ECSS-E-50-12 Space Engineering

SpaceWire:

SERIAL POINT-TO-POINT LINKS

DRAFT

ISSUE C

DOCUMENT NUMBER	UoD-DICE-TN-9201
ISSUE	ISSUE C
DATE	January 2000
ESA CONTRACT NO.	12693/97/NL/FM
ESA TECHNICAL MANAGER	J. Rosello
AUTHOR	Dr SM Parkes, University of Dundee
CONTRIBUTORS	SPACEWIRE WORKING GROUP

Document Revision History			
Revision	**Date**	**Responsible**	**Modifications/ Reasons for Change**
Draft A	Sept. 1999	SM Parkes	Initial release
Draft B	Oct. 1999	SM Parkes	Corrections from initial comments from ESTEC and Dornier Satellitensysteme on draft A.
Draft C	Jan 2000	SM Parkes	Corrections following comments received during SpaceWire Working Group meeting on 27th October and Core Group meeting on 8th December. Also reflects comments received by email.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

ACKNOWLEDGEMENTS

SpaceWire is the result of the efforts of many individuals within the European Space Agency, European Space Industry and Academia.

1. OVERVIEW

1.1 PURPOSE

The SpaceWire standard addresses the handling of payload data on-board a spacecraft. It is a standard for a high-speed data link, which is intended to meet the needs of future, high-capability, remote sensing instruments and other space missions. SpaceWire provides a unified high-speed data-handling infrastructure for connecting together sensors, processing elements, mass-memory units, downlink telemetry sub-systems and EGSE equipment.

The purpose of this standard is

- to facilitate the construction of high-performance on-board data-handling systems
- to help reduce system integration costs
- to encourage re-use of data handling equipment across several different missions.

1.2 SCOPE

The SpaceWire standard specifies the physical interconnection media and data communication protocols to enable data to be sent reliably at high-speed (between 2 Mbps and 100 Mbps or more) from one unit to another. SpaceWire links are full-duplex, point-to-point, serial data communication links.

The scope of this standard is the physical connectors and cables, electrical properties, and logical protocols that comprise the SpaceWire data link. SpaceWire provides a means of sending packets of information from a source node to a required destination node. SpaceWire does not specify the contents of the packets of information.

The SpaceWire standard covers the following normative protocol levels

- **Physical Level:** Defines connectors, cables and EMC specifications.
- **Signal Level:** Defines signal encoding, voltage levels, noise margins and data rates.
- **Character Level:** Defines the data and control characters used to manage the flow of data across a link.
- **Exchange Level:** Defines the protocol for link initialisation, flow control, fault detection and link restart.
- **Packet Level:** Defines how data to be transmitted via a SpaceWire link is split up into packets

The following informative protocol levels are also provided.

- **Network Level:** Defines the form of routing switches used to build a network from SpaceWire links and how packets are transferred from a source node to a destination node across a network.
- **Error Recovery Scheme:** Provides recommendations for error detection, error recovery and error reporting.

SpaceWire is based on two existing commercial standards, IEEE 1355-1995 [RD1] and Low Voltage Differential Signalling [AD1, RD2] which have been combined and adapted for use on-board spacecraft.

1.3 GUIDE TO THE STANDARD

This SpaceWire standard document begins with section 1 (this section) which covers the purpose and scope of the standard, gives a list of applicable and reference documents, summarises the rationale for the standard and introduces the notation used throughout the standard document. A statement regarding intellectual property is also given. Section 2 provides the necessary definitions of conformance keywords and a technical glossary. A brief overview of the standard is given in section 3 to familiarise the reader with the basic SpaceWire concepts, prior to the detailed specification of subsequent sections. Section 3 also aims to provide some explanation of the key decisions made about the SpaceWire standard.

The body of the SpaceWire standard is presented in sections 4-8, which ascend through the various normative levels of the standard.

- Section 4 (Physical Level) covers cables, connectors and cable assemblies.
- Section 5 (Signal Level) deals principally with electrical characteristics, coding and signal timing.
- Section 6 (Character Level) describes how data and control characters are encoded.
- Section 7 (Exchange Level) presents the mechanisms by which a SpaceWire link operates and covers starting a link, normal operation, error detection and recovery.
- Section 8 (Packet Level) describes the support provided for packet switching, including the way in which data is packed into packets for transfer across a SpaceWire link or network.

Sections 9 and 10 are non mandatory (informative) parts of the standard which provide information on the network level and error recovery scheme.

- Section 9 (Network Level) deals with the structure and operation of a SpaceWire network.
- Section 10 (Error Recovery Scheme) provides recommendations for error detection, error recovery and error reporting.

The SpaceWire standard concludes in section 11 with a list of conformance statements, highlighting those parts of the standard that must be followed for a system to be SpaceWire compatible.

There are five annexes which present:-

- Initial requirements for SpaceWire.
- A summary of SpaceWire EMC performance related to the needs of a typical spacecraft.
- The differences between SpaceWire and IEEE standard 1355-1995 [RD1].
- A SpaceWire user guide giving hints and tips on the development and design of SpaceWire systems.
- List of informative references.

1.4 NORMATIVE REFERENCES

This standard shall be used in conjunction with the following publications:

AD1 Telecommunications Industry Association, "Electrical Characteristics of Low Voltage Differential Signaling (LVDS) Interface Circuits", ANSI/TIA/EIA-644-1995, March 1996[1].

AD2 ESA/SCC Generic Specification No. 3902[2]

AD3 ESA/SCC Generic Specification No. 3401/029

Note: Informative bibliography can be found in Annex E.

[1] ANSI publications are available from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA (http://www.ansi.org)

[2] ESA/SCC publications are available from the ESA/SCC secretariat (http://www.estec.esa.nl/qcswww). ESA/SCCNo 3902 and No 3401/029 have also been posted at http://www.estec.esa.nl/tech/spacewire/techmodules.html#Cables_connectors

1.5 RATIONALE FOR STANDARD

SpaceWire technology has grown organically from the needs of on-board processing applications. This SpaceWire Standard document provides a formal basis for the exploitation of SpaceWire in a wide range of future on-board processing and data handling systems. It aims to secure the benefits of equipment compatibility and reuse.

1.5.1 Brief History of Development

The high-speed data links that evolved into SpaceWire were initially developed for use in multiple DSP processor systems. The links were seen as a means of helping to solve demanding on-board signal and image processing problems by connecting together several programmable DSP processors and other processing devices into a high performance parallel processing system [RD3, RD4, RD5]. An integrated architecture was foreseen where various processing elements were connected into a heterogeneous network. The proposed data link was based on IEEE standard 1355-1995 [RD1].

The wider application of the data link became apparent when it was used to form a versatile reconfigurable solid-state memory for space applications [RD6]. This brought benefits of modularity and fault tolerance to a simplified solid-state memory system. IEEE-1355 type links and routing switches were used to send data to and read data from an array of memory modules.

The extension of the emerging architecture to embrace sensors, down-link telemetry and EGSE into a unified on-board data handling infrastructure was natural [RD7, RD8].

With the growing interest in IEEE-1355 type links for space applications it was important to consider issues relating to space qualification. Dornier Satellitensysteme was developing IEEE-1355 encoder/decoder devices in radiation tolerant technology [RD9, RD10], but work remained to be done on the line drivers/receivers, cables and connectors and EMC performance. The Digital Interface Circuit Evaluation (DICE) study was initiated to examine these issues [RD11]. The DICE study resulted in the SpaceWire standard document.

1.5.2 Main Features

An overview of the main features necessary for a data link for use in space applications are listed below [RD11].

- **Data Rate:** A data link shall have sufficient capacity or bandwidth to carry the data for which it was intended. 100Mbaud is an appropriate minimum target for the maximum data rate.
- **Distance:** The data link shall operate over a distance of at least 10m. This distance is commensurate with the size of a large spacecraft enabling data to be transmitted from one extremity to the other.
- **Scalability:** To meet the data rate requirements of particularly demanding applications it shall be possible to use several links in parallel to increase the data rate accordingly.
- **Error Rate:** The error rate on the link shall be low, better than a BER (bit error rate) of 10^{-12} for the basic link and better than 10^{-14} for a link protected by a higher level error detection protocol.
- **Power Consumption:** The power consumption of the link shall be low.
- **Low mass and small size:** The mass and size of the data link interface and the cable shall be as small as possible.
- **Cold Redundancy:** The data link shall support connection within a cold redundant system, i.e. when part of the system is powered and another part is not powered.

- **EM Susceptibility:** The data link shall not be susceptible to interference from external electromagnetic sources. It should meet the EM susceptibility requirements of most space missions.
- **EM Emission:** The data link shall not emit electromagnetic radiation at a level that would interfere with the operation of other systems. It should meet the EM emission requirements of most space missions.
- **Magnetic Emission:** Magnetic emissions from the data link shall be low – ferrous materials should not be used in the data link components.
- **ESD Immunity:** The electronic devices forming a link shall have a high level of immunity to damage by electro-static discharge.
- **Galvanic Isolation:** It should be possible to galvanically isolate the data transmission system from the data reception system.
- **Radiation Tolerance:** The components that implement the data link shall be tolerant of radiation.

Detailed requirements are listed in Annex A.

1.5.3 Reuse

One of the principal aims of SpaceWire is the support of re-use at both the component and sub-system levels. In principle a data-handling system developed for an optical instrument, for example, can be used for a radar instrument by unplugging the optical sensor and plugging in the radar one. Processing units, mass-memory units and downlink telemetry systems developed for one mission can be readily used on another mission reducing the cost of development, improving reliability and most importantly increasing the amount of scientific work that can be achieved within a limited budget.

Integration and test of complex on-board systems is also supported by SpaceWire with ground support equipment plugging directly into the on-board data handling system. Monitoring and testing can be carried out with a seamless interface into the on-board system.

A unified data-handling architecture is essential for cost effective missions – SpaceWire provides the basis for such a high-speed on-board data-handling architecture.

1.6 Disclaimer – Intellectual Property

The implementation of this standard may require the use of intellectual property covered by patent rights. ESA shall not be responsible for identifying all the patents for which a license is required to implement the SpaceWire standard. Furthermore, ESA shall not be responsible for ensuring the existence or legal validity of any patent related to the SpaceWire standard.

1.7 Document Notation

1.7.1 Signal Naming

All electrical signals are shown in uppercase letters.

The two signals making up a differential pair are given the suffixes + and – to indicate the positive and negative components of the differential signal respectively.

The SpaceWire differential signals are referred to as D+,D- and S+,S- for data and strobe respectively. When considering the driven end of a SpaceWire link these signals may be designated Dout+, Dout- and Sout+ and Sout- for data and strobe respectively. Similarly the signals at the input end of a SpaceWire link are Din+, Din- and Sin+, Sin-.

1.7.2 Packet Formats

Packet formats are represented in two ways in this document. The first way is graphical and is shown in Figure 1-1. The field at the top is the one that is transmitted first.

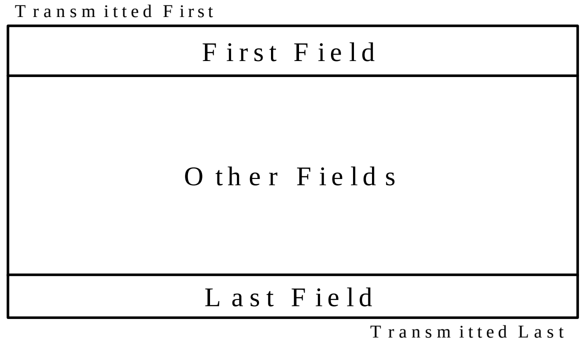

Figure 1-1 Graphical Packet Notation

The second packet representation is textual. Each field is enclosed in chevrons <>. The fields comprising a packet are written left to right in the order that they are transmitted. The example below is equivalent to that shown in Figure 1-1.

EXAMPLE:

<First Field><Other Fields><LastField>

1.7.3 State Diagram Notation

*All state diagrams in this standard use the style shown in Figure 1-2***.** States are represented by ellipses with the state name written inside the ellipse in bold. Actions to be taken while in a particular state are written in italics inside the ellipse underneath the state name. Transitions from one state to another are indicated by arrows. The event that causes a transition is written alongside the arrow. Unconditional transitions are indicated by arrows without an event name written next to it. Reset conditions are indicated by transition arrows that start in empty space.

State names referred to in the text of the standard are in italics e.g. *FirstState.*

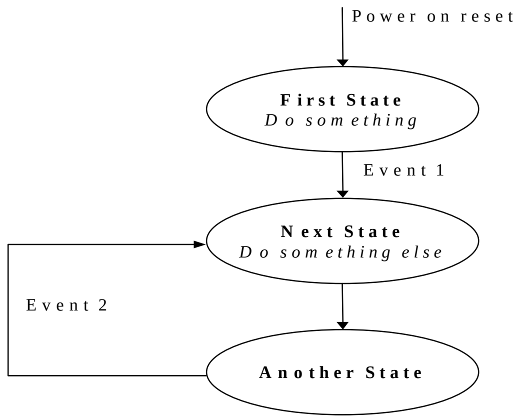

Figure 1-2 State Diagram Style

2. DEFINITIONS

2.1 CONFORMANCE GLOSSARY

The following keywords are used to differentiate between different levels of requirements and optionality, as defined in IEEE Std 100-1992 [RD12].

Shall: indicates a normative requirement. To ensure interoperability with other products conforming to this standard, all normative requirements must be followed strictly with no deviation.

Should: indicates a recommended but not normative requirement. Allows flexibility of choice between several possible alternatives while indicating a strongly preferred alternative. Indicates that a certain course of action is desirable but not normative, or indicates that a certain course of action is deprecated but not prohibited.

May: indicates a suggested course of action without implying preference over any other possible course of action.

2.2 TECHNICAL GLOSSARY

ACK: Acknowledge

Acknowledge: An indication that a message has been received successfully by its intended destination.

AWG: American Wire Gauge

BER: Bit Error Rate

Binder: A layer of tape wrapped around one or more cables to keep them together in a fixed position. The tape is usually PTFE and is wrapped in an overlapping spiral along the length of the cables to be bound.

Bit Error Rate: The ration of the number of bits received in error to the total number of bits sent across a link.

Byte: Eight bits

Character: A control character or data character

Character Level: The protocol level that deals with the encoding of data and control characters into a bit-stream.

Check Sum: A byte or word added to the end of a message or packet used for error detection purposes. The check sum is formed by adding together all the bytes in the message. The addition is done modulo the size of the check sum e.g. 256 for a byte-sized checksum. The receiver also calculates the message checksum and compares it against the one sent with the message to check for errors.

Coding: Translation from one set of bits to another new set of bits.

Control Character: A character that is used to control a link. Control characters include the Link-characters (NULL and Flow Control Token, FCT) and the end of packet markers (EOP and EOM).

Data Character: A data byte encoded ready for transfer across a link.

Data Signalling Rate: The rate at which the bits constituting control and data characters are transferred across a link.

Data-Strobe: An encoding scheme in which a sequence of data bits (and clock) is encoded as the original data bit sequence, together with another bit sequence (strobe) which changes state whenever the data bit sequence does not.

Decoding: The act of translating an encoded set of bits to the original set of bits prior to coding.

De-serialization: Transformation of a serial bit stream into a sequence of control and/or data characters.

Destination: The node or unit that a packet is being sent to.

Destination Address: The route to be taken by a packet in moving from source to destination.

Destination List: A list of destination identifiers which forms the destination address of a packet.

Destination Identifier: The address, or partial address, of the packet destination.

Driver: An electronic circuit design to transmit signals across a particular transmission medium.

DS: Data-Strobe.

EGSE: Electronic Ground Support Equipment.

End of Packet Marker: A control character which indicates the end of a packet.

EOP: End Of Packet market type one – used to indicate the normal end of a packet.

EOM: End of Message.

Error Recovery Scheme: The preferred approach for the handling of errors detected within a SpaceWire link.

ESC: Escape sequence, which is defined in the Character Level.

Exchange Level: The protocol level which handles the flow of packets across a link.

FCT: Flow Control Token.

Filler: A cylindrical piece of PTFE used to fill the gap between insulated wires or cables being grouped together and formed into a larger cable. The filler enhances the structure of the cable helping to keep the constituent wires in a fixed position relative to one another.

Flow Control Token: A control character used to manage the flow of data across a link. Each flow control token indicates that there is space for 8 more normal-characters in the receiver buffer.

Jitter: Random errors in the timing of a signal.

Lay Length: Length of lay (twists per foot). Lay length refers to the number of twists per foot and is expressed in inches or decimals as the length between one complete turn of a single end in the cable.

Link: A link represents the connection of one unit to another unit. A link is bi-directional. Each end of a link contains a link transmitter and a link receiver. Data and control information is passed along a link from one end to the other.

Link-character: A control character used to manage the flow of data across a link (NULL and FCT).

Link Destination: Used to refer to the end of the link that is receiving a particular set of data or control information.

Link Receiver: Used to refer to the receiver at one end of a link.

Link Source: Used to refer to the end of the link that is sending a particular set of data or control information.

Link Transmitter: Used to refer to the transmitter at one end of a link.

Low Voltage Differential Signalling: A particular form of differential signalling using low voltage signals.

LVDS: Low Voltage Differential Signalling.

Mbps: Mega bits per second.

NACK: Negative acknowledge i.e. indication that an error or fault has occurred.

Network: A set of units connected together via links and routing switches.

Network Level: The protocol level that defines the SpaceWire network routers and defines how packets of data are transferred across the network from source node to destination node.

Node: A source and/or destination of a packet. A node may be a processor, memory unit, sensor, EGSE or some other unit connected to a SpaceWire network.

Normal-character: A data character or control character that is passed from the exchange level to the packet level (EOP, EOM or Escape Sequence).

NULL: The NULL character or token is sent to keep the data link active when there are no data or control characters to be sent.

Packet: A sequence of normal-characters comprising a destination address, packet payload and an end of packet marker.

Packet Level: The protocol level that defines how data is organised in packets ready for transfer across a link or network.

Packet Payload: The data that is to be transferred from a source to a destination.

PFA: A type of plastic used to cover wires in cables.

Physical Level: The protocol level that specifies the physical interconnection medium – cables and connectors.

PTFE: Polytetrafluroethylene. A type of plastic used to cover wires in cables.

Receiver: An electronic circuit design to receive signals sent across a particular transmission medium.

