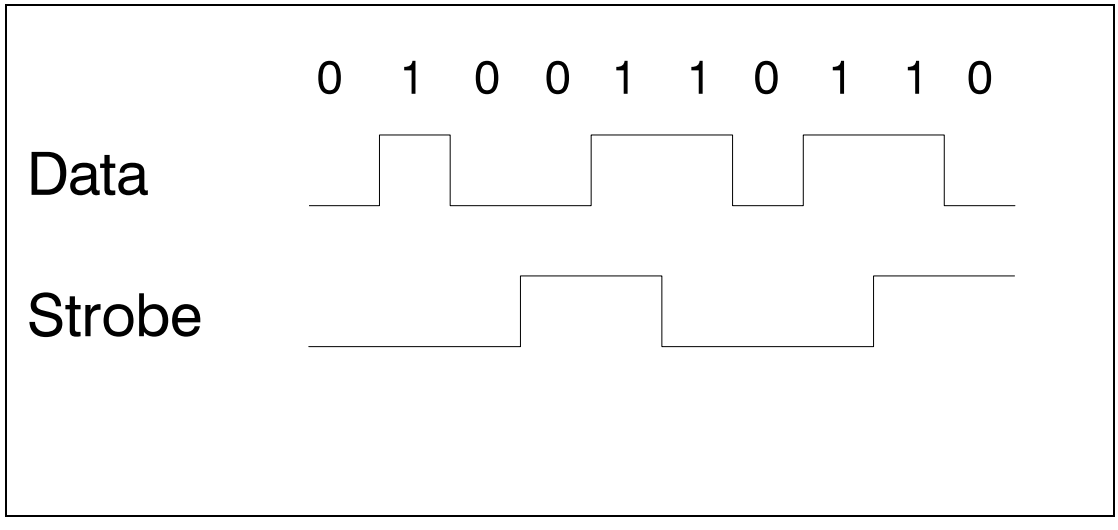


Figure 14: DS-SE Link Signal Encoding

Since the data-strobe system carries a clock, the links are asynchronous; the receiving device synchronizes to the incoming data. This means that DS-SE (and DS-DE) links `autobaud'; the only

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

restriction on the transmission rate is that it does not exceed the maximum speed of the receiver. It also simplifies clock distribution within a system, since the exact phase or frequency of the clock on a pair of communicating devices is not critical.

A DS-SE link output transmits information on a pair of lines, and a DS-SE link input receives information on a pair of lines. A DS-SE link interface shall comprise a DS-SE link output and a DS-SE link input.

A device providing one or more DS-SE link interfaces shall identify the pins used for each link interface by a convention (using subscription where multiple links are supported) based on function and direction of transmission:

| | |
|---|---|
| DS_SE_data_out | for output of data from the device through the link interface |
| DS_SE_strobe_out | |
| DS_SE_data_in | for input of data to the device through the link interface |
| DS_SE_strobe_in | |

A DS-SE link is constructed by connecting the DS_SE_data_out and DS-SE_strobe_out of one link interface to the DS-SE_data_in and DS-SE_strobe_in respectively of the other link interface, and vice versa.

The DS_SE_data_out and DS-SE_strobe_out of a link interface may be connected to the DS-SE_data_in and DS_SE_strobe_in respectively of the same link interface to create a loop-back, eg for link testing.

## 5.5. DS-DE Signal Level

### 5.5.1. General

Differential signalling may be used, eg for connections between equipment. Differential signalling provides the following advantages over single-ended signalling:

a. Increased noise immunity.
b. Avoidance of ground loops between equipment.

A DS-DE link interface may be constructed by connecting two differential drivers and two differential receivers to a DS-SE link interface as follows:

a. Each connection point of the DS-SE link output is connected to the input of a differential driver, generating a pair of balanced signals from the on-board DS-SE electrical specification. The two pairs of signals form the DS-DE link output.

b. Each connection point of the DS-SE link input is connected to the output of a differential receiver, converting two pairs of balanced signals into two signals conforming to the on-board DS-SE electrical specification. The two pairs of signals form the DS-DE link input.

A DS-DE link interface shall have the logical properties which it would inherit if connected to a DS-SE link interface in this way. This is the recommended implementation of a DS-DE link interface.

A DS-DE link is constructed by connecting two DS-DE link interfaces by a suitable eight-wire transmission line, in the manner defined in *.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

### 5.5.2. DS-DE Signals

Each connection in a DS-DE link must be made with a nominal 100 Ohm transmission line. The transmission line must be terminated at the receiver inputs: the combination of receiver input impedance and line termination must be 100 Ohms.

The signalling uses a pair of balanced pseudo-ECL signals. Pseudo-ECL signals have ECL-levels shifted by 5V.

| Parameter | min | typ | max | Unit |
|---|---|---|---|---|
| | | | | |
| Driver Differential output voltage | 0.8 | 1.0 | | V |
| Driver output skew | | 1.0 | 1.0 | ns |
| Driver output rise time (20% - 80%) | | | 2 | ns |
| Skew between drivers of data and strobe | | | 0.5 | ns |
| Receiver differential input voltage | 0.4 | 0.5 | | V |
| Receiver input impedance (including termination resistance) | | 100 | | Ohms |
| Receiver input common mode voltage | -1.2 | | 7.2 | V |

The DS-SE connection consists of four signals. Each of these has two corresponding signals on the differential side.

| **Single-Ended Side** | **Differential Side** | |
|---|---|---|
| | | |
| DS_SE_data_out | DS_DE_data_out_plus | DS_DE_data_out_minus |
| DS_SE_strobe_out | DS_DE_strobe_out_plus | DS_DE_strobe_out_minus |
| DS_SE_data_in | DS_DE_data_in_plus | DS_DE_data_in_minus |
| DS_SE_strobe_in | DS_DE_strobe_in_plus | DS_DE_strobe_in_minus |

In each differential pair, the **plus** version of the signal is positive going when the corresponding single ended signal is positive going, and vice versa.

A link shall connect two sets of differential pairs by connecting the DS_DE_data_out_plus of one set to DS_DE_data_in_plus of the other set, DS_DE_strobe_out_plus of one set to DS_DE_strobe_in_plus of the other set, and similarly for the _minus signals.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

### 5.6. DS-SE and DS-DE Character Level

In order to provide efficient support for higher level protocols, it is useful to be able to encode 'characters' which may contain a data byte or control information (in other standards these might be referred to as 'symbols'). Each character has a parity bit plus a control bit which is used to distinguish between data and control characters. In addition to the parity and control bits, data characters contain 8 bits of data, and control characters have two bits to indicate the character type (e.g. 'end of message'). This is illustrated in figure 15.

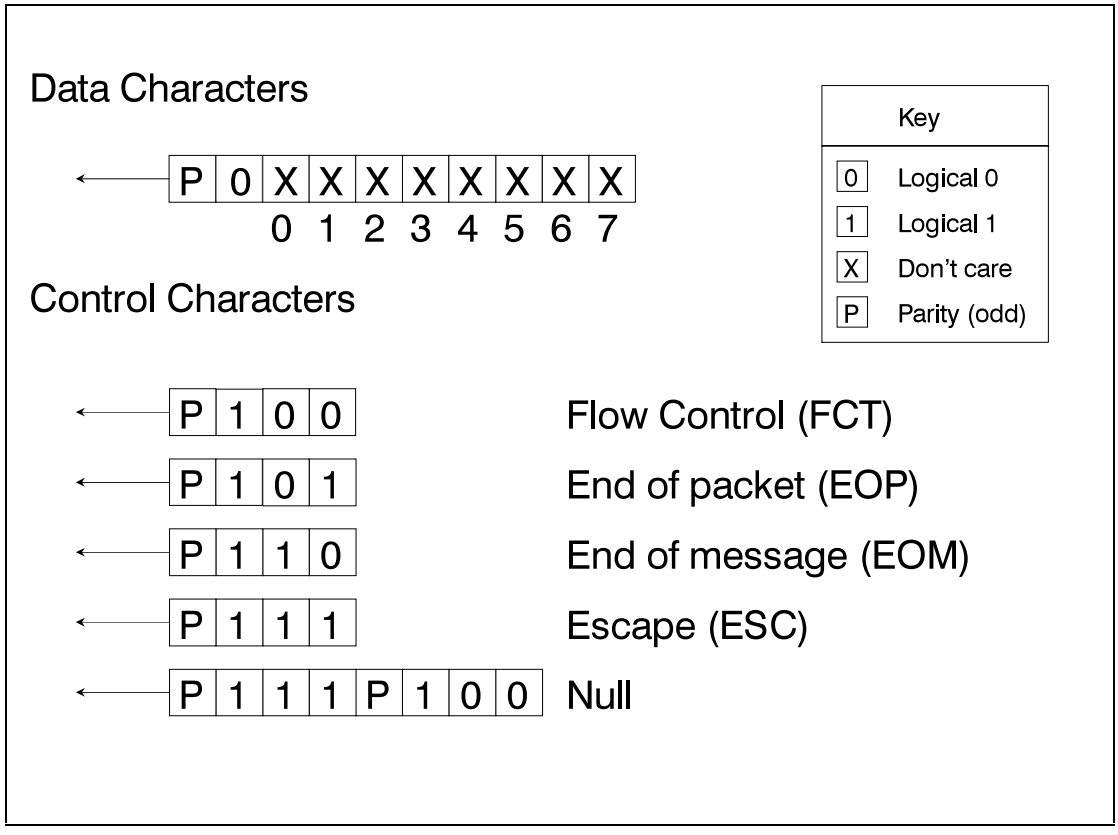


Figure 15: DS-Link Character Encoding

Data bits are transmitted using the 'little endian' convention, i.e. the least significant bit of data is transmitted first (immediately after the zero control bit).

The parity bit in any character covers the parity of the data/control flag in the same character, and the data or control bits in the previous character, as shown in figure 16. This allows an error in any single bit of a character, including the character type flag, to be detected even though the characters are not all the same length. Odd parity checking is used.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

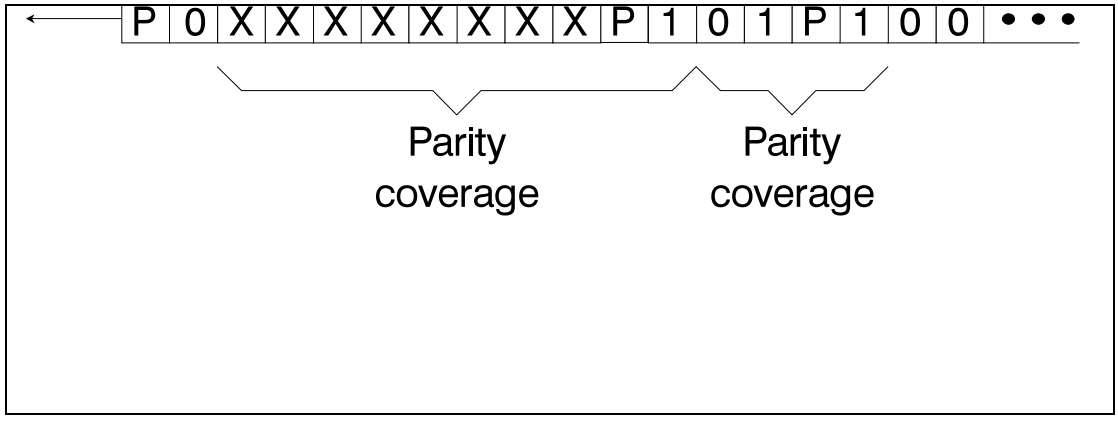


Figure 16: DS-Link Parity Coverage

Characters are either n_chars or l_chars.

N_chars are data characters or 'terminator' characters. The coding of the terminator characters is shown in table 2, in which P indicates the position of the parity bit in the character. The use of terminator characters is discussed in *.

Table 2: Terminator character codings

| | | |
|---|---|---|
| End of packet | EOP | P101 |
| End of message | EOM | P110 |

L_chars are link control characters. The coding of the link control characters is shown in table 3, and their application is discussed in *.

Table 3: Link control character codings

| | | |
|---|---|---|
| Flow control character | FCC | P100 |
| Escape token | ESC | P111 |
| Null character | NUL | ESC P100 |

N.B. The null 'character' is in fact composed of two characters. Therefore, the scope of the parity bit in a character that follows a null character includes only the second 4 bit control character of the null 'character' - i.e. it will be a `1' if the following character is a data character, and will be a `0' if following character is a control character.

Note that the character level of the protocol is independent of details of the higher levels, such as the amount of data contained in a packet, or the particular interpretations of packets of different types.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

## 5.7. DS-SE and DS-DE Exchange Level

### 5.7.1. General

Exchange level flow control (i.e. control of the flow of characters between devices) shall be performed on each link. The additional characters used are not visible to the higher-level packet protocol.

The exchange-level flow control mechanism prevents a link output from overrunning the input buffer of the corresponding link input. Each link input shall contain a buffer for at least 8 characters (more buffering than this may be provided). Whenever a link input has sufficient buffering available for a further 8 characters, the link interface shall transmit a flow control character (FCC) on the associated link output. Reception of this FCC by the link input of the connected link interface gives the corresponding link output permission to transmit a further 8 characters. Once the link output has transmitted a further 8 characters it shall wait until the link interface receives another FCC before transmitting any more characters. More than 8 characters of buffering on each link input may be provided in order to ensure that the next FCC is received before the previous batch of 8 characters has been fully transmitted, so the exchange level flow control does not restrict the maximum bandwidth of the link.

Exchange level flow control greatly simplifies the higher levels of the protocol, since it prevents data from being lost due to buffer overflow and so removes the need for re-transmission unless errors occur. To the user of the system, the net result is that a connected pair of DS-Links function as a pair of fully handshaken FIFOs, one in each direction.

Note that the link interface regulates the flow of data items without regard to the higher level objects that they may constitute. At any instant the data items buffered by a link interface may form part or all of one or more consecutive higher-level objects. L_chars, including FCC, do not belong to such objects and are not buffered.

In the absence of other characters, a DS-Link transmits null characters as defined in Table 3. This ensures the following:

a. Physical disconnection of a link is immediately detected; because a DS-Link transmits characters continuously even in the absence of a message, a physical disconnect can be assumed if the inputting link detects that there are no characters being received, given that some have been received previously.

b. The final parity bit is transmitted; because the parity bit in each character validates the parity of the previous character, the final character of a message would not have its parity bit sent if no characters were sent in the absence of a message.

### 5.7.2. DS-SE and DS-DE Initialisation

On power on, a DS-SE link output shall hold both signals low.

It shall remain in this state until it is instructed to commence operation, in which case it shall transmit only NUL and FCC characters until it is provided with data to transmit. Note that receipt of a character may be taken as an instruction to commence operation.

The parity bit of the first character to be transmitted on a link shall be a 0 (i.e. assuming that the previous character, had it existed, has even parity).

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

### 5.7.3. DS-SE and DS-DE Error Detection

#### 5.7.3.1. General

DS-Links can detect various types of errors. The handling of link errors is a subject of Informative Annex *.

The DS-Link exchange protocol allows two common types of error to be detected. Firstly because each output link, once started, continues to transmit an uninterrupted stream of characters, the physical disconnection of a link can be detected, and secondly, the parity system will detect all single bit errors at the DS-Link exchange level.

#### 5.7.3.2. Disconnection errors

If the links are disconnected for any reason whilst they are running then flow control and character synchronization may be lost. In order to restart the link it is therefore necessary to reset both ends to a known flow control and character synchronization point.

Disconnection is detected if, after a character has been received, no characters are seen by a link input in any 1.6μs window. Once a disconnection error has been detected the link halts its output. This will subsequently be detected as a disconnect error at the other end, and will cause that link to halt its output also. It then resets itself, and waits 12.8 microseconds before allowing communication to restart. This time is sufficient to ensure that both ends of the link have observed disconnection and cycled through reset back into the waiting state. The connection may now be restarted as described above.

#### 5.7.3.3. Parity errors

Following a parity error, both character-level synchronization and flow control status are no longer valid, therefore both ends of the link must be reset. This is done autonomously by the DS-Link using an exchange-of-silence protocol.

When a link interface detects a parity error on its input it halts its output. This will subsequently be detected as a disconnect error at the other end, and will cause that link interface to halt its output also, causing a disconnect to be detected at the first end. The normal disconnect behaviour described above will then ensure that both ends are reset (irrespective of line delay) before either is allowed to restart.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

# 6. TS-FO Link Optical Interconnect

## 6.1. TS-FO: Physical Medium

### 6.1.1. TS-FO Link optical connector

[Editor's note: this section is to be completed]

### 6.1.2. TS-FO Link Optical Cable

The cable is a fiber ribbon cable with 2 to 18 fibers. An example of a fiber is shown in Figure 17.

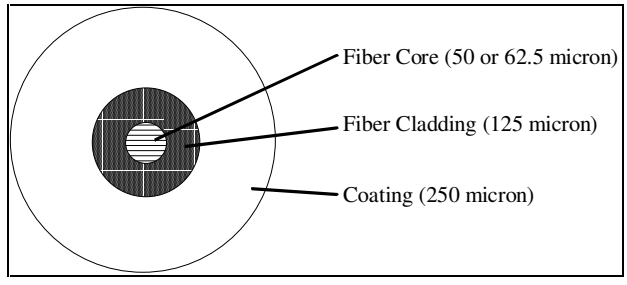


Figure 17: Example of TS Link Fiber Cable Structure

The optical charcteristics of the cable are given in Table 4.

Table 4: Characteristics of TS Link Ribbon Cable.

| Characteristic | Units | Value |
|---|---|---|
| | | |
| Core Diameter | μm | 50 ± 3 |
| N.A. | - | 0.2 ± 0.015 |
| Cladding Diameter | μm | 125 ± 2 |
| Core to Cladding Offset | μm | < 3 |
| Cladding non-circularity | - | < 2% |
| Attenuation at 850 nm | dB/km | 2.4 (min)<br>3.0 (max) |
| Bandwidth at 850 nm | MHz.km | 400 (min)<br>600 (max) |

## 6.2. TS-FO Link Optical Signal Level

Two speeds are specified, TS-FO-01 (125 Mbaud, equivalent to 100 Mbaud on DS links) and TS-FO-02 (250 Mbaud, equivalent to 200 Mbaud on DS links).

