INTERNATIONAL STANDARD

ISO/IEC 14165-222

First edition
2005-05

Information technology –
Fibre channel –

Part 222:
Single-byte command sets-2 mapping protocol (FC-SB-2)

Reference number
ISO/IEC 14165-222:2005(E)

INTERNATIONAL STANDARD

ISO/IEC 14165-222

First edition
2005-05

Information technology – Fibre channel –

Part 222: Single-byte command sets-2 mapping protocol (FC-SB-2)

PRICE CODE XC

For price, see current catalogue

Tables

Figures

INFORMATION TECHNOLOGY – FIBRE CHANNEL –

PART 222: Single-byte command sets-2 mapping protocol (FC-SB-2)

FOREWORD

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

2) In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

3) All users should ensure that they have the latest edition of this publication.

4) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC or ISO National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication of, use of, or reliance upon this ISO/IEC publication or any other IEC or IEC/ISO publications.

5) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

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

International Standard ISO/IEC 14165-222 was prepared by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

Introduction

FC-SB-2 describes the Fibre Channel protocol mapping for the Single-Byte Command Code Sets. The protocol is intended for use with ANSI X3.230, FC-PH, and its subsequent enhancements which specify the Fibre Channel Physical and Signalling Interface. FC-SB-2 is one of a number of Fibre Channel protocol mappings, referred to as FC-4s.

The reader should be familiar with FC-PH, Single-Byte Command Code Sets Connection (SBCON) Architecture (ANSI X3.296-1997).

Figure 1 shows the relationship of this FC-4 proposed working draft standard (highlighted rectangle) with other Fibre Channel standards and draft proposed standards.

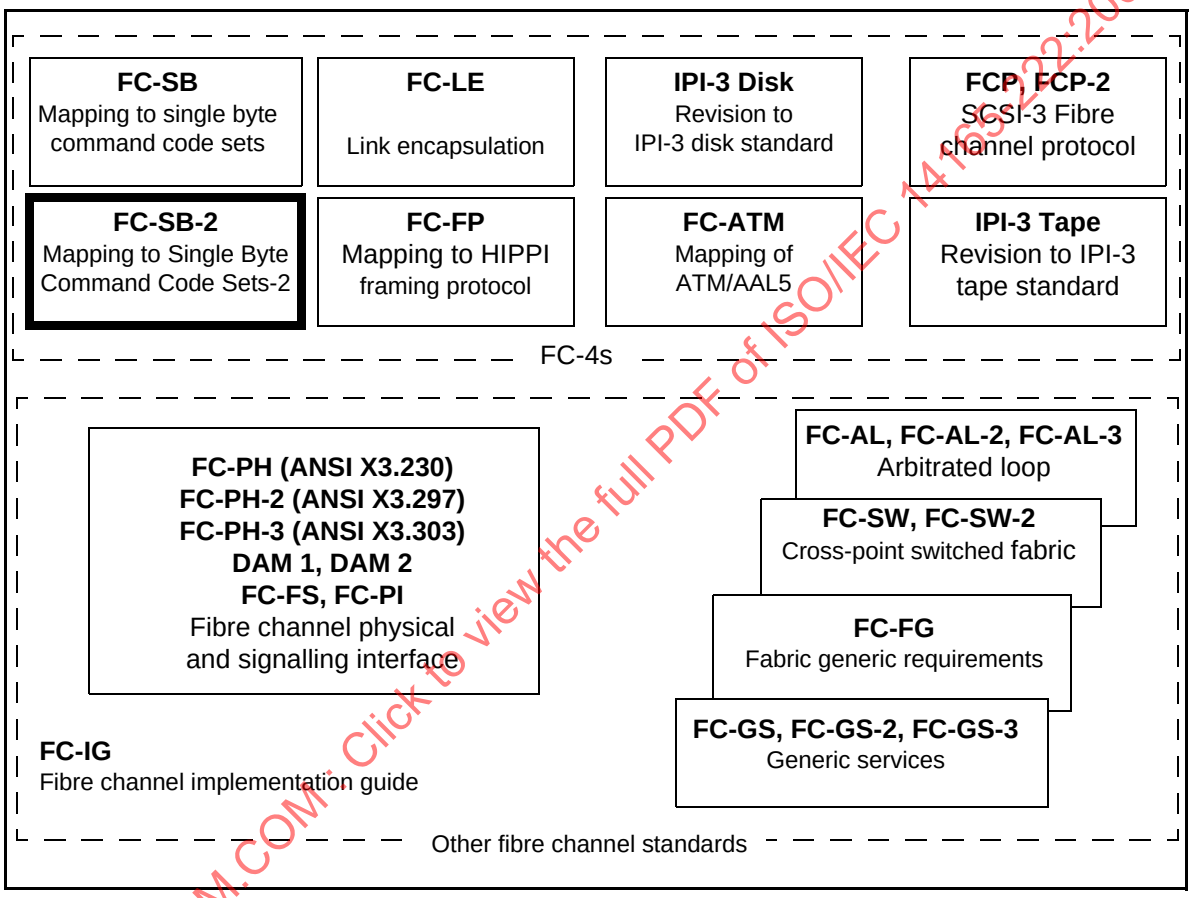

Figure 1 – Document relationship

The information presented in this document is grouped into clauses:

Clause 1, “Scope,” gives a general introduction to the document.

Clause 2, “Normative references,” lists the standards which are referenced in the text and which constitute provisions of this document.

Clause 3, “Definitions and conventions,” describes the basic elements, acronyms, naming conventions, and terminology used in this document.

Clause 4, “Structure and concepts,” provides an overview of the structure, concepts, configurations, and mechanisms used in this document and FC-PH, reference [1].

Clause 5, “FC-PH Link Control,” describes how FC-2 link services, in conjunction with the link header, are used to perform SB-2 link-level and device-level functions.

Clause 6, “Link-Level functions,” describes the link-level services and link-control functions required to establish and maintain the operational state of the FC-link.

Clause 7, “N_Port link initialization,” describes an initialization process, in terms of the hierarchical steps (procedures), that need to be performed over the FC-link in order to establish link-level and device-level communication.

Clause 8, “SB-2 Information units,” describes the SB-2 Information Unit (IU) types, and accompanying formats, that are transferred between conformant SB-2 ULPs.

Clause 9, “Device-Level functions and protocols,” describes the functions and protocols to be used in the execution of I/O operations, exchange of control information, and device-level recovery.

Clause 10, “Link error detection,” describes SB-2 level error detection mechanisms that are used in conjunction with FC-PH level error detection to insure the integrity of the FC-link and the data transferred.

Clause 11, “Error recovery actions,” describes the SB-2 link-level and device-level recovery actions taken for FC-PH and SB-2 errors.

Annex A, “Fabric address assignment,” describes a fabric address-assignment scheme which allows a channel to construct the three-byte N_Port ID of a given control unit from a configuration record which contains only a single byte entry for each control unit.

Annex B, “Correlation of exchanges of an exchange pair,” describes some of the facilities available for correlation of an outbound and inbound exchange of an exchange pair.

Annex C, “LRC calculation,” provides the procedure and an example of the Longitudinal Redundancy Check (LRC) calculation.

Annex D, “Status/chaining summary,” summarizes the conditions controlling whether the control unit breaks or continues command chaining and when to present status to the channel.

INFORMATION TECHNOLOGY – FIBRE CHANNEL –

PART 222: Single-byte command sets-2 mapping protocol (FC-SB-2)

1 Scope

This part of ISO/IEC 14165 describes a communication interface between a channel and I/O control units that utilize the Single-Byte Command Code Sets (SBCCS) as implemented in a wide range of data processing systems. It employs information formats and signalling protocols that provide a uniform means for communicating with various types of I/O control units, facilitating a high bandwidth, high performance and long distance information exchange environment. The signaling protocols and information exchanges are defined at a layer (FC-4) to compatibly utilize the link services and other functions provided by the ANSI Fibre Channel Physical and Signaling Interface (FC-PH) architecture (ANSI X3.230-1994, reference [1]). This FC-4 Upper Level Protocol is referred to as the Fibre Channel-Single-Byte-2 Command Code Sets Mapping Protocol (or for brevity, SB-2).

2 Normative references

2.1 General information concerning references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

Copies of the following documents may be obtained from ANSI: Approved ANSI standards, approved and draft international and regional standards (ISO, CEN/CENELEC) and approved foreign standards (including BSI, JIS and DIN).

2.2 Approved references

[1] ANSI X3.230-1994, *Information Technology – Fibre Channel Physical and Signaling Interface (FC-PH)*, with amendment 1, AM1-1996.

[2] ANSI X3.297-1997, *Information Technology – Fibre Channel – Physical and Signalling Interface-2 (FC-PH-2)*

[3] ANSI X3.303-1998, *Fibre Channel – Physical and Signalling Interface-3 (FC-PH-3)*

[4] ISO/IEC 14165-131:2000, *Information technology – Fibre Channel – Part 131: Switch Fabric Requirements (FC-SW)*

[5] ANSI X3.296-1997, *Single-Byte Command Code Sets Connection Architecture (SBCON)*

[6] ISO/IEC 9314-2:1989, *Information processing systems – Fibre Distributed Data Interface - Part 2: Token Ring – Media Access Control (FDDI-MAC)*

2.3 References under development

When drafting this document, the following referenced standards were still under development. For information on the current status of the document, or regarding availability, contact the relevant standards body or other organizations, as indicated.

NOTE For more information on the current status of a document, contact the Secretariats of the relevant organisations.

[7] ANSI INCITS-230-1994/AM2-1999, *Fibre Channel-Physical and Signaling Interface (FC-PH) Amendment 2*

[8] ISO/IEC 14165-251, *Information technology – Fibre Channel – Part 251: Framing and Signaling (FC-FS)*

3 Definitions and conventions

For SB-2, the following definitions and conventions apply.

3.1 Definitions

3.1.1 channelor: an entity, typically of a host computer, which consists of one N_Port and elements which perform the functions specified by SB-2 to provide access to I/O devices by means of control units or emulated control units.

3.1.2 channel-command word (CCW): (1) A control block which contains an I/O request. (2) A structure of a specific system architecture which specifies the command to be executed along with parameters.

3.1.3 Channel image: a single ULP instance of a channel having the logical appearance of a channel.

3.1.4 Channel-path identifier (CHPID): a system-unique 8-bit value assigned to each installed channel path of a system.

NOTE See 4.4 for information regarding channel paths.

3.1.5 Channel program: A single channel-command word or a sequence of channel-command words executed sequentially that control a specific sequence of channel operations.

3.1.6 Channel-to-channel connection: A channel-to-channel (CTC) connection is an association between two channels which allows two channels to exchange IUs on an exchange pair.

A CTC connection is established through an emulated control unit provided by one of the channels. A CTC connection exists when a logical path is established between one channel and the emulated control unit provided by the other channel.

3.1.7 Connection: Between a channel and control unit, an association established after the successful transfer of IUs that constitute an exchange pair resulting in two open exchanges, one inbound and the other outbound, and both occurring between the channel and the control unit.

A connection, once established, remains until one or both of the exchanges of the exchange pair is no longer open. Typically, an exchange pair exists for the duration of execution of an I/O operation or a chain of I/O operations.

3.1.8 Control unit: a physical or emulated entity, consisting of at least one N_Port and elements which adapt the characteristics of one or more I/O devices to allow their attachment to the N_Port of a channel.

3.1.9 Control-unit image: a single ULP instance of a control unit having the logical appearance of a control unit.

3.1.10 Device: In this document, the term device is used to refer to an I/O device such as a printer, magnetic-tape unit and direct-access-storage device. The I/O device operation is regulated by a control unit that provides the logical and buffering capabilities necessary to operate the I/O device.

3.1.11 Device Information Block (DIB): a data block present in all SB-2 information units which contains a 12-byte DIB header, a 4-byte redundancy check field, and optionally a variable length DIB data field.

NOTE See 8.3 for additional information.

3.1.12 Disconnection: see removal of a connection by closing both exchanges of an exchange pair.

3.1.13 Exchange pair: consists of two FC-PH exchanges with sequence initiative in opposite directions. The two exchanges are linked together by the SB-2 ULP.

The exchange pair carries all the communication required between ULP images during a connection. One exchange is originated by the channel and the other is originated by the corresponding control unit.

3.1.14 FC-SB-2 Channel-to-channel adapter: a channel capable of emulating a control unit which is used by a program in one system to communicate with a program of another system.

The adapter provides a fully functional control unit including support of requests for logical paths and all of the functions of a control unit image (or images) on logical paths over which communication is to occur.

3.1.15 Image: a group of related processes.

Examples of an image are a single system or a single logical partition of a system behind an N_Port.

3.1.16 Inbound exchange: exchange of an exchange pair which originates from a control unit and that carries information to the channel.

3.1.17 Initiation IU: first IU of an exchange.

3.1.18 I/O operation: Execution of an I/O operation is accomplished by decoding, accepting, and executing a command by an I/O device.

One or more CCWs arranged for sequential execution form a channel program and are executed as one or more I/O operations, respectively.

3.1.19 Outbound exchange: exchange of an exchange pair which originates from a channel and that carries information from the channel to the control unit.

3.1.20 SB-2 offline condition: SB-2 offline condition is recognized when a receiver transitions from the FC-PH active state to the FC-PH OLS receive state.

3.1.21 ULP process: A function executing within a channel or control unit which conforms to the Upper Level Protocol (ULP) immediately above the SB-2 service interface.

3.2 Keywords

3.2.1 mandatory: indicates items required to be implemented as defined by this standard.

3.2.2 may: indicates flexibility of choice with no implied preference.

3.2.3 model-dependent: describes an item (for example, bit, field, code value, etc.) or a behavior (for example, number of retries) which is not defined by this standard and may be vendor defined.

3.2.4 optional: describes features that are not required to be implemented by this standard.

However, if any optional feature defined by this standards is implemented, it shall be implemented as defined in this standard.

3.2.5 reserved: refers to bits, bytes, words, fields and code values that are set aside for future standardization.

Their use and interpretation may be specified by future extensions to this or other standards. A reserved bit, byte, word or field shall be set to zero by the sender and ignored by the recipient.

3.2.6 shall: indicats a mandatory requirement.

Designers are required to implement all such requirements to ensure interoperability with other products that conform to this standard.

3.2.7 should: indicats flexibility of choice with a preferred alternative; equivalent to the phrase "it is recommended".

3.2.8 vendor-specific: A keyword describing items (for example, a bit, field, code value, etc.) that are not defined by this standard and may be vendor defined.

3.3 Conventions

3.3.1 English usage conventions

In this standard, a number of conditions, mechanisms, sequences, parameters, events, states, or similar terms that do not have their normal English meaning are printed with the following conventions:

– the first letter of each word in uppercase and the rest lowercase (for example, Exchange, Class, etc.),

– a term consisting of multiple words, with the first letter of each word in uppercase and the rest lowercase, and each word separated from the other by an underscore (_) character. A word may consist of an acronym or abbreviation which would be printed in uppercase. (for example, N_Port),

– a term consisting of multiple words with all letters lowercase and each word separated from the other by a dash (-) character. A word may also consist of an acronym or abbreviation which would be printed in uppercase. (for example, device-level, CUE-with-busy, etc.).

All terms and words not conforming to the conventions noted above have the normal technical English meanings.

Numbered or listed items in this standard do not represent any priority unless explicitly indicated.

If a field or a control bit in a frame is specified as not meaningful, the entity which receives the frame shall not check that field or control bit.

Additional conventions applicable to this document are described in the following sections.

3.3.2 FC-PH

In order to support the SB-2 protocol, all FC-PH functions referenced in this document are required to be supported by a channel, control unit and Fabric (if present).

3.3.3 Bit numbering

All words defined by FC-PH use the convention that bit 31 is the Most Significant Bit (MSB) and bit 0 is the Least Significant Bit (LSB). (See FC-PH, reference [1].) All words defined by the SB-2 protocol use the convention that bit 0 is the MSB and bit 31 is the LSB. All bytes defined by the SB-2 protocol use the convention that bit 0 is the MSB and bit 7 is the LSB. For both FC-PH and SB-2, byte 0 is the most significant byte of a word, and byte 3 is the least significant. In all figures, tables and text of this document, the most significant bit of a binary value is shown as the left-most bit. Figure 2 gives a comparison between the FC-PH convention and the SB-2 convention.

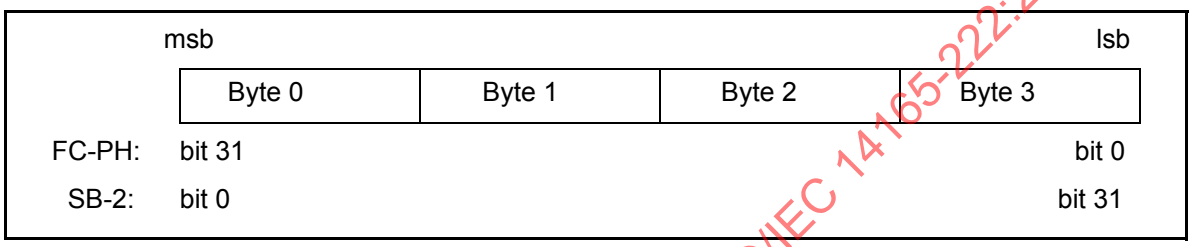

Figure 2 – FC-PH and SB-2 bit numbering conventions

3.3.4 Binary notation

Binary notation may be used to represent some fields. Single bit fields are represented using the binary values 0 and 1. For multiple bit fields, the binary value is enclosed in single quotation marks followed by the letter b. For example, a 4-byte field containing a binary value may be represented as ‘00000000 11111111 10011000 11111010’b.

3.3.5 Hexadecimal notation

Hexadecimal notation may be used to represent some fields. When this is done, the value is enclosed in single quotation marks and preceded by the word hex. For example, a 4-byte field containing a binary value of ‘00000000 11111111 10011000 11111010’b is shown in hexadecimal format as hex ’00 FF 98 FA’.

3.3.6 Abbreviations, acronyms, and symbols

Acronyms applicable to this document are listed below. See FC-PH, reference [1], for FC-PH acronyms and FC-FS, reference [8], for FC-FS acronyms. For all other acronyms, refer to the index.

ABTS Abort Sequence FC-PH BLS
AS Address Specific
BLS FC-PH Basic Link Service
CC Command Chain
CCW Channel-Command Word
CD Chain Data
CH Chaining
CHPID Channel Path Identifier

CI	Channel Initiated
CNP	CRC Not Provided
COC	Chain On Command immediate
CR	Command Retry
CRC	Cyclic Redundancy Check
CRR	Command Response Request
CTC	Channel-to-Channel
CTCA	Channel-to-Channel Adapter
DACK	Device-level Acknowledgment
DIB	Device Information Block
DU	Data chaining Update
E	End
EE	Early End
ELP	Establish Logical Path link control IU
ELS	FC-PH Extended Link Service
ES	Extended Status
FC	Fibre Channel
FC-PH	ANSI standard - FC Physical and signalling interface, reference [1]
FLOGI	F_Port Login FC-PH ELS
IU	Information Unit (See FC-PH, reference [1].)
IUI	Information Unit Identifier
LACK	Link level Acknowledgment link-control IU
LBY	Link level Busy link control IU
LIRR	Link-Incident Record Registration FC-FS ELS
LOGO	N_Port Logout FC-PH ELS
LPE	Logical Path Established link control IU
LPR	Logical Path Removed link control IU
LRC	Longitudinal Redundancy Check
LRI	Long Record/Immediate
LRJ	Link level Reject link control IU
LS_RJT	Link-Service Reject FC-PH ELS Reply
PCI	Program Control Interruption
PLOGI	N_Port Login FC-PH ELS
REX	Repeat Execute
RLIR	Registered Link Incident Record FC-FS ELS
RLP	Remove Logical Path link control IU
RNID	Request Node Identification Data FC-FS ELS
RRQ	Reinstate Recovery Qualifier FC-PH ELS
RSCN	Registered State Change Notification FC-FS ELS
RV	Residual count Valid
SB-2	Single Byte protocol on FC
SCR	State Change Registration FC-FS ELS
SDC	Self-Describing Component
SLI	Suppress Length Indication
SSS	Synchronize Send Status
SYR	Synchronize Response
TIN	Test Initialization link control IU
TIR	Test Initialization Result link control IU
ULP	Upper Level Protocol

4 Structure and concepts

4.1 Introduction

This clause provides an overview of the structure, concepts, and mechanisms used in Fibre Channel (FC), FC-PH, and SB-2.

Communication over Fibre Channel takes place between a pair of N_Ports. FC-PH defines all of the functions and protocols required to transfer information from one N_Port to another. Depending upon the configuration, the communicating N_Ports are between a channel and control unit or in the case of a channel-to-channel connection, between a channel and a channel emulating a control unit. For purposes of brevity, all future references to the term “control unit” assume both configurations of either a physical control unit or an emulated control unit unless stated otherwise.

SB-2 is a mapping protocol, referred to as an FC-4. An FC-4 is a mapping protocol, that maps a particular Upper Level Protocol (ULP) instance to FC-PH. The SB-2 ULP is based on the Single-Byte Command Code Set. See Bibliography for additional information.

An N_Port supports an FC-4 mapping protocol and its associated ULP. More than one FC-4 and ULP instances of the same type or different type may be supported by the same N_Port.

4.2 FC-4 General description

The FC-4 mapping layer uses the FC-PH protocol to transfer Upper Level Protocol (ULP) information. Each FC-4 describes, through mapping rules, how ULP processes of the same ULP type interoperate. An example of an active ULP process is an SB-2 I/O operation in progress at a device of a particular type such as a disk drive.

The protocols are described in terms of the elements visible in the stream of frames, sequences, and exchanges generated by a pair of communicating nodes transferring ULP information.

4.3 SB-2 overview

This clause describes the relationship between some of the SB-2 concepts and Fibre Channel concepts.

4.3.1 SB-2 instance

An SB-2 ULP instance may be either an N_Port-based SB-2 channel image, an N_Port-based SB-2 control unit image, or an N_Port-based emulated control-unit image. Information associated with the execution of an I/O operation and the operation of a device is transferred between a channel image and control unit image.

4.3.2 SB-2 protocols

Two levels of protocol are defined for SB-2:

– link-level,

– device-level.

SB-2 protocols are classified as either link-level or device-level, depending on whether they are for the purpose of managing a logical path and exchanging information over that path or for the purpose of managing the execution of an I/O operation.

The execution of an I/O operation requires both SB-2 link-level and device-level protocols at the channel and control unit. Information exchanged between a channel and control unit as a result of executing an I/O operation is transferred under the control of device-level protocols, which rely on link-level protocols for sending and receiving frames. (See 4.7.2 and 4.7.3.)

4.4 Channel-path elements

4.4.1 Overview of channel-path elements

The channel path provides the communication path between a channel and one or more control units. The physical elements that make up a channel path are a channel, possibly a Fabric, one or more control units, and one or more links.

4.4.2 Channel

A channel provides the functions specified by FC-PH and SB-2 and performs their prescribed protocols. Each channel shall provide one N_Port. When the N_Port of a channel is shared by multiple related processes, such as multiple operating systems, each channel path shall be logically and separately represented within the channel. This logical representation is termed a "channel image." (See 4.4.3.) A channel may also be capable of providing control unit emulation. (See 3.1.6 and 4.5.6.)

4.4.3 Channel image

A "channel image" has the logical appearance of a channel. Each channel image appears to be an independent channel although all channel images on a specific channel share the same N_Port and physical paths. The N_Port of a channel performs certain functions for all sharing channel images (for example, link synchronization and acquisition of address identifiers) and may perform functions for a single sharing channel image (for example, sending frames during the performance of link-level and device-level functions). The N_Port of a channel common to multiple channel images may perform functions for multiple channel images simultaneously by multiplexing work on each function and interleaving frames for each function on the link as allowed by FC-PH.

Figure 3 shows multiple channel images which are sharing the same N_Port.

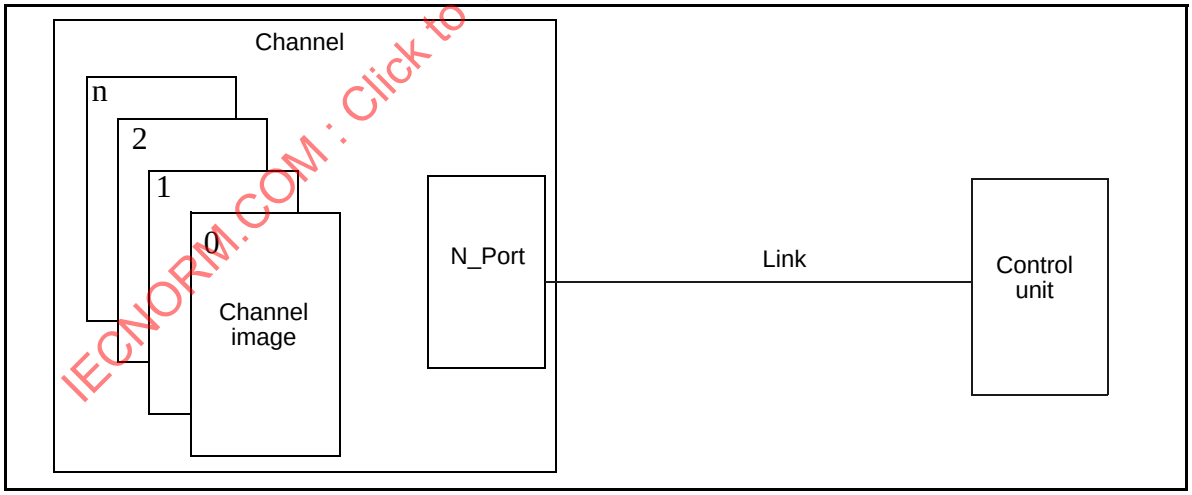

Figure 3 – Channel images

Either single or multiple channel images may exist at an N_Port. Each channel image denotes one SB-2 channel ULP instance.

NOTE Channel images represent different logical channel paths (logical channel paths or for brevity, logical paths) even though the channel images share the same N_Port. Therefore, there is one channel per channel path.

4.4.4 Control unit

A control unit provides the logical capability necessary to operate and control one or more devices and adapts, through the use of common facilities, the characteristics of each device to the N_Port provided by the channel. These common facilities which include N_Port functions and may include SB-2 functions provide for the execution of I/O operations, indications concerning the status of the device and control unit, control of the timing of data transfer over the channel path, and certain levels of device control.

Up to 256 devices may be attached to each control-unit image.

A control unit may have more than one N_Port in order to allow attachment to more than one link, each from a different channel or from a different F_Port on the same or a different Fabric. When the control-unit N_Port is attached to a link from a Fabric, the control unit and its devices may be physically accessible over that N_Port to all channels also attached to links from that Fabric.

When the N_Port on a control unit is shared among multiple channel paths, each channel path shall be logically represented separately within the control unit. This logical relationship is called the logical path. (See 4.5.5.) A control unit shall provide at least one logical path for each operational N_Port, however, the number of logical paths that a control unit permits is model dependent.

NOTE 1 Certain ULP operating systems have a dependency upon the existence of a single path to a control unit for a given channel for purposes of managing the I/O configuration. In order to satisfy this dependency, it is recommended that a control unit have only one N_Port which is configured to communicate with a given channel.

NOTE 2 A control unit may have more than one N_Port, each configured to communicate with a different channel.

NOTE 3 A control unit may be connected to more than one channel at the same time. When a control unit is connected to more than one channel at the same time, each connection may be for the same or a different device.

4.4.5 Control-unit image

A control-unit image has the logical appearance of a control unit. Each control-unit image appears to be an independent control unit, although all control-unit images common to one control unit may share the same common facilities and N_Ports. These common facilities may provide some or all of the SB-2 functions and protocols. Those SB-2 functions and protocols not provided by the common facilities shall be provided by the individual images.

FC-PH and SB-2 protocols operate for each control-unit image independent of all other control-unit images except for resolving contention for the common facilities shared among control-unit images. The N_Port of a control-unit common to multiple control-unit images may perform functions for multiple control-unit images simultaneously by multiplexing work on each function and interleaving frames for each function on the link as allowed by FC-PH.

Figure 4 shows multiple control-unit images which are common to a single N_Port.

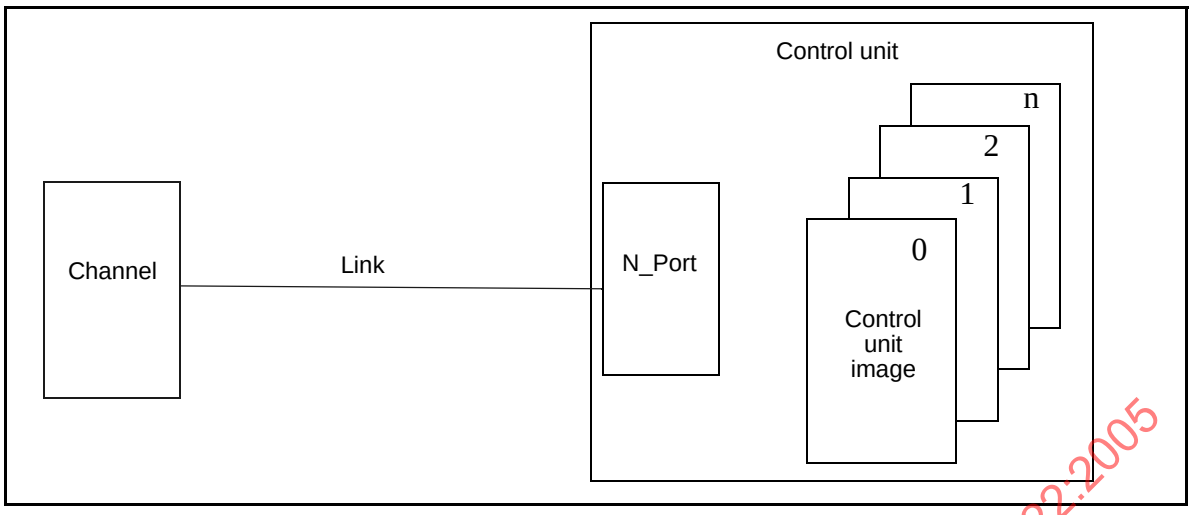

Figure 4 – Control unit images

NOTE Each control-unit image is architecturally identical, even though each image may provide a different device type); therefore, there is no change to the definition of a control unit and no need to use any term other than *control unit* in this document. The presence of multiple control-unit images becomes apparent only through the logical addresses used when logical paths are established. (See 4.7.4 and 4.5.5.)

4.4.6 Link

The link is fully described in FC-PH, reference [1].

4.5 Channel-path configurations

4.5.1 Channel-path configuration overview

Point-to-point and fabric channel path configurations are provided.

NOTE SB-2 does not preclude NL_Port usage, however, the operation of SB-2 on a loop is not supported because not all potential problems associated with operating SB-2 on a loop have yet been considered.

4.5.2 Point-to-point configuration

A channel path that consists of a single link interconnecting one or more control-unit images to one or more channel images forms a point-to-point configuration. A point-to-point configuration shall be permitted between a channel and control unit only when a single control unit is defined on the channel path or when multiple control-unit images all share the same N_Port. Figure 5 shows a point-to-point configuration consisting of a single channel image and single control unit image, and Figure 6 shows a point-to-point configuration consisting of multiple channel and control unit images.

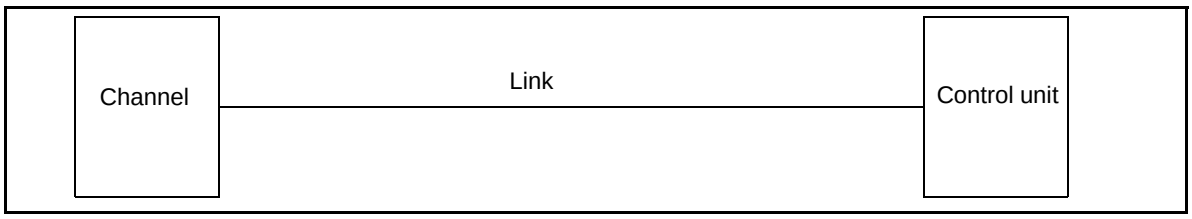

Figure 5 – Point-to-point channel-path configuration (single logical image)

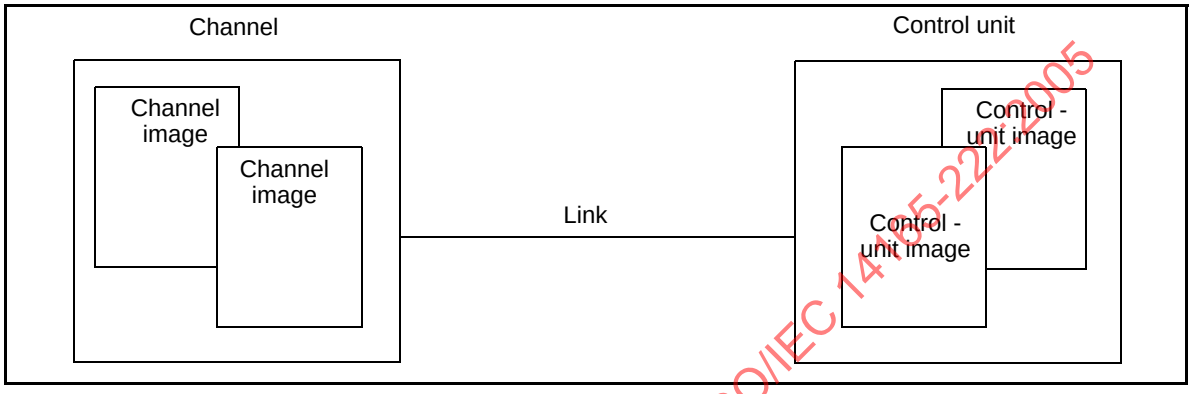

Figure 6 – Point-to-point channel-path configuration (multiple logical images)

The channel N_Port and the control-unit N_Port resolve contention among the logical images for access to the link.

A maximum of one link shall be attached to the channel in a point-to point channel-path configuration. The maximum number of control-unit images that shall be addressed over the link is 256; therefore, the maximum number of devices that shall be addressed over a channel path configured point-to-point is equal to 256 control-unit images times 256 devices per control-unit image, or 65536.

4.5.3 Fabric configuration

A channel path that consists of one link which interconnects one or more channel images with an FC switch fabric (or for brevity Fabric) and one or more links, each of which interconnects the Fabric with one or more control-unit images, forms a fabric configuration. (See Figure 7.)

Multiple channel images and multiple control-unit images may share the resources of the FC links and the Fabric, such that multiplexed I/O operations may be performed.

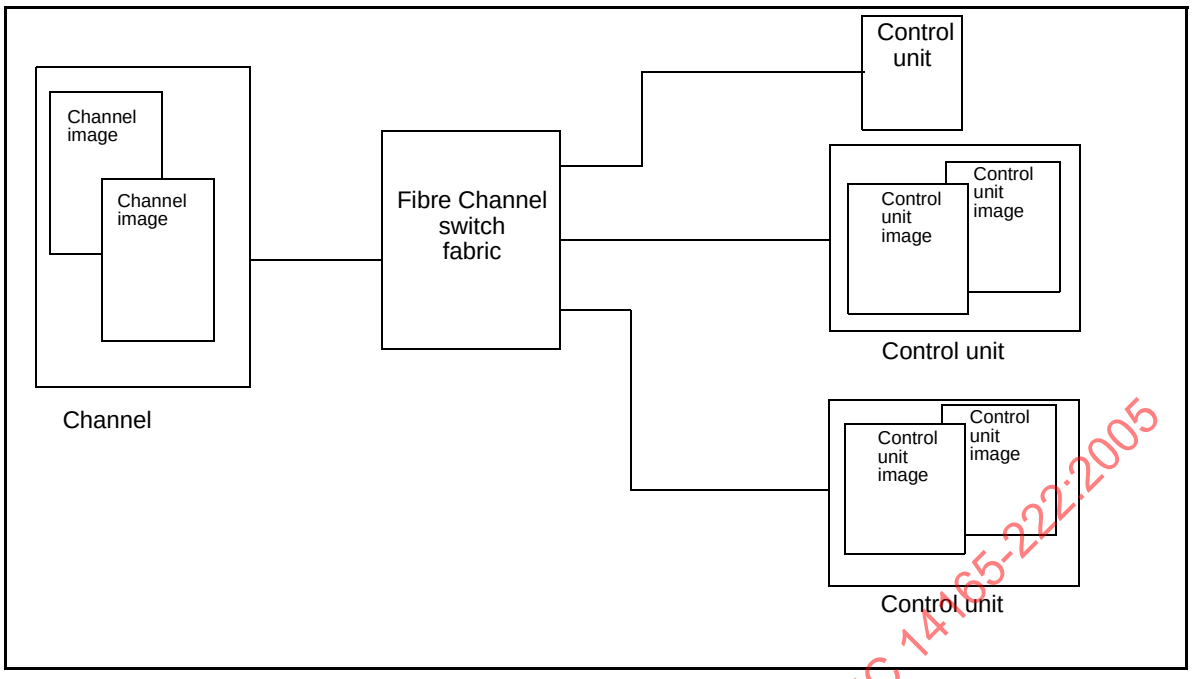

Figure 7 – Fabric channel-path configuration (multiple channel images)

Channels and control units may be attached to the links from the Fabric in any combination, depending on configuration requirements and depending on available resources in the Fabric. Sharing a control unit through a Fabric means that communication with the control unit may take place over one N_Port in the case where the control unit has only one link to the Fabric or over multiple N_Ports in the case where a control unit has more than one link to the Fabric.

A maximum of one link shall be attached to the channel in a fabric channel path configuration. The same control unit and device addressing capability exists as for the point-to-point configuration, however the attachment flexibility is greatly increased with a Fabric through the capability to use the 24 bit N_Port address, to access multiple control units.

For a detailed description of the FC switch fabric refer to FC-PH, reference [4]. The N_Port of the channel or control unit attaches to an F_Port of the Fabric. The F_Port is required to support at least Class 2 and Class 3 service.

NOTE FC-SB-2 supports single-switch fabrics. It is intended to support multiple-switch fabrics, but the special issues that might occur in multiple-switch fabrics have not yet been fully considered.

4.5.4 Physical path

The communication path between a channel and a control unit is composed of two different parts, the physical channel path, or, for brevity, physical path, and the logical channel path, or, for brevity, logical path. The physical path is the link, or the interconnection of two or more links by a Fabric, that provides the physical transmission path between a channel and a control unit. The logical path is the relationship that exists between a channel and a control unit for device-level communication during execution of an I/O operation and presentation of status. (For more information, see 4.5.5 below and 4.7.3.)

4.5.5 Logical path

A logical path is the relationship established between a channel image and control-unit image which identifies a communication path over which device-level information may be transferred and to which certain device-level allegiances may be associated. The logical path is established as part of the channel and control unit initialization procedures by the exchange of SB-2 link-control IUs.

When the logical path is established, device-level communication is allowed on that logical path. All device-level protocols depend upon the existence and identity of logical paths. Device-level protocols are executed over an established logical path by means of the exchange of information units between the channel and the control unit. When a logical path is not established, only link-level communication is permitted.

A logical path is identified within a channel or control unit by the combination of a 24-bit N_Port address identifier assigned to the channel, a 24-bit N_Port address identifier assigned to the control unit, an 8-bit logical address assigned to the channel image establishing the logical path, and an 8-bit logical address assigned to the control-unit image to which the logical path is being established. Both the channel and the control unit recognize a logical path by the same combination of addresses. The two 24-bit address identifiers define the N_Ports to which the images are associated, and the two 8-bit logical addresses identify the images for which the logical path exists.

A channel image may have one logical path to each control-unit image. The number of logical paths a channel or control unit permits depends on system requirements and is model dependent.

Even though multiple logical paths may be associated with the same N_Port on the control unit the channel paths represented by these logical paths are treated as if each were associated with a separate N_Port interface; each logical path represents a logical relationship to a channel.

Allegiances, I/O operations, system resets, and path groups for a particular system are identified by means of the logical path established by the channel for that system. To a control unit, each logical path represents a different channel path or logical subdivision of a channel path.

When multiple control-unit images are provided, their presence is apparent only through associated logical paths.

4.5.6 Channel-to-channel communication

4.5.6.1 Channel-to-channel communication overview

A channel supporting channel-to-channel communication shall be able to accept a request to establish one or more logical paths and shall be able to provide all of the functions of a control unit image on logical paths on which communication is to take place. In addition to providing all of the functions required of a channel for other logical paths, a channel supporting channel-to-channel communication shall provide all of the functions required of a control unit on a logical path for which it has accepted an Establish Logical Path (ELP) IU. (See 6.4.2.) When a channel supports CTC connection, the N_Port of the channel is also the N_Port of the emulated control unit.

Figure 8 shows a Fabric Channel-Path configuration in which a control unit image of one channel, referred to as “channel A” is communicating with another channel, referred to as “channel B.” In the configuration shown, one or more of the control unit images of channel A has accepted a request from channel B to establish a logical path, and the control unit image within channel A is providing all of the functions required of a control unit on the logical path to a corresponding channel image of channel B. Both channel A and channel B may also have established logical paths with other control units attached to the switch.