Router: Routing switch.

Routing Switch: A switch connecting several links that routes packets from one link to another. The destination address of each packet by the switch is used to determine which link a packet will be sent out on.

SCI: Scalable Coherent Interface is an IEEE standard (IEEE 1596) for connecting processors and peripherals in a high performance multiprocessor architecture.

Serialisation: Transformation of a sequence of control and/or data characters into a serial bit stream.

Signal: A measurable quantity that varies with time to transfer information. A signal propagates along a transmission medium.

Signal Level: The protocol level which defines the electrical signals used for SpaceWire together with the Data-Strobe encoding and signal timing.

Skew: The difference in time between the edges of two signals which should ideally be concurrent.

Source: The node or unit that is sending a packet.

TBA: To Be Added

TBC: To Be Confirmed

TBD: To Be Determined

Transmission Medium: The medium over which data is transferred – screened twisted pair cables.

Unit: A box, board or sub-system (that may have one or more SpaceWire interfaces).

3. OVERALL DESCRIPTION

This section provides an overview of the SpaceWire standard giving the rationale behind key decisions made in the development of the standard.

SpaceWire is based on the “DS-DE” part of the IEEE 1355-1995 standard [RD1] combined with the TIA/EIA-644 [AD1] and IEEE-1596.3 [RD2] Low Voltage Differential Signalling (LVDS) standards. See Annex C for details of the differences between SpaceWire and IEEE 1355 and the reasons for those differences.

SpaceWire is a full-duplex, bi-directional, serial, point-to-point data link. It encodes data using two differential signal pairs in each direction. That is a total of eight signal wires, four in each direction.

3.1 PHYSICAL LEVEL

The physical level of the SpaceWire standard covers cables, connectors and EMC specification.

3.1.1 Cables

The SpaceWire cable comprises four twisted pair wires with a separate shield around each twisted pair and an overall shield.

To achieve a high data rate with SpaceWire over distances up to 10m the cable must have the following characteristics:-

- Characteristic impedance matched to the line termination impedance.
- Low signal-signal skew between each signal in a differential pair and between Data and Strobe pairs.
- Low signal attenuation.
- Low cross-talk.
- Good EMC performance.

3.1.2 Connectors

The SpaceWire connector is required to have eight signal contacts plus a screen termination contact. A nine pin micro-miniature D-type is specified as the SpaceWire connector. This type of connector is available qualified for space use.

3.1.3 EMC Specifications

The EMC specifications for SpaceWire have been derived from the EMC specifications for the Rosetta [RD13] and ENVISAT [RD14] missions. The EMC specification is given in Annex B. Initial EMC testing was performed by Patria Finavitec Oy with support from the University of Dundee. The testing covered:

- Radiated emission, electric and magnetic fields,
- Radiated susceptibility, electric and magnetic fields,
- Conducted susceptibility,
- Electro-static discharge,
- Signalling rate,
- Bit error rate,
- Fault isolation, and
- Power consumption.

A summary of the EMC test results is provided in Annex B. Full details are reported in [RD15].

3.2 SIGNAL LEVEL

The signal level part of the SpaceWire standard covers signal voltage levels, noise margins and signal encoding.

3.2.1 Signal Level and Noise Margins

Low Voltage Differential Signalling or LVDS is specified as the signalling technique to be used in SpaceWire.

LVDS uses balanced signals to provide very high-speed interconnection using a low voltage swing (350 mV typical). The balanced or differential signalling provides adequate noise margin to enable low voltages to be used in practical systems. Low voltage swing means low power consumption at high speed. LVDS is appropriate for connections between boards in a unit, and unit to unit interconnections over distances of 10m or more.

A typical LVDS driver and receiver are shown in Figure 3-1, connected by a media (cable or PCB traces) with 100 ohm differential impedance.

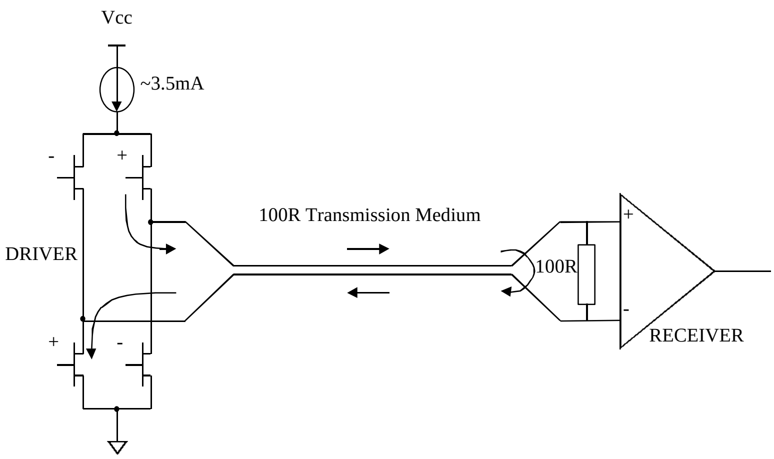

Figure 3-1 LVDS Operation

The LVDS driver uses current mode logic. A constant current source of around 3.5mA provides the current that flows out of the driver, along the transmission medium, through the 100-ohm termination resistance and back to the driver via the transmission medium. Two pairs of transistor switches in the driver control the direction of the current flow through the termination resistor. When the driver transistors marked "+" in Figure 3-1 are turned on and those marked "-" are turned off, current flows as indicated by the arrows on the diagram creating a positive voltage across the termination resistor. When the two driver transistors, marked "-", are turned on and those marked "+" are turned off, current flows in the opposite direction producing a negative voltage across the termination resistor. LVDS receivers are specified to have high input impedance so that most of the current will flow through the termination resistor to generate around ±350mV with the nominal 3.5mA current source.

LVDS has several features that make it very attractive for data signalling [RD16]:-

- Near constant total drive current (+3.5mA for logic 1 and -3.5mA for logic 0) which decreases switching noise on power supplies.

- High immunity to ground potential difference between driver and receiver - LVDS can tolerate at least ±1V ground difference.
- High immunity to induced noise because of differential signaling normally using twisted-pair cable.
- Low EMI because small equal and opposite currents create small electromagnetic fields which tend to cancel one another out.
- Not dependent upon particular device supply voltage(s).
- Simple 100 ohm termination at receiver.
- Failsafe operation - the receiver output goes to the high state (inactive) whenever
  - the receiver is powered and the driver is not powered,
  - the inputs short together,
  - input wires are disconnected.
- Power consumption is typically 50mW per driver/receiver pair for LVDS compared to 120mW for ECL.

There are two standards, which define LVDS

1. TIA/EIA-644 that defines the driver output characteristics and the receiver input characteristics only [AD1].
2. IEEE 1596.3 Low Voltage Differential Signaling (LVDS) for Scalable Coherent Interface (SCI) that defines the signalling levels used and the encoding for packet switching used in SCI data transfers [RD2].

The signal levels and noise margins for SpaceWire are defined using the TIA/EIA-644 standard [AD1] since this deals with LVDS only whereas IEEE 1596.3 [RD2] is concerned with the use of LVDS specifically for SCI.

3.2.2 Data Encoding

SpaceWire uses Data-Strobe (DS) encoding. This is a coding scheme which encodes the transmission clock with the data into data and strobe so that the clock can be recovered by simply XORing the data and strobe lines together. The data values are transmitted directly and the strobe signal changes state whenever the data remains constant from one data bit interval to the next. This coding scheme is illustrated below in Figure 3-2. The DS encoding scheme is also used in the IEEE 1355-1995 [RD1] and IEEE 1394-1995 (Firewire) standard [RD17].

The reason for using DS encoding is to improve the skew tolerance to almost 1-bit time, compared to 0.5 bit time for simple data and clock encoding.

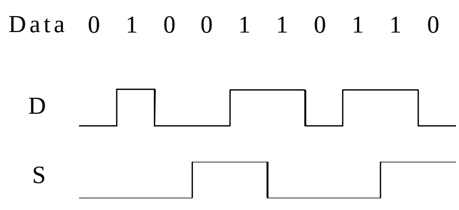

Figure 3-2 Data-Strobe (DS) Encoding

A SpaceWire link comprises two pairs of differential signals, one pair transmitting the D and S signals in one direction and the other pair transmitting D and S in the opposite direction. That is a total of eight wires for each bi-directional link.

3.3 CHARACTER LEVEL

SpaceWire employs the character level protocol defined in IEEE 1355-1995 [RD1].

There are two types of characters:-

- Data characters which hold an eight-bit data value, transmitted least-significant bit first. Each data character contains a parity-bit, a data-control flag and the eight-bits of data. The parity-bit covers the previous eight-bits of data, the current parity-bit and the current data-control flag. It is set to produce odd parity so that the total number of 1's in the field covered is an odd number. The data-control flag is set to zero to indicate that the current character is a data character.
- Control characters which hold a two-bit control code. Each control character is formed from a parity-bit, a data-control flag and the two-bit control code. The data-control flag is set to one to indicate that the current character is a control character. Parity coverage is similar to that for a data character. One of the four possible control characters is the escape code (ESC). This can be used to form longer control codes. One longer control code is specified which is the NULL code. NULL is formed from ESC followed by the flow control token (FCT). NULL is transmitted whenever a link is not sending data or control tokens to keep the link active and to support link disconnect detection.

The data and control characters are illustrated in Figure 3-3.

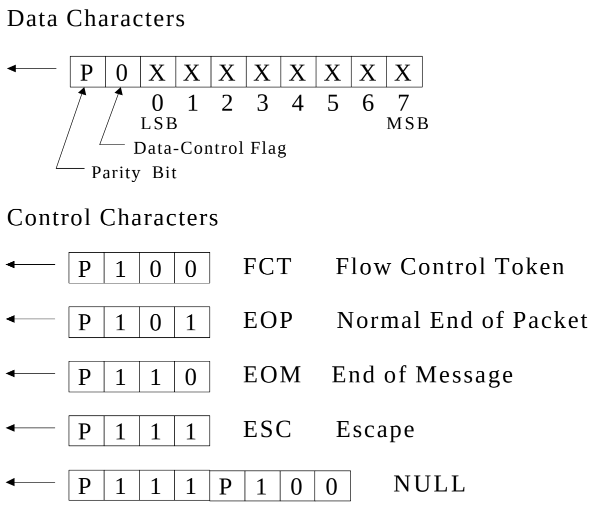

Figure 3-3 Data and Control Characters

3.4 EXCHANGE LEVEL

An exchange level protocol is proposed following that defined in IEEE 1355-1995 [RD1] which provides the following services:-

- **Initialisation:** Following reset the link output is held in the reset state until it is instructed to begin transmission. The link output then starts to transmit NULL characters and the link input is monitored for the reception of a character. Once a NULL character has been received on the input then the output can start normal operation.
- **Flow Control:** A transmitter is only allowed to transmit data characters if there is space in the receiver buffer for them. The receiver indicates that there is space for eight more data characters

by sending a flow control token (FCT). If multiple FCT's are received then it means that there is a corresponding amount of space available in the receiver buffer e.g. four FCT's means that there is room for 32 data characters.

- **Detection of Disconnect Errors:** Link disconnection is detected when following reception of a data bit no new data bit is received within a link disconnect timeout window (850 nsec). Once a disconnection error has been detected the link attempts to restart (see below).
- **Detection of Parity Errors:** Parity errors occuring within a data or control character are detected when the next character is sent, since the parity bit for a data or control token is contained in the next character. Once a parity error has been detected the link will attempt to restart (see below).
- **Link Restart:** Following an error or reset the link attempts to re-synchronise and restart using an "exchange of silence" protocol (see Figure 3-4). The end of the link that is either reset or that finds an error, ceases transmission. This is detected at the other end of the link as a link disconnect and that end stops transmitting too. The first link resets its input and output for 6.4 usec to ensure that the other end will detect the disconnect. The other end will also wait for 6.4 usec after ceasing transmission. Each link then waits a further 12.8 usec before starting to transmit. These periods of time are sufficient to ensure that the two links are resynchronised and ready to receive data.

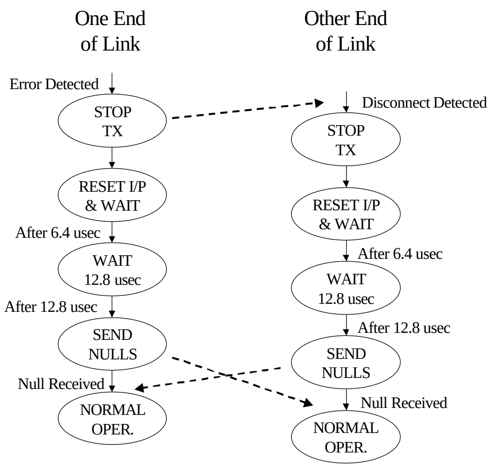

Figure 3-4 Link Restart

3.5 PACKET LEVEL

The packet level protocol follows the packet level protocol defined in IEEE-1355. It defines how message is delivered from source to destination. The format of a packet is illustrated in Figure 3-5.

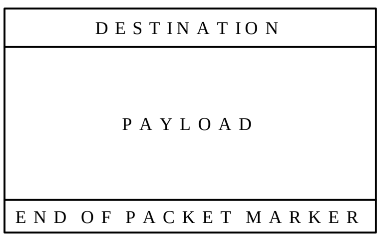

Figure 3-5 Packet Format

The "Destination" is a list of one or more bytes that represent the destination identity. This list of bytes represents either the identity code of the destination node or the path that the packet will take to get to the destination node.

The “Payload” is the data or message to be transferred from source to destination. The format of payload is not defined by the SpaceWire standard at present although extensions to support fault tolerance are being considered (see section 8).

The “End of Packet Marker” is used to indicate the end of a packet. Two end of packet markers are defined.

1. EOP Normal end_of_packet marker - indicates end of packet

2. EOM End_of_message marker - indicates that the packet is the last packet in a message.

Since there is no start of packet marker, the first data character following an end_of_packet marker (either EOP or EOM) is regarded as the start of the next packet.

The packet level protocol provides support for packet routing via wormhole routing switches [RD18].

4. PHYSICAL LEVEL

The physical level provides the actual interface between nodes including both the mechanical and electrical interface. This level of the standard covers:-

- Cable construction
- Connectors
- Cable assemblies, and
- PCB / backplane tracking.

4.1 CABLES

The SpaceWire cable shall be constructed according to ESA/SCC Generic Specification No. 3902 [AD2] and the specific details given below.

The SpaceWire cable shall comprise four twisted pair wires with a separate shield around each twisted pair and an overall shield as illustrated in Figure 4-1.

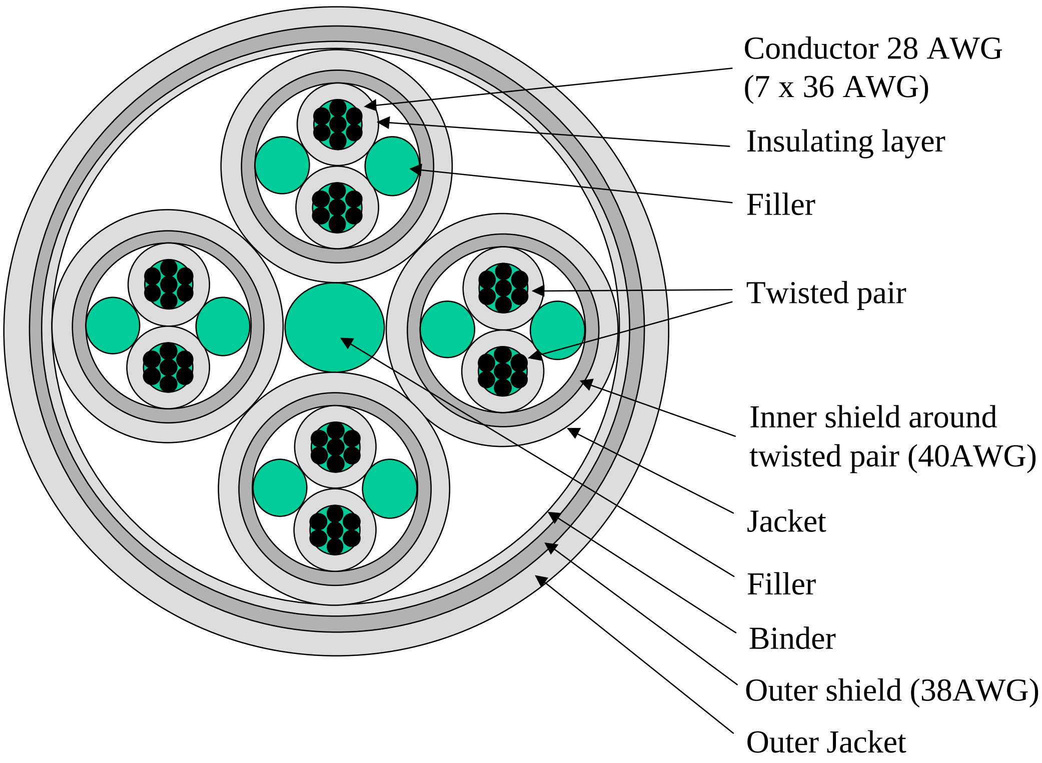

Figure 4-1 SpaceWire Cable Construction

4.1.1 Inner Conductors

4.1.1.1 Conductor

Each signal wire shall be 28 AWG, constructed from seven strands of 36 AWG silver-coated, high-strength copper alloy. The thickness of the silver coating shall be 2.0 ìm minimum.

4.1.1.2 Tensile Characteristics

The minimum elongation of each strand shall be 6.0%.

The tensile strength of each strand shall be at least 35kg/mm^2.

4.1.1.3 Insulator

Each signal shall be insulated using expanded, microporous PTFE with only those additives necessary for processing and pigmentation.

4.1.1.4 Insulator Colour

The insulator around the signal wires shall be white.

4.1.1.5 Electrical Characteristics

The maximum D.C. resistance of the inner conductor shall be 256 ohm/km (TBC).

4.1.2 Twisted Pair

4.1.2.1 Lay Length

The lay of length of the two insulated conductors comprising a differential signal pair shall not be less than 12 times and not more than 16 times the outside diameter of the unshielded twisted pair.

4.1.2.2 Fillers

Fillers shall be used with the differential signal pairs so as to ensure a smooth and uniform diameter under the shielding in order to contribute to a uniform impedance over the cable.