The fiber link has the general characteristics as given in Table 5.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Table 5: TS Link Fiber Optic Link Characteristics

| Fiber Characteristic | Units | GI 62.5/125 | GI 50/125 |
|---|---|---|---|
| N.A. | | 0.275 | 0.2 |
| | | | |
| Transmitter | | | |
| Spectral Centre Wavelength | nm | 860 (min)<br>900 (max) | 860 (min)<br>900 (max) |
| Spectral Width | nm (FWHM) | 60 | 60 |
| Fiber-coupled Power | dBm | -10 (min)<br>-13 (max) | -15 (min)<br>-18 (max) |
| Optical Rise/Fall Time | ns | 2 (max) | |
| | | | |
| Receiver | | | |
| Receiver Power | dBm | -15 (min)<br>-22 (max) | |
| Optical Rise/Fall Time | ns | 2.5 (max) | |

Single ended signalling is used for connections with the front-end circuit for TS Links. Each DS Link output (data and strobe) is connected to an interface generating a single signal towards the input of the optical transmitter. The single output of the optical receiver is connected to an interface generating a DS Link data and strobe input.

The electrical characteristics are given in Table 6.

Table 6: Electrical Characteristics for TS electrical-optical-electrical interface

| Parameter | Unit | Min | Typical | Max |
|---|---|---|---|---|
| | | | | |
| $V_{OL}$ | V | | 0.2 | 0.4 |
| $V_{OH}$ | V | 2.4 | 4.5 | |
| $V_{IL}$ | V | | | $0.3 * V_{DD} = 1.5$ |
| $V_{OH}$ | V | $0.7 * V_{DD} = 3.5$ | | |

The single ended on board TS Link connection consists of two signals: TS-Link-in and TS-Link-out as given in Table 7.

Table 7: DS and TS Link Equivalent Signals.

| DS Link | TS Link |
|---|---|
| | |
| DS-Link-data-out<br>DS-Link-Strobe-out | TS-Link-in |
| DS-Link-data-in<br>DS-Link-Strobe-in | TS-Link-out |

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

## 6.3. TS-FO Link Optical Character and Exchange Levels

When transmitted through optical fibre, DS-SE links need to be converted from the two wire signalling of the Data and Strobe signals into a single wire signalling which includes clock and data. The coding is chosen to ensure that single bit errors do not generate multiple bit errors on decoding, and to avoid the overheads of CRC checking (which would double the length of small packets).

### 6.3.1. Symbols

Each character is coded as two six-bit symbols, where each symbol has three ones and three zeros - hence the code is described as a 'three of six' code. There are twenty valid symbols which have three ones and three zeros, and these are split into sixteen values that are used for data and into four values which are used for control. The table below shows the symbols allocated to the sixteen data nibbles and four control symbols (the symbols are drawn as on a oscilloscope trace, with the least significant bit (ie the first transmitted) on the LEFT).

| Hex | lsb-msb |
|---|---|
| 0 | 011010 |
| 1 | 101001 |
| 2 | 011001 |
| 3 | 110001 |
| 4 | 001101 |
| 5 | 101100 |
| 6 | 011100 |
| 7 | 110100 |
| 8 | 001011 |
| 9 | 110011 |
| A | 010011 |
| B | 110010 |
| C | 001110 |
| D | 100110 |
| E | 010110 |
| F | 100101 |

This selection of data symbols for the particular four bit values has several advantages. Apart from the values 0 and F the coding is systematic in respect of bits 0, 1, 3 and 4. The redundant bits, 2 and 5, are placed in such a manner as to guarantee transitions within each symbol in at least two locations, each location occurring in one of two places. Thus, there will always be a transition either between the first and second bits or the second and third bits, and either between the fourth and fifth bits or the fifth and sixth bits.

The symbols 000111 and 111000 are not utilised because they increase the run length and total disparity count or digital sum variation (DSV) of the code. Furthermore, the symbols 010101 and 101010 are not used for data symbols so as to limit the number of adjacent alternating bits in a data stream. This has the advantage that the symbols 010101 and 101010 can be used for control purposes and are readily identifieable as having a greater number of transitions than any data symbol.

### 6.3.2. Data characters

Data characters are comprised of two data symbols. Bits 0 to 3 of each data character are transmitted by the first symbol and bits 4 to 7 are transmitted in the second symbol.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

### 6.3.3. Control characters

Control characters are formed by composing two symbols, called CONTROL and CONTROL*. Each of these two symbols is always one of 101010 and 010101. Which is which depends upon the final bit of the preceding symbol:-

| Previous symbol | CONTROL | CONTROL* |
|---|---|---|
| xxxxx0 | 010101 | 101010 |
| xxxxx1 | 101010 | 010101 |

This definition enables (i) the boundaries between data tokens and control tokens to be easily identified and (ii) token synchronisation to be checked.

The control characters NUL and FCC are comprised of CONTROL and CONTROL* symbols. The control characters EOM and EOP are comprised of a control symbol and a data symbol, where the data symbol is the longitudinal error check code (described separately below).

| Control character | Symbols | Previous symbol | Symbols (binary) |
|---|---|---|---|
| | | | |
| NUL | CONTROL CONTROL* | xxxxx1 | 101010 101010 |
| | | xxxxx0 | 010101 010101 |
| FCC | CONTROL CONTROL | xxxxx1 | 101010 010101 |
| | | xxxxx0 | 010101 101010 |
| EOP | CONTROL checksum | xxxxx1 | 101010 checksum |
| | | xxxxx0 | 010101 checksum |
| EOM | checksum CONTROL | checksum=xxxxx1 | checksum 101010 |
| | | checksum=xxxxx0 | checksum 010101 |

### 6.3.4. Initialisation

A final control character, comprised of four symbols, is used for initialisation. This is a sequence of as many transitions as possible.

| Control character | Symbols | Previous symbol | Symbols (binary) |
|---|---|---|---|
| Initialise | CONTROL CONTROL*<br>CONTROL* CONTROL* | xxxxx1 | 101010 101010<br>101010 101010 |
| | | xxxxx0 | 010101 010101<br>010101 010101 |

### 6.3.5. Initialisation sequence

Each transmitter starts by transmitting initialise tokens. When a device has been both transmitting and receiving initialise tokens for 125 microseconds, it sends NUL tokens. When a device has been transmitting NUL tokens for at least 375 microseconds and receiving NUL tokens for 125 microseconds, it transmits a single initialise token. When a device has both sent and received a single initialise token, it waits a further 125 microseconds and then is free to start normal operation.

If a device sends the single initialise token but does not receive one for more than 375 microseconds, it sends another after 500 microseconds, and continues with this until it receives the initialise token.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

#### 6.3.6. Error detection

Although DS-Links are designed to be reliable, and fibre interfaced TS-Links are designed to be similarly reliable, the attenuation, dynamic range and distance of fibre connected links mean that they are more prone to errors and that these errors must be detected.

Checks are applied to the connection itself, to packets, to characters, to symbols and to bit-sequences.

##### 6.3.6.1. Disconnection

The absence of an input signal implies disconnection.

##### 6.3.6.2. Packets checked with longitudinal error check code

The code for each symbol is an unusually rigorous check, checking for all single-bit errors and only missing errors when one bit is turned from a one to a zero, and another is turned from a zero to a one. These errors can be detected by a longitudinal check, and the check used is a longitudinal (even?) parity check of the nibbles which have subsequently been encoded as the six bits of each symbol.

The four check bits are converted into the corresponding data symbol for transmission, decoded on receipt, and compared with the check computed n the received data symbols.

##### 6.3.6.3. Character level checks

All characters shall be checked. Any illegal character which is not otherwise reported as being a symbol or synchronisation error shall result in immediate link disconnection.

##### 6.3.6.4. Symbol errors which do not cause loss of synchronisation

Errors in a single symbol, where the received symbol has four ones and two zeros, or two ones and four zeros, may be passed through to any possible error correction logic. Any symbol with five or six ones or zeros indicates an uncorrectable error and the link shall immediately disconnect.

##### 6.3.6.5. Bit-level checks and synchronisation

Once the receiver has established character boundaries, it should stay in synchronisation with these character boundaries thereafter. If a control character is received which is not in synchronism, an error has occurred.

The Initialise, NUL and FCC characters all have at least nine alternating bits, whereas the longest length of alternating bits possible with data characters is eight. If a count is detected, therefore, of nine alternating bits, there should be exactly three more bits to make the character; if there are more or fewer bits to the character boundary, synchronisation has been lost.

The EOP and EOM characters have a larger disparity count (±3) than any other characters. If the disparity count reaches ±3, the next bit must be the fourth of a character, and the remaining bits of the symbol should bring the disparity back to zero; if either of these conditions is not met, disparity has been lost.

A further case of loss of synchronisation is indicated by the disparity at the end of each six-bit symbol, which should be zero. A possible correctable error would cause this disparity to become ±1. If the symbol boundary disparity is reset to zero after this error, and all subsequent character boundaries are at zero disparity until a correctly aligned character, character synchronisation may be regarded as having being maintained. If, however, there is a further non-zero disparity at a symbol boundary, before a control token (whether or not correctly aligned), synchronisation has been lost.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

#### 6.3.6.6. Recovery of character synchronisation

On loss of character synchronisation, the receiver either shall disconnect and re-initialise, or shall attempt to re-synchronise with the new timing. In either case, the (part) packet immediately preceding the loss of synchronisation shall be discarded. If the receiver attempts to re-synchronise without re-initialising, the (part) packets received until after synchronisation has been re-established shall also be discarded. The receiver shall not set itself into synchronisation until it has received at least two control characters with the same character boundary.

There are conditions when a single bit error may cause the count of alternating bits to exceed eight, or the disparity count to exceed ±3, and for such single bit errors to be correctable; it is, however, much simple and safer to class all these errors as uncorrectable, and this practice is recommended.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

# 7. HS-SE-10

## 7.1. Copper Link: Physical Medium

[Editor's note: all sections in this chapter are in the process of being specified, and are therefore incomplete]

### 7.1.1. General Electrical Characteristics

| | |
|---|---|
| Characteristic impedance of the complete transmission line: | 50 ohms ± 10 % |
| Total attenuation power budget (output buffer to input buffer): | TBS |
| Crosstalk: | TBS |
| | |
| EMC at 10 m: | Limit EN 55022 (47 dBµV) |
| ESD susceptibility: | 12 KV |
| Sector noise: | 1 KV |
| E field: | 3 V/m |

### 7.1.2. Printed Circuit Board (PCB)

Characteristic impedance
Microstrip, stripline, vias
IA: Materials, examples

### 7.1.3. Single Minicoaxial Cable

This section describes the basic characteristics of a single minicoaxial cable.

#### 7.1.3.1. Physical

| Characteristic | Value |
|---|---|
| | |
| Central conductor cross section | 0.5mm$^2$ minimum |
| External diameter | 3 mm maximum |

#### 7.1.3.2. Electrical

| Characteristic | Value |
|---|---|
| | |
| Characteristic impedance | 50 ohms ± 10 % |
| Cable attenuation | < 6dB per 10 meters |

### 7.1.4. Link Cable

#### 7.1.4.1. General

The basic bidirectional HS Link uses a 'Link Cable' comprising of two coaxial cables (one for each direction), and is refered to as 'single-way' link cable. A 'Multi-way' link cable refers to two or more single-way cables physically combined to form one link cable (e.g. a two-way link cable contains four

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

cables etc.). All the individual coaxial cables in a link cable conform to the characteristics described above.

### 7.1.4.2. Shielding

A link cable may be shielded or unshielded (double braid or single braid respectively). (See figure 17.) Double braiding is used in order to comply with the EMC specifications and to protect the transmission line from external noise sources. Link cables outside a box should be shielded. Link cables inside a box may be shielded.

### 7.1.4.3. Mechanical Performance

Link cables shall have the following mechanical performance:

| Characteristic | Value (shielded) Single way | Multi-way | Value (unshielded) Single way | Multi-way |
|---|---|---|---|---|
| | | | | |
| Flexibility | TBS | | | |
| Flex life | | | | |
| Clamping envelope | | | | |
| Minimum bending radius (static operation) | | | | |
| | | | | |

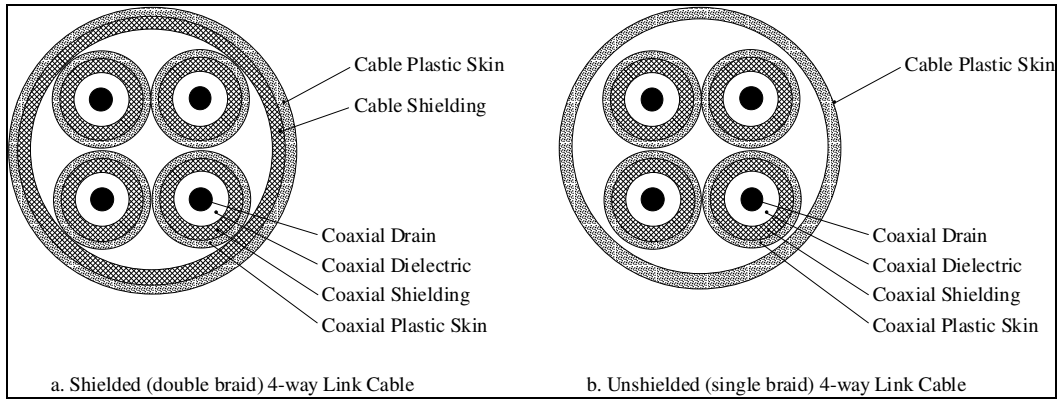


Figure 17: Single braid and Double Braid Cables

### 7.1.5. Link Connectors

### 7.1.5.1. General

Three types of link connector are defined:

a. PCB to link cable (inside a box)
b. PCB to link cable across a panel (from inside to outside a box and v.v.)
c. Link cable to link cable across a panel (from inside to outside a box and v.v.)

A single-way link connector connects a single way link and a multi-way link connector connects a multi-way link. Sketch forms of these connectors are shown in Figure 18.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

[Editor's note: link connectors are in the process of being specified]

#### 7.1.5.2. Shielding

A link connector shall be shielded if the link cable is shielded. There shall be a 360° mechanical connection between the outer braid of the link cable and the connector housing. The connector housing shall have a 360° mechanical connection with the panel with an appropriate fixing mechanism (TBS, e.g. screw) in order to provide a very good electrical connection.

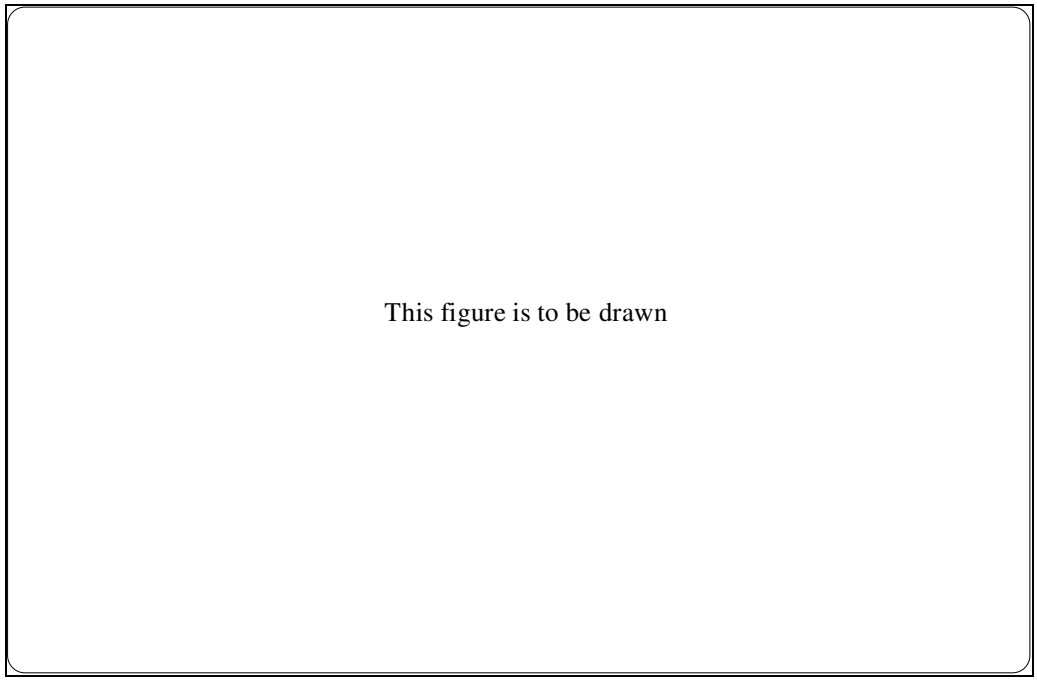


Figure 18: Sketch forms of connectors

### 7.1.6. Coaxial Contacts

#### 7.1.6.1. General

All HS-Link connectors contain coaxial contacts. The number of contacts per connector is equal to the number of cables in each link cable (the minimum being two contacts).