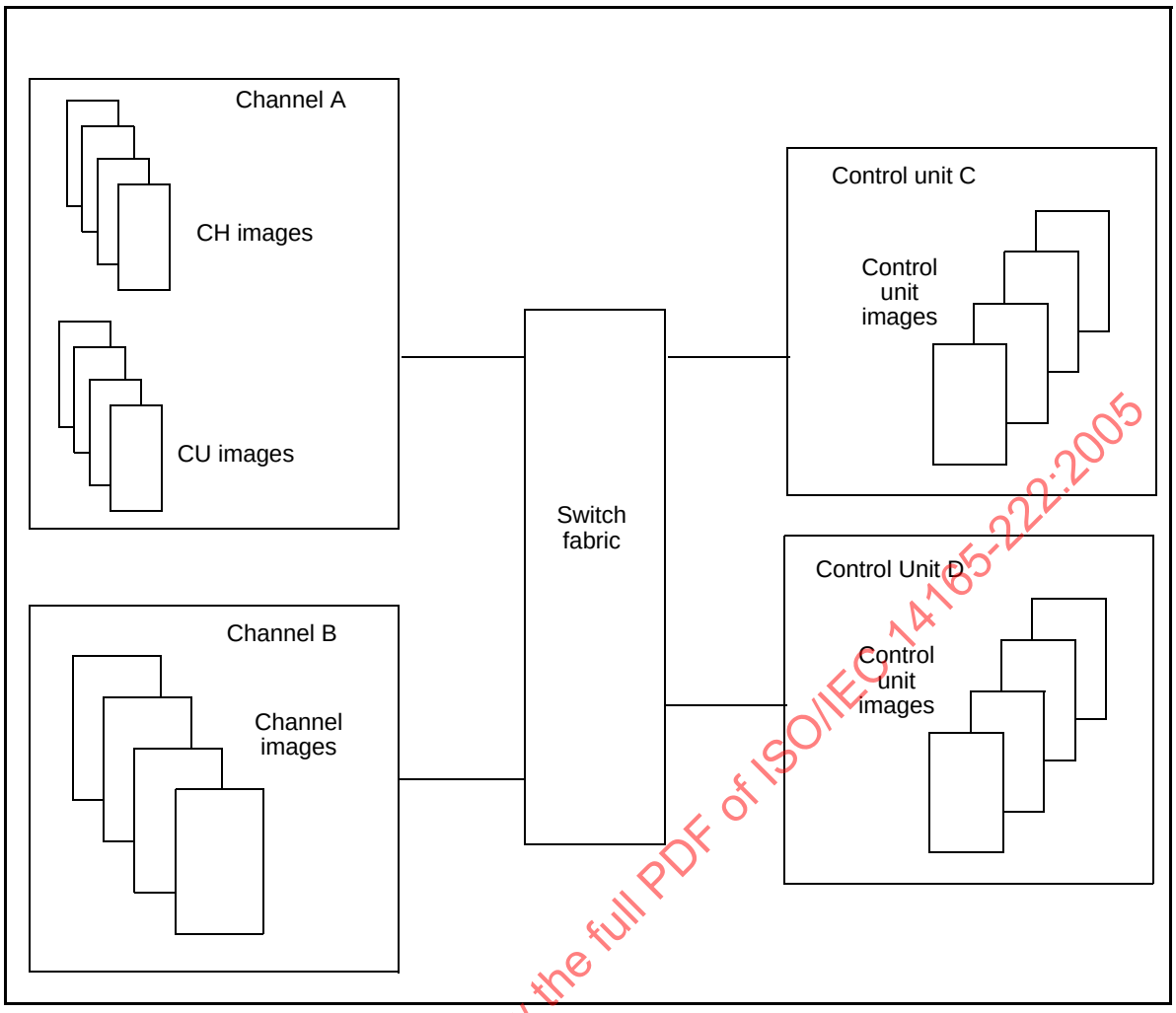

Figure 8 – Fabric channel-path configuration (multiple channel images, CTC connection)

4.5.6.2 FC-SB-2 channel-to-channel adapter

The FC-SB-2 channel-to-channel adapter is a channel capable of emulating a control unit and is used by a program in one system to communicate with a program of another system. The adapter provides the data path and synchronization for a data transfer between two channels; however, in so doing, it presents a view as though it were a fully functional control unit including support of requests for logical paths and all of the functions of a control unit image (or images) on logical paths over which communication is to occur. (All references to functions supported by a “control unit” apply equally to the case where a channel is emulating a control unit.)

4.6 Information transfer

SB-2 information is transferred on an FC link in FC-4 information units. An information unit is a collection of data that is organized according to a particular structure depending on the function being performed or the data content. The types and structures of the IUs defined for SB-2 are described in Figure 12. An information unit is transferred in a sequence which shall be sent as one or more FC-PH device-data frames.

4.7 Protocols

4.7.1 Protocol overview

Protocols for the SB-2 interface are classified as either link level or device level, depending on whether they are for the purpose of managing a channel path and exchanging information over that channel path or for the purpose of managing the execution of an I/O operation. The link level describes the physical characteristics of a channel path and the associated protocols required for the transmission and reception of frames. The link-level protocols for SB-2 are defined both in FC-PH and in this document. (See Clause 6.) The device level primarily relates to the protocols associated with the execution of an I/O operation for a specific device. The device level protocols for SB-2 are defined in this document and they correspond to a new FC-4 definition. These device-level protocols are similar to those of SBCON but different in that multiplexing at the frame level and streaming of commands and data are provided. The frame level multiplexing provided is based on Fibre Channel protocols. This multiplexing allows for multiple simultaneous exchanges, each concurrently transferring information, between a channel and control unit over the same N_Port to different devices.

The presence of multiple channel images or multiple control-unit images only becomes apparent through the logical paths that are established. All device-level protocols depend on the existence of and identity of these logical paths. When the logical path is known, the particular logical images for which the link-level and device-level facilities are dedicated at any one time is also known.

The execution of an I/O operation requires link-level and device-level protocols to be present in both the channel and the control unit. Information exchanged between a channel and control unit as a result of executing an I/O operation is transferred under the control of the device-level protocols with the aid of the link-level protocols for sending and receiving frames. The failure to satisfy protocols at either level results in an error being recognized.

Figure 9 shows the relationship between the link level, device level, physical path, and logical path.

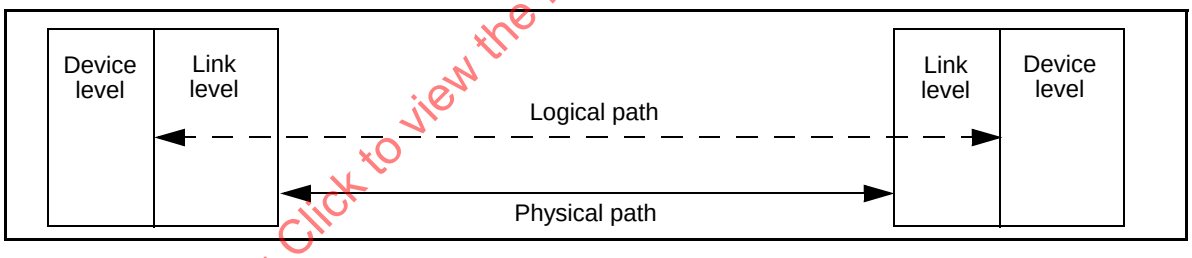

Figure 9 – Relationships among link level, device level, physical path and logical path

4.7.2 Link level

The link-level protocols described in this subclause refer to FC-PH link services and extended link-services. Additionally, SB-2 link-level functions are included.

All communications over a channel path use link-level protocols. Link-level protocols are used to establish and maintain the physical and logical paths in order to provide for transmission and reception of frames and primitive sequences. They include functions such as acquiring address identifiers, establishing frame credit, generating the basic frame structure, coordinating the protocols for frame transmission, and checking the integrity of information sent in frames. (See FC-PH, reference [1] and Clause 6.)

Certain functions require the exchange of information between the channel and control unit at the link level. These functions are performed by means of link-level protocols. Initialization, for example, is such a function. Before communication over a channel path is allowed to occur, and before an I/O operation is allowed to be executed, some form of initialization shall be completed according to the requirements of the system and according to the specifications of the architectural definition. (See FC-PH, reference [1] for link initialization and login definitions.)

Once the N_Port of a channel has performed initialization procedures for a control unit, including logging in and establishment of a logical path to a control-unit image, that control-unit image is considered operational and capable of executing I/O operations over that channel path.

4.7.3 Device level

Device-level protocols allow for transfer of information specifically related to an I/O operation, transfer of status of a device or control unit, and recovery when errors are detected by the channel or the control unit.

The types of device-level information transferred between a channel and control unit that use device-level protocols are: commands, data, control information, and status. Commands are provided by the channel program being executed. Data is the information associated with the command being executed that is to be transferred to or from the device. Control information includes functions that manage the transfer of data between the channel and control unit and functions that manage execution of the I/O operation. Status describes the results of the completion of an I/O operation (successful or unsuccessful) or provides information not associated with an I/O operation that is to be reported to the channel.

Device-level information is transferred between a channel and a control unit in SB-2 information units (IUs). (See Figure 12.) SB-2 information units are transferred using both link-level and device-level functions and protocols. For example, when the channel receives initiative to start an I/O operation, the device-level functions and protocols obtain the command and other parameters from the current CCW and insert them into the appropriate fields within a command IU. When the command IU is ready for transmission, link-level functions and protocols provide additional information, for example, address identifiers and exchange ID in the frame header and then coordinate the actual transmission of the frame on the channel path.

4.7.4 Addressing

Two levels of addressing are used for the link-level and device-level structure. All control units and channels use both levels of addressing, link level and device level. Link-level addressing identifies the N_Port, which in turn identifies the physical path within the channel-path configuration to be used for communication between a channel and a control unit. (See FC-PH, reference [1], clause 18.3.) Link-level addressing also identifies the channel or control-unit images associated with an N_Port.

Device-level addressing identifies the particular device to the particular channel or control unit, once the physical path, as designated by the link-level addressing, has been determined. Link-level addressing is considered to be the first level of addressing because it determines the N_Port, physical path, and the sharing logical image; device-level addressing is considered to be the second level of addressing because it determines the particular device once the N_Port and logical image are determined. Both levels of addressing have specific address-assignment requirements.

Link-level addressing requires each N_Port to be assigned a 24-bit address, called the address identifier. The address identifier of the sender of a frame is in the S_ID field in the FC-PH header, and the address identifier or the recipient of a frame is in the D_ID field of the FC-PH header. (See 5.5.3.) An N_Port that does not have an address identifier assigned is *unidentified*; an N_Port that has an ad-

dress identifier assigned is *identified*. The assignment of an address identifier to an N_Port occurs when the N_Port performs initialization and the required login procedure. (See FC-PH, reference [1], clause 18.3 and clause 23.4.1.) When an unidentified N_Port performs initialization, it acquires its address identifier through the procedures defined for the login process.

Link-level addressing also requires that each channel and control-unit image be assigned an 8-bit address, called the logical address. The logical addresses of the sender and receiver of an IU are in the SB-2 header field of the IU. (See 8.4.) A channel or control-unit image is assigned a logical address when it is created. A logical address shall be unique at a channel or control unit N_Port but need not be unique on a channel path. The assignment of logical addresses and the method by which this assignment is performed are model dependent.

When either multiple channel images or control-unit images are created, they share a single N_Port for each link, and therefore, the same respective address identifier. A maximum of either 256 channel images or 256 control-unit images shall share the same N_Port. The combination of the address identifier and the logical address determines the image to which the device-level addressing applies.

Device-level addressing depends on the assignment of an 8-bit address, called the device address, to every device. A device address shall be unique on a logical path on a control unit but need not be unique on a channel path because the resultant combination of the N_Port address identifier, control-unit logical address, and the device address uniquely identifies a device on a channel path. The assignment of device addresses and the method by which this assignment is performed are model dependent. (See 8.4.4.)

Control units that attach to more than one link provide an N_Port per link, with each N_Port having an assigned address identifier.

All IUs sent on a channel path rely on link-level-addressing information for determining the correct physical path to the required destination N_Port and, for certain IU types, for determining the appropriate image associated with that N_Port. Some IUs require only link-level addressing while others require both link-level and device-level addressing. An IU sent to perform a link-level protocol requires only link-level addressing, and an IU sent to perform a device-level protocol requires link-level addressing and in most cases device-level addressing. Device-level addressing identifies the device that is the source or destination of the information in the IU.

5 FC-PH Link Control

5.1 FC-PH Link control overview

The FC-PH provides services which support the transfer of data between N_Ports in frames which consist of a frame header and a payload. This clause describes how FC-PH services are used to perform SB-2 link-level and device-level functions, and defines how the fields in the frame header are used to perform these functions. For a general description of FC-PH mechanisms and frame header fields which are unrelated to SB-2, see clauses 16 - 29 of FC-PH, reference [1].

5.2 Class of service

For reasons of simplicity and the need to keep protocol interlocks to a minimum, Class 3 service shall be used when sending all SB-2 IUs except for the test initialization IU. See 6.4.7. Class 2 service shall be used when sending the test initialization IU because the N_Port receiving the test initialization IU may be in an non-operational state. When this is the case, Class 2 provides the means for the fabric to provide timely notification of this fact, thereby avoiding costly timeouts and retry operations.

In certain classes of service, including Class 3, a Fabric is not required to deliver frames to a destination in the same order as they were sent by the source. If a Fabric is able to deliver frames to a destination in the same order as they were sent by the source, it indicates this capability during fabric login. The SB-2 ULP requires the Fabric to indicate support for sequential delivery during fabric login. See Clause 7 for additional information.

NOTE Since SB-2 requires the test initialization IU to be sent in Class 2, N_Ports which support SB-2 are required by FC-PH to indicate support for Class 2 at login. This also requires login to be performed using Class 2 service because login is valid only in the class of service used or a class with a higher numerical value. See FC-PH, reference [1] for additional information.

5.3 Buffer-to-buffer credit reclamation

During operation over extended periods of time, errors may occur which result in the loss of one or more R_RDY primitive signals. The loss of R_RDY primitive signals affects the available buffer-to-buffer credit at an N_Port and as a result, may affect that N_Port's ability to send frames and maintain optimum performance. Designs should be tolerant of the loss of some R_RDY primitive signals for short periods of time but also shall be capable of taking some action to offset the cumulative effect of errors over longer periods of time. The cumulative loss of R_RDY primitive signals over long periods of time may result in a reduction in available buffer-to-buffer credit to a point where performance is significantly affected or, in the extreme case where the available buffer-to-buffer credit goes to zero, the inability to send a frame and an FC-PH link time-out error.

To prevent the loss of R_RDY primitive signals from resulting in a performance degradation or FC-PH link time-out error, N_Ports shall perform a model-dependent buffer-to-buffer credit reclamation procedure which reinitializes the available buffer-to-buffer credit at both ends of the link. The preferred reclamation procedure is specified in FC-FS, reference [8].

When either of two attached N_Ports do not support the preferred buffer-to-buffer credit reclamation procedure, an alternative procedure shall be performed periodically at model-dependent time intervals. The time interval of the model-dependent procedure should be as large as possible provided that buffer-to-buffer credit reclamation is performed before the loss of R_RDY primitive signals results in a performance degradation. To minimize the number of IUs lost as a result of buffer-to-buffer credit reclamation, the buffer-to-buffer reclamation procedure is not initiated when open exchanges exist, and an effort is made to quiesce the link before the procedure is performed.

NOTE 1 When the preferred buffer-to-buffer credit reclamation procedure is not performed, the channel may quiesce the link before performing buffer-to-buffer credit reclamation by not initiating new exchange pairs and allowing all channel programs to complete and exchange pairs to end. Control units may quiesce the link by completing all channel programs which are in process and by returning control unit busy status to new commands or control functions other than a system reset or purge path. Since efforts made to quiesce the link do not guarantee that no IUs are lost, occasional errors resulting from buffer-to-buffer credit reclamation should be expected.

NOTE 2 When the preferred buffer-to-buffer credit reclamation procedure is not performed, implementations should use the FC-PH link reset primitive sequence protocol to provide buffer-to-buffer credit reclamation. In addition, it is preferred that data queued in receive and transmit buffers before the protocol is initiated as well as data received into buffers during the time when the protocol is being performed be preserved and processed normally when the protocol is completed. For additional information on the link reset primitive sequence protocol, see FC-PH, reference [1].

NOTE 3 When the preferred buffer-to-buffer credit reclamation procedure is not performed, the suggested default time interval between the performance of each buffer-to-buffer credit reclamation procedure is approximately 24 hours. A means should be provided to adjust the time interval and randomly vary the time interval by up to 10% so that reclamation does not occur at predictable time intervals such as the same time each day. For fabric configurations, a means should be provided which minimizes the probability of simultaneous buffer-to-buffer credit reclamation by a large number of N_Ports because of the system performance impact which results. The means used to avoid simultaneous buffer-to-buffer credit reclamation is model-dependent but may include the use of a slightly different time interval by each N_Port or by causing the time interval of each N_Port to begin at a different time.

NOTE 4 When the preferred buffer-to-buffer credit reclamation procedure is not performed, unnecessary execution of an alternative buffer-to-buffer credit reclamation procedure should be avoided. An implementation should defer buffer credit reclamation for a model-dependent time interval after performing a procedure such as login or link initialization which initializes the available buffer-to-buffer credit. In a point to point configuration, only one of the N_Ports is required to initiate the buffer-to-buffer credit reclamation procedure because the procedure initializes the available buffer-to-buffer credit at both ends of the link. F_Ports are not required to perform the procedure since performance of buffer reclamation by the attached N_Port reinitializes the available buffer-to-buffer credit at both the N_Port and the attached F_Port.

5.4 SB-2 Sequences and exchanges

5.4.1 SB-2 sequences

When receiving frames from the link, FC-PH provides a mechanism which assembles sub-blocks of data contained in the payloads of one or more frames into a single data block called a sequence. Each FC-4 defines the contents of the sequences which are used for ULP functions. When the contents and usage of a sequence are defined by a ULP, it is referred to as an information unit (IU). For SB-2, IUs contain commands, device-level controls, link controls, and related functions. The IUs defined by SB-2 are summarized in Table 1, and the contents of SB-2 IUs conform to the general structure shown in Figure 12. The protocols for using IUs for SB-2 link-level and device-level functions are found in Clause 6 and Clause 9, respectively.

5.4.2 SB-2 exchange pairs

5.4.2.1 Exchange pair overview

IUs which a channel sends during the execution of an SB-2 link-control function or device-level function are restricted to one exchange, and IUs which a channel receives during the operation are restricted to a different exchange. The exchange on which the channel sends IUs is referred to as the outbound exchange, and the exchange on which the channel receives IUs is referred to as an inbound exchange.

Both inbound and outbound exchanges shall transfer IUs in a single direction. When both an outbound exchange and an inbound exchange simultaneously exist between a channel and a control unit for the execution of the same link-level or device-level function, an exchange pair is said to exist, and the control unit is said to be connected to the channel. A channel program which is executed in a single con-

nection uses only one exchange pair. If the connection is removed by the closing of the exchanges during the channel program, a new exchange pair shall be required to complete the channel program.

5.4.2.2 Initiating exchange pairs

5.4.2.2.1 General rules for initiating exchange pairs

A channel shall initiate an exchange pair by sending an IU which opens a new exchange (or, an initiation IU) as an unsolicited command or unsolicited control information category. A control unit shall initiate an exchange pair by sending an initiation IU as an unsolicited control or unsolicited data information category. See Table 1.

5.4.2.2.2 Exchange pairs for link level functions

A control-unit image that initiates an exchange pair for a link-level function shall wait for a response to the initiation IU before it initiates another exchange pair with the N_Port of the same channel if the exchange pair to be initiated is:

– for another link-level function, or

– for a device-level function not associated with a specific device.

5.4.2.2.3 Exchange pairs for device-fevel functions

A control-unit image that initiates an exchange pair for a device-level function not associated with a specific device shall wait for a response to the initiation IU before it initiates another exchange pair if the exchange pair to be initiated is:

– with the same channel image for a device-level function not associated with a specific device, or

– with the N_Port of the same channel for a link-level function.

A channel or control unit may initiate multiple exchange pairs, each for a different device or for the same device provided that there is no more than one exchange pair for a specific device on a logical path. A new exchange pair for a specific device on a logical path shall not be initiated when any of the following conditions apply:

– if an exchange pair already exists for the same device on the same logical path, or

– if the channel or control unit initiating the new exchange pair has initiated another exchange pair for the same device on the same logical path and not received a response.

If either of the above conditions apply, the channel or control unit shall wait for the existing exchange pair to be closed before initiating the new exchange pair for the device on the logical path.

5.4.2.3 Exchange pair processing

In order to avoid errors caused by lost IUs or by the inability to send an IU, sufficient resources should be held in reserve in order to support new exchange pairs for unexpected events. If an IU initiating a new exchange pair is received, sufficient resources should be available to properly receive the IU and to initiate a new exchange in response. If an unexpected event requiring a new exchange pair to be initiated occurs, resources should be available for supporting the new exchange pair.

After a channel or control unit has closed one of the exchanges of an exchange pair for a device on a logical path, an abnormal condition may result in the recognition of a valid initiation IU for the same device on the same logical path before the other exchange of the previous exchange pair has been closed. When this occurs and the initiation IU is not a purge path IU, the IU may be accepted and held until both exchanges of the previous exchange pair have been closed or a link-busy condition may be recognized. (See 6.4.10 and 9.3.1.)

5.5 FC-PH frame header fields

5.5.1 Frame header field overview

The FC-PH frame header identifies the source and destination of a frame, the exchange and sequence to which a frame belongs, the order in which a frame was sent, and the type of information in the frame payload. The frame header consists of a frame routing field (R_CTL), source and destination link address fields (S_ID and D_ID), an FC-4 identification field (TYPE), an optional header control field (DF_CTL), fields which identify and control FC-PH sequences and exchanges (F_CTL, OX_ID, RX_ID, SEQ_ID, SEQ_CNT), and a parameter field whose function depends on the particular frame type. The usage of these fields for frames which are part of an IU is described below. The frame header format is described in FC-PH, reference [1], and FC-FS, reference [8]. For a definition of the settings of frame header fields in frames which contain FC-PH link-control functions, FC-PH basic link services, and FC-PH extended link services, see FC-PH, reference [1] and FC-FS, reference [8].

5.5.2 R_CTL field

The R_CTL field contains two sub-fields, the routing bits and the information field. The routing bits identify the frame payload as either FC-4 related or FC-PH related. For all frames used to send an IU, the routing bits shall be set to ’0000’b to indicate that the payload contains FC-4 Device_Data.

When the routing bits are set to ’0000’b and the frame is used to send an IU, the information field contains an information category field which identifies the category of the information in the payload. (See FC-PH, reference [1] for a list of information categories and their corresponding values of the information field.) All frames in an SB-2 IU shall contain the same information category. The information category of an IU depends on the contents of the IU and the conditions under which the IU is sent. Table 1 summarizes the information categories which shall be used to send each IU.

Table 1 – Information categories of SB-2 IUs

SB-2 IU name	Information category	Content
Command IU	Unsolicited command	Command DIB or command/data DIB
Solicited data IU	Solicited data	Data DIB or Status DIB
Unsolicited data IU	Unsolicited data	Status DIB
Solicited control IU	Solicited control	Link control DIB or control DIB
Unsolicited control IU	Unsolicited control	Link control DIB or control DIB

5.5.3 D_ID and S_ID fields

Each channel and control unit is assigned a unique N_Port ID during the initialization procedure. See Clause 7. When sending an IU, the D_ID field shall be set to the N_Port ID of the recipient of the frame, and the S_ID field shall be set to the N_Port ID of the sender of the frame.

5.5.4 CS_CTL

See FC-FS, reference [8].

5.5.5 TYPE field

The TYPE field identifies the FC-4 protocol which defines the frame payload. This field shall be set to hex'1B' in all frames of an IU which are sent by a channel. This field shall be set to hex'1C' in all frames of an IU which are sent to a channel.

If a channel which does not provide control-unit emulation receives an ELP IU with the TYPE field set to hex'1B', it shall send a link-level reject (LRJ) IU in response; if the channel provides control-unit emulation, the emulated control unit may send an LPE IU with the TYPE field set to hex'1C' in response. See 6.4.9 and 11.2.8, and 6.4.2 for additional information. IUs other than the ELP IU, LRJ IU, and LACK IU, which have the TYPE field set to hex'1B', are discarded by the channel and no response is sent.

5.5.6 F_CTL field

The F_CTL field controls sequences and exchanges. These bits are defined in FC-PH, reference [1]. SB-2 has the following requirement on the relative offset present bit in the F_CTL field:

Relative Offset present: The relative offset present bit shall be set to one for all IUs of information category solicited data. For all other information categories, the relative offset present bit shall be set to zero. See 7.3.2.2, 7.4.2.2, and FC-PH, reference [1], for additional information on relative offset.

For the settings of all other bits in the F_CTL field, see FC-PH, reference [1].

5.5.7 SEQ_ID

The SEQ_ID field is defined by FC-PH, reference [1].

5.5.8 DF_CTL

The DF_CTL field shall be set to zero since optional headers are not used.

5.5.9 SEQ_CNT

The SEQ_CNT field is defined by FC-PH, reference [1].

NOTE When sending IUs, the use of continuously increasing sequence count is recommended. This increases the probability that errors caused by lost frames are detected. See FC-PH, reference [1] for additional information.

5.5.10 OX_ID

The OX_ID field is defined by FC-PH, reference [1].

5.5.11 RX_ID

The RX_ID field is defined by FC-PH, reference [1].

5.5.12 Parameter field

When the information category of an IU is solicited data, the relative offset present bit is set to one and the parameter field shall contain the offset from the base address into which the data in a frame is stored. The base address for the solicited data information category is zero and refers to the first byte of the IU. When the information category of an IU is not solicited data, the relative offset present bit is set to zero and the parameter field shall be ignored by the recipient. (See 5.5.6 for information on the relative offset present bit, and see FC-PH, reference [1] for additional information about relative offset.)

5.6 Other FC-PH mechanisms

Other required FC-PH mechanisms include selected basic and extended link services. See Clause 6 for additional information.

6 Link-Level functions

6.1 Link-level function overview

FC-PH link control, FC-PH basic link services, FC-PH extended link services, and SB-2 link control provide the means by which the elements interconnected by the physical paths are brought to and maintained in an operational state. (See Clause 5 for a description of those aspects of FC-PH link control specific to SB-2, and see FC-PH, reference [1] for general aspects of FC-PH link control, FC-PH basic link services, and FC-PH extended link services.) The functions provided by FC-PH extended link services, FC-PH basic link services, and SB-2 link control are collectively referred to as “link-level functions” in this document. Those link-level functions provided by FC-PH extended link services are referred to as “extended-link-service commands;” those link-level functions provided by FC-PH basic link services are referred to as “basic-link-service commands;” and those link-level functions provided by SB-2 link control are referred to as “SB-2 link-control functions.” The link-level functions required for SB-2 support are described in the following subclauses.

6.2 FC-PH Basic-link-services

This subclause describes the basic-link-service (BLS) commands required by SB-2. Only a brief overview of how these commands are used is provided. For additional details, see FC-PH, reference [1].

6.2.1 Abort sequence

When an exchange is to be aborted, the abort sequence (ABTS) basic-link-service shall be performed. For the conditions under which one or more exchanges are aborted, refer to Clause 11. The settings of fields in the FC-PH header of the ABTS BLS command are given in FC-PH, reference [1]. When the recipient of an ABTS BLS command accepts it, the recipient shall send a basic accept (BA_ACC) response indicating that the entire exchange is being aborted and a recovery qualifier is to be established by setting the Last_Sequence bit in the F_CTL field equal to one, the Low SEQ_CNT field equal to hex’0000’, and the High SEQ_CNT field to hex’FFFF’. The recovery qualifier causes all frames for all sequences of the exchange to be discarded until the recovery qualifier is reinstated. After receiving the BA_ACC response and after waiting for an R_A_TOV time-out period, the sender of the ABTS BLS command shall send a reinstate recovery qualifier extended-link-service command to allow the OX_ID and RX_ID combination to be reused. See 6.3.4 and FC-PH, reference [1].

6.3 FC-PH Extended link services

6.3.1 F_Port login

Explicit F_Port login shall be performed by means of the F_Port login (FLOGI) extended link service during the initialization process. See Clause 7 for a definition of other aspects of explicit F_Port login specific to SB-2, and see FC-PH, reference [1] for general aspects of the explicit FLOGI extended link service.

6.3.2 N_Port login

Explicit N_Port login shall be performed by means of the N_Port login (PLOGI) extended link service during the initialization process. When an N_Port has performed N_Port login with another N_Port, that N_Port is said to be logged in with the other N_Port.

When a PLOGI ELS command is received from an N_Port with which the recipient is currently logged in, the recipient shall perform implicit N_Port logout with the source N_Port and shall consider all open exchanges with the source N_Port to be abnormally terminated before accepting the PLOGI ELS com-

mand. (See FC-PH, reference [1] clause 24.3.13 and 10.5.7 for additional information on the abnormal termination of an exchange.)

If the N_Port of a control unit completes explicit N_Port login with the N_Port of a channel at a time when one or more logical paths are indicated as being established to the channel, initiative to send a test initialization IU to that channel shall be generated. See 6.4.7 for additional information. See Clause 7 for a definition of other aspects of explicit N_Port login specific to SB-2, and see FC-PH, reference [1] for general aspects of the explicit PLOGI extended link service.

6.3.3 N_Port logout

N_Port logout may be performed either explicitly or implicitly. For the conditions under which implicit N_Port logout occurs, see FC-PH, reference [1]. When communication with the N_Port of a control unit is no longer required and no logical paths exist for which the channel is providing control unit emulation, the N_Port of a channel shall initiate explicit N_Port logout by sending a LOGO ELS command and awaiting a response. In addition, the N_Port of a channel shall initiate explicit N_Port logout with the N_Port of a control unit if it receives an IU from the control unit and it is not configured to communicate with the control unit.

The N_Port of a control unit shall initiate explicit N_Port logout with the N_Port of a channel only if the control unit receives an IU from a channel with which the N_Port of the control unit is neither logged in nor in the process of logging in. When this occurs, the received IU shall be discarded, explicit N_Port logout shall be performed, and no further action is taken by the N_Port of the control unit.

When the N_Port of a control unit accepts a LOGO ELS command from the N_Port of a channel to which logical paths exist, it shall reset its internal indicators for the logical paths established with the channel, and perform the equivalent of a system reset for the affected logical paths. For a description of system reset, see 9.2.4. See FC-PH, reference [1] for additional information concerning the handling of open sequences and exchanges which exist at the time when explicit N_Port logout is to be performed, and see 11.2.5 for information concerning the actions of channels and control units upon implicit N_Port logout.

NOTE FC-PH, reference [1], requires that during the N_Port login procedure, other communication with the destination N_Port is neither initiated nor accepted. Therefore if a control unit has sent a PLOGI ELS to a channel and receives an IU from the channel before receiving a response to the PLOGI ELS, the received IU is discarded and no LOGO ELS is sent.

6.3.4 Reinstate recovery qualifier

The reinstate recovery qualifier (RRQ) extended link service shall be performed in order to allow reuse of the OX_ID and RX_ID of an exchange which was aborted. When the sender of an ABTS BLS command receives a BA_ACC response, it sends an RRQ ELS command after waiting an R_A_TOV time-out period. Other aspects of the Reinstate Recovery Qualifier extended link service are given in FC-PH, reference [1].

6.3.5 Registered state change notification

In a point-to-point configuration, the registered state change notification (RSCN) extended link service shall be performed in order to notify the directly-attached N_Port of an event which has affected the state of the sending channel or control unit, provided that the attached N_Port has previously registered to receive state-change notifications. (See 6.3.6.) In a fabric configuration, the RSCN ELS shall be performed in order to notify the fabric controller of an event which has affected the state of the sending N_Port. Additional aspects of the registered state change notification extended link service includ-

ing the format of the RSCN ELS command, the format of the response, and the conditions under which a fabric controller sends an RSCN ELS command are given in FC-FS, reference [8].

The N_Port of a channel or a control unit may optionally perform the RSCN extended link service when an event causes a change in existing logical paths or, for a control unit, in the ability to accept new logical paths. When the resources common to the N_Port of a channel are shared among channel images, the N_Port of that channel shall be capable of reporting a state change by performing the RSCN extended link service.

The N_Port of a channel or control unit shall perform the RSCN extended link service by sending the RSCN ELS command to the fabric controller if a Fabric is present, or to the N_Port of the attached control unit or channel if no Fabric is present. The address format byte (byte 1 of the Affected N_Port ID page of the RSCN ELS command) shall be set to zero to indicate that all three bytes of the N_Port ID are valid, and the affected N_Port ID shall be set to the N_Port ID of the sender. The sender of the RSCN ELS command shall not issue the command if it is aware that it has already signaled a condition which would cause the fabric controller, if present, to issue an RSCN ELS command on behalf of its N_Port ID.

Once the N_Port of a channel or control unit recognizes a condition for which an RSCN ELS command is sent, initiative to send an RSCN ELS command is generated. Subsequent state changes in the sender shall not create initiative to send another RSCN ELS command if the initiative to send the first RSCN ELS command has not been discharged. Once the initiative to send the first RSCN ELS command has been discharged, a subsequent state change shall create a new initiative.

Initiative to send an RSCN ELS command shall be discharged when an accept (ACC) link service reply is received in response to the RSCN ELS command.

When an RSCN ELS command is accepted, the recipient shall check each affected N_Port ID page. (See FC-FS, reference [8].) All three allowed values of the address format byte shall be supported. If an affected N_Port ID indicated in any affected N_Port ID page corresponds to an N_Port ID to which one or more logical paths are indicated as being established, initiative to send a test initialization IU to that N_Port may be generated. See 6.4.7 for additional information. If an affected N_Port ID corresponds to an N_Port ID to which no logical path is indicated as being established, initiative to send a test initialization IU shall not be generated and no further action is needed. However, at a channel, initiative to establish logical paths may be generated if its model-dependent configuration information indicates that a new control-unit image may have become available.

NOTE Sending the RSCN ELS command is not functionally equivalent to performing the FC-PH online to offline primitive sequence protocol. (See FC-PH, reference [1] and 11.2.4.) However, the RSCN ELS command may be used by a channel or control unit to report a state change that has affected logical paths. When multiple logical images share a common N_Port, RSCN is used in place of the online to offline primitive sequence protocol to report state changes that have not affected all of the logical images.

6.3.6 State-change registration

The state-change-registration (SCR) extended link service shall be performed in order to register to receive RSCN ELS commands. General aspects of the state change registration extended link service, including the format of the SCR ELS command and response, are given in FC-FS, reference [8]. If a Fabric is present, the D_ID field in the FC-PH header of the SCR ELS command shall be set to the N_Port ID of the fabric controller. If no Fabric is present, the D_ID field shall be set to the N_Port ID of the attached channel or control unit. The registration function field in the payload of the SCR ELS command shall be set to hex'03' to request notifications to be sent for events detected by both the fabric controller and the affected N_Ports. After the SCR ELS command is accepted, registration remains in effect until N_Port logout occurs with the recipient of the SCR ELS command. Although it is possible

to clear registration for state change notifications by setting the registration function field in the SCR ELS command to hex'FF', the sender should not set the registration function field to this value.

6.3.7 Request Node-Identification Data

6.3.7.1 Request node-identification data requests and responses

The request-node-identification-data (RNID) extended link service shall be performed in order to acquire the node identification data of the recipient. General aspects of the RNID extended link service, including the format of the RNID ELS command and response, are given in FC-FS, reference [8]. When either the RNID ELS command or the RNID ACC ELS reply is sent, the node-identification data format field shall be set to hex'18'. In the RNID ACC ELS reply, the 32 byte node descriptor described below shall be returned in the specific node-identification data field. When the RNID ACC ELS is received, the common node-identification data, if present, may be ignored.

6.3.7.2 Specific node-ndentification data

For SB-2, the specific node-identification data is referred to as the node descriptor. A node descriptor (ND) is a 32-byte field that describes a node. The ND consists of a 1 byte flags field, a 3 byte node parameters field, and a 28 byte node-ID.

The node-ID is composed of the two parts listed below.

a) SDC ID: The first 26 bytes of the node-ID identify the Self-Describing Component (SDC) containing the interface that determines the node. The SDC ID shall correspond to the information provided on a serial-number plate attached to the external surface of the structure containing the SDC.

b) Interface ID (Tag): The last two bytes of the node-ID shall contain an interface identifier (ID) that uniquely identifies the physical location of the associated SDC interface.

Node-IDs with the same SDC ID shall not use the same interface ID.

Collectively, the 25 bytes of information contained in word 1, bytes 0 and 1 of word 2, bytes 1-3 of word 3, and words 4-7 of the node descriptor shall provide a vendor-specific node identifier.

A node descriptor shall have the format given in Figure 10. (See 3.3.3 for a description of the bit numbering convention used.)

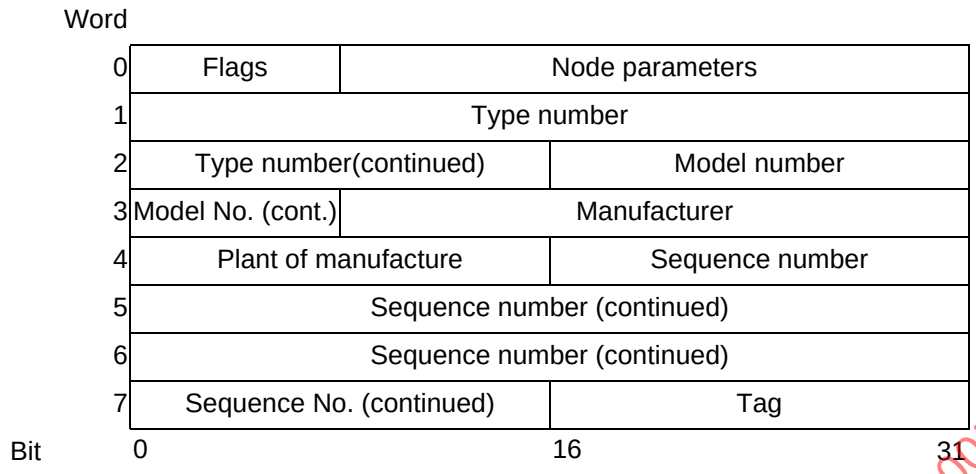

Figure 10 – Contents of the node descriptor

Flags: Byte 0 of word 0 describes the manner in which selected fields of the node descriptor are to be interpreted. The meaning of bits 0-7 is as follows:

Bits *Meaning*

0-2 Node-ID validity. Bits 0-2 contain a code that describes the validity of bits 3-7 of the flags field, the node parameters field, and the node-ID contained in words 1-7. The codes and their meanings are as follows:

Value *Meaning*

0 Bits 3-7 of the flags field, the node parameters field, and the node-ID are valid.

1 Bits 3-7 of the flags field, the node parameters field, and the node-ID are valid, however, they may not be current. This value shall be used when the SDC has obtained the requested node ID but subsequently has observed some event (such as the loss of signal on a link) which may have resulted in a configuration change. The SDC has been unable to obtain the node ID again.

2 Node-ID is not valid. The SDC is unable to obtain the requested node-ID. Except for the node-ID-validity field, the contents of the node descriptor shall have no meaning.

When the N_Port of a channel or control unit receives an RNID ELS command, the channel or control unit shall determine the node-ID validity and set the appropriate node-ID validity code in the flag field of the ND. If, while obtaining its node-ID, an error is detected or it is determined for some other reason that the node-descriptor information is suspect (that is, not valid), then the node-ID validity code shall be set to the value 2 in the flag field of the node descriptor being sent in the accept response.

In the accept response to the RNID ELS command, a node-ID validity code of either 0 or 2 in the flag field of the node descriptor shall be indicated. Since a node descriptor is either *valid* or *not valid*, a node-ID validity code of 1 shall never be sent in the accept response to the RNID command.

3 Node type. When zero, bit 3 shall specify that the node described by this node descriptor is a device-type node; when one, this bit shall specify that the node described by this node descriptor is a central-processor-complex-type (CPC-type) node.

4-7 Reserved.

Node Parameters: Bytes 1-3 of word 0 shall contain additional information about the node.

When bit 3 of the flag field is zero, indicating that this is a device-type node, the contents of bytes 1-3 of word 0 shall be as follows:

Byte *Description*

1 Bits 0-2 of byte 1 contain a code that shall specify the interface protocol type of the interface identified by the node descriptor. The codes and their meanings are as follows:

Value	*Meaning*
0	Reserved.
1	FC-SB-2
2-6	Reserved
7	Vendor-specific

Bits 3-7 of byte 1 shall be reserved.

2 Class. Byte 2 of word 0 shall contain a code that specifies the class to which the device belongs. The codes and their meanings are as follows:

Value	*Meaning*
0	Unspecified class
1	Direct access storage (DASD)
2	Magnetic tape
3	Unit record (input)
4	Unit record (output)
5	Printer
6	Communications controller
7	Terminal (full screen)
8	Terminal (line mode)
9	Reserved
10	Switch
11-255	Reserved

3 Byte 3 shall contain zero, except in the following case:

When code 10 is specified in the class field, indicating that this is a switch, byte 3 shall contain the area field (bits 15 - 8) of the Port address identifier of the associated switch interface.

When bit 3 of the flag field is one, indicating that this is a CPC-type node, the contents of bytes 1-3 of word 0 shall be as follows:

Byte *Description*

1 *Type*. When the class field contains a value other than 1, byte 1 of word 0 shall be reserved and set to zeros.

When the class field contains a value of 1, byte 1 of word 0 shall be defined as follows:

– Bits 0-2 contain a code that specifies the interface protocol type of the interface identified by the node descriptor. The codes and their meanings shall be as follows:

Value	*Meaning*
0	Reserved.
1	FC-SB-2
2-6	Reserved
7	Vendor-specific

– Bits 3-7 of byte 1 shall be reserved.

2 *Class*. Byte 2 of word 0 shall contain a code that specifies the class to which the interface belongs. The codes and their meanings shall be as follows:

Value	*Meaning*
0	Unspecified class
1	Channel path, not CTC capable
2-6	Reserved
7	FC-SB-2 channel-to-channel adapter (FC-SB-2 CTCA)
8	Emulated control unit support only
9-255	Reserved

3 *Identification*. Byte 3 of word 0 shall contain the CHPID of the channel path that contains the specified interface.

