# ISO/IEC 14165-122

Edition 1.1 2008-11

# INTERNATIONAL STANDARD

## Information technology – Fibre channel – Part 122: Arbitrated loop-2 (FC-AL-2)

ISO/IEC 14165-122:2005+A1:2008(E)

Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

**THIS PUBLICATION IS COPYRIGHT PROTECTED**

**Copyright © 2008 ISO/IEC, Geneva, Switzerland**

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester.

If you have any questions about ISO/IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

## About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

## About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

- Catalogue of IEC publications: www.iec.ch/searchpub
The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub
Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org
The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv
If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00

Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

# ISO/IEC 14165-122

Edition 1.1 2008-11

# INTERNATIONAL STANDARD

## Information technology – Fibre channel – Part 122: Arbitrated loop-2 (FC-AL-2)

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE CB

ICS 35.200

ISBN 2-8318-9775-0

Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

CONTENTS

FOREWORD ... 8
INTRODUCTION ... 10
1 Scope ... 12
2 Normative references ... 12
3 Definitions and conventions ... 13
3.1 Definitions ... 13
3.2 Editorial conventions ... 16
3.3 Abbreviations, acronyms and other special words ... 16
3.4 Symbols ... 19
4 Structure and concepts ... 19
4.0 Purpose ... 19
4.1 Overview ... 19
4.2 General description ... 20
4.3 Access fairness algorithm ... 22
4.3.0 Access fairness overview ... 22
4.3.1 Access fairness for NL_Ports ... 22
4.3.2 Access unfairness for NL_Ports ... 22
4.3.3 Access unfairness for FL_Ports ... 23
4.4 Relationship to FC-PH ... 23
5 Addressing ... 25
5.1 Arbitrated Loop Physical Address (AL_PA) ... 25
5.1.1 Valid AL_PAs ... 27
5.1.2 Special AL_PAs and flags ... 27
5.2 Native address identifier ... 27
6 FC-AL Ordered Sets ... 28
7 FC-AL Primitive Signals and Sequences ... 29
7.0 Overview ... 29
7.1 Arbitrate Primitive Signals (ARByx) ... 29
7.1.0 ARByx overview ... 29
7.1.1 ARB(AL_PA) ... 29
7.1.2 ARB(F0) ... 29
7.1.3 ARB(FF ... 30
7.2 Open Primitive Signals (OPNy) ... 30
7.2.0 OPNy overview ... 30
7.2.1 Open full-duplex (OPNyx) ... 30
7.2.2 Open half-duplex (OPNyy) ... 30
7.3 Open Replicate Primitive Signals (OPNr) ... 31
7.3.0 OPNr overview ... 31
7.3.1 Open selective replicate (OPNyr) ... 31
7.3.2 Open broadcast replicate (OPNfr) ... 31
7.4 Close Primitive Signal (CLS) ... 32
7.5 Dynamic Half-Duplex Primitive Signal (DHD) ... 32
7.6 Mark Primitive Signal (MRKtx) ... 32
7.7 Loop Port Bypass/Enable Primitive Sequences ... 33

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

7.7.0 LPE/LPB overview ... 33
7.7.1 Loop Port Bypass (LPByx) ... 33
7.7.2 Loop Port Bypass all (LPBfx) ... 34
7.7.3 Loop Port Enable (LPEyx) ... 34
7.7.4 Loop Port Enable all (LPEfx) ... 34
7.8 Loop initialization primitive sequences (LIP) ... 35
7.8.0 LIP overview ... 35
7.8.1 Loop Initialization — no valid AL_PA ... 35
7.8.2 Loop Initialization — Loop Failure; no valid AL_PA ... 35
7.8.3 Loop Initialization — valid AL_PA ... 35
7.8.4 Loop Initialization — Loop Failure; valid AL_ ... 35
7.8.5 Loop Initialization — reset L_Port ... 35
7.8.6 Loop Initialization — reserved ... 35
8 L_Port operation ... 36
8.0 Overview ... 36
8.1 History ... 36
8.1.1Access fairness history ... 36
8.1.2Duplex mode history ... 37
8.1.3Replicate mode history ... 37
8.1.4Operational mode history ... 37
8.1.5DHD received history ... 38
8.1.6ARB(FF) history ... 38
8.2 Timeouts ... 38
8.2.1FC-PH timeout values ... 38
8.2.2Arbitrated Loop timeout value ... 38
8.2.3Loop timeout ... 39
8.3 Operational characteristics ... 39
8.3.1Transmission Word ... 39
8.3.1.1Power-on Transmission Words ... 39
8.3.1.2Invalid Transmission Words and Transmission Characters ... 39
8.3.2Clock skew management ... 39
8.3.3Error detection and recovery ... 40
8.3.4BB_Credit and Available_BB_Credit ... 40
8.3.4.0BB_Credit overview ... 40
8.3.4.1BB_Credit management per Loop circuit ... 41
8.3.4.2Available_BB_Credit management per Loop ... 42
8.4 Loop Port State Machine (LPSM) ... 42
8.4.0 LPSM overview ... 42
8.4.1 State names ... 42
8.4.2 State diagram ... 43
8.4.3 Reference items ... 45
9 L_Port state transition tables ... 61
10 Loop Initialization procedure ... 86
10.0 Loop Initialization overview ... 86
10.1 Loop Initialization summary ... 86
10.2 Loop Initialization introduction ... 87
10.3 Loop Initialization timers ... 87

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

10.4 Node-initiated L_Port initialization ... 88
10.5 L_Port initialization ... 88
10.5.0 Initialization overview ... 88
10.5.1 Loop Initialization Sequences ... 89
10.5.2 Assigned AL_PA values ... 90
10.5.3 Loop Initialization steps ... 92
10.5.4 Loop Initialization state diagram ... 96
10.5.4.0State diagram overview ... 96
10.5.4.1Validity of AL_PA ... 98
10.5.4.2POWER-ON state diagram ... 100
10.5.4.3OLD-PORT state diagram ... 101
10.5.4.5Normal Initialization state diagram ... 105
10.5.4.7Slave Initialization state diagram ... 110
10.5.4.8Slave AL_PA position map state diagram ... 113
10.5.4.9Master Initialization state diagram ... 115
10.5.4.10Master AL_PA position map state ... 117
Annex A (normative) L_Port Elasticity buffer management ... 119
A.0Overview ... 119
A.1L_Port elasticity buffer implementation ... 119
A.2Clock skew management ... 119
A.3Clock skew management states ... 120
A.3.0Clock skew overview ... 120
A.3.1Insertion pending ... 120
A.3.2Quiescent ... 120
A.3.3Deletion pending ... 120
A.3.3.1Low priority deletion pending ... 121
A.3.3.2 High priority deletion pending ... 121
A.4Buffer size ... 122
Annex B (informative) Loop Port State Machine examples ... 123
B.0Overview ... 123
B.1L_Port initialization example ... 123
B.2N_Port Login example ... 124
Annex C (informative) Dynamic Half-Duplex ... 126
C.0DHD Overview ... 126
C.1Close initiative description ... 126
C.2Dynamic Half-Duplex examples ... 127
Annex D (informative) Access unfairness ... 128
D.0Overview ... 128
D.1Improving Loop performance ... 128
D.2Emptying ACK ... 128
Annex E (informative) Half-duplex operation ... 129
Annex F (informative) BB_Credit and Available_BB_Credit management example ... 130
L_Port A ... 130
L_Port B ... 130

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

Annex G (informative) L_Port clock design options....132
G.0Overview....132
G.1L_Port synchronous clock design....132
G.2L_Port asynchronous clock design....132
G.3Clock skew management function periodicity....133
Annex H (informative) Mark synchronization examples....134
H.0Overview....134
H.1Clock synchronization....134
H.2Disk spindle synchronization....134
Annex I (informative) Port Bypass Circuit example and usage....136
I.0Overview....136
I.1Port Bypass Circuit....136
I.1.0Overview....136
I.1.1Default bypass....137
I.1.2Power-on reset bypass....137
I.2Using a Port Bypass Circuit....137
I.2.1Diagnostic Test of the Port Bypass Circuit....137
I.2.2Recovery from Loop Failure....138
I.2.3Power-on with a failing L_Port....138
I.2.4Reconfiguring a Loop with LPB and LPE....138
Annex J (informative) Public L_Ports and Private NL_Ports on a Loop....139
Annex K (informative) Assigned Loop Identifier....140
Annex L (informative) Selective replicate for parallel query acceleration....141
L.0Overview....141
L.1Parallel query technology....141
L.2Shared disk cluster....142
L.3Parallel query example....142
Annex M (informative) Controlled FC-AL configurations....145
M.0Overview....145
M.1Address Control....145
M.1.0Overview....145
M.1.1Preferred Hard Addressing....145
M.1.2Required Hard Addressing....145
M.2Configuration Change Control....146
M.2.0Overview....146
M.2.1Port Bypass Circuit Control....146
M.2.2Loop Initialization Control....146
Annex N (Informative) Insertion modes of Hubs....147
Annex O (informative) L_Port power-on considerations....148
Annex P (Informative) L_Port initialization flow diagram....149
Annex Q (informative) Examples of Switch Port Initialization....150
Q.0Overview....150
Q.1Example 1: two E/F/FL_Port-capable Switch....150

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

Q.2Example 2: two E/F/FL_Port-capable Switch Ports and one Nx_Port .....151
Q.3Example 3: one E/F/FL_Port-capable Port and one E/F_Port-capable Port .....152

Table 1 — 8B/10B characters with neutral disparity..... 26
Table 2 – Primitive Signals..... 28
Table 3 — Primitive Sequences ..... 28
Table 4 — MONITORING (State 0) transitions ..... 63
Table 5 — ARBITRATING (State 1) transitions ..... 67
Table 6 — ARBITRATION WON (State 2) transitions ..... 70
Table 7 — OPEN (State 3) transitions..... 72
Table 8 — OPENED (State 4) transitions ..... 74
Table 9 — XMITTED CLOSE (State 5) transitions..... 77
Table 10 — RECEIVED CLOSE (State 6) transitions..... 80
Table 11 — TRANSFER (State 7) transitions ..... 83
Table 12 — INITIALIZATION process (State 8) transitions ..... 85
Table 13 — Reserved ..... 85
Table 14 — OLD-PORT (State A) transitions..... 85
Table 15 — AL_PA mapped to bit maps..... 91
Table C.1 — Dynamic Half-Duplex.....127
Table K.1 — Assigned Loop Identifier .....140

Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

Figure 1 — Fibre channel roadmap ..... 10
Figure 2 — Examples of the Loop topology ..... 21
Figure 3 — FC-PH with Arbitrated Loop addition ..... 23
Figure 4 — State Diagram ..... 44
Figure 5 — Loop Initialization Sequences ..... 89
Figure 6 — Loop Initialization Sequence AL_PA bit map ..... 93
Figure 7 — Loop Initialization state diagram example ..... 97
Figure 8 — POWER-ON state diagram ..... 100
Figure 9 — OLD-PORT state diagram ..... 101
Figure 10 — Loop Fail Initialization state diagram ..... 103
Figure 11 — Normal Initialization state diagram ..... 105
Figure 12 — OPEN-INIT state diagram ..... 107
Figure 13 — Slave Initialization state diagram ..... 110
Figure 14 — Slave AL_PA position map state diagram ..... 113
Figure 15 — Master Initialization state diagram ..... 115
Figure 16 — Master AL_PA position map state diagram ..... 117
Figure A.1 — Elasticity buffer ..... 119
Figure A.2 — Clock skew management states ..... 120
Figure G.1 — Example of a synchronous L_Port design ..... 132
Figure G.2 — Example of an asynchronous L_Port design ..... 132
Figure I.1 — Example Port Bypass Circuit ..... 136
Figure J.1 — Public L_Ports and Private NL_Ports on a Loop ..... 139
Figure L.1 — FC-AL parallel query server ..... 142
Figure P.1 — L_Port initialization flow diagram ..... 149
Figure Q.1 — Switch Initialization example 1 ..... 150
Figure Q.2 — Switch Initialization example 2 ..... 151
Figure Q.3 — Switch Initialization example 3 ..... 152

Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

# INFORMATION TECHNOLOGY – FIBRE CHANNEL –

## Part 122: Arbitrated loop-2 (FC-AL-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. Their preparation is entrusted to technical committees; any ISO and IEC member body interested in the subject dealt with may participate in this preparatory work. International governmental and non-governmental organizations liaising with ISO and IEC also participate in this preparation.

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) The formal decisions or agreements of IEC and ISO on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC and ISO member bodies.

4) IEC, ISO and ISO/IEC publications have the form of recommendations for international use and are accepted by IEC and ISO member bodies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC, ISO and ISO/IEC publications is accurate, IEC or ISO cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

5) In order to promote international uniformity, IEC and ISO member bodies undertake to apply IEC, ISO and ISO/IEC publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any ISO/IEC publication and the corresponding national or regional publication should be clearly indicated in the latter.

6) ISO and IEC provide no marking procedure to indicate their approval and cannot be rendered responsible for any equipment declared to be in conformity with an ISO/IEC publication.

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

8) 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 member bodies 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, ISO or ISO/IEC publications.

9) 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.

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

International Standard ISO/IEC 14165-122 was prepared by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

This consolidated version of ISO/IEC 14165-122 consists of the first edition (2005) and its Amendment 1 (2008).

The technical content is therefore identical to the base edition and its amendment and has been prepared for user convenience.

It bears the edition number 1.1.

A vertical line in the margin shows where the base publication has been modified by Amendment 1.

Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

The list of all currently available parts of ISO/IEC 14165 series, under the general title *Information technology – Fibre channel*, can be found on the IEC web site.

This International Standard has been approved by vote of the member bodies, and the voting results may be obtained from the address given on the second title page.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

# INTRODUCTION

International Standard ISO/IEC 14165-122 specifies an enhancement to the signaling protocol of the Fibre Channel Physical and Signaling Interface (FC-PH), ISO/IEC 14165-251, to support communication among two or more Ports without using the Fabric topology. The following diagram shows the relationship of this document to other parts of Fibre Channel. FC-PH-n refers to *n* versions of FC-PH. The roadmap is intended to show the general relationship of documents to one another, not a hierarchy, protocol stack or system architecture. It does not show the complete set of Fibre Channel documents.