4.1.2.3 Filler Material

The filler material as used for the differential signal pairs shall be expanded microporous PTFE with only those additives necessary for processing.

4.1.2.4 Construction of Filler

The filler material shall be extruded or wrapped from tapes to a diameter of 1.0mm TBC.

4.1.2.5 Shield

Each differential signal pair shall be shielded by a braided shield. The braided shield type shall be of push-back type and provide not less than 90% coverage.

4.1.2.6 Shield Wire Size

The shield wire size shall be 40 AWG.

4.1.2.7 Shield Material

All strands used in the manufacture of the braided shield shall be silver-coated, soft or annealed oxygen-free high conductivity copper. The thickness of silver shall be 2.5 microns minimum. Any strand shall show an elongation of 10% minimum.

4.1.2.8 Protective Sheath

The protective sheath for the shielded differential signal pairs shall be a layer of extruded fluorpolymer PFA with only those additives necessary for processing and pigmentation.

4.1.2.9 Protective Sheath Wall Thickness

The wall thickness of the protective sheath for the shielded differential signal pair shall be 0.15mm nominal.

4.1.2.10 Protective Sheath Colour

The jacket colour of the differential signal pairs shall be white.

4.1.2.11 Characteristic Impedance

The characteristic impedance of each differential signal pair shall be 100 ± 6 ohm differential impedance.

4.1.2.12 Skew

The skew between each signal in each differential signal pair shall be less than 0.1nsec/m (TBC).

4.1.3 Complete Cable

4.1.3.1 Construction

Four sets of differential signal pairs shall be twisted together not less than 12 times and not more than 16 times of the outside diameter of a shielded and jacketed differential signal pair.

4.1.3.2 Filler

A filler shall be used in the centre of the four differential signal pairs so as to ensure a smooth and uniform diameter under the shielding in order to contribute to a uniform impedance over the cable.

4.1.3.3 Filler Material

The filler material as used for the differential signal pairs shall be expanded microporous PTFE with only those additives necessary for processing.

4.1.3.4 Construction of Filler

The filler material shall be extruded or wrapped from tapes to a diameters of TBD mm.

4.1.3.5 Binder

A binder shall be applied over the four differential signal pairs and central filler to keep the signal pairs and filler together in a fixed position.

4.1.3.6 Binder Material

The material shall be virgin, wrapped, expanded microporous PTFE with only those additives necessary for processing.

4.1.3.7 Binder Construction

The material shall be wrapped with an overlap of 50% maximum.

4.1.3.8 Outer Shield

The set of four jacketed and screened differential signal pairs shall be shielded by an outer braided shield. The braided shield type shall be of push-back type and provide not less than 90% coverage.

4.1.3.9 Outer Shield Wire Size

The shield wire size shall be 38 AWG.

4.1.3.10 Outer Shield Material

All strands used in the manufacture of the braided shield shall be silver-coated, soft or annealed oxygen-free high conductivity copper. The thickness of silver shall be 2.5 microns minimum. Any strand shall show an elongation of 10% minimum.

4.1.3.11 Shield Isolation

The twisted pair shields shall NOT make contact with one another NOR with the outer shield.

4.1.3.12 Outer Jacket

The outermost jacket over the four twisted screened and jacketed differential signal pairs shall be a layer of extruded Fluoropolymer PFA with only those additives necessary for processing and pigmentation.

4.1.3.13 Outer Jacket Wallthickness

The wall thickness of the jacket for the shielded differential signal pair shall be 0.25mm nominal.

4.1.3.14 Jacket Colour

The jacket shall be white in colour.

There shall be NO identifying marking on the cable jacket. Applying pressure to the cable during the marking process can adversely affect the electrical properties of the cable.

4.1.3.15 Signal Skew

The skew between the differential signal in one differential signal pair and the differential signal in each other differential signal pair within the cable shall be less than 0.1nsec/m (TBC).

4.1.4 Cable Physical Parameters

4.1.4.1 Cable Diameter

The outside diameter of the complete cable shall be less than 7mm (TBC).

4.1.4.2 Cable Minimum Bend Radius

The minimum bend radius of complete cable shall be less than 50mm (TBC).

4.1.4.3 Adhesion of Inner Conductor

The minimum stripping force shall be 1.0 N.

4.1.4.4 Cable Weight

The maximum weight of the SpaceWire cable shall be 60 g/m (TBC).

4.1.4.5 Cable Maximum Ratings

The maximum ratings defined in Table 4-1 shall be met.

Table 4-1-SpaceWire Cable Maximum Ratings

No.	Characteristics	Symbol	Maximum Ratings	Unit	Remarks
1	Operating Voltage (Continuous)	V_{op}	200	V_{rms}	
2	Current	I	1.5	A	
3	Operating Frequency	F_M	400	Mbits/sec	
4	Operating Temperature Range	T_{op}	-200 to +180	C	T_{amb} NOTE 1
5	Storage Temperature Range	T_{stg}	-200 to +180	C	
NOTE 1: The above specified current will generate a temperature rise of approximately 50C above ambient temperature in a vacuum environment. Precautions shall be taken to prevent the total temperature of the wire (ambient plus rise) exceeding the continuous operating temperature of the wire.					

4.2 CONNECTORS

The SpaceWire connector shall be a nine contact micro-miniature D-type as defined in ESA/SCC Detailed Specification No. 3401/029.

4.2.1 Sockets

Sockets shall be used on board and unit assemblies.

The socket used on board and unit assemblies should be of type 340102901B 9SFR114 (TBC) as defined in ESA/SCC Detailed Specification No. 3401/029 [AD3]. Note: These sockets with flying leads are recommended for connection to a PCB rather than PCB mounting connectors to improve mechanical shock and vibration resistance of the unit.

4.2.2 Plugs

Plugs shall be used on cable assemblies.

The plugs used on cable assemblies should be of type 340102901B 9P with the SpaceWire conductors directly crimped to the crimp-type contacts before assembly and potting of the plug (see section 4.3 for SpaceWire cable assembly details).

4.2.3 Connector Pin-Out

The connector pin-out given in Table 4-2 and Figure 4-2 shall be used.

Table 4-2 Connector Pin-Out

Pin Number	Signal Name
1	DIN+
2	SIN+
3	Inner Shield
4	SOUT-
5	DOUT-
6	DIN-
7	SIN-
8	SOUT+
9	DOUT+

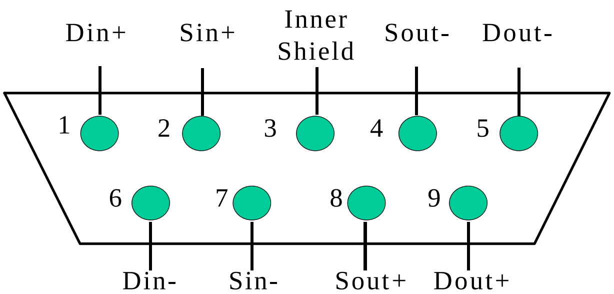

Figure 4-2 SpaceWire Connector Pin-Out

The inner shield connection is for connection to the inner shield of the SpaceWire cable. This should be connected to signal ground (possibly via a parallel resistor and capacitor) according to the EMC requirements or guideline for a particular mission. See section 4.3 for the cable connection to pin 3 of the connector.

4.2.4 Flying Lead Connectors

Flying lead connectors are recommended for connection to a PCB. Flying lead connectors used for connection to a PCB should have all the leads cropped to the same short length (less than 25mm) and the wires comprising the differential signal pairs should be twisted together. This will help to minimise the discontinuity in impedance caused by the connector.

4.2.5 PCB Mounting Connectors

PCB mounting right-angled connectors are not recommended. If a PCB mounting right-angled connector has to be used care must be taken to compensate in the PCB layout for the different lengths of signal path though the connector. The topmost row of pins on the right-angled connect have longer leads than the bottom row. Signals connected to the top row must be given correspondingly shorter PCB track lengths than tracks going to the bottom row. Track length compensation must be performed at the connector end of the PCB tracks to maintain the differential signal across the PCB.

4.3 CABLE ASSEMBLY

Cable assemblies shall consist of two identical plug connectors joined by a length of cable.

4.3.1 Cable Length

The maximum length of the cable assembly should normally be 10m to ensure that the end to end skew and jitter introduced by the cable assembly does not exceed the maximum budget for the cable. Longer length cables may be used at slow data rates provided that the signal attenuation and system jitter and skew limits are not violated at the operating data rate (see section 5.6).

4.3.2 Cable Connections

The connector contacts shall be terminated as shown in Figure 4-3. The cable signal wires cross over to achieve a transmit to receive interconnection, i.e. DOUT+ is connected to DIN+ etc.

The individual shields of the differential signal pairs carrying the output signals DOUT+, DOUT- and SOUT+ and SOUT- shall be connected together and to pin 3 of the connector. The shields are terminated at the end of the cable that the signals are being driven, following good EMC practice. In this way two of the differential pairs are connected at one end of the cable and the remaining two at the other end. A symmetrical arrangement results, avoiding the problem of having to know which end of the cable is which during installation.

A metal shell shall be used for each connector to provide necessary shielding of the connector.

The outer shield of the cable shall be bonded to the connector shell via a low impedance connection.

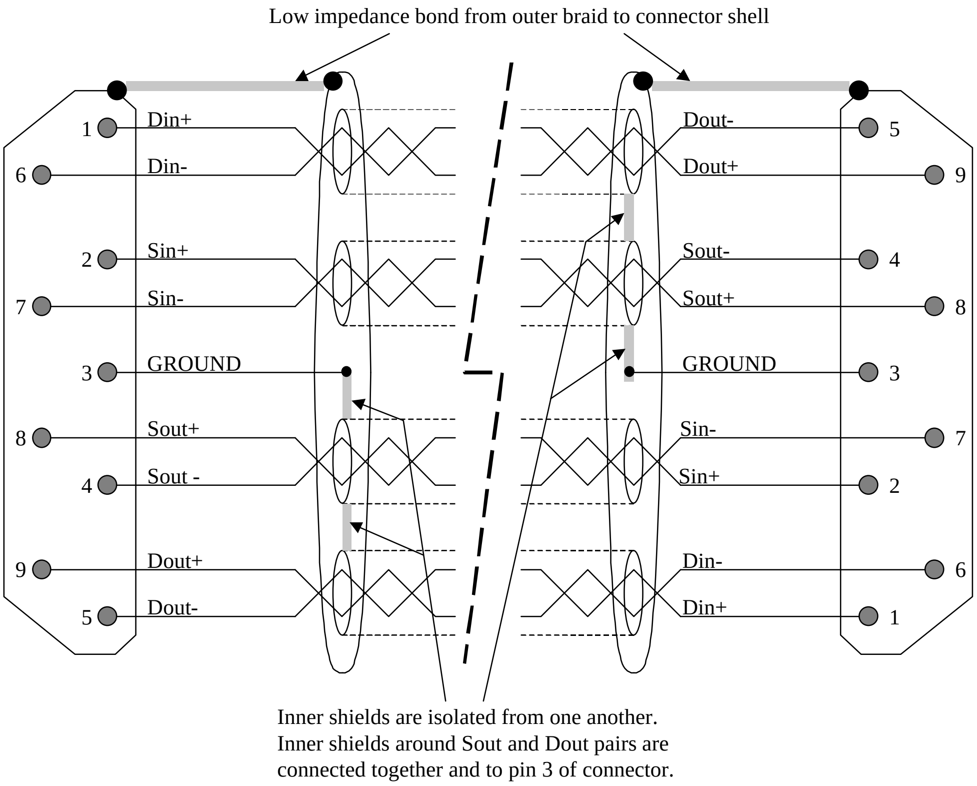

Figure 4-3 SpaceWire Cable Assembly

4.3.3 Connector Back Shell

TBD

4.4 PCB/BACKPLANE LINK

As well as routing SpaceWire signals through a cable the signals may also be transmitted across a PCB or along a backplane.

4.4.1 Differential Signal Pairs

4.4.1.1 Differential Impedance

Differential pair signals shall run on a pair of close, parallel PCB tracks with a differential impedance of 100 ± 6 ohms. The required differential impedance may be achieved by adjusting the track thickness, width, separation and height above the ground plane.

4.4.1.2 Difference in Track Length

The difference in track length between the two signals from a differential pair shall be less than 5% of the track length and no more than 5mm. This is to avoid skew between the two parts of the differential signal.

4.4.2 Differential Signals

4.4.2.1 Skew

The skew introduced between the data and strobe (D and S) signals must be minimised. For PCB tracks this is controlled by making the tracks all close to the same length. The difference in track length between the data and strobe signals shall be less than 5% of the track length and no more than 5mm.

4.4.3 Example Track Dimensions

4.4.3.1 Microstrip

TBA

4.4.3.2 Stripline

TBA

5. SIGNAL LEVEL

5.1 LVDS

SpaceWire shall use Low Voltage Differential Signalling (LVDS) with electrical characteristics as defined in the TIA/EIA-644 standard "Electrical Characteristics of Low Voltage Differential Signalling (LVDS) Interface Circuits" [AD1].

The signalling levels used by LVDS are illustrated in Figure 5-1.

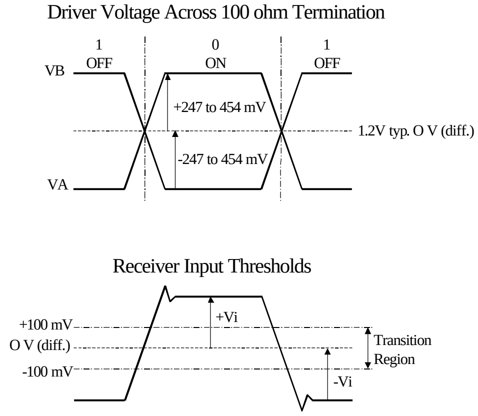

Figure 5-1 LVDS Signalling Levels

5.2 FAILSAFE OPERATION OF LVDS

When any of the following fault conditions occur the receiver outputs shall not oscillate and shall be locked to logic high provided that a noise threshold of 10 mV (TBC) is not exceeded at the receiver input.

1. Driver not powered
2. Driver disabled
3. Driver not connected to receiver
4. Interconnecting cable open circuit

When the receiver inputs are shorted together the receiver output shall remain in a logic high state.

When the driver is not powered its output should be high impedance i.e. > TBD kohm.

When the receiver is not powered its input should be high impedance i.e. > TBD kohm.

5.3 SIGNAL CODING

5.3.1 DS

SpaceWire shall use Data-Strobe (DS) encoding as defined in section 5.3.5 of IEEE 1355-1995 [RD1]. Note DS encoding is also defined in IEEE 1394-1996 [RD17]. See Annex C for details of the differences between SpaceWire and IEEE 1355 and the reasons for those differences.

The data bit stream to be transmitted shall be encoded using two signals Data and Strobe. The Data signal shall follow the required data bit stream i.e. be high when the data bit is 1 and low when the data bit is 0. The Strobe signal shall change state whenever the Data does not change from one bit to the next.

The DS encoding is illustrated in Figure 5-2.

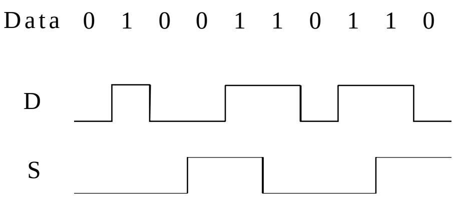

Figure 5-2 Data-Strobe (DS) Encoding

5.3.2 Simultaneous Transition on Data and Strobe Signals

Simultaneous transitions on the data and strobe lines are not part of the normal operation of SpaceWire. They can occur, however, either when a SpaceWire cable is plugged in while the transmitter is trying to make a connection, or when the LVDS driver or receiver circuits are enabled while the transmitter is trying to make a connection.

The SpaceWire receiver shall be tolerant of simultaneous transitions on the data and strobe lines i.e. the receiver shall not hang-up. Data corruption following simultaneous transitions on the data and strobe lines is to be expected.

5.4 DIFFERENTIAL DS

SpaceWire shall use low voltage differential signalling (LVDS) for the data and strobe signals.

5.5 SPACEWIRE LINK

A SpaceWire link shall comprise two pairs of differential signals, one pair transmitting the D and S signals in one direction and the other pair transmitting D and S in the opposite direction. That is a total of eight wires for each bi-directional link.

5.6 DATA RATE

5.6.1 Minimum Data Rate

The minimum data rate is the lowest data rate at which a SpaceWire link can operate. The minimum data rate is set by the disconnect time-out (section 7.7.1) to greater than 1.18 Mbps, i.e. 1/(850 nsec). The minimum data rate at which a SpaceWire link shall operate is 2 Mbps.

5.6.2 Maximum Data Rate

The maximum data rate is the highest data rate at which a SpaceWire link can operate and is defined by consideration of signal skew and jitter (see section 5.6.4).

5.6.3 Operational Data Rate

A SpaceWire link may operate at any data rate between the minimum data rate and the maximum possible data rate.

The link in one direction may operate at a different data rate to the same link in the opposite direction. Links within a system may operate at different data rates.

5.6.4 Effects of Skew and Jitter

The maximum data rate that can be achieved will be different from one system to another depending on cable length, driver/receiver technology, encoder/decoder design etc, and is limiter by skew and jitter. Figure 5-3 illustrates the effect of skew and jitter on the data and strobe signals.