The contents of the following five fields shall correspond to the information provided on a serial-number plate attached to the external surface of the SDC. The ASCII character code used in these fields shall be ASCII/Latin 1.

Type Number: Word 1 and bytes 0-1 of word 2 shall contain the six-character (0-9) ASCII type number of the SDC. The type number shall be right justified with leading ASCII zeros if necessary.

Model Number: Bytes 2-3 of word 2 and byte 0 of word 3 shall contain, if applicable, the three-character (0-9 or uppercase A-Z) ASCII model number of the SDC. The model number shall be right justified with leading ASCII zeros if necessary.

Manufacturer: Bytes 1-3 of word 3 shall contain a three-character (0-9 or uppercase A-Z) ASCII code that identifies the manufacturer of the SDC.

Plant of Manufacture: Bytes 0-1 of word 4 shall contain a two-character (0-9 or uppercase A-Z) ASCII plant code that identifies the plant of manufacture for the SDC.

Sequence Number: Bytes 2-3 of word 4, words 5-6, and bytes 0-1 of word 7 shall contain the 12-character (0-9 or uppercase A-Z) ASCII sequence number of the SDC. The sequence number shall be right justified with leading ASCII zeros if necessary.

A serial number shall consist of the concatenation of the plant-of-manufacture designation with the sequence-number designation.

Tag: Bytes 2-3 of word 7 shall contain the physical identifier for the SDC interface that is identified by the preceding 26 bytes of the node descriptor.

6.3.8 Registered Link-Incident Record

6.3.9 Registered Link-Incident Record requests and responses

The registered link-incident record (RLIR) extended link service shall provide the means to send a link-incident record to the N_Port of a channel. General aspects of the RLIR extended link service, including the format of the RLIR ELS command and ACC reply, are given in FC-FS, reference [8]. When the RLIR ELS command is sent, the link-incident record format field shall be set to hex'18'. When the RLIR ELS command is received, the link-incident descriptor may be ignored and the common link-incident record, if present, may be ignored. The specific link-incident record field shall contain a 100 byte SB-2 specific link-incident record of the format described below.

Control units shall use the link-incident reporting procedure to report link-incidents.

6.3.9.1 SB-2 specific Link-Incident Record

The SB-2 specific link-incident record is shown in Table 2.

Table 2 – SB-2 specific Link-Incident Record

Byte	Contents		
0-3	IQ	IC	Reserved
4-35	Incident node descriptor		
36-67	Attached node descriptor		
68-99	Incident specific information		

Incident Qualifier (IQ): Byte 0 shall describe the manner in which the contents of the link-incident record shall be interpreted. The meaning of bits 0-7 is as follows:

Bits *Meaning*

0 Reserved

1 Reserved

2 Switch: When one, bit 2 shall indicate that the incident node, identified by the incident-node descriptor, is a switch node. When zero, bit 2 shall indicate that the incident node is not a switch node.

3 Reserved

4-5 Bits 4 and 5 constitute a two-bit code which shall identify the reporting class for the link-incident. The codes and their meanings are as follows:

Value	*Meaning*
0	Informational report: All link-incidents reported with incident-code bit 0 set to one shall use a reporting class value of zero.
1	Link degraded but operational: Link-incidents reported with incident-code bit 0 set to zero shall use reporting class 1 if the link associated with the incident node is not in a link-failure or offline state as a result of the event which generated the link-incident record.
2	Link not operational: Link-incidents reported with incident-code bit 0 set to zero shall use reporting class 2 if the link associated with the incident node is in a Link-Failure or Offline state as a result of the event which generated the link-incident record.
3	Reserved

6 Subassembly type: When one, bit 6 shall specify that the type of subassembly used for the node that is the subject of this link-incident record is laser. When zero, bit 6 shall specify that the type of subassembly used for the node that is the subject of this link-incident record is not laser.

7 FRU identification: When one, bit 7 shall specify that the incident-specific-information field is in a format that provides field-replaceable-unit (FRU) identification. When zero, bit 7 shall specify that the incident-specific-information field is vendor-specific.

Incident Code (IC): Byte 1 shall contain the incident code which describes the incident that was observed by the incident node.

Bits	*Meaning*
0	Bit 0 of the incident code shall indicate whether the link-incident record is a primary or secondary report of the link-incident. When bit 0 is set to zero, the link-incident record shall be a primary report. When bit 0 is set to one, the link-incident record shall be a secondary report.
1-7	Bits 1-7 of the incident code shall contain a value that specifies the type of incident which was observed. The values that may be specified shall be as follows.

Value	*Meaning*
0	Reserved
1	Implicit Incident: A condition which has been caused by an event known to have occurred within the incident node has been recognized by the incident node. The condition affects the attached link in such a way that it may cause a link-incident to be recognized by the attached node.
2	Bit-error-rate Threshold Exceeded: The number of code violation errors recognized by the incident node has exceeded a threshold. (See FC-FS, reference [8].)

3 FC-PH Link Failure - Loss of Signal or Synchronization: A loss of synchronization condition has been recognized by the incident node, and it persisted for more than the R_T_TOV time-out period. A loss of signal condition has been recognized by the incident node. (See FC-PH, reference [1] clause 16.4.2.)

4 FC-PH Link Failure - Not-Operational Primitive Sequence (NOS) Recognized: The NOS primitive sequence has been recognized by the incident node. (See FC-PH, reference [1] clause 16.5.3.2.)

5 FC-PH Link Failure - Primitive Sequence Time-out: The incident node has recognized either a link reset protocol time-out (See FC-PH, reference [1] clause 16.5.2.) or a time-out when timing for the appropriate response while in the NOS Receive state and after the NOS is no longer recognized. (See FC-PH, reference [1] clause 16.5.3.2.)

6 FC-PH Link Failure - Invalid Primitive Sequence for Port State: Either a link reset (LR) or link reset response (LRR) primitive sequence was recognized by the incident node while in the wait-for-OLS state. (See FC-PH, reference [1] clause 16.5.4.3.)

A link-incident record shall be generated and reported for an FC-PH link-failure condition only if the FC-PH link-failure condition persists for longer than SB_TOV.

7-127 Reserved

Incident-Node Descriptor: Bytes 4-35 shall contain the node descriptor of the incident node. The contents of a node descriptor are described in 6.3.7.2.

Attached-Node Descriptor: Bytes 36-67 shall contain the node descriptor of the node attached to the incident node at the time the link-incident was detected. The contents of a node descriptor are described in 6.3.7.2.

Incident-Specific Information: When bit 7 of the incident-qualifier field is set to zero, bytes 68-99 shall contain node-dependent incident information, which may provide additional information related to the incident.

When bit 7 of the incident-qualifier field is set to one, bytes 68-99 shall contain field-replaceable-unit (FRU) identification information.

When the incident-specific-information field contains FRU identification information, the format of the incident-specific information is illustrated in Table 3:

Table 3 – Incident-Specific Information

Bytes	Contents	
68 - 69	FRU flags	reserved
70 - 81	First FRU callout (12 Bytes)	
82 - 93	Second FRU callout model-dependent information (12 Bytes)	
94 - 99	Reserved	

Byte 68 shall contain the FRU-flags field. The meaning of bits 0-7 shall be as follows:

Bits *Meaning*

0 Reserved.

1 Format bit; FRU-callout-field format:

Value *Meaning*

0 FRU-part-number format

1 FRU-code format

2-5 Reserved.

6-7 Validity code for FRU-callout fields:

Value *Meaning*

00 Reserved.

01 First-FRU-callout field valid; Second-FRU-callout field shall contain 12 bytes of model-dependent data.

10 First-FRU-callout and second-FRU-callout fields valid.

11 Reserved.

Byte 69 shall be reserved and set to zero.

Bytes 70-81 shall contain the first-FRU-callout identification information.

Bytes 82-93 shall contain either the second-FRU-callout identification information or 12 bytes of model-dependent information, depending on the value of bits 6-7 of the FRU-flags field.

Bytes 94-99 shall be reserved and set to zeros.

The format of a valid FRU-callout field shall depend on the value of bit 1 (format bit) of the FRU-flags field within the same link-incident record.

When the format bit is set to zero, the FRU-callout field shall be in ASCII, right justified, with either leading blanks (hex'20') or leading ASCII zeros (hex'30').

When the format bit is set to one, the FRU-callout field shall be in hexadecimal, right justified, with leading zeros (hex'00').

6.3.10 Link-Incident-Record Registration

The link-incident-record registration (LIRR) extended link service shall provide the means for the N_Port of a channel to register to receive link-incident records from a control unit. General aspects of the LIRR extended link service including the format of the LIRR ELS command and ACC response are given in FC-FS, reference [8].

The N_Port of a channel shall send an LIRR command during the initialization procedure. (See Clause 7.) The registration function field shall be set to hex'01' to indicate the "set registration-conditionally receive" function, and the link-incident-record registration type field shall be set to hex'18'. Although it is possible to clear registration for link-incident records by setting the registration function field to hex'FF', the sender shall never set the registration function field to this value.

6.4 SB-2 Link-control functions

6.4.1 SB-2 Link-control function overview

SB-2 link-control functions provide the means by which the logical paths between a channel and control-unit are established and maintained. SB-2 link-control functions also provide information about conditions on the physical and logical paths that affect the transmission or reception of information units (IUs). These functions are performed by means of control IUs containing a link-control DIB (or, link-control IUs). (See 8.12.) For brevity, a link-control IU containing a link-control DIB with a link header specifying one of the link-control functions, is referred to by the name of the link-control function specified in the link header. For example, when the link-control function is a request to establish a logical path, the link-control IU is referred to as an establish logical path IU. (See Table 4 for a list of the link-control functions.)

SB-2 link-control functions are performed primarily during initialization or when certain error conditions occur on a link.

SB-2 link-control requests shall be part of a request-response pair, followed by a link-level acknowledgment (LACK) response. The link-control IU containing the request (or, the link-control-request IU) shall be sent, opening an exchange, as an unsolicited control information category. The exchange containing the link-control-request IU shall be left open. The link-control-response IU shall be sent as a solicited control information category in a new exchange. The exchange containing the link-control-response IU shall be closed. When the sender of the link-control-request IU receives the link-control-response IU, it shall send a link-level-acknowledgment IU as a solicited control information category on the same exchange used to send the link-control-request IU. The exchange containing the link-lev-

el-acknowledgment IU shall be closed. An example of an SB-2 link-control function is a link-control-request IU sent by the channel to request the establishment of a logical path, a link-control-response IU sent by the control unit indicating whether or not the logical path was established, and a link-level-acknowledgment IU sent by the channel to close the outbound exchange.

Link-control-request IUs shall always be sent on a new exchange, that is, they shall never be sent on an existing exchange. If a link-control-request IU is received on an existing exchange, an SB-2 link-level protocol error (or, for brevity, a link-level protocol error) shall be detected.

If a channel or control unit receives an IU when any of the following apply, then the IU shall be discarded and a link-level protocol error shall be detected:

– the received IU is a link-control-response IU and no link-control-request IU was sent, or a link-control-request IU was sent and the link-control-response IU received is not an allowed response,

– the received IU is a link-control IU specifying a link-control function that is not recognized,

– the received IU has an information category other than an information category allowed for the IU,

– the received IU is a link-control IU with a link-payload byte-count field set to a value different from that described in Table 4.

Table 4 summarizes the SB-2 link-control functions and the contents of fields within the IU. For a definition of the format of fields in link-control IUs, see Figure 12 and 8.12.

Table 4 – Summary of link-control request and response IUs

IU	Sent by channel	Sent by control unit	Expected reply or reply to	SB-2 header		Link header		Link payload
				Channel image ID	Control-unit image ID	Link-control field	Link-control info. field	
Link-control request IUs								
ELP	yes	no	LPE,LPR	Channel image ID	Control-unit image ID	0100 0001	Optional features	None
RLP	yes	no	LPR	Channel image ID	Control-unit image ID	0100 1001	n/a	None
TIN	yes	yes	TIR	Note 1	Note 1	0000 1001	n/a	None
Link-control response IUs								
LPE	no	yes	ELP	Channel image ID	Control-unit image ID	0101 0001	Optional Features	None
LPR	no	yes	ELP,RLP	Channel image ID	Control-Unit image ID	0101 1001	Reason	None
TIR	yes	yes	TIN	Note 2	Note 2	0000 0001	n/a	Initialization state (32 Bytes)
LRJ	yes	yes	Initiation IU	Note 3	Note 3	0001 0001	Reason	None
LBY	yes	yes	Initiation IU	Note 3	Note 3	0010 0001	n/a	None
LACK	yes	yes	Note 4	0	0	0110 0001	n/a	None

Note 1) See 6.4.7.
Note 2) See 6.4.8.
Note 3) This field is set to the same value as the corresponding field in the discarded initiation IU.
Note 4) See 6.4.6.

The functions performed by the SB-2 link-control IUs in the above table are described below.

6.4.2 Establish logical path

The establish logical path (ELP) function shall be sent from a channel image to a control-unit image to indicate the optional features supported by the channel and to request the establishment of a logical path. The number of CTC connections for which the channel is providing emulated control unit functionality may also be indicated. A logical path, when established, shall identify the channel and control unit images and shall specify the optional features which are used when IUs are sent between the channel and the control unit. Optional features are optional methods of operation which may be used in place of the default method of operation.

A channel shall perform the ELP function by sending an ELP IU to the control unit. The link-control DIB shall contain a link header with a link-control field specifying the ELP function, a link-control-information field specifying optional features, and a CTC counter field that may specify the number of CTC connections for which the channel is providing emulated control unit functionality. (See 8.12.2.) The

link-control DIB shall not contain a link payload. The normal response to an ELP IU is the logical path established IU.

The channel image for the path to be established shall be identified by the combination of the channel N_Port ID and channel image ID.

The control-unit image shall be identified by the combination of the control-unit N_Port ID and control-unit image ID.

All control units shall be able to accept any value in the channel image ID field. The control unit may still restrict the quantity of logical paths permitted based on resources and system requirements.

Although a control unit that does not allow multiple control-unit images may restrict the number of logical paths it accepts to one, that one logical path may have any value of channel logical address.

The optional features which the channel supports shall be specified in the link-control-information field. See 8.12.2.3.

Each bit in the link-control-information field corresponds to a different optional feature. The bit for an optional feature shall be set to one if that optional feature is supported; otherwise, the bit shall be set to zero. Bits for which no optional feature is defined shall be set to zero by the channel and ignored by the control unit.

With the exception of the CTC connection feature, the channel shall indicate the same optional features in all ELP IUs which it sends.

Optional features for the following bits in the link-control information field have been defined:

Bits *Meaning*

0 Enhanced CRC Generation: When bit zero of the link-control information field is set to zero in the ELP IU, optional CRC generation initialization is not supported. When bit zero of the link-control information field is set to one in the ELP IU, optional CRC generation initialization is supported. See 8.6.5 for additional information.

15 CTC Connection: When bit 15 of the link-control information field is set to zero in the ELP IU, the channel does not support control-unit emulation for a CTC connection on the indicated logical path. When bit 15 of the link-control information field is set to one in the ELP IU, the channel does support control-unit emulation for a CTC connection on the indicated logical path. When control-unit emulation for a CTC connection is supported, the value specified in the CTC counter field is valid. (See 8.12.2.4.)

When a control unit which is not an emulated control unit receives the ELP IU, it shall compare the optional features of the channel with its own supported optional features. The optional features to be used shall be those features which both the channel and the control unit support.

When an emulated control unit receives an ELP IU, the optional features to be used are those optional features which both channels support with the exception of the CTC connection feature. A channel supporting CTC connection may use the CTC connection optional feature and respond with an LPE IU regardless of whether the other channel supports CTC connection. If the sending channel has indicated support for CTC connection by setting the CTC connection bit to one in the ELP IU, the CTC connection bit in the LPE IU is set to one; otherwise the CTC connection bit is set to zero.

The requested logical path shall be established provided that the optional features for the requested logical path do not affect the optional features associated with other previously established logical paths with the same channel. If the request to establish a logical path would affect the optional features associated with other previously established logical paths with the same channel, the requested logical path shall not be established, and an LPR IU shall be sent in response to the ELP IU with an appropriate reason code. (See 6.4.5.)

When a logical path is established, the optional features which both the channel and the control unit support shall be saved and associated with the logical paths for that channel in such a way that all subsequent IUs for which those features apply are transferred on that logical path using those optional features.

A channel shall attempt to establish logical paths to the control-unit images that are described in its configuration definition. This shall be done when a channel image is initialized, when configuration changes are made, or when the channel receives an indication that the logical path no longer exists. The generation of a configuration definition is beyond the scope of SB-2.

Each logical path shall be established with a separate exchange of IUs, an ELP request IU and an LPE response IU. Failure to establish a logical path shall not affect other existing logical paths or the ability to establish other logical paths.

If an ELP IU is received with a request for a logical path that is already established, the IU shall be accepted, provided that no errors are detected. If the establishment of the logical path would not affect optional features associated with other previously established logical paths with the same channel, a system reset shall be performed with respect to that logical path and a logical path established response shall be sent. The logical path established response does not indicate initiation or any degree of progress made for the associated system reset.

The channel shall not consider the logical path to be established until it receives an error-free logical path established IU. The control unit shall not consider the logical path to be established until the necessary action is taken at the control unit to establish the logical path, and the sending of an error-free logical path established IU is completed. If an error is detected when an ELP IU is received, the appropriate response, if any, shall be made, and the logical path shall not be established. If the control-unit image is not able to perform the ELP function for reasons other than an error condition or a link-busy condition, a logical path removed IU containing the appropriate logical path removed reason code shall be sent in response. If a logical path removed IU is received in response to an ELP IU, the logical path shall be considered not established. If a logical path removed IU response is received, the ELP IU may be retried until the logical path is established. The number of retries is model dependent. In the case of logical path removed, whether or not the ELP is retried shall depend on the reason code. (See 6.4.5.)

If a channel receives an ELP IU with a TYPE field of hex'1B', a link-level-reject (LRJ) IU with a reject-reason code of protocol error shall be sent in response if the conditions for sending an LRJ IU are satisfied. See 6.4.9 for additional information. Otherwise, when CTC connection is supported, a Logical Path Established (LPE) IU is sent to complete the establishment of the logical path.

In the event that there is an error in the response to a channel request to establish a logical path, the channel shall not assume that the requested action has or has not taken place. If no valid response is received by the channel to the ELP request, the channel may retry the request. The number of retries is model dependent.

6.4.3 Remove logical path

The remove logical path (RLP) function shall request the control unit to remove a logical path. A channel shall perform the RLP function by sending an RLP IU to the control unit. The link-control DIB shall contain a link header with a link-control field specifying the RLP function and no link payload.

The logical path to be removed shall be specified by the combination of the channel N_Port ID, the channel image ID, the control-unit N_Port ID, and the control-unit image ID.

An RLP IU shall be sent when a change in the channel-path configuration requires a control unit to be either physically or logically removed.

The removal of a logical path shall cause the control-unit image and its associated devices to be logically removed from the channel path. When an RLP IU is received and accepted, the logical path shall be removed, and the equivalent of a system reset shall be performed only for the affected logical path; that is, only the control-unit image associated with the logical path shall be affected, and only those operations and allegiances within the control-unit image for this logical path shall be reset. (See 9.2.4 for information regarding system reset.) The logical path removed IU shall be the normal response. Other logical paths associated with the same channel or different channels and the allegiances maintained to them shall be unaffected. After a logical path is removed, IUs for device-level functions shall not be sent or received using that logical path.

An RLP IU shall be received over the same physical path over which the logical path was established. If an RLP IU is received for a logical path that does not exist, the RLP shall be accepted, provided that no errors are detected, and the logical path removed response shall be sent.

If an error is detected when an RLP IU is received, the IU shall be discarded, the specified logical path is not removed, and the appropriate response, if any, for the error recognized shall be sent. The channel shall not consider the logical path removed until it receives the LPR IU and no errors are detected. The control unit shall not consider the logical path removed until the logical path removed IU is sent.

Unless the channel receives a valid response to an RLP, the channel shall not assume that the requested action has or has not taken place. If an invalid response is received by the channel to the remove-logical-path request, the channel shall retry the request. The number of retries is model dependent.

6.4.4 Logical path established

The logical path established (LPE) function shall confirm the successful completion of an ELP request and the establishment of the logical path, and shall indicate the optional features to be used on all IUs sent between the channel and the control unit. An optional feature shall be used only if the channel has indicated support for that optional feature, the control unit supports the optional feature, and use of the feature does not affect the optional features associated with other previously established logical paths with the same channel. A control unit shall perform the LPE function by sending an LPE IU to the channel. The link-control DIB shall contain a link header with a link-control field specifying the LPE function and a link-control-information field specifying optional features. The link-control DIB shall not contain a link payload.

The logical path which was established shall be identified by the combination of the control-unit N_Port ID, control-unit image ID, the channel N_Port ID, and the channel image ID in the LPE IU. The optional features to be used shall be specified in the link-control-information field. See 8.12.2.3.

Each bit in the link-control-information field shall correspond to a different optional feature. The bit for an optional feature shall be set to one if that optional feature is to be used; otherwise, the bit shall be

set to zero. See 6.4.2 for information concerning the CTC connection optional feature. Bits for which no optional feature is defined shall be set to zero by the control unit and ignored by the channel.

The control unit shall indicate the same optional features in all LPE IUs sent on a particular channel path.

See 6.4.2 for a definition of the bits in the link-control information field for which optional features have been defined.

When a channel receives an LPE response to the ELP IU, the indicator for the requested logical path shall be established provided that the optional features for the requested logical path do not affect the optional features associated with other previously established logical paths with the same control unit. When the logical path is established, the optional features to be used shall be saved and associated with the logical paths for that control unit in such a way that all subsequent IUs for which those features apply are transferred on that logical path using those optional features. If establishment of a logical path would affect the optional features associated with other previously established logical paths, an RLP IU shall be sent to remove the logical path.

When an ELP IU is accepted, the LPE IU shall be the normal response. A logical path shall be considered not established by the recipient of an ELP IU until it has sent the LPE IU and shall be considered not established by the sender of an ELP IU until the LPE IU is received.

6.4.5 Logical path removed

The logical path removed (LPR) function shall confirm the successful completion of an RLP request and the removal of a logical path. A control unit shall perform the LPR function by sending an LPR IU to the channel. The link-control DIB shall contain a link header with a link-control field specifying the LPR function and a link-control-information field specifying a reason code. The LPR IU shall have no link payload.

The combination of the control-unit N_Port ID, control-unit image ID, the channel N_Port ID, and the channel image ID in the LPR IU shall identify the logical path that was removed.

An LPR IU shall be the normal response to an RLP request. A logical path shall be considered not removed by the recipient of an RLP frame until it has sent the LPR IU in response and shall be considered not removed by the sender of the RLP IU until the LPR IU is received.

An LPR IU may also be sent in response to an ELP IU when no error or link-busy conditions are detected but the requested logical path is not established.

The first byte of the link-control-information field in the link header is defined as follows:

Bits *Meaning*

0-3 Bits 0-3 of the link-control-information field in the link header shall be reserved for future use; these bits shall be set to zeros by the sender of the LPR IU and shall be ignored by the recipient of the IU.

4-7 Bits 4-7 of the link-control-information field shall contain a reason code which indicates why the logical path was removed. The values (in binary) for the following reason codes shall have these meanings:

Value	*Meaning*
0000	This LPR IU is a response to RLP.
0001	The supported optional features received would affect those in use on already established logical paths with the same channel. The ELP should not be retried for this condition.
0010	The control-unit image has no resources available for establishing new logical paths on this physical path. The channel may retry the ELP a model-dependent number of times.
0011	Device-level initialization is not complete; the control unit is not ready to perform device-level operations. The channel should retry the ELP a model-dependent number of times.
0100	A control-unit image corresponding to the control-unit image ID field in the SB-2 header of the ELP IU does not exist. The channel should not retry the ELP for this condition.
0101-1111	Reserved

A link-level protocol error shall be detected if an LPR IU containing a reserved reason-code value is received.

6.4.6 Link-level acknowledgment

The LACK function shall be used to close an exchange used to initiate a link-control request, and exchanges which can not be closed with either a link-control response or a device-level-control function. Examples of exchanges which may be ended with the LACK IU are listed below:

– an exchange left open as a result of a link-control request,

– an exchange left open as a result of a cancel, system reset, or request status sent in an initiation IU as information category unsolicited control which received a DACK IU as an allowed response,

– an exchange left open as a result of a selective reset device-level control function which received a DACK IU as an allowed response,

– an exchange left open as a result of an initiation IU receiving an LRJ IU or LBY IU as an allowed response, or

– an exchange left open as a result of a command or a device-level-control function which received an address exception IU as an allowed response.

The LACK function shall be performed by sending the LACK IU. The link-control DIB shall contain a link header with a link-control field specifying the LACK function and no link payload.

The channel and control-unit image IDs of the LACK IU shall be set to zero by the sender and ignored by the recipient.

6.4.7 Test initialization

The test-initialization (TIN) function shall provide a method for determining which logical paths are considered established for a channel or control-unit. The TIN function shall be performed by sending a TIN IU using Class 2 service. The link-control DIB shall contain a link header with a link-control field specifying the TIN function and no link payload.

When a channel sends the TIN IU, the logical paths to be tested for establishment shall be specified by the channel N_Port ID and channel image ID. The control-unit image ID field shall be set to zero by the channel and ignored by the control unit.

When a control unit sends the TIN IU, the logical paths to be tested for establishment shall be specified by the control-unit N_Port ID and control-unit image ID. The channel image ID field shall be set to zero by the control unit and is ignored by the channel.

A channel or control unit may send the TIN IU at any time; however, when initiative to perform the test-initialization function is generated, the TIN IU shall be sent. The occurrence of any of the following events at a channel or control unit shall create the initiative to send a TIN IU as defined below.

– If an RSCN is accepted by the N_Port of a channel or a control unit and one or more of the affected N_Ports is the N_Port of a channel or control unit to which one or more logical paths are indicated as being established, initiative to send a TIN IU to each channel or control unit to which a logical path is indicated as being established shall be generated.

– If the N_Port of a channel receives the LOGO ELS request from the N_Port of a control unit at a time when internal indicators at the channel indicate that one or more logical paths exist with the control unit, initiative to send a TIN IU to the control unit shall be generated.

– If the N_Port of a control unit receives a PLOGI ELS request from the N_Port of a channel at a time when one or more logical paths are indicated as being established to the channel, initiative to send a TIN IU to the channel shall be generated.

– If a control unit exits the FC-PH link failure state at a time when one or more logical paths are indicated as being established to one or more channels, initiative to send a TIN IU to each channel to which a logical path is indicated as being established shall be generated.

Once initiative to send a TIN IU to either a channel or control unit is generated, subsequent occurrences of any of the above events shall not create initiative to send another TIN IU to the same channel or control unit if initiative to send the first TIN IU has not been discharged. For a channel, initiative to send a TIN IU to a control unit shall be discharged after the TIN IU has been sent and a TIR IU has been received in response or an event occurs which requires the removal of all logical paths to the control unit. For a control unit, initiative to send a TIN IU to a channel shall be discharged if one of the following occurs:

– a TIN IU has been sent to the channel and a TIR IU is received in response,

– a logical path time-out condition for the channel has been recognized by any control-unit image, and the control unit has attempted to send a TIN IU to that channel after recognizing the time-out condition (see 10.2.4 for information on the recognition of a logical path time-out condition), or

– an event occurs which requires the removal of all existing logical paths to the channel.

Upon accepting a TIN IU, the recipient shall check whether it has logical paths with the source of the TIN IU. The normal response is a TIR IU.

6.4.8 Test initialization result

The test-initialization-result (TIR) function shall confirm the successful completion of the TIN function and shall indicate whether the logical paths are considered established for the channel or control unit performing the TIR function.

The TIR function shall be performed by sending a TIR IU. A TIR IU shall be sent only as a response to a TIN IU that is accepted. The link-control DIB shall contain a link header with a link-control field specifying the TIR function and a 32 byte logical path field in the link payload. (See Table 5 for the format of the logical path field.)

If a channel sends the TIR IU, the channel image ID field shall be set to zero by the channel and ignored by the control unit. The control-unit image ID field shall be the same as the control-unit image ID field received in the TIN IU. If a control unit sends the TIR IU, the channel image ID field in the SB-2 header shall be the same as the channel image ID field received in the TIN IU, and the control-unit image ID field shall be set to zero by the control unit and ignored by the channel.

The 32 byte link payload for the TIR IU has the format given in Table 5.

Table 5 – Logical path field

Word	Contents
0	Logical paths 0 - 31
1	Logical paths 32 - 63
2	Logical paths 64 - 95
3	Logical paths 96 - 127
4	Logical paths 128 - 159
5	Logical paths 160 - 191
6	Logical paths 192 - 223
7	Logical paths 224 - 255

The logical-path field shall indicate whether each of the 256 possible logical paths are considered established. There is a bit in the logical-path field for each possible logical path that may be established with the source of the TIN frame. Each bit shall correspond to an image ID. Starting with bit 0, word 0, logical addresses 0 through 255 are assigned in ascending order. The bit for a logical address corresponding to a logical path shall be set to one if that logical path is established and shall be set to zero otherwise. (See 3.3.3 for a definition of the SB-2 bit numbering convention used in Table 5.)

When the TIR IU is received in response to a TIN IU and one or more logical paths are indicated as not established when they were previously considered to be established, a test-initialization-result error shall be recognized. If one or more logical paths are indicated as being established when they were previously considered not established, a test-initialization-result error shall be recognized. For recov-

ery from a test initialization result error, see 11.2.9. If a test-initialization-result error is not recognized, no action shall be taken.

6.4.9 Link-level reject

The link-level-reject (LRJ) function shall indicate that the initiation IU received was not accepted by a channel or control unit and the IU was discarded because of an error condition. The LRJ function shall be performed by sending an LRJ IU.

The channel and control-unit image ID fields shall be set to the corresponding values in the IU for which the reject is being sent. If the IU for which the reject is being sent contains other than a link-control DIB and the AS bit is set to one, then the AS bit shall be set to one in the LRJ IU and the device address is provided. If the AS bit is set to zero in the IU being rejected, the AS bit shall be set to zero in the LRJ IU.

The link-control DIB shall contain a link header with a link-control field specifying the LRJ function and a link-control information field specifying the reject-reason code. The DIB shall not contain a link payload.

A channel or control unit shall send an LRJ IU in response to an IU when a logical path not established error condition is detected and all of the following conditions are satisfied:

– the IU is an initiation IU,

– the IU contains no FC-PH errors,

– the IU contains no exchange errors,

– the IU is either an unsolicited control IU, an unsolicited command IU, or an unsolicited data IU, and

– no other condition exists or has been recognized that requires the sending of either a link-level-busy IU or an FC-PH primitive sequence.

An LRJ IU shall also be sent by a channel as the response to an ELP IU received with a TYPE field of hex'1B', provided all of the above conditions are satisfied and any of the following conditions are satisfied.

– The channel does not support CTC connection or does not support CTC connection for the requested logical path. (See 6.4.2.)

– The channel is not waiting for a response from an ELP IU previously sent to the channel from which it received the ELP IU, and the CTC counter field of the received ELP IU contains a valid CTC counter value which is less than the number of CTC connections for which the channel is providing the emulated control unit functionality. (See 8.12.2.4.)

– The channel is not waiting for a response from an ELP IU previously sent to the channel from which it received the ELP IU, the CTC counter field of the received ELP IU contains a valid CTC counter value which is equal to the number of CTC connections for which the channel is providing the emulated control unit functionality, and the N_Port ID of the channel is less than the N_Port ID of the channel which sent the received ELP IU.

– The channel is waiting for a response from an ELP IU previously sent to the channel from which it received the ELP IU, and the CTC counter field of the received ELP IU contains a valid value which is less than the CTC counter value specified in the ELP IU previously sent.

– The channel is waiting for a response from an ELP IU previously sent to the channel from which it received the ELP IU, the CTC counter field of the received ELP IU contains a valid value which is equal to the CTC counter value specified in the ELP IU previously sent, and the N_Port ID of the channel is less than the N_Port ID of the channel which sent the received ELP IU.

When an LRJ IU is received in response to an IU which initiates a connection, the type of error shall be indicated by the reject-reason code. If the LRJ IU is received in response to an IU sent during a connection, a link-level protocol error shall be recognized.

The LRJ IU shall have a two byte link-control information field. Bits 1-7 of the first byte of the link-control information field shall contain the reject-reason code. The link-control information field of an LRJ IU shall have the format shown in Figure 11.

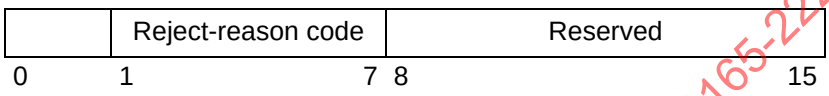

Figure 11 – Link-control information field of the LRJ IU

The reject-reason code shall appear in bits 1-7 of the link-control information field of an LRJ IU. This field shall identify the error condition that caused the IU to be rejected.

When a channel sends the LRJ IU, the only valid reject-reason codes shall be ‘000 0110’b, corresponding to the logical path not established reject-reason code, and ‘000 1001’b, corresponding to protocol error reject-reason code, respectively. All other reject-reason codes shall be reserved. When a control unit sends the LRJ IU, the only valid reject-reason code is ’000 0110’b, corresponding to the logical path not established reject-reason code. All other reject-reason codes shall be reserved.

The protocol error reject-reason code is used by the channel to indicate an error condition if an ELP IU with a type code of hex’1B’ is received. The logical path not established reject-reason code is used by the channel and the control unit when a logical path not established error condition is detected.

Bit zero and bits 8-15 of the link-control-information field shall be reserved for future use; these bits shall be set to zero by the sender and are ignored by the recipient of an LRJ IU.

A link-level protocol error shall be detected if an LRJ IU is received with a reserved reject-reason-code value.

6.4.10 Link-level busy

The link-level-busy (LBY) function shall indicate that an initiation IU was received and discarded by a channel or control unit because of a link-busy condition. The LBY function shall be performed by sending an LBY IU to the source of the IU which was discarded. The link-control DIB shall contain a link header with a link-control field specifying the LBY function and no link payload.

The channel and control-unit image ID fields shall be set to the corresponding values in the IU for which the LBY IU is being sent. If the IU for which the LBY IU is being sent contains other than a link-control DIB and the AS bit is set to one, then the AS bit shall be set to one in the LBY IU and the device address shall be provided. If the AS bit is set to zero in the IU for which the LBY is being sent, the AS bit shall be set to zero in the LBY IU.

An LBY IU shall be sent as a response when an IU is not accepted because of a link-busy condition. Link-busy conditions for which the LBY IU is sent shall be temporary conditions which cause an initiation IU to be discarded such as when a self-initiated function is being performed or when an SB-2 resource required for the reception of the initiation IU is temporarily unavailable.

An LBY IU shall be sent only if all of the following conditions are satisfied:

– the IU is an initiation IU,

– the IU contains no FC-PH errors,

– the IU contains no exchange errors,

– the IU is either an unsolicited control IU, an unsolicited command IU, or an unsolicited data IU, and

– no other condition exists or has been recognized that requires the sending of either a link-level-reject IU or an FC-PH primitive sequence.

When an LBY IU is received in response to an initiation IU a temporary busy condition shall be recognized, and the link-level or device-level function may be immediately retried or retried at a later time. When an LBY IU is received in response to an IU while a connection exists, a link-level protocol error shall be detected.

7 N_Port link initialization

7.1 N_Port link initialization overview

This clause describes the initialization procedures for a channel and control unit unique to SB-2.

The initialization process establishes the necessary conditions for elements of a channel path to be able to sustain both link-level and device-level communication. Under normal conditions, the initialization process occurs infrequently; for example, it is usually required only after a power on procedure or during a system-initialization procedure and is not part of the execution of an I/O operation.

The initialization process may be thought of as a series of hierarchical steps.

For the channel, the steps of the initialization process and their associated procedures are:

a) link-initialization procedure,

b) channel login procedure,

c) channel node-identifier-acquisition procedure,

d) channel state-change-registration procedure,

e) channel-link-incident-record-registration procedure,

f) channel logical-path-establishment procedure.

For the control unit, the steps of the initialization process and their associated procedures are:

a) link-initialization procedure,

b) control unit login procedure,

c) control unit node-identifier-acquisition procedure,

d) control unit state-change-registration procedure,

e) control unit logical-path-establishment procedure.

If a procedure encounters an FC-PH or SB-2 response indicating a busy condition, the procedure shall be retried until either the busy condition no longer exists and the procedure is successful or a condition other than a busy condition is encountered, in which case, the protocols defined for that condition determine the action to be taken. The retry for a busy condition may be deferred until attempts to complete the procedure with other N_Ports have been made.

If a procedure is not successful because of an unrecoverable error, that portion of the initialization process shall be terminated and all or a portion of the channel or control unit shall be considered uninitialized, depending on which initialization procedures were not successful.

The hierarchy of initialization steps shall also be followed when there is an error or another event that causes regression within the initialization process. When an error or other event indicates that the results achieved at a previously completed step are no longer valid, information associated with that step and possibly subsequent steps in the hierarchy shall be discarded or may be considered no longer current for node-identifier information, and the initialization process for the affected channel or control

unit shall be repeated for all affected steps. (See 11.2 and 6.3.7.) A node-identifier that is not current is an identification of the last known node attached to the N_Port, but as the result of some event, such as an FC-PH link-failure condition, may not reflect the current attached node, since a configuration change may have occurred. For example, if a channel observes that a link which was previously considered operational is now not operational because of an FC-PH link-failure condition, then the channel N_Port is implicitly logged out with the N_Port or F_Port at the other end of the link, and if the FC-PH link failure condition persists for longer than SB_TOV, it removes any logical paths established for that link, and may discard the neighboring-node identifier or may consider the identifier to be not current. For this situation, the initialization process would start again with the link initialization procedure when the FC-PH link failure condition no longer exists.

A Registered State Change Notification (RSCN) ELS command received from a fabric controller or N_Port which contains the N_Port ID of a channel or control unit for which the recipient of the RSCN considers the initialization process complete, may cause a test-initialization IU to be sent in order to verify the effect of the state change on previously established logical paths. If the test-initialization-result IU confirms that logical paths previously established by the initialization process are no longer considered established, the procedures shall be attempted, beginning at the appropriate step within the initialization-process hierarchy, to reestablish the logical paths.

An RSCN ELS command received from a fabric controller or N_Port which contains the N_Port ID of a channel or control unit for which the recipient of the RSCN considers the initialization process incomplete, may cause a test-initialization IU to be sent in order to verify the effect of the state change on logical paths previously established. If the test-initialization-response IU confirms that logical paths previously established by the initialization process are no longer considered established, the procedures shall be attempted, beginning at the appropriate step within the initialization-process hierarchy, to reestablish logical paths. If the test-initialization-result IU confirms that logical paths previously established are still valid, then the previously incomplete step shall be retried, and, if successful, the next step in the hierarchy shall be attempted for each logical path that is required on the channel path but that is not yet established. If the retry does not result in the successful completion of the initialization process, the recipient of the RSCN shall continue to consider the initialization process incomplete for that N_Port.

When an error or other event causes regression to the node-identifier acquisition step or to a previous step within the initialization process and the existing node-identifier of its neighbor is valid, the N_Port shall set the flag-field node-ID-validity code to the value 1 (node-ID is valid but may no longer be current) in the node descriptor for the affected neighboring node.

NOTE Many of the processes used during initialization require that the appropriate action or response be taken within a prescribed period of time. The values selected for these periods of time are significantly greater than the worst-case propagation delay. To make shure that these time values are not compromised, it is important that each component in the path be designed to contribute the minimum amount of delay and that no one component in the path use a significant portion of the allowed time value. It should also be recognized that factors other than those directly associated with the procedure being performed may subtract from the time allowed. For example, when propagating a frame through a Fabric, the time required to route the frame through the Fabric should be taken into account.

7.2 Link-initialization procedure

Link initialization is described in FC-PH, reference [1]. When link initialization is complete, the N_Port or F_Port is in the active state. (See FC-PH, reference [1], clause 16.6.2.)

Once link initialization is complete for an N_Port or F_Port, the N_Port or F_Port is considered to be operational as long as it remains in the active state. (See FC-PH, reference [1] clause 16.5.)

When an N_Port or F_Port exits the active state, the attached link is considered not operational, and the link initialization procedure shall be successfully completed in order to consider it operational.

7.3 Initialization process for a channel

7.3.1 Channel initialization overview

For a channel, the initialization process consists of the following procedures: link initialization, channel login, channel node-identifier acquisition, channel state-change registration, channel link-incident-record registration and channel logical-path establishment.

7.3.2 Channel login

The N_Port of a channel shall attempt to perform channel login following the completion of link-initialization. Channel login consists of two steps: fabric login and N_Port login. The login protocol is described in FC-PH, reference [1], clause 23.

7.3.2.1 Channel F_Port login

The N_Port of a channel, including the N_Port of a channel providing CTC connection, shall initiate channel login by sending an FLOGI ELS command with S_ID=hex'00 00 00' and D_ID=hex'FF FF FE'. The command shall be sent using Class 2 service, and support for Class 2 and Class 3 shall be indicated in the service parameters. If the channel is attached to a Fabric, the response from the Fabric assigns the channel an N_Port ID. (Annex A, "," describes an address assignment scheme which allows the use of simplified configuration record formats.) If an accept response is received, and if support for sequential delivery and Classes 2 and 3 service is indicated, then the channel shall proceed with further steps in the initialization procedure. If errors prevent the successful receipt of an accept response, or if the service parameters of the response do not indicate support for sequential delivery and Classes 2 and 3 service, further steps in the initialization procedure shall not be performed.