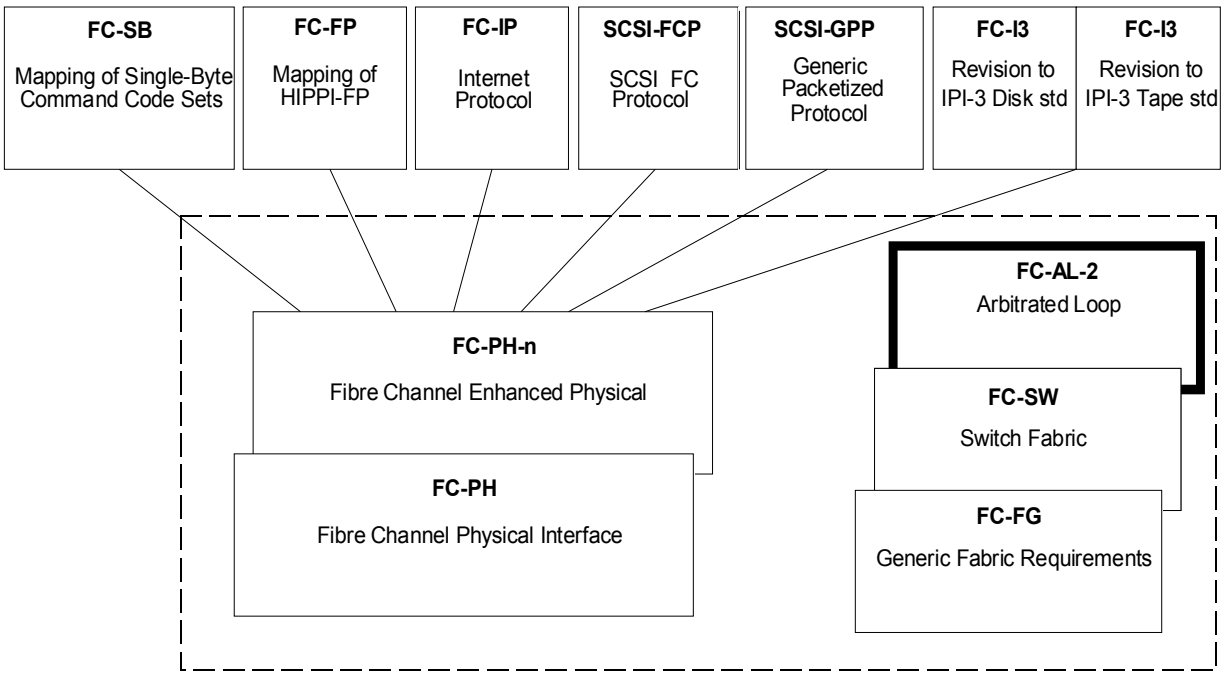


**Figure 1 — Fibre channel roadmap**

FC-AL features enhanced Ports, called L_Ports, which arbitrate to access an Arbitrated Loop. Once an L_Port wins arbitration, a second L_Port may be opened to complete a single point-to-point circuit (i.e., communication path between two L_Ports). When the two connected L_Ports release control of the Arbitrated Loop, another point-to-point circuit may be established. An L_Port may have the ability to discover its environment and works properly, without outside intervention, with an F_Port, an N_Port or with other L_Ports.

There is no change to the framing protocol of FC-PH-n, however, modification to the Port hardware is required to transmit, receive and interpret the new Arbitrated Loop Primitive Signals and Sequences.

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

# INTRODUCTION to Amendment 1

Since the publication of ISO/IEC 14165-122:2005 (FC-AL) important technical corrections have been developed.

FC-AL-2 defines a method for the operation of a FC arbitrated loop. This amendment revises the base document, FC-AL-2, with respect to some inconsistencies found since approval of the base document.

This amendment contains the following changes:

- the transmission word delay through an L_Port is changed from 6 words to 12 words to accommodate higher fibre channel speeds; for example, 8 Gbit/s and 16 Gbit/s.
- the OPEN state is corrected to assure fairness for an L_Port that is using the TRANSFER state while another L_Port is using ARBf.

Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

# INFORMATION TECHNOLOGY – FIBRE CHANNEL –

## Part 122: Arbitrated loop-2 (FC-AL-2)

## 1 Scope

This part of ISO/IEC 14165 specifies signaling interface enhancements for FC-PH, to allow L_Ports to operate with an Arbitrated Loop topology. This standard defines L_Ports that retain the functionality of Ports as specified in FC-PH. The Arbitrated Loop topology attaches multiple communicating points in a loop without requiring switches.

The Arbitrated Loop topology is a distributed topology where each L_Port includes the minimum necessary function to establish a Loop circuit. A single FL_Port connected to an Arbitrated Loop allows multiple NL_Ports to attach to a Fabric.

When an L_Port is operating on a Loop with at least one other L_Port, the L_Port uses the protocol extensions to FC-PH that are specified in this standard.

When an L_Port is connected with an N_Port or an F_Port, the L_Port communicates using the protocol defined in FC-PH.

Each L_Port may use a self-discovering procedure to find the correct operating mode without the need for external controls.

## 2 Normative 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.

ISO/IEC 14165-131, *Information technology — Fibre Channel — Part 131: Switch Fabric Requirements (FC-SW)*

ISO/IEC 14165-141, *Information technology — Fibre Channel — Part 141: Fabric Generic Requirements (FC-FG)*

ISO/IEC 14165-251, *Information technology — Fibre Channel — Part 251: Framing and Signaling (FC-FS)*[1]

INCITS 230:1994 [R2004], *Fibre Channel — Physical and Signaling Interface (FC-PH) [T11]*

Amendment 1 (1996)

Amendment 2 (1999)

[1] Under consideration.

Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

# 3 Definitions and conventions

## 3.1 Definitions

For the purposes of this International Standard, the definitions in FC-FH and the following definitions apply. Definitions in this clause take precedence over any definitions in FC-FH.

**3.1.1**
**Arbitrated Loop**
Fibre Channel topology where Ports use arbitration to gain access to the Loop

**3.1.2**
**Arbitrated Loop Physical Address (AL_PA)**
unique one-byte valid value as established in 5.1

**3.1.3**
**Arbitrated Loop Destination Address (AL_PD)**
Arbitrated Loop Physical Address of the L_Port on the Loop that should receive the Primitive Signal or Primitive Sequence
For example, the AL_PD is the y value of the OPNyx or OPNyy Primitive Signal.

**3.1.4**
**Arbitrated Loop Source Address (AL_PS)**
Arbitrated Loop Physical Address of the L_Port on the Loop that transmitted the Primitive Signal or Primitive Sequence

For example, the AL_PS is the x value of the OPNyx Primitive Signal.

**3.1.5**
**close**
procedure used by an L_Port to terminate a Loop circuit

**3.1.6**
**current Fill Word**
Fill Word currently selected by the LPSM to be transmitted when needed. The initial value is the Idle Primitive Signal (see 8.4)

**3.1.7**
**Dynamic Half-Duplex**
procedure initiated by the L_Port in the OPEN state to change a full-duplex transfer to a half-duplex transfer

The resulting half-duplex transfer is from the L_Port in the OPENED state to the L_Port in the OPEN state (see 7.4 and annex C).

**3.1.8**
**fairness window**
period during which a fair L_Port can arbitrate and win access to the Loop only once (see 4.3)

**3.1.9**
**Fill Word**
Transmission Word which is an Idle or an ARByx Primitive Signal
These words are transmitted between frames, Primitive Signals and Primitive Sequences to keep a fibre active. For information about Primitive Signals and Primitive Sewuences, see FC-FH.

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

**3.1.10**
**FL_Port**
an F_Port (i.e., Fabric Port) which contains the Loop Port State Machine as defined in this standard

**3.1.11**
**F/NL_Port**
an NL_Port that detects OPN(00,x) and provides Fibre Channel services in the absence of an FL_Port

**3.1.12**
**full-duplex**
communication model 2 referred to as *duplex* in FC-PH. Both L_Ports are allowed to transmit and receive Data frames

**3.1.13**
**half-duplex**
communication model 1 in FC-PH. Only one L_Port is allowed to transmit Data frames

**3.1.14**
**Hub**
device for interconnecting L_Ports

**3.1.15**
**Loop**
the Arbitrated Loop as described in this standard

**3.1.16**
**Loop circuit**
bidirectional path that allows communication between two L_Ports on the same Loop

**3.1.17**
**Loop Failure**
loss of word synchronization for greater than R_T_TOV; or loss of signal (see FC-PH)

**3.1.18**
**L_Port**
either an FL_Port or a Public or Private NL_Port as defined in FC-PH

**3.1.19**
**NL_Port**
N_Port (i.e., Node Port) which contains the Loop Port State Machine defined by this document

Without the qualifier "Public" or "Private," an NL_Port is assumed to be a Public NL_Port.

**3.1.20**
**non-L_Port**
Port that does not support the Loop functions defined in this standard (see *Port* in FC-PH)

**3.1.21**
**Non-Participating mode**
operational mode of an L_Port which does not have an AL_PA, but is enabled into the Loop (see 8.1.4)

Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

**3.1.22**
**Non-Participating Bypassed mode**
operational mode of an L_Port which does not have an AL_PA and is bypassed from the Loop (see 8.1.4)

**3.1.23**
**open**
procedure used by an L_Port to establish a Loop circuit

**3.1.24**
**Participating mode**
operational mode of an L_Port which has an AL_PA and is enabled into the Loop (see 8.1.4)

**3.1.25**
**Participating Bypassed mode**
operational mode of an L_Port which has an AL_PA, but is bypassed from the Loop (see 8.1.4)

**3.1.26**
**Port_Name**
A unique 64-bit identifier as defined in the LOGI or ACC frame (see FC-PH)

**3.1.27**
**Primitive Sequence**
three identical consecutive Ordered Sets before the function conveyed by the Primitive Sequence is performed (see FC-PH)

**3.1.28**
**Private Loop**
a Loop that does not include a Participating FL_Port (see Figure 2 and annex J)

**3.1.29**
**Private NL_Port**
an NL_Port that does not attempt a Fabric Login and does not transmit OPN(00,x) (see Figure 2 and 5.2)

**3.1.30**
**Public Loop**
a Loop that includes a Participating FL_Port and may contain both Public and Private NL_Ports (see Figure 2 and annex J)

**3.1.31**
**Public NL_Port**
an NL_Port that attempts a Fabric Login (see Figure 2 and 5.2)

**3.1.32**
**replicate frame**
a Class 3 frame which may be received and processed by one or more NL_Ports while being forwarded (see 7.2)

**3.1.33**
**transfer**
a procedure used by an L_Port to close an existing Loop circuit in order to establish a new Loop circuit without relinquishing control of the Loop

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

**3.1.34**
**trusted AL_PA**
an AL_PA which is assumed to be valid and unique through a vendor-specified means (e.g., a Hard Address)

### 3.2 Editorial conventions

In FC-AL, many conditions, mechanisms, sequences, events or similar terms are printed with the first letter of each word in upper case and the rest lower case (e.g., Loop). States are defined in all upper case letters. Any lower case words not defined in 3.1 have the normal technical English meaning.

In case of conflicts between text, tables and figures, the following precedence shall be used: text, tables, figures. State diagrams have precedence as stated in the appropriate clauses.

The word *shall,* when used in this standard, states a mandatory rule or requirement. The word *may,* when used in this standard, states an optional rule. The word *should*, when used in this standard, denotes flexibility of choice with a strongly preferred alternative (equivalent to the phrase 'is recommended').

The words, *recognize, recognizes or recognized*, when used in this standard, indicate that an L_Port has detected a Primitive Signal or Primitive Sequence.

Each individual entry that appears within parentheses of FC-AL Ordered Sets (e.g., ARByx and OPNy) represents the hexadecimal value of an AL_PA or special flags (e.g., hex 'F7', hex 'F8' and hex 'FF').

All history variables, when used in this standard, are assumed to be set to zero (0) at power-on time. Any optional history variable that is not implemented tests as zero (0) in all tests of that variable.

The ISO convention of numbering is used; i.e., the thousands and higher multiples are separated by a space and a comma is used instead of the decimal point (e.g., 1 062,5 Mb/s).

Whenever FC-PH is referenced, all FC-PH documents referenced in clause 2 are implied.