#### 7.1.6.2. Characteristics

General

| Characteristic | Value |
|---|---|
| | |
| Impedance | 50 ohms ± 10% |
| Bandwidth | 3 GHz |
| Operating temperature | TBS |

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Electrical

| | | |
|---|---|---|
| Insertion loss | 0.3 √F (GHz) | |
| HF leakage | - 40 dB (min) at 1 GHz | |
| Central contact resistance | Initial<br>≤ 5 mΩ | After test<br>≤ 15 mΩ |
| Isolation resistance | 10E6 ohms | |
| Test voltage | 500 V eff. | |
| Operation voltage | TBS | |

Mechanical

| | | |
|---|---|---|
| Engaging force | TBS | |
| Withdrawl force | TBS | |
| Endurance | TBS | |
| Tensil strength | Single braid<br>TBS | Double braid |

Enviroment

| | |
|---|---|
| Vibration | TBS |
| Shock | TBS |
| High temperature resistance | TBS |

Materials

| | |
|---|---|
| Body | Brass |
| Central contacts | Copper-berylium |
| Insulators | PTFE |

Protection

| | |
|---|---|
| Body | Nickel or Gold |
| Central contacts | Gold |

## 7.2. Copper Link: Signal Level

### 7.2.1. General

HS-SE-10 signal levels are single ended.

The guaranteed operating data rate shall be the following:

| Characteristic | Value |
|---|---|
| | |
| Operating data rate | 700 Mbaud to 1 Gbaud |
| Bit period | 1.428 ns to 1.0 ns |
| Maximum rise and fall time (10% - 90%) | 300 ps |
| Minimum rise and fall time (10% - 90%) | TBS |

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

The electrical model of the link input and output buffers is shown in Figure 19.

The receiver amplifier should be polarized around its switching threshold, nominally at $V_{DD}/2$ for a CMOS inverter. This results in a maximum receiver sensitivity. The receiver input impedance (defined by $R_3$ // $R_4$) shall be 50 ohms ± 10 %.

The driver is modelled by two current sources, pulling the line to logic level '1' or logic level '0' as a function of the bit to be sent.

The nominal signal levels corresponding to these logic levels are as follows:

$$Vs = \text{threshold voltage of the receiver amplifier} \approx \frac{V_{DD}}{2} \text{ and}$$

$$(V_S - V_L) = (V_H - V_S) \approx 400 \text{ mV}$$

where $V_L$ = logic level '0' on the line
$V_H$ = logic level '1' on the line

and $R_3 // R_4 = Zc$

where Zc = characteristic impedance of the transmission line (50 ohms).

### 7.2.2. Line Signal Levels - Driver Side

Owing to the nature of the line signal levels, the latter should be observed via a high impedance probe on a fully connected line. For observation purposes at the driver side of the line, the line loss of the transmission medium shall be less than 0.5dB. The levels shall meet the requirements shown in Table 4 for $V_{DD}$ = 3.3 volts (nominal) and Table 6 for $V_{DD}$ = 5.0 volts (nominal).

Table 4: Driver side line logic levels for $V_{DD}$ = 3.3 volts (nominal)

| Symbol | Parameter | Units | Min | Typ | Max |
|---|---|---|---|---|---|
| | | | | | |
| $V_{OLS}$ | Serial output low level | V | 1.15 | 1.25 | 1.35 |
| $V_{OHS}$ | Serial output high level | V | 1.95 | 2.05 | 2.15 |

Table 5: Driver side line logic levels for $V_{DD}$ = 5.0 volts (nominal)

| Symbol | Parameter | Units | Min | Typ | Max |
|---|---|---|---|---|---|
| | | | | | |
| $V_{OLS}$ | Serial output low level | V | 2.0 | 2.1 | 2.2 |
| $V_{OHS}$ | Serial output high level | V | 2.8 | 2.9 | 3.0 |

Alternatively, the output of the driver may be AC coupled (see below) directly into the 50 ohm termination of an oscilloscope. This allows measurement of the line swing only (as DC levels are masked by the AC coupling). The swings shall meet the requirements shown in Table 6.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Table 6: Driver side line swing when AC coupled into 50 ohm termination

| Symbol | Parameter | Units | Min | Typ | Max |
|---|---|---|---|---|---|
| | | | | | |
| $V_{SW}$ | Serial line swing | V | 0.6 | 0.8 | 1.0 |

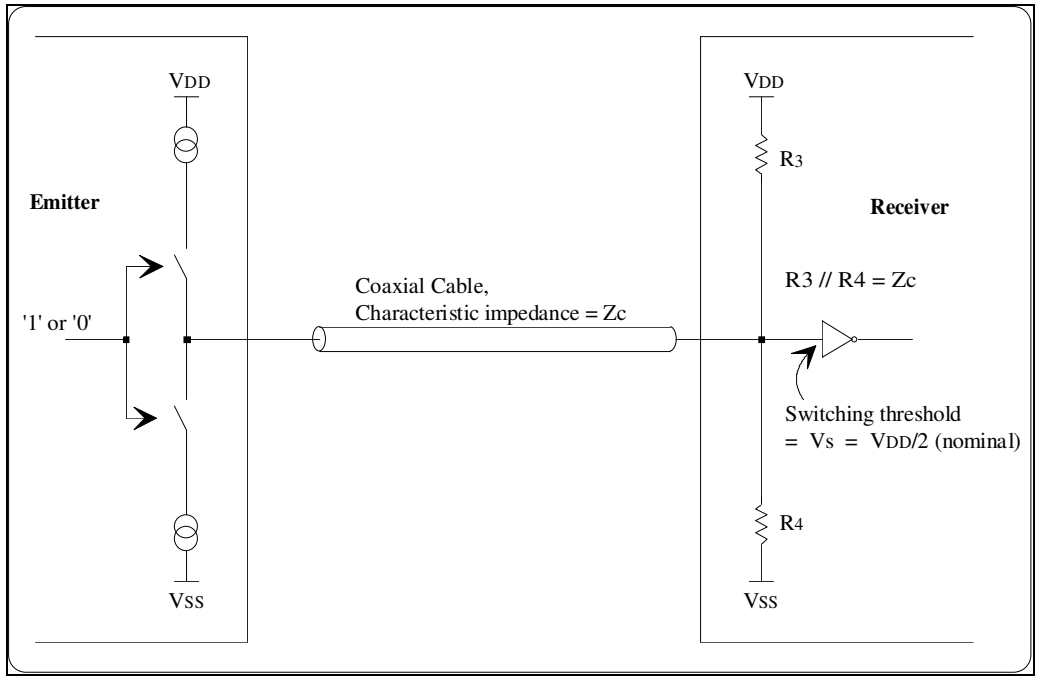


Figure 19: Input and Output Buffer Electrical Model

### 7.2.3. AC Coupling of the link

The HS Link 8B/12B coding scheme is DC balanced, with a maximum disparity of 8 bits.This DC balance allows the code to be transmitted across AC coupled links, which is advantageous for both fiber and coaxial cable systems. Therefore the HS-SE-10 links using such media should be AC coupled.

Transformer coupling or capacitive coupling may be used.

Because of static electricity considerations, it is undesirable to leave the cable conductors completely isolated from ground. A bleeder resistor (10 MΩ) should be included.

[Editor's note: this section is to be completed]

For HS Links that operate between two components on the same PCB (i.e. using only PCB tracks), AC coupling may be used, but is less essential.

### 7.2.4. Receiver Electrical and BER

[Editor's note: this section is to be completed]

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

## 7.3. Character Level (8B/12B Code)

### 7.3.1. General

The HS-SE-01 uses an 8B/12B DC balanced code. This code provides the following features:

a. A positive going synchronization transition at the beginning of every character, to enable clock recovery.
b. An odd parity bit per character, enabling the detection of single bit errors.
c. DC balance, with a maximum disparity of 8 bits.
d. 126 out-of-bound control characters.

### 7.3.2. Transmission Characters

Define d[7:0] to be the byte to be transmitted, where d[7] is the most significant bit. To each byte, an odd parity, AP, and an inversion bit, I, are added to form the encoded character (on 10 bits). The transmission character (on 12 bits) is the encoded character plus the Start ('1') and Stop ('0') bits which provide the positive going synchronization transition.

The transmission character therefore has the form:

Start ('1') AP e[0] e[1] e[2] e[3] e[4] e[5] e[6] e[7] Invert, I Stop ('0')

where e[7:0] is d[7:0] or $\overline{d[7:0]}$, as defined below.

On the serial line the bits are sent in the order Start to Stop ([0] = LSB).

The AP bit is defined as follows:

AP = 1 if there is an even number of 1's in e[7:0] and I.
AP = 0 if there is an odd number of 1's in e[7:0] and I.

### 7.3.3. DC balance

A code that is DC balanced is one that has a constant DC component regardless of the data pattern. This provides many advantages for high data rate transmission on fibre optic and long distance copper media, as described in *.

Assigning +1 and -1 to bit levels 1 and 0 respectively provides a measure of the DC component in each transmitted character. The *disparity*, D, of a data word is the difference between the number of 1's and the number of 0's in the word - positive and negative disparities refer to an excess of 1's and 0's respectively. The *running disparity*, RD, is the running (cumulative) sum of the disparities of all previous characters. In a code that is 'ideally' DC balanced each transmitted character has a D = 0 and thus *RD* is always equal to 0. However, perfect DC balance is not required in practice.

To maintain DC balance, the RD has always to tend towards zero. For each character to be sent, the following algorithm is used: calculate the disparity of the encoded character (since the Start and Stop bits are 1 and 0 respectively, they do not affect the disparity) with I = 0. If the result will reduce the absolute value of RD, then transmit the character as it is (i.e. e[7:0] = d[7:0], I = 0). If the result will increase the absolute value of RD, then invert the encoded character and set I = 1 (i.e. e[7:0] = $\overline{d[7:0]}$, I = 1), so that the inverted character reduces or leaves unchanged the absolute value of RD. The receiver uses the bit I to determine whether the character has been inverted or not. By its definition, the AP is also inverted if d[7:0] and I are inverted. In summary:

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

| | | |
|---|---|---|
| If RD ≤ 0 and D > 0 | then I = 0, e[7:0] = d[7:0] | (character non-inverted) |
| If RD < 0 and D ≤ 0 | then I = 1, e[7:0] = $\overline{d}$[7:0] | (character inverted) |
| If RD ≥ 0 and D < 0 | then I = 0, e[7:0] = d[7:0] | (character non-inverted) |
| If RD > 0 and D ≥ 0 | then I = 1, e[7:0] = $\overline{d}$[7:0] | (character inverted) |

### 7.3.4. Control Characters

When the disparity of the encoded character is zero it does not effect the RD, and can therefore be transmitted either non-inverted or inverted. This property enables the transmission of control characters. As defined above, if D = 0, the character is transmitted inverted. Control characters are therefore defined as characters which have D = 0 but are transmitted non-inverted. This gives 126 possible control characters.

### 7.3.5. The Code

Table 7 details the 256 data characters, giving the decimal value, the binary value, the transmission character and the disparity. For data characters which have a D ≠ 0, the non-inverted (first row) and inverted forms (second row) of the character is shown. Note that D can take the values 0, ±4 and ±8 only.

Table 8 details the 126 control characters, giving the control character ID (0 to 125), the decimal value, the binary value and the transmission character (the disparity being zero and the character being non-inverted).

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

**Table 7: Data Characters**

| Dec | Bin, d[0:7] | S-AP-e[0:7]-I-S | D |
|---|---|---|---|
| | | | |
| 0 | 00000000 | 1 1 000000000 0 | -8 |
| | | 1 0 111111111 0 | 8 |
| 1 | 10000000 | 1 0 100000000 0 | -8 |
| | | 1 1 011111111 0 | 8 |
| 2 | 01000000 | 1 0 010000000 0 | -8 |
| | | 1 1 101111111 0 | 8 |
| 3 | 11000000 | 1 1 110000000 0 | -4 |
| | | 1 0 001111111 0 | 4 |
| 4 | 00100000 | 1 0 001000000 0 | -8 |
| | | 1 1 110111111 0 | 8 |
| 5 | 10100000 | 1 1 101000000 0 | -4 |
| | | 1 0 010111111 0 | 4 |
| 6 | 01100000 | 1 1 011000000 0 | -4 |
| | | 1 0 100111111 0 | 4 |
| 7 | 11100000 | 1 0 111000000 0 | -4 |
| | | 1 1 000111111 0 | 4 |
| 8 | 00010000 | 1 0 000100000 0 | -8 |
| | | 1 1 111011111 0 | 8 |
| 9 | 10010000 | 1 1 100100000 0 | -4 |
| | | 1 0 011011111 0 | 4 |
| 10 | 01010000 | 1 1 010100000 0 | -4 |
| | | 1 0 101011111 0 | 4 |
| 11 | 11010000 | 1 0 110100000 0 | -4 |
| | | 1 1 001011111 0 | 4 |
| 12 | 00110000 | 1 1 001100000 0 | -4 |
| | | 1 0 110011111 0 | 4 |
| 13 | 10110000 | 1 0 101100000 0 | -4 |
| | | 1 1 010011111 0 | 4 |
| 14 | 01110000 | 1 0 011100000 0 | -4 |
| | | 1 1 100011111 0 | 4 |
| 15 | 11110000 | 1 0 000011111 0 | 0 |
| 16 | 00001000 | 1 0 000010000 0 | -8 |
| | | 1 1 111101111 0 | 8 |
| 17 | 10001000 | 1 1 100010000 0 | -4 |
| | | 1 0 011101111 0 | 4 |
| 18 | 01001000 | 1 1 010010000 0 | -4 |
| | | 1 0 101101111 0 | 4 |
| 19 | 11001000 | 1 0 110010000 0 | -4 |
| | | 1 1 001101111 0 | 4 |
| 20 | 00101000 | 1 1 001010000 0 | -4 |
| | | 1 0 110101111 0 | 4 |
| 21 | 10101000 | 1 0 101010000 0 | -4 |
| | | 1 1 010101111 0 | 4 |
| 22 | 01101000 | 1 0 011010000 0 | -4 |
| | | 1 1 100101111 0 | 4 |
| 23 | 11101000 | 1 0 000101111 0 | 0 |
| 24 | 00011000 | 1 1 000110000 0 | -4 |
| | | 1 0 111001111 0 | 4 |
| 25 | 10011000 | 1 0 100110000 0 | -4 |
| | | 1 1 011001111 0 | 4 |
| 26 | 01011000 | 1 0 010110000 0 | -4 |
| | | 1 1 101001111 0 | 4 |
| 27 | 11011000 | 1 0 001001111 0 | 0 |
| 28 | 00111000 | 1 0 001110000 0 | -4 |
| | | 1 1 110001111 0 | 4 |
| 29 | 10111000 | 1 0 010001111 0 | 0 |
| 30 | 01111000 | 1 0 100001111 0 | 0 |
| 31 | 11111000 | 1 1 000001111 0 | 0 |
| 32 | 00000100 | 1 0 000001000 0 | -8 |
| | | 1 1 111110111 0 | 8 |
| 33 | 10000100 | 1 1 100001000 0 | -4 |
| | | 1 0 011110111 0 | 4 |
| 34 | 01000100 | 1 1 010001000 0 | -4 |
| | | 1 0 101110111 0 | 4 |
| 35 | 11000100 | 1 0 110001000 0 | -4 |
| | | 1 1 001110111 0 | 4 |
| 36 | 00100100 | 1 1 001001000 0 | -4 |
| | | 1 0 110110111 0 | 4 |
| 37 | 10100100 | 1 0 101001000 0 | -4 |
| | | 1 1 010110111 0 | 4 |
| 38 | 01100100 | 1 0 011001000 0 | -4 |
| | | 1 1 100110111 0 | 4 |
| 39 | 11100100 | 1 0 000110111 0 | 0 |
| 40 | 00010100 | 1 1 000101000 0 | -4 |
| | | 1 0 111010111 0 | 4 |
| 41 | 10010100 | 1 0 100101000 0 | -4 |
| | | 1 1 011010111 0 | 4 |
| 42 | 01010100 | 1 0 010101000 0 | -4 |
| | | 1 1 101010111 0 | 4 |
| 43 | 11010100 | 1 0 001010111 0 | 0 |
| 44 | 00110100 | 1 0 001101000 0 | -4 |
| | | 1 1 110010111 0 | 4 |
| 45 | 10110100 | 1 0 010010111 0 | 0 |
| 46 | 01110100 | 1 0 100010111 0 | 0 |
| 47 | 11110100 | 1 1 000010111 0 | 0 |
| 48 | 00001100 | 1 1 000011000 0 | -4 |
| | | 1 0 111100111 0 | 4 |
| 49 | 10001100 | 1 0 100011000 0 | -4 |
| | | 1 1 011100111 0 | 4 |
| 50 | 01001100 | 1 0 010011000 0 | -4 |
| | | 1 1 101100111 0 | 4 |
| 51 | 11001100 | 1 0 001100111 0 | 0 |
| 52 | 00101100 | 1 0 001011000 0 | -4 |
| | | 1 1 110100111 0 | 4 |
| 53 | 10101100 | 1 0 010100111 0 | 0 |
| 54 | 01101100 | 1 0 100100111 0 | 0 |
| 55 | 11101100 | 1 1 000100111 0 | 0 |
| 56 | 00011100 | 1 0 000111000 0 | -4 |
| | | 1 1 111000111 0 | 4 |
| 57 | 10011100 | 1 0 011000111 0 | 0 |
| 58 | 01011100 | 1 0 101000111 0 | 0 |
| 59 | 11011100 | 1 1 001000111 0 | 0 |
| 60 | 00111100 | 1 0 110000111 0 | 0 |
| 61 | 10111100 | 1 1 010000111 0 | 0 |
| 62 | 01111100 | 1 1 100000111 0 | 0 |
| 63 | 11111100 | 1 1 111111000 0 | 4 |
| | | 1 0 000000111 0 | -4 |
| 64 | 00000010 | 1 0 000000100 0 | -8 |
| | | 1 1 111111011 0 | 8 |
| 65 | 10000010 | 1 1 100000100 0 | -4 |
| | | 1 0 011111011 0 | 4 |
| 66 | 01000010 | 1 1 010000100 0 | -4 |
| | | 1 0 101111011 0 | 4 |
| 67 | 11000010 | 1 0 110000100 0 | -4 |
| | | 1 1 001111011 0 | 4 |
| 68 | 00100010 | 1 1 001000100 0 | -4 |
| | | 1 0 110111011 0 | 4 |
| 69 | 10100010 | 1 0 101000100 0 | -4 |
| | | 1 1 010111011 0 | 4 |
| 70 | 01100010 | 1 0 011000100 0 | -4 |
| | | 1 1 100111011 0 | 4 |
| 71 | 11100010 | 1 0 000111011 0 | 0 |
| 72 | 00010010 | 1 1 000100100 0 | -4 |
| | | 1 0 111011011 0 | 4 |
| 73 | 10010010 | 1 0 100100100 0 | -4 |
| | | 1 1 011011011 0 | 4 |
| 74 | 01010010 | 1 0 010100100 0 | -4 |
| | | 1 1 101011011 0 | 4 |
| 75 | 11010010 | 1 0 001011011 0 | 0 |
| 76 | 00110010 | 1 0 001100100 0 | -4 |
| | | 1 1 110011011 0 | 4 |
| 77 | 10110010 | 1 0 010011011 0 | 0 |
| 78 | 01110010 | 1 0 100011011 0 | 0 |
| 79 | 11110010 | 1 1 000011011 0 | 0 |
| 80 | 00001010 | 1 1 000010100 0 | -4 |
| | | 1 0 111101011 0 | 4 |
| 81 | 10001010 | 1 0 100010100 0 | -4 |
| | | 1 1 011101011 0 | 4 |
| 82 | 01001010 | 1 0 010010100 0 | -4 |
| | | 1 1 101101011 0 | 4 |
| 83 | 11001010 | 1 0 001101011 0 | 0 |
| 84 | 00101010 | 1 0 001010100 0 | -4 |
| | | 1 1 110101011 0 | 4 |
| 85 | 10101010 | 1 0 010101011 0 | 0 |
| 86 | 01101010 | 1 0 100101011 0 | 0 |
| 87 | 11101010 | 1 1 000101011 0 | 0 |
| 88 | 00011010 | 1 0 000110100 0 | -4 |
| | | 1 1 111001011 0 | 4 |
| 89 | 10011010 | 1 0 011001011 0 | 0 |
| 90 | 01011010 | 1 0 101001011 0 | 0 |
| 91 | 11011010 | 1 1 001001011 0 | 0 |
| 92 | 00111010 | 1 0 110001011 0 | 0 |
| 93 | 10111010 | 1 1 010001011 0 | 0 |
| 94 | 01111010 | 1 1 100001011 0 | 0 |
| 95 | 11111010 | 1 1 111110100 0 | 4 |
| | | 1 0 000001011 0 | -4 |
| 96 | 00000110 | 1 1 000001100 0 | -4 |
| | | 1 0 111110011 0 | 4 |
| 97 | 10000110 | 1 0 100001100 0 | -4 |
| | | 1 1 011110011 0 | 4 |
| 98 | 01000110 | 1 0 010001100 0 | -4 |
| | | 1 1 101110011 0 | 4 |
| 99 | 11000110 | 1 0 001110011 0 | 0 |
| 100 | 00100110 | 1 0 001001100 0 | -4 |
| | | 1 1 110110011 0 | 4 |
| 101 | 10100110 | 1 0 010110011 0 | 0 |
| 102 | 01100110 | 1 0 100110011 0 | 0 |
| 103 | 11100110 | 1 1 000110011 0 | 0 |
| 104 | 00010110 | 1 0 000101100 0 | -4 |
| | | 1 1 111010011 0 | 4 |
| 105 | 10010110 | 1 0 011010011 0 | 0 |
| 106 | 01010110 | 1 0 101010011 0 | 0 |
| 107 | 11010110 | 1 1 001010011 0 | 0 |
| 108 | 00110110 | 1 0 110010011 0 | 0 |
| 109 | 10110110 | 1 1 010010011 0 | 0 |
| 110 | 01110110 | 1 1 100010011 0 | 0 |
| 111 | 11110110 | 1 1 111101100 0 | 4 |
| | | 1 0 000010011 0 | -4 |
| 112 | 00001110 | 1 0 000011100 0 | -4 |
| | | 1 1 111100011 0 | 4 |
| 113 | 10001110 | 1 0 011100011 0 | 0 |
| 114 | 01001110 | 1 0 101100011 0 | 0 |
| 115 | 11001110 | 1 1 001100011 0 | 0 |
| 116 | 00101110 | 1 0 110100011 0 | 0 |
| 117 | 10101110 | 1 1 010100011 0 | 0 |
| 118 | 01101110 | 1 1 100100011 0 | 0 |
| 119 | 11101110 | 1 1 111011100 0 | 4 |
| | | 1 0 000100011 0 | -4 |
| 120 | 00011110 | 1 0 111000011 0 | 0 |
| 121 | 10011110 | 1 1 011000011 0 | 0 |
| 122 | 01011110 | 1 1 101000011 0 | 0 |
| 123 | 11011110 | 1 1 110111100 0 | 4 |
| | | 1 0 001000011 0 | -4 |
| 124 | 00111110 | 1 1 110000011 0 | 0 |
| 125 | 10111110 | 1 1 101111100 0 | 4 |
| | | 1 0 010000011 0 | -4 |
| 126 | 01111110 | 1 1 011111100 0 | 4 |
| | | 1 0 100000011 0 | -4 |
| 127 | 11111110 | 1 0 111111100 0 | 4 |