If the N_Port of a channel receives an FLOGI ELS command before it is able to send an FLOGI ELS command, or if the response to the FLOGI ELS command indicates that the channel is not attached to a Fabric, then the N_Port of the channel shall proceed with N_Port login instead of F_Port login.

7.3.2.2 Channel N_Port login

7.3.2.2.1 Channel N_Port login overview

After the N_Port of a channel (including a channel providing CTC connection) has logged in with the Fabric, or if the attempted fabric login has revealed that the N_Port of the channel is directly attached to another N_Port, the N_Port of the channel shall perform N_Port login. When performing N_Port login, Class 2 service shall be used, and support for the following features shall be indicated:

– Classes 2 and 3, and

– continuously increasing relative offset for solicited data information category.

7.3.2.2.2 Channel N_Port login: point-to-point configurations

If the N_Port of the channel is directly attached to another N_Port, the N_Port of the channel shall perform login only with the attached N_Port. If no errors occur during N_Port login, and if support for Class 2, Class 3, and continuously increasing relative offset for solicited data information category is indicated, then the channel considers N_Port login successful, and further steps in the initialization process shall be performed. If errors occur, if support for Class 2 and Class 3 is not indicated, or if support for

continuously increasing relative offset for solicited data information category is not indicated, then the channel considers N_Port login unsuccessful, and further steps in the initialization process shall not be performed.

7.3.2.2.3 Channel N_Port login: fabric configurations

If the N_Port of a channel is attached to a Fabric, then after F_Port login is complete, the N_Port of the channel shall initiate N_Port login with the N_Ports of each control unit which is configured to the channel, with each channel which supports CTC connection which is configured to the channel, with the management server, and with the fabric controller as required in further steps in the initialization process. When an initialization procedure requires communication with an N_Port with which the N_Port of the channel is not logged in, then the N_Port of the channel shall initiate N_Port login with that N_Port before performing the procedure. The channel also performs N_Port login with a control unit if it receives a PLOGI ELS from a control unit.

If no errors occur during N_Port login, if support for Class 2 and Class 3 is indicated, and (for the case of login with a control unit or with a channel which supports CTC connection) if support for continuously increasing relative offset for solicited data information category is indicated, then the channel shall consider N_Port login successful with respect to the N_Port, and further steps in the initialization process with respect to that N_Port shall be performed; otherwise, the channel shall consider N_Port login with that N_Port unsuccessful, and further steps in the initialization process with respect to an N_Port shall not be performed.

7.3.3 Channel Node-Identifier acquisition

The channel node-identifier-acquisition procedure provides the channel a means by which the channel shall acquire the neighboring-node identifier. The neighboring-node identifier refers to the identification of the node attached to the other end of the link. The N_Port of a channel, including the N_Port of a channel providing CTC connection, shall attempt to acquire the neighboring-node identifier by sending a RNID ELS command as soon as the channel has completed login with the neighboring node. (For additional information on obtaining a neighboring-node identifier, see 6.3.7.) When the recipient of the RNID ELS command is able to provide node-identification data of type SB-2, it shall respond with the node-identifier and a flag indicating the validity of the node-identifier. The preferred implementation is for the neighboring node to immediately return a valid node-identifier.

If, an LS_RJT is received in response to the RNID ELS command, or if errors occur which prevent the N_Port of the channel from sending the RNID ELS request or receiving the requested node-identifier, the channel shall consider node-identifier acquisition of its neighbor unsuccessful. See FC-PH, reference [1], for information on LS_RJT.

The channel shall continue the initialization process even if the attempt to acquire the node-identifier of its neighbor was unsuccessful. The lack of success in acquiring a valid and current node-identifier shall not prevent the establishment of logical paths or the execution of link-level and device-level functions. The retry attempt to acquire the node-identifier may be deferred for a link error, or an FC-PH busy or reject indication.

A Channel that checks the node-descriptor flag field shall perform the following actions.

a) When the node-ID-validity code is zero, the channel shall establish the received 32-byte node descriptor as the node descriptor of its neighbor, and channel node-identifier acquisition is complete.

b) When the node-ID-validity code is not zero, the channel shall either

1) Check the existing node-descriptor node-ID-validity code if it had previously established the node descriptor of its neighbor and, if valid and not current, maintain the existing 32-byte node descriptor with an indication that the node-identifier is not current, or maintain an indication that the node descriptor of its neighbor is not valid, or

2) Not check the existing node-descriptor node-ID-validity code and maintain an indication that the node descriptor of its neighbor is not valid.

See note 1 for additional information.

When the node-ID-validity code is not zero, the channel shall retry channel node-identifier acquisition until a valid node descriptor is acquired. If there is reason to suspect that subsequent retries may not be successful, the node-identification procedure shall be suspended. If the channel has to provide node descriptors in a function such as link-incident reporting prior to acquisition of a valid and current node descriptor, the channel shall send the node descriptor of its neighbor with a node-ID-validity code of 1 or 2, as appropriate.

NOTE 1 When a channel checks the node-descriptor flag field and recognizes that the node-ID-validity code is not zero, the preferred implementation is option b1).

NOTE 2 When a condition occurs that affects the validity of the neighboring-node identifier, the channel should consider the neighboring-node identifier as not current. This requirement applies for all conditions except for powering off or initial machine loading (IML), in which case, the channel may discard the neighboring-node identifier.

7.3.4 Channel state-change registration

The channel state-change-registration procedure enables the N_Port of a control unit or channel supporting CTC connection (in a point-to-point configuration), or the fabric controller (in a fabric configuration) to send registered-state-change notifications to the N_Port of a channel. (See 6.3.5.)

If the N_Port of a channel, including a channel supporting CTC connection, is directly attached to another N_Port, the N_Port of the channel shall register for state-change notification by sending the SCR ELS command to the attached N_Port; however, if the channel is already registered to receive state-change notifications, it need not re-register. If the channel, including a channel supporting CTC connection, is attached to a Fabric, the N_Port of the channel shall register for state change notification by sending an SCR ELS command to the fabric controller; however, if the channel is already registered to receive state-change notifications, it need not re-register. For additional information on the SCR ELS command, see 6.3.6. After accepting the SCR ELS command, the fabric controller sends an RSCN ELS command to the N_Port of the channel when other N_Ports have potentially changed their states. When an accept response to the SCR ELS command is received, all IUs containing link-level information which are allowed for a channel may be sent unless link-incident-record registration is required. If errors occur which prevent the channel from sending the SCR ELS command or receiving an accept response, then the channel shall not proceed with further steps in the initialization process.

7.3.5 Channel Link-Incident-Record registration

The channel link-incident-record-registration procedure enables control units which are not emulated control units and the management server to send RLIR ELS commands to the channel. (See 6.3.8 and 6.3.10 for additional information on the RLIR ELS and the LIRR ELSs.) The channel is not required to perform link-incident record registration with other channels providing CTC connection.

The N_Port of a channel shall perform the link-incident-record-registration procedure by sending the LIRR ELS command as soon as it has completed the channel state-change-registration procedure. If the N_Port of a channel is directly attached to a control unit which is not an emulated control unit, the

LIRR ELS command shall be sent to the N_Port of the control unit. If the N_Port of the channel is attached to a Fabric, the LIRR ELS command shall be sent to the management server. In addition for a fabric configuration, link-incident-record registration shall be performed with each control unit with which the channel completes the N_Port login procedure; however, link-incident record registration is not required to be performed with other channels providing CTC connection.

If the N_Port of a channel receives an accept response to the LIRR ELS command, the channel shall consider link-incident-record registration successful with respect to the N_Port to which the request was sent, and the channel shall continue with further steps in the initialization process with respect to that N_Port. If an LS_RJT is received in response to the LIRR command, or if errors occur which prevent the N_Port of a channel from sending the LIRR ELS command to an N_Port or receiving an accept response, the channel shall consider link-incident-record registration unsuccessful with respect to the N_Port, and the channel shall continue the initialization process even though link-incident-record registration with the N_Port was unsuccessful. When the channel considers link-incident record registration unsuccessful, it is allowed to register for another type of link incident record, or for common link-incident records.

7.3.6 Channel Logical-Path establishment

The last initialization procedure performed at the channel is the establishment of logical paths between the channel images sharing the N_Port and the control-unit images configured to the channel images. The channel logical-path-establishment procedure creates, at the channel, the information necessary for a particular channel image to communicate with a particular control-unit image to perform I/O operations.

The channel shall initiate the establishment of the logical path by sending an ELP IU. If the control unit is able to form a logical path between the channel image and control-unit image, it shall respond with the LPE IU indicating that the request is accepted and the logical path has been established; otherwise, it shall indicate that the logical path is not established and the reason why. When the logical path is established, the channel shall then allow device-level communication on that logical path between the specified channel image and control-unit image. (See 6.4.2.)

The channel shall consider the logical path to be established upon receiving an error-free LPE IU from the control unit. When the channel considers the logical path to be established, the channel shall consider the initialization process to be complete for the combination of that channel image and control-unit image. When the logical path is established, the N_Ports at each end of the physical path, and the device-level facilities associated with those N_Ports shall be considered operational and have the capability of performing their respective functions. (See Clause 8 for information about IUs containing device-level information.)

A logical path shall no longer be usable for the exchange of IUs when either end of the path no longer considers the logical path to be established. A channel shall consider a logical path to be no longer established when either of the following events occurs.

– The channel sends a RLP request to the control unit, and an error-free LPR response is returned. (See 6.4.3.)

– A channel detects an error for which the recovery action includes removing the logical path. (See 11.2.)

When a logical path is not established, the channel shall not perform device-level functions and protocols with respect to the corresponding control-unit image.

When the channel is initialized, one or more attempts to establish a logical path with each control-unit image in the channel-path configuration shall be made. The number of times beyond one that the channel attempts to establish the logical path for a control-unit image during initialization is model dependent.

A logical path may be removed by a control unit because of an SB-2 link failure, a logical path timeout error, an SB-2 offline condition, or a condition internal to the control unit; when this occurs, it may be asynchronous to the activity of the channel. (See 11.2.) If the control unit is not actively communicating with the channel image, the channel may not be immediately aware of the loss of the logical path. When the channel image later attempts to perform a device-level function using the logical path, abnormal conditions which affect the initialization of the channel are encountered. (See 11.2.8.)

During the initiation of a device-level function for a device or control unit, if the channel determines that a logical path, which it considers to exist, is to be removed, then the channel shall consider the device or control unit not operational with respect to the logical path and shall terminate the device-level function on this logical path. The channel shall also recognize a not-operational condition for all other devices for which an I/O operation was active or disconnected with respect to that logical path. All other devices associated with the logical path that is removed shall not be affected.

During the initiation of a device-level function for a device or control unit, if the channel determines that the logical path was previously not established, the channel shall attempt, by means of the initialization process, to establish the logical path before performing the device-level function. If the logical path is successfully established, the channel shall proceed to attempt to initiate the device-level function if the device-level function is still pending. If the initialization process is terminated during the initiation of the device-level function, the logical path shall remain not established and the channel shall consider the device or control unit to be not operational with respect to that logical path.

7.4 Initialization process for a control unit

7.4.1 Control unit initialization overview

For a control unit, the initialization process consists of the following procedures: link initialization, control unit login, control unit node-identifier acquisition, control unit state-change registration, and control unit logical-path establishment.

7.4.2 Control-unit login

The N_Port of a control unit shall attempt to perform control-unit login following the completion of link-initialization. Control-unit login consists of two steps: F_Port login and N_Port login. For a description of F_Port and N_Port login protocol, see FC-PH, reference [1], clause 23.

7.4.2.1 Control unit F_Port login

Except in the case of an emulated control unit in which F_Port login is performed by the channel, the N_Port of a control unit shall initiate control-unit login by sending an FLOGI ELS command with S_ID=hex'00 00 00' and D_ID=hex'FF FF FE'. The command shall be sent using Class 2 service, and support for Class 2 and Class 3 shall be indicated in the service parameters. If the N_Port of the control unit is attached to a Fabric, the response from the Fabric assigns the N_Port of the control unit an N_Port ID. (Annex A describes an address assignment scheme which allows the use of simplified configuration record formats.) If an accept response is received, and if support for sequential delivery and Classes 2 and 3 service is indicated, then the control unit shall proceed with further steps in the initialization procedure. If errors prevent the successful receipt of an accept response, or if the service parameters of the response do not indicate support for sequential delivery and Classes 2 and 3 service, further steps in the initialization procedure shall not be performed.

If the N_Port of a control unit receives an FLOGI ELS command before it is able to send an FLOGI ELS command, or if the response to the FLOGI ELS command indicates that the N_Port of a control unit is not attached to a Fabric, then the control unit shall proceed with N_Port login instead of F_Port login.

7.4.2.2 Control unit N_Port login

7.4.2.2.1 Control unit N_Port login overview

After the N_Port of the control unit has logged in with the Fabric, or if fabric login has indicated that the N_Port of the control unit is directly attached to another N_Port, the control unit shall perform N_Port login. When performing N_Port login, Class 2 service shall be used and support for the following features shall be indicated:

– Classes 2 and 3, and

– continuously increasing relative offset for solicited data information category.

7.4.2.2.2 Control unit N_Port login: point-to-point configurations

If the N_Port of the control unit is directly attached to another N_Port, the N_Port of the control unit shall perform login with the attached N_Port; however, for the case of an emulated control unit, the N_Port of the channel is the same as the N_Port of the control unit and N_Port login is not repeated on behalf of the N_Port of the emulated control unit. If no errors occur during N_Port login, and if support for Class 2, Class 3, and continuously increasing relative offset for solicited data information category is indicated, then the control unit considers N_Port login successful, and further steps in the initialization process shall be performed. If errors occur, if support for Classes 2 and 3 is not indicated, or if support for continuously increasing relative offset for solicited data information category is not indicated, then control-unit login is regarded as unsuccessful, and further steps in the initialization process shall not be performed.

7.4.2.2.3 Control unit N_Port login: fabric configurations

If the N_Port of the control unit is attached to a Fabric, then after F_Port login is complete, the N_Port of the control unit shall initiate N_Port login with the management server and fabric controller as required in further steps in the initialization process. (See 7.4.3 and 7.4.4.) When an initialization procedure requires communication with an N_Port with which the control unit is not logged in, the N_Port of the control unit shall initiate N_Port login with that N_Port before performing the procedure. The N_Port of a control unit also initiates N_Port login with a channel when the control unit has received initiative to send a TIN to that channel but the N_Port of the control unit is not logged in with the N_Port of the channel. (See 6.4.7.) In all other cases, the control unit performs N_Port login only upon receiving a PLOGI ELS request.

If no errors occur during N_Port login, if support for both Class 2 and Class 3 is indicated, and (for the case of login with the N_Port of a channel) if support for continuously increasing relative offset for solicited data information category is indicated, then the control unit shall consider N_Port login successful with respect to that N_Port, and further steps in the initialization process with respect to that N_Port shall be performed; otherwise, the control unit shall consider login with that N_Port unsuccessful, and further steps in the initialization process with respect to that N_Port shall not be performed.

7.4.3 Control unit node-identifier acquisition

The control unit node-identifier-acquisition procedure provides a means by which the control unit shall acquire the neighboring-node identifier. Except in the case of an emulated control unit where node-

identifier acquisition is done by the channel, the N_Port of a control unit shall attempt to acquire the neighboring-node identifier by sending an RNID ELS command as soon as it has completed login with the N_Port of the neighboring node. (See 6.3.7 for additional information about the RNID ELS.)

When the recipient of the RNID ELS command is able to provide node-identification data of type SB-2, it shall respond with the node-identifier and a flag indicating the validity of the node-identifier. The preferred implementation is for the neighboring node to immediately return a valid node-identifier.

If an LS_RJT is received in response to the RNID ELS, or if errors occur which prevent the control unit from sending the RNID ELS request or receiving a reply containing the requested node-identifier, the control unit shall consider node-identifier acquisition of its neighbor unsuccessful.

The control unit shall continue the initialization process even if the attempt to acquire the node-identifier was unsuccessful. The lack of success in acquiring a valid and current node-identifier shall not prevent the establishment of logical paths or the execution of link-level and device-level functions.

A control unit that checks the node-descriptor flag field shall perform the following actions:

a) When the node-ID-validity code is zero, the control unit shall establish the received 32-byte node descriptor as the node descriptor of its neighbor, and control unit node-identifier acquisition is complete.

b) When the node-ID-validity code is not zero, the control unit shall either:

   1) Check the existing node-descriptor node-ID-validity code if it had previously established the node descriptor of its neighbor, and, if valid and not current, maintain the existing 32-byte node descriptor with an indication that the node-identifier is not current, or maintain an indication that the node descriptor of its neighbor is not valid, or

   2) Not check the existing node-descriptor node-ID-validity code and maintain an indication that the node descriptor of its neighbor is not valid.

   See note 1 for additional information.

When the node-ID-validity code is not zero, the control unit shall retry control unit node-identifier acquisition until a valid node descriptor is acquired. If there is reason to suspect that subsequent retries may not be successful, the node-identification procedure shall be suspended. If the control unit has to provide node descriptors in a function such as link-incident reporting prior to acquisition of a valid and current node descriptor, the control unit shall send the node descriptor of its neighbor with a node-ID-validity code of 1 or 2, as appropriate.

NOTE 1 When a control unit checks the node-descriptor flag field and recognizes that the node-ID-validity code is not zero, the preferred implementation is option b1).

NOTE 2 When the response to the RNID ELS request contains a value of 2 in the Node-ID-validity field, the control unit should not resend the RNID ELS request so frequently that the frame exchanges seriously degrade the execution of other operations on the link.

NOTE 3 When a condition occurs that affects the validity of the neighboring-node identifier, the control unit should consider the neighboring-node identifier as not current. This requirement applies for all conditions except for powering off or initial machine loading (IML), in which case, the control unit may discard the neighboring-node identifier.

NOTE 4 In a preferred implementation involving a Fabric, the Management Server attempts to acquire the node-identifiers of each N_Port which has acquired a node-identifier from the Management Server.

7.4.4 Control unit State-Change registration

The control unit state-change-registration procedure enables the N_Port of a channel (in a point-to-point configuration) or a fabric controller (in a fabric configuration) to send registered state change notifications to the control unit. (See 6.3.5.)

If the N_Port of a control unit is attached to another N_Port, the N_Port of the control unit shall register for state-change notification by sending the SCR ELS command to the other N_Port except in the case of an emulated control unit where state-change registration is performed by the channel, or in the case in which the control unit is already registered to receive state-change notifications. If the N_Port of the control unit is attached to a Fabric, the N_Port of the control unit shall register for state-change notifications by sending an SCR ELS command to the fabric controller, except in the case of an emulated control unit, where state change registration is done by the channel, or in the case in which the control unit is already registered to receive state-change notifications. For additional information on the SCR ELS command, see 6.3.6. After the SCR ELS command is accepted, RSCN ELS commands are sent to the N_Port of the control unit when other N_Ports have potentially changed their states. When an accept response to the SCR ELS command is received, the control unit shall consider state-change registration successful, and all IUs containing link-level information which are allowed for a control unit may be sent. If errors occur which prevent the sending the SCR ELS command or receiving an accept response, the control unit shall consider state-change registration unsuccessful and shall not proceed with further steps in the initialization process.

7.4.5 Control unit logical-path establishment

The control unit shall attempt to establish a logical path when a link-control IU containing a request for the establishment of a logical path between a specified channel image and control-unit image is received. The control-unit-logical-path-establishment procedure consists of either accepting or refusing the requested logical path. If the control unit is able to accept the specified logical path, it shall respond with a link-control IU containing an indication that the request is accepted and the logical path has been established; otherwise, it shall indicate that the logical path is not accepted. When the logical path is established, the control unit shall then allow device-level communication on that logical path between the channel image and control-unit image. (See 6.4.2.)

The control unit shall consider the logical path to be established upon sending the logical-path-established response without errors. When the control unit considers the logical path to be established, the control unit shall consider the initialization process to be complete for the combination of that channel image and that control-unit image.

A logical path shall no longer be usable for the exchange of IUs containing device-level information when either end of the path no longer considers the logical path to be established. A control unit shall consider a logical path to be no longer established when any of the following events occurs.

– A remove-logical-path request is received, and an error-free response is returned indicating logical path removed. (See 6.4.3.)

– The control unit detects an error for which the recovery action includes removing the logical path. (See 11.2.)

– The control unit is powered off.

When a logical path is not established, the control unit shall not perform device-level functions and protocols with respect to the logical path.

NOTE A logical path may be lost and reestablished without the recognition of an error condition. When a logical path is not established, the equivalent of a system reset occurs with respect to that logical path. This reset without the recognition of an error condition is acceptable only because there is an explicit requirement for all SB-2-I/O-interface devices to provide reset-event notification. (See 9.4.) If a logical path is removed, either the program is aware of the reset or the device reports that a reset has occurred. The report occurs only if the logical path is reestablished and an I/O operation is initiated to the device on the logical path.

8 SB-2 Information units

8.1 SB-2 Information unit overview

FC-SB-2 is based upon the FC-4 Information Unit construct described in FC-PH, reference [1].

Information associated with the execution of an I/O operation and the operation of a device is transferred between the channel and control unit as Information Units. SB-2 Information Units (IUs) contain SB-2 device-level commands, status, data, or control information or SB-2 link-level control information. All SB-2 IUs are sent as FC-4 device-data frames, FC-PH routing control bits set to '0000'b.

There are five basic SB-2 IU types that correspond to five of the FC-PH information categories. They are: the unsolicited-command IU, the solicited-data IU, the unsolicited-data IU, the solicited-control IU, and the unsolicited-control IU. For brevity these IUs may be referred to as the command IU, data IU, and control IU in the remainder of this document.

8.2 Rules for sending SB-2 IUs

8.2.1 Overview of rules for sending SB-2 information units

This subclause defines the information category a channel or control unit uses when it sends an SB-2 IU, and the placement of the IU within an exchange. All of the allowed information categories and placements within an exchange of SB-2 IUs are summarized in Table 6 and Table 7. When a particular IU is sent, however, its information category and its placement within an exchange may depend on the conditions under which the IU is sent. For those cases in which such a dependency exists, rules define the conditions under which an information category is used and the conditions under which the IU starts or ends an exchange. Rules for IUs which contain device-level information are given following the tables. Rules for IUs which contain link-level information are given in 6.4. Rules for control IUs are given in 8.11.

Table 6 – Characteristics of IUs sent by a channel

DIB type	F, M, or L	Information category
Command	F or M	Unsolicited command
Command-Data	F or M	Unsolicited command
Data	M	Solicited data
Control	F	Unsolicited control
Control	F or M or L or FxL	Solicited control
Link-Control (request)	F	Unsolicited control
Link-Control (response)	FxL or L	Solicited control

F: IU opens an exchange but does not close it
M: IU is neither first nor last IU of an exchange
L: IU closes a pre-existing exchange
FxL: IU opens and closes an exchange

.

Table 7 – Characteristics of IUs Sent by a Control Unit

DIB Type	F,M, or L	Information Category
Data	M	Solicited data
Control	F or M or L or FxL	Solicited control
Status	F or M or L or FxL	Solicited data
Status	F	Unsolicited data
Link-Control (request)	F	Unsolicited control
Link-Control (response)	FxL or L	Solicited control

F: IU opens an exchange but does not close it
M: IU is neither first nor last IU of an exchange
L: IU closes a pre-existing exchange
FxL: IU opens and closes an exchange

8.2.2 Rules for device-level functions

a) A channel shall send an IU containing a command or command-data DIB in an unsolicited command IU as the first or middle sequence of an exchange, depending on whether there is an exchange already open for that device. The exchange shall be left open.

b) A status DIB shall be sent as an initiation IU unless sent for a specific device when an inbound exchange already exists for that device.

c) Status shall be sent as an information category solicited data if sent on an existing exchange or if sent as an initiation IU in direct response to an initiation IU from the channel. The data IU containing the status DIB shall close the inbound exchange unless one of the following conditions applies.

– All of the conditions for chaining are satisfied and device-end status is included in the status DIB.

– The status is sent in response to a control IU.

– The status is sent as a result of receiving an early end indication.

– Supplemental status is included in the status DIB.

Status sent in response to a selective-reset IU requesting retry or unit check with the CI bit set to one shall be regarded as sent in response to the selective-reset IU (a control IU), and the inbound exchange shall be left open.

d) Status shall be sent as an information category unsolicited data if it is sent as an initiation IU and is not a direct response to an initiation IU from the channel. The inbound exchange shall be left open.

e) A control unit shall not send a status DIB for any specific device, on the same logical path, on more than one exchange at the same time.

f) If a channel sends an initiation IU containing a device-level control function or a command to a device on a logical path and simultaneously a control unit sends an initiation IU as information category unsolicited data containing status for the same device on the same logical path, then the control unit shall send a response to the initiation IU using the existing inbound exchange with an IU which contains the same token as the initiation IU from the channel. (See 8.5.5.) The channel shall discard the initiation IU from the control unit and shall wait for another IU on the inbound exchange from the same device which contains a response to its initiation IU.

8.3 SB-2 IU structure

All SB-2 IUs contain an 8-byte SB-2 header followed by an 8-byte IU header. Immediately following the IU header is a field referred to as “the device-information block (DIB).” A DIB shall contain a minimum of 16 bytes; consisting of a 12 byte DIB header followed by four bytes of longitudinal-redundancy check (LRC). (See 8.6.3.) For certain DIB types, a variable length DIB data field may immediately follow the LRC. The maximum length of the DIB data field shall be 8160 bytes. Six different DIB types are defined; they are: the data DIB, command DIB, status DIB, control DIB, command-data DIB, and link control DIB. The type and structure of the DIB is determined by bits in the information unit identifier (IUI) field of the IU header. (See 8.5.2.)

The structures of SB-2 IUs are shown in Figure 12.

SB-2 general IU payload structure

SB-2 header	IU header	DIB header	LRC	DIB data
(8)	(8)	(12)	(4)	(0 - 8160)

SB-2 unsolicited command IU with command DIB

SB-2 header	IU header	Command header	LRC
(8)	(8)	(12)	(4)

SB-2 unsolicited command IU with command-data DIB

SB-2 header	IU header	Command header	LRC	Data	PAD	CRC
(8)	(8)	(12)	(4)	(0 - 8160)	(0-3)	(0/4)

SB-2 solicited data IU with data DIB

SB-2 header	IU header	Data header	LRC	Data	PAD	CRC
(8)	(8)	(12)	(4)	(0 - 8160)	(0-3)	(0/4)

SB-2 solicited/unsolicited data IU with status DIB

SB-2 header	IU header	Status header	LRC	Supplemental status	PAD	CRC
(8)	(8)	(12)	(4)	(0 - 32)	(0-3)	(0/4)

SB-2 solicited/unsolicited control IU with control DIB

SB-2 header	IU header	Control header	LRC
(8)	(8)	(12)	(4)

SB-2 solicited/unsolicited control IU with link control DIB

SB-2 header	IU header	Link header	LRC	Link payload	PAD	CRC
(8)	(8)	(12)	(4)	(0 - 8 156)	(0-3)	(0/4)

Figure 12 – IU payload structures

8.4 SB-2 header

8.4.1 SB-2 header format

The SB-2 header provides the FC-4 addressing information needed to identify the logical path and the device for the exchange. Figure 13 shows the SB-2 header format.

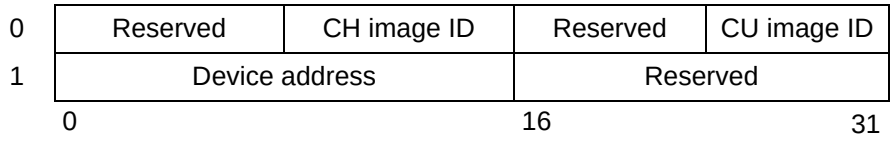

Figure 13 – SB-2 header

Bits 0-7 and 16-23 of word 0 and bits 16-31 of word 1 shall be reserved.

8.4.2 Channel image ID

Bits 8-15 of word 0 contain the 8-bit ID of the channel image. For all IUs except a control IU containing a link-control DIB the channel image ID specified shall correspond to a logical path previously initialized by the establish-logical-path procedure; otherwise a logical-path-not-established error shall be recognized. For a link-control DIB, the channel-image ID shall be meaningful for the following link-control functions: establish-logical path (ELP), logical-path established (LPE), remove-logical path (RLP), logical-path removed (LPR), test-initialization (TIN), test-initialization result (TIR), link-level reject (LRJ), and link-level busy (LBY). For the LACK link-control function the channel-image ID shall be set to zero by the sender and ignored by the recipient.

8.4.3 Control-unit image ID

Bits 24-31 of word 0 contain the 8-bit ID of the control-unit image. For all IUs except a control IU containing a link-control DIB, the control-unit image ID specified shall correspond to a logical path previously initialized by the establish-logical-path procedure; otherwise a logical-path-not-established error shall be recognized. For a link-control DIB, the control-unit image ID shall be meaningful only for the following link-control functions: establish-logical path (ELP), logical-path established (LPE), remove-logical path (RLP), logical-path removed (LPR), test initialization (TIN), test initialization result (TIR), link-level reject (LRJ), and link-level busy (LBY). For the LACK link-control function the control-unit image ID shall be set to zero by the sender and ignored by the recipient.

8.4.4 Device address

When the AS bit in the IU header is set to one, bits 0-15 of word 1 shall contain the address of the device for which this exchange is being initiated. The device address specified shall correspond to a device configured to the control-unit image specified by the control-unit image ID; otherwise, an address exception condition shall be recognized.

Bits 0-7 of the device address shall be set to zeros by the source of an IU. If bits 0-7 of the device address are not set to zeros when the AS bit is set to one, a device-level protocol error shall be recognized. When the AS bit is set to zero, the 16-bit device-address field shall be set to zero by the sender of the IU and is ignored by the recipient of the IU.

Only one device address shall be used during a single connection. Once a device address has been identified for a connection, the use of a different device address during this same connection shall result in a device-level protocol error being recognized.

If a control unit receives an IU with the AS bit set to one and a device address of a not-ready device, it shall either perform the specified function and provide the appropriate response, if any, or, when the specified function requires a ready device, generate unit-check status; the sense data associated with the unit check shall indicate intervention required. If a control unit receives an IU with the AS bit set to one and the device address of an uninstalled device, it shall generate either an address-exception condition or, optionally, unit-check status. (See 8.11.2.10.)

The manner in which device addresses are assigned is model dependent.

8.5 IU header

8.5.1 IU header format

The information unit (IU) header is eight bytes which immediately follow every SB-2 header and precedes the DIB header. The IU header, shown in Figure 14, provides SB-2 control flags and necessary information in order to associate an IU to a specific CCW.

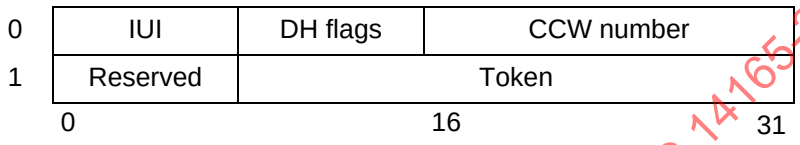

Figure 14 – IU header

The IU header consists of five fields, the information-unit identifier (IUI) field, device header (DH) flags field, CCW number field, a reserved field, and the token field. Byte 0 of word 1 shall be reserved and set to zero by the sender and ignored by the recipient.

8.5.2 Information-unit identifier

8.5.2.1 Information-unit identifier format

Byte 0 of word 0 of the IU header is the information-unit identifier (IUI). The IU type and the functions that affect field formats and interpretation are identified by bits in the IUI, which have the format shown in Figure 15.

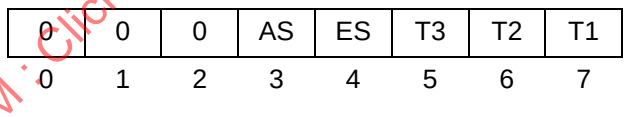

Figure 15 – IU identifier

Bits 0-2 of the IUI shall be reserved and are set to zeros by the sender for all IU types and ignored by the recipient.

8.5.2.2 Address specific (AS)

The AS bit, bit 3 of the IUI, when set to one, shall indicate that the IU is associated with the specific device identified by the device-address field of the SB-2 header. When the AS bit is set to zero, the IU shall not be associated with a specific device, and the device address shall not be used.

For a data IU containing a data DIB or command IU, the AS bit shall be set to one; otherwise, a device-level protocol error shall be detected.

For a data IU containing a status DIB, the AS bit may be set to either one or zero, depending on whether the status is associated with the device or the control unit. (See Table 18.)

For a control IU, the AS bit may be either one or zero, depending on the particular control function. (See 8.11, Table 18, and 6.4.)

8.5.2.3 Supplemental status (ES)

The ES bit, bit 4 of the IUI, may be set to either one or zero. The meaning of the bit when set to one depends on the IU type.

For a data IU containing a data DIB or a command IU the ES bit shall be set to zero by the sender and ignored by the recipient.

For a data IU containing a status DIB, the ES bit may be set to either one or zero. If the ES bit is set to one, supplemental status, sometimes referred to as extended status, shall be present in the supplemental status field. If the ES bit is set to zero, the length of the supplemental status field shall be zero. The ES bit may be set to one only when the status DIB contains unit check status.

For a control IU specifying the status-accepted function, the ES bit may be set to either one or zero. If the ES bit is set to one, the channel shall be indicating that it has accepted the supplemental status that was presented along with the status byte. If the ES bit is set to zero in a status-accepted IU that is sent in response to a status DIB with the ES bit set to a one, it shall indicate that the status has been accepted, but the supplemental status has not been accepted. (See 9.1.5.2.) The ES bit shall be set to one in a status-accepted IU only in direct response to status with the ES bit set to one. For all other control functions, the ES bit shall be set to zero by the sender and ignored by the recipient.

8.5.2.4 DIB type (T3,T2,T1)

Bits 5-7 of the IUI define the type of DIB contained in the IU. The DIB type also determines the format of the DIB. Table 8 summarizes the setting of the T3-T1 bits.

Table 8 – DIB-type settings

T3	T2	T1	DIB Type	DIB Format
0	0	0	Data	Data
0	0	1	Command	Command header
0	1	0	Status	Flags, status, count, supplemental status
0	1	1	Control	Control function and parameters
1	0	0	Command-data	Command header and data
1	0	1	Link control	Link-control functions
1	1	0	Reserved	
1	1	1	Reserved	

A data DIB shall be sent only in a data IU and shall be received by either the channel or control unit.

A command DIB shall be sent only in a command IU and shall only be received by a control unit. If a channel receives a command DIB, a device-level protocol error shall be detected.

A command-data DIB shall be sent only in a command IU and shall only be received by a control unit If a channel receives a command-data DIB, a device-level protocol error shall be detected.

A status DIB shall be sent only in a data IU and shall only be received by the channel. If a control unit receives a status DIB, a device-level protocol error shall be detected.

A control DIB shall be sent only in a control IU and shall be received by the channel or control unit.

A link-control DIB shall be sent only in a control IU and shall be received by the channel or control unit.

8.5.3 Device-header flags

8.5.3.1 Format of device-header flags

Byte 1 of word 0 of the IU header is the device-header flags (DHF) field. The device-header flag bits are used individually or collectively to invoke certain SB-2 protocols to control the execution of an I/O operation.

The format of the device-header flags field and the description of each of its flag bits follow. Bits 1,2,6, and 7 of the device-header flag field shall be reserved and are set to zeros by the sender and ignored by the recipient. The format of the device-header flags field is shown in Figure 16.

E	0	0	CH	EE	CNP	0	0
0	1	2	3	4	5	6	7

Figure 16 – Device-header flags

8.5.3.2 End (E)

The E bit, bit 0, when set to one for either a command-data DIB or data DIB, shall be used to indicate that the data sent exactly satisfies the CCW count and that the CRC field is present in the DIB. Also, the E bit, when set to one for a status DIB, shall be used to indicate that the quantity of data sent by the channel with an early end indication exactly satisfied the record length. (See 8.5.3.4.) The CCW number and token in the IU header, collectively and in a model dependent manner, identify the CCW to which the E bit applies.

For a data DIB, the E bit may be set to either one or zero. If the E bit is set to one, the CRC field shall be present in the data DIB and the total quantity of data sent for the CCW shall exactly satisfy the CCW count. If the E bit is set to zero, and the data sent for the CCW does not exactly satisfy the CCW count, the presence of the CRC field shall be determined by the EE bit.

For a command-data DIB, the E bit may be set to either one or zero. If the E bit is set to one, the CRC field shall be present in the command-data DIB and the data in the IU exactly shall satisfy the CCW count. If the E-bit is set to zero, and the data in the command-data DIB does not exactly satisfy the CCW count, the presence of the CRC field shall be determined by the EE bit.

If the DIB type being sent is a command DIB for a write operation and the CCW count is zero, the E bit shall be set to one. If the DIB type being sent is a command DIB for a read operation, the E bit shall be set to zero.

For a status DIB, the E bit may be set to one or zero. The E bit shall be set to one for a status DIB to indicate that for a write operation with an early end indication, the data received from the channel exactly satisfied the record length; otherwise, the E bit shall be set to zero.

For a control DIB, the E bit shall be set to zero by the sender and ignored by the recipient. (See 8.11 and Table 18.)

For a link-control DIB, the E-bit may be set to one or zero. The E bit shall be set to one when a link payload including the CRC field is present.

The presence of the CRC field does not mean that a valid CRC is provided. The CNP bit shall be checked to determine if a valid CRC is provided in the CRC field.

8.5.3.3 Chaining (CH)

The CH bit, bit 3, when set to one, is used by the channel to signal its intention to chain or to confirm that chaining is continuing.

For a data IU, the CH bit shall be set to zero by the sender and is ignored by the recipient.

For a command IU, the CH bit may be set to either one or zero. If the CH bit is set to one, the command IU shall be a command update as a result of either command chaining or data chaining and the DIB shall contain information from the current CCW. Whether the command update is for data chaining or command chaining shall be indicated by the DU bit in the command flag field. If the CH bit is set to zero, the command IU shall not be a command update. The CH bit shall be set to zero in a command IU initiating the channel program; otherwise, a device-level protocol error shall be detected. The CH bit shall be set to one in all remaining command IUs received during the execution of the channel program; otherwise, a device-level protocol error shall be detected.

For a control IU from the channel, the CH bit may be set to one only when the status-accepted function is indicated either during command chaining and the status does not include device end or when the status represents a retry request for an I/O operation and the status does not include device end. The CH bit shall be set to one by the channel when chaining is still indicated for the I/O operation and set to zero when chaining is not or no longer indicated for the I/O operation or when a retry request is not accepted. For all control IUs specifying other than the status accepted function, the CH bit shall be set to zero by the sender and ignored by the recipient.

8.5.3.4 Early End (EE)

The EE bit, bit 4, when set to one for either the control-end function in a control DIB, a data DIB, or a command-data DIB shall be used to indicate that the quantity of data sent for the CCW indicated by the CCW number is less than the CCW count specified for that CCW and for the data DIB and command-data DIB, the CRC field is present. (See 8.11.2.3 for additional information.) If the EE bit is set to zero, an early end condition is not being indicated and the presence of the CRC field shall be determined by the E bit.

When, for a write operation, the channel is able to send some, but not all, of the data specified for a CCW, the EE bit shall be set to one for either the command-data DIB or data DIB containing the CRC field. The CRC field may be present either in the same DIB containing data or in a data DIB containing no data immediately following the IU containing the last data bytes. If, for a CCW with a nonzero CCW count, the channel is unable to send any data, the EE bit shall be set to one for a command-data DIB which contains the CRC field. The CRC value sent shall be the initialized value of the CRC generator. See 8.6.5 for additional information.

When during data chaining the channel is unable to obtain the next CCW and perform the command update, the EE bit shall be set to one in a control-end IU in order to indicate that an early end condition was detected and the channel is unable to send or receive any more data for the current command.

When, for a read operation, the control unit is able to send some, but not all, of the data specified for a CCW, the EE bit shall be set to one in the data DIB containing the CRC field. The CRC field may be present either in the same DIB containing data or in a data DIB containing no data immediately following the IU containing the last data bytes.

Table 9 summarizes the use of the EE bit for CCWs with a nonzero CCW count:

Table 9 – EE-bit table

Conditions	Sent by	Number of Bytes sent	DIB type	EE	CRC field
All write CCWs. Some data sent	CH	< CCW count	Command-data or data	1	Yes
All write CCWs No data sent	CH	0 Bytes	Command-data	1	Yes
Data chain CCW - Read or write No command update sent	CH	0 Bytes	Control	1	No
All read CCWs Some data sent	CU	< CCW count	Data	1	Yes

When an early end indication is received by a control unit, status shall be returned at the completion of the operation for which the early end indication was sent.

When the Early End condition is recognized, the control unit shall disregard detection of an incorrect-length condition, if any, and shall depend on the channel to make the decision to continue chaining or not. (See 8.7.2.3 for additional information on an incorrect length condition.)

For a status DIB in a data IU and control IU specifying a control function other than the control-end function, the EE bit shall be set to zero by the sender and ignored by the recipient.

The EE bit and the E bit shall never be both set to one in the same IU and if both are found set to one in an IU for which both are meaningful, a device-level protocol error shall be recognized.

8.5.3.5 CRC not provided (CNP)

The CNP bit, bit 5, when set to one for a data, command-data, status or link-control DIB shall indicate that the value contained in the CRC field is not valid, and that CRC checking for the data associated with the CRC field should not be performed. The CNP bit shall only have meaning when the CRC field is present. (See 8.6.5.) When the CNP bit is set to zero, the CRC field shall contain a valid value, and if the receiver provides CRC checking, and the receiver's calculated value does not equal the received value, an SB-2 CRC error shall be detected.