### 3.3 Abbreviations, acronyms and other special words

| | |
|---|---|
| **ACCESS** | **Access** history variable – a two-valued variable (i.e., 0/1) to indicate access fairness history (see 8.1.1) |
| **AL_PA** | **A**rbitrated **L**oop **P**hysical **A**ddress (see 5.1) |
| **AL_PD** | **A**rbitrated **L**oop **D**estination Physical Address (e.g., the y value in OPNyx and OPNyy) |
| **AL_PS** | **A**rbitrated **L**oop **S**ource Physical Address (e.g., the x value in OPNyx) |
| **AL_TIME** | **A**rbitrated **L**oop **time**out value (see 8.2.2) |
| **ARB_PEND** | **Arb**itration **PEND**ing history variable – a two-valued variable (i.e., 0/1) to help an L_Port remember that it was arbitrating |
| **ARB_WON** | **Arb**itration **Won** history variable – a two-valued variable (i.e., 0/1) to remember whether the L_Port won arbitration (see 8.1.1) |

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

**ARB(AL_PA)** **Arb**itrate Primitive Signal – an ARByx in which y = x = AL_PA of the L_Port (see 7.1.1)

**ARB(val)** **Arb**itrate Primitive Signal – an ARByx in which y = x = val ('val' represents a one byte hexadecimal value)

**ARByx** **Arb**itrate Primitive Signal – any Ordered Set that begins with K28.5, D20.4 ('y' and 'x' may be used in adjacent text to denote the value of characters 3 and 4 within the Ordered Set) (see 7.1)

**ARBf_SENT** **Arb**itrate hex 'FF' **Sent** history variable – a two-valued variable (i.e., *0/1*) that indicates that the L_Port has requested REQ(arbitrate (FF)) and the LPSM has modified the current Fill Word to ARB(FF) (see 8.1.6 and 8.3.3)

**Available_BB_Credit** **Available** **B**uffer-to-**B**uffer **Credit** (see 8.2.4)

**BB_Credit** **B**uffer-to-**B**uffer **Credit** as established during Login (see 8.2.4 and FC-PH)

**BYPASS** **BYPASS** history variable – a two-valued variable (i.e., 0/1) which indicates whether an L_Port is bypassed (see 8.1.4)

**CLS** **CL**o**S**e Primitive Signal (see 7.4)

**CFW** **C**urrent **F**ill **W**ord (i.e., Idle or ARByx) (see 3.1.10 and 7.1)

**DHD** **D**ynamic **H**alf-**D**uplex Primitive Signal (see 7.5)

**DHD_RCV** **D**ynamic **H**alf-**D**uplex **R**e**C**ei**V**ed history variable – a two valued variable (i.e., 0/1) to indicate that the L_Port in the OPENED state has detected and supports DHD (see 8.1.5)

**DUPLEX** **DUPLEX** history variable – a two-valued variable (i.e., 0/1) to indicate whether the L_Port is allowed to originate Data frames (see 8.1.2)

**EE_Credit** **E**nd-to-**E**nd **Credit** (see FC-PH)

**LIFA** **L**oop **I**nitialization **F**abric **A**ssigned – Loop Initialization Sequence (see 10.5)

**LIHA** **L**oop **I**nitialization **H**ard **A**ssigned – Loop Initialization Sequence (see 10.5)

**LILP** **L**oop **I**nitialization **L**oop **P**osition – Loop Initialization Sequence (see 10.5)

**LIM** **L**oop **I**nitialization **M**aster – the L_Port which is responsible for initializing the Loop (see clause 10)

**LIP** **L**oop **I**nitialization **P**rimitive Sequence – any of the LIP Primitive Sequences (see 7.8)

**LIPfx** **L**oop **I**nitialization **P**rimitive Sequence – perform a vendor unique reset of all (except AL_PA = x) L_Ports (f = hex 'FF') (see 7.8.5)

**LIPA** **L**oop **I**nitialization **P**reviously **A**cquired – Loop Initialization Sequence (see 10.5)

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

| | |
|---|---|
| **LIPyx** | **L**oop **I**nitialization **P**rimitive Sequence – perform a vendor unique reset of an L_Port at AL_PA = y (see 7.8.5) |
| **LIRP** | **L**oop **I**nitialization **R**eport **P**osition – Loop Initialization Sequence (see 10.5) |
| **LISA** | **L**oop **I**nitialization **S**oft **A**ssigned – Loop Initialization Sequence (see 10.5) |
| **LISM** | **L**oop **I**nitialization **S**elect **M**aster – Loop Initialization Sequence (see 10.5) |
| **LI_FL** | **L**oop **I**nitialization **FL**ag – Loop Initialization flag (see 10.5) |
| **LI_ID** | **L**oop **I**nitialization **ID**entifier – Loop Initialization identifier (see 10.5) |
| **LP_TOV** | **LooP TimeOut Value** (see 8.2.3) |
| **LPB** | **L**oop **P**ort **B**ypass Primitive Sequence – either LPByx or LPBfx (f = hex 'FF') (see 7.7.1 and 7.7.2) |
| **LPBfx** | **L**oop **P**ort **B**ypass Primitive Sequence – used to bypass all (except AL_PA = x) L_Ports (f = hex 'FF') (see 7.7.2) |
| **LPByx** | **L**oop **P**ort **B**ypass Primitive Sequence – used to bypass an L_Port at y = AL_PA (see 7.7.1) |
| **LPE** | **L**oop **P**ort **E**nable Primitive Sequence – either LPEyx or LPEfx (see 7.7.3 and 7.7.4) |
| **LPEfx** | **L**oop **P**ort **E**nable Primitive Sequence – used to enable all bypassed L_Ports (f = hex 'FF') (see 7.7.4) |
| **LPEyx** | **L**oop **P**ort **E**nable Primitive Sequence – used to enable a bypassed L_Port at y = AL_PA (see 7.7.3) |
| **LPSM** | **L**oop **P**ort **S**tate **M**achine (see 8.4) |
| **MRKtx** | **Mark** Primitive Signal (see 7.6) |
| **MK_TP** | **Mark Type** – used to identify the type of Mark Primitive Signal (see 7.6) |
| **OPNr** | **Open** Replicate Primitive Signal – either OPNyr or OPNfr (see 7.2) |
| **OPNfr** | **Open** Primitive Signal – broadcast replicate(see 7.2.2) |
| **OPNy** | **Open** Primitive Signal – either OPNyx or OPNyy (see 7.1) |
| **OPNyr** | **Open** Primitive Signal – selective replicate(see 7.2.1) |
| **OPNyx** | **Open** Primitive Signal – full-duplex (see 7.1.1) |
| **OPNyy** | **Open** Primitive Signal – half-duplex (see 7.1.2) |
| **PARTICIPATE** | **PARTICIPATE** history variable – a two-valued variable (i.e., *0/1*) that indicates whether an L_Port has an AL_PA and is participating on the Loop (see 8.1.4) |

AVA Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

**REPEAT** symbol whose value is derived from BYPASS and PARTICIPATE, which indicates that an L_Port merely repeats received Transmission Words (see 8.1.4)

**REPLICATE** **Replicate** history variable – a two valued variable (i.e., *0*/*1*) to indicate if an L_Port has transmitted OPNr while in the OPEN state or an NL_Port has received OPNr while in the MONITORING or ARBITRATING states (see 8.1.3)

**SOFiL** **S**tart_**o**f_**F**rame Primitive Signal (K28.5 D21.5 D22.2 D22.2) used during Loop Initialization (see 10.5)

**XMIT_2_IDLES** **X**mit **2** **I**dles history variable – a two-valued variable (i.e., 0/1) that indicates whether the L_Port needs to transmit two (2) Idles (see 8.1.1)

### 3.4 Symbols

Logic symbols are represented in state tables and diagrams as follows:

— the logical 'or' is represented as '|';

— the logical 'and' is represented as '&';

— the logical negation is represented as '~' (tilda);

— the 'less-than' is represented as '<';

— the 'greater-than' is represented as '>';

— comparisons are represented as '=' (equal) and '<>' (not equal);

— setting a variable is done using the colon equal operator, ':='; and,

— the concatenation symbol is represented as '||'.

## 4 Structure and concepts

### 4.0 Purpose

This clause provides an overview of the structure, concepts and mechanisms that allow two or more L_Ports to communicate without using a Fabric topology. Readers unfamiliar with FC-AL should read or scan clauses 1 and 4 before attempting to master the detailed material in clauses 5 through 10.

### 4.1 Overview

FC-AL is a serial data channel, structured for low-cost connectivity, that provides a logical bidirectional, point-to-point service between two L_Ports. Each L_Port represents a communication point. The additional functions, that are added to allow an N_Port or F_Port to operate on a Loop, permit the L_Ports to form a simple, blocking, non-meshed, switching environment.

— Blocking refers to the number of circuits that can be concurrently active. Only one pair of L_Ports may communicate at one time although there may be up to 127 participating L_Ports attached on one Loop. All other communication must wait (i.e., is blocked).

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

— Non-meshed refers to the attribute of a Loop where there is exactly one path on each Loop between L_Ports. Non-meshed implies that any single fibre problem may stop all activity on the Loop Meshed, in this context, means that there may be alternate paths available between L_Ports.

— Switching refers to the Loop circuitry added to each L_Port compared to a non-L_Port. The circuitry acts as a two-port switch where information received on the inbound fibre of the L_Port is directed to either the local FC-2 function or placed on the outbound fibre for another L_Port to process.

The Loop supports a maximum of one point-to-point circuit at a time. When two L_Ports are communicating, the Loop topology supports simultaneous, symmetrical, bidirectional flow between the two L_Ports. All other L_Ports are monitoring or arbitrating for access to the Loop.

The Loop supports all Classes of Service as specified in FC-PH. Individual L_Ports may choose to implement, at the FC-2 level, only a subset of the Classes of Service which are available. Such an implementation does not affect the operation of the Loop protocol.

The Loop guarantees in-order delivery of frames in all Classes of Service when the source and destination are on the same Loop. Frames transmitted from an NL_Port to an FL_Port are received at the FL_Port in the transmitted order; frames transmitted from an FL_Port are received at the NL_Port in the transmitted order. Out-of-order frames may be received at a destination NL_Port, but that out-of-order characteristic is not caused by the FL_Port or the Loop.

Unlike the Fabric topology where a circuit is established only for a dedicated connection or virtual circuit, a Loop circuit shall be established between two L_Ports on the Loop before the FC-PH framing protocol may be used. The two L_Ports may use the framing protocol and any Class of Service appropriate for their implementations and for the FC-4 protocol being used. Other L_Ports on the same Loop may form their own Loop circuit after the current Loop circuit is closed.

### 4.2 General description

In a Fabric topology, one or more Fabric Element(s) are required to connect more than two Ports together (see FC-FG, for the minimum requirements of a Fabric).

The Loop topology reduces the number of transceivers required to interconnect L_Ports to one transceiver per L_Port. Up to 127 L_Ports may be in the Participating mode on one Loop.

The different topologies which are defined in FC-PH have certain pertinent distinguishing characteristics.

— The **point-to-point topology** is non-blocking. Each N_Port may transmit frames to the other at any time within the limits of the implemented protocols of the N_Ports.

  The number of transceivers needed to completely interconnect *n* N_Ports using multiple links may be calculated using the formula: $t = n\ (n - 1)$ where *t* and *n* represent the number of transceivers and N_Ports, respectively. For example, to connect six (6) N_Ports requires 30 transceivers.

  Also, if a link in a point-to-point topology fails, communication between that pair of Ports stops. Communication between other point-to-point connected Ports continues.

— The **Fabric topology** may be configured to be non-blocking between any two N_Ports. It is commonly acknowledged that most data processing-type Nodes cannot sustain high-speed data transfer for long periods of time to all peripheral devices (although there may be some exceptions). A Fabric offers a way to take advantage of these natural pauses in communication, allowing fewer interconnects. The available bandwidth is shared between

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

the N_Ports, but this sharing adds contention and therefore a management function is required.

One advantage for the Fabric topology is that when there is at least one free F_Port in the Fabric, a new N_Port can be added to the free F_Port without disrupting the remaining N_Ports. The new N_Port has the potential to communicate with all other N_Ports in the Fabric. However, adding an N_Port does not guarantee that the new N_Port can communicate with any of the currently attached N_Ports (see FC-FG).

Because a Fabric topology may permit multiple paths between any two F_Ports in the Fabric (i.e., the meshing capability of the Fabric topology), a Fabric topology may be more robust. For a Node with only one N_Port, there is always a single point of failure at the link to the Fabric Element.

— The **Loop topology** functions are the result of reducing the Fabric topology to its simplest form. There is exactly one link bandwidth to share among all L_Ports. This makes the Loop the ultimate blocking topology, yet it retains considerable connectivity. There can be only one active Loop circuit at a time, independent of the number of L_Ports on a Loop. New L_Ports can be added at any time, although only a maximum of 127 may be participating. Should any link in a Loop fail, communication between all L_Ports stops on that Loop.

  Fabric management is reduced to a minimum with the remaining functions distributed in each L_Port on the Loop. This eliminates the central management function of a Fabric and at least one-half of the transceivers compared to a Fabric topology. Once communication is established between two L_Ports, the normal FC-PH protocol is used for all operations.

Figure 2 shows two independent Loop configurations each with multiple L_Ports connected. Each line in the figure between L_Ports represents a single fibre. The configuration in Figure 2(a) shows two Loops: one includes two NL_Ports (i.e., point-to-point) and the other includes six NL_Ports (i.e., Private Loops). The configuration in Figure 2(b) shows a Loop which includes one FL_Port (i.e., a Public Loop) and five NL_Ports (either Public or Private NL_Ports). (See annex J.)

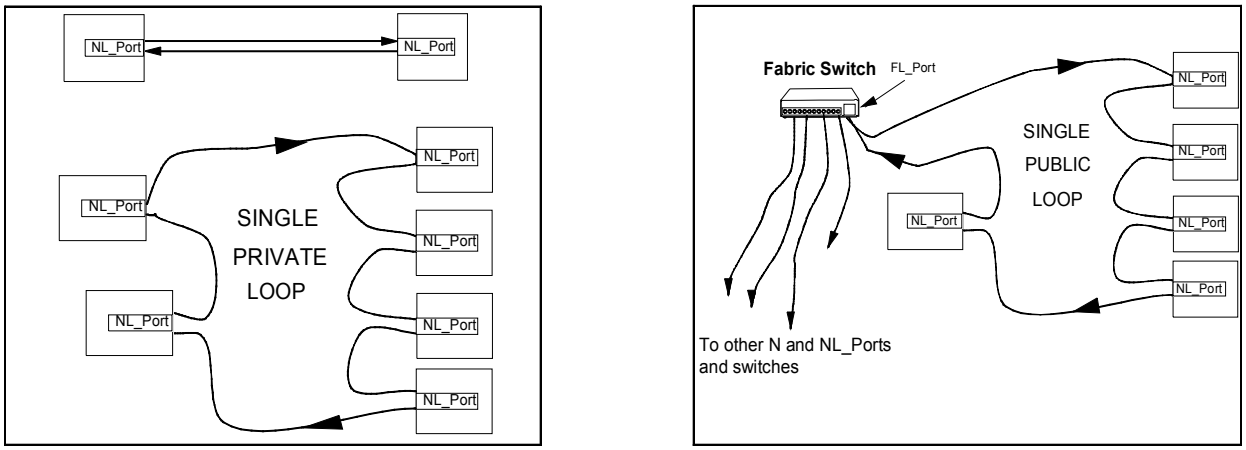


**Figure 2 — Examples of the Loop topology**

The Loop topology and the Fabric topology together provide a compromise between connectivity and performance. A number of Loops may be connected through a Fabric. For example, four sixteen-port Loops (one FL_Port and fifteen NL_Ports) may be connected through a four-port Fabric to achieve a connectivity of sixty L_Ports with better performance than if all sixty NL_Ports were on one Loop.