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

| | | | |
|---|---|---|---|
| | | 1 1 0000000011 0 | -4 |
| 128 | 00000001 | 1 0 0000000010 0 | -8 |
| | | 1 1 1111111101 0 | 8 |
| 129 | 10000001 | 1 1 1000000010 0 | -4 |
| | | 1 0 0111111101 0 | 4 |
| 130 | 01000001 | 1 1 0100000010 0 | -4 |
| | | 1 0 1011111101 0 | 4 |
| 131 | 11000001 | 1 0 1100000010 0 | -4 |
| | | 1 1 0011111101 0 | 4 |
| 132 | 00100001 | 1 1 0010000010 0 | -4 |
| | | 1 0 1101111101 0 | 4 |
| 133 | 10100001 | 1 0 1010000010 0 | -4 |
| | | 1 1 0101111101 0 | 4 |
| 134 | 01100001 | 1 0 0110000010 0 | -4 |
| | | 1 1 1001111101 0 | 4 |
| 135 | 11100001 | 1 0 0001111101 0 | 0 |
| 136 | 00010001 | 1 1 0001000010 0 | -4 |
| | | 1 0 1110111101 0 | 4 |
| 137 | 10010001 | 1 0 1001000010 0 | -4 |
| | | 1 1 0110111101 0 | 4 |
| 138 | 01010001 | 1 0 0101000010 0 | -4 |
| | | 1 1 1010111101 0 | 4 |
| 139 | 11010001 | 1 0 0010111101 0 | 0 |
| 140 | 00110001 | 1 0 0011000010 0 | -4 |
| | | 1 1 1100111101 0 | 4 |
| 141 | 10110001 | 1 0 0100111101 0 | 0 |
| 142 | 01110001 | 1 0 1000111101 0 | 0 |
| 143 | 11110001 | 1 1 0000111101 0 | 0 |
| 144 | 00001001 | 1 1 0000100010 0 | -4 |
| | | 1 0 1111011101 0 | 4 |
| 145 | 10001001 | 1 0 1000100010 0 | -4 |
| | | 1 1 0111011101 0 | 4 |
| 146 | 01001001 | 1 0 0100100010 0 | -4 |
| | | 1 1 1011011101 0 | 4 |
| 147 | 11001001 | 1 0 0011011101 0 | 0 |
| 148 | 00101001 | 1 0 0010100010 0 | -4 |
| | | 1 1 1101011101 0 | 4 |
| 149 | 10101001 | 1 0 0101011101 0 | 0 |
| 150 | 01101001 | 1 0 1001011101 0 | 0 |
| 151 | 11101001 | 1 1 0001011101 0 | 0 |
| 152 | 00011001 | 1 0 0001100010 0 | -4 |
| | | 1 1 1110011101 0 | 4 |
| 153 | 10011001 | 1 0 0110011101 0 | 0 |
| 154 | 01011001 | 1 0 1010011101 0 | 0 |
| 155 | 11011001 | 1 1 0010011101 0 | 0 |
| 156 | 00111001 | 1 0 1100011101 0 | 0 |
| 157 | 10111001 | 1 1 0100011101 0 | 0 |
| 158 | 01111001 | 1 1 1000011101 0 | 0 |
| 159 | 11111001 | 1 1 1111100010 0 | 4 |
| | | 1 0 0000011101 0 | -4 |
| 160 | 00000101 | 1 1 0000010100 | -4 |
| | | 1 0 1111101010 | 4 |
| 161 | 10000101 | 1 0 1000010100 | -4 |
| | | 1 1 0111101010 | 4 |
| 162 | 01000101 | 1 0 0100010100 | -4 |
| | | 1 1 1011101010 | 4 |
| 163 | 11000101 | 1 0 0011101010 | 0 |
| 164 | 00100101 | 1 0 0010010100 | -4 |
| | | 1 1 1101101010 | 4 |
| 165 | 10100101 | 1 0 0101101010 | 0 |
| 166 | 01100101 | 1 0 1001101010 | 0 |
| 167 | 11100101 | 1 1 0001101010 | 0 |
| 168 | 00010101 | 1 0 0001010100 | -4 |
| | | 1 1 1110101010 | 4 |
| 169 | 10010101 | 1 0 0110101010 | 0 |
| 170 | 01010101 | 1 0 1010101010 | 0 |
| 171 | 11010101 | 1 1 0010101010 | 0 |
| 172 | 00110101 | 1 0 1100101010 | 0 |
| 173 | 10110101 | 1 1 0100101010 | 0 |
| 174 | 01110101 | 1 1 1000101010 | 0 |
| 175 | 11110101 | 1 1 1111010100 | 4 |
| | | 1 0 0000101010 | -4 |
| 176 | 00001101 | 1 0 0000110100 | -4 |
| | | 1 1 1111001010 | 4 |
| 177 | 10001101 | 1 0 0111001010 | 0 |
| 178 | 01001101 | 1 0 1011001010 | 0 |
| 179 | 11001101 | 1 1 0011001010 | 0 |
| 180 | 00101101 | 1 0 1101001010 | 0 |
| 181 | 10101101 | 1 1 0101001010 | 0 |
| 182 | 01101101 | 1 1 1001001010 | 0 |
| 183 | 11101101 | 1 1 1110110100 | 4 |
| | | 1 0 0001001010 | -4 |
| 184 | 00011101 | 1 0 1110001010 | 0 |
| 185 | 10011101 | 1 1 0110001010 | 0 |
| 186 | 01011101 | 1 1 1010001010 | 0 |
| 187 | 11011101 | 1 1 1101110100 | 4 |
| | | 1 0 0010001010 | -4 |
| 188 | 00111101 | 1 1 1100001010 | 0 |
| 189 | 10111101 | 1 1 1011110100 | 4 |
| | | 1 0 0100001010 | -4 |
| 190 | 01111101 | 1 1 0111110100 | 4 |
| | | 1 0 1000001010 | -4 |
| 191 | 11111101 | 1 0 1111110100 | 4 |
| | | 1 1 0000001010 | -4 |
| 192 | 00000011 | 1 1 0000001100 | -4 |
| | | 1 0 1111110010 | 4 |
| 193 | 10000011 | 1 0 1000001100 | -4 |
| | | 1 1 0111110010 | 4 |
| 194 | 01000011 | 1 0 0100001100 | -4 |
| | | 1 1 1011110010 | 4 |
| 195 | 11000011 | 1 0 0011110010 | 0 |
| 196 | 00100011 | 1 0 0010001100 | -4 |
| | | 1 1 1101110010 | 4 |
| 197 | 10100011 | 1 0 0101110010 | 0 |
| 198 | 01100011 | 1 0 1001110010 | 0 |
| 199 | 11100011 | 1 1 0001110010 | 0 |
| 200 | 00010011 | 1 0 0001001100 | -4 |
| | | 1 1 1110110010 | 4 |
| 201 | 10010011 | 1 0 0110110010 | 0 |
| 202 | 01010011 | 1 0 1010110010 | 0 |
| 203 | 11010011 | 1 1 0010110010 | 0 |
| 204 | 00110011 | 1 0 1100110010 | 0 |
| 205 | 10110011 | 1 1 0100110010 | 0 |
| 206 | 01110011 | 1 1 1000110010 | 0 |
| 207 | 11110011 | 1 1 1111001100 | 4 |
| | | 1 0 0000110010 | -4 |
| 208 | 00001011 | 1 0 0000101100 | -4 |
| | | 1 1 1111010010 | 4 |
| 209 | 10001011 | 1 0 0111010010 | 0 |
| 210 | 01001011 | 1 0 1011010010 | 0 |
| 211 | 11001011 | 1 1 0011010010 | 0 |
| 212 | 00101011 | 1 0 1101010010 | 0 |
| 213 | 10101011 | 1 1 0101010010 | 0 |
| 214 | 01101011 | 1 1 1001010010 | 0 |
| 215 | 11101011 | 1 1 1110101100 | 4 |
| | | 1 0 0001010010 | -4 |
| 216 | 00011011 | 1 0 1110010010 | 0 |
| 217 | 10011011 | 1 1 0110010010 | 0 |
| 218 | 01011011 | 1 1 1010010010 | 0 |
| 219 | 11011011 | 1 1 1101101100 | 4 |
| | | 1 0 0010010010 | -4 |
| 220 | 00111011 | 1 1 1100010010 | 0 |
| 221 | 10111011 | 1 1 1011101100 | 4 |
| | | 1 0 0100010010 | -4 |
| 222 | 01111011 | 1 1 0111101100 | 4 |
| | | 1 0 1000010010 | -4 |
| 223 | 11111011 | 1 0 1111101100 | 4 |
| | | 1 1 0000010010 | -4 |
| 224 | 00000111 | 1 0 0000011100 | -4 |
| | | 1 1 1111100010 | 4 |
| 225 | 10000111 | 1 0 0111100010 | 0 |
| 226 | 01000111 | 1 0 1011100010 | 0 |
| 227 | 11000111 | 1 1 0011100010 | 0 |
| 228 | 00100111 | 1 0 1101100010 | 0 |
| 229 | 10100111 | 1 1 0101100010 | 0 |
| 230 | 01100111 | 1 1 1001100010 | 0 |
| 231 | 11100111 | 1 1 1110011100 | 4 |
| | | 1 0 0001100010 | -4 |
| 232 | 00010111 | 1 0 1110100010 | 0 |
| 233 | 10010111 | 1 1 0110100010 | 0 |
| 234 | 01010111 | 1 1 1010100010 | 0 |
| 235 | 11010111 | 1 1 1101011100 | 4 |
| | | 1 0 0010100010 | -4 |
| 236 | 00110111 | 1 1 1100100010 | 0 |
| 237 | 10110111 | 1 1 1011011100 | 4 |
| | | 1 0 0100100010 | -4 |
| 238 | 01110111 | 1 1 0111011100 | 4 |
| | | 1 0 1000100010 | -4 |
| 239 | 11110111 | 1 0 1110111100 | 4 |
| | | 1 1 0000100010 | -4 |
| 240 | 00001111 | 1 0 1111000010 | 0 |
| 241 | 10001111 | 1 1 0111000010 | 0 |
| 242 | 01001111 | 1 1 1011000010 | 0 |
| 243 | 11001111 | 1 1 1100111100 | 4 |
| | | 1 0 0011000010 | -4 |
| 244 | 00101111 | 1 1 1101000010 | 0 |
| 245 | 10101111 | 1 1 1010111100 | 4 |
| | | 1 0 0101000010 | -4 |
| 246 | 01101111 | 1 1 0110111100 | 4 |
| | | 1 0 1001000010 | -4 |
| 247 | 11101111 | 1 0 1110111100 | 4 |
| | | 1 1 0001000010 | -4 |
| 248 | 00011111 | 1 1 1110000010 | 0 |
| 249 | 10011111 | 1 1 1001111100 | 4 |
| | | 1 0 0110000010 | -4 |
| 250 | 01011111 | 1 1 0101111100 | 4 |
| | | 1 0 1010000010 | -4 |
| 251 | 11011111 | 1 0 1101111100 | 4 |
| | | 1 1 0010000010 | -4 |
| 252 | 00111111 | 1 1 0011111100 | 4 |
| | | 1 0 1100000010 | -4 |
| 253 | 10111111 | 1 0 1011111100 | 4 |
| | | 1 1 0100000010 | -4 |
| 254 | 01111111 | 1 0 0111111100 | 4 |
| | | 1 1 1000000010 | -4 |
| 255 | 11111111 | 1 1 1111111100 | 8 |
| | | 1 0 0000000010 | -8 |