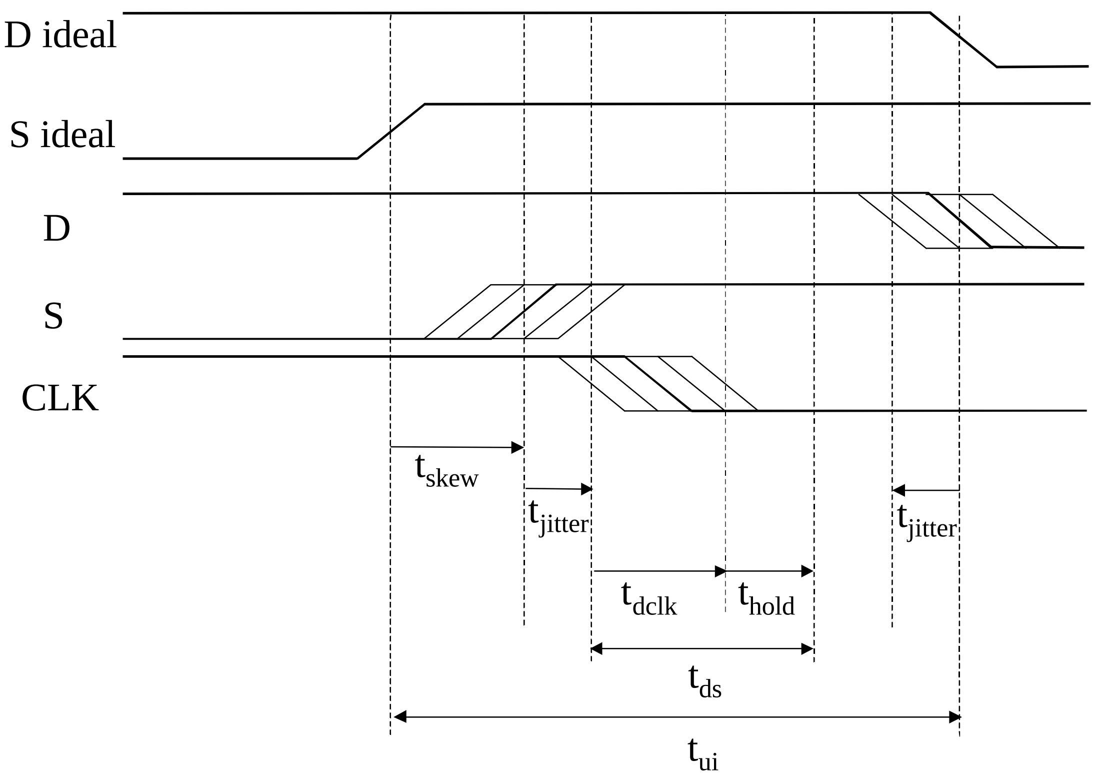

Figure 5-3 Skew and Jitter

t_{skew} is the skew between the data and strobe signals.

t_{jitter} is the jitter on the data or strobe signal. t_{jitter}data = t_{jitter}strobe since they follow identical signal paths (as close as possible).

t_{dclk} is the delay in the receiver from the edge of the data or strobe signal, through the XOR operation which produces the clock signal, to the clocking in of the data in the input flip-flop. This may be regarded as the set-up time for the data input flip-flop from the edge of the data or strobe signal.

t_{hold} is the hold time required for the data signal after the clocking of the data into the input flip-flop.

t_{ui} is the unit interval or bit period. $t_{ui} = 1/F_{op}$ where F_{op} is the link operating frequency.

The t_{dclk} and t_{hold} parameters may be combined into a minimum specification for the separation of consecutive edges on the data and strobe signals at the input to the decoder, $t_{ds} = t_{dclk} + t_{hold}$.

t_{margin} is the available margin. $t_{margin} = t_{ui} – (t_{skew} + 2*t_{jitter} + t_{dclk} + t_{hold})$.

For reliable operation the margin shall be greater than zero.

Figure 5-4 illustrates the contributors to skew and jitter in a typical system.

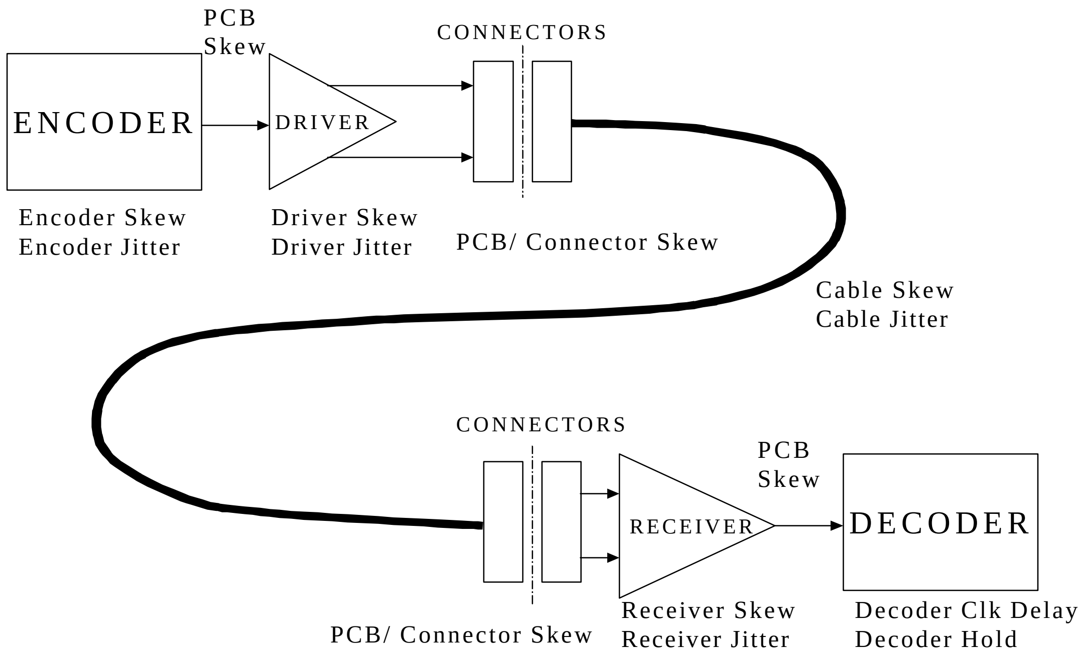

Figure 5-4 Contributors to Skew and Jitter

The following three tables (Table 5-1, Table 5-2 and Table 5-3) provide the jitter and skew budgets at three different operating frequencies used as examples (100 Mbps, 200 Mbps and 400 Mbps). The table for 400 Mbps operation contains suggested jitter and skew budgets for future systems and is provided for information only.

FIGURES IN THE FOLLOWING TABLES THAT ARE ITALIC AND UNDERLINED ARE TBC.

Table 5-1 Jitter and Skew Budget at 100 Mbps (tui = 10 ns)

	Data Jitter (ns)	Strobe Jitter (ns)	Skew (ns)	Min edge separation (ns)	Total (ns)
	t jitter	t jitter	t skew	t ds	
Encoder Skew			0.50		
Encoder Jitter	*0.50*	*0.50*			
PCB Skew			0.05		
Driver Skew			1.00		
Driver Jitter	*0.50*	*0.50*			
PCB/ Connector Skew			0.10		
Total Transmitter	**1.00**	**1.00**	**1.65**		**3.65**
Cable Jitter	*0.50*	*0.50*			
Cable Skew			1.00		
Total Cable	**0.50**	**0.50**	**1.00**		**2.00**
PCB/ Connector Skew			0.10		
Receiver Skew			1.50		
Receiver Jitter	*0.50*	*0.50*			
PCB Skew			0.05		
Decoder Clock Delay and Hold				1.00	
Total Receiver	**0.50**	**0.50**	**1.65**	**1.00**	**3.65**
Total System	**2.00**	**2.00**	**4.30**		**8.30**
Margin					1.70

Table 5-2 Jitter and Skew Budget at 200 Mbps (tui = 5 ns)

	Data Jitter (ns)	Strobe Jitter (ns)	Skew (ns)	Min edge separation (ns)	Total (ns)
	t jitter	t jitter	t skew	t ds	
Encoder Skew			0.50		
Encoder Jitter	*0.10*	*0.10*			
PCB Skew			0.05		
Driver Skew			0.07		
Driver Jitter	*0.20*	*0.20*			
PCB/ Connector Skew			0.10		
Total Transmitter	**0.30**	**0.30**	**0.72**		**1.32**
Cable Jitter	*0.50*	*0.50*			
Cable Skew			1.00		
Total Cable	**0.50**	**0.50**	**1.00**		**2.00**
PCB/ Connector Skew			0.10		
Receiver Skew			0.12		
Receiver Jitter	*0.20*	*0.20*			
PCB Skew			0.05		
Decoder Clock Delay and Hold				1.00	
Total Receiver	**0.20**	**0.20**	**0.27**	**1.00**	**1.67**
Total System	**1.00**	**1.00**	**1.99**	**1.00**	**4.99**
Margin					0.01

Table 5-3 Proposed Jitter and Skew Budgets for 400 Mbps Operation (tui = 2.5 ns)

	Data Jitter (ns)	Strobe Jitter (ns)	Skew (ns)	Min Edge Separation (ns)	Total (ns)
	t jitter	t jitter	t skew	t ds	
Encoder Skew			*0.20*		
Encoder Jitter	*0.10*	*0.10*			
PCB/ Connector Skew			0.05		
Total Transmitter	**0.10**	**0.10**	**0.25**		**0.45**
Cable Jitter	*0.35*	*0.35*			
Cable Skew (5m max. length)			0.50		
Total Cable	**0.35**	**0.35**	**0.50**		**1.20**
PCB/ Connector Skew			0.05		
Receiver Jitter	*0.10*	*0.10*			
Decoder Clock Delay and Hold				*0.50*	
Total Receiver	**0.10**	**0.10**	**0.05**	**0.50**	**0.75**
Total System	**0.55**	**0.55**	**0.80**	**0.50**	**2.40**
Margin					0.10

NOTE: The jitter and skew figures for 400 Mbps operation (Table 5-3) assume that the LVDS driver / receiver are integrated in the same package as the encoder / decoder.

The maximum data rate for a SpaceWire link must be set so that the timing margin (t_{margin}) is greater than zero.

5.6.5 Initial Operating Data Rate

After reset the SpaceWire link transmitter shall initially commence operating at a data rate of 10 ± 2 Mbps. It will operate at this rate until commanded to operate at a different data rate. This requirement is intended to provide all systems with a common, slow, initial data rate so that system operation can be validated before switching to higher and possibly widely different data rates. All SpaceWire systems will be capable of this initial slow data rate but may not necessarily be capable of higher data rates. This minimum data rate constrains the speed at which a link may reconnect either initially or after a disconnect (see section 7).

6. CHARACTER LEVEL

The character level protocol follows the DS-SE and DS-DE character level encoding given in section 5.6 of the IEEE Std 1355-1995 [RD1].

6.1 DATA

A data character shall contain a parity-bit, a data-control flag and eight-bits of data. The data-control flag shall be set to zero to indicate that the current character is a data character. The eight-bit data value shall be transmitted least-significant bit first. This is illustrated in Figure 6-1.

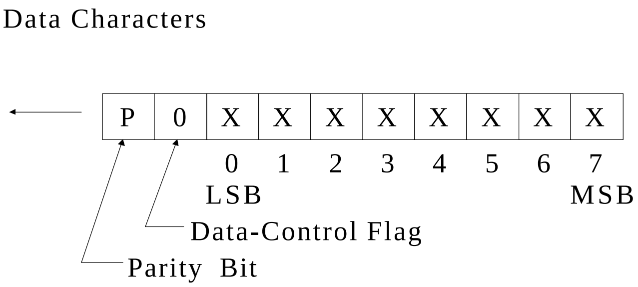

Figure 6-1 SpaceWire Data Characters

6.2 CONTROL

A control character shall be formed from a parity-bit, a data-control flag and a two-bit control code. The data-control flag shall be set to one to indicate that the current character is a control character. The different control characters are illustrated in Figure 6-2.

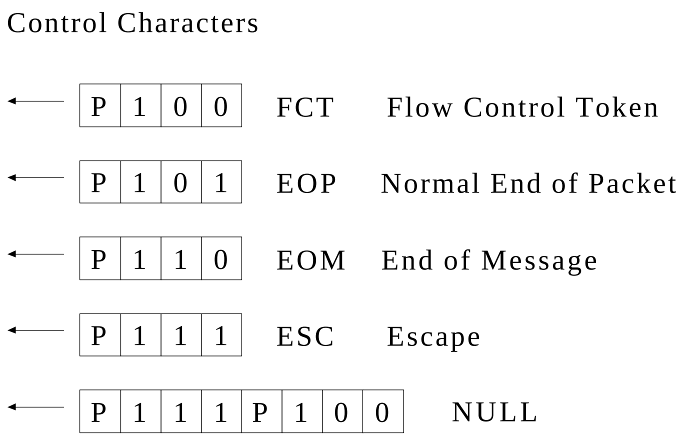

Figure 6-2 SpaceWire Control Characters

One of the four possible control characters is the escape code (ESC).

The NULL control code shall be formed from ESC followed by the flow control token (FCT). NULL shall be transmitted whenever a link is not sending data or control tokens, to keep the link active and to support link disconnect detection (see section 7).

The meaning of an escape character (ESC) which is not followed by a flow control token (FCT) shall be defined by the transaction or service layer or by the application. When an ESC occurs not as part of a NULL then it shall be passed up to the higher protocol levels (as a flag, interrupt or in some other way) to be interpreted and responded to.

6.3 PARITY FOR ERROR DETECTION

A parity bit shall be assigned to each data or control character to support the detection of transmission errors.

The parity-bit shall cover the previous eight data bits or two control bits, and the current data-control flag. It shall be set to produce odd parity so that the total number of 1's in the field covered is an odd number. The coverage of the parity bit is illustrated in Figure 6-3.

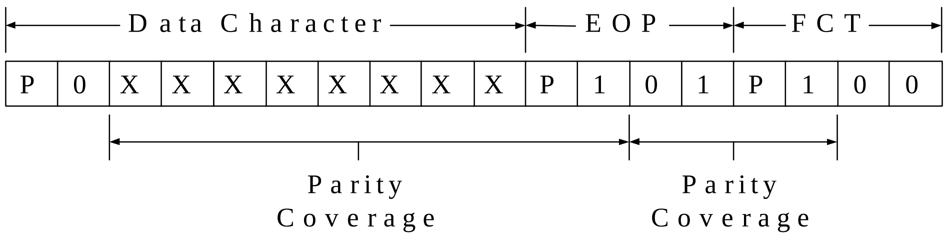

Figure 6-3 Parity Coverage

After power on the first bit to be sent is a parity bit, this bit shall be set to zero.

7. EXCHANGE LEVEL

The exchange level protocol is an improved version of the DS-SE and DS-DE exchange level protocol given in section 5.7 of the IEEE Standard 1355-1995 [RD1]. The Exchange level state machine has been modified to eliminate problems with the **ResetLinkCommand** ***and several ambiguities within the IEEE 1355-1995 standard have been resolved. See Annex C for details of the differences between SpaceWire and IEEE 1355 and the reasons for those differences.***

NOTE THAT THIS SECTION IS NOT IN ITS FINAL FORM AND SHOULD BE REGARDED AS A DISCUSSION DOCUMENT. AT PRESENT IT AIMS TO EXPLAIN THE OPERATION OF THE EXCHANGE LEVEL RATHER THAN TO SPECIFY IT FORMALLY.

7.1 LINK-CHARACTERS AND NORMAL-CHARACTERS

At the exchange level, data and control characters are separated into two types L-char and N-char.

Link-characters (L-char) are those that are used in the exchange level and which are not passed on to the packet level. The flow control token (FCT) and NULL are Link-characters.

Normal-characters (N-char) are all the other characters including data characters, end-of-packet markers (EOP and EOM) and escape sequences (ESC). These characters are all passed up to the packet level either as flags, interrupts or in some other way.

It shall be possible to interleave Link-characters in a stream of Normal-characters.

Normal-characters from one packet shall NOT be interleaved with Normal-characters from another packet.

7.2 FLOW CONTROL

The flow of data across a link must be controlled to avoid receiver input overflow and subsequent loss of data. Data flow shall be controlled using flow control tokens sent by the receiver to the transmitter. The flow control token or flow control token (FCT) is defined in section 6.2.

Each flow control token sent out by a receiver shall indicate that there is space for eight more Normal-characters in the receiver buffer. A receiver with a large buffer should send out multiple FCTs according to the amount of space in the buffer. For each FCT sent the receiver shall reserve room in its buffer to accommodate eight Normal-characters.

The transmitter shall not transmit any data until it has received one or more FCTs to indicate that the receiver is ready. The transmitter shall keep a credit count of the number of Normal-characters that it has been authorised to send. Each time the transmitter receives a FCT then it shall increment this credit count by eight. Whenever the transmitter sends an N-char it shall decrement the credit count by 1. If the credit count reaches zero the transmitter shall cease sending Normal-characters until it receives another FCT increasing the credit count again to eight. When the credit count is zero the transmitter shall continue to send Link-characters (NULLs or FCTs).

After a reset the initial value of the credit count shall be zero.

If the credit count is at its maximum value when a FCT is received that FCT shall be ignored.

Consider the following situation. A link interface has a credit count of a limited size (e.g. max. credit count = 7). The link interface that it is attached to has a larger buffer than can be handled by the limited credit count size (e.g. receive buffer > 8*7 = 56). In this case any additional FCTs that are sent to the transmitter will be ignored. Thus if a FCT is received when the credit count is at its maximum value the credit count will not be incremented. The system will operate as if the receive buffer size is eight times the maximum credit count (e.g. 8*7 =56).

7.3 ENCODER/DECODER BLOCK DIAGRAM

An example block diagram of a SpaceWire Encoder/Decoder is illustrated in Figure 7-1 below.