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

## 4.3 Access fairness algorithm

### 4.3.0 Access fairness overview

The protocol for the Loop permits each L_Port to continuously arbitrate to access the Loop. A priority is assigned to each Participating L_Port based on the Arbitrated Loop Physical Address (AL_PA). As with other prioritized protocols, this could lead to situations where the lower priority L_Ports cannot gain access to the Loop. The access fairness algorithm sets up an access window in which all L_Ports are given an opportunity to arbitrate and win access to the Loop. When all L_Ports have had an opportunity to access the Loop once, a new access window is started. An L_Port may arbitrate again and eventually win access to the Loop in the new access window. Not every L_Port is required to access the Loop in any one access window.

When an L_Port which uses the access fairness algorithm has arbitrated for and won access to the Loop, the L_Port shall not arbitrate again until at least two (2) Idles have been transmitted by the L_Port. The access window is defined as the time period between when no L_Ports are arbitrating until all L_Ports requesting to arbitrate have won arbitration. An access window may vary in size depending on the number of arbitrating L_Ports. A special arbitration Primitive Signal (i.e., ARB(F0)) is used as the Fill Word during this interval to prevent an early reset of the access window. The details of the access fairness algorithm are contained in the Loop state machine (see 8.4).

The access fairness algorithm does not limit the time that an L_Port controls the Loop once it wins arbitration, just as FC-PH does not limit the time for a Class 1 connection. However, if access is denied longer than LP_TOV, the access window is reset and an L_Port may begin arbitrating.

All L_Ports shall implement the access fairness algorithm, FL_Ports or NL_Ports are not required to use the access fairness algorithm nor use it consistently. For example, if one L_Port requires more Loop accesses than the other L_Ports, that L_Port may choose to be unfair. Although, the standard encourages all L_Ports to use the access fairness algorithm, the decision when to be fair or unfair is beyond the scope of this standard (see annex D).

### 4.3.1 Access fairness for NL_Ports

To provide equal access to the Loop for all NL_Ports, it is recommended that each NL_Port use the access fairness algorithm. When an NL_Port is using the access fairness algorithm, it is called a *fair* NL_Port.

When a fair NL_Port has access to the Loop and detects that another L_Port is arbitrating, the fair NL_Port should close the Loop at the earliest possible time and arbitrate again in the next access window.

### 4.3.2 Access unfairness for NL_Ports

The configuration of some Loops may require that certain NL_Ports have more access to the Loop than just once per access window. Examples of these NL_Ports include, but are not limited to, a subsystem controller or a file server.

An NL_Port may be initialized (or may temporarily choose) not to use the access fairness algorithm. When an NL_Port is not using the access fairness algorithm, it is called an *unfair* NL_Port. The decision whether to use the access fairness algorithm is beyond the scope of this standard. (See annex D.)

When an unfair NL_Port has arbitrated for and won access to the Loop and does not detect that another L_Port is arbitrating, that NL_Port may keep the existing Loop circuit open indefinitely or the L_Port may use the transfer procedure to open another L_Port on the Loop.

Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

When an unfair NL_Port controls the Loop and detects that another L_Port is arbitrating, the unfair NL_Port may close the Loop, keep the existing Loop circuit open, or it may use the transfer procedure to open another L_Port on the Loop.

### 4.3.3 Access unfairness for FL_Ports

A Participating FL_Port is always the highest priority L_Port on the Loop based on its AL_PA. An FL_Port is encouraged to use the access fairness algorithm, but it may choose to be unfair since the majority of its traffic is with the rest of the Fabric. If the FL_Port were required to use the fairness algorithm at all times, it would be more likely to fill buffers in the Fabric, causing non-Loop communications to be affected.

When an FL_Port controls the Loop and detects that an NL_Port is arbitrating, the FL_Port may close the Loop, keep the existing Loop circuit open, or it may use the transfer procedure to open another NL_Port on the Loop.

## 4.4 Relationship to FC-PH

If a Port uses FC-AL, it extends the FC-2 and FC-1 functions of FC-PH. Figure 3 shows logically where the Loop (FC-AL) function is located. This functional level does not have a formally defined interface to the other levels.

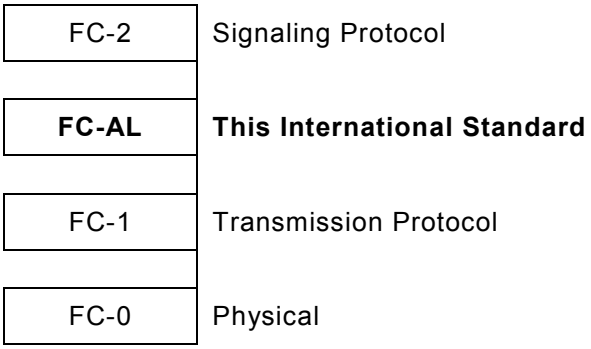


**Figure 3 — FC-PH with Arbitrated Loop addition**

When two L_Ports are communicating, the L_Ports may use all of the functions specified in FC-PH. The following list is a clause-by-clause analysis of the differences between N_Ports or F_Ports and NL_Ports or FL_Ports, respectively. The Loop:

— supports communication models 1 and 2 identified in FC-PH, but it does not support model 3. Any two L_Ports may operate in half-duplex mode during one Loop circuit. The direction of the half-duplex mode may be changed by establishing a new Loop circuit in the opposite direction;

— adds new error detection or recovery protocols in 8.2 in addition to those identified in FC-PH, and related clauses;

— places no limit on the use of any one type of transmitter (although they shall all be of the same data rate) for the cable plant of a Loop. Some requirements (e.g., Open Fibre Control) may prevent interoperability when mixed on a single Loop (see FC-PH);

— specifies that all NL_Ports and the optional FL_Port on a Loop shall use the same data rate (see FC-PH);

— the FC-PH buffer-to-buffer flow control is not used for L_Ports that are monitoring the Loop. (See 8.2.3);

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

— expands the number of Ordered Sets beyond those specified in FC-PH (see clause 6);

— expands the number of Primitive Signals and Sequences beyond those specified in FC-PH, (see clause 7);

— extends the Ordered Sets that may be deleted to include Idle, ARByx, and all Primitive Sequences, (See 8.2.2);

— specifies the Primitive Signals that may be inserted on a Loop between frames for clock skew management (see 8.2.2);

— modifies an F_Port behavior to allow clock skew management by L_Ports on a Loop. An FL_Port in the OPEN, OPENED, or RECEIVED CLOSE state shall originate at least six (6) Primitive Signals between Class 2 or Class 3 frames. In a Class 1 connection, the clock skew needs to be managed between the two NL_Ports (i.e., from one end of the circuit to the other end);

— defines a local physical address and native address identifier assignment algorithms when an FL_Port is not present on a Loop (see clause 10);

— requires a minimum payload size of 132 bytes for Loop Initialization (see 10.5);

— permits an L_Port to manage a separate BB_Credit for each L_Port on the Loop or the L_Port may choose to use a single value for BB_Credit. The single value shall be between zero (0) and the minimum value for all L_Ports;

— requires that the L_Port set the "Alternate BB_Credit Management" bit to 1 in the N_Port Common Service Parameters during Login (see FC-PH);

— permits a Loop circuit to be terminated when Available_BB_Credit is unbalanced;

— requires that each L_Port is capable of mapping the S_ID in each frame it receives to the AL_PA of the L_Port that transmitted this frame;

— requires that the destination of a connect request (SOFc1) sent through an FL_Port is to a Port not on the Loop; the FL_Port is not able to open another NL_Port on the same Loop (this would require three open L_Ports);

— requires Loop Initialization Sequences to be used during the initialization procedure; and,

— allows an NL_Port (in the absence of an FL_Port) to act as an F/NL_Port. The F/NL_Port shall provide the Fabric Login service associated with well-known address identifier hex 'FFFFFE'. The F/NL_Port may also provide services associated with other well-known address identifiers.

When a Loop circuit has been established between two L_Ports (i.e., an FL_Port to an NL_Port or an NL_Port to an NL_Port) (see FC-PH for flow control), FC-2 uses

— the point-to-point topology model, when both communicating NL_Ports are on the same Loop or

— the Fabric topology model, when one communicating Port is outside the Loop.

Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

# 5 Addressing

## 5.1 Arbitrated Loop Physical Address (AL_PA)

Each L_Port (if it chooses to participate on the Loop, see 8.1.4) shall be assigned a local Arbitrated Loop Physical Address (AL_PA). The AL_PA establishes the priority of an arbitrating L_Port (i.e., the lower the AL_PA, the higher the priority).

Each L_Port shall use an AL_PA value that results in neutral disparity (see FC-PH). The algorithm described below or in Table 1 provides a means for the L_Port to select an AL_PA.

The AL_PA shall be a valid data character as specified in FC-PH that does not change the current running disparity of a Transmission Word. The algorithm below is dependent on the FC-1 naming convention for an information byte in FC-PH and Table 26, identified as Dxx.y. The xx portion of the FC-1 naming convention is based on bits identified as E, D, C, B, and A in FC-PH in that order. The y portion of the FC-1 naming convention is based on bits identified as H, G, and F in FC-PH, in that order. A decimal value is assigned to each bit combination with the range of 0 to 31 for xx and 0 to 7 for y, respectively. The entire range for valid data characters using the FC-1 naming convention is D00.0 through D31.7.

Disparity for a valid data character is calculated as follows:

— arrange an information byte in the manner prescribed for the naming convention in FC-PH, to obtain the Dxx.y data byte name;

— if the xx portion of a valid data character is (in decimal) 0, 1, 2, 4, 8, 15, 16, 23, 24, 27, 29, 30, or 31, set HI to 1. If the xx portion is (in decimal) 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 17, 18, 19, 20, 21, 22, 25, 26, or 28, set HI to 0;

— if the y portion of a valid data character is (in decimal) 0, 4, or 7, set LO to 1. If the y portion is (in decimal) 1, 2, 3, 5, or 6, set LO to 0; and,

— compute the XOR function for HI and LO (i.e., XOR(HI,LO)).

If the computed value of the XOR function is 0, the value is disparity neutral and is a valid AL_PA. If the computed value of the XOR function is 1, the value is disparity biased and the FC-2 byte is not a valid AL_PA.

Table 1 identifies with an asterisk (*) each 8B/10B character that has neutral disparity ordered by the Dxx.y naming convention. The right-most column shows the FC-2 byte notation values for each row with neutral disparity in Table 1.

Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

**Table 1 — 8B/10B characters with neutral disparity**

| D xx . y | y | | | | | | | | Hex value |
|---|---|---|---|---|---|---|---|---|---|
| xx | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| 00 | * | | | | * | | | * | 00, 80, E0 |
| 01 | * | | | | * | | | * | 01, 81, E1 |
| 02 | * | | | | * | | | * | 02, 82, E2 |
| 03 | | * | * | * | | * | * | | 23, 43, 63, A3, C3 |
| 04 | * | | | | * | | | * | 04, 84, E4 |
| 05 | | * | * | * | | * | * | | 25, 45, 65, A5, C5 |
| 06 | | * | * | * | | * | * | | 26, 46, 66, A6, C6 |
| 07 | | * | * | * | | * | * | | 27, 47, 67, A7, C7 |
| 08 | * | | | | * | | | * | 08, 88, E8 |
| 09 | | * | * | * | | * | * | | 29, 49, 69, A9, C9 |
| 10 | | * | * | * | | * | * | | 2A, 4A, 6A, AA, CA |
| 11 | | * | * | * | | * | * | | 2B, 4B, 6B, AB, CB |
| 12 | | * | * | * | | * | * | | 2C, 4C, 6C, AC, CC |
| 13 | | * | * | * | | * | * | | 2D, 4D, 6D, AD, CD |
| 14 | | * | * | * | | * | * | | 2E, 4E, 6E, AE, CE |
| 15 | * | | | | * | | | * | 0F, 8F, EF |
| 16 | * | | | | * | | | * | 10, 90, F0 |
| 17 | | * | * | * | | * | * | | 31, 51, 71, B1, D1 |
| 18 | | * | * | * | | * | * | | 32, 52, 72, B2, D2 |
| 19 | | * | * | * | | * | * | | 33, 53, 73, B3, D3 |
| 20 | | * | * | * | | * | * | | 34, 54, 74, B4, D4 |
| 21 | | * | * | * | | * | * | | 35, 55, 75, B5, D5 |
| 22 | | * | * | * | | * | * | | 36, 56, 76, B6, D6 |
| 23 | * | | | | * | | | * | 17, 97, F7 |
| 24 | * | | | | * | | | * | 18, 98, F8 |
| 25 | | * | * | * | | * | * | | 39, 59, 79, B9, D9 |
| 26 | | * | * | * | | * | * | | 3A, 5A, 7A, BA, DA |
| 27 | * | | | | * | | | * | 1B, 9B, FB |
| 28 | | * | * | * | | * | * | | 3C, 5C, 7C, BC, DC |
| 29 | * | | | | * | | | * | 1D, 9D, FD |
| 30 | * | | | | * | | | * | 1E, 9E, FE |
| 31 | * | | | | * | | | * | 1F, 9F, FF |
| TOTAL 134 | 13 | 19 | 19 | 19 | 13 | 19 | 19 | 13 | |
| Legend: * — character with neutral disparity | | | | | | | | | |

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

### 5.1.1 Valid AL_PAs

The following valid AL_PAs are assigned to the first 127 neutral disparity values from Table 1:

hex '00': (1) AL_PA for FL_Port or alias AL_PA of F/NL_Port

Highest priority.