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

**Table 8: Control Characters**

| ID | Dec | Bin, d[0:7] | S-AP-e[0:7]-I-S |
|---|---|---|---|
| | | | |
| 0 | 15 | 11110000 | 1 1 11110000 0 0 |
| 1 | 23 | 11101000 | 1 1 11101000 0 0 |
| 2 | 27 | 11011000 | 1 1 11011000 0 0 |
| 3 | 29 | 10111000 | 1 1 10111000 0 0 |
| 4 | 30 | 01111000 | 1 1 01111000 0 0 |
| 5 | 31 | 11111000 | 1 0 11111000 0 0 |
| 6 | 39 | 11100100 | 1 1 11100100 0 0 |
| 7 | 43 | 11010100 | 1 1 11010100 0 0 |
| 8 | 45 | 10110100 | 1 1 10110100 0 0 |
| 9 | 46 | 01110100 | 1 1 01110100 0 0 |
| 10 | 47 | 11110100 | 1 0 11110100 0 0 |
| 11 | 51 | 11001100 | 1 1 11001100 0 0 |
| 12 | 53 | 10101100 | 1 1 10101100 0 0 |
| 13 | 54 | 01101100 | 1 1 01101100 0 0 |
| 14 | 55 | 11101100 | 1 0 11101100 0 0 |
| 15 | 57 | 10011100 | 1 1 10011100 0 0 |
| 16 | 58 | 01011100 | 1 1 01011100 0 0 |
| 17 | 59 | 11011100 | 1 0 11011100 0 0 |
| 18 | 60 | 00111100 | 1 1 00111100 0 0 |
| 19 | 61 | 10111100 | 1 0 10111100 0 0 |
| 20 | 62 | 01111100 | 1 0 01111100 0 0 |
| 21 | 71 | 11100010 | 1 1 11100010 0 0 |
| 22 | 75 | 11010010 | 1 1 11010010 0 0 |
| 23 | 77 | 10110010 | 1 1 10110010 0 0 |
| 24 | 78 | 01110010 | 1 1 01110010 0 0 |
| 25 | 79 | 11110010 | 1 0 11110010 0 0 |
| 26 | 83 | 11001010 | 1 1 11001010 0 0 |
| 27 | 85 | 10101010 | 1 1 10101010 0 0 |
| 28 | 86 | 01101010 | 1 1 01101010 0 0 |
| 29 | 87 | 11101010 | 1 0 11101010 0 0 |
| 30 | 89 | 10011010 | 1 1 10011010 0 0 |
| 31 | 90 | 01011010 | 1 1 01011010 0 0 |
| 32 | 91 | 11011010 | 1 0 11011010 0 0 |
| 33 | 92 | 00111010 | 1 1 00111010 0 0 |
| 34 | 93 | 10111010 | 1 0 10111010 0 0 |
| 35 | 94 | 01111010 | 1 0 01111010 0 0 |
| 36 | 99 | 11000110 | 1 1 11000110 0 0 |
| 37 | 101 | 10100110 | 1 1 10100110 0 0 |
| 38 | 102 | 01100110 | 1 1 01100110 0 0 |
| 39 | 103 | 11100110 | 1 0 11100110 0 0 |
| 40 | 105 | 10010110 | 1 1 10010110 0 0 |
| 41 | 106 | 01010110 | 1 1 01010110 0 0 |
| 42 | 107 | 11010110 | 1 0 11010110 0 0 |
| 43 | 108 | 00110110 | 1 1 00110110 0 0 |
| 44 | 109 | 10110110 | 1 0 10110110 0 0 |
| 45 | 110 | 01110110 | 1 0 01110110 0 0 |
| 46 | 113 | 10001110 | 1 1 10001110 0 0 |
| 47 | 114 | 01001110 | 1 1 01001110 0 0 |
| 48 | 115 | 11001110 | 1 0 11001110 0 0 |
| 49 | 116 | 00101110 | 1 1 00101110 0 0 |
| 50 | 117 | 10101110 | 1 0 10101110 0 0 |
| 51 | 118 | 01101110 | 1 0 01101110 0 0 |
| 52 | 120 | 00011110 | 1 1 00011110 0 0 |
| 53 | 121 | 10011110 | 1 0 10011110 0 0 |
| 54 | 122 | 01011110 | 1 0 01011110 0 0 |
| 55 | 124 | 00111110 | 1 0 00111110 0 0 |
| 56 | 135 | 11100001 | 1 1 11100001 0 0 |
| 57 | 139 | 11010001 | 1 1 11010001 0 0 |
| 58 | 141 | 10110001 | 1 1 10110001 0 0 |
| 59 | 142 | 01110001 | 1 1 01110001 0 0 |
| 60 | 143 | 11110001 | 1 0 11110001 0 0 |
| 61 | 147 | 11001001 | 1 1 11001001 0 0 |
| 62 | 149 | 10101001 | 1 1 10101001 0 0 |
| 63 | 150 | 01101001 | 1 1 01101001 0 0 |
| 64 | 151 | 11101001 | 1 0 11101001 0 0 |
| 65 | 153 | 10011001 | 1 1 10011001 0 0 |
| 66 | 154 | 01011001 | 1 1 01011001 0 0 |
| 67 | 155 | 11011001 | 1 0 11011001 0 0 |
| 68 | 156 | 00111001 | 1 1 00111001 0 0 |
| 69 | 157 | 10111001 | 1 0 10111001 0 0 |
| 70 | 158 | 01111001 | 1 0 01111001 0 0 |
| 71 | 163 | 11000101 | 1 1 11000101 0 0 |
| 72 | 165 | 10100101 | 1 1 10100101 0 0 |
| 73 | 166 | 01100101 | 1 1 01100101 0 0 |
| 74 | 167 | 11100101 | 1 0 11100101 0 0 |
| 75 | 169 | 10010101 | 1 1 10010101 0 0 |
| 76 | 170 | 01010101 | 1 1 01010101 0 0 |
| 77 | 171 | 11010101 | 1 0 11010101 0 0 |
| 78 | 172 | 00110101 | 1 1 00110101 0 0 |
| 79 | 173 | 10110101 | 1 0 10110101 0 0 |
| 80 | 174 | 01110101 | 1 0 01110101 0 0 |
| 81 | 177 | 10001101 | 1 1 10001101 0 0 |
| 82 | 178 | 01001101 | 1 1 01001101 0 0 |
| 83 | 179 | 11001101 | 1 0 11001101 0 0 |
| 84 | 180 | 00101101 | 1 1 00101101 0 0 |
| 85 | 181 | 10101101 | 1 0 10101101 0 0 |
| 86 | 182 | 01101101 | 1 0 01101101 0 0 |
| 87 | 184 | 00011101 | 1 1 00011101 0 0 |
| 88 | 185 | 10011101 | 1 0 10011101 0 0 |
| 89 | 186 | 01011101 | 1 0 01011101 0 0 |
| 90 | 188 | 00111101 | 1 0 00111101 0 0 |
| 91 | 195 | 11000011 | 1 1 11000011 0 0 |
| 92 | 197 | 10100011 | 1 1 10100011 0 0 |
| 93 | 198 | 01100011 | 1 1 01100011 0 0 |
| 94 | 199 | 11100011 | 1 0 11100011 0 0 |
| 95 | 201 | 10010011 | 1 1 10010011 0 0 |
| 96 | 202 | 01010011 | 1 1 01010011 0 0 |
| 97 | 203 | 11010011 | 1 0 11010011 0 0 |
| 98 | 204 | 00110011 | 1 1 00110011 0 0 |
| 99 | 205 | 10110011 | 1 0 10110011 0 0 |
| 100 | 206 | 01110011 | 1 0 01110011 0 0 |
| 101 | 209 | 10001011 | 1 1 10001011 0 0 |
| 102 | 210 | 01001011 | 1 1 01001011 0 0 |
| 103 | 211 | 11001011 | 1 0 11001011 0 0 |
| 104 | 212 | 00101011 | 1 1 00101011 0 0 |
| 105 | 213 | 10101011 | 1 0 10101011 0 0 |
| 106 | 214 | 01101011 | 1 0 01101011 0 0 |
| 107 | 216 | 00011011 | 1 1 00011011 0 0 |
| 108 | 217 | 10011011 | 1 0 10011011 0 0 |
| 109 | 218 | 01011011 | 1 0 01011011 0 0 |
| 110 | 220 | 00111011 | 1 0 00111011 0 0 |
| 111 | 225 | 10000111 | 1 1 10000111 0 0 |
| 112 | 226 | 01000111 | 1 1 01000111 0 0 |
| 113 | 227 | 11000111 | 1 0 11000111 0 0 |
| 114 | 228 | 00100111 | 1 1 00100111 0 0 |
| 115 | 229 | 10100111 | 1 0 10100111 0 0 |
| 116 | 230 | 01100111 | 1 0 01100111 0 0 |
| 117 | 232 | 00010111 | 1 1 00010111 0 0 |
| 118 | 233 | 10010111 | 1 0 10010111 0 0 |
| 119 | 234 | 01010111 | 1 0 01010111 0 0 |
| 120 | 236 | 00110111 | 1 0 00110111 0 0 |
| 121 | 240 | 00001111 | 1 1 00001111 0 0 |
| 122 | 241 | 10001111 | 1 0 10001111 0 0 |
| 123 | 242 | 01001111 | 1 0 01001111 0 0 |
| 124 | 244 | 00101111 | 1 0 00101111 0 0 |
| 125 | 248 | 00011111 | 1 0 00011111 0 0 |

1
2

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

## 7.4. Exchange Level

### 7.4.1. General

All HS Links shall be bidirectional.

The EL shall perform the following functions that correspond to normal operation of a bidirectional link:

a. Start-up
b. Shutdown

The EL shall ([Editor's note: some of these function may be classified under 'should']) perform the following additional functions:

a. Flow control
b. Parity error detection and recovery
c. Receiver calibration loss and recovery
d. Disconnection detection and recovery
e. Reset

All EL functions remain invisible to the PL during normal operation.

The EL uses the reserved l_chars detailed in Table 9.

Table 9: Reserved l_chars for EL functions

| Symbol | Control character ID | Name |
|---|---|---|
| | | |
| IDLE | 0 | Idle |
| START_REQ | 5 | Start request |
| START_ACK | 1 | Start acknowledge |
| STOP_REQ | 2 | Stop request |
| STOP_ACK | 3 | Stop acknowledge |
| STOP_NACK | 4 | Stop negative acknowledge |
| FCC | 125 | Flow control |
| RESET | 6 | Reset |

Consider two connected *nodes*, NODE_A and NODE_B. (See figure 20.) Each node uses a *port* or *link interface* (PORT_A and PORT_B respectively) in order to communicate with the other node. Note that a node may have more than one port. Each port is divided into two parts: a transmitter and a receiver, refered to as transmitter_A and receiver_A for NODE_A and transmitter_B and receiver_B for NODE_B.

Each port has a n_char source (SOURCE_A and SOURCE_B) which supply n_chars to transmitter_A and transmitter_B respectively and a n_char sink (SINK_A and SINK_B) which accept data from receiver_A and receiver_B respectively.

Exchange level information (i.e. the l_chars detailed in Table 9) generated and sent by, for example, transmitter_A are received and filtered by the receiver_B and passed directly to transmitter_B (and vice-versa). L_chars shall not be generated by the source nor written into the sink.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Each receiver has a flag, CAL, which indicates its state. When CAL is low, this indicates that the receiver is not calibrated to the incoming serial data stream. When in this state, all characters are filtered by the receiver. When CAL is high, this means that the receiver is calibrated to the incoming serial data stream. When in this state, all n_chars are passed to the n_char sink and all l_chars are passed to the port's transmitter.

Each transmitter and receiver is controlled by a state machine (See Figure 21, Figure 22, Figure 23 and Figure 24.)

### 7.4.2. Start-up

A request to transmit in one direction automatically activates the complete bidirectional link, this being necessary to allow the EL functions. There are two different types of start-up possible: Unidirectional and Bidirectional. In both cases the start-up procedure calibrates and acknowledges the bidirectional link.

The start-up procedure uses the l_char IDLE (control character ID = 0) in order to calibrate the receiver. IDLE has the characteristic that the only positive going edges in the character are the synchronization edges. This enables the clock recovery circuit to correctly lock onto these edges.

#### 7.4.2.1. Unidirectional Start-up

In figure 20, NODE_A requests to transmit to NODE_B. At first the transmitter_A is started and IDLEs are sent to calibrate receiver_B. When receiver_B is calibrated (flag CAL_B goes high), this automatically starts the transmitter_B and IDLEs are sent to calibrate receiver_A. When receiver_A is calibrated, PORT_A sends a start_request (START_REQ) to PORT_B. PORT_B responds by sending a start_acknowledge (START_ACK) to PORT_A.

On receipt of START_ACK, PORT_A replies by sending a START_ACK to PORT_B. PORT_A becomes *functional* (see below). N_chars in the SOURCE_A are transmitted from A to B (if SOURCE_A is empty, IDLEs shall be sent) and flow control information is transmitted from A to B when necessary.

On receipt of START_ACK, PORT_B becomes *functional*. N_chars in the SOURCE_B are transmitted from B to A (if SOURCE_B is empty, IDLEs shall be sent) and flow control information is transmitted from B to A when necessary. Note that PORT_B is functional, even if NODE_B has not been explicitly requested to transmit to NODE_A.

This exchange is shown in Figure 25.

#### 7.4.2.2. Bidirectional Start-up

This is the case in which NODE_A and NODE_B request to transmit quasi-simultaneously (for example, they request to transmit on power-up). Both transmitter_A and transmitter_B send START_REQ and both reply with START_ACK. Both then re-reply to the START_ACK with another START_ACK (since each side does not 'know' that it is a bidirectional start-up); these last 2 START_ACK's have no effect.

This exchange is shown in Figure 26.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

### 7.4.3. Errors on Start-up

#### 7.4.3.1. Loss of START_REQ

In the case of a bidirectional start-up, it is possible that the START_REQ is sent before the opposite receiver is calibrated (this is not possible during an unidirectional start-up). If this is the case, the START_REQ sent will be filtered and therefore lost by the opposite port. Therefore, after a timeout period, $t_{TO1}$, the START_REQ is resent. The value of $t_{TO1}$ is given in Table 10.

#### 7.4.3.2. Destination node not connected or not under power

If a node requests to transmit and the opposite node is not connected or not under power, the tranmitter state machine will block in state Tx_Working_1. If after a timeout period, $t_{TO2}$, the transmitter should return to the Not_Working state. It is then for the port controller to signal the failure of start-up and to re-request to transmit as and when appropriate (these details are outside the scope of this standard).

This functionality is desirable for two reasons:

a. The system is notified that there has been a failure of start-up on the link.
b. Fiber-optic saftey. If the link is on a fiber-optic medium, and the fiber is disconnected, the laser must not continue to indefinitely emit into an open fiber for eye safety reasons.

Table 10: Timeout Values

| Symbol | Unit | Value |
|---|---|---|
| | | |
| $t_{TO1}$ | | TBS |
| $t_{TO2}$ | | TBS |

#### 7.4.3.3. Other Errors

It is possible that other transmission errors (e.g. loss of receiver calibration, parity error, unexpected or no response from opposite node) could interrupt the start-up mechanism. If any of these errors occur during start-up the transmitter should return to the Not_Working state for the reasons stated above.

### 7.4.4. Functional

#### 7.4.4.1. General

When functional, the following information shall be sent across the link in order of decreasing priority:

a. Link characters other than IDLE (primarily FCC i.e. flow control information for the opposite going link)
b. N_chars in the SOURCE, in a flow-controlled manner (see below).
c. IDLEs when there are no SOURCE n_chars to send (the SOURCE is empty) .

All l_chars are filtered by the receiver controller: IDLEs are discarded; other link characters are passed to the transmitter controller which acts appropriately on them. N_chars are written into the SINK.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

#### 7.4.4.2. Flow Control

The n_char sink has a maximum capacity, defined by the physical implementation. N_chars are written into the sink by the receiver and are read from the sink by the application using the link. In order to prevent sink overun a flow control mechanisms should be [Editor's note: 'shall be'?] used:

The source has a 'credit' (FCC_CT) which indicates how many *blocks* (of F n_chars) of data it can transmit without overunning the sink. For every n_char sent, DATA_CT is incremented; when DATA_CT reaches F, the credit is decremented and DATA_CT is reset to 0. When the transmitter has zero credit, it only sends l_chars (primarily IDLEs, but also FCCs i.e. flow control information for the opposite going link). The source's credit is incremented when the tranmistter receives *Flow Control Link Character* (FCC) from the opposite receiver. When a transmitter has one or less credits, it shall record the arrival of at minimum one FCC and it may record the arrival of more than one FCC.

F is set at 32 n_chars and the transmitter has an initial (before start-up) credit of 0.

The sink authorizes the sending of a FCC when it has the space to receive F n_chars. Each time the sink authorizes the sending of a FCC it shall reserve the space corresponding to F n_chars. While the sink has unreserved space available it should continue sending FCCs until all its space is reserved. The first action of each sink after start-up should therefore be to send the number of FCCs corresponding to its unreserved space.

#### 7.4.4.3. Maximum transmission line length and Latency

[Editor's Note: to be moved to Informative Annex].

Using the flow control mechanism described in above, there exists a theoretical maximum line length over which 'continuous transmission' can be obtained in the case where every n_char written into the sink is immediately read out of the sink by the receive application. This maximum length is determined by the size of the sink (which determines the number of FCCs that the receiver sends to the transmitter after start-up and thus the initial transmitter credit).

Considering the general situation in which:

1. the sink size is B n_chars
2. the flow control block size is F n_chars
3. the transmitter sends a n_char every cycle of T ns
4. the length of the transmission line is D meters (at 5ns/meter propagation time)
5. the bidirectional link has a latency parameter of L cycles.

The source thus acquires an initial credit of [int(B/F).F] n_chars and therefore starts to transmit its n_chars. As the receiver receives the data, it writes it into the sink, from which the receive user immediately reads it. When the user has read F n_chars, the sink authorizes the sending of a FCC. The condition for continuous transmission is that the FCC arrives at the transmitter before the latter's credit is consumed. The timings are shown in Figure 27:

1. the last n_char of the first data block (where a data block is F n_chars) is read by the receive user ($L_1$.T + 5 D) ns after it is sent.
2. there is a delay of $L_2$.T ns before the FCC is sent.
3. the FCC arrives at the opposite receiver after ($L_3$.T + 5 D) ns
4. the transmitter's credit is incremented after $L_4$.T ns

The total time is therefore : (LT + 10 D) ns.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

where: $L = L_1 + L_2 + L_3 + L_4$

For continuous transmission, this time has to be inferior to ([int(B/F).F]-F).T.