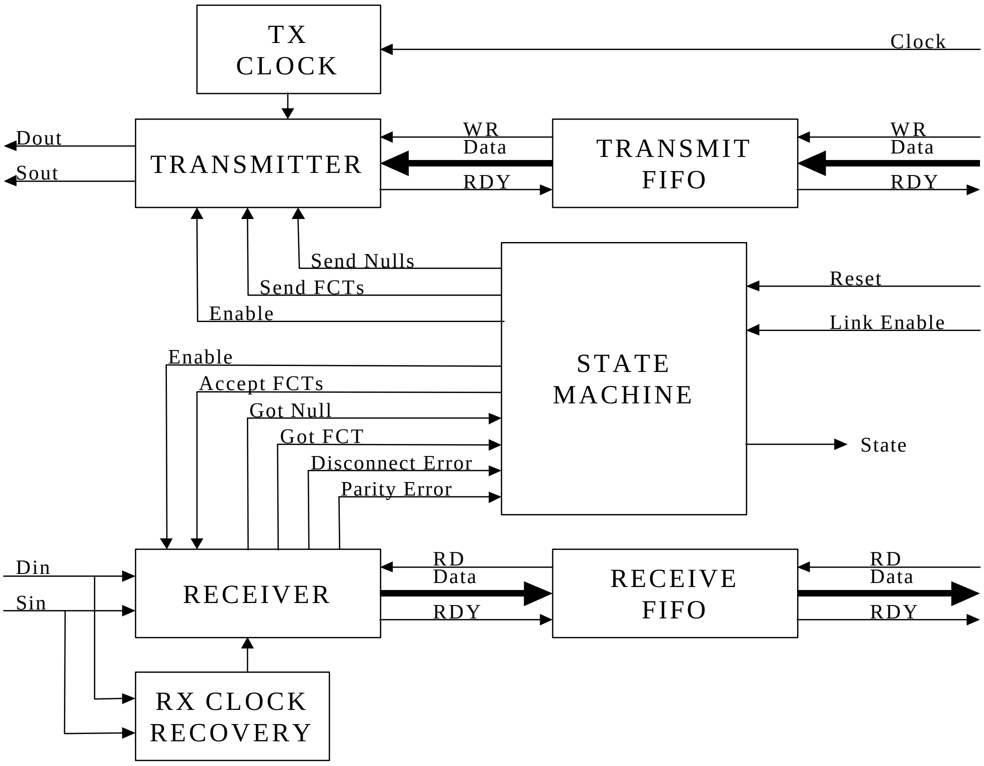

Figure 7-1 Example SpaceWire Link Interface Block Diagram

7.3.1 Transmitter

The Transmitter is responsible for encoding data and transmitting it using the DS encoding technique. It receives its data from the Transmit FIFO. If there is no data to transmit the Transmitter will send NULL characters. The Transmitter is only allowed to send data if the Receiver at the other end of the link has room in its input buffer (Receive FIFO). This is indicated by the link interface at the other end of the link sending a FCT, indicating that there is space for another 8 data characters in its Receive FIFO. The Transmitter is responsible for keeping track of the FCTs received and the number of data characters sent to avoid input buffer overflow at the other end of the link. To do this the Transmitter holds a credit count of the number of characters it has been given permission to send. The transmitter is also responsible for sending FCTs whenever the local Receiver has space for eight more data characters.

7.3.2 Transmit Clock

The Transmit Clock is responsible for producing the clock signals needed by the transmitter. The transmit-clock signals are typically derived from the local system clock or from a special transmit clock circuit.

7.3.3 Transmit FIFO

The Transmit FIFO provides the interface between the Transmitter and the local system source of data. The local system writes data into the FIFO at any time provided the FIFO is not full.

7.3.4 Receiver

The Receiver is responsible for decoding the DS signals (Din and Sin) to produce a sequence of data characters that are written into the Receive FIFO. It also receives NULL characters, FCTs and other control characters (EOP, EOM, ESC). NULL characters simply represent an active link and are otherwise ignored. When an FCT is received the Receiver must inform the Transmitter so that it can update its credit count accordingly. All other control characters received are flagged to the host system.

7.3.5 Receive Clock Recovery

The receive-clock is recovered by simply XORing the received data and strobe signals together. The Receive Clock Recovery circuit provides all the clock signals needed by the receiver.

7.3.6 Receive FIFO

The Receive FIFO provides the interface between Receiver and the local system destination for the data. As data is received by the Receiver it is written into the Receive FIFO. The local host system can then access this data as required. The Receive FIFO is responsible for informing the Transmitter whenever there is space for eight more data characters within the Receive FIFO so that the Transmitter can send an FCT to the interface at the other end of the link.

7.3.7 State Machine

The state machine controls the overall operation of the link interface. It provides link initialisation, normal operation and error recovery services. The operation of the state machine is discussed in detail in the remainder of section 7.

7.4 STATE MACHINE

The complete state transition diagram for the SpaceWire link interface is illustrated in Figure 7-2 below.

The state diagram notation is explained in section 1.7.3.

The definition of states is represented in section 7.4.1.

The definition of transition events and conditions is explained in section 7.4.2.

The operation of the state machine is explained in sections 7.5 through 7.11.

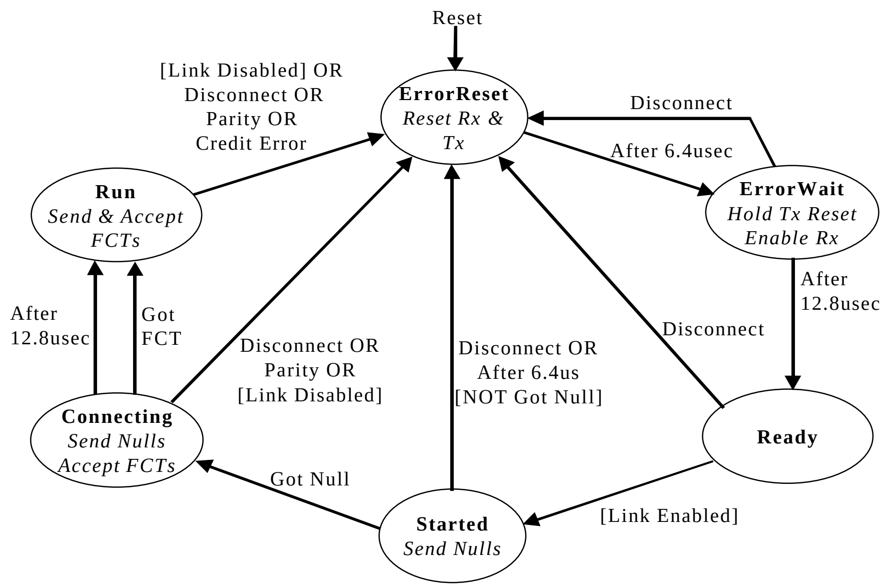

Figure 7-2 State Diagram for SpaceWire Link Interface

7.4.1 Definition of States

In this section the states represented in Figure 7-2 are described.

7.4.1.1 ErrorReset

The *ErrorReset* is entered after a system reset, after link operation has been terminated for any reason or if there is an error during link initialisation. In the *ErrorReset* state the Transmitter and Receiver are both halted and the Transmit FIFO and Receive FIFO are both flushed. The *ErrorReset* state is left unconditionally after a delay of 6.4 usec (nominal) and the state machine moves to the *ErrorWait* state.

7.4.1.2 ErrorWait

The *ErrorWait* state is entered only from the *ErrorReset* state. In the *ErrorWait* state the Receiver is enabled but the Transmitter is disabled. This allows the Receiver to start the disconnection detection mechanism (after registering a transition on the D or S line) and to begin looking for the arrival of a NULL character. The *ErrorWait* state is left unconditionally after a delay of 12.8 usec (nominal) and the state machine moves to the *Ready* state. If a disconnection error is detected while in the *ErrorWait* state the state machine moves back to the *ErrorReset* state.

7.4.1.3 Ready

The *Ready* state is entered only from the *ErrorWait* state. In the *Ready* state the link interface is ready to initialise as soon as it is allowed to do so. The Transmitter remains disabled and Receiver is enabled listening for NULL characters. The state machine waits in the *Ready* state until the [Link Enabled] guard becomes true (see section 7.9) and then it moves on into the *Started* state. If a disconnection error is detected while in the *Ready* state the state machine moves back to the *ErrorReset* state.

7.4.1.4 Started

The *Started* state is entered from the *Ready* state when the link interface is enabled. In the *Started* state the state machine begins making a connection with the link interface at the other end of the link by sending NULL characters. On entering the *Started* state the Transmitter is instructed to send NULL characters and a 6.4 usec timeout timer is started. The state machine remains in the *Started* state until either a NULL character is received (Got Null) or the 6.4 usec timeout timer expires or a disconnect error is detected. If a NULL is received or was received in the *ErrorWait*, *Ready* or *Started* state then the state machine will move to the *Connecting* state. If the timeout timer expires or a disconnect error is detected then the state machine will move to the *ErrorReset* state.

7.4.1.5 Connecting

The *Connecting* state is entered from the *Started* state after a NULL character has been received. The *Connecting* state is entered when the link interface has received a NULL character so that is has achieved partial connection across the link. The link interface has to wait in the *Connecting* state for the other end of the link to receive NULL from this end. It does this by waiting long enough for the other end of the link to disconnect if it has not received a NULL character. On entering the *Connecting* state a 12.8 usec timeout timer is started. In the *Connecting* state the Transmitter continues to send NULL characters and the Receiver is enabled to receive FCTs. The state machine waits in the *Connecting* state until either the 12.8 usec timeout occurs, a FCT is received, a disconnect error is detected, a parity error occurs or the link interface is disabled. If the 12.8 usec time out expires or a FCT is received the state machine moves to the *Run* state, otherwise it moves to the *ErrorReset* state.

7.4.1.6 Run

The *Run* state is entered only from the *Connecting* state. When in the *Run* state the link interface is operating normally. The Transmitter is enabled to send FCTs and the Receiver to accept FCTs. The *Run* state is left and the *ErrorReset* state entered when the link interface is disabled or if an error occurs (disconnect, parity or credit error).

7.4.2 Definition of Transition Events and Conditions

In this section the transitions represented in Figure 7-2 are described.

7.4.2.1 Reset

Reset represents power on reset, other hardware reset or software commanded reset.

7.4.2.2 After T usec

After 6.4 usec or after 12.8 usec represents a delay of the specified time measured from when the current state is entered. The actual time intervals are nominal delays.

7.4.2.3 [Link Enabled]

[Link Enabled] is a condition that must be met for the transition to occur (i.e. a guard). [Link Enabled} can be set true by software or hardware (see section 7.9).

7.4.2.4 Got Null

Got Null means that a NULL character has been received. NULL detection could be enabled only in the *Started* state or throughout the *ErrorWait, Ready and Started* states (TBD).

7.4.2.5 Got FCT

Got FCT means that a FCT has been received. FCTs are only recognised in the *Connecting* and *Run* states.

7.4.2.6 [Link Disabled]

[Link Disabled] is a condition set by external hardware or software in order to disable and stop the link interface.

7.4.2.7 Disconnect

Disconnect is an error condition asserted when the length of time since the last transition on the D or S lines was longer than 850 nsec (nominal) ago. The disconnect detection mechanism is activated after leaving the *ErrorReset* state as soon as the first edge is detected on the D or S line.

7.4.2.8 Parity

The parity event occurs if a parity error is detected (see section 6.3). Parity detection is enabled only in the *Connecting* and *Run* states.

7.4.2.9 Credit Error

Credit error occurs when the Receive FIFO is written to when it is already full. This should never occur and indicates that an undetected error has occurred affecting the credit count.

7.5 LINK INITIALISATION

This section explains how the state diagram given in section 7.4 handles link initialisation, going from the reset of one end of a link through to the link operating normally sending data in both directions. The basic state diagram is illustrated in Figure 7-3 below.

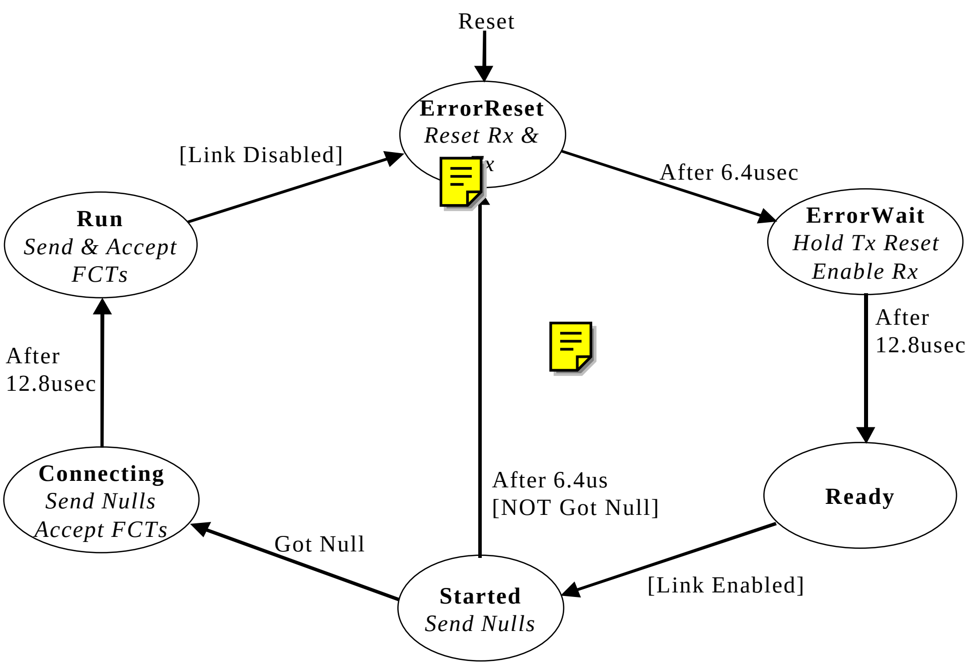

Figure 7-3 Basic State Diagram for SpaceWire Link Interface

After a link interface (one end of a link) has been reset, it enters the *ErrorReset* state where the transmitter and receiver are reset. The transmitter reset is a controlled reset, resulting first in the transmitter stopping transmission followed by resetting of the data signal and then the strobe signal. This sequence avoids the simultaneous transition of both data and strobe signals.

The link interface will remain in the *ErrorReset* state for approximately 6.4 usec and then move to the *ErrorWait* state. In the *ErrorWait* state the transmitter remains disabled, but the receiver is enabled so that it can begin searching for NULL characters.

The link interface remains in the *ErrorWait* state for 12.8 usec and then moves into the *Ready* state. The significance of the 6.4 usec and 12.8 usec delays will become apparent when error detection is discussed in section 7.7.

The link interface may be enabled in many possible ways, for example, by software command, automatically when the receiver detects a NULL character, or the link may be permanently enabled. When a link interface is enabled the *[LinkEnabled]* condition becomes true. The link interface will move from the *Ready* state to the *Started* state as soon as the link is enabled.

In the *Started* state the link interface instructs the transmitter to start sending NULL characters. It will remain in this state until the receiver detects that a NULL character has been received over the link or until a connection timeout has expired. The connection timeout is set to a nominal 6.4 usec since this period has to be generated for the *ErrorReset* state timeout. If a NULL character is received then the link interface will move to the *Connecting* state otherwise it will move (after 6.4 usec) to the *ErrorReset* state. In the latter case the link interface will go through the reset sequence and attempt to make a connection again a short time later.

In the *Connecting* state the link interface waits for 12.8 usec and then moves on to the *Run* state. This delay is used to make sure that the interface at the other end of the link is operating correctly (see section 7.8.2).

When the link enters the *Run* state it starts normal operation, sending and receiving data and control characters. It remains in the *Run* state until the link is disabled (i.e. is no longer enabled). The link interface then moves through the reset sequence (*ErrorReset*, *ErrorWait*, *Ready*) and stays in the ready state until the link is enabled once more.

The following table illustrates a typical initialisation sequence. Link interface A is at one end of the link and link interface B is at the other end.

Table 7-1 EXAMPLE: typical initialisation sequence.

Link Interface End A	Link Interface End B	Event/Condition Causing Transition
ErrorReset	*ErrorReset*	End A times out after 6.4 usec and moves to the *ErrorWait* state.
ErrorWait	*ErrorReset*	End B times out after 6.4 usec and moves to the *ErrorWait* state.
ErrorWait	*ErrorWait*	End A times out after 12.8 usec and moves to the *Ready* state.
Ready	*ErrorWait*	End A is link enabled so moves to the *Started* state.
Started Sending Nulls	*ErrorWait*	End B detects NULL sent from end A. This is registered as Got Null by end B. There is no state change.
Started Sending Nulls	*ErrorWait*	End B times out after 12.8 usec and moves to the *Ready* state.
Started Sending Nulls	*Ready*	End B is link enabled so moves straight to the *Started* state.
Started Sending Nulls	*Started* Sending Nulls	End B sends a NULL. It has already detected a Null (Got Null) so can now move to the *Connecting* state.
Started Sending Nulls	*Connecting*	End A detects NULL sent from end A (Got Null) and can move to the *Connecting* state.
Connecting	*Connecting*	End B times out after 12.8 usec and moves to the *Run* state.
Connecting	*Run*	End A times out after 12.8 usec and moves to the *Run* state.
Run	*Run*	Both ends are in the *Run* state and begin normal operation.

7.6 NORMAL OPERATION

Once normal operation is started a link shall send one or more FCTs to indicate that there is space available in the link receiver. The link shall not send any normal-characters until it receives one or more FCTs. Whenever the link's flow control credit count is above zero it may send Normal-characters.

If there is no data (Normal-characters) to transmit then the link shall send NULL characters.

At least one Null character must be sent before any FCTs are sent.

The link receiver shall not recognise or record FCTs until it has received at least one NULL character (i.e. is the *Run* state).

7.7 ERROR DETECTION

There are two forms of error that can be detected and acted upon at the exchange level – disconnect errors and parity errors. Whenever a disconnect or parity error occurs both character level synchronisation and flow-control status cease to be valid. Both ends of the link must be reset and re-initialised to recover character synchronisation and flow control status.

7.7.1 Disconnect Error

An operational link sends normal-characters, FCTs or NULLs continuously, thus the data and/or strobe signals are always changing. The receiver shall detect a disconnection when the time interval from the last transition on either the data or strobe signal exceeds the disconnect-detection time. The disconnect-detection time shall be 850 nsec (nominal).

Before being able to detect a disconnect error the link must have received at least one bit.

When a disconnect error is detected the link shall move immediately to the *ErrorReset* state.

The disconnect error detection (and parity error detection) transitions have been added to the basic state diagram in Figure 7-4 below.