8.5.4 CCW number

The CCW number, bytes 2 and 3 of word 0, of the IU header contains a model-dependent 16-bit binary value that is assigned by the channel to the CCW associated with the IU being sent.

A CCW number for a command IU initiating an I/O operation shall not be reused in a subsequent command IU until the command which initiated the I/O operation is no longer retryable. (See 9.3.2.) A CCW number for a command IU sent during data chaining and following a command IU initiating an I/O operation shall not be reused in a subsequent command IU until the channel has recognized that the CCW corresponding to this CCW number has been executed. This occurs when a data or control IU has been received indicating that execution of the channel program has progressed to a CCW contained in a subsequently transferred command IU. See 9.1.4.4 for additional information on data chaining.

The CCW number uniquely identifies a CCW specifying a retryable command or a CCW not specifying a command sent during data chaining which the channel has not yet recognized as executed.

The value hex'0000' is not a valid CCW number and shall not be used by the channel or control unit to identify a CCW. This value shall be used in those IUs that do not specify a specific CCW number. All values other than hex'0000' shall be valid for assignment by the channel to CCWs executed. Whenever a value other than hex'0000' is present in the CCW number field, a valid token shall also be present in the token field in word 1. (See 8.5.5.) If the channel receives an IU with the value hex'0000' in the CCW number field and that IU is required to have a valid CCW number, a device-level protocol error shall be recognized.

A control unit shall not check the CCW number.

Command IUs and data IUs containing a data DIB shall require a valid CCW number. The channel shall include the CCW number from the command IU in all data IUs sent to satisfy the CCW count specified by the command IU, and the control unit shall include this CCW number in all data IUs sent to satisfy the CCW count specified by the command IU.

Some IUs do not pertain to a CCW, and therefore, do not require a valid CCW number. For example, an unsolicited data IU containing a status DIB for asynchronous status or a link control request IU.

When a disconnection occurs as a result of the acceptance of status by the channel, all CCW numbers assigned to CCWs associated with the connection being removed shall become void, and as such shall be available for re-use by the channel. If the control unit subsequently reconnects to present status, a CCW number of hex'0000' shall be used. If a disconnection occurs as a result of an abnormal condition and a system reset or selective reset is not performed, the CCW numbers being maintained at the time of disconnection shall be preserved. The CCW numbers shall be preserved at least until the device-level recovery performed after the disconnection has completed and may remain in effect afterward, depending on the result of the recovery action.

The channel may indicate in a control IU the CCW number for which the control function is to be performed. When, for example, the channel sends a selective reset control IU with the RO bit, RU bit, or RC bit set to one, a valid CCW number shall be required if the request is for a specific command; otherwise, a CCW number of hex'0000' shall be used.

When the control function specified does not require a valid CCW number to be specified, the value hex'0000' shall be used. (See the individual control function descriptions for CCW number requirements.)

For a status DIB, a valid CCW number may or may not be required depending on the status being presented. If the status DIB is for the presentation of channel-end status during an existing connection, status in response to a command received in a command IU, or stacked or pending status with the busy indication in response to a command IU, the CCW number used shall be the CCW number from the command IU. If the status DIB is for the presentation of device-end status after a disconnection, the CCW number used shall be hex'0000'. If the status DIB is sent in direct response to a request-

status IU that is used to retrieve stacked status, the CCW number from the request-status IU shall be used. If the status DIB is in response to a channel initiated recovery procedure for a specific CCW number, then the CCW number from the recovery request from the channel shall be used. If a control unit sends retry status in an unsolicited data IU, the CCW number used shall be the CCW number from the command IU that is to be retried. For all other cases where a status DIB is sent, a valid CCW number shall not be required and the value hex'0000' shall be used.

8.5.5 Token

The token field, bytes 1-3 of word 1, of the IU header shall contain a 24-bit binary value that is provided by the channel. The value hex'000000' is not a valid token and shall be used only in those IUs that do not require a valid token. All other values of the token are valid and may be used by the channel. The method of determining the token value to be sent in an IU and its use by the channel is model dependent.

The token field shall contain a valid token, whenever the IU contains a valid CCW number. If the token field contains a value of all zeros, the CCW number is meaningless, and shall be set to all zeros by the sender and ignored by the receiver. When the token field contains a valid token, the CCW number field may or may not contain a valid CCW number, depending on other information associated with the IU.

A control unit shall not check the token.

A valid token shall always be sent in a command IU and data IU containing a data DIB. The control unit shall include the token from the command IU in all data IUs sent to satisfy the CCW count specified by the command IU. If the channel detects that a valid token is not present when it receives a data DIB, a device-level protocol error shall be recognized.

A valid token shall be sent by the channel in an unsolicited control IU. Examples are: system reset, cancel, and link control requests, ELP or TIN.

A solicited control IU, sent by a control unit shall contain the token from the IU to which it is being sent in response. A solicited control IU, sent by a channel shall contain a valid token.

A valid token shall be sent in a solicited data IU containing a status DIB. If the status DIB is for either the presentation of channel-end status during an existing connection, status in response to a command received in a command IU, or stacked or pending status with the busy indication in response to a command IU, the token used shall be the token from the command IU. Examples are: any status containing channel-end status and control unit busy status. If the channel receives a solicited data IU with a status DIB with a token that is not expected, or a token of all zeros, a device-level protocol error shall be recognized.

The invalid token (hex'000000') shall be sent in an unsolicited data IU containing a status DIB with asynchronous status and in an unsolicited control IU sent by a control unit. Examples are: asynchronous status presentation and the link-control request, TIN.

8.6 Device information block (DIB) structure

8.6.1 DIB structure overview

A DIB consists of a DIB header, LRC, and for some DIB types a DIB-data field. The DIB type is identified by bits in the IU header. Figure 17 shows the DIB structure common to all DIB types .

Figure 17 – Basic DIB structure

8.6.2 DIB header

8.6.2.1 DIB header format

The DIB header is the first 12 bytes of every DIB type and immediately follows the IU header. The contents of the DIB header and the DIB data field are determined by the DIB type. Figure 18 shows the basic DIB header structure common to all DIB types.

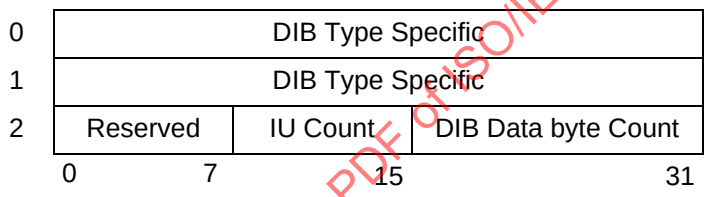

Figure 18 – DIB header structure

The format and contents of the DIB header fields are described in other sections in this clause. The following subclauses describe those fields in a DIB that are common to more than one DIB type.

8.6.2.2 IU count

Byte 1 of word 2, of each DIB header type shall contain the IU count. The IU count is an 8-bit binary number that shall be included in every IU sent during an open exchange.

The IU count value shall be initialized to hex'00' when an exchange is opened and the first IU sent. The value of zero shall be inserted in the IU count field of the DIB header for the sequence sent to open an exchange. After each IU is sent on a given exchange, the IU count shall be advanced by one and the new value included in the IU count field of the next IU sent. When the IU count reaches hex'FF' and additional sequences are to be sent, the IU count shall be permitted to wrap and the next value used is hex'00'.

When an exchange is closed, there is no further need for an IU count parameter to be maintained by the channel or control unit for that exchange.

If an IU is received with an IU count value that is not one greater than the IU count value received in the previous IU received for a given exchange, or, if the previous IU count was hex'FF' and the next IU count received is not hex'00', an IU count error shall be recognized and the IU shall be discarded. If an IU is received that opens an exchange, it shall have an IU count value of zero, otherwise an IU

count error shall be recognized and the IU shall be discarded. When an IU is discarded because of an IU count error, the recovery performed depends on the operation or function being performed. In some cases a retry of the operation may be attempted and in other cases the operation or function may be terminated without retry.

8.6.2.3 DIB data byte count

Bytes 2 and 3 of word 2, of the DIB header shall contain the DIB data byte count. The DIB data byte count is a 16 bit binary number that represents the number of bytes of data contained in the DIB-data field. Pad bytes and CRC shall not be included in the DIB data byte count. For a command DIB, the DIB data byte count shall be set to zero by the channel, and ignored by the control unit.

For a status DIB, the DIB data byte count shall be valid only when the ES status flag bit is set to one. When the ES bit is set to one, the DIB data byte count shall represent the number of bytes of supplemental status in the DIB data field. When the ES bit is set to zero, the DIB data byte count shall be ignored.

For a link control DIB, the DIB data byte count shall be valid only when the TIR link-control function is indicated. The DIB data byte count shall represent the number of bytes of link payload in the DIB data field.

For a control DIB, the DIB data byte count field shall be set to zero by the sender and ignored by the receiver.

8.6.3 Longitudinal-Redundancy-Check field

The longitudinal-redundancy-check (LRC) field shall contain a 32-bit redundancy-check code, immediately following the DIB header in bytes 28 to 31 of the IU. (See Figure 12.) LRC shall be provided only on the bytes contained in the SB-2 header, IU header, and DIB header.

The sender of an IU shall generate LRC and include the value calculated in the LRC field of that IU. The recipient of an IU shall also generate LRC and compare the value calculated to the LRC contained in the IU. If the value of the LRC contained in the IU equals the value calculated by the recipient, a valid LRC shall be recognized; otherwise, an invalid LRC shall be recognized. If a valid LRC is recognized, the contents of the SB-2 header, IU header, and DIB header shall be considered valid; otherwise, the contents of these headers shall be considered invalid and an LRC error shall be recognized.

See Annex C for a description of the procedure and an example of the LRC calculation.

8.6.4 DIB data field

8.6.4.1 DIB data field format

The DIB Data field, if present, is a variable length field which contains either data, data and pad bytes, data and a CRC field, data and pad bytes and a CRC field, or just a CRC field. The quantity of data contained in the DIB data field shall be indicated in the DIB data byte count field of the DIB header. A command DIB and a control DIB shall always be sent with a DIB data field of zero bytes. The maximum number of bytes that shall be sent in the DIB data field shall be 8160 bytes. The format of the DIB data field is shown in Figure 19.

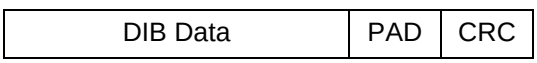

Figure 19 – DIB data field

When the data field contains the maximum of 8160 bytes of data, the PAD field and CRC field shall both contain 0 bytes. When the CRC field is present in the DIB, the maximum number of bytes in the DIB data field and PAD shall be 8 156. All DIB data fields sent for a CCW shall contain 8160 bytes with the possible exception of the last data DIB sent for the CCW. See 8.8 and 8.9 for additional information.

8.6.4.2 Pad

Pad bytes, if present, are contained in the last word of the data field and shall be used to pad the data field to the next word boundary.

For a command-data DIB or a data DIB, pad bytes shall be present only in the last IU containing data that is sent for a CCW and then only when boundary alignment is necessary. For a command IU with a command-data DIB, pad bytes shall be present only if all of the data to be sent for the current CCW is included and word alignment is necessary. For a status DIB with supplemental status and a link-control DIB with a link payload, pad bytes shall be present only if word alignment is necessary.

The value used for a pad byte is model dependent.

Pad bytes, if present, shall be included in the generation of CRC but shall not be included in the byte count for the DIB-data field.

8.6.5 Cyclic-Redundancy-Check field

When CRC generation is provided by the sender, the cyclic-redundancy-check (CRC) field, when present in an IU, shall contain a word-aligned 32-bit redundancy-check code, and the CNP bit shall be set to zero. When CRC is not provided, the word-aligned CRC field shall contain a model-dependent value, and the CNP bit shall be set to one. The requirements for the presence and location of the CRC field shall be the same, independent of whether CRC generation is provided or not.

The CRC field may be present in a command-data DIB, data DIB, status DIB, and link control DIB. For a command-data DIB and a data DIB, the CRC field shall be present only when either the E bit or the EE bit in the device-header flags field is set to one. For a status DIB, the CRC field shall be present only when the supplemental status (ES) bit is set to one. For a link control DIB containing a link payload, the CRC field shall be present and the E bit is set to one.

CRC shall be generated on data, supplemental status, and link payloads that are sent in the DIB-data field of an IU. When data is sent, CRC shall be generated on the data associated with a CCW. If more than one CCW is used to transfer the data associated with a command (data chaining), CRC shall be generated and sent with the data for each CCW, that is, CRC generation shall not span the data associated with more than one CCW. If it is possible to send the data associated with a CCW in one IU, then the CRC for that data shall be sent as the last four bytes of the DIB of that IU. If it is not possible to send the data associated with a CCW in one IU, then the CRC for that data shall be sent as the last four bytes of the last data IU required. In some cases it may be necessary to send an IU with a DIB containing only a DIB header and a CRC, as when either the remaining quantity of data for a CCW to be sent exceeds 8 156 bytes but is less than or equal to 8160 bytes, or an early end condition occurs and CRC is not included with the last data sent for the CCW.

When CRC generation is provided, the sender shall generate CRC on the data sent and insert the value calculated at the end of the last data DIB. When CRC checking is provided and the CNP bit is set to zero, the recipient shall generate CRC on the data received and compare the value calculated to the CRC contained in the IU. If the value of the CRC contained in the IU equals the value calculated by the recipient, a valid CRC check shall be recognized; otherwise, an SB-2 CRC error shall be rec-

ognized. If the receiver does not provide CRC checking or the CNP bit is set to one, the contents of the CRC field shall be ignored.

Whenever some, but not all, of the data from a CCW is used (such as when a cancel is received during an I/O operation), and CRC checking is provided, and the CNP bit is set to zero, that data shall be verified by checking the CRC for the entire CCW.

The processing of CRC, in terms of generation and checking, shall follow the equations in Annex N of FC-PH, reference [1]; however, the coefficients of the polynomial representing the DIB data field are not chosen according to bit transmission order specified in Annex N. The coefficient of the highest order term of F(x), which is the polynomial representing the DIB data field, shall represent the most-significant bit of the DIB data field; lower-order coefficients of the polynomial F(x) correspond to less-significant bits of the DIB data field in sequential order. The 32-bit CRC shall be the 32 coefficients of the frame check sequence (FCS) polynomial in Annex N, equation (1). The most significant bit of the CRC shall be the coefficient of the highest order term of the FCS polynomial; sequentially less-significant bits shall be the coefficients of sequentially lower order terms of the FCS polynomial. The CRC is transmitted on the link in the same bit-transmission order as all other words of the DIB data field. Additional information may be found in appendix B of reference [6].

When optional CRC generation initialization is not being used, the initialized value of the CRC generator shall be hex'FFFFFFFF'.

When optional CRC generation initialization is being used, the initialized value of the CRC generator shall be set to the modulo 2^{32} sum of the four 32-bit addends shown in Figure 20. The most significant bit of the initialized value of the CRC generator shall correspond to the most significant bit of the sum. (See note below.) The channel image ID, control unit image ID, S_ID, and D_ID are those being used in the exchange on which the data is sent. For a command-data DIB, a data DIB, and a status DIB, the device address is that being used on the exchange on which the data is sent and the CCW number is that of the command IU for which the data is being sent. For a link-control DIB, the device address and CCW number fields are set to zero.

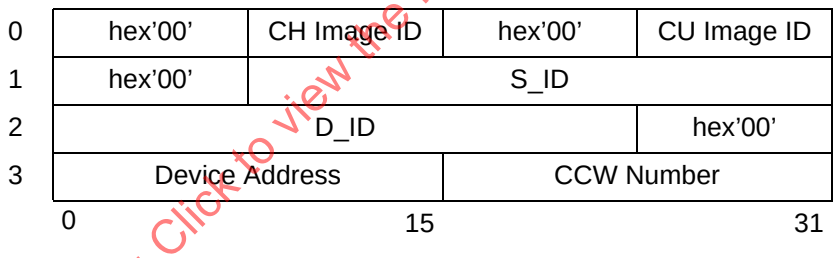

Figure 20 – Addends of the alternative initialized value of the CRC generator

The initialized value of the CRC generator shall be represented by the coefficients of the polynomial "L(x)" in equations (2) and (4), Annex N, FC-PH, reference [1]. When optional CRC generation initialization is being used, the bits of the modulo 2^{32} sum of the quantities in Figure 20 shall be used as coefficients in the polynomial L(x). The most significant bit (bit 0) of the modulo 2^{32} sum corresponds to the coefficient of X^{31}; lower order bits of the modulo 2^{32} sum correspond to lower-order coefficients in the polynomial L(x), respectively.

NOTE The preferred implementation is for CRC generation and checking to be provided. This provides end-to-end protection for data at the FC-4 processing layer. Optional alternative CRC generation initialization provides a higher level of protection by enabling the recipient of an IU to detect a larger percentage of possible errors caused by lost frames and possible errors caused by the association of incorrect header information with a block of data.

8.7 Command DIB structure

8.7.1 Command DIB overview

A command DIB shall be sent only in a command IU by the channel. It shall be used to transfer information associated with the current CCW to the control unit. At the beginning of an I/O operation, a command IU containing a command DIB may be used to initiate an operation with a device. When data chaining is performed for a read operation, a command IU containing a command DIB shall be used to update the information held about the current command at the control unit.

Each command DIB shall contain a command header and LRC. The structure of the command IU containing the command DIB is shown in Figure 12.

8.7.2 Command header

8.7.2.1 Command header format

The command header shall be the first 12 bytes of either a command DIB or command-data DIB. The command header shall contain both information from the current CCW and information supplied by the channel necessary for the current command to be executed. The command header describes how execution of the I/O operation being initiated is to be performed and may also specify how a subsequent I/O operation is to be performed.

The command header has the format shown in Figure 21.

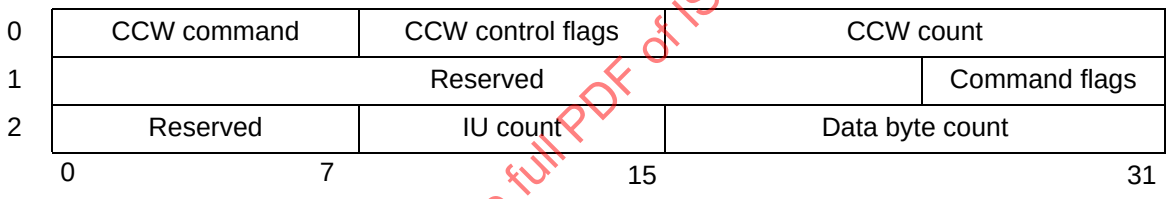

Figure 21 – Command header

Bytes 0-2 of word 1 and byte 0 of word 2 shall be reserved and set to zero by the channel and ignored by a control unit.

8.7.2.2 Command field

Byte 0 of word 0 of the command header shall contain the command specifying the I/O operation to be executed with a device. The basic operations shall be specified by the following commands: read, read backward, write, control, and sense.

The rightmost bit positions shall indicate the type of operation; the leftmost bit positions shall indicate a modification code which expands the meaning of the basic command that is to be executed. The modifier codes and the commands executed when they are decoded are model dependent.

Table 10 lists the basic commands and the bit settings of the command field. See 3.3.3 for a definition of the bit-numbering convention used in the table.

Table 10 – Contents of the command field

Command	Bit Position							
	0	1	2	3	4	5	6	7
Reserved	M	M	M	M	0	0	0	0
Sense	M	M	M	M	0	1	0	0
Reserved	M	M	M	M	1	0	0	0
Read Backward	M	M	M	M	1	1	0	0
Write	M	M	M	M	M	M	0	1
Read	M	M	M	M	M	M	1	0
Control	M	M	M	M	M	M	1	1
Note: M represents a modifier bit.								

When the command is initiating an I/O operation, the command field may contain any value that is not reserved. If a reserved command code is received by a device, the device shall respond with unit-check status, and the sense data associated with the unit check shall indicate command reject.

There are some commands (particular combinations of eight bits in the command field) which shall be executed by all devices. Table 11 indicates these required commands.

Table 11 – Required Commands

Command Code (hex)	Function
04	Basic Sense
03	No-operation (no-op)
md	Read configuration data
md	Read node identifier
E4	Sense ID
md	Set interface identifier
Explanation: md Model-dependent command code. The command code may be obtained by using a sense-ID command.	

When a command IU is used to update the count and flags during data chaining (the DU flag of the command DIB header is set to one), the contents of the command field is unpredictable and shall be ignored by the control unit receiving the command IU.

Some commands, when executed, do not result in the transfer of data but either cause the device to respond to the command with status which contains channel end, with or without device end (and without busy status) or cause the device to chain and start execution of the next command for the case where all of the conditions for not transferring status are satisfied. When this occurs, the I/O operation is referred to as an immediate operation. There are other situations in which an I/O operation may result in no data being transferred, for example, when a CCW count of zero has been validly specified. However, the operation shall not be an immediate operation if the CRR CCW control flag is not set for the command and the device responds to the command by sending a command response control IU. Each of the basic operations is described in the following section.

Read: A read command shall initiate execution of an I/O operation that performs device-to-channel data transfer. The bytes of data within a block shall be provided in the same order as those received by the write command.

Read Backward: A read-backward command shall initiate execution of an I/O operation in the same manner as the read command, except that bytes of data within a block shall be sent to the channel in an order which is the reverse of that used in writing. The bits within an 8-bit byte shall be in the same order as sent to the device on writing. The control unit may be designed to cause mechanical motion in the device in a direction opposite to that for a read command, or it may be designed to operate the device as it would for a read command.

Unless otherwise noted, any description that applies to read shall also apply to read backward.

Write: A write command shall initiate execution of an I/O operation that performs channel-to-device data transfer.

Control: A control command shall initiate execution of an I/O operation that performs channel-to-device data transfer. The device shall interpret the data as control information.

Sense: The sense command is similar to a read command, except that the data shall be obtained from sense indicators rather than from a record source.

8.7.2.3 CCW control flag field

Byte 1 of word 0 shall contain the CCW control flags. Bits 3 and 5-7 shall be reserved and shall be ignored by the control unit receiving the command IU. The value to which bits 3 and 5-7 are set is unpredictable. The format of the CCW control flag field shall be as shown in Figure 22.

CD	CC	SLI		CRR			
0	1	2	3	4	5	6	7

Figure 22 – CCW control flag field

Chain Data (CD): The CD flag, bit 0, when set to one, shall specify an intent to perform chaining of data. It shall cause the CCW control flags, CCW count, and command flags designated in the next command IU in which the DU flag and the CH bit are set to ones to be used with the current command. When the CD flag is set to one, the CC flag shall be ignored by the control unit. (See 9.1.4.4.) If a command IU with the DU flag set to one is followed by a command IU specifying a command, the CD flag shall be set to zero and the CC flag shall be set to one.

When data chaining occurs, a new set of CCW control flags, CCW count, and command flags shall go into effect for the current command.

It is model dependent whether the control unit permits data chaining for a particular command or for a particular device. When the control unit does not support data chaining or if the command is rejected for another reason, the control unit shall reject the command in the first command IU of a data chain (i.e the command IU with the CD flag set to one and the DU flag set to zero) with unit-check status. If the command is rejected due to data chaining, unit-check status shall be returned in a solicited data IU containing a status DIB sent either in direct response to the command IU or after sending a command response control IU. However, in either case, any data received for the rejected command shall be discarded by the control unit. When data chaining is not permitted, the sense data associated with the unit check shall indicate command reject along with a model-dependent indication that the unit check occurred because data chaining is not permitted for the command.

Chain Command (CC): The CC flag, bit 1, when set to one while the CD flag is set to zero, shall specify an intent to perform chaining of commands. Upon normal completion of the current I/O operation and after recognition of device end at the device, chaining of commands shall cause the command code specified in the command header of the next command IU to be initiated.

In the channel or in the control unit, when the CD flag is set to one, the CC flag shall be ignored.

When command-chaining occurs at the channel, the command, CCW control flags, and CCW count from the current CCW along with the command flags shall be transferred from the channel to the control unit in a command IU; the CH bit shall be set to one in the device header-flags field; and the DU flag shall be set to zero in the command-flag field of the command header. Other command flag bits may also be set to one.

In the case where command chaining at the channel is the direct result of having received normal status containing device end, the CH bit set to one and the DU flag set to zero shall indicate acceptance of this status by the channel.

When command chaining occurs at the control unit, the command, CCW control flags, CCW count, and command flags contained in the command header of the next command IU received, with the CH bit set to one and the DU bit set to zero, shall take control.

Suppress Length Indication (SLI): The SLI bit, bit 2, shall control whether or not command chaining is to occur on an incorrect-length condition, that is a condition recognized when the CCW count is not equal to the number of bytes requested or offered by the device. When the SLI bit is set to one and an incorrect-length condition exists for the current command, command chaining, if indicated, shall be permitted. If the SLI bit is set to zero and an incorrect-length condition exists for a command that is executed as a non-immediate operation, command chaining, if indicated, shall not be permitted and ending status with the residual count and appropriate status flag bits shall be transferred to the channel. If the SLI bit is set to zero and an incorrect-length condition exists for an immediate command, command chaining, if indicated, may or may not occur depending on the setting of the COC command flag bit. If the COC command flag bit is set to zero command chaining shall not be permitted and ending status with the residual count and appropriate status flag bits shall be transferred to the channel. (See 8.7.2.5.)

Command Response Request (CRR): The CRR flag, bit 4, when set to one for a command IU, other than the first command IU of a channel program, shall cause the control unit to indicate that it has started execution of that command by sending a command-response IU.

The CRR flag shall only have meaning when the CH bit in the DH_Flags field is set to one. The control unit shall ignore the CRR bit in the first command IU of a channel program, but not in subsequent command IUs that are command-chained or data-chained.

When a control unit starts execution of a command received as a result of command chaining or command retry and for which the CRR bit is set to one, a command-response IU shall be sent to the channel.

When both the CRR CCW control flag and the REX command flag are set to one, a control unit shall send only one command-response IU, independent of the number of times the command is executed.

The channel shall use the CRR flag in order to solicit a command response from the control unit when using the IU pacing function. (See 9.1.4.5.) The conditions under which the channel uses the CRR flag in this manner and the frequency it is used during the channel program are model dependent. The channel may also periodically set the CRR flag to one in a command IU other than the first command IU of a channel program for purposes of tracking the execution of the channel program. For example, when the channel is executing a CCW with the Program-Controlled Interruption (PCI) flag set to one, which is not the first command of a channel program, it shall set the CRR flag in the command IU.

8.7.2.4 CCW count field

Bytes 2 and 3 of word 0 shall contain the byte count specified in the current CCW.

The CCW count field is a 16-bit field that shall indicate the number of bytes to be transferred between the channel and control unit during execution of the CCW, not including the PAD and CRC bytes. (See 8.6.4.) The field shall be interpreted as a 16-bit unsigned binary integer. The value in the field may range from 0 to 65 535.

8.7.2.5 Command-Flag field

Byte 3 of word 1, shall contain the command flags. The channel shall use the command flags to provide the control unit with additional information on how the I/O operation, specified by the contents of the CCW in the command header, is to be executed.

Bits 0-2 shall be reserved, shall be set to zeros by the channel, and shall be ignored by a control unit. The format of the command flag field shall be as in Figure 23.

Reserved			DU	COC	SYR	REX	SSS
0	1	2	3	4	5	6	7

Figure 23 – Command-flag field

Data-Chaining Update (DU): The DU flag, bit 3, when set to one, shall indicate that the CC flag, the CD flag, and the count sent in this command IU are associated with a new CCW used during data chaining. The CH bit in the device-header-flags field shall be set to one when the DU flag is set to one.

Continue on Command Immediate (COC): The COC flag, bit 4, when set to one along with the CC flag also set to one shall indicate that suppression of the incorrect length condition is recognized by the channel and command chaining to the next command is permitted at the end of execution of a command immediate operation. When a command is executed as an immediate operation and the CCW count for the command is nonzero, an incorrect length condition shall be recognized and command chaining, if indicated, shall be under control of the COC flag or SLI flag. If the COC flag or the

SLI flag is set to one, command chaining, if indicated, shall be permitted. If the COC flag and the SLI flag are both set to zero, command chaining shall not be permitted. (See 8.7.2.3.)

Synchronize Response (SYR): The SYR flag, bit 5, when set to one, shall indicate to the control unit that the command IU contains a command being sent in response to status received for the immediately preceding command. The channel may set the SYR bit to one only when the CH bit is set to one and the DU bit is set to zero in the command header.

Repeat Execute (REX): The REX flag, bit 6, may be set to one only for an output operation and when the command and all of the data associated with the operation is contained in a single IU. When the REX flag is set to one, it shall indicate that the channel is requesting the control unit to perform the transfer-in-channel function by repetitively executing the command in the command header.

When the channel has fetched a CCW containing a command, all of the conditions for setting the REX bit to one are satisfied, and the next CCW (the one to which chaining occurs) contains a transfer-in-channel command with an address of the previous CCW, it is recommended that the channel set the REX bit to one in the command IU. Transfer-in-channel commands shall not be sent to the control unit.

When the control unit recognizes this indication, re-execution of the current command shall be performed until one of the following conditions occur.

a) Execution of the command results in status of other than channel end and device end alone being recognized.

b) Execution of the command has been performed 128 times.

c) Execution of the command is terminated by cancel, selective reset, or system reset.

The REX flag shall be set to one only when the CD, DU, and SSS flag bits in the command header are all set to zero. If a command IU is received with the REX bit set and any of the CD, DU, or SSS flag bits are set to one, a device-level protocol error shall be recognized.

When an execution of the command results in status of channel end, device end, and status modifier alone being recognized, chaining to the next command shall occur at the control unit.

When re-execution of the command is performed 128 times without status other than channel end and device end alone being recognized, re-execution of the command shall be terminated and the status recognized for the last execution of the command shall be sent to the channel.

When execution of the command results in status of other than channel end and device end alone or channel end, device end, and status modifier being recognized, re-execution of the command shall be terminated at the control unit and status shall be sent to the channel. (See 8.10.)

When the REX flag is set to zero, re-execution of the command is not requested.

If for certain commands the control unit does not support the repeat execute function, then it may always reject this command with unit-check status. If for other commands it does support the repeat execute function, it shall accept and execute the command with the REX bit set to one as required by the REX bit. If for all commands it does not support the repeat execute function, it may reject all commands with the REX bit set to one with unit-check status. Whenever a command is rejected with unit-check status because the REX bit is set to one, model-dependent sense data shall be provided.

Synchronize Send Status (SSS): The SSS flag, bit 7, when set to one, shall indicate that the channel has recognized a condition requiring synchronization with I/O execution at the control unit and is re-

questing that status be generated and presented at the completion of the I/O operation for this command IU. When the I/O operation has been concluded at the device, a status DIB containing device-end status is sent. When channel-end status is not accompanied by device-end status, a status DIB containing channel-end status is sent, and the control unit subsequently sends a status DIB containing device-end status when the I/O operation has been concluded at the device.

When conditions at the channel require synchronization with execution of a command at the control unit, the SSS flag bit shall be set to one in the command IU. If data chaining is indicated for the command, the SSS flag bit shall be set to one in the command IU sent for the last CCW providing data chaining.

When a command IU with the SSS flag bit is sent, command chaining at the channel shall be temporarily suspended. When chaining is suspended at the channel, execution of the channel program shall temporarily end with the execution of the last CCW for the current command, and until initiative to perform command chaining is provided by the receipt of status that allows chaining, command IUs and data IUs shall no longer be sent.

When a control unit completes the execution of a command for which the SSS flag is set, status shall be generated, and sent to the channel. If a command IU is received with both the SSS flag and REX flag set to one, a device-level protocol error shall be recognized.

8.8 Command-Data DIB Structure

A command-data DIB shall be sent only in a command IU by the channel. It shall be used to transfer both information associated with the current CCW for write commands with a nonzero CCW count and all, some portion or none of the data associated with the command. When data chaining is performed for a write operation, a command-data DIB shall be used to update the information held about the current command at the control unit.

A command-data DIB shall contain a command header, LRC, and a DIB-data field. (See Figure 12.) The format of the command header shall be the same as that of a command DIB. The DIB data field shall be a maximum of 8 160 bytes. If the CCW count for the command is 8 156 bytes or less, all of the data for the command shall be sent along with the CRC field in the DIB-data field. If the CCW count for the command is greater than 8 156 bytes, up to the first 8 160 bytes shall be sent in the DIB-data field, and the CRC field and remaining data, if present, shall be sent in one or more subsequent data IUs. If the CCW count for the command is greater than 8 156 bytes, and less than 8 160 bytes, pad bytes shall be added at the end of the command-data DIB, and the CRC field shall be sent in a subsequent data IU.

The DIB data byte count, bytes 2 and 3 of word 2, of the command header shall specify the number of data bytes contained in the DIB-data field. (See 8.6.2.3.)

8.9 Data DIB structure

8.9.1 Data DIB overview

A data DIB shall be sent only in a solicited data IU by either the channel or control unit.

For a write operation, the channel shall use one or more data IUs containing a data DIB to transfer any remaining portion of data not sent in the command-data DIB. When the quantity of data to be sent for a write operation exceeds the maximum number of bytes that is allowed to be sent in an IU, the channel shall send a command IU containing a command-data DIB for the command and the maximum number of bytes allowed; followed by one or more data IUs until either the quantity of data specified by the CCW count is satisfied or an early end condition is recognized.

For a read operation, the control unit shall use one or more data IUs containing a data DIB to transfer to the channel the data for the read command. When the quantity of data to be sent requires more than one data IU, the control unit shall send a data IU followed by one or more additional data IUs until one of the following occurs: 1) the quantity of data specified by the CCW count is satisfied, 2) an early end condition is recognized, or 3) a condition requiring status to be sent is recognized.

All data IUs contain the maximum number of bytes, 8 160, except the last data IU sent for a CCW which may have less than the maximum.

When the data DIB containing the last data byte for the current CCW contains more than 8 156 bytes of data, the CRC field shall not be included in the data DIB and shall be sent in a subsequent data DIB by itself. The data DIB containing greater than 8 156 bytes of CCW data shall have pad bytes added, if necessary, and the CRC value sent in the subsequent data IU (if generated) shall cover both the CCW data and the pad bytes. In the last data IU, containing only the 4-byte CRC field, the byte count shall be set to zero.

The CCW number and token provided in the IU header of each data IU sent shall be the same as that provided in the command IU.

8.9.2 Data Header

The data header is the first 12 bytes of a data DIB. The data header shall have the format shown in Figure 24.

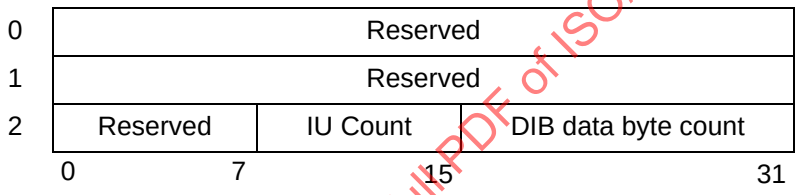

Figure 24 – Data header

Word 0, word 1, and byte 0 of word 2 shall be reserved and set to zero by the sender and ignored by the recipient.

8.10 Status DIB

8.10.1 Status DIB processing

A status DIB shall be sent in either a solicited data IU or unsolicited data IU by a control unit. It shall be used to transfer status, with or without supplemental status, to the channel. (See Figure 12.) A solicited data IU containing a status DIB may be sent in direct response to a command IU, or to certain initiation IUs when the request-status, selective-reset, or cancel control function is specified.

Additionally, a solicited data IU containing a status DIB may be sent during the data transfer portion of an exchange.

An unsolicited data IU containing a status DIB shall be used to present asynchronous status or any status that initiates a connection and is not sent in direct response to an IU received from the channel.

After sending a data IU containing a status DIB, the control unit shall discard all command IUs with the SYR flag set to zero and data IUs associated with that device address and logical path, either queued at the control unit, or received from the channel; discarding continues until a command IU with the SYR

flag set to one is recognized by the control unit for the device and logical path. Acceptance of the status by the channel may be indicated explicitly when a status-accepted IU is received or may occur implicitly when a command IU with the SYR bit in the command header set to one, is received.

When any of the following conditions exist, status shall be sent to the channel at the completion of the current I/O operation.

a) The SSS command flag bit is set to one.

b) An early end indication is received.

c) The CC flag bit is set to zero.

d) The first command of the channel program is executed as an immediate operation.

e) The CC flag bit is set to one and the SLI flag bit is set to zero and an incorrect length condition is detected for a command that is executed as a non-immediate operation.

f) The CC flag bit is set to one, both the SLI flag bit and COC command flag bit are set to zero, the CCW count is nonzero, and the command is executed as an immediate operation.

g) The REX bit is set to zero and status other than channel end (with or without device end) is recognized.

h) The REX bit is set to one and status other than channel end and device end, or channel end, device end and status modifier is recognized.

i) The REX bit is set to one, and the command has been executed 128 times.

j) Status of channel end (without device end) is recognized, and conditions at the control unit require disconnection from the channel to suspend the transferring of command IUs and data IUs.

If none of the above conditions exist, status shall not be sent and chaining, if indicated at the control unit, shall be permitted. (See Annex D.) Additionally, status shall not be sent if during an existing exchange any previously sent status has not yet been accepted by the channel.

For a Status DIB, the AS bit may be set to either zero or one depending on whether the status is associated with the control unit or a device. If the status contains one of the following combinations, the status is associated with the control unit, and the AS bit shall be set to zero.

a) Control Unit Busy. If a control unit receives an IU that initiates an exchange and a busy condition exists which prevents the acceptance of any SB-2 IU with the exception of a control IU indicating system reset, then the control unit shall send a solicited data IU with a status DIB with only the status-modifier and busy status bits set to ones and the AS bit set to zero. This status is referred to as control-unit-busy status. (See 9.2.10.)

b) Control-Unit End Alone. If a control unit does not recognize a stack status sent in response to a status DIB containing control-unit-busy status, the control unit shall later send a data IU containing a status DIB with the control-unit-end status bit set to one. When all other status bits are set to zeros, the AS bit is set to zero, and the status is referred to as control-unit end alone.

c) Control-Unit End with Busy. When control-unit-end status is pending and the control unit recognizes a valid command IU that initiates an exchange pair, the control unit may return a solicited

data IU with a status DIB with only the control-unit-end and busy status bits set to ones and the AS bit set to zero. This status combination is referred to as control-unit end with busy.

d) Control-Unit End with Control Unit Busy. If a control unit receives an IU that initiates an exchange pair and the IU is not a control IU indicating system reset, the control unit may return a solicited data IU with a status DIB and with only the status-modifier, control-unit-end, and busy status bits set to ones, and the AS bit set to zero. This status combination is referred to as control-unit end with control unit busy. When the channel recognizes this combination of status bits, shall consider the status to be an indication that the control unit is not busy.

In all other cases, the status shall be associated with a device address, and the AS bit shall be set to one; if the AS bit is set to zero with any status combination not listed above, the channel shall detect a device-level protocol error.

A data IU with the AS bit set to zero containing a status DIB shall be sent only if the status is one of the preceding combinations and meets one of the following conditions.

a) The status is control-unit end alone and is to be sent in an unsolicited data IU.

b) The status is in response to a valid request-status IU that initiated a connection, and the status byte is not control-unit end with busy.

c) The status is in response to a valid command IU that initiated a connection, and the status is not control-unit end alone.

d) The status is in response to a valid IU other than a system-reset IU, that initiated a connection, and the status is control unit busy.

The channel shall detect a device-level protocol error if a data IU containing a status DIB is received that has the AS bit set to zero and does not meet one of the preceding conditions.

NOTE 1 Control units and devices should provide interlocks so that status is not lost, hidden, or included with other status when the result would cause the program or channel to misinterpret the original meaning and intent of the status.

NOTE 2 The protocol prohibits the sending of status except under the the conditions listed in order to minimize the performance impacts caused when status is presented.

8.10.2 Status DIB structure

The format of the data IU for a status DIB is shown in Figure 12. A status DIB shall consist of a 12-byte status header, 4-byte LRC, and when supplemental status is included, from 1 to 32 bytes of sense data (and, if necessary, pad bytes) and the 4-byte CRC field. When supplemental status is not included in the status DIB, the status DIB shall consist only of the status header and the 4-byte LRC.

8.10.3 Status header

8.10.3.1 Status header format

The status header is the first 12 bytes of a status DIB. The status header shall have the format shown in Figure 25.

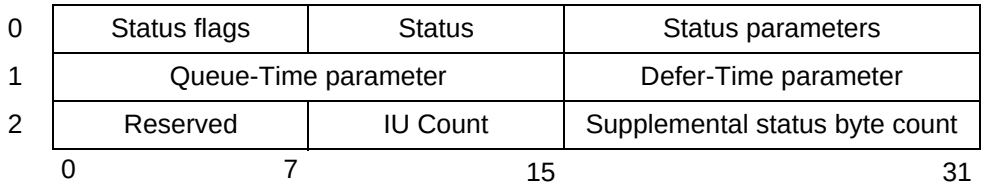

Figure 25 – Status header

Byte 0 of word 2 shall be reserved and set to zero by a control unit and ignored by the channel.

8.10.3.2 Status-Flags field

Byte 0 of word 0 of the status header shall contain the status flags. The status-flag field shall be used to provide additional information to the channel concerning the conditions that were present at the control unit when status was generated and conditions that pertain to the status DIB. These flags shall assist the channel in determining how to handle the status DIB and what status, if any, to report. The status-flag field shall have the format shown in Figure 26.