Therefore $D_{max} = \frac{T}{10}\left\{[int(B/F).F] - F - \left(L_1 + L_2 + L_3 + L_4\right)\right\}$

**7.4.5. Errors when Functional**

**7.4.5.1. Loss of Receiver Calibration**

Loss of receiver calibration on the incoming serial data stream is indicated by the flag CAL passing low. The port whose receiver has lost calibration sends the RESET l_char and its transmitter passes to the Tx_Working_1 state. On receipt of RESET, the opposite port's transmitter passes into the Tx_Working_2 state if it calibrated or into the Tx_Cal_1 state if it is not calibrated. The unidirectional start-up procedure then proceeds as described above.

[Editor's note: open issue on what to do with the flow control]

**7.4.5.2. Parity Error Detected**

[Editor's note: this section is to be completed]

**7.4.5.3. Request Reset**

If a port requests to reset a link, it follows the same procedure as described in *.

**7.4.6. Shutdown**

The bidirectional link is shutdown only when both NODE_A and NODE_B have requested a shutdown. This is done by a request to stop demand from the port. The transmitter sends a stop_request (STOP_REQ). The response to a STOP_REQ is either a stop_acknowledge (STOP_ACK) if the opposite user is ready to stop, or a stop_not_acknowledge (STOP_NACK) if the opposite user is not ready to stop. Once a request to stop has been demanded, any new n_chars placed in the source are not transmitted: the transmitter only sends l_chars (primarily IDLEs and FCCs). If the port receives a STOP_NACK in reply to its request_to_stop, it can recommence transmission by a request to transmit.

When shutdown, the transmitter is in the Not_Working state and the serial output is static.

**7.4.7. Built-in Loopback Test**

[Editor's note: this section is to be completed]