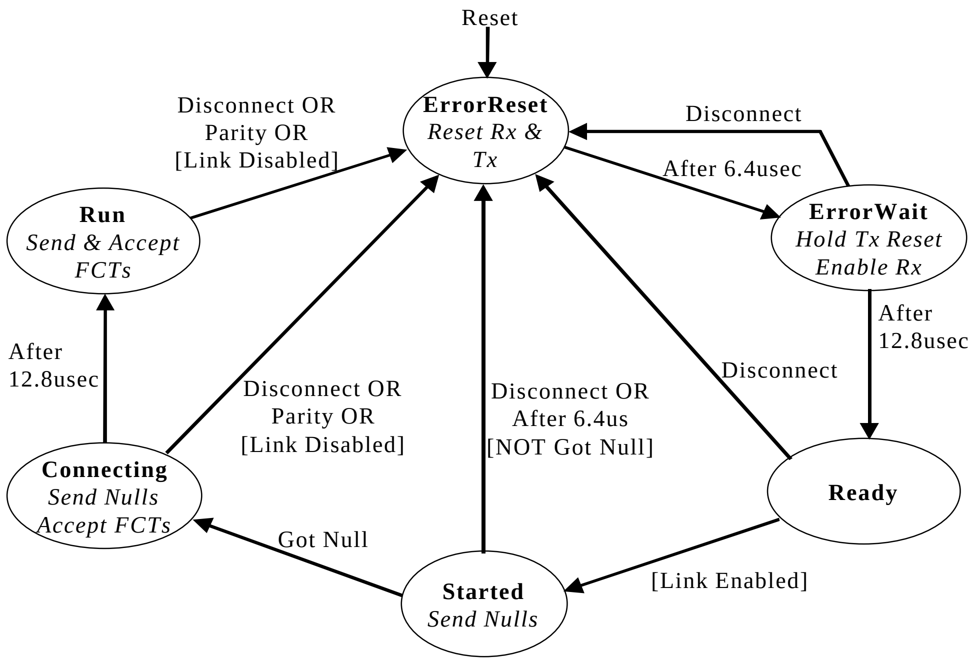

Figure 7-4 State Diagram for SpaceWire Link Interface with Error Transitions

When one end of the link (end A) is disabled and halts its output it causes a disconnect error at the other end of the link (end B). The end B will then cease transmission resulting in the disconnect error at end A. This procedure is known as an "exchange of silence". Both ends of the link will cycle through the reset sequence (*ErrorReset, ErrorWait, Ready*) ending up in the *Ready* state ready to begin operation once enabled. If end B is still enabled then it will move from the *Ready* state to the *Started* state and will send NULLs for 6.4 usec. Since end A is disabled it cannot respond. End A will, however, have started its disconnect timer and will also have registered that a NULL has been received. When end B completes the 6.4 usec timeout it will move to the *ErrorReset* state and disconnect (stop its output). End A is able to detect the disconnection so will also move to the *ErrorReset* state. Both ends will once again move through the reset sequence. This series of events will continue until either end A is enabled or end B is disabled.

A disconnect error can either be caused when one end of the link is disabled or when the link is physically disconnected (intentionally or unintentionally). If a physical disconnection is the cause of the disconnect error the both ends of the link will try repeatedly to make a connection until the link is reconnected or until the link interfaces are disabled.

7.7.2 Parity Error

Each data and control character has a parity bit associated with it (see section 6.3). When the parity bit for a data or control character is received, the parity of the character shall be checked. If a parity error occurs in the *Connecting* or *Run* states the link shall follow the disconnect error procedure described above in section 7.7.1.

7.8 EXCEPTION CONDITIONS

Several exception conditions have been identified where things, for one reason or another, do not follow the usual sequence of events. These exceptions are considered in this section.

7.8.1 Disconnect error while waiting to start

"Waiting to start" means that a link interface is in either the *ErrorReset*, *ErrorWait*, *Ready* or possibly the *Started* state. For a disconnect error to be detected while waiting to start the other end of the link (end B say) must have sent several NULLs (i.e. been in the *Started* state) so that the disconnect detect mechanism at end A could be activated. End B must have then given up waiting for end A to send a NULL and moved to the *ErrorReset* state and stopped its transmitter – thus causing the disconnect. An alternative possibility is that the link became physically disconnected. The following tables illustrates the various sequences of events starting from when end B has just moved to the *ErrorReset* state.

Table 7-2 End A in ErrorReset state

Link Interface End A	Link Interface End B	Event/Condition Causing Transition
ErrorReset	*Started*	End B times out while waiting to receive a NULL from end A. End B moves to the *ErrorReset* state.
ErrorReset	*ErrorReset*	End A and end B are both in the *ErrorReset* state they will step through the reset sequence and will start again when both ends are enabled.

Table 7-3 End A in ErrorWait state

Link Interface End A	Link Interface End B	Event/Condition Causing Transition
ErrorWait	*Started*	End B times out while waiting to receive a NULL from end A. End B moves to the *ErrorReset* state.
ErrorWait	*ErrorReset*	End B stops transmission. This is detected at end A as a disconnect error. End A moves to the *ErrorReset* state.
ErrorReset	*ErrorReset*	Both ends step through the reset sequence and will start again when both ends are enabled.

Table 7-4 End A in Ready state

Link Interface End A	Link Interface End B	Event/Condition Causing Transition
Ready	*Started*	End B times out while waiting to receive a NULL from end A. End B moves to the *ErrorReset* state.
Ready	*ErrorReset*	End B stops transmission. This is detected at end A as a disconnect error. End A moves to the *ErrorReset* state.
ErrorReset	*ErrorReset*	Both ends step through the reset sequence and will start again when both ends are enabled.

Table 7-5 End A in Started state

Link Interface End A	Link Interface End B	Event/Condition Causing Transition
Started	*Started*	End B times out while waiting to receive a NULL from end A. End B moves to the *ErrorReset* state.
Started	*ErrorReset*	End B stops transmission. This is detected at end A as a disconnect error. End A moves to the *ErrorReset* state.
ErrorReset	*ErrorReset*	Both ends step through the reset sequence and will start again when both ends are enabled.

If a physical disconnection has occurred then both ends of the link will continue to try to make a connection, cycling around the reset sequence, until they are disabled or until the connection is re-established.

7.8.2 Link connected in one direction but not in the other

A link may be connected in one direction and not in the other while a link is in the process of being plugged in (contact bounce time may be significantly larger than tens of usec) or if there is a break in the link cable.

In this case the sequence of events listed in the table below will be followed. Consider for convenience that both links are in the *started* state and that end A is connected to end B, but end B is not connected to end A.

Table 7-6 Link connected in one direction (A to B) but not in other

Link Interface End A	Link Interface End B	Event/Condition Causing Transition
Started	*Started*	End A is sending NULLs to end B and these are received, starting the disconnect timer of end B and registering as Got Null at end B. End B will therefore move to the *Connecting* state. End B is also sending NULLs to end A but these are not received at end A because the link is not connected in this direction.
Started	*Connecting*	End A is in the *Started* state waiting for NULLs to arrive. After waiting for 6.4 usec end A times out and moves to the ErrorReset state. End B continues to send NULLs and is able to accept FCTs. After waiting for 12.8 usec it would move into the *Run* state but before this can happen the 6.4 usec timeout for end A expires.
ErrorReset	*Connecting*	End A ceases transmission and this is detected at end B as a disconnect. End B moves to the *ErrorReset* state.
ErrorReset	*ErrorReset*	Both ends are in the *ErrorReset* state and will now cycle through the reset sequence (*ErrorReset*, *ErrorWait*, *Ready*, etc) until the link is properly connected or until one or both link interfaces are disabled.

Note that the 6.4 usec timeout in the *Started* state and the 12.8 usec timeout in the *Connecting* state are important. The timeout in the Connecting state must be longer than that in the *Started* state. This ensures that the end of the link that is not receiving data (and is in the *Started* state) has time to time out before the end that is receiving data (and is in the *Connecting* state) times out and moves to the *Run* state. If this precaution is not taken it is possible for FCTs to be lost if the partial link connection becomes a full connection while one end is in the *Run* state and the other end is in the *Started* state.

7.8.3 Parity error while waiting to start

Parity errors shall be ignored until the link interface is in the *Connecting* or *Run* state.

NULL detection in the *ErrorWait*, *Ready* and *Started* states shall include the parity bits within the NULL character i.e. the NULL character will only be detected if its parity is correct. Hence the NULL character will be detected when the 111010011 sequence of bits is received as illustrated in Figure 7-5.

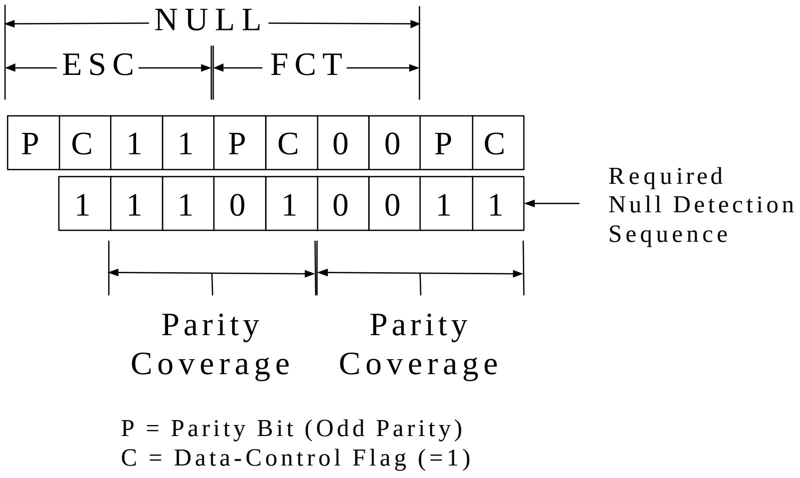

Figure 7-5 NULL Detection Sequence

Note that the sequence 111010010 (i.e. last data-control flag is 0 indicating next character is a data character) is also normally valid for a NULL. This should not register as a "Got Null" event in the *ErrorWait*, *Ready* or *Started* state since a data character following the NULL character is not valid during link initialisation.

7.8.4 One end starts as other end disconnects

One end (end A) arrives at the Start state 6.4 usec before the other end (end B) arrives at the Start state. The sequence of event is illustrated the following table.

Table 7-7 One end starts as other end disconnects

Link Interface End A	**Link Interface End B**	**Event/Condition Causing Transition**
Started	*Ready*	The time out timer at end A expires (after 6.4 usec) so end A moves to the *ErrorReset* state. End B is either just enabled or was previously enabled and has just entered the *Ready* state. In either case it will now move to the *Started* state.
ErrorReset	*Started*	End B has already received a NULL from end A so moves straight on into the *Connecting* state.
ErrorReset	*Connecting*	End A stops transmitting and causes end B to detect a disconnect. End B then moves on into the *ErrorReset* state.
ErrorReset	*ErrorReset*	Both ends step through the reset sequence and will start properly next time round.

7.8.5 D connected, S disconnected

If D is connected and S disconnected then the clock generated in the receiver will follow the data signal i.e. there will be a clock edge every time the data signal changes. This results in a continuous sequence of 0101010101.

This sequence will quickly produce a parity error because the parity is even for both control characters (4-bits) and data characters (10-bits) whereas the required parity is odd (see section 6.3).

During initialisation this sequence will be ignored as it does not produce a NULL character so initialisation will fail until the strobe signal is properly connected.

7.8.6 S connected, D disconnected

If S is connected and D disconnected then the clock generated in the receiver will follow the data signal i.e. there will be a clock edge every time the data signal changes. This results in a continuous sequence of 1111111111 since the data signal input will go to 1 when the data line is disconnected. If the data line is shorted to ground then the continuous sequence 0000000000 will be received.

This sequence will quickly produce a parity error because the parity is even for both control characters (4-bits) and data characters (10-bits) whereas the required parity is odd (see section 6.3).

During initialisation either sequence will be ignored as it does not produce a NULL character so initialisation will fail until the data signal is properly connected.

7.8.7 One side of differential pair disconnected

The effect of disconnecting one side of the differential pair will depend upon the values of the internal bias resistors at the interface, on the length of cable and on the grounding arrangements. There are three possibilities.

1. The data and strobe signals are still received correctly but with a much reduced noise margin. The link will continue to operate with a significant increase in the number of detected parity errors.
2. The stobe signal sits at logic 0 or logic 1. This is similar in effect to the D connected, S disconnected case of section 7.8.5 above.
3. The data signal sits at logic 0 or logic 1. This is similar in effect to the S connected, D disconnected case of section 7.8.6 above.

Further testing needs to be done to confirm the above.

7.9 AUTOSTART

A link interface should be able to be commanded to start automatically on receipt of a NULL. In this case the Link Enabled condition in the state machine is set as follows.

[Link Enabled] = (NOT [Link Disabled]) AND ([LinkStart] OR ([AutoStart] AND GotNull))

LinkDisabled is a flag set by software or hardware to indicate that the link is disabled. This corresponds to the Link Disabled condition in the state diagram.

LinkStart is a flag set by software or hardware to start a link i.e. to cause the transition from the *Ready* state to the *Started* state.

AutoStart is a flag set by software or hardware to indicate that the link should start automatically on receipt of a NULL character.

GotNull is a flag indicating that the link interface has received a NULL character.

LinkStart and AutoStart only operated when the link interface is not disabled i.e. [LinkDisabled] = False.

The AutoStart facility enables a system to be set up where one end (end A) of the link is held waiting for the other end (end B) to attempt connection. As soon as end B tries to connect end A will respond immediately. This allows connection of a link to b controlled from one end of the link only.

7.10 CREDIT ERROR

In the *Run* state if the receiver attempts to try to write to the receive FIFO when it is full then the link interface shall move to the *ErrorReset* state. This situation can arise if an error occurs undetected by the parity bit. It is then possible that a credit error can occur with the resultant effect that the receiver will try to write to the receive FIFO when it is full. This condition must be flagged as an error and cause a disconnect.

7.11 FASTER CONNECTION

In the basic state diagram both ends of a link have to wait for 12.8 usec before moving from the *Connecting* state to the *Run* state. This is unnecessary. If an FCT is received by a link interface in the *Connecting* state then it should move immediately to the *Run* state. The following table provides and illustration of this situation. Both ends have reached the *Connecting* state.

Table 7-8 Faster connection

Link Interface End A	Link Interface End B	Event/Condition Causing Transition
Connecting	*Connecting*	End B times out after 12.8 usec and moves to the *Run* state.
Connecting	*Run*	End B sends one or more FCTs. End A receives the first FCT and moves to the *Run* state.
Run	*Run*	Both ends are in the *Run* state and begin normal operation.

7.12 ALTERNATIVE STATE MACHINE

An alternative state machine to that proposed in section 7.4 has been discussed within the SpaceWire Core Working Group. The alternative state machine is based on the handshake principle rather than the wait for response approach of the proposed state machine. The alternative state machine is presented in Figure 7-6.

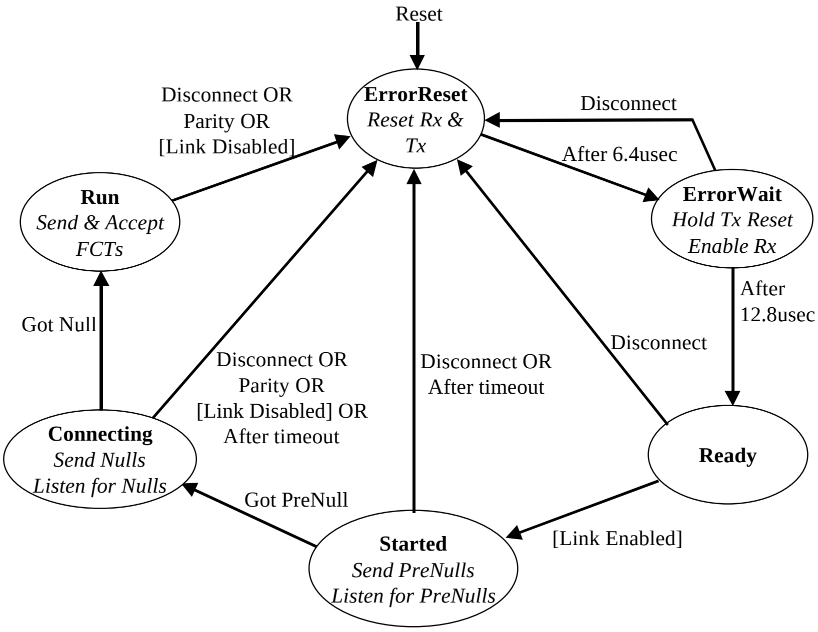

Figure 7-6 Alternative State Machine Discussed by SpaceWire Core Working Group

This approach has the advantage that is does not have the "After 12.8 usec" wait in the connection path needed in the state diagram of section 7.4.

It has the disadvantage that it requires transmission and detection of PreNull characters (which could possibly be any data or control token other than the NULL character – it could even be a NULL character with an incorrect parity bit).

It has a further disadvantage in that it may well not be compatible with existing IEEE-1355 DS encoder-decoder devices (TBC).

8. PACKET LEVEL

The packet level protocol follows the principles of the DS-SE and DS-DE character level encoding given in section 6, "Common Packet Level", of the IEEE Standard 1355-1995 [RD1]. Some ambiguities in the IEEE1355 specification have been resolved in the SpaceWire standard. See Annex C for details of the differences between SpaceWire and IEEE 1355 and the reasons for those differences.

8.1 PACKETS

A packet shall comprise a destination address, a payload and an end_of_packet marker.

<destination address> <payload> <EOP>

8.1.1 Destination

The destination address shall consist of a list of zero or more destination identifiers (dest_id).

<destination address> = <dest_id1> <dest_id2> … <dest_idN>

A destination identifier comprises one or more bytes.

The destination list is not delimited.

The case of zero destination identifiers in the destination list (i.e. the destination list is empty) is intended to support a network which is simply a single point-to-point link from source to destination.

The case of one or more destination identifiers in the destination list is intended to support routing of a packet across a network.

8.1.2 Payload

The payload comprises the data characters that are to be transferred from the source to the destination.

The payload shall contain one or more characters.

8.1.3 End_of_Packet Markers

There are two possible end of packet markers: EOP and EOM.

The EOP and EOM control character formats are described in section 6.2 of this standard.

EOP shall be used as the normal end of packet marker indicating the end of a packet.

EOM shall be used to indicate the end of the last packet of a message.

The first data character following an end of packet marker shall be taken as the first character of the next packet.

9. NETWORK LEVEL

The network level has yet to be defined.

9.1.1 Networks and Routing

A network comprises zero or more routing switches interconnected by SpaceWire links. Nodes interface to the network via a SpaceWire link. Nodes may be packet sources, packet destinations or both.

A routing switch shall connect several SpaceWire links together and be able to route packets from one SpaceWire link to another. The leading (first) destination identifier of a packet shall define the output link of the routing switch that the packet is to be routed to. A routing switch shall strip zero or one leading destination identifiers from the front of the destination list.

EXAMPLE:

<dest_id1> <dest_id2> <payload> <EOP>

After passing through a routing switch that deletes the first destination identifier the second one is at the front of the packet ready for interpretation by the next routing switch encountered.