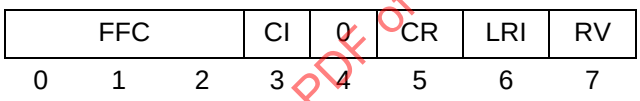

Figure 26 – Status-flag field

Flag-Field Code (FFC): The FFC, bits 0-2, is a three-bit encoded field that either in conjunction with or independent of the other status flag bits shall further describe the status information contained in the status byte, the status-parameter field, and the queueing-information fields. The flag-field code assignments shall be:

000 No Function. The status byte, status-parameter field, and queueing information fields are not affected by this FFC code.

001 Queueing Information Valid. The queue-time-parameter field contains control-unit queueing information. This code is set by the control unit and checked by the channel.

010 Resetting Event. A resetting-event condition exists for the logical path and the device for which unit-check status is being presented. This code shall be permitted only when unit-check status is presented for a resetting-event condition and then only as initial status in response to a command IU for the first command of the channel program; if this code is indicated at any other time, the channel shall detect a device-level protocol error. (See 9.4.)

011-111Reserved.

Channel Initiated (CI): The CI bit, bit 3, when set to one, shall indicate that this solicited data IU containing a status DIB is in direct response to a control IU indicating selective reset with either a request for retry or a request for unit check. (See 9.3.3.2 and 9.3.3.3.) If the CI bit is set to zero, this data IU

containing a status DIB is not in direct response to a control IU indicating selective reset with a request for retry or a request for unit check.

Reserved: Bit 4 shall be reserved.

Command Retry (CR): The CR bit, bit 5, when set to one, shall be used to request command retry if the status byte also contains retry status. If the CR bit is set to zero or the status byte does not contain retry status, command retry is not being requested. The status byte shall contain retry status if it contains unit check and status modifier together with 1) channel end alone (meaning the control unit or device is not yet ready to retry the command), or 2) channel end and device end (meaning the control unit and device are prepared for immediate command retry). All other status combinations shall not be considered retry status; if the CR bit is set to one with any other status combination, a device-level protocol error shall be detected.

Long Record/Immediate (LRI): The LRI bit, bit 6, when set to one, shall indicate that either a long-record condition was detected by the control unit or that the status being presented with a residual count equal to the CCW count is associated with a command that was executed as an immediate operation.

When the control unit detects a long-record condition (that is, additional data would have been sent to the channel or written to the device beyond the data provided for the current command), the LRI and RV bits shall be set to one and the residual count shall be set to zero. When data chaining is not used, the amount of data provided for the current command is equal to the count provided in the CCW. When data chaining is used, the amount of data provided for the current command is equal to the sum of the counts of all CCWs used for the current command.

When the control unit executes a command, that is not the first command of a channel program, as an immediate operation and the CCW count is greater than zero, the LRI and RV bits shall be set to one and the residual count shall be set to the value of the CCW count. When the control unit executes the first command of a channel program as an immediate operation, and the CCW count is greater than zero, the LRI bit is meaningless, may be set to zero or one, and shall not be checked by the channel.

The LRI bit may be set to one by the control unit only when the channel-end status bit is set to one. If the channel-end status bit is set to zero in a status DIB, the control unit shall set the LRI bit to zero, and the LRI bit shall be ignored by the channel.

See Table 12 for a summary of the usage of the LRI bit.

Residual-Count Valid (RV): The RV bit, bit 7, when set to one, shall indicate that the status-parameter field contains the residual count. For write commands, the residual count is equal to the difference between the CCW count for the write command and the number of bytes actually written to the device. For read commands, the residual count is the difference between the CCW count and the number of bytes actually read from the device and transferred to the channel. The RV bit may be set to one by the control unit only when the channel-end status bit is set to one. If the channel-end status bit is set to zero, the control unit shall set the RV bit to zero and the RV bit is ignored by the channel.

When status is sent in response to the first command of a channel program that is executed as an immediate operation, the RV bit shall be set to zero.

When status with channel end is sent in response to a command other than the first command of a channel program that is executed as an immediate operation, the RV bit shall be set to one unless an abnormal condition exists that precludes calculation or transfer of the residual count, in which case the RV bit shall be set to zero and unit check shall be indicated in the status byte.

When the LRI bit is meaningful and set to one, the RV bit shall be set to one. (See 8.10.3.2.)

When the RV bit is set to zero, both the E and LRI bits shall not be meaningful, and the status-parameter field shall not contain a residual count. (See 8.10.3.4.)

Table 12 summarizes the use of the RV and LRI status flags and the channel-end status bit. In addition to these bits, the E bit in the device-header-flags field shall be set to one if during a write command an early end condition was indicated and the data sent by the channel is equal to the record length. In all other cases, the E bit is set to zero for a status DIB.

Table 12 – LRI and RV bit usage

Condition	CH END	LRI	RV	Status parameter field
CCW count = data read or written	1	0	1	Residual count = 0
CCW count < # bytes in record	1	1	1	(Note 1)
READ command: CCW count > # bytes read and transferred	1	0	1	CCW count minus # bytes read & transferred
WRITE command: CCW count > # bytes written	1	Note 2	1	CCW count minus # bytes written
Non-Immediate Command: CCW count > 0	1	Note 2	1	CCW count (no data transferred)
Immediate command (not first) CCW count > 0	1	1	1	CCW count
Immediate command (first) CCW count > 0	1	X	0	IU pacing parameter (Note 3)

NOTE 1 If the channel indicates early end, the residual count = CCW count minus # bytes written, otherwise the status parameter field is set to zero.
NOTE 2 If the channel indicates early end, the LRI bit is set to one if the CCW count < # bytes in record, otherwise the LRI bit is set to zero.
NOTE 3 See 8.10.3.4 for additional information.
NOTE 4 X = don't care.

8.10.3.3 Status byte

Byte 1 of word 0 shall contain the status byte. The status byte indicates device and control unit status. The status byte shall have the format shown in Table 13. See 3.3.3 for a definition of the bit-numbering convention used in Table 13.

Table 13 – Status byte

Status bit position	Description
0	Attention
1	Status modifier
2	Control-unit end
3	Busy
4	Channel end
5	Device end
6	Unit check
7	Unit exception

A brief summary of each of these status bits follows.

If the channel receives a status DIB with the status byte set to zero, a device-level protocol error shall be detected. The status DIB shall contain a status byte with a combination of status bits set to ones which is appropriate for the conditions existing when the status is presented. If the combination of status bits is not appropriate for the existing conditions, the channel may detect a device-level protocol error.

The channel shall indicate acceptance of the status by means of either a status-accepted control IU, or, when retry or chaining is performed and the status includes device end, by means of a command IU having the CH bit set to one.

See 8.10 for the situations in which the following status conditions are sent to the channel.

Attention: The attention condition shall be generated when some asynchronous condition occurs in the control unit or device. The condition may be accompanied by other status. Attention shall not be associated with the initiation, execution, or termination of any I/O operation.

Status Modifier: Status modifier shall be generated by the device when the device is not able to provide its current status in response to interrogation by the channel subsystem, when the control unit is busy, when the normal order of commands has to be modified, or when command retry is to be initiated.

Control-Unit End: Only control units that indicate a control-unit-busy condition shall indicate a control-unit-end condition. Control-unit-end status shall be returned from the control unit to the channel after the channel accepts control-unit-busy status from the control unit. Control-unit-end status shall be returned after the control-unit-busy condition no longer exists. This is sometimes referred to as no-longer-busy status for the control unit. Only one control-unit-end indication shall be returned on a log-

ical path, regardless of the number of times the channel accepted control-unit-busy status on that logical path during the busy period.

The control unit shall not associate pending control-unit-end status with any device address. A selective reset shall not reset pending control-unit-end status. If the channel stacks or does not accept a status byte that contains the control-unit-end status bit, the control-unit-end status shall not be held along with the status byte, and it shall remain pending and unstacked at the control unit. Control-unit-end status may be withdrawn by the control unit if the control unit becomes busy again before the status is accepted by the channel; in such a case, the control unit shall return control-unit-end status later, after the control-unit-busy condition no longer exists.

When control-unit end is included with other status bits set, other than those status combination required to have the AS bit set to zero, the AS bit shall be set to one, and the device address used shall be the device address for which the other status is being sent. (See 8.10.) A pending control-unit end shall not necessarily preclude initiation of new operations. Whether the control unit allows initiation of new operations is at the option of the control unit.

Control-unit end shall not necessarily cause command chaining to be suppressed. Control-unit end shall not cause command chaining to be suppressed when presented with the AS bit set to zero as described in 8.10.

When control-unit-end status is presented on a logical path along with status modifier and busy, with no other status bits set to one, the combination should be interpreted as control-unit-end status. When this status is accepted, the control unit shall no longer owe a control-unit-end status indication on that logical path. If the control unit was interrogated while it was in the busy state, and then system reset is recognized by the control unit before control-unit-end status is accepted by the channel, the control unit shall not owe control-unit-end status.

A control unit shall only present control-unit-end status when a no-longer-busy condition is owed. However, the channel shall not detect an error if control-unit-end status is received when no control-unit-busy condition was indicated (that is, a no-longer-busy condition was not owed).

NOTE Presentation of the control-unit-end status bit without any other status bits is the preferred implementation.

Busy: The busy indication shall occur only when conditions existing at the device or control unit preclude execution of the intended I/O operation because of one of the following four situations.

a) A previous I/O operation or chain of I/O operations is being initiated or being executed.

b) Stacked or pending status conditions exist, and the pending status conditions are returned in response to a command IU, busy is appended to the status returned.

c) The control unit is shared by channels or devices, or a device is shared by control units, and the shared facility is not available.

d) A self-initiated function (for example, microdiagnostics or data movement internal to the device) is being performed.

Status conditions for the addressed device, if any, shall accompany the busy indication.

Channel End: Channel end shall indicate that the portion of an I/O operation involving transfer of data or control information between the channel and a control unit has been completed. Acceptance of channel-end status shall indicate the completion of the channel portion of the I/O operation.

Each I/O operation initiated at the device shall cause one and only one channel end condition to be recognized at the control unit, for which channel-end status may or may not be generated and presented to the channel.

Device End: Device end shall be recognized at the control unit (1) when the device portion of an I/O operation is completed, (2) when the device, having previously responded busy, transitions from the busy to the not-busy state, (3) when the device changes from the not-ready to the ready state, and (4) when the control unit or device recognizes an asynchronous condition.

Unit Check: Unit check shall indicate that the device or control unit has detected an unusual condition that is detailed by sense information. (See 8.10.4.2.) The occurrence of unit check may indicate that a programming error or an equipment error has been detected, that the not-ready state of the device has affected the execution of the command, or that an exception condition other than the one identified by unit exception has occurred. The unit-check bit provides a summary indication of the conditions identified by sense data.

Unit Exception: Unit exception shall mean that the device detected an unusual condition that needs to be reported to the program, such as recognition of a tape mark, and does not necessarily indicate an error.

8.10.3.4 Status-Parameter field

The status-parameter field, bytes 2 and 3 of word 0 of the status header, is a 16-bit field that may contain a residual count or IU pacing parameter. The IU pacing parameter is provided in the status-parameter field when status is presented for the first command of a channel program which is executed as an immediate operation or when presenting device end in order to reconnect when the chaining condition is set. If the conditions are such that neither the residual count nor IU pacing parameter is to be presented, the control unit shall set the status-parameter field to zero, and the channel receiving the status DIB shall ignore the status-parameter field.

Residual Count: The residual count is a 16-bit unsigned binary number that represents the difference between the CCW count for a command and the quantity of data actually transferred either from or to the device for that command.

For write commands, the residual count shall be equal to the difference between the CCW count of the current CCW for the command and the number of bytes actually written to the device. For read commands, the residual count shall be the difference between the CCW count for the current CCW for the command, and the number of bytes actually read from the device and transferred to the channel. The residual count shall be meaningful only when the residual-count-valid (RV) bit is one.

IU Pacing Parameter: The IU pacing parameter is an eight bit value that is carried in the least-significant byte of the status-parameter field.

The IU pacing parameter shall be sent by the control unit to indicate to the channel the maximum number of IUs a channel may send on a given outbound exchange until it receives a command response IU, which was sent because the CRR bit was set to one, on the existing inbound exchange. An IU pacing parameter of zero indicates that the control unit wishes to leave the default value of the IU pacing credit unchanged. See 9.1.4.5 for additional information.

8.10.3.5 Queue-Time parameter

When control-unit queueing is provided, bytes 0 and 1 of word 1 contain the queue-time parameter (QTP). (See Figure 27.) Queue time represents the time the I/O operation is queued at the control unit. When control unit queueing is provided, I/O operations specified by the channel program shall be executed up to a point where the I/O resources are unavailable to execute an I/O operation in that channel program. (Such resources would typically be in use executing I/O operations specified in other channel programs.) Depending upon the control unit design, queueing of a command may occur after receipt of the first command or after receipt of a subsequent command within the channel program. I/O operations executed up to and including the command to be queued may involve data transfer as a normal course of I/O operation execution. In all cases, queueing of a command for the channel program shall not affect ongoing I/O operations or cause the sequential order of I/O operation execution to be altered. Queueing shall occur only once within execution of the channel program and may occur only between execution of successive I/O operations; however, the point in execution chosen and the length of the queueing time are model dependent. When a command is queued, it shall remain queued at the control unit until either the needed resources become available or when a system reset or selective reset has been executed.

When queueing occurs, the control unit shall either perform a disconnection from the channel by presenting status containing channel end without device end (this status may or may not include command retry status), or temporarily delay, to a later time, further channel program execution without causing a disconnection from the channel. In the former case, execution of the command shall be completed at the channel when the control unit performs the disconnection by presenting channel-end status. When the unavailable resources become available, the queue interval shall be ended when the command is dequeued. The specified I/O operation shall be executed and shall be completed at the device. Connection with the channel occurs and device-end status shall be presented, along with the queue-time parameter. The control unit shall retain the queue-time parameter until it receives acknowledgment that the status transfer was accepted. At this time command chaining, if specified, may occur which allows execution of the channel program to continue. In the latter case, when disconnection from the channel does not occur, the queueing interval shall be the same duration as would have been accumulated had disconnection and subsequent reconnection to the channel occurred. In this situation, the accumulated queueing information shall be held pending at the control unit until either execution of the channel program has completed at the device (device-end status is presented) or a later disconnection and reconnection has occurred and device-end status is presented. In all cases, the control unit shall retain the queueing information until it regards the status as accepted. When this has occurred, all previously stored or accumulated values shall be zeroed at the control unit.

The contents of the queue-time parameter field shall be meaningful only when the flag-field code is set to ‘001’b and the device-end status bit is set to one in the accompanying status byte.

When the queue-time parameter field contains valid information, it shall be obtained in the following manner with the format specified in Figure 27.

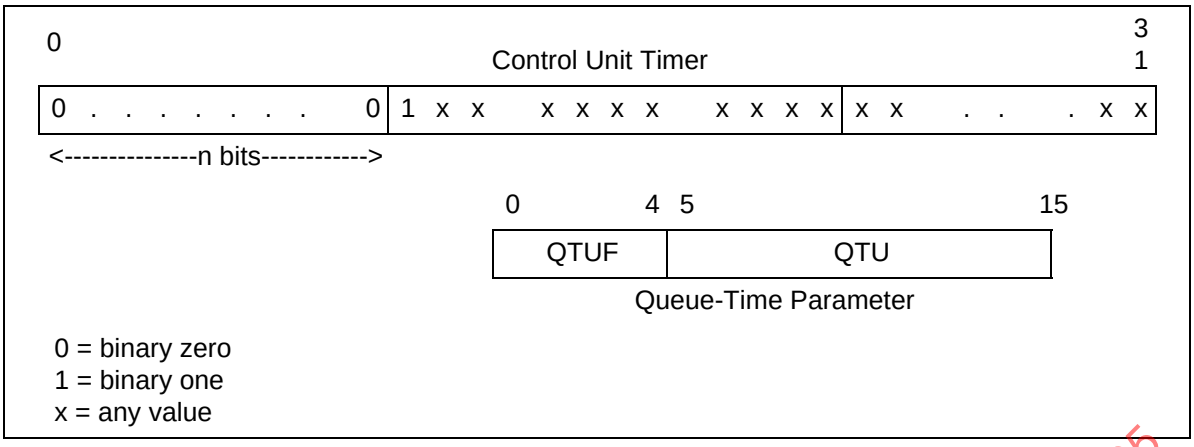

Figure 27 – Queue-time parameter (QTP) format

The queue-time parameter shall be calculated by the control unit using a 4-byte timer having a 1 µs resolution. The value accumulated by the 4-byte control unit timer shall specify the queue-time unit to be stored in the queue-time parameter field.

Queue Time Unit Factor (QTUF): Bits 0-4, when non-zero, shall specify the number of bit positions the 11-bit queue-time unit shall be shifted left for interpretation by the channel. If a shift is required, the QTUF shall be calculated by subtracting the value of the highest bit position, which was set to one, of the queue-time unit from 21. (This applies only in those cases where the bit position is equal to or less than 21.) If no adjustment is required, the QTUF value shall be '00000'b.

Queueing Time Unit (QTU): Bits 5-15 shall contain the 11-bit queueing time unit accumulated by the 4-byte control unit timer. The bit significance of the time unit shall be specified by the QTUF.

Table 14 describes the relationship between the values of QTUF and the bit-significance of QTU.

Table 14 – QTF/QTU relationship

When QTUF is:	Bit 15 of QTP represents:
00000	1 µs
00001	2 µs
00010	4 µs
00011	8 µs
.	.
.	.
.	.
10101	2 097 152 µs

In the example given in Figure 28, the control unit timer has accumulated a queue time of 75 000 µs (hex'124F8') and calculated the queue-time parameter. When received by the channel, the QTUF in-

dicates a shift of the most significant bit of the QTU to bit position 15 of the channel 4-byte-time value, and bit 15 of the QTU represents a value of 64 µs. The low-order 6-bit positions are set to zeros. See 3.3.3 for a definition of the bit-numbering convention used in the Figure 28.

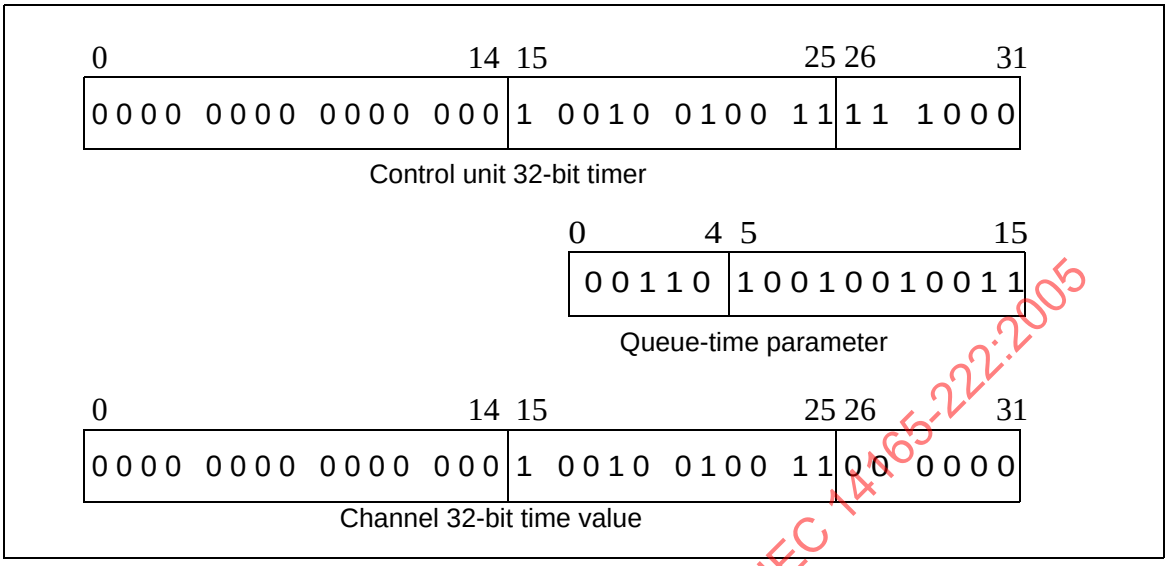

Figure 28 – Queue-time parameter example

A queue time of zero shall be indicated by setting the QTUF to ‘00000’b and the queueing-time unit to ‘00000000000’b.

A control unit that performs queueing shall have a queue-timing facility which is capable of measuring a time interval of up to its expected maximum queueing time, not exceeding 4 294,967 295 s.

When a control unit converts from a 4-byte timer value to a queue-time unit, if any of the bits of the 4-byte timer value are lost while shifting, the least significant bits shall be discarded (truncated).

Whenever the queue-time-unit factor is value ‘00001’b - ‘11111’b and the queue-time unit is zero, the resultant value of the queue-time parameter (for example, the values hex’0800, hex’1000’, hex’4000, hex’8000, hex’C000’, etc.) shall be not meaningful. This provides for usage of these values to indicate unique conditions recognized by the control unit. In particular, the value hex’8000’ (the QTUF is set to ‘10000’b and the QTU is set to ‘00000000000’b), and the queueing-information-valid code in the flag-field code shall indicate the control unit is not capable of presenting meaningful queue-time as a result of either a queue-timing-facility failure or an overflow condition.

If a queue-timing facility failure has occurred, the control unit shall send a status DIB with unit check status and sense information describing the failure to allow device-dependent program recovery. If a queue-time overflow condition has been recognized, unit check status shall not be presented to the channel. When a queue-timing facility failure has been recognized and after unit-check status has been presented and accepted, the control unit may either

– disable its queue-timing facility and process subsequent I/O operations without queueing until this failure condition is corrected, or

– continue queueing the subsequent I/O operations and present an indication of meaningless queue-time by setting the flag-field code in the status flag field of the status header to ‘001’b and setting the queue-time parameter to the value hex’8000’.

(In the latter case, unit-check status shall not be presented after the initial indication of queueing information not meaningful.)

NOTE The preferred implementation for control units that perform queueing and have detected a queueing-timing-facility-failure condition is to continue queueing subsequent I/O operations and present the indication of meaningless queueing time.

8.10.3.6 Defer-Time parameter

When the defer-time function is provided by the control unit, bytes 2 and 3 of word 1 contain the defer-time parameter (DTP). (See Figure 29.) Device-defer time shall be accumulated by the control unit whenever ongoing channel program execution needs to be temporarily delayed because a device-dependent action needs to be performed. This condition typically occurs whenever the control unit requires the device to access the physical medium for the purpose of reading or writing data. However, because of the device-dependent nature of the action, other conditions may also be recognized by the control unit creating the need for a temporary delay of channel program execution. In all cases, device-defer time shall continue to be accumulated with each successive temporary delay until such time when either the control unit forces a disconnection from the channel or execution of the channel program has been completed. If the control unit disconnects from the channel, the previously accumulated defer-time value shall be presented as the device-defer-time parameter along with status containing channel-end without device end. (This status may or may not include command retry status.) When the temporary delay of channel program execution is no longer required, the connection to the channel shall be reestablished and device-end status shall be accepted from the control unit. Device-defer time shall not be accumulated during the time interval of a disconnection from the channel. If no disconnection from the channel occurs during execution of the channel program, the previously accumulated device-defer time shall be presented as the device-defer-time parameter along with status of channel end, signaling completion of channel program execution at the channel. (Status of device end may also be included with the channel-end status.) In all cases, the control unit shall retain the defer time information until it regards the status as accepted. When this has occurred, all previously-stored or accumulated defer-time values shall be zeroed at the control unit and accumulation, if any, begins anew. The defer-time parameter shall be obtained in the following manner with the format specified in Figure 29.

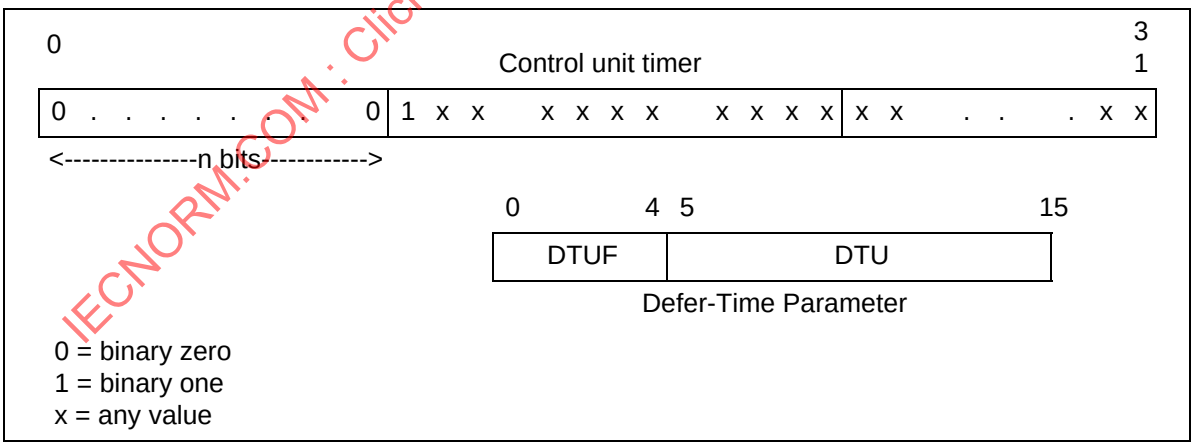

Figure 29 – Defer time parameter format

The defer-time parameter shall be calculated by the control unit using a 4-byte timer having a 1 µs resolution. The value accumulated by the 4-byte control unit timer shall specify the defer-time unit to be stored in the defer-time parameter field. When the channel accepts the accompanying status, it uses the control information, and is able to re-establish the 4-byte time value (with low-order truncation).

Defer Time Unit Factor (DTUF): Bits 0-4, when non-zero, shall specify the number of bit positions the 11-bit defer time unit shall be shifted. If a shift was required, the DTUF shall be calculated by subtracting the value of the highest bit-position (which was set to one) of the queue-time unit from 21. (This shall apply only in those cases where the bit position is equal to or less than 21.) If no adjustment was required, the DTUF value shall be '00000'b.

Defer Time Unit (DTU): Bits 5-15 shall contain the 11-bit defer-time unit accumulated by the control unit timer. The bit-significance of the time unit shall be specified by the DTUF.

Table 15 shown below describes the relationship between the values of DTUF and the bit-significance of DTU.

Table 15 – DTF/DTU relationship

When DTUF is:	Bit 15 of DTP represents:
00000	1 µs
00001	2 µs
00010	4 µs
00011	8 µs
.	.
.	.
.	.
10101	2 097 152 µs

In the example given in Figure 30, the control unit timer has accumulated a defer time of 9 464 µs (hex'24F8') and calculated the defer-time parameter. When received by the channel, a shift of the most significant bit of the DTU to bit-position 18 of the channel 4-byte-time-value, and bit 15 of the DTP represents a value of 8 µs. The low-order 3-bit positions are set to zeros. See 3.3.3 for a definition of the bit-numbering convention used in Figure 30.

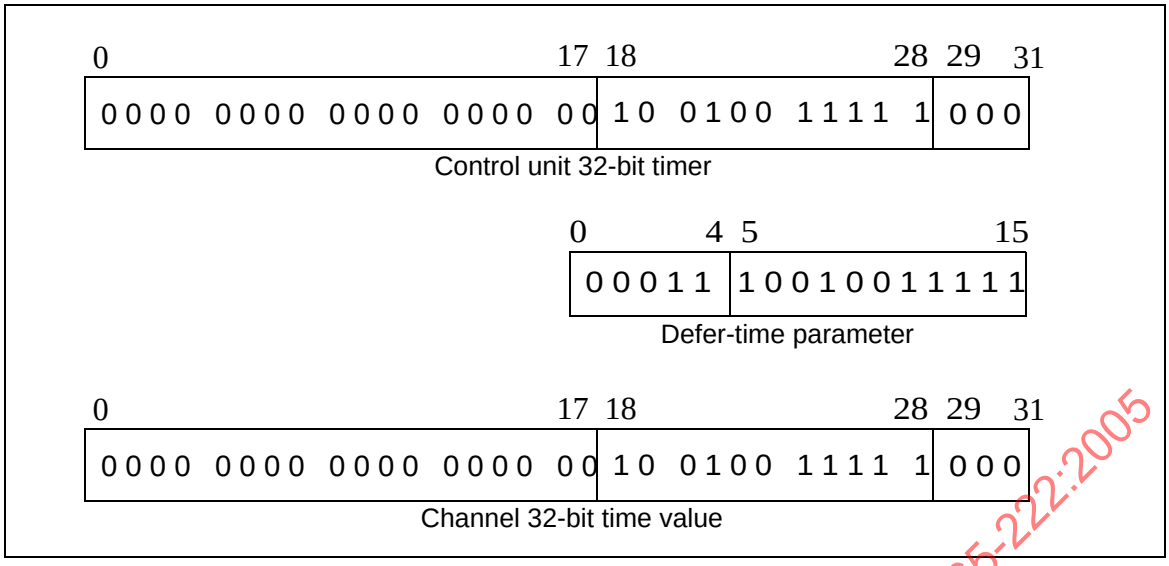

Figure 30 – Defer-time parameter example

A control unit that performs the defer function shall have a defer-timing facility which is capable of measuring a time interval of up to its expected maximum defer time, not exceeding 4 294,967 295 s.

When a control unit converts from a 4-byte timer value to a defer-time unit value, if any of the bits of the 4-byte timer value are lost while shifting, the least significant bits shall be discarded (truncated).

Whenever the defer-time-unit factor is value ‘00000’b and the defer-time unit is ‘00000000000’b, either no accumulation shall have occurred, the defer-time function is not supported, a failure of the defer-timing facility shall have been recognized, or an overflow condition shall have been recognized.

If a defer-timing facility failure has occurred, the control unit shall send a status DIB with unit check status and sense information describing the failure to allow device-dependent program recovery. If a defer-timing overflow condition has been recognized, unit check status shall not be presented to the channel. When a defer-timing facility failure has been recognized and after unit-check status has been presented and accepted, the control unit shall continue with execution of subsequent channel programs as before; however, it shall present an indication of meaningless defer-time values by setting the DTUF and DTU fields to zeros. (In this case, unit-check status shall not be presented after the initial indication of defer-timing information is not meaningful.)

NOTE 1 The preferred implementation for control units that perform the defer-timing function and have detected a defer-timing-facility-failure condition is to continue with channel program execution, when appropriate, and present the indication of meaningless defer time as described previously.

NOTE 2 Accumulation of excessive defer time when a connection exists may result in the recognition of an SB-2 exchange error. To minimize the probability of this occurrence, the control unit should keep its defer time interval at a minimum by disconnecting from the link.

8.10.4 Supplemental Status Field

8.10.4.1 Supplemental Status Field Overview

Supplemental status, when present, shall provide additional information concerning conditions at the control unit or device for which status information is being provided. Supplemental status shall exist

only for unit-check status. When supplemental status is included in a status DIB, it shall be sent in the supplemental-status field. When supplemental status is not included in the status DIB with its associated status information, the supplemental status shall be held by the control unit until one of the following occurs:

– It is read by the appropriate sense command.

– The associated status is reset.

The supplemental-status field shall be valid when the ES bit, bit 4 of the information-unit identifier, is set to one for a status DIB. Up to 32 bytes of sense information may be sent in the supplemental-status field. The contents and exact number of bytes in the supplemental-status field are model dependent. (See 8.10.4.2.)

When sense data is sent as supplemental status, it shall be sent in ascending order starting with sense-data byte 0 as the first byte of the supplemental-status field; it shall be sent in a single status DIB along with the status byte. The length of sense data sent in a status DIB with supplemental status shall be from a minimum of one byte to a maximum of 32 bytes. The number of bytes of sense data beyond one is model dependent.

Supplemental status shall not be used to report sense data associated with a resetting-event indication. (See 9.4.)

Supplemental status shall be sent in a single status DIB.

8.10.4.2 Sense information

The data that is transferred during a basic sense operation or that is contained in the supplemental-status field of a status DIB when unit check is indicated, shall provide information concerning unusual conditions detected in a previous I/O operation and concerning the actual state of the device. Sense information shall provide more detailed information than the status byte and may describe reasons for the unit-check indication. It may also indicate, for example, that the device is in the not-ready state or that a drive is in the write-protected state.

Basic sense data shall not be reset as a result of a device executing the sense-ID command.

Bits 0-5 of the first sense-data byte (sense byte 0) shall be common to all devices. The six bits shall be independent of each other and, when set to ones, shall specify the events described in Table 16. See 3.3.3 for a definition of the bit-numbering convention used in the table.

Table 16 – Bits 0 - 5 of Sense-data byte 0

Sense bit	Description
0	Command Reject
1	Intervention required
2	Bus-out check
3	Equipment check
4	Data check
5	Overrun

Presentation of sense data as supplemental status is described in 9.1.5.2.

8.11 Control DIB structure

8.11.1 Control DIB structure overview

A control DIB shall be sent only in a control IU by either a channel or control unit. It shall be used to transfer control information necessary for the execution of an I/O operation or for device-level recovery. The structure of the control IU and its DIB is shown in Figure 12. The control DIB consists of a 12-byte control header and a 4-byte LRC.

8.11.2 Control header

8.11.2.1 Control header format

The control header shall be the first 12 bytes of a control DIB. The control header shall contain information indicating the control function to be performed and, for some functions, the parameters needed to perform the specified function. The control header shall consist of a one-byte control-function field, a three-byte control-parameters field, IU count field, and reserved bytes. The format of the control header shall be as shown in Figure 31.

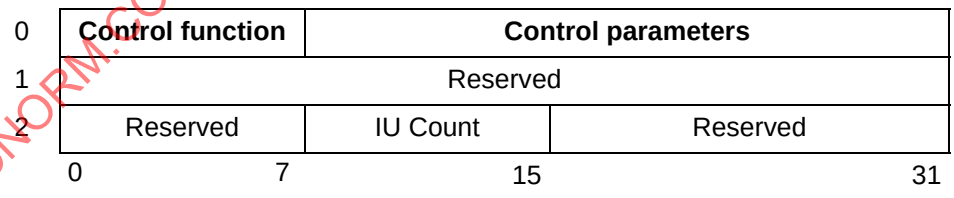

Figure 31 – Control header

Word 1 and bytes 0 and 2-3 of word 2 shall be reserved and set to zeros by the sender and ignored by the recipient.

8.11.2.2 Control function and control parameters

Byte 0 of word 0 shall contain the control function, which shall be interpreted in conjunction with the bits in the device-header-flags field of the IU header to determine the device-level function to be performed and the format of the control-parameter field. A summary of the functions represented by coding of bits 0-4 of the control-function field is shown in Table 17. See 3.3.3 for a definition of the bit-numbering convention used in the table.

Table 17 – Summary of Device-Level control functions

Bit positions					Control function
0	1	2	3	4	
0	0	0	0	0	Control end
0	0	0	1	0	Command response
0	0	1	0	0	Stack status
0	0	1	1	0	Cancel
0	1	0	0	0	System reset
0	1	0	1	0	Selective reset
0	1	1	1	0	Request status
1	0	0	0	0	Device-level exception
1	0	1	0	0	Status accepted
1	0	1	1	0	Device-level acknowledgment
1	1	0	0	1	Purge path
1	1	0	1	0	Purge path response

NOTE All combinations of bits 0-4 that are not listed are reserved.

If a reserved combination of bits 0-4 of the control-function field is used, a device-level protocol error shall be detected.

For all control DIBs, bits 5, 6, and 7 of the control-function field shall be set to zeros by the sender and are ignored by the recipient. Table 18 on page 113 provides a definition, in summary form, of the IUI bits and device-header flag bits for the various control functions. The recipient of a control DIB shall check the IUI bits and device-header flag bits for adherence to the description contained in the table. Bits that are required to be checked by the recipient which are set to a value other than specified by the table for the control function, shall result in a device-level protocol error being detected.

If the channel receives a control IU specifying a control function that only the channel is allowed to send and a control unit is allowed to receive, for example, a cancel function, a device-level protocol error shall be detected. If a control unit receives a control IU specifying a control function that only a

control unit is allowed to send and the channel is allowed to receive, for example, a command response function, a device-level protocol error shall be detected.

Control IUs shall be sent as either FC-2 solicited or FC-2 unsolicited control information categories. A control IU that initiates a connection shall be sent as an unsolicited control information category. A control IU that does not initiate a connection shall be sent as a solicited control information category. Table 18 gives a summary of the control functions, and the sections following Table 18 describe each control function and any control parameters required. See 3.3.3 for a definition of the bit-numbering convention used in the table.

Table 18 – Summary of control IUs

Control function	IUI								Device-Header flag bits								FROM	F, M, or L	Control parameter field
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7			
				AS	ES	T3	T2	T1	E			CH	EE	CNP					
Control end	z	z	z	1	z	0	1	1	z	z	z	z	1	z	z	z	CH	M	z
Command response	z	z	z	1	z	0	1	1	z	z	z	z	z	z	z	z	CU	F or M	z
Stack status	z	z	z	x	z	0	1	1	z	z	z	z	z	z	z	z	CH	FxL or L	z
Cancel	z	z	z	1	z	0	1	1	z	z	z	z	z	z	z	z	CH	F or L	z
System reset	z	z	z	0	z	0	1	1	z	z	z	z	z	z	z	z	CH	F	z
Selective reset	z	z	z	1	z	0	1	1	z	z	z	z	z	z	z	z	CH	F or L	*
Request status	z	z	z	1	z	0	1	1	z	z	z	z	z	z	z	z	CH	F	z
Device-level exception: Address exception	z	z	z	1	z	0	1	1	z	z	z	z	z	z	z	z	CU	FxL	$
Status accepted	z	z	z	x	x	0	1	1	z	z	z	x	z	z	z	z	CH	FxL or L	z
Device-level acknowledgment	z	z	z	x	z	0	1	1	z	z	z	z	z	z	z	z	CU	FxL or L	z
Purge path	z	z	z	1	z	0	1	1	z	z	z	z	z	z	z	z	CH	F	z
Purge path response	z	z	z	1	z	0	1	1	z	z	z	z	z	z	z	z	CU	FxL	z

Explanation:

*	Includes modifier bits in control-parameter field
$	Includes exception code in control-parameter field
F	Start new exchange pair. (First IU of exchange pair)
M	Middle IU of exchange. (Exchange pair remains)
L	Last IU of a pre-existing exchange (Exchange pair removed)
FxL	IU opens and closes an exchange.
CH	Channel sends IU to control unit
CU	Control unit sends IU to channel
x	Bit is set to one or zero as appropriate for conditions
z	Bit or field shall always be set to zero by the sender and is ignored by the recipient
0	Bit shall always be set to zero and is checked for zero
1	Bit shall always be set to one and is checked for one

8.11.2.3 Control end

The channel shall use the control-end function to indicate that it is unable to perform a command update for data chaining. For brevity, a control IU indicating the control-end function is referred to as a control-end IU.

A control-end IU shall be sent only by a channel, and the AS bit and the EE bit shall always be set to one.

When the channel sends a control-end IU, it shall be indicating that, as a result of an abnormal condition, the transfer of the command update for data chaining was suppressed at the channel, and for a write operation, no more data is being sent for the current operation. A control-end IU sent by the channel shall require a status response from the control unit. The sending of additional IUs by the channel shall be held in abeyance until the required response is received.

A control unit that receives a control-end IU shall continue to execute the operation with the device until the conditions for ending an I/O operation are met and then it shall transfer status to the channel. The status DIB shall carry the residual count for the current CCW being executed when the control-end IU was received.

A control-end IU shall require both a valid CCW number and valid token. The CCW number used shall be the value the channel would have sent in the command update had it been able to perform the data chain. The token assigned is model dependent.

A control-end IU shall be sent as a solicited control FC-2 information category. It shall be sent as the middle sequence of an exchange.

8.11.2.4 Command response

A control unit shall use a command-response function to indicate that execution of the first command of a channel program has started as a non-immediate operation, or, a command IU containing both the CRR and CH bits set to one and the DU bit set to either zero or one, has become current. For brevity, a control IU indicating command response is referred to as a command-response IU.

A command-response IU shall be sent only by a control unit. A command-response IU shall be sent only under the following conditions:

a) when the first command of the channel program is to be executed as a non-immediate command, or

b) when execution has started for a command IU, that is command-chained or data-chained, and for which the CRR bit in the command header is set to one.

The sending of a command-response IU shall not be contingent on sending or receiving any of the data associated with the command IU for which it is sent.

A command-response IU shall always require a valid CCW number and token, which shall be the CCW number and token received with the command for which the command response is being sent.

A command-response IU shall be sent as a solicited control FC-2 information category. It shall be sent as the first or middle sequence of an exchange, depending on whether there is an exchange already open for the same device on that logical path.

When the command-response IU is sent in response to the first command of a channel program, the least significant byte of the control-parameters field contains the 8-bit IU pacing parameter. (See 8.10.3.4 for additional information.)

8.11.2.5 Stack status

The channel shall use the stack-status function to indicate that status is not accepted. The control unit or device shall hold the status information (status flags, status byte, and status parameter) and associated supplemental status, if any, until the status is requested by a request-status IU, until the status is presented as the response to a command IU, or until the status information is cleared or withdrawn. For brevity, a control IU indicating stack status is referred to as a stack-status IU.

A stack-status IU shall be sent only by the channel. When a stack-status IU is received at a time when the inbound exchange has been closed with a data IU containing a status DIB, no response shall be sent; otherwise the response sent shall be a DACK IU.

The AS bit may be set to either one or zero in a stack-status IU. The setting of the AS bit in the stack-status IU shall be the same as the setting of the AS bit in the data IU containing a status DIB for which the stack-status IU is sent in response; otherwise, a device-level protocol error shall be detected.

A stack-status IU shall require a valid token but shall not require a valid CCW number. The CCW number field shall be meaningless for a stack-status IU. The token field for a stack-status IU shall not be checked by the recipient.

A stack-status IU shall be sent as a solicited control FC-2 information category. A stack-status IU shall be sent as the first and last sequence of an exchange if it is sent in response to an unsolicited data IU. It shall be sent as the last sequence if it is sent in response to a solicited data IU.