AL_PA hex '00' shall be assigned to the FL_Port in the Participating mode. The maximum number of FL_Ports in the Participating mode on a single Loop shall not exceed one. Additional FL_Ports may be present, but they shall be in the Non-Participating or Non-Participating Bypassed mode.

NOTE Adding a new FL_Port to a loop with an existing FL_Port, may cause the access to the ports on that loop to change to the new FL_Port and the fabric that it is a part of, including possible changes in address for the ports on the loop. If the fabric that the two FL_Ports are part of is not the same, this may impact access to the ports on the loop from the fabric.

If there is no Participating FL_Port on the Loop, a Participating NL_Port may accept this value as an alias AL_PA for its LPSM, but not as its only AL_PA.

hex '01' through hex 'EF': (126) AL_PA for NL_Ports

Descending priority is assigned as AL_PA values increase in the range from hex '01' through hex 'EF'. All valid values in this range are lower in priority than hex '00'.

Each Participating NL_Port shall be assigned one valid AL_PA in this range. The maximum number of Participating NL_Ports on a single Loop shall not exceed 126.

### 5.1.2 Special AL_PAs and flags

The following special AL_PAs and flags are assigned to the last 7 neutral disparity values from Table 1. These values replace an AL_PA to provide special functions:

hex 'F0': (1) Value used in ARB for fairness and during Loop Initialization. Value hex 'F0' is the next lower priority outside the range hex '00' through hex 'EF'.

hex 'F1' through hex 'F6': (0) Reserved

hex 'F7': (1) Value used in LIP to indicate that the L_Port is initializing and also a value for an L_Port which does not have an AL_PA.

hex 'F8': (1) Value used in LIP to indicate a Loop Failure has been detected at the receiver of the L_Port.

hex 'F9' through hex 'FE': (3) Reserved

hex 'FF': (1) Value used to address all L_Ports (except the originating L_Port) in OPNfr, LPBfx, LPEfx, and LIPfx and as a special ARB(FF) Fill Word.

## 5.2 Native address identifier

A native address identifier shall be assigned to each Participating NL_Port (up to the maximum of 126 NL_Ports) with the following characteristics:

— The low-order byte (bits 7-0) of the native address identifier is the AL_PA of the L_Port. The AL_PA shall be unique on a Loop, shall be in the range of hex '01' through hex 'EF', and shall be valid according to Table 1.

— All Private NL_Ports shall have the upper two bytes of their native address identifier (bits 23-8) equal to hex '0000'.

Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

— All Public NL_Ports shall have the upper two bytes of their native address identifier (bits 23-8) equal to the upper two bytes of the native address identifier of the FL_Port (hex 'XXXX00'). The FL_Port shall acquire this value (which shall not equal hex '000000') from its Fabric Element. The upper two bytes shall be unique for each Loop to allow multiple Loops to attach to the same Fabric.

All Public NL_Ports shall process PLOGI, LOGO, and ABTS frames with a D_ID of hex '0000'||AL_PA or hex 'XXXX'||AL_PA.

If an FL_Port is not present, these upper two bytes shall be set to zero (hex '0000').

— a native address identifier may be assigned to another NL_Port if a Participating NL_Port enters the Non-Participating mode.

## 6 FC-AL Ordered Sets

Table 2 specifies the Ordered Sets that shall be detected and may be originated by L_Ports on the Loop as additional Primitive Signals (see FC-PH). Table 3 specifies the Ordered Sets that shall be recognized and may be originated by L_Ports on the Loop as additional Primitive Sequences (see clause 7).

**Table 2 – Primitive Signals**

| Primitive Signal | | Beginning RD | Ordered Set 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| Arbitrate | ARByx | Negative | K28.5 | D20.4 | y | x |
| Arbitrate | ARB(val) | Negative | K28.5 | D20.4 | val | val |
| Close | CLS | Negative | K28.5 | D5.4 | D21.5 | D21.5 |
| Dynamic Half-Duplex | DHD | Negative | K28.5 | D10.4 | D21.5 | D21.5 |
| Mark | MRKtx | Negative | K28.5 | D31.2 | MK_TP | AL_PS |
| Open full-duplex | OPNyx | Negative | K28.5 | D17.4 | AL_PD | AL_PS |
| Open half-duplex | OPNyy | Negative | K28.5 | D17.4 | AL_PD | AL_PD |
| Open selective replicate | OPNyr | Negative | K28.5 | D17.4 | AL_PD | D31.7 |
| Open broadcast replicate | OPNfr | Negative | K28.5 | D17.4 | D31.7 | D31.7 |

Characters 3 and 4 of Ordered Sets ARByx, ARB(val), MRKtx, OPNyx, OPNyy, OPNyr, and OPNfr shall contain one of the neutral disparity values from Table 1, whenever such an Ordered Set is originated (see 7.1 to 7.7).

**Table 3 — Primitive Sequences**

| Primitive Signal | | Beginning RD | Ordered Set 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| Loop Initialization--F7,F7 | LIP(F7,F7) | Negative | K28.5 | D21.0 | D23.7 | D23.7 |
| Loop Initialization--F8,F7 | LIP(F8,F7) | Negative | K28.5 | D21.0 | D24.7 | D23.7 |
| Loop Initialization--F7,x | LIP(F7,x) | Negative | K28.5 | D21.0 | D23.7 | AL_PS |
| Loop Initialization--F8,x | LIP(F8,x) | Negative | K28.5 | D21.0 | D24.7 | AL_PS |
| Loop Initialization--reset | LIPyx | Negative | K28.5 | D21.0 | AL_PD | AL_PS |
| Loop Initialization--reset all | LIPfx | Negative | K28.5 | D21.0 | D31.7 | AL_P |
| Loop Initialization--reserved | LIPba | Negative | K28.5 | D21.0 | b | a |
| Loop Port Bypass | LPByx | Negative | K28.5 | D9.0 | AL_PD | AL_PS |
| Loop Port Bypass all | LPBfx | Negative | K28.5 | D9.0 | D31.7 | AL_PS |
| Loop Port Enable | LPEyx | Negative | K28.5 | D5.0 | AL_PD | AL_PS |
| Loop Port Enable all | LPEfx | Negative | K28.5 | D5.0 | D31.7 | AL_PS |

AL_PD and AL_PS characters of the above Ordered Sets shall contain one of the neutral disparity values from Table 1, whenever such an Ordered Set is originated (see 7.7 to 7.8).

The b and a characters of the above Ordered Sets shall contain one of the neutral disparity values from Table 1, whenever such an Ordered Set is originated (see 7.7 to 7.8).

AVA Documents Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

# 7 FC-AL Primitive Signals and Sequences

## 7.0 Overview

The Arbitrate Primitive Signal (ARByx) may be transmitted in place of an Idle and therefore becomes a Fill Word which may be removed for clock skew management. The Mark Primitive Signal (MRKtx) may also be transmitted in place of a Fill Word, but it shall not be removed for clock skew management. All Primitive Signals (except MRKtx and Fill Words) defined in this standard shall follow the FC-PH rule for transmitting R_RDYs (i.e., two (2) Fill Words shall precede and follow these Primitive Signals with at least six (6) Primitive Signals between frames). (See FC-PH and clause 6 for a specification of the following Ordered Sets.)

Except as specifically described, the LPSM shall fully decode (test the value of all bits in all four characters of) received Ordered Sets to detect reception of a Primitive Signal or Primitive Sequence. Any received Ordered Set that does not fully match one defined in this standard or in FC-PH shall be treated as an 'Other Ordered Set.'

## 7.1 Arbitrate Primitive Signals (ARByx)

### 7.1.0 ARByx overview

A received Ordered Set shall be detected as an Arbitrate Primitive Signal (ARByx) by detecting that its first two characters (fully decoded) are equal to the value shown in Table 2, regardless of the value of characters 3 and 4 (y and x). L_Ports shall only originate an Arbitrate Primitive Signal (ARByx) where y = x. All Arbitrate Primitive Signals shall be treated as Fill Words for clock skew management. An Arbitrate Primitive Signal is further classified by examining the values of y and x. The values of y and x used, shall be the data bytes resulting from 8B/10B decoding.

If the values of y and x are equal, a received Arbitrate Primitive Signal shall be detected as an ARB(val), where 'val' is the value of y or x. If the values of y and x are different, a received Arbitrate Primitive Signal should not be recognized as an ARB(val).[2]

### 7.1.1 ARB(AL_PA)

ARB(AL_PA) is an ARB(val) which is transmitted on a Loop by a Participating L_Port to request access to the Loop. Each ARB(AL_PA) shall contain the AL_PA of the L_Port making the request (REQ(arbitrate own AL_PA)).

### 7.1.2 ARB(F0)

ARB(F0) is an ARB(val) which is transmitted on a Loop to manage the access fairness algorithm. Since this is a low-priority ARByx, any arbitrating L_Port shall replace the ARB(F0) with its ARB(AL_PA). ARB(F0) is also used while selecting a temporary Loop Initialization Master during Loop Initialization.

[2] Some L_Ports may not implement this recommendation, and may detect a received ARByx as an ARB(val) when y <> x. With such L_Ports it is vendor specific whether y or x is used as the value of 'val'. However, the Primitive Signals ARB(F0), ARB(FF) and ARB(AL_PA) (where AL_PA is the AL_PA of the receiving L_Port) should not be detected except when y = x (i.e., by checking all four characters of the Primitive Signal). L_Ports should only originate Arbitrate Primitive Signals (ARByx) where y = x, regardless of whether they test this on received Ordered Sets. These recommendations may be required by future standards.

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

### 7.1.3 ARB(FF

ARB(FF) is an ARB(val) which may be originated by any L_Port in the MONITORING state when the access fairness window has been reset and no Fill Words other than ARB(FF) or Idle are received. Since this is the lowest-priority ARByx, any arbitrating L_Port may replace the ARB(FF) with its ARB(AL_PA). ARB(FF) has fewer bit transitions than the Idle. ARB(FF) may replace Idle to reduce the Electromagnetic Interference (EMI) which may be caused by the Idle.

## 7.2 Open Primitive Signals (OPNy)

### 7.2.0 OPNy overview

A received Ordered Set shall be detected as an Open Primitive Signal (OPNy) by detecting that its first two characters (fully decoded) are equal to the value shown in Table 2 and that its fourth character is not equal to hex 'FF'. If characters 3 and 4 are not equal, the received OPNy shall be detected as an Open full-duplex (OPNyx); if characters 3 and 4 are equal, the received OPNy shall be detected as an Open half-duplex (OPNyy). The values used for comparison shall be the data bytes resulting from 8B/10B decoding.

An originating L_Port may determine the AL_PD (y value) of OPNy by checking the D_ID of the frame. If the left-most two bytes of the D_ID are the same as the left-most two bytes of the native address identifier of the originating L_Port, the left-most two bytes of the D_ID are hex '0000', or the originating L_Port is a Private NL_Port, then the AL_PD shall be the right most byte of the D_ID. Otherwise, the AL_PD shall be hex '00' (the FL_Port); the D_ID is addressed to the Fabric or to a Port not on the same Loop.

### 7.2.1 Open full-duplex (OPNyx)

Open full-duplex (OPNyx) is transmitted on a Loop by a Participating L_Port to indicate that it is ready for Data and Link_Control frame transmission and reception (i.e., full-duplex) (see FC-PH). The OPNyx shall contain the AL_PD (destination = y value) of the L_Port to be opened and the AL_PS (source = x value) of the L_Port which transmitted OPNyx.

OPNyx that is received by a Participating L_Port (where y = AL_PA of the L_Port) in the MONITORING or ARBITRATING states indicates that another Participating L_Port desires to communicate in full-duplex mode with the L_Port that received OPNyx. The opened L_Port may transmit Data frames.

### 7.2.2 Open half-duplex (OPNyy)

Open half-duplex (OPNyy) is transmitted on a Loop by a Participating L_Port to indicate that it is ready for Data and Link_Control frame transmission and Link_Control frame reception (i.e., half-duplex) (see FC-PH). The OPNyy shall contain the AL_PD (destination − y value) of the L_Port to be opened.

OPNyy that is received by a Participating L_Port (where y = AL_PA of the L_Port) in the OPEN state indicates that the L_Port opened itself. OPNyy that is received by an L_Port (where y = AL_PA of the L_Port) in the MONITORING, ARBITRATING states indicates that another Participating L_Port desires to communicate in half-duplex mode with the L_Port that received OPNyy. In half-duplex mode, the opened L_Port shall not transmit Data frames.

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

## 7.3 Open Replicate Primitive Signals (OPNr)

### 7.3.0 OPNr overview

A received Ordered Set shall be detected as an Open Replicate Primitive Signal (OPNr) by detecting that its first two characters (fully decoded) are equal to the value shown in Table 2 and that its fourth character is equal to hex 'FF'. If character 3 is not equal to hex 'FF', the received OPNr shall be detected as an Open selective replicate (OPNyr); if character 3 is equal to hex 'FF', the received OPNr shall be detected as an Open broadcast replicate (OPNfr). The values used for comparison shall be the data bytes resulting from 8B/10B decoding.

Open Replicate (OPNr) is transmitted on a Loop by a Participating L_Port which desires to communicate with a group of NL_Ports on the same Loop. The requesting L_Port has won arbitration and is in the OPEN state. Transmitted frames shall be Class 3, although no buffer-to-buffer flow control (R_RDY) is used. If R_RDYs are transmitted by the L_Port in the OPEN state, they shall not be passed to FC-2. Frame reception is not guaranteed at each designated NL_Port (i.e., D_ID of the frame header may not be recognized by FC-2 or receive buffers may not be available). To avoid overflowing buffers and to assure that all designated NL_Ports can receive each replicate frame, the requesting L_Port should limit the number and size of frames that it transmits. The L_Port in the OPEN state shall discard all received frames.

NOTE Although an FL_Port does not replicate frames through the Fabric, an FL_Port may transmit OPNr to communicate with multiple NL_Ports.