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

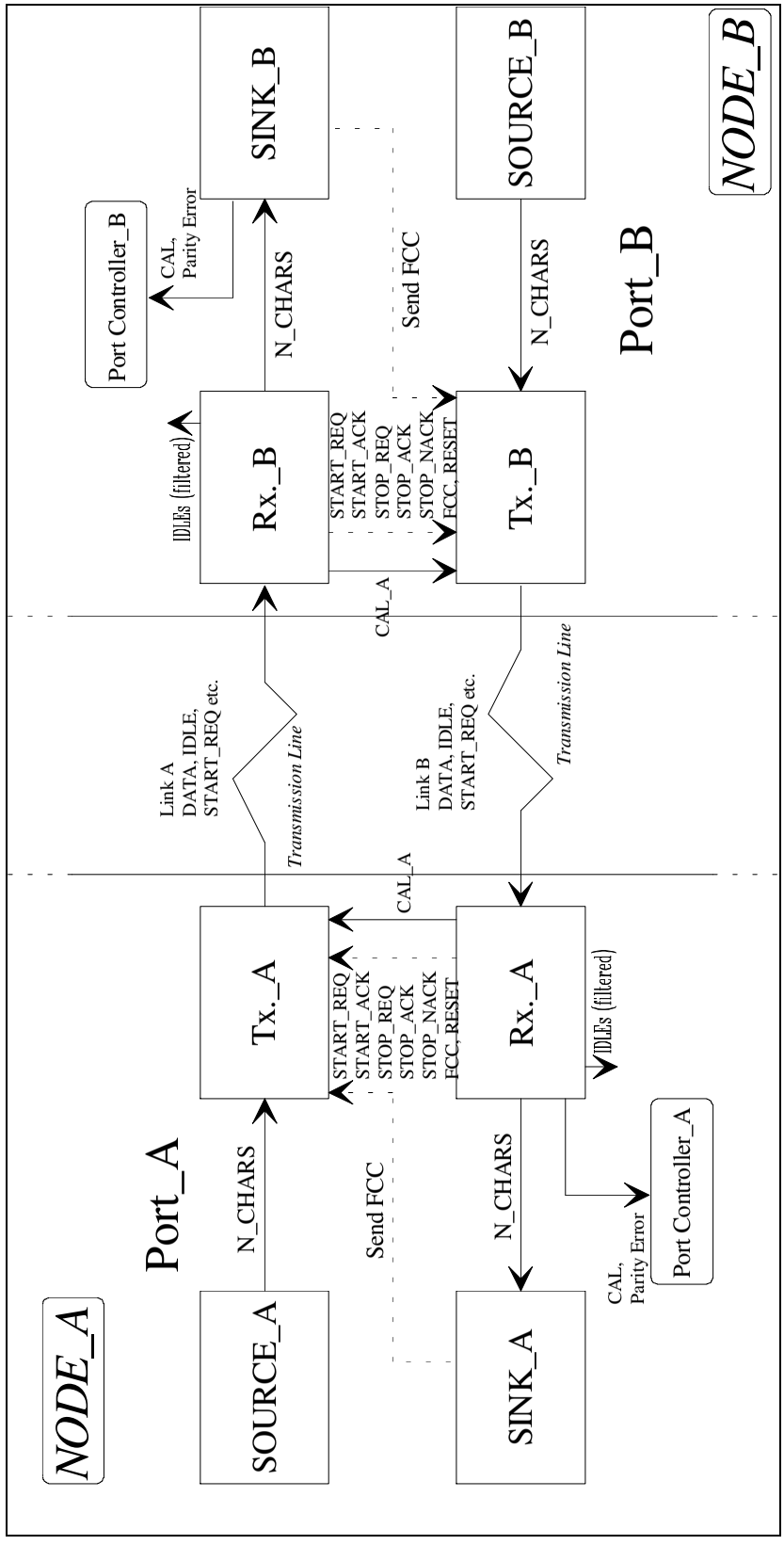


Figure 20: Exchange Level interconnection between two nodes

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

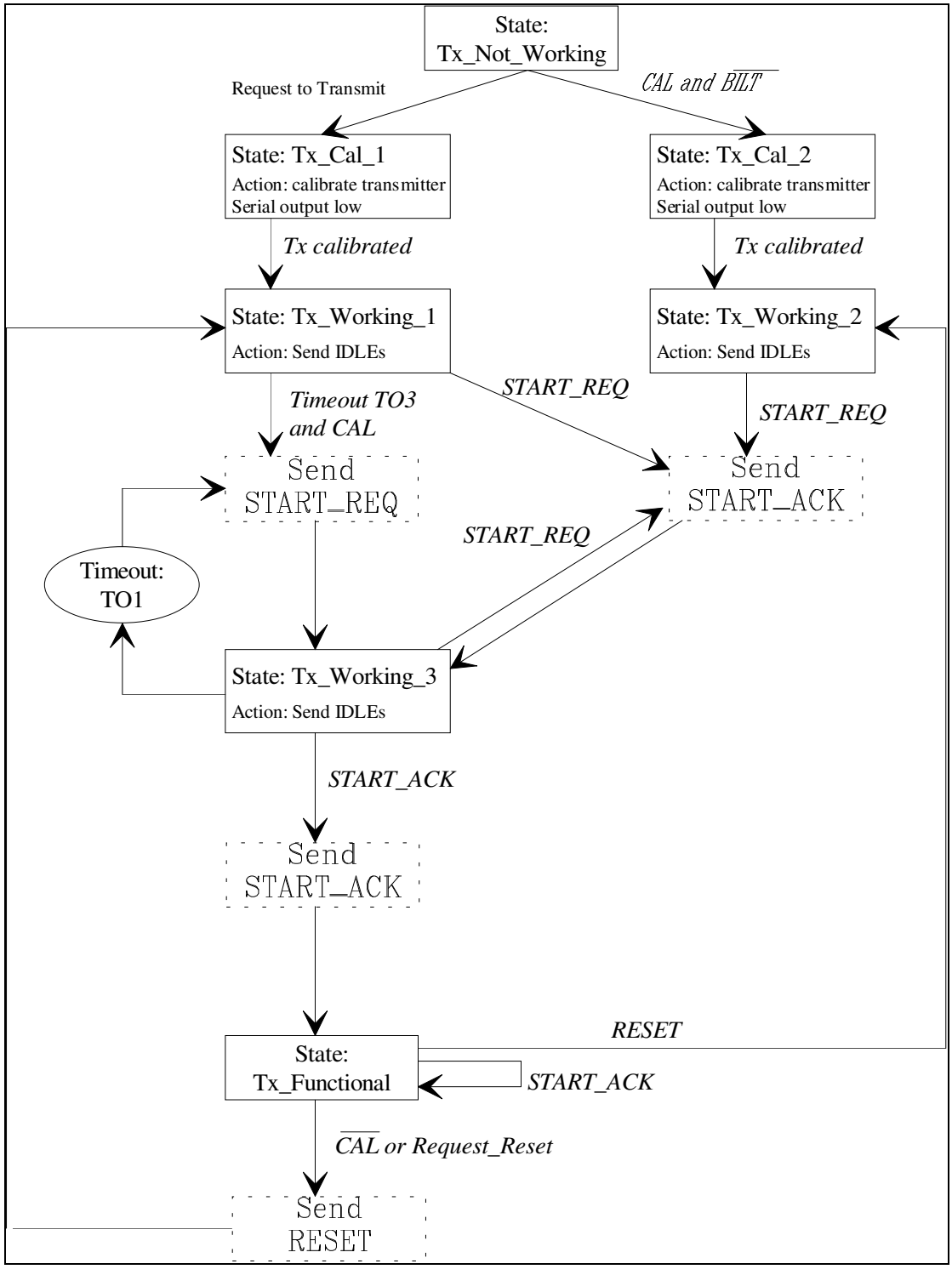


Figure 21: Transmitter State Machine Controller - Start-up

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

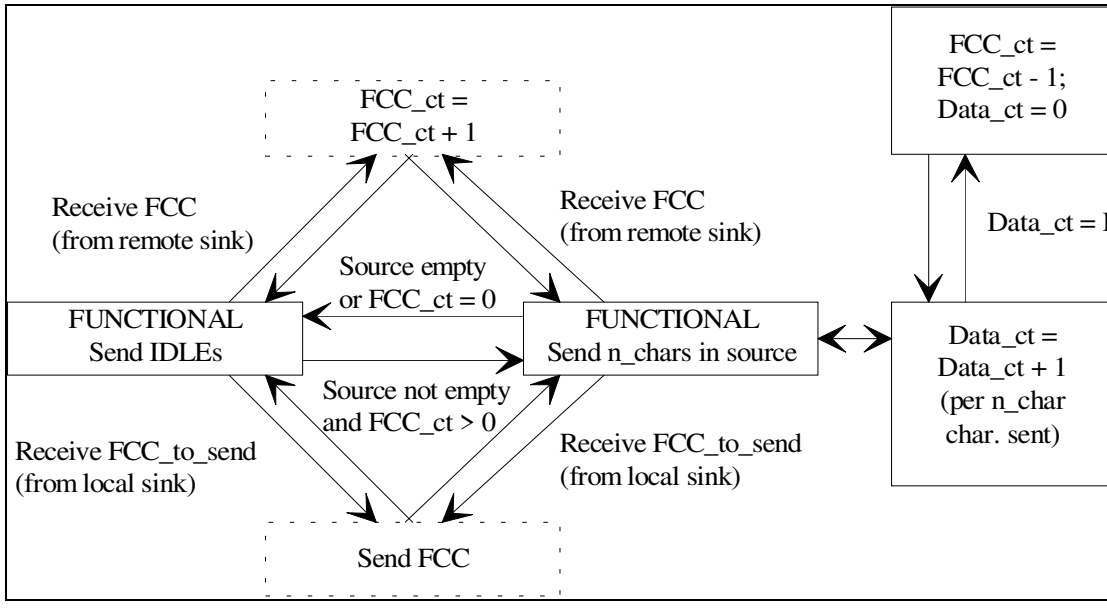


Figure 22: Transmitter State Machine Controller - Functional

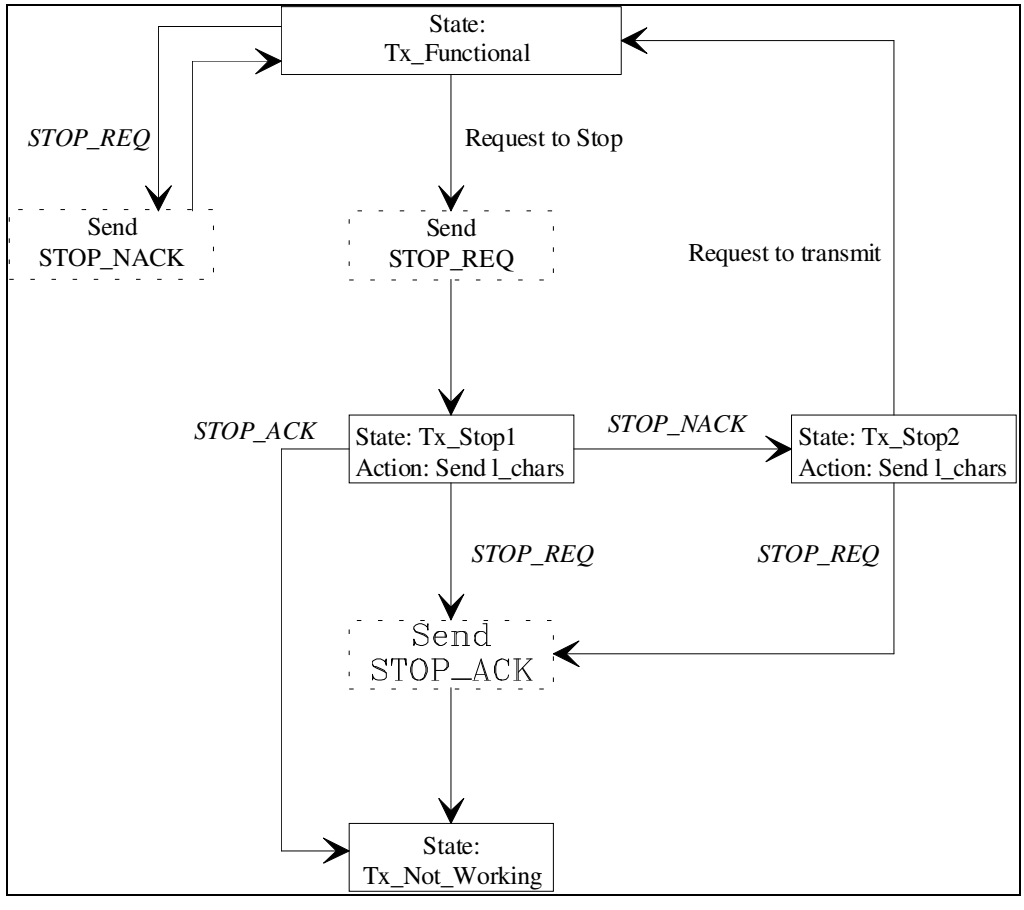


Figure 23: Transmitter State Machine Controller - Shutdown

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

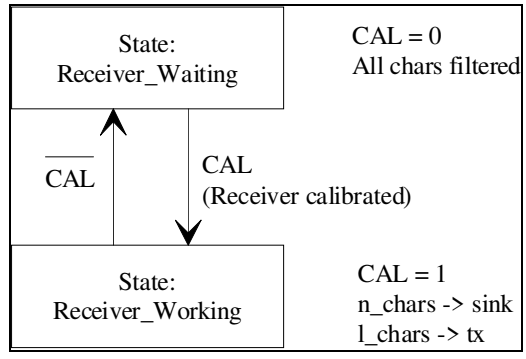


Figure 24: Receiver State Machine Controller

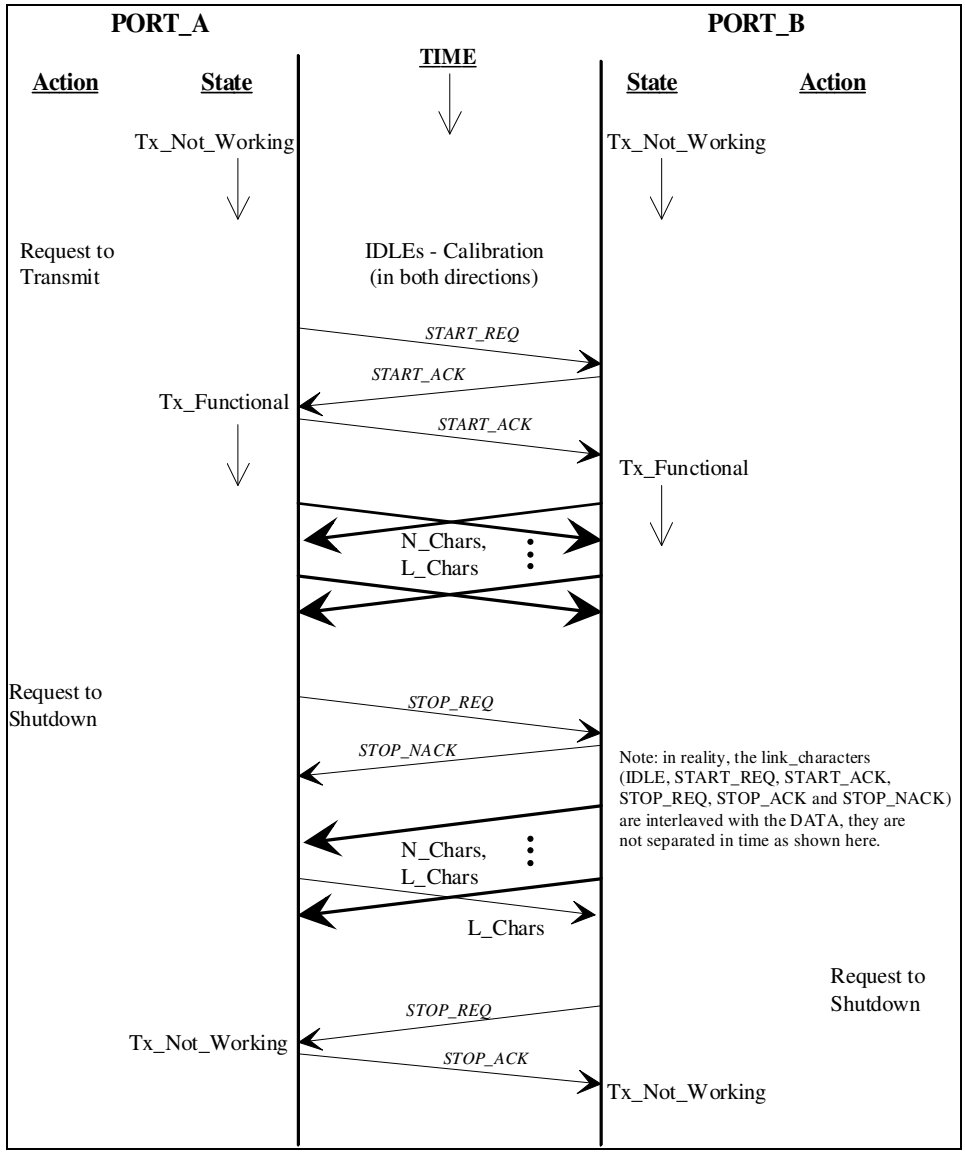


Figure 25: Exchange for Start-up, Functional and Shutdown

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

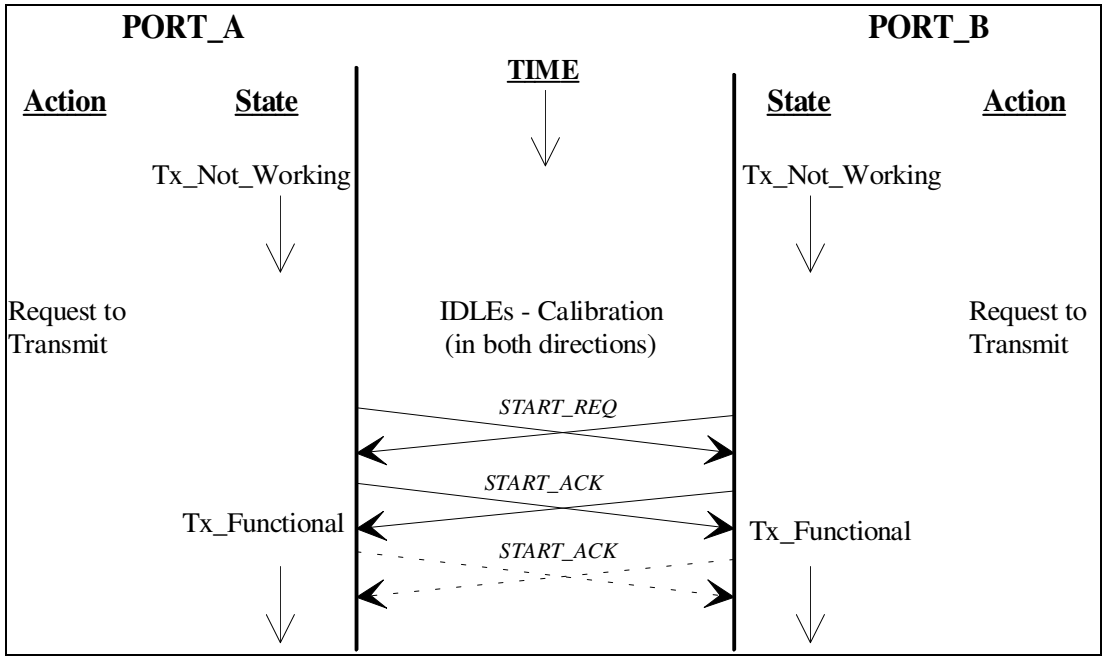


Figure 26: Exchange for Bidirectional Start-Up

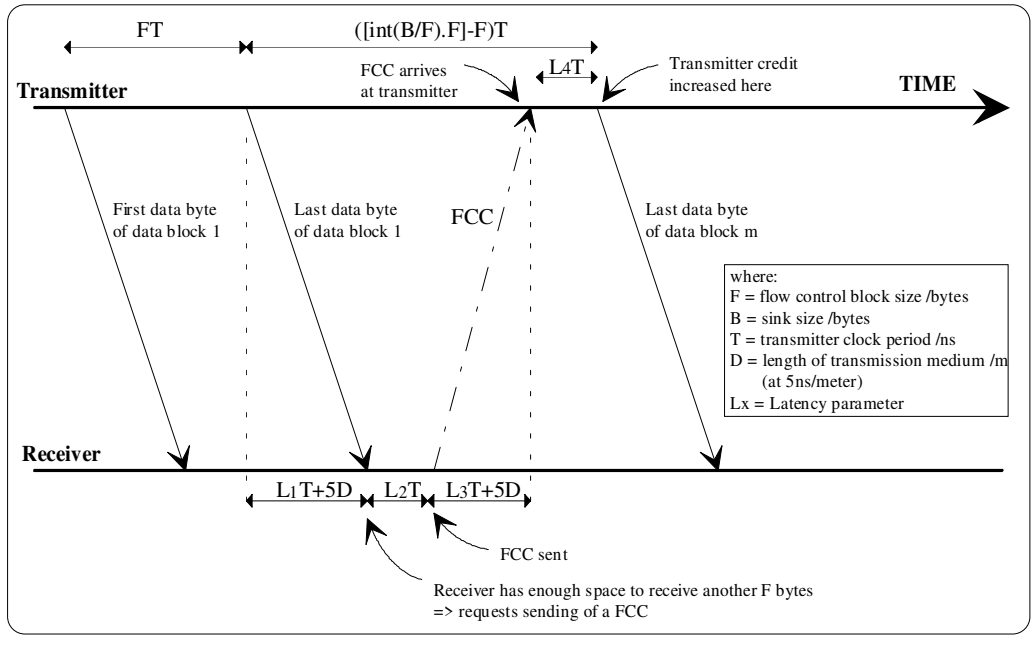


Figure 27: Theoretical Maximum Transmission Length Calculation

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

# 8. HS-FE-10 Fiber Optic Link

## 8.1. Physical Medium

Fiber optic cable: Monomode 9µ/125µ.
Fiber optic driver: Laser at 1300 nm.

[Editor's note: this section is to be completed]

## 8.2. Signal Level

### 8.2.1. General

The HS-SE-10 electrical signal levels operate with a nominal swing of 800mV, equivalent to ECL or PECL swings. Because the link may be AC coupled, the electrical driver and receiver can be directly interfaced with standard fiber optic transceiver components (which operate on ECL or PECL levels).

[Editor's note: this section is to be completed]

## 8.3. Character Level and Exchange Level

The CL and PL are identical to the HS-SE-10 implementation. See *.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

# 9. Common Packet Level

## 9.1. General Discussion

HIC does not define a specific size (or maximum size) for packets. This enables different packet formats to be carried by a HIC network. However, packets should be limited in size so that no one very long packet can occupy the network for an extended period of time. A limited known packet size also enables an upper bound to be placed on the latency of any packet transmission.

## 9.2. Packet Format

A packet consists of a *Destination* followed by a *Payload*, and is delimited by an end_of_packet marker.

<DEST> <PAYLOAD> <End_Of_Packet>

### 9.2.1. Destination

The destination contains one or more destination identifiers (dest_id) which are used as the input to the routing function, enabling the routing network to transmit the packet from its source-terminal-node to its destination-terminal-node. The destination could be null for specific case of a point-to-point link (i.e. the network is just one link).

Two types of destination are defined:

1. Simple
   The destination is one or more dest_id(s) in a list. A dest_id is a fixed size field (from 1 to n bytes long), its size being known to the network. The list isn't delimited and only the first dest_id is used by the routing function. The router treats the other dest_id(s) in the list (if they exist) as part of the payload (this is known as a *multiheader*).

3. Complex
   The destination contains a list of dest_ids, the list having a start_list and a stop_list marker. In practice, these two markers are the same character (a n_char). Since different types of list are envisioned, the list marker is parametrised: complex_identifier_n (cid_n).

### 9.2.2. Payload

The payload is the data (a message, a memory access request, an acknowledgment etc.) that has to be transferred from the source-terminal-node to the destination-terminal-node. It has a specific format, defined in the TL. A payload may be null. The definition of any packet format is outside the scope of this standard.

### 9.2.3. End_Of_Packet Marker

The end_of_packet Marker ('EOP' in its generic form) is a n_char, which is always the last character of a packet and acts as the packet delimiter. Packets do not have a start_of_packet marker: the first n_char received after an EOP is taken as the first character of the next packet.

Different EOP markers are defined in Table 11.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Table 11: End_of_Packet markers

| Name | Description |
|---|---|
| | |
| EOP_1 | Normal end_of_packet marker |
| EOP_2 | Exceptional end_of_packet marker |

The exceptional_end_of_packet marker may be used to signal an end of message or an error. Implementations may provide more than one type of exceptional_end_of_packet marker.

## 9.3. Routing

### 9.3.1. Routing Function

The (sub)network[1] routes the packet using the information contained in the destination field of the packet. The routing algorithm takes the fixed size[2] dest_id as its input in order to determine how to route the packet within the (sub)network. The routing algorithm is outside the scope of this standard.

### 9.3.2. Routing Functionalities

Other functionalities may be supported: multiheader (header deletion), multicast and broadcast.

a. Multiheader
A multiheader system allows the hierarchical building of complex networks from a series of subnetworks. The destination field contains a list of dest_ids, each one being the input to the routing function of a subnetwork. The complete identification of the destination-terminal-node is given by the ensemble of dest_ids. Each subnetwork routes the packet using the corresponding dest_id and, as the packet leaves the subnetwork to enter the next one, the dest_id is deleted (this is known as *header deletion*). This exposes the next dest_id in the list which routes the packet through its corresponding subnetwork. This process continues until the packet arrives at the subnetwork which contains its destination-terminal-node. The final dest_id routes the packet to this node; as the packet enters the node, this dest_id is deleted exposing the payload to the node.
A network that relies on the header deletion function (i.e. it has been built hierarchically) is known as a *hierarchical network*, one that does not is known as a *flat network*.

b. Multicast
Multicast is the distribution of a packet to a set of destination-terminal-nodes. The destination field contains a list of dest_ids, each one being the identifier of a destination-terminal-node to which the packet is to be sent.

c. Broadcast
Broadcast is the distribution of a packet to all destination-terminal-nodes in a network.

### 9.3.3. Use of simple and complex destinations

#### 9.3.3.1. Multiheader

Both simple and complex destinations can be used to support a multiheader:

[1] Subnetworks refer to the building blocks of a hierarchical network (see Multiheader)
[2] `Fixed size' means that the size is constant for a given (sub)network and is known to the (sub)network. The size determines the number of destinations that can be reached by the dest_id e.g. a one byte dest_id can address 256 nodes, two bytes 65536 nodes etc.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

#### 9.3.3.1.1. Simple

The destination format is:

dest_id*0* dest_id*1* dest_id*2* .... dest_id*n* <PAYLOAD> <EOP>

where the complete identification of the destination-terminal-node is given by the ensemble {dest_id*0* to dest_id*n*}.
The first subnetwork routes the packet using dest_id*0*. The subnetwork considers dest_id*1* to dest_id*n* as part of the payload. As the packet leaves this subnetwork, dest_id*0* is deleted, exposing dest_id*1* to the second subnetwork. This process continues in the manner described above.

#### 9.3.3.1.2. Complex

A complex destination needs to be used in the case where the router is required to perform some function on the payload. The destination format is:

**cid_0** dest_id*0* dest_id*1* dest_id*2* .... dest_id*n* **cid_0** <PAYLOAD> <EOP>

The functionality is as for a simple destination; however the router can now differentiate between the destination and payload parts of the packet. The header deletion function removes one by one the dest_ids, leaving in place the cid_0 markers. In the final subnetwork, as the packet enters the destination-terminal-node, the structure {cid_0 dest_id*n* cid_0} is removed to expose the payload. cid_0 is reserved to indicate a multiheader.

### 9.3.3.2. Multicast

Only a complex destination can be used for a multicast packet. Two packet formats are defined:

### 9.3.3.3. Multicast in a flat network

The destination format is:

**cid_1** dest_id*0* dest_id*1* dest_id*2* .... dest_id*n* **cid_1** <PAYLOAD> <EOP>

where each dest_id is the complete identification of a destination-terminal-node. This format therefore only allows multicast in a flat network. cid_1 is reserved to indicate a multicast packet in a flat network.

### 9.3.3.4. Multicast in a hierarchical network

The destination format is:

**cid_2** dest_id*0.0* dest_id*0.1* ... dest_id*0.n* **cid_2** dest_id*1.0* ... dest_id*1.m* **cid_2** ... ... **cid_3** <PAYLOAD> <EOP>

where the complete identification of each destination-terminal-node is the ensemble of dest_ids between two cid_2 markers (and between cid_2 and cid_3 for the last destination in the multicast list).

### 9.3.3.5. Broadcast

[Editor's note: this section is to be completed]

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

## 9.4. Error Detection, Recovery and Reporting

### 9.4.1. General

The function of the protocol is to deliver an error free[3] packet from its source-terminal-node to its destination-terminal-node. At the Packet Level or above, a system of *high level* error checking may be required. It is the responsibility of the Transaction Level to define the high level check that is to be used and therefore any definition of a high level check is outside the scope of this standard. In general, a high level check is required at the Packet level if the low level is suspected of being error prone and there is no higher level check (e.g. a message level CRC in a message passing system).

### 9.4.2. Packet Level Check: General Discussion

A Packet level check is used to check the validity of each packet's payload at its destination-terminal-node. This is normally achieved by appending an error checking code (for example a Cyclic Redundancy Code, CRC) to the payload. Since this code is normally calculated on the fly as the data is transmitted, it is added to the end of the payload. This implies that the destination-terminal-node cannot check the validity of the payload until it has received the whole packet. It is for the Transaction Level to decide when to use the incoming packet.

A hardware retry, at the packet level, would be difficult to implement since it implies that the packet level needs to know the source address of the erroneous packet and also a packet ID number, so that it can request retransmission[4]. It would be easier to leave the task to the Transaction Level and higher: any erroneous packet is discarded at arrival by the Packet Level, and a signal is passed up the levels to indicate the problem. Since the Transaction Level knows with which other terminal node it is communicating, retransmission can be safely requested. Note however that the Transaction Level (or part of it) could be implemented in hardware.

In some implementations, the error checking code could be checked in each routing node. In some implementations such as combining networks, the payload may be changed in the routing node, and hence the error checking code has to be recalculated.

The error checking code may not cover the destination field of the packet, because the latter's contents may be modified by header deletion and multicast functionalities. Adding a error checking code to each destination identifier is possible and relatively simple. By doing this, each destination is checked before the routing decision is taken.

[3] For some applications such as video data, the error free criterium is not so important, and so any error recovery action that interrupts the flow of data may need to be disabled.
[4] Even if it could extract these from the packet, it cannot be sure that these were not the bytes in error.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

# 10. Conformance criteria

## 10.1. Definition of subsets

An implementation of a connector shall conform to all of the specifications given in all of the sections for one of the following identifications:-

| Identification | Relevant sections |
|---|---|
| | |
| DS-DE PCB connector | (to be completed when the numbering system has settled down) |
| DS-DE cable connector | |
| TS-FO PCB connector | |
| TS-FO fibre connector | |
| HS-SE | |
| | |
| | |
| | |
| | |

An implementation of a cable or optic fibre shall conform to all of the specifications given in all of the sections for one of the following identifications:

| Identification | Relevant sections |
|---|---|
| | |
| DS-DE cable | |
| | |
| | |
| TS-FO fibre | |
| HS-SE | |
| | |
| | |
| | |
| | |

An implementation of a link interface shall conform to all of the specifications given in all of the sections for one of the following identifications:

| Identification | Relevant sections |
|---|---|
| | |
| DS-SE | (electrical, signal, character, exchange, packet) |
| DS-DE | (female connector, electrical, signal, character, exchange, packet) |
| TS-FO-01 | |
| TS-FO-02 | |
| HS-SE-10 | |
| | |
| | |
| | |
| | |

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

(perhaps the link interface should go up to the exchange level only, in which case we would also have, logically, the following. . . ..)

An implementation of a switch chip . . .

An implementation of a switch . . .

### 10.2. Conformance testing

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

# Normative Annexes

[Edior's note: this section is to be completed]

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

# Informative Annexes

# Annex A. Rationale

Need for a New Standard

It is widely recognised today that the most economic way to build very high-performance systems is by using parallelism. Parallel systems can provide very high computational power (in machines such as the CM-5 and the Parsytec GC-machine), fast response (for transaction processing or distributed control), very large I/O throughput (in RAID systems) and extremely high reliability (in redundant fault-tolerant systems). They also have the potential to be more maintainable and more expandable than conventional, monolithic systems.

The construction of high-performance systems with parallel processing and/or parallel i/o demands a fast, low-cost, low-latency interconnect. It must be fast and low-latency, otherwise it will be the limiting factor in system performance, and it must be low-cost, or else it will dominate the system cost. It must also scale well in both performance and cost relative to the system size, otherwise highly parallel systems will be limited in performance or too expensive. Existing standards do not meet these criteria, either because they are designed for communication over long distances (which incurs high costs), or because they aim at the extreme of currently achievable performance (which again increases costs), or because they are based on a restricted model such as a bus, which limits overall performance and scalability.

It is proposed to create a new interconnect standard to enable high-performance, scalable, modular, parallel systems to be constructed with low cost, where "cost" must include not only the price of components, but also the engineering effort required to use them successfully. This standard will specify the physical connectors and cables to be used, the electrical properties of the interconnect, and a cleanly-separated set of logical protocols to perform the interconnection in the simplest possible way.

Aspects of a Parallel Systems Interconnect

Parallel Interconnect

How can the requirements of performance, scalability and low cost be met? Since these are no different from the requirements of the whole parallel system, they can be satisfied in the same way; by allowing many instances of a cheap component to operate concurrently. Thus the interconnect should consist of many separate connections operating simultaneously to give a high aggregate performance. Provided each connection can be utilised at a reasonable level, its raw performance need not be very high, allowing both component and engineering costs to be kept down. For maximum simplicity, modularity and fault-tolerance, each connection should be point-to-point.