<dest_id2> <payload> <EOP>

The node producing the packet must know the address of the destination which may require knowledge of the network topology.

10. ERROR RECOVERY SCHEME

The error recovery scheme has yet to be defined.

11. CONFORMANCE CRITERIA

11.1 CONFORMANCE STATEMENTS

Several SpaceWire compatible products may be identified each of which implement only a part of the SpaceWire standard:-

- SpaceWire Cable
- SpaceWire Connector
- SpaceWire Cable Assembly
- SpaceWire Interface

Corresponding subsets of the SpaceWire standard are defined to which implementations may claim conformance. The conformance statement used should take the form given in the appropriate subset definition.

11.2 DEFINITION OF SUBSETS

11.2.1 SpaceWire Cable

An implementation of SpaceWire cable shall conform to all of the normative specifications given in all of the sections listed in Table 11-1. A cable meeting this specification may use the following conformance statement:-

This cable conforms to the SpaceWire cable specification of the ESA SpaceWire Standard.

Table 11-1 SpaceWire Cable Conformance

Relevant Sections	Title
4.1	Cables

11.2.2 SpaceWire Connector

An implementation of a SpaceWire connector shall conform to all of the normative specifications given in all of the sections listed in table 11-1. A connector meeting this specification may use the following conformance statement:-

This connector conforms to the SpaceWire connector specification of the ESA SpaceWire Standard.

Table 11-2 SpaceWire Connector Conformance

Relevant Sections	Title
4.2	Connectors

11.2.3 SpaceWire Cable Assembly

An implementation of a SpaceWire cable assembly shall conform to all of the normative specifications given in all of the sections listed in Table 11-3. A cable assembly meeting this specification may use the following conformance statement:-

This cable assembly conforms to the SpaceWire cable assembly specification of the ESA SpaceWire Standard.

Table 11-3 SpaceWire Cable Assembly Conformance

Relevant Sections	**Title**
4.3	Cable Assembly

11.2.4 SpaceWire Interface

An implementation of a SpaceWire interface shall conform to all of the normative specifications given in all of the sections listed in Table 11-4. A product fitted with an interface meeting this specification may use the following conformance statement:-

This product conforms to the SpaceWire interface specification of the ESA SpaceWire Standard.

Table 11-4 SpaceWire Interface Conformance

Relevant Sections	**Title**
4.2	Connectors
4.4	PCB Tracks
5	Signal Level
6	Character Level
7	Exchange Level
8	Packet Level

Together with the above conformance statement the following parameters shall be specified for the interface.

1. Total transmitter data-strobe skew (worst case[3]) measured at the interface connector (Dout to Sout skew).
2. Total transmitter data jitter (worst case) measured at the interface connector (Dout jitter).
3. Total transmitter strobe skew (worst case) measured at the interface connector (Sout jitter).
4. Total receiver minimum separation between data and strobe (worst case) measured at the interface connector (Din to Sin minimum separation[4]).

Typical figures for the above parameters may also be provided, in which case the conditions applying for the typical figure measurements must be clearly stated (e.g. temperature, operating voltage).

A detailed explanation of the above parameters is provided in section 5.

11.2.5 SpaceWire Encoder/Decoder

An implementation of a SpaceWire encoder/decoder shall conform to all of the normative specifications given in all of the sections listed in Table 11-5. A product fitted with an interface meeting this specification may use the following conformance statement:-

This product conforms to the SpaceWire encoder/decoder specification of the ESA SpaceWire Standard.

---

[3] Worst case over process, temperature, voltage range and irradiation.

[4] This must include all D-S skew, D jitter and S jitter between the interface connector and the decoder device.

Table 11-5 SpaceWire Encoder/Decoder Conformance

Relevant Sections	Title
5	Signal Level
6	Character Level
7	Exchange Level
8	Packet Level

Together with the above conformance statement the following parameters shall be specified for the interface.

1. Encoder data-strobe skew (worst case[5]) measured at the output of the encoder device.
2. Encoder data jitter (worst case) measured at the output of the encoder device.
3. Encoder strobe skew (worst case) measured at the output of the encoder device.
4. Decoder minimum separation between data and strobe (worst case) measured at the input of the decoder device.

Typical figures for the above parameters may also be provided, in which case the conditions applying for the typical figure measurements must be clearly stated (e.g. temperature, operating voltage).

A detailed explanation of the above parameters is provided in section 5.

[5] Worst case over process, temperature, voltage range and irradiation.

Annex A (Normative) SpaceWire Requirements

The initial requirements for the physical and signalling layers of the SpaceWire data link as defined during the DICE study [RD11] are given in this section.

A.1 DATA RATE AND CABLE LENGTH

A.1.1 Maximum Signalling Rate

A maximum signalling rate of at least 100 Mbits per second (100 Mbaud) shall be provided over a single differential pair.

Rationale: Many instruments already have data rates approaching 100Mbits per second or higher. Future instruments are likely to require data rates of up to 1Gbit per second. To be useful in future as a high-speed data link at least 100Mbaud should be supported. Higher data rates may be implemented by using several data links in parallel. The overhead of a serial link protocol will reduce the data rate supported by a single link compared to its baud rate.

Need: Essential

A.1.2 Minimum Signalling Rate

A minimum signalling rate of less that one tenth of the maximum signalling rate shall be supported.

Rationale: The minimum signalling rate will depend upon the signalling method used (e.g. Data/Strobe) and the specific interface components. Depending upon the particular drivers/receivers used, reducing the signalling rate when no data is being sent may reduce system power consumption.

Need: Desirable

A.1.3 Maximum Cable Length

The maximum signalling rate (R1.1) shall apply over a maximum cable length of at least 10m.

Rationale: A maximum distance of 10m should be adequate for most spacecraft applications, given the size of typical spacecraft.

Need: Essential

A.2 EMC

These requirements have been derived from the Rosetta and Envisat EMC requirements specifications.

A.2.1 Conducted Emission

The data link shall not emit common mode conductive interference appearing on adjacent signal lines.

TBC

A.2.2 Conducted Susceptibility

The data link receiver circuits shall have immunity to common mode interference levels of

± 360mV (1kHz square wave injected into signal lines).

Rationale: This requirement is taken from the Rosetta EMC Requirements Specification (R5.1.2.3-1).

Need: Essential

A.2.3 Radiated Emission, Electric Field

A single data link shall not radiate electric fields, measured at 1m distance, in excess of the levels given in Table A-1 below:-

Table A-1: Radiated Electric Field Emission Requirements

Reference	Frequency Range	Field Strength	Bandwidth
A.2.3-1	10kHz - 1GHz	40 dBuV/m (rms)	Narrow band
A.2.3-2	1GHz - 10GHz	60 dBuV/m (rms)	Narrow band
A.2.3-3	10kHz - 1GHz	70 dBuV/m (rms)	Broad band

Rationale: These requirements are in line with the radiated emission requirement for Rosetta (Rosetta EMC Requirements Specification R5.2.1.1-1, which gives a minimum requirement of 50dBuV/m) and the sub-system requirements for ENVISAT (ENVISAT EMC Requirements Specification section 4.4.1, which give a requirement of 44dBuV/m narrowband and 74dBuV/m broadband, 10kHz - 1GHz).

Need: Essential

A.2.4 Radiated Susceptibility, Electric Field

The data link shall not exhibit any malfunction, degradation of performance or deviation from specified parameters when irradiated with an Electric Field of strength up to the levels given in Table A-2 below:-

Table A-2: Radiated Electric Field Susceptibility Requirements

Reference	Frequency Band	Strength	Modulation
A.2.4-1	14kHz -1GHz	1V/m (rms)	30-80%AM, 1kHz squarewave
A.2.4-2	1GHz - 40GHz	20V/m (rms)	Pulsed 1kHz PRF, 10-30% duty cycle

Rationale: These requirements are in line with those of Rosetta (Rosetta EMC Requirements Specification R5.2.2.1-1) and ENVISAT (ENVISAT EMC Requirements Specification section 4.5.1).

Need: Essential

A.2.5 Radiated Emission, Magnetic Field

The single data link shall not radiate magnetic fields, measured at 1m distance, in excess of the limits given in Table A-3 below:-

Table A-3: Radiated Magnetic Field Emission Requirements

Reference	Frequency Range	Field Strength	Bandwidth
A.2.5-1	AC (30Hz - 50kHz)	≤ 50dBpT	Narrow band
A.2.5-2	DC (Static)	≤ 200nT	Static

Rationale: These requirements are in line with the magnetic radiated emission requirements for Rosetta (Rosetta EMC Requirements Specification R5.2.1.2-1) and ENVISAT (ENVISAT EMC Requirements Specification section 4.4.2).

Need: Essential

A.2.6 Radiated Susceptibility, Magnetic Field

The data link shall not exhibit any malfunction, degradation of performance or deviation from specified parameters when irradiated with a Magnetic Field of strength up to the levels given in Table A-4 below:-

Table A-4: Radiated Magnetic Field Susceptibility Requirements

Reference	Frequency Band	Strength
A.2.6-1	AC (30Hz-500Hz)	< 136dBpT
A.2.6-2	AC (500-50kHz)	< 120dBpT
A.2.6-3	DC (static)	155dBpT (Earth Field)

Rationale: These requirements are in line with those of Rosetta (Rosetta EMC Requirements Specification R5.2.2.2-1) and ENVISAT (ENVISAT EMC Requirements Specification section 4.5.2). The DC magnetic field susceptibility requirement (R2.5-2) represents operation in the Earth's magnetic field and need not be tested explicitly.

Need: Essential

A.3 ESD

A.3.1 ESD Rating

Driver and receiver devices shall have ESD protection for greater than 3000 Volts.

Rationale: This is in-line with standard CMOS components. Normal antistatic handling procedures will be required to prevent any ESD damage to driver and receiver devices.

Need: Essential

A.4 Error Rate

A.4.1 Bit Error Rate (BER)

A bit error rate of less than 10^{-12} shall be achieved over a distance of 10m using the recommended cable type, connectors, and PCB trace layout.

Rationale: At a signalling rate of 100 Mbaud this corresponds to one error every 10^4 seconds or 2.78 hours. A higher level error detection and correction protocol must be provided to reduce the error rate below this level. This is provided by IEEE-1355.

Need: Essential

A.5 Radiation

A.5.1 Total Dose

A radiation total dose of greater than 100krad (Si) shall be tolerated.

Rationale: This would make the LVDS devices suitable for most space applications.

Need: Desirable – this is a target specification, lower radiation performance values may be acceptable (TBD).

A.5.2 Single Event Upset

The single event upset threshold (SEU LET) shall be greater 30MeV/mg/cm^2.

Rationale: This would make the LVDS devices suitable for most space applications.

Need: Desirable – this is a target specification, lower radiation performance values may be acceptable (TBD).

A.5.3 Latch-Up

The single event latch-up threshold (SEL LET) shall be greater 100MeV/mg/cm^2.

Rationale: This would make the LVDS devices suitable for most space applications.

Need: Desirable – this is a target specification, lower radiation performance values may be acceptable (TBD).

A.6 POWER CONSUMPTION

A.6.1 Link Power Consumption

A driver/receiver pair together with necessary termination and biasing components, implementing a single uni-direction link, shall have a typical power consumption of less than 0.1 Watt.

Rationale: The aim as far as power consumption is concerned is to significantly improve over the high power consumption of ECL, PECL and other high speed interface devices.

Need: Essential

A.7 FAIL SAFE / FAULT ISOLATION

A.7.1 Receiver Inputs Open

When the receiver inputs are open circuit the receiver output shall remain in a known logic state (either HIGH or LOW). The receiver output must not oscillate.

Rationale: With some logic families (e.g. CMOS) an open circuit input will float to somewhere around the logic threshold level. Any noise on the input will then cause the receiver output to oscillate. This results in high power consumption in the receiver and unpredictable behaviour of the system to which it is connected.

Need: Essential

A.7.2 Receiver Inputs Shorted

When the receiver inputs are short circuited the receiver output shall remain in a known logic state (either HIGH or LOW) and the receiver shall not be damaged.

Rationale: A short circuit must not cause damage to the receiver or unpredictable behaviour of the system to which it is connected.

Need: Essential

A.7.3 Driver Outputs Open

When the driver outputs are open circuit no damage shall result to the driver.

Rationale: It is possible that a driver may be powered and enabled before a receiver and termination resistor is connected to it. In this case the driver should not be damaged.

Need: Essential

A.7.4 Driver Outputs Shorted

When the driver outputs are shorted together or to ground no damage shall result to the driver.

Rationale: If a fault occurs which shorts the outputs of the driver together or to ground, then the driver should not be damaged or propagate the fault.

Need: Essential

A.7.5 Driver Powered, Receiver Not Powered

When the driver is powered and the receiver is not powered neither the driver nor the receiver shall be damaged.

Rationale: In a cold redundant system active signals may be connected to a unit which is powered down. In this case no damage should result to either the driver or receiver circuits. NOTE: normally the driver would be tri-stated to prevent any unnecessary power consumption.

Need: Essential

A.7.6 Driver Not Powered, Receiver Powered

When the driver is not powered and the receiver is powered neither the driver nor the receiver shall be damaged and the receiver output shall remain in a known logic state (either HIGH or LOW).

Rationale: In a cold redundant system a receiver may be powered and the driver not powered. In this case it is important that the receiver remains in a known logic state to prevent unpredictable behaviour of the system to which the receiver is connected.

Need: Essential

A.7.7 Driver Tri-State

It shall be possible to tri-state the driver.

Rationale: This is required to prevent unnecessary power consumption in cold redundant systems.

Need: Essential

A.8 MASS AND SIZE

A.8.1 Driver / Receiver Packaging

A small outline device package containing several drivers and/or receivers shall be used.

Rationale: Small outline packages with several drivers and/or receivers per package will save on size and mass.

Need: Desirable

A.8.2 Driver / Receiver Integration

Driver and receiver circuits that can be integrated with higher-level system functions are preferred.

Rationale: Integrating the drivers and receivers with the higher-level functions will eliminate the separate driver/receiver components saving mass and board area.

Need: Desirable

A.8.3 Line Termination

The line termination shall normally require no more than three components.

Rationale: A single resistor should suffice for line termination unless centre tap capacitance termination is used to filter common mode noise. This requires a pair of resistors and a capacitor.

Need: Desirable

A.8.4 Driver/Receiver Power

The driver and receiver circuits shall run off the same power supplies as the system to which they are connected, unless galvanic isolation is required.

Rationale: Extra power supplies will add to system mass, size and complexity.

Need: Desirable

A.9 MAGNETIC EMISSION

A.9.1 Non-Ferrous Materials

Ferrous materials shall NOT be used in any component forming the data link.

Rationale: Magnetic emission is an important consideration for some missions. To support these missions no magnetic materials should be used in the data link components.

Need: Desirable

A.10 GALVANIC ISOLATION

A.10.1 Galvanic Isolation

It should be possible to galvanically isolate two systems connected via the data link.

Rationale: Some space instruments or sub-systems require to be galvanically isolated. This is not generally the case but will depend upon specific mission requirements. The data link should be capable of supporting galvanic isolation through the addition of extra isolation circuitry.

Need: Desirable

A.11 REQUIREMENTS SUMMARY

A table summarising the requirements for space-link is given below:

Table A-5 Requirements summary

Ref.	Title	Requirement Summary
A.1.1	Maximum signalling Rate	> 100 Mbaud
A.1.2	Minimum Signalling Rate	< 0.1 x Maximum Signalling Rate
A.1.3	Maximum Cable Length	> 10m at 100 Mbaud
A.2.1	Conducted Emission	TBC
A.2.2	Conducted Susceptibility	Immunity to ±360mV 1kHz. sq. wave
A.2.3-1	Radiated Emission, Electric Field	< 40dBuV/m rms (10kHz - 1GHz, narrow band)
A.2.3-2	Radiated Emission, Electric Field	< 60dBuV/m rms (1GHz - 10GHz, narrow band)
A.2.3-3	Radiated Emission, Electric Field	< 70dBuV/m rms (10kHz - 1GHz, broad band)
A.2.4-1	Radiated Susceptibility, Electric Field	immunity to 1 V/m rms (14kHz - 1 GHz, 1kHz sq.wave)
A.2.4-2	Radiated Susceptibility, Electric Field	immunity to 20V/m rms (1GHz - 40GHz, Pulsed 1kHz PRF)
A.2.5-1	Radiated Emission, Magnetic Field	< 50dBpT (30Hz - 50kHz, narrow band)
A.2.5-2	Radiated Emission, Magnetic Field	< 200nT (DC)
A.2.6-1	Radiated Susceptibility, Magnetic Field	< 136dBpT (30Hz-500Hz)
A.2.6-2	Radiated Susceptibility, Magnetic Field	< 120dBpT (500Hz-50kHz)
A.2.6-3	Radiated Susceptibility, Magnetic Field	< 155dBpT (DC, Earth Field)
A.3.1	ESD Rating	> 3000 V
A.4.1	Bit Error Rate(BER)	< 10^{-12} over 10m
A.5.1	Total Dose	> 100krad (Si)
A.5.2	Single Event Upset Threshold	> 30 MeV/mg/cm^2
A.5.3	Latch-Up Threshold	> 100 MeV/mg/cm^2
A.6.1	Link Power Consumption	< 0.1 W (uni-directional Link)
A.7.1	Receiver Inputs Open	Known output state, no oscillation
A.7.2	Receiver Inputs shorted	Known output state, no damage to receiver
A.7.3	Driver Outputs Open	No damage to driver
A.7.4	Driver Outputs Shorted	No damage to driver
A.7.5	Driver Powered, Receiver Not Powered	No damage to driver or receiver
A.7.6	Driver Not Powered, Receiver Powered	No damage to driver or receiver and known output state
A.7.7	Driver Tri-State	Possible to tri-state driver
A.8.1	Driver / Receiver Packaging	Small outline package with multiple drivers and/or receivers
A.8.2	Driver / Receiver Integration	Possible to integrate with encoder etc.
A.8.3	Line Termination	No more than three components
A.8.4	Driver/Receiver Supply Voltage	Same Vcc as system
A.9.1	Non-Ferrous Materials	No ferrous material to be used
A.10.1	Galvanic Isolation	Possible to galvanically isolate two systems connected by link

Annex B (Normative) SpaceWire EMC Performance

This section provides an evaluation of SpaceWire (in particular LVDS) against the data link requirements set out in the Annex A report. This testing was performed as part of the DICE study. Full details of the EMC test results are reported in [RD15].