8.11.2.6 Cancel IU

The channel shall use the cancel function to cause the control unit to terminate or nullify the current I/O operation, if any, for the specified device. (See 9.2.3.) For brevity, a control IU indicating cancel, is referred to as a cancel IU.

A cancel IU shall be sent only by the channel. When the cancel IU is received on the outbound exchange of an exchange pair at a time when the inbound exchange of the exchange pair has been closed, no response shall be sent; otherwise the response sent shall be a DACK IU.

A cancel IU shall require a valid token but shall not require a valid CCW number. The CCW number field for a cancel IU shall be meaningless. The acknowledgment sent in response to a cancel IU shall contain the CCW number and token associated with the current command for which the cancel function was performed and for which ending status is to be presented later.

A cancel IU sent during a connection shall be sent as a solicited control FC-2 information category, and shall be sent as the last sequence of the exchange, thus closing the outbound exchange. A cancel IU that initiates a connection shall be sent as an unsolicited control FC-2 information category, shall be sent as the first sequence of the exchange, and shall leave the exchange open. The outbound exchange shall be closed when the channel sends the LACK in response to the DACK.

NOTE A cancel function results in a disconnection when the DACK is received by the channel. In order to avoid having to save the active CCW numbers and tokens used during a connection and after a cancel function has been performed, the control unit indicates in the DACK sent, the CCW number and token for which the cancel was performed.

8.11.2.7 System reset

The channel shall use the system-reset function to cause the control unit and associated devices to be reset with respect to the logical path on which the function was received. (See 9.2.4.) For brevity, a control IU indicating system reset is referred to as a system-reset IU.

A system-reset IU shall be sent only by the channel and when recognized by the control unit the response sent shall be a DACK.

A system-reset IU shall require a valid token but shall not require a valid CCW number. The CCW number field for a system-reset IU shall be meaningless. The token field for a system-reset IU shall not be checked by the recipient. The token value received in the system-reset IU, shall be returned in the DACK response.

A system-reset IU shall always initiate a connection, shall be sent as an unsolicited control FC-2 information category, shall be sent as the first sequence of the exchange, and shall leave the exchange open. The outbound exchange shall be closed when the channel sends the LACK in response to the DACK.

8.11.2.8 Selective reset

The channel shall use the selective-reset function to cause the control unit to end execution of the current operation, if any, for the specified device, and, depending on the bits within the control-parameter field, to perform one of the following functions:

– selective reset,

– channel-initiated retry, or

– channel-initiated unit check.

(See 9.2.5 and 9.3.3.) For brevity, a control IU indicating selective reset is referred to as a selective-reset IU.

A selective-reset IU shall be sent only by the channel. When the selective-reset IU is received on the outbound exchange of an exchange pair at a time when the inbound exchange of the exchange pair has been closed, no response shall be sent; otherwise the response sent shall be a DACK IU. When a selective reset IU which initiates a connection is received, the response sent shall be either a DACK IU or a data IU containing a status DIB. (See 9.2.5 for additional information.)

The control-parameter field for a selective-reset IU shall have the format shown in Figure 32.

RC	0	0	RU	RO	Reserved
0	1	2	3	4	5 23

Figure 32 – Control-parameter field for the selective-reset IU

Bits 1, 2, and 5-23 shall be reserved.

When the RC bit, RU bit, and RO bit of the control-parameter field are all set to zeros, a selective reset shall be performed.

When one or more of the RC, RU, and RO bits of the control-parameter field are set to one and the requested function or functions are not able to be performed, a selective reset shall be performed.

When the RC bit of the control-parameter field is set to one, a channel-initiated retry shall be performed, if possible; if retry is not possible, the RU bit and RO bit of the control-parameter field shall determine the preferred method of terminating the I/O operation.

When either the RU bit or the RO bit of the control-parameter field is set to one and channel-initiated retry is either not requested or requested and not performed, a channel-initiated unit check shall be performed, if possible; if channel-initiated unit check is not performed, a selective reset shall be performed.

A selective-reset IU shall require a valid token and, depending on conditions at the channel and the setting of the RC, RU, and RO bits, a valid CCW number may or may not be required. If the RC, RU, or RO bit is set to one and the channel is requesting one of these functions to be performed for a specific CCW, then a valid CCW number shall be required. If the RC, RU, or RO bit is set to one but the channel is requesting that one of these functions be performed for the current CCW, a valid CCW number shall not be required and the value in the CCW number field shall be set to zero. (See 9.3.3.) If the RC, RU, and RO bits are all set to zero, a valid CCW number shall not be required. In this case the CCW number field shall be meaningless. If a selective-reset IU is sent after the control unit has disconnected and the token and CCW numbers have been reinitialized at the channel, a valid CCW number shall not be required and the CCW number field shall contain the value zero. The token field for a selective-reset IU shall not be checked by the recipient.

The DACK IU sent in response to a selective-reset IU shall contain the token and CCW number associated with the current command at the control unit which was terminated by the selective-reset function. If status is returned in a data IU as a result of performing either the request for retry or request for unit check functions, the token and CCW number associated with the command for which the function is being performed shall be used. If a disconnection occurs as a result of an abnormal condition and a selective reset is received as a result of recovery actions at the channel, the CCW number and token being maintained at the time of the disconnection shall be returned in the DACK. If, as a result of a normal disconnection from the channel, CCW numbers and tokens have been discarded, the DACK sent shall contain the token and CCW number received in the selective-reset IU.

Request Command Retry (RC): The RC bit, bit 0, when set to one, shall indicate that the channel is requesting the device to perform command retry on behalf of the channel.

If the RC bit is set to one, the device shall perform command retry if possible; if command retry is not able to be performed, the RU and RO bits shall determine the action to be taken. If the RC bit is set to zero, the RU and RO bits shall determine the action to be taken.

Request Unit Check (RU): Bit 3, when set to one, shall indicate that the channel is requesting the device to present unit check status. If the command is currently being executed by the device, the I/O operation shall be terminated and either unit check status shall be presented to the channel or the selective-reset operation performed.

Request Unit Check with Overrun (RO): Bit 4, when set to one, shall indicate that the channel is requesting the device to recognize an overrun condition and present unit check status. If the command is currently being executed by the device, the I/O operation shall be terminated and either unit check status shall be presented to the channel or the selective-reset operation performed.

The channel may set to one either the RO bit or the RU bit depending on the error condition. When command retry is either not requested or requested and not performed, the device shall interpret the RU and RO bits in the control-parameter field as shown in Table 19:

Table 19 – Interpretation of the RO and RU bits

RU	RO	Interpretation
0	0	A selective reset is performed
0	1	The device shall recognize an overrun condition and generate unit-check status with the sense data. Retry may be requested.
1	0	The device shall recognize a unit check and generate unit-check status with the appropriate sense data.
1	1	Error - the device shall ignore the requested functions and perform a selective reset.

A selective-reset IU that initiates a connection shall be sent as an unsolicited control FC-2 information category, shall be sent as the first sequence of the exchange, and shall leave the exchange open.

A selective reset shall be sent during a connection only if the RC, RU, and RO bits are all set to zero. It shall be sent as a solicited control FC-2 information category and as the last sequence of the exchange, closing the outbound exchange.

8.11.2.9 Request status

The channel shall use the request-status function to indicate that the channel is prepared to have status information presented for the addressed device for which status is pending. (See 9.2.2 and 9.2.6.) For brevity, a control IU indicating request status is referred to as a request-status IU.

A request-status IU shall be sent only by the channel.

The AS bit shall be set to one in a request-status IU.

For a request-status IU the CCW number and token assigned by the channel are model dependent.

A request-status IU shall always initiate a connection; it shall be sent as an unsolicited control FC-2 information category as the first sequence of the exchange.

8.11.2.10 Device-Level exception

A control unit shall use the device-level-exception function to indicate that an abnormal condition was recognized in the IU received for which the sending of status is inappropriate or not permitted. The abnormal condition shall be indicated by the exception code present in the first byte of the control-parameter field. For brevity, a control IU indicating device-level exception is referred to as a device-level-exception IU.

A device-level-exception IU shall be sent only by a control unit and may only be sent in response to an initiation IU.

A device-level exception shall have a control-parameter field with the format shown in Figure 33.

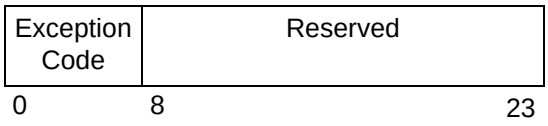

Figure 33 – Control-parameter field for the device-level exception IU

Bits 8-23 of the control-parameter field shall be set to zeros by the sender of the device-level-exception IU and shall be ignored by the channel.

Bits 0-7 shall specify the abnormal condition detected in the IU for which this device-level-exception IU is the response. Table 20 shows code assignments for the exception codes. See 3.3.3 for a definition of the bit-numbering convention used in the table.

Table 20 – Exception code assignments

Bits								Exception code
0	**1**	**2**	**3**		**4**	**5**	**6**	**7**	
0	0	0	0		0	0	0	0	Reserved
0	0	0	0		0	0	0	1	Address exception
0	0	0	0		0	0	1	0	Reserved
				through					
1	1	1	1		1	1	1	1	

If a reserved exception code is used, a device-level protocol error shall be detected by the recipient of the device-level-exception IU.

A device-level-exception IU shall use the token and CCW number from the IU to which it is being sent in response.

Address Exception: Address exception shall indicate that the IU received contained the device address of an uninstalled device or a device that is offline with respect to the logical path. A device-level-exception IU for an address exception shall be sent only in response to certain IUs containing device-level information. (See 9.3.4 for a description of the use of this exception code.) For brevity, a control IU indicating address exception is referred to as an address-exception IU.

An address-exception IU shall use the token, CCW number and device address from the IU to which it is being sent in response.

An address-exception IU shall be sent as a solicited control FC-2 information category, and it shall be sent as the first and last sequence of the exchange.

NOTE The device may be offline for some logical paths but not others. For example, the device may be precluded from going to the offline state if the offline switch was set to the offline position but the allegiances for some paths have not been cleared.

8.11.2.11 Status accepted

The channel shall use the status-accepted function to indicate that the status received in a status DIB was accepted. For brevity, a control IU indicating status accepted is referred to as a status-accepted IU.

A status-accepted IU shall be sent only by the channel. If the control unit did not close the inbound exchange with a data IU containing the status DIB, the response sent by the control unit shall be a DACK; otherwise no response is sent.

The CH bit in the device-header flag field for a status-accepted IU may be set to either one or zero, depending on conditions at the channel. (See 9.2.8.)

The AS bit may be set to either one or zero. The setting of the AS bit in the status-accepted IU shall be the same as the setting of the AS bit in the data IU containing a status DIB for which the status-accepted IU is sent in response; otherwise, a device-level protocol error shall be detected.

The ES bit may be set to either one or zero in a status-accepted IU. The ES bit shall be set to one to indicate that the channel has accepted the supplemental status received in a status DIB with the ES bit set to one. If the channel has not accepted the supplemental status received, or there was no supplemental status presented in the status DIB, the ES bit shall be set to zero. (See 9.1.5.2.)

The status-accepted IU shall require a valid token but does not require a valid CCW number. The token value used in the status-accepted IU is model dependent. The CCW number field and token field for a status-accepted IU shall not be checked.

A status-accepted IU shall be sent as a solicited control FC-2 information category. It shall be sent as the first and last, or last sequence of the outbound exchange, depending on whether there is an outbound exchange already open for that device or logical path.

8.11.2.12 Device-Level Acknowledgment (DACK)

The device-level acknowledgment (DACK) function shall indicate acceptance of a stack-status IU, cancel IU, system-reset IU, selective-reset IU, request-status IU, or status-accepted IU and that the requested function was or is to be performed. For brevity, a control IU indicating a device-level ACK is referred to as a DACK IU.

A device-level acknowledgment shall be sent only by a control unit.

The AS bit may be set to either one or zero. The setting of the AS bit in the DACK IU shall be the same as the setting of the AS bit in the IU for which the DACK is sent in response; otherwise, a device-level protocol error shall be detected.

Except for a response to a cancel or selective-reset IU, a DACK IU shall use the token and CCW number from the IU to which it is being sent in response. When a DACK IU is sent in response to a cancel or selective-reset IU, the token and CCW number used shall depend on whether or not the cancel IU initiated a connection, conditions at the control unit and, for a selective-reset IU, the settings of the RC, RU and RO bits. See 8.11.2.8.

A DACK IU shall be sent as a solicited control FC-2 information category. It shall be sent as the first and last, or last sequence of the inbound exchange, depending on whether there is an inbound exchange already open for that device or logical path.

8.11.2.13 Purge path

A channel shall use the purge-path function as part of device-level recovery in order to purge the transmit-receive path of any residual IUs associated with the logical path and device for which recovery is being performed. For brevity, a control IU indicating purge path is referred to as a purge-path IU.

A purge-path IU shall be sent only by a channel and when recognized by a control unit the response shall be a purge-path-response IU.

A purge-path IU shall require a valid token but shall not require a valid CCW number. The CCW number field for a purge-path IU shall be meaningless. The token value received in the purge-path IU shall be returned in the token field of the purge-path-response IU.

A purge-path IU shall be sent as the first sequence of an outbound exchange as an unsolicited control FC-2 information category.

8.11.2.14 Purge-Path response

A control unit shall use the purge-path-response function to indicate receipt and recognition of the purge-path function. For brevity, a control IU indicating purge-path response is referred to as a purge-path-response IU.

A purge-path-response IU shall be sent only by a control unit in response to a purge-path IU. The channel shall send a LACK IU to close the outbound exchange when it receives a purge-path-response IU.

A purge-path-response IU shall be sent as a solicited control FC-2 information category, and shall be sent as the first and last sequence of the exchange.

8.12 Link-Control DIB structure

8.12.1 Link-Control DIB structure overview

A link-control DIB shall be sent in a control IU by the channel or control unit. It shall be used to transfer SB-2 link-level information. A control IU containing a link-control DIB is referred to as a link-control IU.

A link-control IU shall be used to perform the following link-level functions: (See 6.4.)

– establish a logical path,

– remove a logical path,

– test initialization,

– acknowledge, reject or indicate a busy condition.

A link-control DIB shall contain a link header, an LRC, and for some functions a link payload with its associated CRC field. See Figure 12.

8.12.2 Link header

8.12.2.1 Link header format

The link header shall be generated by the channel or control unit and shall appear immediately after the IU header of either a solicited or unsolicited control IU. The link header shall contain information indicating the link-control function to be performed and, for some functions, the parameters needed to perform the specified function. The link header shall consist of a one-byte link-control-function field, a two-byte link-control-information field, a two-byte CTC counter field, a one-byte IU count, a two-byte link payload byte count, and reserved bytes. The link header shall have the format shown in Figure 34.

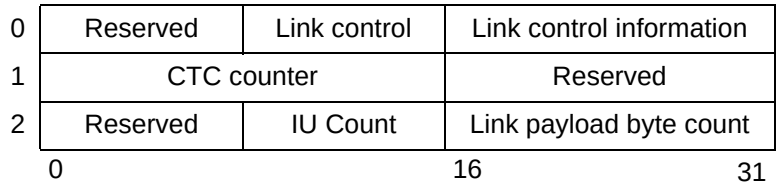

Figure 34 – Link header

Byte 0 of word 0, bytes 2 and 3 of word 1, and byte 0 of word 2 shall be reserved and set to zero by the sender and ignored by the recipient. For all IUs except the ELP IU, bytes 0 and 1 of word 1 are also set to zero by the sender and ignored by the recipient.

8.12.2.2 Link control

The link-control field, byte 1 of word 0, shall contain the link-header format bits, bits 5-7, and the link-control function bits, bits 0-4. The link control field shall have the format shown in Figure 35.

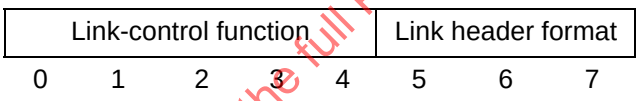

Figure 35 – Link-control field

When bits 5-7 are set to the value ‘001’b, bits 0-4 of the link-control field shall contain the link-control function. See 6.4 for the link-control functions and their assigned codes. All other values of bits 5-7 shall be reserved. If a value of other than ‘001’b is specified, a link-level protocol error shall be recognized.

A link-control function, when sent as a request by the channel, shall require a valid token and when sent as a request by a control unit, the token value shall be meaningless. A link control response sent by a control unit shall use the token and CCW number from the IU to which it is being sent in response. A link control response sent by a channel shall have a valid token assigned by the channel (but not a valid CCW number) and shall not use the token and CCW number from the IU to which it is being sent in response.

8.12.2.3 Link-Control information

The link-control information field, bytes 2 and 3 of word 0, shall contain additional parameters necessary in order to perform the link-control function specified by bits 0-4 of byte 1 of word 0. The parameters shall be determined by the link-control function specified.

The link-control information field shall be meaningful only when the ELP, LPE, LPR, and LRJ link-control functions are specified. (See 6.4.2, 6.4.4, 6.4.5 and 6.4.9 for a description of the parameters.)

The parameters associated with the LPR and LRJ link-control functions shall be contained in the first byte of the link-control information field, byte 2 of word 0. The second byte of the link-control information field, byte 3 of word 0, shall always be set to zero by the sender and ignored by the recipient.

The parameters associated with the ELP and LPE link-control functions shall be contained in both bytes of the link-control information field, bytes 2 and 3 of word 0.

8.12.2.4 Channel-to-Channel (CTC) counter

The CTC counter field, bytes 0 and 1 of word 1, is only meaningful when ELP is the link-control function specified, and bit 15 of the link-control information field is set to one. The CTC counter specifies the number of CTC connections for which the channel sending the ELP IU is providing emulated control unit functionality at the time when the ELP IU is sent.

8.12.3 Link payload

The link payload shall be a variable length field with a maximum length of 8 156 bytes. The link payload field shall be present only for the TIR link-control function. (See 6.4.8.)

9 Device-Level functions and protocols

This clause describes the functions and protocols necessary for the execution of an I/O operation, the exchange of control information, and device-level recovery. These functions depend on the successful completion of all FC-PH protocols.

9.1 Device-Level operations

9.1.1 Overview of Device-Level operations

Initiation of an I/O operation, the transfer of data, and the ending of an I/O operation are device-level operations which rely on SB-2 device-level functions and protocols. The device-level functions discussed in this clause have all been defined in Clause 8. This clause describes how these functions are used in order to perform device-level operations.

9.1.2 Channel program execution

A channel may concurrently execute one or more channel programs on the same or different logical paths. Each channel program executed by a channel shall be for a different device and shall require a different exchange pair to be used.

The number of channel programs that may be concurrently executed by a channel or control unit is model dependent.

9.1.3 Initiating an I/O operation

An I/O operation shall be initiated with a device when the channel transfers the command from the current CCW to that device. The command shall be transferred in the command header of a command IU. (See 8.7.2.)

A channel executing a channel program may concurrently initiate multiple I/O operations with the same device. Each I/O operation shall consist of the transfer of a command in a command IU containing either a command DIB or a command-data DIB and, in the case of a write operation with a nonzero CCW count, the transfer of data in a data IU.

A control unit that receives one or more command IUs, each initiating an I/O operation, shall execute each command in the order in which it is received from the channel. Execution of the command first shall require acceptance of the command at the control unit. When a command is accepted, that command shall be considered current at the control unit until execution at the device is considered completed. At least the following conditions shall be satisfied in order for the command to be accepted.

a) The AS bit shall be set to one in the IUI.

b) For those commands that require the device to be installed and ready, the device address shall specify a device that is installed and ready.

c) For command chaining, the device address shall designate the same device that was designated for the command IU which initiated the channel program.

d) The chaining (CH) bit shall be set to zero for the first command of a channel program and set to one for all subsequent commands of the channel program.

e) For those devices that provide commands executed as immediate operations, the command shall be checked to determine whether a command response or status is to be sent.

f) If the REX flag bit is set to one, the CD, DU, and SSS flags shall be set to zero in the command header.

At least the following conditions shall be satisfied in order for the control unit to return busy status to the channel.

a) The status DIB indicating device busy or control unit busy (status modifier and busy) may be sent only in response to the first command of a channel program (CH bit set to zero in the command IU).

b) A response of control unit busy shall require that the IUI of the command IU be checked. No other device-level checking shall be required.

c) A response of device busy shall require that the IUI and device address be checked. No other device-level checking shall be required.

When the command is the first command of the channel program, the channel shall consider a connection to exist when the command response or status is received. The control unit shall consider a connection to exist when the command response or status is sent.

For the first command of a channel program, the channel shall send the command in a command IU with either a command DIB or command-data DIB. If the command is for a write operation and the CCW count is nonzero, a command-data DIB containing both the command and data for that command shall be used. If the command is for either a write operation with a zero CCW count or a read operation, a command DIB shall be used.

The channel shall consider the first command of a channel program to be accepted when either a command response or status indicating channel-end is received.

For the first command of a channel program, the channel shall consider the I/O operation to be in progress at the device when the indication that the command has been accepted by the device is received. In this case, the device shall consider the I/O operation to be in progress when the indication that the command was accepted is sent to the channel. Acceptance of the command by the control unit shall signal the start of execution of the channel program.

When the first command of a channel program is handled by the control unit as a nonimmediate operation, acceptance of the command for execution shall be indicated by sending a command response to the channel. If the operation is a read, data may be sent by the control unit after the command-response IU is sent. If the operation is a write with a nonzero CCW count, data shall be sent by the channel with the command and, if necessary, in data IUs immediately following the command IU.

When the first command of a channel program is handled by the control unit as an immediate operation, acceptance of the command for execution shall be indicated by sending a solicited data IU containing a status DIB containing channel-end status without busy status. Command IUs with the SYR bit set to zero and data IUs received subsequent to the command IU containing the first command of the channel program, shall be discarded. If device-end status is not included with the channel-end status, the channel shall indicate that the status is accepted by sending a status-accepted IU. The chaining bit (CH bit) shall be set to one in the status-accepted IU if chaining is to be indicated by the channel on acceptance of the status; if chaining is not to be indicated, the CH bit shall be set to zero on acceptance of the status. Execution of the command shall be permitted when the command is accepted.

When device-end status is included with the channel-end status for an immediate operation, the response from the channel indicating status is accepted depends on whether or not chaining is to be indicated by the channel. If chaining is not to be indicated, the channel shall send a status-accepted

IU with the CH bit set to zero. If chaining is to be indicated by the channel, then the channel shall indicate status is accepted by sending the next command in a command IU with the CH and SYR bits set to one.

When the status DIB received by the channel in response to the first command of a channel program indicates that supplemental status is available, the status may be accepted as previously described, and acceptance of the supplemental status shall be indicated by the ES bit. (See 9.1.5.2.)

When the status received by the channel in response to the first command of a channel program requests that command retry be performed, the channel shall indicate acceptance of status and its intent to perform the retry by sending either a status-accepted IU or a command IU, depending on whether device-end status is present in the status DIB and on whether the channel intends to perform the command retry. The response from the channel shall depend on the following conditions.

– If device-end status is not present, the channel shall indicate acceptance of the status by sending a status-accepted IU. If command retry is to be performed, the CH bit shall also set to one in the status-accepted IU. From the time when the status-accepted IU is sent, the channel shall consider the rules for command chaining to be in effect. (See 9.3.2.) Conversely, if command retry is not to be performed, the CH bit shall be set to zero in the status-accepted IU.

– If device-end status is present and command retry is to be performed, the channel shall indicate acceptance of the status and of the command retry by sending a command IU with the command to be retried, along with the flags and parameters associated with the command. The CH and SYR bits shall be set to one in the command IU. The CH bit shall indicate that this command is a command update as a result of command chaining and the SYR bit shall identify this command as the one being retried. From the time the command IU is sent to the device, the channel shall consider the I/O operation to be in effect at the device and the rules for command chaining in effect. (See 9.3.2.)

– If device-end status is present and command retry is not to be performed, the channel shall indicate acceptance of the status by sending a status-accepted IU. The CH bit shall be set to zero in the status-accepted IU to indicate that command retry is not to be performed.

Commands which are not executed as the first command of a channel program shall not result in a command response or status being sent to the channel, unless one of the following conditions exists:

– the CRR flag is set to one for the command, or

– conditions at the control unit require the sending of status. (See 8.10.)

The following protocols shall be used for commands which are not executed as the first command of a channel program.

– When the command is handled by the control unit as a nonimmediate operation, acceptance and execution of the command may occur immediately. If the CRR flag is set to one for the command, a command response shall be sent to the channel upon acceptance of the command. If the CRR flag is set to zero for the command, a command response shall not be sent to the channel upon acceptance of the command. If the operation is a read, data transfer may proceed immediately after sending the command-response IU, if required. If the operation is a write, the data associated with the command contained in the command-data DIB and data DIBs received may immediately be transferred to the device.

  The control unit shall consider the I/O operation to be in progress at the device when the conditions for acceptance of the command are satisfied.

– When the command is handled by the control unit as an immediate operation, acceptance and execution of the command may occur immediately. If the CRR flag is set to one for the command, a command response shall be sent to the channel. (See 8.7.2.3.) If the CRR flag is set to zero for the command, a command response shall not be sent to the channel upon acceptance of the command. At the completion of execution of the command, status may or may not be sent to the channel depending on the conditions previously described. If status is sent, and the CCW count for this command is non-zero, the control unit shall indicate that it was executed as an immediate operation by setting the LRI bit in the status flags field to a one and the residual count to the value of the CCW count.

9.1.4 Data-Transfer protocol

9.1.4.1 Data-Transfer protocol overview

Data is transferred between the channel and the control unit as part of the execution of an I/O operation. The transfer of data from the channel to the control unit is a write operation. The transfer of data from the control unit to the channel is a read operation. The operation, read or write, is determined by the current command. Read commands have bit 7 of the command byte set to zero, and write commands have bit 7 of the command byte set to one.

Some read and write commands do not result in the transfer of data. These commands are executed as immediate operations, are designed to be executed without transferring data, have counts of zero in the CCW, or have data-record lengths of zero. (See 9.1.3 for the protocols used when the command is executed by the control unit as an immediate operation. See 9.1.4.6 for the protocols used when the CCW count is zero.)

When, for the first command of a channel program with a nonzero CCW count, the data-record length at the control unit is zero but the command is not accepted as an immediate operation, a command-response IU is sent to the channel. Following transmission of the command response, if status is required to be sent to the channel, status containing channel end with or without device end is sent to the channel, the residual-count-valid bit is set to one, and the residual count field contains the CCW count received with the command. For this case, if the command is a read command, no data is transferred to the channel. If the command is a write command and data was sent for the command, the data is discarded by the control unit.

When, for a command other than the first command of a channel program with a nonzero CCW count, the data record length at the control unit is zero, a command-response IU is sent to the channel only if the CRR flag is set to one for the command. If conditions require the sending of status to the channel, status is sent. In the case in which a command response is required and conditions require the sending of status, the status is sent after the command response is sent. When status is sent, the residual counts and data transfer rules for the first command of a channel program apply. If conditions do not require the sending of status and chaining is to take place at the control unit, only a command response, if requested is sent; otherwise, nothing is sent to the channel.

Data is transferred from the channel to the control unit in a command-data DIB sent in a command IU or data DIB sent in a data IU. Data is transferred from the control unit to the channel only in a data DIB. See 8.8 and 8.9 for the maximum DIB sizes.

For each CCW that results in a command IU being sent, the CCW count is included in the command header. For a write operation, the CCW count is used to determine the quantity of data to expect from the channel. For a read operation, the CCW count is used to determine the quantity of data that may be sent to the channel. For both a read and write operation the CCW count is used to determine the residual count to be returned to the channel, if status is to be presented, and it is used in conjunction with the CCW control flags and command flags, to determine if chaining is to occur.

9.1.4.2 Write operation

A write operation is the transfer of data from the channel to the control unit as part of the execution of a command that has bit 7 set to one.

For a write operation with a nonzero CCW count, the channel shall send the command and the data for the command in a command-data DIB. If the quantity of data specified by the count for the current CCW exceeds the maximum allowable DIB size, the remaining data shall be sent in data DIBs until all of the data specified by the CCW count is sent. (See 8.8 and 8.9.) If all of the data specified by the CCW count is sent, the E bit shall be set to one in the last IU sent containing the CRC field. If all of the data specified by the CCW count is not sent, the EE bit shall be set to one in the last IU sent containing the CRC field.

When, during a write operation, the channel is able to send some but not all of the data for a CCW, the EE bit shall be set to one in the last IU sent. The channel, for the affected channel program, shall suspend sending IUs to the control unit until status is received. When status is received, the channel may or may not resume execution of the channel program with the control unit, depending on the contents of the status DIB and conditions at the channel.

When, during a write operation, the channel is unable to send any of the data specified for a CCW, the EE bit shall be set to one for the command-data DIB, the command header shall contain the required information for the CCW, and the DIB data field shall contain only a CRC field. If CRC generation and checking is provided, then the CRC field shall contain only the initialized value of the CRC generator in the DIB data field and the CNP bit shall be set to zero.

When, for a write operation, the quantity of data required by a device is equal to or greater than the CCW count, the resulting residual count shall be zero. If the quantity of data required by the device is less than the CCW count, the resulting residual count shall be the difference between the quantity of data required and the CCW count.

When, at the end of a write operation which is not the first command of a channel program executed as an immediate operation, status including channel end is sent to the channel, the residual count calculated shall be included in the status DIB. If the quantity of data required by the device is greater than the CCW count, the LRI and RV status flag bits shall be set to one and the residual count set to zero to indicate that the device would have used more data had it been provided.

When the quantity of data received by the control unit for a write operation is less than the quantity specified by the CCW count and the EE bit is set to one in the last IU received, the I/O operation shall be executed and the data received, if any, shall be used by the device. If the quantity of data required by the device is exactly equal to the quantity of data received, the E bit shall be set to one for the status DIB sent. If the quantity of data required by the device is either less than or greater than the quantity of data received, the E bit shall not be set to one for the status DIB sent; if a valid CRC value is received and CRC checking is provided, CRC shall be checked on the entire amount of data received.

9.1.4.3 Read operation

A read operation is the transfer of data from the control unit to the channel as part of the execution of a command that has bit 7 set to zero.

For a read operation the channel shall send the command to the control unit in a command DIB in a command IU. The CCW count and CCW control flags for the command shall be included in the command header.

When, for a read operation, the quantity of data sent to the channel exactly equals the count for the current CCW, the resulting residual count shall be zero and the E bit shall be set to one in the last data IU sent for that CCW. If the quantity of data sent to the channel is less than the count for the current CCW, the resulting residual count shall be the difference between the quantity of data sent to the channel and the CCW count. The E bit shall be set to zero and the EE bit shall be set to one in the last data IU sent for that CCW. If more data has been transferred to the channel and has a larger count been provided or data chaining indicated, the resulting residual count shall be zero and the E bit shall be set to one in the last data IU sent for that CCW.

When, at the end of a read operation which is not the first command of a channel program executed as an immediate operation, status including channel end is sent to the channel, the residual count calculated shall be included in the status DIB. If the quantity of data sent to the channel by the device, is exactly equal to the CCW count but the device would have sent more data had a larger CCW count been provided, the LRI and RV status flag bits shall be set to one and the residual count is set to zero.

9.1.4.4 Data chaining

When there are successive CCWs to be executed by the channel for a single I/O operation, the execution of these CCWs is referred to as data chaining. When each CCW associated with a channel program becomes the current CCW being executed, the channel shall indicate to the control unit the initiation of a data chaining operation by transferring a command-IU containing the chain-data flag, CCW count, and other flags associated with the CCW. Subsequent IUs containing data DIBs may also be transferred, either by the channel or control unit, depending upon whether input or output was specified in the command IU initiating the I/O operation. When execution of the current CCW is completed and the chain-data flag is set to one, data chaining shall take place, provided no abnormal conditions are detected and all other conditions for data chaining are satisfied. Data chaining may occur only when the CD bit is set to one for the current CCW. When data chaining takes place, the command IU shall be used to update the flags, CCW number, token, and count held at the control unit, the chaining bit (CH bit) of the device header flag field and the data-chaining-update (DU) flag of the command-flag field shall both be set to ones.

The device shall ensure that data chaining is occurring at the proper times by recognizing a data-chaining condition. The data-chaining condition shall be recognized for each device and shall be used to verify that successive CCWs are being executed by the channel. The data-chaining condition shall be set whenever the device accepts a command IU and the chain-data bit is set to one in the command-flag field. The data-chaining condition shall be reset whenever a control IU is received and system reset, selective reset, cancel or stack status is performed. The data-chaining condition shall also be reset whenever a command IU is received and the chain-data bit is set to zero, or status containing device end has been accepted by the channel.

If a command IU is received with the DU flag bit and the CH bit set to ones, and if the data-chaining condition is set, then the command flags, CCW number, token, and count from the command IU shall be accepted and become current.

The chain-data bit in the CCW-flag field of the command IU shall indicate that there is a subsequent count and command-flag update for the device that is to immediately follow the execution of the current command IU, provided that no abnormal conditions are encountered.

In some cases while data chaining, the channel may request the transfer of status when the last CCW of a data chain is executed. The channel shall do this by setting the SSS bit to one signalling the control unit to provide ending status in an IU containing a status DIB when the I/O operation has ended. In this case, if command chaining was set to one in the last CCW of the data chain, then normal command chaining would occur after the channel received the IU containing the status DIB. If data chaining is ended and no further chaining is specified, ending status shall be sent to the channel.

When data-chaining occurs at the channel during a write operation, a command IU with a command-data DIB containing the CCW number, the CCW count and required flags for the next CCW, a valid token, and data shall be sent to the control unit. If the quantity of data specified for the CCW exceeds the quantity of data that is allowed to be sent in a command-data DIB, one or more data DIBs may be sent. If the quantity of data sent for the CCW is exactly equal to the count in the CCW, the E bit shall be set to one in the last IU sent for the CCW. Data chaining shall continue until all of the data is transferred for the write operation or until the data chaining condition is not set at the end of a CCW. If the quantity of data sent for the CCW is less than the count in the CCW, the EE bit shall be set to one in the last IU sent for the CCW. For this case, data chaining, if indicated for the CCW, shall not be performed and execution of the channel program shall be suspended until status is received from the control unit. If the quantity of data used by the device is less than the total count provided by the channel, then a status DIB shall be sent; the residual count field shall indicate the amount of data which was not used. If the quantity of data required by the device is greater than the total count provided and the EE bit is not set, then status, if sent, shall have the LRI and RV bits in the status-flag field set to one and the residual count set to zero.

When, at the channel for a read operation, data chaining is indicated in the current CCW, a command IU with a command DIB containing the CCW number, the CCW count and required flags for the next CCW, and a valid token shall be sent to the control unit. The channel shall continue to do this, provided sufficient IU pacing credit is available, for each CCW for which data chaining was specified until the command IUs for all of the CCWs for data chaining have been sent. (See 9.1.4.5.) For a read operation, the sending of the command IUs indicating data chaining does not constitute execution of the command or CCWs; the operation and data chaining shall only be considered to have been initiated. There is an indefinite period of time from the initiation of a read operation with data chaining and when data for that operation arrives at the channel. When a data IU is received by the channel, the CCW number shall indicate the CCW for which this data is being sent. At that point in time, the CCW identified shall become current again at the channel and execution of the CCW shall be considered to be occurring. As each successive data IU is received for a read operation, the channel shall ensure that the CCW number specified corresponds to the appropriate CCW. The quantity of data received for a CCW shall exactly equal the CCW count for that CCW in order for data chaining to occur at the channel. If, before all of the data for a CCW is received, the next data IU received from the control unit specifies a different CCW than the current CCW for the read operation, a device-level protocol error shall be recognized. If either status or a data IU with the EE bit set to one is received, data chaining shall be ended for the current read operation.

When, at the control unit for a read operation, the quantity of data to be sent to the channel exceeds the CCW count in effect from the command IU currently being executed, the E bit shall be set to one in the last data IU sent for the CCW and if the data chaining condition is set, data chaining shall occur at the control unit. When data chaining occurs at the control unit, the next CCW shall go into effect, that is, the contents of the next command IU with the CH bit set to one and a command DIB containing the DU flag set to one shall become current and go into effect for the read operation. The remaining data shall be sent in data IUs under the control of the new CCW. The CCW number from the current command IU shall be used in each data DIB sent, and the CCW count shall determine the maximum quantity of data that may be sent.

Data chaining shall continue at the control unit until all of the data is transferred for the read operation or at the completion of the current CCW, the data chain condition is not set. If all of the data is transferred for the read operation and the data chaining condition is set, data chaining shall not occur, and status shall be sent to the channel. If all of the data has not been transferred to the channel for the read operation and the data chaining condition is not set, data chaining shall not occur and a status DIB, if sent, shall contain the LRI and RV bits in the status-flags field set to one and the residual count set to zero.

When the data-chaining condition is set and the control unit initiates the ending of the I/O operation by transferring channel-end status to the channel, with or without device-end status, and if the quantity of data transferred is less than the CCW count, then the data-chaining indication shall be reset, and chaining shall not be performed.

Data chaining shall occur whenever the data-chaining condition is set, and the chaining (CH) bit in the device-header flag field and the data-chaining-update flag (DU flag) are both set to ones in the command IU.

If the data-chaining condition is set in the control unit and if a command IU is recognized with the chaining bit (CH bit) or the data-chaining-update flag (DU flag) set to zero, then a device-level protocol error shall be detected. If the data-chaining and command-chaining conditions are both not set and if a command IU is received with either the DU flag set to one or the CH bit set to one, then a device-level protocol error shall be detected. The CH bit and DU flag shall be tested before the data-chaining condition is changed to conform to the new value of the chain-data bit in the command-flag field.

If the data-chaining condition is set, the control unit shall ensure that the path to the device remains available when the current count is exhausted, until the next command IU is recognized or until the data-chaining condition is reset.

A channel may minimize the number of data chain updates for successive data chain CCWs. This is accomplished by adding the counts from successive CCWs and presenting only the sum of these counts to the control unit in a single command IU. When this is done, the original integrity of the channel program shall be maintained by ensuring that PCI and command chaining conditions occur at the appropriate points within the channel program.

9.1.4.5 IU Pacing

Each channel provides an IU pacing credit which is initialized at either the start of each channel program or reconnection to continue execution of a channel program. The IU pacing credit is the maximum number of IUs that a channel may send on a given outbound exchange before it receives a command-response IU, which was sent because the CRR bit was set to one, on the existing inbound exchange.

At the start of a channel program, the IU pacing credit shall be set to a model-dependent value no greater than the default value of 16. The control unit may request that the IU pacing credit be increased by the channel at the start of a channel program or at each time the control unit reconnects with device-end status. At the start of a channel program, the control unit may request that the IU pacing credit be increased by providing an IU pacing parameter in either the command response or status sent in response to the first command of a channel program; when reconnecting with device-end status, the control unit may request that the IU pacing credit be increased by providing an IU pacing parameter in the status DIB. See 8.11.2.4 and 8.10.3.4 for a description of how the control unit provides the IU pacing parameter.

If the control unit sets the IU pacing parameter to a value less than or equal to the default value, the channel shall not increase the IU pacing credit above the default value. If the control unit sets the IU pacing parameter to a value greater than the default value, then the channel may increase the IU pacing credit by any amount up to the value indicated by the IU pacing parameter.

At the start of a channel program or at each reconnection, the channel shall send a model dependent number of IUs to the control unit. The number of IUs sent shall not exceed the IU pacing credit value. Prior to or at the last command IU sent, the channel shall request a command response to be returned by setting the CRR bit in a command or command-data DIB. The selection of the command or command-data DIB for the setting of the CRR bit shall be such that the remaining IU pacing credit (that is,

the number of additional IUs the channel is allowed to send before it receives a command-response IU) does not prevent the transmission of all of the IUs for a CCW. For example, if the channel has not set the CRR bit since the command or command-data DIB for which the last command response was received and remaining IU pacing credit is less than the number of IUs required to transfer all of the data indicated by the CCW count field in a command-data DIB, then the CRR bit shall be set to one in the command-data DIB; otherwise the channel shall be unable to proceed with the channel program.

When a command response is received, it shall indicate which CCW is currently being executed and, therefore, the number of IUs that have been processed since the start of the channel program or since the IU for which the previous command response was received. Upon receipt of the command response, the channel is then permitted to send an additional number of IUs beyond the current remaining credit equal to the number of IUs indicated as having been processed.

When a control unit sends a data IU containing a status DIB, the control unit shall discard all command IUs with the SYR bit set to zero and data IUs which are received subsequent to the IU for which the status was sent. (See 8.10.) When a data IU containing a status DIB is received, the channel sets its remaining IU pacing credit to a value equal to the IU pacing credit for the exchange; the number of IUs a channel is then permitted to send, including and subsequent to the IU sent in response to the status, is equal to the IU pacing credit.

When an IU which closes the inbound exchange is received, a channel is allowed to respond to the IU without regard to IU pacing credit.

NOTE 1 Care should be exercised when selecting the command or command-data DIB in which to set the CRR bit. Setting the CRR bit too early results in an excessive number of command-response IUs; setting the CRR bit too late may result in interrupting the flow of command and data IUs on the outbound exchange until a command-response is received. In order to avoid this and as a “rule of thumb” the CRR bit should be set in a command IU as close as possible to the point of sending IUs where half of the IU pacing credit value remains.

NOTE 2 An IU pacing credit higher than 16 is recommended for link speeds above 1 Gbit/s. For distances of up to 100 km, an IU pacing credit of 32 is recommended for a link speed of 2 Gbit/s.

NOTE 3 The intent of the IU pacing function is to prevent data-intensive channel programs such as a channel program performing a data archive operation from unfairly utilizing all of the resources of a control unit. IU pacing is not intended to be a means of flow control.