When a Participating L_Port is in the MONITORING or ARBITRATING state and detects OPNr (where the AL_PD is either hex 'FF' or the AL_PA of the NL_Port), it shall set REPLICATE to TRUE(1). While REPLICATE is TRUE(1), each received Transmission Word (except for normal Fill Word processing - updating the CFW appropriately) shall be retransmitted to the next L_Port on the Loop and shall also be provided to the FC-2 of the NL_Port for further processing; FL_Ports shall not propagate any frame through the Fabric.

NOTE Restricting the FL_Port prevents duplicate frames from being delivered to an NL_Port on the same Loop as the originator of the OPNr from a broadcast or multicast server in the Fabric.

When CLS is received, all L_Ports with REPLICATE set to TRUE(1), shall set REPLICATE to FALSE(0).

### 7.3.1 Open selective replicate (OPNyr)

Open selective replicate (OPNyr where y = AL_PD and r = hex 'FF') is transmitted on a Loop by a Participating L_Port which desires to communicate with a subset of NL_Ports on the Loop. The requesting L_Port shall transmit OPNyr (where y is a member of the subset) to each NL_Port in the subset group. OPNyr may be transmitted to group members in any order. (See annex L.)

NOTE The following sequence of events is a valid example and shows some of the versatility of using OPNyr.

Arbitrate and win
Transmit OPN(17,FF), transmit frame (17 processes)
Transmit OPN(23,FF), transmit frame (17 and 23 process)
Transmit OPN(76,FF), transmit frame (17, 23, and 76 process)
CLS

### 7.3.2 Open broadcast replicate (OPNfr)

Open broadcast replicate (OPNfr where f and r = hex 'FF') is transmitted on a Loop by a Participating L_Port which desires to communicate with all Participating NL_Ports on the Loop.

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

### 7.4 Close Primitive Signal (CLS)

Close (CLS) is transmitted on a Loop by an L_Port in the OPEN, OPENED, or RECEIVED CLOSE state. Once an L_Port has transmitted CLS, the L_Port shall not transmit frames or R_RDYs in the current Loop circuit. CLS indicates that the transmitting L_Port is prepared to or has ended the current Loop circuit (see 8.4). CLS is also transmitted to indicate that the INITIALIZATION process has completed (see 10.5.4).

### 7.5 Dynamic Half-Duplex Primitive Signal (DHD)

Dynamic Half-Duplex (DHD) is transmitted on a Loop by the L_Port in the OPEN state to indicate to the L_Port in the OPENED state that the L_Port in the OPEN state has no more Data frames to transmit. DHD shall only be requested by the L_Port in the OPEN state if both L_Ports in the current Loop circuit have indicated support of DHD via Login and if the L_Port in the OPEN state had transmitted OPNyx (full-duplex open). The DHD supported Login bit is found in FC-PH-3 (see ISO/IEC 14165-251). DHD may allow L_Ports to make more efficient use of the established Loop circuit (see annex C) by allowing an L_Port which is in the OPENED state to transmit all Data frames, even though the L_Port in the OPEN state has finished its data transfer.

NOTE DHD adds minimal delay to the closing process (i.e., the next arbitrating L_Port will win at nearly the same time whether DHD is used or not.

Transmitting DHD only affects Data frames (i.e., Link_Control frames and R_RDYs may still be transmitted) just as in the definition of OPNyy (half-duplex open) (see 7.2.2). The recipient of DHD shall transmit CLS when it has finished its transmissions.

NOTE DHD does not prohibit either L_Port from transmitting the first CLS. However, the L_Port in the OPEN state, if it had transmitted DHD, would normally wait for the L_Port in the OPENED state to transmit the first CLS.

### 7.6 Mark Primitive Signal (MRKtx)

A received Ordered Set shall be detected as a Mark Primitive Signal (MRKtx) by detecting that its first two characters (fully decoded) are equal to the value shown in Table 2.

Mark (MRKtx) is transmitted on a Loop by a master control point to inform other Nodes of a certain action (e.g., synchronization; see annex H). The L_Port shall request to transmit MRKtx at the appropriate time (REQ(mark as tx)) and the LPSM shall attempt to transmit one MRKtx for this request by transmitting MRKtx instead of the next Fill Word. Since MRKtx shall only replace a Fill Word, it is possible that the mark may not be sent in the desired time. The REQ(mark as tx) may be withdrawn before the MRKtx can be transmitted (i.e., no MRKtx is transmitted).

NOTE In order to avoid any delay when transmitting MRKtx, Fill Words are not required to precede or follow the MRKtx (i.e., Fill Words are not inserted before or after MRKtx).

The Mark Type (MK_TP) is expressed in character 3; the AL_PA of the originator of the MRKtx is in character 4 (x value). MK_TP is vendor unique and the interpretation and use is beyond the scope of this standard. The value(s) shall be assigned from the neutral disparity characters in Table 1.

When MRKtx is received by the originator (i.e., x = AL_PS) and REPEAT is FALSE(0), the MRKtx shall be replaced with the CFW. All other L_Ports which are in the MONITORING, ARBITRATING, XMITTED CLOSE, or TRANSFER state or with REPEAT is TRUE(1) shall retransmit the received MRKtx.

NOTE Since not all states retransmit MRKtx, in order to guarantee that all L_Ports receive MRKtx, the originator should be in the OPEN state and no other L_Ports in the OPENED state (i.e., all other L_Ports are either in the MONITORING or ARBITRATING state).

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

## 7.7 Loop Port Bypass/Enable Primitive Sequences

### 7.7.0 LPE/LPB overview

If an L_Port receives three consecutive identical Ordered Sets whose first two characters (fully decoded) are equal to the values shown in Table 3, then the L_Port shall recognize a Loop Port Bypass Primitive Sequence or a Loop Port Enable Primitive Sequence. If character 3 is not equal to hex 'FF', a normal Loop Port Bypass or Loop Port Enable Primitive Sequence shall be recognized; if character 3 is equal to hex 'FF', a Loop Port Bypass all or Loop Port Enable all Primitive Sequence shall be recognized.

The Loop Port Bypass and Loop Port Enable Primitive Sequences are used to control access of an L_Port to the Loop as well as to control the optional Port Bypass Circuit. The Port Bypass Circuit may be used to physically bypass an L_Port, however, the L_Port is also logically bypassed (i.e., the L_Port shall not originate Transmission Words on the Loop). (See 8.1.4 and annex I.)

When an L_Port is in either Participating Bypassed or Non-Participating Bypassed mode, the L_Port shall not originate Transmission words (except for clock skew). The L_Port shall only monitor the Loop (as in the Non-Participating mode). If a Participating Bypassed L_Port recognizes LIP, it shall relinquish its AL_PA and enter the Non-Participating mode.

Although LPB or LPE may be transmitted in a number of states, not all states retransmit LPB or LPE. To guarantee that L_Ports receive LPB or LPE, the originator shall be in the OPEN state and all other L_Ports shall be in the MONITORING or ARBITRATING state; or all L_Ports shall be in the MONITORING, NORMAL-INITIALIZE, LOOP-FAIL-INITIALIZE, OPEN-INIT-SELECT-MASTER, or SLAVE-WAIT-FOR-MASTER state. To ensure that all L_Ports are in the MONITORING, NORMAL-INITIALIZE, LOOP-FAIL-INITIALIZE, OPEN-INIT-SELECT-MASTER, or SLAVE-WAIT-FOR-MASTER state, the originator of LPB or LPE shall enter NORMAL-INITIALIZE, LOOP-FAIL-INITIALIZE, or OPEN-INIT-START and not forward any LISMs from other L_Ports or any LIPs until it has completed its management of bypassed L_Ports. After the originator of LPB or LPE has completed its management of bypassed L_Ports it shall enter the NORMAL-INITIALIZE state.

Since an L_Port cannot guarantee that other L_Ports will not begin sending LIPs, it cannot be guaranteed that an L_Port which has an AL_PA and is bypassed will not lose its AL_PA while it is bypassed due to another L_Port transmitting LIPs. Due to this, LPB or LPE operation should be used with caution except with L_Ports that have trusted AL_PAs as described in 3.1.34.

### 7.7.1 Loop Port Bypass (LPByx)

Loop Port Bypass (LPByx) is transmitted on a Loop to bypass an L_Port and to activate the optional Port Bypass Circuit. The originator of the LPByx (as identified by AL_PS in character 4 — x value) may be a diagnostic manager or an operating L_Port that has determined that a "defective" L_Port (identified by AL_PD in character 3 — y value) exists on the Loop. (See annex I.)

When LPByx is recognized, where y = AL_PA of the L_Port, the L_Port shall set BYPASS to TRUE(1). LPByx may be used to diagnose the optional Port Bypass Circuit, for error recovery, or for any reason to cause an L_Port to be bypassed.

Each L_Port in the MONITORING, ARBITRATING, NORMAL-INITIALIZE, LOOP-FAIL-INITIALIZE, OPEN-INIT-SELECT-MASTER, or SLAVE-WAIT-FOR-MASTER state shall retransmit the received LPByx. When LPByx is received by the originator (i.e., x = AL_PA of the L_Port) and REPEAT is FALSE(0), the LPByx shall be replaced with the CFW.

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

Once an L_Port is bypassed and the optional Port Bypass Circuit has been activated, the L_Port shall only monitor the Loop for LPEyx (where y = AL_PA of the L_Port), LPEfx, or LIP. LIP is only used as a signal to relinquish its AL_PA; i.e., upon receipt of LIP, if BYPASS is TRUE(1), then the LPSM shall set PARTICIPATE to FALSE(0) and shall not go to the OPEN-INIT-START state.

### 7.7.2 Loop Port Bypass all (LPBfx)

Loop Port Bypass all (LPBfx where f = hex 'FF') is transmitted on a Loop to bypass all L_Ports and activate the optional Port Bypass Circuit(s) of all L_Ports (except the L_Port at x) . The originator of the LPBfx is identified by the AL_PS in character 4 (x value). LPBfx may be used to verify that an operating Loop is possible. It may also be useful to bypass a Non-Participating L_Port (i.e., the L_Port does not have an AL_PA). (See annex I.)

When LPBfx is recognized, all L_Ports on the Loop (participating or non-participating) except the L_Port at x, shall set BYPASS to TRUE(1).

NOTE If multiple L_Ports are simultaneously transmitting LPBfx, all L_Ports will be bypassed. An L_Port which transmitted LPBfx and which was bypassed by another LPBfx (where x <> AL_PA of the L_Port), may at a later time attempt to deactivate the optional Port Bypass Circuit and participate on the Loop. The L_Port, which is attempting Loop recovery with LPBfx, may have a faulty transmitter and therefore can by this means be bypassed by another L_Port.

Each L_Port in the MONITORING, ARBITRATING, NORMAL-INITIALIZE, OPEN-INIT-SELECT-MASTER, or SLAVE-WAIT-FOR-MASTER state shall retransmit the received LPBfx. When LPBfx is received by the originator (i.e., x = AL_PA of the L_Port) and REPEAT is FALSE(0), the LPBfx shall be replaced with the CFW.

### 7.7.3 Loop Port Enable (LPEyx)

Loop Port Enable (LPEyx) is transmitted on a Loop to enable an L_Port that had been previously bypassed and to deactivate the optional Port Bypass Circuit without an intervening LIP being received. The destination L_Port is identified by the AL_PD in character 3 (y value). The originator of the LPEyx is identified by the AL_PS in character 4 (x value). (See annex I.)

When LPEyx is recognized, where y = AL_PA of the L_Port, the L_Port shall set BYPASS to FALSE(0).

Each L_Port in the MONITORING, ARBITRATING, NORMAL-INITIALIZE,OPEN-INIT-SELECT-MASTER, or SLAVE-WAIT-FOR-MASTER state shall retransmit the received LPEyx. When LPEyx is received by the originator (i.e., x = AL_PA of the L_Port) and REPEAT is FALSE(0), the LPEyx shall be replaced with the CFW.

### 7.7.4 Loop Port Enable all (LPEfx)

Loop Port Enable all (LPEfx where f = hex 'FF') is transmitted on a Loop to deactivate all Port Bypass Circuits and enable all L_Ports into the Loop. The originator of the LPEfx is identified by the AL_PS in character 4 (x value). When an L_Port has been bypassed, it may have lost its AL_PA (e.g., the L_Port is required to relinquish its AL_PA upon recognizing LIP). Therefore, LPEfx allows these L_Ports (which no longer have an AL_PA) to be enabled on the Loop. (See annex I.)

When LPEfx is recognized, the L_Port shall set BYPASS to FALSE(0).

Each L_Port in the MONITORING, ARBITRATING, NORMAL-INITIALIZE,OPEN-INIT-SELECT-MASTER, or SLAVE-WAIT-FOR-MASTER state shall retransmit the received LPEfx. When LPEfx is received by the originator (i.e., x = AL_PA of the L_Port) and REPEAT is FALSE(0), the LPEfx shall be replaced with the CFW.

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

## 7.8 Loop initialization primitive sequences (LIP)

### 7.8.0 LIP overview

If an L_Port receives three consecutive identical Ordered Sets whose first two characters (fully decoded) are equal to the values shown in Table 3, then the L_Port shall recognize a Loop Initialization Primitive Sequence. If character 3 is equal to hex 'F7', then a normal LIP (LIP(F7)) shall be recognized; if character 3 is equal to hex 'F8', then a Loop Failure LIP (LIP(F8)) shall be recognized; if character 3 is equal to the AL_PA of the L_Port, then a reset LIP (LIPyx) shall be recognized; and, if character 3 is equal to hex 'FF', then a reset all LIP (LIPfx) shall be recognized.

Loop Initialization (LIP) is a Primitive Sequence used by an L_Port to detect if it is part of a Loop or to recover from certain Loop errors (see 8.4.3, item 21 and item 23 and clause 10).

The LIP contains information on why the LIP was transmitted in the right-most two characters (characters 3 and 4). Other L_Ports may make decisions based on this information (e.g., inform an operator of a Loop Failure).

### 7.8.1 Loop Initialization — no valid AL_PA

Loop Initialization (LIP(F7,F7)) is used by the originating L_Port to acquire an AL_PA.