The following table (Table B-1) lists the requirements and considers whether LVDS meets each requirement. A double tick ( ) in the "OK" column of the table means that tests have shown conclusively that LVDS meets a requirement. A single tick () means that the performance is thought to be satisfactory although the requirement has not been met in full. TBC in the table means that the tests were inconclusive or that the requirement has not been tested during the current DICE study.

The bit error rate (BER) test has checked both LVDS and IEEE-1355 i.e. the complete SpaceWire link. The EMC testing has tested the LVDS part of the data link using where appropriate IEEE-1355 like data patterns.

Table B-1 Does LVDS Meet The Requirements

Ref.	Title	Requirement Summary	Test Result	OK?
A.1.1	Maximum signalling Rate	> 100Mbaud	LVDS eye diagram results indicated a maximum performance of 200Mbaud	√√
A.1.2	Minimum Signalling Rate	< 0.1 x Maximum Signalling Rate	Not Tested	√√
A.1.3	Maximum Cable Length	> 10m at 100 Mbaud	10m requirement achieved at 100Mbaud	√√
A.2.1	Conducted Emission	TBD – depends on particular power supply arrangement used.	Cross-coupling from signal to power supply measured at around 4mA i.e. very low.	√
A.2.2	Conducted Susceptibility	Immunity to ±360mV 1kHz. sq. wave	AC and DC susceptibility tested	√√
A.2.3-1	Radiated Emission, Electric Field	< 40dBuV/m rms (10kHz - 1GHz, narrow band)	Problem with interference from pattern generator.	√ TBC
A.2.3-2	Radiated Emission, Electric Field	< 60dBuV/m rms (1GHz - 10Ghz, narrow band)	Requirement met.	√√
A.2.3-3	Radiated Emission, Electric Field	< 70dBuV/m rms (10kHz - 1GHz, broad band)	Not tested.	TBC
A.2.4-1	Radiated Susceptibility, Electric Field	immunity to 1 V/m rms (14kHz - 1 GHz, 1kHz sq.wave)	Requirement met	√√
A.2.4-2	Radiated Susceptibility, Electric Field	immunity to 20V/m rms (1GHz - 40GHz, Pulsed 1kHz PRF)	Tested in range 1 GHz to 20GHz with field strength of at least 4.5V/m	√
A.2.5-1	Radiated Emission, Magnetic Field	< 50dBpT (30Hz - 50kHz, narrow band)	Requirement met across frequency band.	√√
A.2.5-2	Radiated Emission, Magnetic Field	< 200nT (DC)	Not tested – DC magnetic emission is caused by the use of magnetic materials see R9.1	√
A.2.6-1	Radiated Susceptibility, Magnetic Field	< 136dBpT (30Hz-500Hz)	Requirement met.	√√
A.2.6-2	Radiated Susceptibility, Magnetic Field	< 120dBpT (500Hz-50kHz)	Requirement met.	√√
A.2.6-3	Radiated Susceptibility, Magnetic Field	< 155dBpT (DC, Earth Field)	Indirectly Tested – All tests performed in Earth's magnetic field.	√√
A.3.1	ESD Rating	> 3000 V	Specified by LVDS device manufacturer.	√
A.4.1	Bit Error Rate (BER)	< 10^{-12} over 10m	Achieved better than 10^{-13} over 21m under normal lab conditions.	√ TBC
A.5.1	Total Dose	> 100krad (Si)	50krad for receiver. Driver not tested.	TBC
A.5.2	Single Event Upset	> 30 MeV/mg/cm^2	20 MeV/mg/cm^2 for	TBC

			receiver.	
A.5.3	Latch-UP	> 100 MeV/mg/cm^2	>77.3 MeV/mg/cm^2 for receiver. Driver showed latch-up at 27.5 MeV/mg/cm^2.	TBC
A.6.1	Link Power Consumption	< 0.1 W (uni-directional Link)	Measured as 38.5 mW at room temp., 5.5V supply operating at 100 Mbits/s	√√
A.7.1	Receiver Inputs Open	Known output state, no oscillation	Output high. Oscillation sometimes occurred – though to be due to oscilloscope probes.	√
A.7.2	Receiver Inputs shorted	Known output state, no damage to receiver	Output high. No damage.	√√
A.7.3	Driver Outputs Open	No damage to driver	No damage.	√√
A.7.4	Driver Outputs Shorted	No damage to driver	No damage.	√√
A.7.5	Driver Powered, Receiver Not Powered	No damage to driver or receiver	No damage.	√√
A.7.6	Driver Not Powered, Receiver Powered	No damage to driver or receiver and known output state	Output high. No damage.	√√
A.7.7	Driver Tri-State	Possible to tri-state driver	Facility provided on LVDS driver device.	√√
A.8.1	Driver / Receiver Packaging	Small outline package with multiple drivers and/or receivers	Not Tested – the LVDS devices used have four drivers or receivers in each small outline package	√√
A.8.2	Driver / Receiver Integration	Possible to integrate with encoder etc.	Not Tested – LVDS can be implemented in a range of different technologies.	√√
A.8.3	Line Termination	No more than three components	Indirectly Tested – single resistor termination used for all tests.	√√
A.8.4	Driver/Receiver Supply Voltage	Same Vcc as system	Indirectly Tested – LVDS devices used operate with a single supply rail at a nominal 5V.	√√
A.9.1	Non-Ferrous Materials	No ferrous material to be used	Not Tested – cables and connector must be specified to NOT have magnetic materials.	√√
A.10.1	Galvanic Isolation	Possible to galvanically isolate two systems connected by link	Not Tested – A circuit suitable for the galvanic isolation of LVDS has been identified (patent owed by TI).	√√

Annex C (Informative) Differences Between Spa ceWire and IEEE-1355

There are several differences between SpaceWire and IEEE 1355-1995 [RD1]. Improvements have been made in the SpaceWire standard to improve ruggedness, power consumption, EMC performance, and to eliminate problems and ambiguities that exist with IEEE 1355.

The differences between the two standard and the reasons for them are detailed below looking at each level of the SpaceWire standard in turn.

C.1 PHYSICAL LEVEL

- The SpaceWire cable is designed to be suitable for space applications. IEEE 1355 cable is not suitable.
- SpaceWire connectors are 9-way micro-miniature D-type connectors which are used in space applications. The IEEE 1355 connector is not rugged enough for space use.

C.2 SIGNAL LEVEL

- LVDS adopted for SpaceWire provides improved electromagnetic emission characteristics compared to the PECL signals used in IEEE 1355. LVDS supports failsafe operation, which is not the case with PECL. LVDS can be implemented in a range of semiconductor technologies. This enables integration of completed SpaceWire interfaces with other system functions.
- The DS encoding used by SpaceWire is identical to IEEE 1355 with the exception that SpaceWire interfaces must be tolerant of simultaneous transitions on data and strobe signals. This is not a requirement of IEEE 1355 and may lead to faults occurring within the interface.
- The SpaceWire timing specification is tightened up compared to that in IEEE 1355. IEEE 1355 only considers skew whereas SpaceWire considers both jitter and skew.

C.3 CHARACTER LEVEL

- The character level protocol for SpaceWire is identical to that in IEEE 1355.
- See also minor differences section 72.

C.4 EXCHANGE LEVEL

- IEEE 1355 section 5.7.2 states “Thereafter it shall send only NULL characters unless and until at least <u>one character</u> has been received by the corresponding link input since reset. After the link output has been started and at least <u>one character</u> has been received by the corresponding link input since reset, the link shall begin normal operation.” The state diagram in Figure 5-11 and timing sequence diagram in Figure 5-12 of IEEE 1355 show a different situation. Instead of one character being received it is specifically a NULL that must be received. This ambiguity is resolved in the SpaceWire standard – a NULL must be received.
- IEEE 1355 section 5.7.4.2 states “If a link interface detects a disconnect error before it has started, it shall start, transmit at least one character, and then halt, to ensure that a disconnection error is also detected by the other end …” The state diagram in Figure 5-11 of IEEE 1355 shows a different reaction to a disconnect error before the link has started. The link is simply halted, it is NOT started and a character is NOT sent. This ambiguity is resolved in the SpaceWire standard by modification of the state machine. If a disconnect error is detected before the link has started then the link resets immediately – it does not send a character first. This requires modification to the state machine to work reliably.
- IEEE 1355 section 5.7.2 states “After reset, a DS-SE link output shall maintain both signals at their reset level until started, i.e., instructed to begin operation (note that receipt of a character by the corresponding link input may be taken as such an instruction).” The state diagram in Figure 5-11 of IEEE 1355 requires the *LinkStart* command even when a character (specifically a NULL)

has been received. When a NULL is received in the *Ready* state the link moves to the *NullReceived* state. To move on to the *Run* state it must then receive the *LinkStart* command. The SpaceWire standard requires the *LinkEnable* (equivalent to *LinkStart)* command to be given before a link can move to the *Run* state.

- The state machine illustrated in Figure 5-11 of IEEE 1355 will hang up if the *ResetLinkCommand* is given to both ends of the link while they are both in the *Ready* state. In the *Ready* state no characters have been sent so the disconnect time-out has not started (disconnect time-out is started only after a bit has been received – IEEE 1355 section 5.7.2). When the *ResetLinkCommand* is given the state machine moves to the *WaitInStop* state where it waits for a disconnect error – that cannot occur! Further application of the *ResetLinkCommand* will not resolve this problem, the state machine remains firmly stuck in the *WaitInStop* state until a *PowerOnReset* is applied. The SpaceWire standard has a modified state machine in the exchange level that resolves this problem. The *WaitInStop* state has been removed completely and any Reset command causes a transition to the *ErrorReset* state.

- A similar problem to that described above can occur if one end of the link (end A) has started (in *Started* state) and the other end (end B) is in the *NullsReceived* state, but has not yet been commanded to start. If end A receives a *ResetLinkCommand* it will move to the *WaitInStop* state. In the *WaitInStop* state the outputs are halted so end A stops transmitting NULLs and this is detected as a disconnect error at end B. End B then moves through the *ErrorReset* and *ErrorWait* states ending up in the *Ready* state. Meanwhile end A has remained in the *WaitInStop* state unable to detect the disconnect error because it has not been sent a character. The modified state machine specified in the SpaceWire standard resolves this problem.

C.5 PACKET LEVEL

- The IEEE Std 1355-1995 specification section 9.2.1 is unclear in the case of a destination being null. It is not clear whether this means a destination address of zero, which is a valid destination address, or whether it means a non-existent destination, i.e. the destination list is empty (contains zero destination identifiers). The latter case is assumed in the SpaceWire standard.

- The IEEE Std 1355-1995 specification section 9.2.1 is also ambiguous in its definition of a dest_id. It is defined as a fixed size field, its size being known to the (sub)network. It is not clear whether a network comprising several sub-networks can have different size dest_ids. E.g. the first sub-network encountered by a packet may expect a dest_id of 2-bytes and the next sub network encountered may expect a dest_id of 1-byte. In the SpaceWire standard it is assumed that a destination list can contain dest_ids of different lengths. Each routing switch will know how many bytes to strip off. The packet source must know the destination address across the network. Alternative paths to the destination available to the source can have different format destination lists. Alternative paths to a destination determined by an intelligent routing device must have the same format destination lists.

C.6 NETWORK LEVEL

- There is no Network level in IEEE-1355.

C.7 ERROR RECOVERY SCHEME

- There is no Error Recovery Scheme defined in IEEE-1355.

C.8 OTHER MINOR DIFFERENCES

- The name flow control token (FCT) is used consistently throughout the SpaceWire standard in place of FCT and FCC in IEEE-1355.

- The two EOP tokens of IEEE-1355 (EOP-1 and EOP-2) have been renamed EOP (End of Packet) and EOM (End of Message). EOP of SpaceWire and EOP-1 of IEEE-1355 have the same function. The EOM of SpaceWire is used to indicate an end of message only and is not used for error recovery purposes as is implied for the EOP-2 in IEEE-1355.

Annex D (Informative) SpaceWire User Guide

Annex E (Informative) BIBLIOGRAPHY

E.1 REFERENCE DOCUMENTS[6]

RD1 IEEE Computer Society, "IEEE Standard for Heterogeneous Interconnect (HIC) (Low-Cost, Low-Latency Scalable Serial Interconnect for Parallel System Construction)", IEEE Standard 1355-1995, IEEE, June 1996[7].

RD2 IEEE Computer Society, "IEEE Standard for Low-Voltage Differential Signals (LVDS) for Scalable Coherent Interface (SCI)", IEEE Standard 1596.3-1996, IEEE, July 1996.

RD3 Parkes SM, Gillions AR et al, "Digital Signal Processing Technology Study Final Report", ESTEC contract no. 8741/90/NL/JG(SC), British Aerospace Space Systems Ltd TP9459, 450 pages, September 1992.

RD4 Parkes SM et al, "Multi-Processor Evaluation", Section in "High Performance 32-bit DSP Evaluation Study" ESTEC contract no. 9736/91/NL/JG, Deutsche Aerospace, Analog Devices, BAe Space Systems Ltd, MHS, TCS, June 1994.

RD5 Parkes SM and Price SD, "DSP-Link Study Final Report", ESTEC contract no. 8741/90/NL/JG(SC), Report No. ESA-CR(P)-3864, Matra Marconi Space TR 0104, October 1994.

RD6 PALASIM[8]

RD7 Parkes SM, "On-Board Processing: Cheaper, Faster, Better", 32nd ESLAB Symposium on Remote Sensing Methodology for Earth Observation and Planetary Exploration, European Space Technology Centre (ESTEC), Noordwijk, The Netherlands, 15-18 September 1998, European Space Agency (ESA) publication SP-423, ISBN 92-9092-692-9.

RD8 Parkes SM, "DSP (Demanding Space-based Processing!): the Path Behind and the Road th International Workshop on Digital Signal Processing Techniques for Space Applications, European Space Technology Centre (ESTEC), Noordwijk, The Netherlands, 23-25 September 1998, Sponsored by IEEE Benelux Chapter on Communications and Vehicular Technology, European Space Agency (ESA) publication no. WPP-144, paper 4.1.

RD9 DIPSAP I[9]

RD10 DIPSAP II[10]

RD11 Parkes SM et al, "Review of Standard and Status", Digital Interface Circuit Evaluation Study WP1000 Technical Report, Document No. UoD-DICE-TN-1000, ESA Contract No. 12693/97/NL/FM, University of Dundee, July 1998.

---

[6] During the DRAFT stages within the development of the SpaceWire Standard, most of the references below will be available on the LITERATURE and COMPONENTS-BOARDS section of http://www.estec.esa.nl/tech/spacewire .

[7] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 455 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (http://www.standards.ieee.org/).

[8] Further information on Solid State Mass Memories can be found at ftp://ftp.estec.esa.nl/pub/ws/wsd/ssmm/intronew.htm .

[9] Find further details at http://www.cordis.lu/esprit/src/omi06347.htm

[10] Find further details at http://www.cordis.lu/esprit/src/omi08068.htm

RD12 IEEE, "The New IEEE Standard Dictionary of Electrical and Electronics Terms", IEEE Standard 100-1992, IEEE, 1992.

RD13 Konig W, "Rosetta EMC Requirements Specification", RO-DSS-RS-1002, Issue 2, Daimler-Benz Aerospace, January 1998.

RD14 Gottschlag D and Aulton J, " ENVISAT-1 EMC Requirements Specification", PO-RS-DOR-PL-0006, Issue 2, Dornier GmbH, February 1994.

RD15 Parkes SM and Tuominen T, "EMC Test Results", Digital Interface Circuit Evaluation Study WP3000 Technical Report, Document No. UoD-DICE-TN-3000, ESA Contract No. 12693/97/NL/FM, University of Dundee, April 1999.

RD16 Parkes SM, "High-Speed, Low-Power, Excellent EMC: LVDS for On-Board Data Handling", Proceedings of the 6th International Workshop on Digital Signal Processing Techniques for Space Applications, ESTEC, Sept. 1998.

RD17 IEEE Computer Society, "IEEE Standard for a High Performance Serial Bus", IEEE Standard 1394-1995, IEEE, August 1996.

RD18 May MD, Thompson PW & Welch PH, "Networks, Routers & Transputers: Function, Performance and Application", IOS Press, ISBN 90 5199 129 0, 1993.

RD19 Rosello J, Parkes SM and Christen AZ, " From IEEE-1355 to SpaceWire", Proceedings DASIA'99, Lisbon, Portugal, May 1999, To be published.

RD20 ESA-PSS-07-101, Packet Utilisation Standard

RD21 ECSS-Procedure-13, Rules for the drafting and presentation of ECSS Standards[11]

RD22 ECSS-P-00, Standardisation policy

RD23 ECSS-P-001, Glossary of terms

RD24 ECSS-E-00, Space Engineering - Policy and principles

RD25 ECSS-E-10, Space Engineering - System Engineering

RD26 ECSS-E-10-02, Space Engineering - Verification

RD27 ECSS-E-10-03 Draft, Space Engineering - Testing

RD28 ECSS-E-20, Space Engineering - Electrical and Electronic

RD29 ECSS-E-50, Space Engineering - Communications

RD30 ECSS-E-70, Space Engineering - Ground Systems and operations

RD31 ECSS-M-00, Space project management – Policy and principle

RD32 ECSS-Q-00, Space product assurance – Policy and principle

RD33 ECSS-Q-60, Space product assurance - EEE components

RD34 ECSS-Q-70, Space product assurance - Materials, Mechanical Parts & Processes

RD35 ECSS-Q-80, Space product assurance - Software Product Assurance

RD36 ISO/IEC Guide 25, General requirements for the competence of testing and calibration laboratories[12]

---

[11] ECSS publications are available from the ESA/ECSS secretariat (http://www.estec.esa.nl/ecss/)

[12] ISO/IEC publications are available from the International Organization for Standardization, 1, rue de Varembé, Case postale 56, CH-1211 Genève 20 (http://www.iso.ch) and from the International Electro-technical Commission , 1, rue de Varembé, Case postale 56, CH-1211 Genève 20 (http://www.iec.ch) .

RD37 ISO/IEC Directives, Part 3, Rules for the structure and drafting of International Standards, 3rd edition, 1997