NOTE 4 A control unit may return control unit busy status if resource limitations prevent the control unit from accepting all of the IUs sent by a channel during the initiation of a new exchange pair.

NOTE 5 The IU pacing function is defined only for transmissions from the channel to the control unit. It is not required for control unit to channel transmissions because resources are already allocated in the channel prior to execution of the channel program.

NOTE 6 A preferred channel implementation makes use of the IU pacing parameter provided by the control unit.

NOTE 7 A preferred control unit implementation provides a means by which the IU pacing parameter may be modified based on operating conditions.

9.1.4.6 Zero CCW count

When a CCW is fetched containing a count field of zeros, the CCW count field of the command IU is likewise set to zeros.

When a command with a CCW count of zero is executed, the type of response sent to the channel, if any, and whether or not command chaining, if indicated, occurs depend on how the command is executed and the flag bits in the command header.

If the first command of a channel program is executed as an immediate operation, a data IU containing a status DIB shall be sent. The LRI and RV bits shall be set to zero. The E-bit shall be ignored for this case. Command chaining, if indicated, shall be initiated by the channel when device-end status is received and the conditions for command chaining at the channel are satisfied.

If the first command of a channel program is executed as a nonimmediate operation, a command-response IU shall be sent. Execution of the command may proceed after the command has been accepted and the command response sent; chaining to the next CCW may proceed if all of the conditions for chaining are satisfied. (See 9.1.6.)

If other than the first command of a channel program is executed, the CRR bit and SSS bit in the command header are both set to zero, and all of the conditions for chaining are satisfied, then execution of the command may proceed, no response shall be sent, and chaining to the next CCW may occur.

If other than the first command of a channel program is executed and the CRR bit and the SSS bit are both set to one, a command-response IU followed by a data IU containing a status DIB shall be sent. In the status DIB, the E bit in the IU header shall be set to zero by the sender and ignored by the recipient, and the RV bit in the status-flags field shall be set to one for a residual count of zero. If the command would have been executed as a nonimmediate operation had a count other than zero been provided, the LRI status flag bit shall be set to one and the residual count is set to zero. Command chaining, if indicated, shall be initiated by the channel when device-end status is received and the conditions for command chaining at the channel are satisfied. (See 9.1.6.)

If other than the first command of a channel program is executed and the CRR bit is set to one and the SSS bit is set to zero, a command-response IU shall be sent. Command chaining, if indicated, occurs at the control unit and status shall not be sent to the channel.

If other than the first command of a channel program is executed and the CRR bit is set to zero and the SSS bit is set to one, a command response shall not be sent but a data IU containing a status DIB shall be sent. In the status DIB, the E bit in the IU header shall be set to zero by the sender and ignored by the recipient, and the RV bit in the status-flags field shall be set to one for residual count of zero. If the command would have been executed as a nonimmediate operation had a count other than zero been provided, the LRI status flag bit shall be set to one and the residual count shall be set to zero. Command chaining, if indicated, shall be initiated by the channel when device-end status is received and the conditions for command chaining at the channel are satisfied. (See 9.1.6.)

The presence of a zero count in the command IU shall not be used to determine whether a command should be executed as an immediate operation, executed as a nonimmediate operation, or rejected by returning unit-check status. For example, when a command-response IU would be returned for the first command of the channel program if the CCW count were nonzero, a command response shall also be returned when the count is zero. This is to ensure that compatibility is maintained among the Fibre Channel interface, SBCON and the parallel-I/O interface.

9.1.5 Ending an I/O operation

9.1.5.1 General rules for ending an I/O operation

The ending of an I/O operation is either channel initiated or control-unit initiated. The channel may initiate the ending of an I/O operation as the result of an abnormal condition or a non-error condition. The control unit may initiate the ending of an I/O operation as the result of the completion of the execution of the command or the transfer of all data associated with the command or as the result of an abnormal condition detected during the execution of the command.

When the channel initiates the ending of the I/O operation, it shall send a control IU indicating one of the following control functions:

a) cancel, (See 9.2.3)

b) selective reset, (See 9.2.5) or

c) system reset. (See 9.2.4.)

When conditions require the ending of an I/O operation with the channel, status in a status DIB shall be sent and all command IUs with the SYR bit set to zero and data IUs received subsequent to the current command shall be discarded.

When conditions permit the ending of an I/O operation without the transfer of status to the channel, chaining shall occur with the start of execution of the next command. (See 8.10.)

In the absence of errors, when the current command is executed as a nonimmediate operation, the control unit shall initiate the ending of the I/O operation when one of the following conditions is satisfied.

a) All of the data specified by the CCW count has been transferred and the data-chaining condition is not set.

b) The data required by the device for the command has been transferred.

When the current command is executed as an immediate operation, the control unit shall initiate the ending of the I/O operation when the conditions for the device allow channel-end status is to be generated.

Ending an I/O Operation without Transferring Status: When execution of the current command reaches completion with the device and conditions not requiring status to be presented to the channel, the I/O operation shall be considered to have ended and execution of the next command in the chain may proceed.

Ending an I/O Operation with Transferring Status: When the control unit ends execution of the current command, including the transfer of data, and conditions require status to be transferred to the channel, a status DIB indicating channel end with or without device end shall be sent along with the appropriate status flags and residual count. If the operation is ended because of an abnormal condition, the appropriate status for this condition is included. If device-end status is included, the operation shall be considered ended by the device when this status is accepted by the channel. If device-end status is not included, the I/O operation shall be considered ended by the device when the control unit later transfers device-end status and receives acknowledgment that the status has been accepted by the channel. If unit-check status is included, the control unit may optionally present supplemental status along with this status, and indicate this by setting the ES bit to one in the IU header.

If the control unit is connected when it is ready to transfer status in order to initiate the ending of the I/O operation with the channel, a status DIB with channel end with or without device-end status shall be sent. (See 8.2.2 for information on when the inbound exchange is closed with the sending of the status DIB.)

If the control unit is not connected when it is ready to transfer status in order to complete the ending of the I/O operation with the channel, a data IU containing a status DIB indicating device-end status shall be sent. The inbound exchange shall be left open.

9.1.5.2 Supplemental status

Supplemental status provides sense information in a status DIB to describe conditions at the control unit or device for which status information containing unit check is being provided.

The control unit may optionally transfer supplemental status along with unit-check status (but not retry status). The presence of the supplemental status shall be indicated by the ES bit being set to one in the IU header of the status DIB containing unit-check status. The channel shall indicate that it has accepted the supplemental status along with the status by setting the ES bit to one in the status-accepted IU. If the channel has accepted the status but not accepted the supplemental status, the ES bit shall be set to zero in the status-accepted IU. In this case, the supplemental status shall be held by the device until retrieved by the channel with sense information by the appropriate sense command, or cleared.

The ability to accept supplemental status is determined by conditions present at the channel and the design of the channel. Whether or not supplemental status is included in a status DIB sent to the channel depends on the status information being transferred and conditions at the control unit, which are model dependent.

NOTE 1 The preferred implementation is for a control unit to provide supplemental status in the status DIB whenever permitted.

NOTE 2 Supplemental status is presently defined only for unit-check status (but not retry status).

9.1.5.3 Residual count

The control unit shall determine the residual count by calculating the difference between the CCW count and the actual number of bytes either written to a device or read from a device and sent to the channel. The value of the residual count is dependent on the CCW count, the number of bytes transferred between the channel and control unit, and the number of bytes actually read from or written to the device.

For a write operation the channel may transfer more data to the control unit than the device requires, but never more data than specified by the CCW count. If the channel transfers all of the data specified by the CCW count and all of the data is used by the device, the residual count shall be equal to zero for this command. If a write command with a nonzero CCW count is executed as an immediate I/O operation, the residual count shall be equal to the CCW count.

When, for a write operation, the device requires less data than specified by the CCW count, the residual count shall be equal to the difference between the CCW count and the actual number of bytes of data used. Alternatively, if the device requires more data than specified by the CCW count, the residual count shall be zero for this command.

For the case where, for a write operation, the channel transfers less data than specified by the CCW count (See 8.11.2.3 and 9.1.4.2.), the residual count shall still be determined by calculating the difference between the CCW count and the actual number of bytes used by the device.

For a read operation only a quantity of data equal to or less than the CCW count shall be transferred to the channel. The residual count shall be the difference between the CCW count and the actual number of bytes sent. If all of the data specified by the CCW count is transferred, the residual count shall be equal to zero for this command. If a read command with a nonzero CCW count is executed as an immediate I/O operation, the residual count shall be equal to the CCW count.

When, for a read operation, less data than specified by the CCW count is transferred to the channel, the residual count shall be equal to the difference between the CCW count and the actual number of

bytes sent. If, for the read operation, all of the data specified by the CCW count is sent but more data would have been sent to the channel had a larger count been specified, the residual count shall be equal to zero.

Note that there are only three cases in which the residual count is greater than zero:

a) when the record length is less than the CCW count,

b) when, during a write operation, the channel is unable to send all of the data for a CCW and has set the EE bit in the last IU sent, and

c) when a command with a non-zero CCW count is executed as an immediate operation.

Table 12 shows the valid combinations for the LRI and RV bits, the residual count, and channel-end status. Also, note that for all cases in which a valid residual count is presented, channel-end status shall also be presented.

The following abnormal situations may require additional flags to be set in the device-header-flags field or may prevent including a valid residual count in a status DIB.

a) The E bit shall be set to one in the device-header-flags field for a status DIB only for a write operation involving an early end indication for which the amount of data received from the channel and written to the device exactly equals the record length.

b) If an abnormal condition prevents determining the exact amount of data either read from or written to a device, a valid residual count shall not be calculated. In this case, the RV bit shall be set to zero in the status DIB. Unit-check status shall be indicated along with channel-end status. Command retry may be requested. See the definition of the CR bit in 9.3.2. If the channel accepts the command retry request, the retry proceeds normally. If command retry is not honored, the channel shall terminate the I/O operation.

c) If, for other than the first command of a channel program, the channel receives a status DIB with the channel-end status bit set to one, the unit-check status bit set to zero, and the residual count valid bit set to zero, a device-level protocol error shall be recognized.

d) If during a read operation, an abnormal condition prevents determining the exact amount of data transferred to the channel, then a valid residual count shall not be calculated. In this case, the RV bit shall be set to zero and unit check along with channel-end status is indicated. Command retry may also be requested.

When, during data chaining, the next CCW takes effect, a new residual count shall be started for the data transfer associated with this CCW. If status is generated and sent to the channel, the residual count shall be included, provided that a valid residual count is calculated.

9.1.6 Command chaining

When there are successive I/O operations to be executed by a channel and a control unit for a single channel program, the execution of these successive I/O operations is referred to as command chaining. When an I/O operation is initiated by the channel, it shall use a command IU to transfer the command, CCW control flags, CCW count, and command flags associated with the current CCW to the control unit. In this manner, the control unit is informed whether or not command chaining is specified for the current CCW. When an I/O operation is completed and the chain-command flag is set to one and the chain-data flag is set to zero, command chaining shall be permitted, provided no abnormal conditions are detected and all other conditions for chaining are satisfied. When command chaining

takes place, the next command and associated flags and count shall become current and the next I/O operation shall be considered to have started. The next command and associated flags and count shall be found in the next command IU received which shall have the chaining bit (CH bit) set to one and the data-chaining-update flag (DU command flag) set to zero.

The control unit shall ensure that command chaining is occurring at the proper times by recognizing a command-chaining condition. The command-chaining condition shall be recognized for each device. The command-chaining condition shall be set as a result of command chaining or as a result of command retry. During command chaining, the command chaining condition shall be set whenever a command with the CC flag set to one and the CD flag set to zero in the CCW-flag field of a command IU is accepted. The command-chaining condition shall be set for command retry when the CH bit is set to one in the IU used by the channel to accept the status that requests a command retry. (See 9.3.2.)

When the command chaining condition is set and the conditions for not sending status at the completion of execution of the current command are all met, command chaining shall be performed by starting execution of the command in the next command IU which shall have the CH bit set to one and the DU command flag bit set to zero. (See 8.10 and Annex D for the conditions when status is sent and when chaining occurs at the control unit.) If the command chaining condition is set but the channel has requested that status be sent at the completion of execution of the current command, status shall be sent to initiate command chaining with the channel. In this case command chaining shall not occur until the channel receives status with device end indicated and the channel sends a command IU for the next command to be executed.

The command-chaining condition shall be reset whenever (1) system reset or selective reset is performed, (2) the device receives a cancel or stack-status in a control IU, (3) status containing channel end but not device end for an I/O operation has been accepted by the channel and the chaining bit is set to zero in the status-accepted IU, (4) status containing device end for an I/O operation has been accepted by the channel using a status-accepted IU, (5) an address-exception condition is recognized and a control IU with a device-level-exception indication is sent, or (6) the channel accepts a valid status DIB with the AS bit set to one and the status contains unit check (except when command retry is requested and the retry requested is to be honored by the channel), unit exception, busy, attention, status modifier without any other bits set, or control-unit end, provided it is not control-unit end alone. A valid status DIB for which the AS bit set to zero shall not suppress command chaining.

Command chaining shall occur whenever the command-chaining condition is set at the device, and the CH bit and the DU bit are set to one and zero, respectively.

If the command-chaining condition is set at the device, the control unit shall ensure that the path to the device remains available when device-end status is presented until the next command is recognized or until the command-chaining condition is reset. If the command-chaining condition is set at the device and if a command IU is recognized with the CH bit set to zero or the DU flag set to one, a device-level protocol error shall be detected. If the command-chaining and the data-chaining conditions are both not set and if a command IU is recognized with the CH bit set to one, then a device-level protocol error shall be detected.

9.2 Device-Level controls

9.2.1 Overview of Device-Level control functions

Device-level controls are used to modify or control the execution of an I/O operation, or to place the control unit and device in a known state. Most device-level controls are specified by combinations of bits in the device-header flag field and the IUI field in the IU header. For command and control IUs, device-level controls are also specified in the command or control DIB. For other IU types, device-level controls are specified only by the device-header flag field and IUI field.

9.2.2 Stacking Status Function

The stack-status control function shall indicate that the channel does not accept the status; the control unit or device shall hold the status information (status flags, status byte, and status parameter) and associated supplemental status, if any, until requested by the channel or until status information is cleared, withdrawn, or unstacked. In addition, a control unit may unstack any status which has remained stacked for at least 1 minute. In certain unusual control-unit-recovery situations, such as when the control unit has lost the indication of whether status should or should not be unstacked for a particular device, the control-unit-recovery action may possibly cause some stacked status to become not stacked without a request-status IU, and subsequently that status may be presented by the control unit. A channel shall be allowed to stack any status sent in an unsolicited data IU. Also, a channel shall be allowed to stack any status sent in a solicited data IU unless the status is received while the channel program is being executed at the channel and the busy bit is set to a zero. If a stack-status IU is received in response to a status transfer which the channel is not allowed to stack, a device-level protocol error shall be detected.

Stacking of status other than control-unit end or control unit busy shall cause the suppression of chaining when chaining is in progress.

The busy bit shall not be considered part of the status to be held by the control unit or device when the busy bit has been set to one in order to indicate that the status being presented to a command was already pending.

The control unit shall reset the data-chaining or command-chaining condition when it receives a stack-status IU in response to a status transfer and the status byte does not contain either (1) the control-unit-end status bit set to one and all other status bits set to zeros, or (2) the busy and status-modifier bits set to ones and all other status bits set to zeros. (See 9.1.4.4 and 9.1.6.)

Status shall be stacked and unstacked with respect to a specific logical path. Status that is not associated exclusively with the current logical path may be presented using another logical path for which it either has been unstacked or has not been stacked.

The channel may request pending status or request the unstacking of status by using the request-status control function. (See 9.2.6.)

If conditions change at the control unit or device which affect status information that is being held as a result of receiving stack status, the most current information shall be sent in a data IU containing a status DIB (when permitted by receipt of a request-status IU) or in response to a command IU.

When the control unit has sent a data IU containing a status DIB and left the inbound exchange open, the control unit shall indicate to the channel that it recognized the stack status by sending a DACK IU. The channel shall consider the stack-status function to have been performed when it receives the DACK IU.

When the control unit has sent a data IU containing a status DIB and closed the inbound exchange, the channel shall consider the stack-status function to have been performed when it sends the stack-status IU.

If a control unit recognizes a stack-status IU in response to a status DIB for which the AS bit is set to zero, the control unit shall do one of the following.

a) If the status was control unit busy, the status shall be withdrawn. In addition, control-unit-end status shall not be owed unless control-unit-busy status was previously accepted by the channel.

b) If the status was control-unit end alone, the status shall remain pending and unstacked.

c) If the status was control-unit end with busy, the busy bit shall not be retained, and the control-unit end shall remain pending and unstacked.

d) If the status contained control unit busy along with control-unit end, the status shall be withdrawn. In addition, control-unit-end status shall not be owed unless control-unit-busy status was previously accepted by the channel.

When status information is caused to be stacked for one device and the control unit is no longer connected to the channel, the transfer of pending not-stacked status for a different device shall be permitted.

When status signaling the ending of an I/O operation has been previously stacked at a device, a Request Status IU may be sent to retrieve this status only if no other I/O operations have been initiated with the subchannel subsequent to the stacking having occurred at the device. If a channel initiates an I/O operation with a device for which status for a previous I/O operation has been stacked but not retrieved, initiative for sending a Request-Status IU to retrieve the stacked status for the previous I/O operation with the device shall be discarded or suppressed until the subchannel again becomes idle.

9.2.3 Cancel function

The cancel function shall cause the designated device to terminate execution of the current operation, if any. When the channel has initiative to send a cancel IU during data transfer for a write operation, it shall stop data transfer for the write operation. The last IU sent for the data transfer shall contain either a command-data DIB or data DIB with the E or EE bit set to one, and contain a CRC field.

If the channel has initiative to send a cancel IU during an existing connection for an I/O operation, but is unable to perform the cancel function because of inadequate pacing credit, an SB-2 exchange error may be recognized if the function remains pending at the channel after 5 s have elapsed. See 10.5.6 for additional information.

When an I/O operation is terminated by the cancel function, the device shall proceed to its normal ending point (including mechanical motion) and, as a result of having gone to its normal ending point, the device shall generate channel-end and device-end status, as appropriate, for the I/O operation. If an I/O operation is neither being initiated nor in progress, the cancel function shall cause no action at the device.

When the cancel function is performed and the device address specified corresponds to the device for which the control unit is connected to the channel, a DACK IU shall be sent to the channel and the cancel function shall be performed for the specified device after performing one of the following actions, if applicable.

a) If the CCW being terminated by the cancel is a write operation and the device has used any of the data for the CCW, CRC shall be checked on the data it received from the channel provided that CRC checking is provided and a valid CRC was received.

b) If the CCW being terminated by the cancel is for a read operation, any data transfer in progress shall be stopped at an appropriate point and a CRC field shall be provided in the last data DIB with the E or EE bit set to one.

The DACK IU sent shall contain the CCW number and token associated with the command terminated by the cancel. If the cancel function occurs after the completion of execution of a command but before the start of execution of the next command, the CCW number and token associated with this next com-

mand, if available, shall be used in the DACK; otherwise the CCW number and token from the previous command shall be used in the DACK. The channel shall consider the cancel function to be performed when it receives the DACK IU in response.

When the cancel IU is received on the outbound exchange of an exchange pair and the device address specified corresponds to a device for which a solicited data IU containing a status DIB has been sent which closed the inbound exchange of the exchange pair, no response shall be sent. In this case, if an I/O operation is in process at the device, the control unit shall perform the cancel function for the specified device. The channel shall accept the status and no response shall be sent. Initiative to perform the cancel function shall continue to exist at the channel.

When the cancel IU is received and the device address specified does not correspond to the device for which the control unit is connected to the channel, the DACK IU shall not be sent, and a device-level protocol error shall be detected.

When the cancel IU is received and no connection to the channel is considered to exist, the DACK IU sent shall contain the CCW number and token from the cancel IU for the specified device. The channel shall consider the cancel function to be performed when it receives the DACK IU response.

The cancel function shall not cause any pending or stacked status to be cleared. If the cancel function is performed after the sending of status containing status modifier and busy but before the receipt of the status-accepted IU, then the status shall be withdrawn.

When the cancel function is performed, chaining shall be suppressed for the affected I/O operation.

When the cancel function is performed by a device before the device considers an I/O operation in progress or as being initiated, no action shall be taken by the device. Stacked or pending status for that device shall not be affected. The device shall not become busy as a result of performing the cancel function. The following conditions shall be included:

– the control unit is not connected to the channel, and no operation is considered as either being in progress or being initiated with the specified device, and

– the control unit is connected to the channel, has sent a status DIB for asynchronous status, and has not received a status-accepted response.

When the cancel function is performed after an I/O operation is considered to be in progress at the device, the I/O operation shall be terminated; status containing channel end, device end, or both, as appropriate shall be generated as a result of the device proceeding to its normal ending point. (See 9.1.3 and 9.1.5.) In this case, the presentation of the channel-end or device-end status shall require a new connection to the channel.

If channel end is included in this status, then the residual count shall also be included in the status DIB with the RV bit set to one, unless conditions prevent transfer of a valid residual count, in which case the RV bit is set to zero, and unit check is included in the status. If the LRI bit is set to one with the channel-end status, and the residual count is zero, the long-record condition shall be ignored by the channel. The device shall remain busy until all status for the I/O operation is accepted by the channel. Any abnormal condition shall be indicated by unit check in the status, and the sense information shall provide additional details concerning the operation. The cancel function shall cause an I/O operation to be terminated at the control unit for the following conditions.

– The control unit is connected to the channel executing an I/O operation and the conditions requiring ending status to be sent are satisfied, but status containing channel-end status has not been sent or, if sent, has not yet been accepted by the channel.

– The control unit is connected to the channel executing a chain of I/O operations and the conditions for not transferring status are all satisfied, but the current I/O operation has not completed and chaining to the next command has not yet occurred.

– The control unit is connected to the channel as a result of a request status, and status containing channel-end or device-end status for an I/O operation has been sent but not yet accepted by the channel.

– Status containing channel-end or device-end status for an I/O operation has been sent but not yet accepted by the channel.

If the cancel function is performed while the device is in the process of sending an IU, the device shall finish sending that IU and in the case of data transfer, any additional IUs required in order to provide a valid CRC; however, any additional data beyond that which is required in order to provide a valid CRC shall not be sent.

When the channel sends a cancel IU during an existing connection for an I/O operation, there may be one or more IUs already in transit to the channel that normally would have been part of the I/O operation had the cancel not been sent. In this case, valid data IUs containing a data DIB and command-response IUs that are received after the cancel IU is sent but before an IU which closes the inbound exchange is received shall be accepted and processed in the same manner as if the cancel had not been sent.

Except when dynamic reconnection is being used, performing the cancel function shall not affect I/O operations or status associated with a logical path other than the logical path over which the cancel IU was received. (For devices using dynamic reconnection, see 9.5.2.)

9.2.4 System-Reset function

The system-reset function shall be performed by the control unit with respect to a logical path whenever any of the following conditions is recognized by the control unit.

– A valid system-reset IU is received on the logical path. A system-reset IU is always sent with the AS bit set to zero.

– The logical path is removed or established.

– Certain internal recovery procedures occur within the control unit.

In addition, whenever a logical path does not exist, the control unit and all attached devices shall appear as if they had been in the reset state with respect to the logical path during the time when the logical path did not exist.

A system-reset IU shall always be sent on a new exchange as an initiation IU. IUs received on other open exchanges for the same logical path shall be discarded by the channel.

After a valid system-reset IU is received from the channel, the control unit, for the same logical path, shall close all exchanges except the one on which the system reset was received. The control unit shall close all inbound and outbound exchanges by sending the ABTS basic link service frame. When all open exchanges for the logical path are closed, a DACK IU shall be sent on a new exchange to the channel. The CCW number and token from the system-reset IU shall be used in the DACK IU sent.

After receiving the DACK IU but before sending the LACK IU, the channel shall send ABTS to close any exchanges which remain open on the logical path except for the exchange on which the system

reset was sent. When a system-reset IU is sent, the channel shall consider the system-reset function performed after the DACK IU is received and the LACK IU is sent to close the outbound exchange.

Except for the closing of exchanges on the logical path on which the system reset was sent, no FC-PH functions shall be performed or reset as a result of the system reset function.

A system reset, when performed by the control unit, shall cause a reset for all devices, I/O operations, and pending status with respect to the logical path. The interpretation of the reset state of a device is model dependent. A system reset may possibly reset forthcoming status which is owed to the path signaling reset but not yet pending, such as no-longer-busy status. Any I/O operation in progress shall be terminated, and the device shall proceed to a normal mechanical stopping point, if applicable. No further IUs shall be sent pertaining to the I/O operation that was reset.

While the system reset is either pending or being performed by the control unit, busy status may be returned in response to any command IU from the logical or physical path with which the system reset is associated. While system reset is being performed, busy status may be returned in response to any command IU from any path. If busy status is returned in response to a command IU during one of these intervals, then appropriate no-longer-busy status shall be made pending for the logical path when the busy condition no longer exists.

The system reset associated with a logical path shall not directly affect I/O operations or status associated with any other logical path. However, if the status for another logical path has not been presented because of an inhibiting condition and the reset shall clear the inhibiting condition, then previously owed or pending status associated with that other logical path may be presented along that other logical path after the reset.

The ready or not-ready state of the control unit or device is generally not changed by a system reset. When, however, the online/offline switch was changed before the reset but is not yet effective because of required inhibiting conditions, the ready or not-ready state may change if the reset clears those inhibiting conditions.

A system reset shall not reset activity that occurs logically subsequent to the moment that system reset is recognized by the control unit.

If a system reset terminates an active I/O operation, but leaves the device busy until the end of mechanical motion, then device-end status or device-end and unit-check status at the end of mechanical motion may be presented. If presented, this status may be presented using any logical path which is established with respect to the device. The CCW number field and token field for this Status DIB shall be set to zero.

When a system reset is performed, a resetting-event condition shall be generated for each affected device. (See 9.4.)

9.2.5 Selective-Reset function

The selective-reset function shall cause a device and its status to be reset with respect to the particular logical path. Only the device and certain allegiances associated with the designated logical path shall be reset. The definition of the reset state of the device is model dependent.

No FC-PH functions are reset as a result of the selective-reset function.

The selective-reset IU shall pass information to the device which results in one of two types of action.

a) When one or more of the RC, RU, and RO bits is set to one, the channel shall be requesting that command retry be requested or that the device reply with status containing unit check. The device shall signal that it is capable of complying with the request by returning status with the CI bit set to one in a status DIB. (See 9.3.3.)

b) When the device neither requests the indicated retry nor returns the indicated status, the device shall return a DACK IU and perform the selective-reset operation.

When a device performs the selective-reset operation, the device and its status shall be reset. Any I/O operation in progress for the device on that logical path shall proceed to a normal ending point, if applicable, with no further data transfer. All unexecuted command IUs and their associated data shall be discarded. If an IU is in the process of being sent, the control unit shall finish sending that IU. No further IUs shall be sent pertaining to the I/O operation that was reset.

Performing the selective-reset operation may reset forthcoming status which is owed to the path signaling reset but not yet pending, such as no-longer-busy status.

If the selective-reset IU is received on the outbound exchange of an exchange pair at a time when the inbound exchange has been closed, the selective-reset operation shall be performed and no response shall be sent to the channel; if the selective-reset IU is received on an outbound exchange of an exchange pair at a time when the inbound exchange of the exchange pair exists, the control unit shall indicate to the channel that the selective-reset operation has been or is to be performed by sending a DACK IU. The DACK IU sent shall contain the CCW number and token associated with the command terminated by the selective-reset operation. If the selective-reset operation is performed after the execution of a command has completed normally at the control unit and chaining is indicated but has not yet occurred, the CCW number and token used in the DACK shall be those of the next command to be executed, if available; otherwise, the CCW number and token from the previous command shall be used in the DACK. If the selective-reset operation is performed when either a channel program is not in progress with the device or the device has disconnected from the channel, the control unit shall indicate to the channel that the selective-reset operation has been or is to be performed by sending the DACK IU. The DACK IU sent shall contain the CCW number and token from the selective-reset IU.

When the channel sends a selective-reset IU when no connection to the device exists, the channel shall consider the selective-reset function to be performed when it receives the DACK IU.

When the channel sends a selective-reset IU during an existing connection for an I/O operation, there may be one or more device-level IUs already in transit from the control unit to the channel that normally would have been part of the I/O operation had the selective-reset IU not been sent. Valid IUs that are received after the selective-reset IU is sent but before an IU which closes the inbound exchange is received shall be ignored by the channel. If a data IU containing a status DIB which closes the inbound exchange is received, the channel shall process the status as if it had been received before the selective-reset IU was sent, and no response shall be sent to the control unit. Initiative to perform the selective-reset function shall continue to exist at the channel.

While the selective-reset operation is pending or is being performed by the control unit, busy status may be returned in response to any command IU from the logical path which initiated the selective-reset operation. If the selective-reset operation is being performed by the control unit, busy status may be returned in response to any command IU from any path. If busy status is returned in response to a command IU during one of these intervals, then appropriate no-longer-busy status shall be made pending for the logical path when the busy condition no longer exists.

If performing the selective-reset operation results in the termination of an active I/O operation, but leaves the device busy until the end of mechanical motion, then device-end status or device-end and

unit-check status at the end of mechanical motion may be presented. If presented, this status may be presented to any logical path which is established with respect to the device.

The ready or not-ready state of the control unit or device is generally not changed when performing the selective-reset operation. When, however, the online/offline switch was changed before performing the selective-reset operation, but is not yet effective because of certain inhibiting conditions, the ready or not-ready state may change if performing the selective-reset operation clears those inhibiting conditions.

Performing the selective-reset operation shall not reset any activity that occurs logically subsequent to the moment that selective-reset operation is recognized by the control unit.

NOTE 1 The architecture allows the option of resetting or not resetting forthcoming status, which is status owed to the path signaling reset but not yet pending, such as no-longer-busy status. The preferred implementation is to reset any forthcoming status, that is, status such as device-end status for no longer busy or for the end of an I/O operation.

NOTE 2 Devices should not generate status to notify the program that the selective-reset operation was performed.

9.2.6 Request-Status function

The request-status function shall indicate that the channel is prepared to have pending status information presented for the addressed device. The status shall be associated with the logical path on which the request-status IU was received. Request status shall have no effect on status associated with logical paths other than that on which the request status was received. The request-status control function may be used to obtain stacked status.

When the control unit accepts a request-status IU, it shall unstack any status previously stacked for the specified device address. The unstacked status may be sent in response to the request status or it may be sent at a later time. If the unstacked status is sent in direct response to the request status, the CCW number and token from the request-status IU shall be used for the status transfer. If status is not sent, a DACK IU shall be sent and the status remains pending. The CCW number and token from the request-status IU shall be used in the DACK sent.

If the control unit receives a request-status IU but is unable to honor it, the control unit shall send a status DIB with the AS bit set to zero, the status-modifier status bit set to one, and the busy status bit set to one, indicating a control-unit-busy condition. If the channel accepts status indicating a control-unit-busy condition, control-unit end shall be owed. If the channel responds with a stack-status IU, control unit busy shall be withdrawn, and control-unit end shall not be owed unless it was previously owed.

If the control unit receives a request-status IU and the designated device is not installed, one of the following shall occur:

a) If pending status exists for the designated device, the control unit shall unstack the status if it was previously stacked and optionally transfer the pending status information to the channel. If status is not returned in direct response to the request status, a DACK IU shall be sent.

b) If pending status does not exist for the designated device, the control unit shall either recognize an address-exception condition and send a device-level-exception IU, or a DACK IU. (See 9.2.7 and 9.2.9.)

9.2.7 Device-Level-Exception function

The device-level-exception control function shall indicate certain abnormal conditions. For the use of the device-level-exception control, see 9.3.4.

9.2.8 Status-Acceptance function

The channel shall indicate that status is accepted in several ways, depending on whether the status is for an I/O operation, whether the channel intends to perform command chaining, and, if command chaining is to be performed, whether the status contains device end.

If the status is considered by the channel to be unrelated to an I/O operation or if the channel does not intend to perform command chaining, the channel shall indicate that status is accepted by sending a status-accepted IU with the chaining (CH) bit in the device-header flag field set to zero.

If the channel intends to perform command chaining and the status DIB has the channel-end status bit set to one and the device-end status bit set to zero, the channel shall indicate that status is accepted by sending a status-accepted IU with the CH bit set to one.

If the channel intends to perform command chaining and the status DIB has the device-end status bit set to one, the channel shall indicate implicitly that the status is accepted by sending the next command in a command IU with the CH bit and the SYR flag set to one. (See 9.1.6.) The control unit shall consider the status to have been accepted when it receives the command IU with the CH bit and the SYR flag set to one.

If a status-accepted IU is sent in response to a status DIB with the ES bit set, indicating that supplemental status is also present, the acceptance of the supplemental status shall be indicated by setting the ES bit to a one in the status-accepted IU. If the channel accepts the status but not the supplemental status, the ES bit shall be set to a zero in the status-accepted IU. In this case, the supplemental status shall be held by the device until retrieved by the channel as sense information by the appropriate sense command, or cleared.

When the control unit has sent a data IU containing a status DIB, left the inbound exchange open, and received a status-accepted IU in response, the control unit shall indicate to the channel that it recognizes the status-accepted IU by sending a DACK IU. The control unit shall consider the status to have been accepted when it sends the DACK IU. The CCW number and token from the status-accepted IU shall be used in the DACK sent. The channel shall consider the status-accepted function to have been performed when it receives the DACK IU.

When the control unit has sent a solicited data IU containing a status DIB and closed the inbound exchange, the control unit shall consider the status to have been accepted when it sends the status. The channel shall consider the status to be accepted when it sends the status-accepted IU.

If the control unit considers that command chaining is suppressed or not indicated, or the control unit has presented status not related to an I/O operation and then detects the CH bit set to one in either a status-accepted IU or a command IU which was sent by the channel in response to the status DIB, then the control unit shall detect a device-level protocol error and shall not allow command chaining to occur. If a command IU with the CH bit and SYR flag set to one is received in response to a status DIB that had the device-end status bit set to zero, a device-level protocol error shall be detected and command chaining shall not occur. If the control unit receives a status-accepted IU with the CH bit set to one in response to a status DIB that had the device-end status bit set to one, a device-level protocol error shall be detected and command chaining shall not occur.

9.2.9 Device-Level-Acknowledgment function

The device-level-acknowledgment (DACK) function shall indicate that the control unit has recognized a corresponding stack-status, cancel, system-reset, selective-reset, request-status, or status-accepted IU. For this use of the DACK function, refer to

– 9.2.2 for the stack-status function,

– 9.2.3 for the cancel function,

– 9.2.4 for the system-reset function,

– 9.2.5 for the selective-reset function,

– 9.2.6 for the request-status function, and

– 9.2.8 for the status-acceptance function.

9.2.10 Control-Unit-Busy condition

Once a logical path is established, all IUs which contain device-level information that initiate a connection which are received by a control unit over that logical path may encounter a busy condition that prevents the acceptance of any device-level function except for a system reset or purge path. When the device-level function is not accepted due to a control-unit-busy condition, a data IU with the AS bit set to zero and with a status DIB containing only the status-modifier and busy status bits set to ones shall be sent. This status is called control unit busy.

The control unit may return control-unit-busy status when all of the following conditions are met.

a) There are no errors.

b) The control unit considers no prior connection to exist for the same channel program.

c) An IU containing device-level information is received that initiates a connection.

d) The control unit is incapable of recognizing the function contained in the IU due to a control-unit-busy condition.

e) The IU is not a valid system-reset or purge-path IU.

If the status is regarded as accepted, then when the busy condition no longer exists, control-unit-end shall be sent to the destination that accepted the status byte with the control-unit-busy indication. If the response to the status transfer is a stack-status IU, then the status shall be withdrawn and the control unit shall proceed as though the status was never sent. If the response to the status DIB is anything else, then a device-level protocol error shall be detected.

9.3 Error handling at the device level

9.3.1 Purge path function

When link-level recovery results in the aborting of an outbound exchange at the channel for a known logical path and device, the channel shall have initiative to send a purge-path IU on a new outbound exchange. The purge path shall be sent in an unsolicited control IU and is the first device-level recovery action taken. The purge-path IU shall be sent on the same logical path and for the same device as

was used for the outbound exchange that was aborted. If, for the same logical path and device, an inbound exchange is opened before either the purge-path IU is sent or after the purge-path IU is sent but before a purge-path-response IU is received, the IU opening the inbound exchange shall be discarded. This inbound exchange shall be aborted by the channel after the purge-path IU is sent. The aborting of the inbound exchange by the channel shall not affect the requirement to receive a purge-path-response IU.

If, in the interval after receiving initiative to send a purge-path IU but before receipt of a purge-path-response IU, the channel receives an IU on the same logical path but either with the AS bit set to zero or with the AS bit set to one but for a different device address, the IU shall be accepted. Acceptance of the IU shall not imply acceptance of the function or operation; only that the channel treats this IU as being independent of the recovery action in progress and that aborting this inbound exchange is not performed. The channel shall continue to wait for the purge-path IU response on a new inbound exchange.

If after the channel sends a purge-path IU an error free purge-path-response IU is not received before a PTOV timeout, link-level recovery for an exchange error shall be performed. (See 11.2.6.) A purge-path IU shall be retried once and if still unsuccessful, initiative to perform device-level recovery shall be removed and the operation or function being attempted shall be terminated.

When a control unit receives a purge-path IU, any inbound or outbound exchange still open for the same logical path and device shall be closed by the sending of ABTS and a purge-path-response IU is sent on a new inbound exchange. A purge-path IU shall not result in a reset for the device nor shall it reset any operation, function, or status for the device. After a purge-path-response IU is sent and the required LACK closing the outbound exchange received, any recovery action, operation, or function pending for the logical path and device may resume.

9.3.2 Command retry

A control unit may request command retry for a command either when an error detected at the control unit prevents execution of the command or when an error detected at the channel results in the control unit receiving a channel-initiated-retry request and the request is accepted.

When command retry is performed by the channel as the result of receiving a command retry request from the control unit, the command specified by the CCW number received with the command retry request shall be re-sent. The command IU shall have the CH bit and the SYR bit set to one and the DU bit set to zero. If the CCW number received is not valid or does not correspond to a CCW that is able to be made current again at the channel, the status shall not be accepted, a selective reset without a request for command retry shall be sent, and execution of the channel program shall be terminated.

A control unit may request command retry anytime after accepting a command and before status indicating channel end has been sent and regarded as accepted. (See 9.2.8.)

Once channel-end status has been accepted for a command, that command and all previous commands shall no longer be retried by the channel.

When command retry is requested, the CCW number and token associated with the command to be retried shall be provided by the control unit. If the CCW number or token is not able to be identified for the command, a command retry request shall not be permitted. A control unit shall determine which command to retry based on conditions at the control unit and whether the request for retry is initiated by the channel or control unit.

If retry is initiated by the channel, the command to be retried may be specified through a valid CCW number sent with the channel-initiated-retry request. In this case, if the CCW number provided is associated with a command that is able to be retried at the device, this CCW number and the associated token shall be returned in the command retry request sent to the channel. If a CCW number of zero is provided with the channel-initiated-retry request, the command to be retried shall be determined by the control unit. The control unit shall determine the command to be retried based on which commands have been executed and resulted in either a data or control IU being sent to the channel and on model dependent criteria. The last command for which channel-end status has not yet been accepted and which resulted in either a data or control IU being sent to the channel shall become the furthest point back in the channel program that is able to be retried at the channel. In determining the command on which to start retry, the control unit may select either the current command being executed or any previous command executed, back to and including the last command for which a data or control IU was sent.

If retry is initiated by the control unit, the command to be retried shall be determined based on which commands have been executed and resulted in either a data or control IU being sent to the channel and on model dependent criteria. In determining the command on which to start retry, the control unit may select either the current command being executed or any previous command executed back to and including the last command for which a data or command IU was sent and for which status containing channel end is not regarded as having been accepted.

To request command retry, the control unit shall send a status DIB with the status byte containing retry status and with the CR bit set to one. The status byte shall contain retry status if it contains unit check and status modifier together with 1) channel end alone (meaning the control unit or device is not yet ready to retry the command), or 2) channel end and device end (meaning the control unit and device are prepared for immediate command retry).

Command retry may be immediate or deferred, depending on whether device end is part of the retry status in the status byte. If device end is part of the retry status, then the device shall be immediately ready to perform command retry. The channel shall accept the immediate-retry request by responding with a command IU with the CH bit and SYR bit set to one and the DU bit set to zero. If the channel responds with a status-accepted IU, retry shall not be performed, and command chaining shall be reset.

If the retry status does not include device end, then the retry shall be deferred until the device sends in device end, with or without status modifier, in a subsequent status DIB. If the channel accepts the request for deferred command retry, the channel shall respond with a status-accepted IU that has the CH bit set to one. Later, when the device is ready to perform the command retry, device-end status with the CR bit set to zero shall be sent to the channel. If the channel accepts the status and performs command retry, the requested command shall be re-sent in a command IU with the CH bit and SYR bit set to one and the DU bit set to zero. If conditions at the channel have changed so that the channel is no longer able to retry the command, or if the status presented contains anything besides device end or device end and status modifier, the channel shall accept the status by means of a status-accepted IU with the CH bit set to zero. If the channel responds to the device-end status with the status-accepted IU, command retry shall not be performed, and command chaining shall be reset.

A channel shall have the capability to perform command retry; however, a control unit may or may not provide command retry.

It is left as an option of the control unit whether to verify if the command returned via command retry is in fact the same command in effect when command retry was invoked. If the control unit performs