### 7.8.2 Loop Initialization — Loop Failure; no valid AL_PA

Loop Initialization (LIP(F8,F7)) is used by the originating L_Port to indicate that a Loop Failure has been detected at its receiver (hex 'F8'); hex 'F7' is used to indicate that the L_Port does not have a valid AL_PA.

### 7.8.3 Loop Initialization — valid AL_PA

Loop Initialization (LIP(F7,AL_PS)) is used by the originating L_Port (identified by AL_PS) to reinitialize the Loop. The L_Port may have noticed a performance degradation (e.g., it has been arbitrating longer than it deemed reasonable) and is trying to restore the Loop into a known state.

### 7.8.4 Loop Initialization — Loop Failure; valid AL_

Loop Initialization (LIP(F8,AL_PS)) is used by the originating L_Port (identified by AL_PS) to indicate that a Loop Failure has been detected at its receiver.

### 7.8.5 Loop Initialization — reset L_Port

Loop Initialization (LIP(AL_PD,AL_PS)) is used by the originating L_Port (identified by AL_PS) to reset the NL_Port (identified by AL_PD). All L_Ports shall treat this LIP as specified in 7.8.3, however, the NL_Port at AL_PD shall also perform a vendor specific reset. If AL_PD = hex 'FF', a vendor specific reset shall be performed by all L_Ports (including those without an AL_PA, but not the one at AL_PS).

### 7.8.6 Loop Initialization — reserved

Loop Initialization (LIPba) is reserved for future use. The 'b' and 'a' characters of the LIPba Ordered Set shall contain one of the neutral disparity values from Table 1, and the LIPba shall not be a member of any LIP defined in 7.8.1 through 7.8.5. A LIPba shall be treated just like any other LIP (see 10.5).

Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

# 8 L_Port operation

## 8.0 Overview

To simplify L_Port design and minimize Transmission Word propagation delay, the following rules apply:

— all routing decisions by the LPSM (except during the Loop INITIALIZATION process) shall be made based on the AL_PA in the Primitive Signals (i.e., during normal operation, no LPSM routing decisions are made based on frame content);

— logging errors that are detected when retransmitting Transmission Words is optional; and,

NOTE While an L_Port is not required to log errors encountered while retransmitting Transmission Words, fault isolation and error analysis may be enhanced by doing so.

— Transmission Words are not routed to the FC-2 of the NL_Port in the ARBITRATING and MONITORING states unless REPLICATE is set to TRUE(1) (see 7.3).

The maximum delay of a Transmission Word through an L_Port in the MONITORING or ARBITRATING state should not exceed twelve (12) Transmission Word periods. See 8.2.2 for maximum round trip latency and Annex A for L_Port Elasticity buffer management.

The following steps provide an example for how an L_Port transfers one or more frames on a Loop.

a) The L_Port requests the LPSM to obtain access to the Loop.

b) The LPSM enters the ARBITRATING state and transmits its ARB(AL_PA) in place of the appropriate received Fill Word (see 7.1) until a matching ARB(AL_PA) is received. When the matching ARB(AL_PA) is received, the L_Port opens the Loop (i.e., stops retransmitting received Transmission Words).

c) The LPSM transmits OPNy to establish a point-to-point Loop circuit on the Loop with another L_Port. OPNy may be followed by frame(s). The number of frames that can immediately be transmitted is based on BB_Credit (see 8.3.4).

d) Either L_Port (of the Loop circuit) may transmit CLS when the L_Port desires to close the Loop circuit. When an L_Port receives CLS, it completes transmitting its frame(s), retransmits the CLS, and closes its end of the Loop circuit. When the CLS returns to the L_Port which originated the CLS, this L_Port closes its end of the Loop circuit.

NOTE Since either open L_Port may transmit CLS, an L_Port should be prepared to handle CLS simultaneously with or on the next Transmission Word after entering the XMITTED CLOSE state.

## 8.1 History

### 8.1.1 Access fairness history

The access fairness algorithm requires four memory elements that shall be maintained and used by each L_Port (see 8.4 for management requirements of each memory element):

a) **ACCESS** — the value of this variable is used by an L_Port to determine the status of the fairness window (i.e., whether the L_Port may arbitrate for access to the Loop). If ACCESS is FALSE(0), then an L_Port which is using the fairness algorithm, shall not arbitrate for access to the Loop; if ACCESS is TRUE(1), then an L_Port may arbitrate for access to the Loop and the L_Port may use the TRANSFER state to open a Loop circuit with another L_Port.

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

b) **ARB_WON** — the value of this variable is used by an L_Port to indicate that this L_Port has won arbitration. If ARB_WON is FALSE(0), then the L_Port did not win arbitration; if ARB_WON is TRUE(1), then the L_Port won arbitration.

c) **ARB_PEND** — the value of this variable is used by an L_Port which has been opened while arbitrating to remember that it was arbitrating. If ARB_PEND is FALSE(0), then the L_Port was not arbitrating; if ARB_PEND is TRUE(1), the L_Port was arbitrating.

NOTE This history variable forces the L_Port to finish arbitrating even if the L_Port no longer desires access to the Loop to assure that the fairness window is reset.

d) **XMIT_2_IDLES** — the value of this variable is used by an L_Port to remember that after receiving ARB(F0), two (2) Idles shall be transmitted when Idle is received. The current Fill Word when set to Idle shall not be modified until XMIT_2_IDLES is FALSE(0).

NOTE Since Idle is used to reset the fairness window, by transmitting two Idles, the probability of at least one Idle traversing the Loop is increased. If only one Idle is transmitted, it could be removed by another L_Port if that L_Port needs to delete a Fill Word for clock skew.

### 8.1.2 Duplex mode history

The OPEN, OPENED and RECEIVED CLOSE state requires one memory element, called DUPLEX, to determine whether the L_Port is allowed to originate Data frames. If DUPLEX is FALSE(0), the Loop circuit is operating in half-duplex mode; if DUPLEX is TRUE(1), the Loop circuit is operating in full-duplex mode (see 8.4 for management requirements of DUPLEX).

### 8.1.3 Replicate mode history

The MONITORING and ARBITRATING states require one memory element, called REPLICATE, to remember if OPNr had been received. If REPLICATE is FALSE(0), the states operate normally; if REPLICATE is TRUE(1), all received Transmission Words (except for normal Fill Word processing - updating the CFW appropriately) shall be retransmitted and also shall be provided to the FC-2 of the NL_Port for further processing. (See 7.3 and 8.4.3, item 13 and item 14.)

The OPEN state requires REPLICATE to remember that OPNr was transmitted in the ARBITRATION WON or TRANSFER state. If REPLICATE is FALSE(0), the state operates normally; if REPLICATE is TRUE(1), the L_Port may originate additional OPNr's; the L_Port shall not use BB_Credit management. (See 7.3 and 8.4.3, item 15 and item 16.)

### 8.1.4 Operational mode history

An L_Port uses two memory elements to record the L_Port's operational mode:

a) **PARTICIPATE** — set TRUE(1) if the L_Port has an AL_PA, set FALSE(0) if the L_Port does not have an AL_PA.

b) **BYPASS** — set TRUE(1) if the L_Port activates its optional Port Bypass Circuit, set FALSE(0) if the L_Port deactivates its optional Port Bypass Circuit.

REPEAT is a symbol that is defined to simplify the LPSM description. REPEAT is TRUE(1), if PARTICIPATE is FALSE(0) or BYPASS is TRUE(1) or both. REPEAT is FALSE(0), if PARTICIPATE is TRUE(1) and BYPASS is FALSE(0). When REPEAT is TRUE(1), the LPSM repeats most incoming Transmission Words (except for normal Fill Word processing—updating the CFW appropriately) without responding to them. When REPEAT is FALSE(0), the LPSM actively participates on the Loop.

The combined values of the PARTICIPATE and BYPASS variables record the four L_Port operational modes:

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

— **Participating** (PARTICIPATE = 1, BYPASS = 0) — the L_Port has an AL_PA and is enabled into the Loop. The L_Port may use the Loop and respond to all requests directed to it. This is the normal operational mode in which most Loop access occurs. In this mode, REPEAT is FALSE(0).

— **Non-Participating** (PARTICIPATE = 0, BYPASS = 0) — the L_Port does not have an AL_PA, but is enabled into the Loop. The L_Port repeats transmission words (except for normal Fill Word processing—updating the CFW appropriately) and only responds to a limited number of requests such as Loop Initialization. If the L_Port wishes to obtain an AL_PA and participate in the Loop, the L_Port may initiate Loop Initialization; it shall attempt to obtain an AL_PA if Loop Initialization occurs. In this mode, REPEAT is TRUE(1).

— **Participating Bypassed** (PARTICIPATE = 1, BYPASS = 1) — the L_Port has an AL_PA, but is bypassed (i.e., not enabled) from the Loop. The L_Port activates its optional Port Bypass Circuit if one is present. The L_Port also repeats transmission words (except for normal Fill Word processing—updating the CFW appropriately) in case no Port Bypass Circuit is present. The L_Port shall respond to an LPEyx directed to its AL_PA. In this mode, REPEAT is TRUE(1).

— **Non-Participating Bypassed** (PARTICIPATE = 0, BYPASS = 1) — the L_Port does not have an AL_PA and is bypassed (i.e., not enabled) from the Loop. The L_Port activates its optional Port Bypass Circuit if one is present. The L_Port also repeats transmission words (except for normal Fill Word processing—updating the CFW appropriately) in case no Port Bypass Circuit is present. The L_Port does not respond to any Primitive Signal or Primitive Sequences directed to a specific AL_PA. In this mode, REPEAT is TRUE(1).

### 8.1.5 DHD received history

The OPENED state requires one memory element if DHD is supported, called DHD_RCV. This variable is set to TRUE(1) if DHD is received. The variable is checked when the L_Port in the OPENED state has completed all transmissions to the L_Port in the OPEN state. If DHD_RCV is FALSE(0), then the L_Port may continue to wait to receive CLS (normal operation) or it may transmit CLS. If DHD_RCV is TRUE(1), then the L_Port shall transmit CLS. (See annex C.)

### 8.1.6 ARB(FF) history

The MONITORING state uses a memory element, called ARBf_SENT, to indicate that the L_Port has requested the LPSM to modify its CFW to ARB(FF) from Idles. If ARBf_SENT is FALSE(0), the current Fill Word is managed normally; if ARBf_SENT is TRUE(1), then the current Fill Word shall remain ARB(FF) until an ARB(AL_PA) is received or ARB(F0) (if REPEAT is TRUE(1)) is received. (See 8.4.)

## 8.2 Timeouts

### 8.2.1 FC-PH timeout values

Timeout values (e.g., R_T_TOV) and related timeout procedures in FC-PH, shall be used as appropriate.

### 8.2.2 Arbitrated Loop timeout value

The Arbitrated Loop timeout value (AL_TIME) is 15 ms, which represents two times the worst case round-trip latency for a very large Loop. AL_TIME is based on twice the sum of the following values:

— 134 times an L_Port internal latency of six (6) Transmission Word periods at 1,062 5 Gbits/s of the L_Port and

— 134 times 10 km, the cable latency (5 ns/m).

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

AL_TIME is primarily used during the INITIALIZATION process to control events that require a Loop round-trip latency to complete. The sequencing of these events must be coordinated between multiple L_Ports, which requires that all L_Ports use AL_TIME consistently. During the INITIALIZATION process, L_Ports shall measure AL_TIME with a tolerance of -0 %, +20 % (i.e., an AL_TIME timeout shall expire from a minimum of 15 ms up to a maximum of 18 ms).

NOTE It is conceivable that the maximum round-trip delay of a Loop configuration is greater than the AL_TIME. However, determining interoperability when using a different AL_TIME value is outside the scope of this standard.

### 8.2.3 Loop timeout

The Loop timeout value (LP_TOV) is 2 s. LP_TOV is used to keep a Loop from deteriorating due to protocol errors or lost Ordered Sets. For example, LP_TOV is used to reset the fairness window (see 4.3) and during the INITIALIZATION process to time start-up events (see 10.5.4).

## 8.3 Operational characteristics

### 8.3.1 Transmission Word

#### 8.3.1.1 Power-on Transmission Words

At power-on, the L_Port shall turn off its transmitter until it is ready to participate in Loop Initialization.

#### 8.3.1.2 Invalid Transmission Words and Transmission Characters

An L_Port shall make substitutions for invalid received Transmission Words and Transmission Characters (see 8.4) as follows:

— in the MONITORING or ARBITRATING states:

  – if an invalid Transmission Word is detected, the L_Port shall substitute the CFW for that Transmission Word.

  – if an invalid Beginning Running Disparity is detected on an Ordered Set, the L_Port shall substitute the CFW.

— in any other state the L_Port shall follow the rules defined in FC-PH and clause 29.

### 8.3.2 Clock skew management

When an L_Port implements receive and transmit clocks with different reference sources, a buffer is required between the receiver and transmitter logic to manage the clock frequency and phase differences (see annex G for clock design options). When a buffer is required, the L_Port shall implement the buffer as defined in annex A. To prevent buffer over-run or under-run, the L_Port shall use the clock skew management rules defined in annex A to control the level of data.

When processing Transmission Words between frames, any ARByx shall be treated the same as Idle. Fill Words or any Ordered Set defined for use as a Primitive Sequence shall be treated equally. (See clause 7, annex A and annex G and FC-PH.)

Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

### 8.3.3 Error detection and recovery

Each state in 8.4 contains the procedures for handling failures. State transitions are considered to take place instantaneously and no error detection takes place during a state transition. Any failure or subsequent state request that occurs during a state transition shall be detected in the subsequent state.

Following recovery from a failure, the L_Port shall comply with the provisions for Sequence integrity, error detection, and Sequence recovery specified in FC-PH.