The requirement of low cost implies that, at the very least, a connection to the interconnect can be implemented with a relatively small amount of circuitry in a non-exotic technology. Ideally such a connection could be integrated onto a chip with a processor or other device to minimise costs. The requirement of a small amount of circuitry implies that protocols must be simple and require minimal buffering. At the system level, cost limits the number of signal connections which can be used. Too large a number would make integration with a processor impractical, for example, limit the maximum number of connections available on a device, and lead to skew-control problems at the system level.

Routing

To connect many devices together, it is not possible to provide a direct, physical connection between every pair. Nor is it acceptable to connect only certain pairs unless the connection pattern happens to

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

match that required by the application. To enable every pair of devices to communicate without necessarily having a direct connection, data must be *routed* in some way through intermediate nodes.

In order for *every* pair of devices to be able to communicate, data must be routed in different ways at different times. This can be achieved either by configuring intermediate nodes to make each connection before the data is sent, or by making the data *self-routing* so that it contains within it information which determines which way it should be routed. Configuring the nodes in advance of the data requires that the destination of the data must be supplied to a central controller, which increases latency, and creates the danger that the controller will become a system bottleneck. Self-routed systems enable the control function to be distributed, and hence to scale in performance with the size of the system. Consequently, this proposal will consider only self-routed systems.

A piece of data together with its associated routing information is called a *packet*. If a packet is sent directly from sender to receiver it is not necessary for the length of the packet to be represented within it; it is only necessary to ensure that the sender and receiver agree on the length. However, where packets pass through intermediate nodes, it must be possible for these nodes to determine the length of the packet passing through, to determine when the task of routing a particular packet is complete. The requirement of low-latency implies that packets must be limited in length, since otherwise connections could be occupied indefinitely by long packets. Since for many purposes short messages are required, the overhead on each packet must be small, and the packet size must be variable. This requires that the protocol provides an indication of the packet length, such as a termination marker or an initial length count.

Wormhole Routing

In most serial packet-switching networks each intermediate node inputs a packet, decodes the header, and then forwards the packet to the next node. This is undesirable for a parallel systems interconnect for two reasons:

a. It requires storage in each node for transmitted packets, which either limits the capacity of the node or requires a separate memory (which increases costs).

b. It causes potentially long delays between the output of a packet and its reception, because each node waits for the whole packet to be received before starting re-transmission, thereby increasing latency.

A more suitable approach is *wormhole routing* (Dally), in which only the header of the packet is initially read in by the node. The routing decision is taken, the header is output, and the rest of the packet is sent directly from the input to the output without being stored in the node. This means that a packet can be passing through several nodes at the same time, and the head of the packet may be received by the destination before the whole packet has been transmitted by the source. Thus this method can be thought of as a form of dynamic circuit switching, in which the header of the packet, in passing through the network, creates a temporary circuit (the `wormhole') through which the data flows. As the tail of the packet is pulled through, the circuit vanishes[5]. The transmission of a single packet may thus be *pipelined* through a series of devices.

Note that, as far as the senders and receivers of *packets* are concerned, the wormhole routing is invisible. Its only effect is to minimise the latency in the message transmission. If one or more intermediate nodes were to store-and-forward the packet it would still be delivered correctly. Note also, however, that wormhole routing has the further advantage that it is independent of the packet length. In a store-and-forward system, the maximum packet size must be determined in advance so that buffering can be provided; if few packets are of this size, then the extra buffering is largely unused. In addition, independence from the packet length is desirable because it achieves a clean separation of layers of the protocol.

[5] The analogy of an earthworm pushing its way through sand is very apt.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Flow Control

Guaranteeing that physical connections will be available for each stage of a packet's journey requires global information about the state of the system. Accumulating global information is time-consuming and inherently non-scalable. Thus in a low-latency, scalable system, the possibility exists that the header of a packet will be input by a node but be unable to proceed because the required output is already in use. However the body of the packet will still be passing through previous nodes. There are then three possibilities:

a. The incoming packet body is buffered in the node where the packet is stalled;

b. The incoming data is discarded;

c. A flow-control mechanism stops the flow of data.

The first of these options is a return to store-and-forward routing[6], with the disadvantage of requiring buffer resources in each routing node, which increases cost and destroys the packet length independence of the routing. The second option is undesirable because it forces the end-equipment to engage in complex protocols to deal with the possibility of data loss. In addition, once part of a packet has been lost the packet is probably useless and may as well be entirely destroyed. A system using this scheme could then easily degenerate into a state where most of the connections were carrying packets to the point of their destruction! Clearly it would be preferable to propagate information about a stall back along the path of the packet, which brings us to the third option, to implement a system of flow-control.

Since we do not wish to provide buffering for an entire packet, the flow-control system must be capable of stalling the flow of data part-way through a packet. This implies that it operates on a level *below* that of packets. It is the flow of the sub-units of which packets are composed which must be controlled. With this scheme, when a packet is unable to proceed, data may continue to move until all buffering along the path of the packet is filled. The flow-control mechanism must then ensure that data movement ceases so that no buffers are overwritten. This means that all links which are still occupied by the packet will be idle; however this is an improvement on the previous scenario in which they would be busy moving data to the point at which it is discarded.

Comparison with Other Types of System

In this section we consider how the requirements for a parallel system interconnect compare with those of other types of communications system.

Telecommunications Systems

The principal feature of digital telecommunications systems is that they operate over long distances. This means that the actual, physical medium of communication is large and expensive[7]; most of the attributes of such systems are a consequence of this. Because the medium (be it an ordinary cable, optical fibre or satellite) is expensive, it is worthwhile using expensive end equipment to extract the most from it. It is justifiable to push the basic operating frequency to a level at which the BER is non-negligible and to compensate for the errors with very sophisticated protocols, if this results in a net improvement in data rate. Protocols may be further complicated by the need to perform a number of functions via the same medium, since in general there will not be an alternative connection (and if there were it would not be economic to dedicate it only to functions which are used infrequently, however important they may be).

[6] This combination of methods is called 'virtual cut-through'

[7] It is estimated that 50% of the assets of major telecommunications companies are in the form of buried cables!

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

The extended nature of the medium makes it vulnerable to noise and transient interruption[8]. Thus information may be lost; and so for example if fine-grained flow-control were employed it would frequently fail due to loss of the flow-control information[9]. This implies that flow-control must be coarse-grained, and so large buffers must be provided. Since packets may be lost, they must be buffered for re-transmission until positive confirmation of reception is received. This may in turn cause further packets to be lost if the buffers of a forwarding node are full. Thus telecommunications systems are intrinsically "lossy" and higher-level protocols must compensate for this (see Black).

Local-area Networks

LANs share some of the attributes of telecommunications systems, but the balance of cost between the physical medium and the end-equipment is quite different. Because the cost of the medium is still significant, however, LANs work on the principle that a number of pieces of end-equipment effectively share the same medium (and hence share its bandwidth). BERs are lower, but the protocols used can still be quite complicated in order to organise the sharing of the medium.

To economise on wiring, the most popular physical configuration of LANs is a ring. Rings may be linked together via `bridges'; this improves the bandwidth because the bandwidth of each ring is shared between the active users connected to it, whilst separate rings joined by bridges may operate concurrently. It also improves the fault-tolerance, since the failure of one ring can be isolated from the others. However the overall level of concurrency in the interconnect is low.

Parallel Systems Interconnect

For communications in a parallel system, the constraints are rather different. When connecting devices on a board or in a card-cage, the actual cost of the wires involved is negligible. However the pins and board area consumed by wide, parallel connections and the difficulty of correctly routing them is not negligible, and this leads to a preference for serial communications.

Since the cost of the copper tracks is not significant, it is not necessary to maximise the data rate on each one. Nor is it essential to use a complex protocol to perform a variety of different functions on a single connection, since the cost of providing separate connections for, for example, configuration, control or monitoring purposes need not be prohibitive.
Within a system with limited physical dimensions and reasonably clean electrical properties it is entirely feasible to make connections with extremely low BERs ($10^{-15}$ or less). Thus connections can be assumed reliable, and complicated systems of acknowledgements, anti-acknowledgements and re-tries can be avoided. Moreover since the system can be designed and controlled as a whole, each part can be relied upon to obey strict rules, and need not behave defensively with respect to other parts of the system, allowing further simplifications. Although in such systems it is possible to make communications synchronous with a common clock, this can create serious problems of high-speed clock distribution and should be avoided to keep costs down.

In such systems, the cost of the end-equipment for each connection is very significant. This is because at least one such end must be provided for each device, and hence contributes to the cost of the node. For reasons of both cost and performance it is very advantageous to be able to integrate the end-equipment with the device chip(s). Thus it is preferable not to require sensitive analogue circuitry, exotic technology (GaAs etc.), or large amounts of logic. Replicating a simple piece of end-equipment is more cost-effective than providing a complex one with maximum performance.

High bandwidth and low latency are essential for the communications to be matched with the speed of the devices. In particular it is important to provide a high total bandwidth when many devices need to communicate simultaneously, and for the latency of each communication to be moderate even under high load. Fault-tolerance is highly desirable, and these requirements together imply that the

[8] For example an aeroplane flying between a ground station and a satellite.
[9] It would also be inappropriate because of the large latencies involved.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

communication should be performed via a number of independent, concurrently operating connections rather than one higher-speed connection.

References

Uyless D. Black,
Data Networks: Concepts, Theory, and Practice
Prentice-Hall International 1989.

Dally, W.
A VLSI Architecture for Concurrent Data Structures
Computer Science Department, California Institute of Technology 5209:TR:86 1986

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

# Annex B. Switch chips, switches and fabrics

A switch provides a number of node interfaces, and determines how these interfaces are identified (numbered). The standard specifies the protocols on each of these interfaces, which includes routing information defining the interface(s) to which an individual packet is to be routed. A switch may be implemented from one or more switch chips (or indeed by other methods). A switch chip may provide pins which directly implement switch interfaces, in which case it can claim appropriate conformance to the standard. It is proposed that the standard leaves open how switch chips can be combined to implement a switch. The standard is not prescribing how routing works, but saying some things about it, essential to the use of a switch. This allows competitive proprietary technologies for implementing the internals of switches, without compromising the open nature and value of a standard for using the switches.

A fabric is a general routing capability, constructed from one or more switches. The standard shall define the protocol for packets which are routed through a number of such switches (the technique of `header deletion' as a packet leaves a switch is proposed).

The standard thus can guarantee that a user can use switches from different vendors (bought possibly at different times) in combination (subject to their using the same subset!).

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

# Annex C. Use of the Transaction Layer

This annex gives guidelines for the use of the transaction layer to support higher level protocols, the constraints of any such mappings to ensure that they are compatible, and example mappings for SCI, ATM, Fiber Channel, . . . .

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

# Annex D. DS-Link error handling

DS-Links are designed to be highly reliable within a single subsystem and can be operated in one of two modes, determined by a flag at each end of the link, called **LocalizeError.**

In applications where all connections are on a single board or within a single box, the communications system can reasonably be regarded as being totally reliable. In this environment errors are considered to be extremely rare, but are treated as being catastrophic should one occur. If an error occurs it will be detected and reported. Normal practice will then be to reset the subsystem in which the error has occurred and to restart the application. This minimizes the overheads on each communication, but if an error does occur there will be an interruption in the operation of the system.

For other applications, for instance when a disconnect or parity error may be expected during normal operation, a higher level of fault-tolerance is required. This is supported by localizing errors to the link on which they occur, by setting the **LocalizeError** flag of the link to 1. If an error occurs, packets in transit at the time of the error will be discarded or truncated, and the link will be reset automatically. This minimizes the interruption of the operation of a system, but imposes an overhead on all communications in order to deal with the possibility that data may be lost.

The flag **LocalizeError** is set to 0 or 1 either as part of the specification of the link implementation, or as part of the configuration of the link. A device implementing one or more links shall either specify the setting of the flag **LocalizeError** for some or all of its links, or shall be configurable to provide a means of selecting the setting of the flag **LocalizeError** for one or more of its links. A device with multiple links may provide separate configuration for each of its links, or may group various of the links together for error configuration purposes. The error configuration shall be either fixed or configurable during an initialisation process. It shall not be selectable during operation of the link. The means of setting he flag **LocalizeError** for a link is not defined within this standard.

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

# Annex E. DS-DE Connector PCB recommendation

| | | |
|---|---|---|
| Electrical termination, PCB connector | solder | |
| PC hole diameter contact | 0.9mm min | Draft IEC 1076-4-101 |
| PC hole diameter location peg | 2.0mm ± 0.05 | Draft IEC 1076-4-101 |
| Printed board thickness, daughter card | max 2.7 mm | |

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

# Annex F. DS-DE Cable Recommendation

The cable required for a differential DS-link connection consists of four, shielded, twisted pairs, which are then shielded overall. Each differential pair must have a characteristic impedance of 100 Ohms. A suitable cable would be a 9 conductor 30 AWG composite cable, as shown in the following diagram:

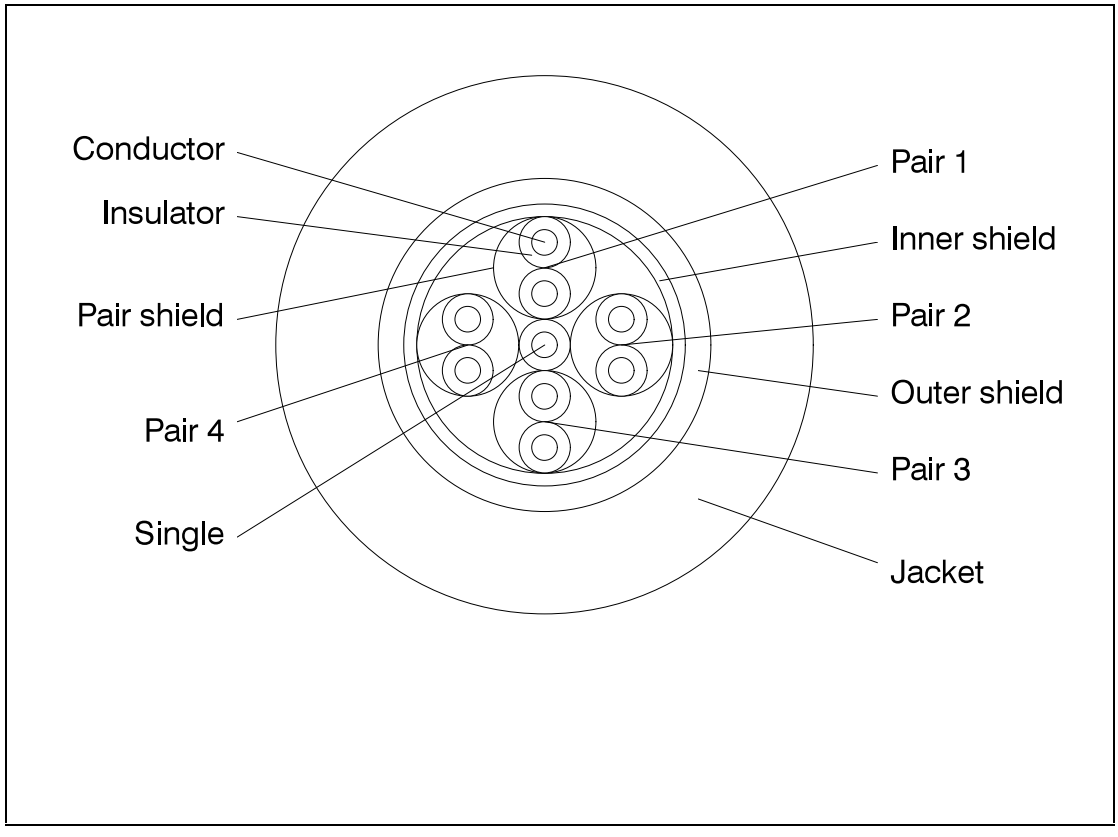


Figure 28: DS-Link Cable

*Construction*

| Pair | |
|---|---|
| | |
| Conductor | 30 AWG 7/38 tin plated copper, 30mm diameter |
| Insulation | 0.22mm polypropylene, 0.74mm diameter, dielectric constant is 2.24 |
| Pair | two insulated conductors twisted together |
| Pair shield | 0.025 mm/0.025 mm polyester tape, aluminium side facing out, 25% overlap |
| | |
| Single | |
| | |
| Conductor | 30 AWG 7/38 tin plated copper,30mm diameter |
| Insulation | 0.14mm polypropylene, 0.58mm diameter, dielectric constant is 2.24 |

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Table (contd)

| Cable | |
|---|---|
| | |
| Core | one single |
| Layer 1 | four pairs twisted around the core |
| Inner shield | 35 AWG tin plated copper braid, 65% nominal coverage |
| Jacket | 0.56mm super flexible PVC, loosely applied to enhance flexibility, colour - black |
| Diameter | 5.33 +/- 0.25 mm nominal |

*Colour code*

| Component | Conductor 1 | Conductor 2 |
|---|---|---|
| | | |
| Pair 1 | Red | Green |
| Pair 2 | Brown | Blue |
| Pair 3 | Orange | Yellow |
| Pair 4 | Violet | Grey |
| Single | Black | |

The following table gives the recommended wiring schedule for the cable and the connector at either end:

| Pin | Conductor | Pin |
|---|---|---|
| | | |
| 1a | Pair 2 Brown | 2e |
| 2a | Pair 2 Blue | 1e |
| 1b | Pair 1 Red | 2d |
| 2b | Pair 1 Green | 1d |
| 1c | *No Connect* | 1c |
| 2c | Black | 2c |
| 1d | Pair 3 Orange | 2b |
| 2d | Pair 3 Yellow | 1b |
| 1e | Pair 4 Violet | 2a |
| 2e | Pair 4 Grey | 1a |

Copyright © 1993 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.