### 8.3.4 BB_Credit and Available_BB_Credit

#### 8.3.4.0 BB_Credit overview

BB_Credit and Available_BB_Credit are used when transmitting a SOFc1, a Class 2, or a Class 3 frame. Before Login, the "Alternate BB_Credit Management" bit (see FC-PH) and BB_Credit shall be set to 0 and one (1) in the OLD-PORT state and to 1 and zero (0) in the OPEN-INIT-START state, respectively (see 8.4.3 item 21 and item 23, and 10.5.4). During Login, BB_Credit shall be set to a value that represents the number of receive buffers that the L_Port shall guarantee to have available when a Loop circuit is established.

When on a Loop, L_Ports have unique characteristics (unlike point-to-point or Fabric-attached N_Ports):

— Loop circuits are dynamic;

— if not properly managed, an L_Port may have frames in the receive buffers from the previous Loop circuit when a new Loop circuit is established; even a BB_Credit equal to one (1) may overrun the receive buffers;

— using BB_Credit equal to zero (0) requires a turn-around delay and impedes performance at the beginning of each Loop circuit; and,

— balancing BB_Credit at the end of a Loop circuit may impede performance.

"Alternate BB_Credit Management" is used to achieve the best performance while addressing these unique Loop characteristics. To avoid a turn-around delay at the beginning of a Loop circuit, L_Ports may take advantage of the BB_Credit which is established during Login. Although balancing BB_Credit is not required (receive buffers may be emptied after the Loop circuit is closed), the BB_Credit value represents the number of receive buffers that an L_Port is assumed to have available when the next Loop circuit is established. Therefore, an L_Port shall not enter the MONITORING state until the number of available receive buffers is at least equal to the largest BB_Credit value which the L_Port disseminated during Login.

BB_Credit in the following discussion is identified as *open* BB_Credit (i.e., the BB_Credit of the L_Port which transmits the OPNy) and *opened* BB_Credit (i.e., the BB_Credit of the L_Port which receives the OPNy) (see annex F).

A positive opened BB_Credit allows the L_Port to follow OPNy with frames, without waiting for an R_RDY (i.e., there is no round-trip delay).

NOTE "Alternate BB_Credit Management" is written from the view of the L_Port which transmits the OPNy. This L_Port knows the opened BB_Credit and its open BB_Credit, but it has no knowledge of what the opened L_Port will use for the open BB_Credit. The L_Port which receives the OPNy may choose to use the open BB_Credit, or immediately use Available_BB_Credit.

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

#### 8.3.4.1 BB_Credit management per Loop circuit

For each Loop circuit, BB_Credit for the L_Ports in the OPEN and OPENED state is any value less than or equal to the BB_Credit which the other L_Port in the Loop circuit advertised during Login. L_Ports shall not have more than 255 outstanding R_RDYs during any Loop circuit.

NOTE If the L_Port in the OPEN state is using an opened BB_Credit of zero (0), a Loop turn-around delay is required (i.e., an R_RDY must be received) before the L_Port is allowed to transmit the first frame.

**The L_Port which transmits OPNy** shall obey the following rules for transmitting R_RDYs:

NOTE Since a minimum of six (6) Fill Words are required between the OPNy and the first frame, the L_Port may transmit one R_RDY instead of one Fill Word without any performance penalty. The number of R_RDYs which the L_Port transmits before the first frame is a balance between delaying the transmission of the first frame and delaying receiving frames.

— If the open BB_Credit equals zero (0), the L_Port shall transmit one R_RDY for each currently available receive buffer.

— If the open BB_Credit is greater than zero (0), the L_Port shall transmit one R_RDY for each BB_Credit which this L_Port advertised plus one R_RDY for each additional available receive buffer.

— If CLS is received before all R_RDYs have been transmitted, the remaining R_RDYs are not required to be transmitted in the Loop circuit.

The L_Port may transmit the number of frames specified by the opened BB_Credit before receiving an R_RDY. The L_Port shall discard one received R_RDY for each of these frames sent. When the number of discarded R_RDYs equals the opened BB_Credit, the L_Port shall use Available_BB_Credit management.

**The L_Port which receives OPNy** shall obey the following rules for transmitting R_RDYs:

NOTE The number of R_RDYs which the L_Port transmits before the first frame is a balance between delaying the transmission of the first frame and delaying receiving frames.

— if the opened BB_Credit equals zero (0), the L_Port shall transmit one R_RDY for each currently available receive buffer.

— if the opened BB_Credit equals zero (0), the L_Port may transmit CLS if there are no available receive buffers.

— if the opened BB_Credit is greater than zero (0), the L_Port shall transmit one R_RDY for each BB_Credit which this L_Port advertised plus one R_RDY for each additional available receive buffer.

— If CLS is received before all R_RDYs have been transmitted, the remaining R_RDYs are not required to be transmitted in the Loop circuit.

The L_Port shall initialize the open BB_Credit to zero (0). If the L_Port can determine the open BB_Credit, it may transmit the number of frames specified by the open BB_Credit. If the L_Port transmitted frames based on the open BB_Credit, it shall discard one received R_RDY for each of these frames sent. When the number of discarded R_RDYs equals the open BB_Credit, the L_Port shall use Available_BB_Credit management.

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

#### 8.3.4.2 Available_BB_Credit management per Loop

Once the L_Port has discarded the same number of R_RDYs as BB_Credit, the L_Port shall use Available_BB_Credit for transmitting additional frames.

Available_BB_Credit is one of the following values:

— zero (0) – the initial value until one or more R_RDYs have been received; or.

— the number of R_RDYs received less the number of frames transmitted.

The L_Port may transmit the number of frames specified by Available_BB_Credit. For each frame sent, Available_BB_Credit is decremented by one (1); for each R_RDY received, Available_BB_Credit is incremented by one (1). As long as Available_BB_Credit greater than zero, the L_Port may transmit frames during this Loop circuit.

## 8.4 Loop Port State Machine (LPSM)

### 8.4.0 LPSM overview

The Loop Port State Machine (LPSM) shall be used to define the behavior of the L_Ports when they require access to and use of a Loop. The following subclauses specify the state names, state diagram, and item references for the LPSM.

### 8.4.1 State names

The state names and numbers used in the LPSM, along with a brief description, are given below. Reference items for each state are considered part of each state. The reference item numbers are identified in the L_Port state machine diagram in 8.4.2. The reference item text follows the state machine diagram in 8.4.3.

**MONITORING (0):** The LPSM is transmitting received Transmission Words and, if it is in the Participating mode, monitoring the Loop for certain Ordered Sets (e.g., OPNy and OPNr). This is the default state of any L_Port.

**ARBITRATING (1):** The LPSM is arbitrating for control of the Loop.

**ARBITRATION WON (2):** The LPSM has received a matching ARB(AL_PA) (i.e., AL_PA = AL_PA of this L_Port) while arbitrating.

**OPEN (3):** The LPSM has transmitted OPNy while in the ARBITRATION WON state. Normal FC-2 protocol follows.

**OPENED (4):** The LPSM has received a matching OPNy (i.e., y = AL_PA of this L_Port) while in the MONITORING or ARBITRATING state. Normal FC-2 protocol follows.

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

**XMITTED CLOSE (5):** The LPSM has transmitted CLS and intends to relinquish control of the Loop.

**RECEIVED CLOSE (6):** The LPSM has received CLS.

**TRANSFER (7):** The LPSM, while in the OPEN state, has transmitted CLS and requires the Loop to communicate with another L_Port.

**INITIALIZATION process (8):** The LPSM is initializing or re-initializing.[3]

**OLD-PORT (A):** The LPSM has determined that it wants to operate in a point-to-point mode utilizing FC-PH protocol without FC-AL.

### 8.4.2 State diagram

The state diagram is shown in Figure 4. The numbered reference items for states and state transitions in 8.4.3 are normative parts of the LPSM definition. If the details were in the state diagrams, the diagrams would be difficult to read and interpret.

States are identified with a single letter or digit followed by a single colon character (e.g., 6:). Transitions identified as "(Xn):", where *n* is a single digit or letter, represent valid transitions from multiple states to the ending state, *n*, caused by an event outside the steady state operation of the LPSM. A transition identified as "(mn):", where *m* and *n* are single digits or letters, represents a transition from state *m* to state *n*. Each transition and state is accompanied by detailed specifications and requirements identified by the numbered reference item.

[3] The INITIALIZATION process encompasses the INITIALIZING and OPEN-INIT states that were defined in the first publication of this standard.

Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

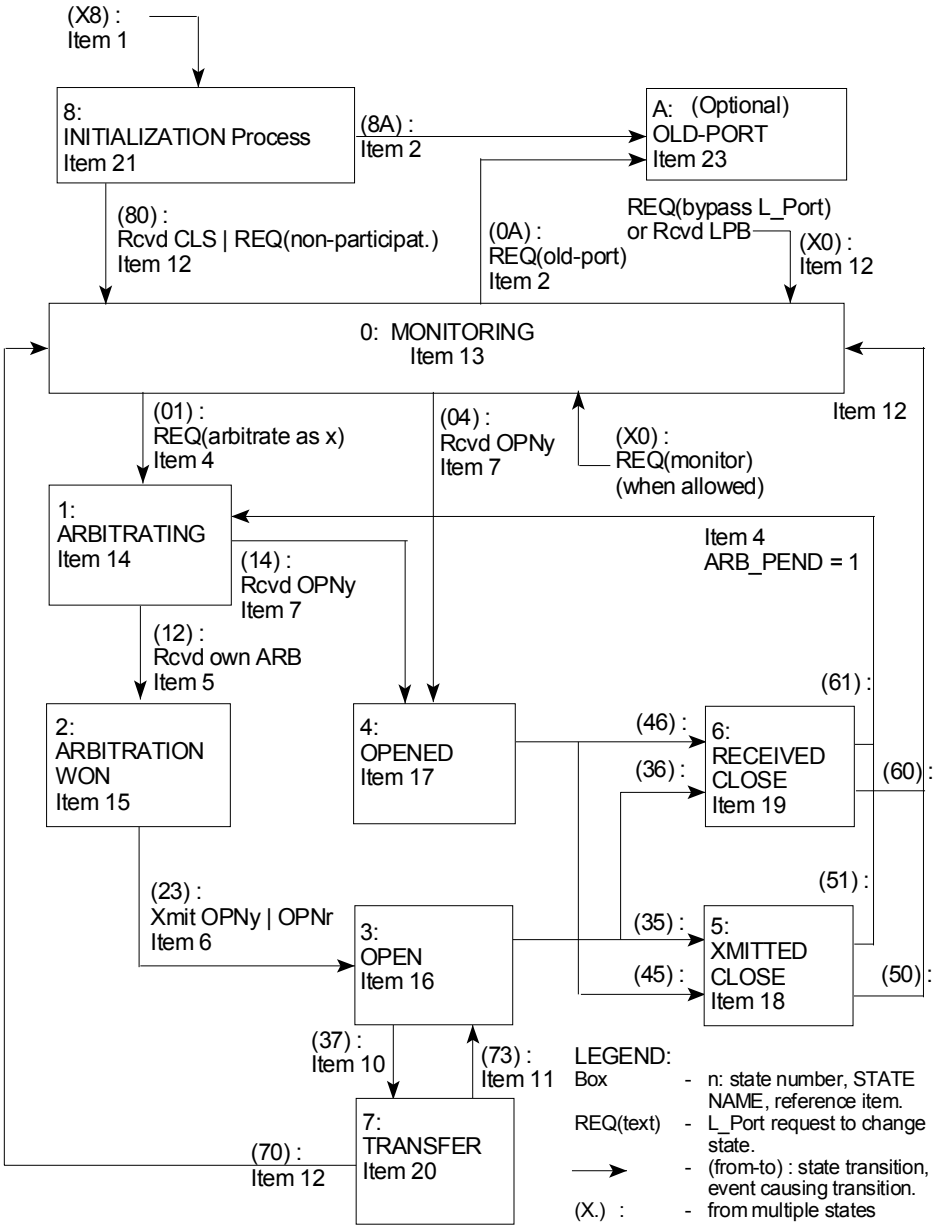


**Figure 4 — State Diagram**

Documents
Document Sharing Hub
Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV

### 8.4.3 Reference items

For detailed information about a state or state transition, refer to the item number in the list below.

For conditions that are not explicitly listed in this section as causing state changes to occur, the LPSM shall remain in the current state.

1 **Transition (X8):** This transition[4] shall be made at power-on of an L_Port, after detecting a failure (see clause 10 and FC-PH), when a LIP is recognized, or from any state when the L_Port requests it (see 10.5.3).

All fibre-type dependent operations shall be complete before making this transition (e.g., Open Fibre Control) (see FC-PH).

2 **Transition (0A:; (8A):** The LPSM shall make the transition to the OLD-PORT state (if supported) (see item 13, item 23, and 10.5.4).

3 **Transition (01):** The LPSM shall make the transition to the ARBITRATING state (see item 13 and item 14).

4 **Transitions (51):, (61):** The LPSM shall make the transition to the ARBITRATING state (see item 14, item 18, and item 19).

5 **Transition (12):** The LPSM shall make the transition to the ARBITRATION WON state (see item 14 and item 15).

6 **Transition (23):** The LPSM shall make the transition to the OPEN state (see item 15 and item 16).

7 **Transitions (04):, (14):** The LPSM shall make the transition to the OPENED state (see item 13, item 14, and item 17).

8 **Transitions (35):, (45):** The LPSM shall make the transition to the XMITTED CLOSE state (see item 15, item 16, item 17 and item 18).

9 **Transitions (36):, (46):** The LPSM shall make the transition to the RECEIVED CLOSE state (see item 16, item 17, and item 19).

10 **Transition (37):** The LPSM shall make the transition to the TRANSFER state (see item 16 and item 20).

11 **Transition (73):** The LPSM shall make the transition to the OPEN state (see item 16 and item 20).

12 **Transitions (X0):, (50):, (60):, (70):** The LPSM shall make the transition to the MONITORING state (see item 13, item 15, item 18, and item 20).

[4] Some implementations may choose to allow an L_Port to exit the INITIALIZATION process without completing initialization. These L_Ports 'initialize' outside the scope of this standard.

Documents
Document Sharing Hub

Full document details: ISO_IEC 14165-122_2005+AMD1_2008 CSV