FIBRE CHANNEL
A Functional Performance model of the Fibre Channel – Arbitrated Loop (FC-AL)

Foresight Systems M&S

Attention: Please Read This Page

We are currently in the process of updating this white paper. While we believe that the material is important and will be interesting to current users of Foresight Systems software, we have been through some changes since the paper was originally prepared.

Foresight Systems, Inc. was a predecessor company to Foresight Systems M&S. All references to Foresight Systems, Inc. are obsolete. Readers should assume that Foresight Systems M&S replaces all instances of Foresight Systems, Inc.

The current contact information for Foresight Systems M&S is:

Foresight Systems M&S
12550 N. 89th St.
Scottsdale, AZ 85260

http://www.foresightsystems-mands.com
fs_marketing@foresightsystems-mands.com

The diagrams and illustrations included here were prepared with versions of the software current at the time that the paper was prepared. Some of them may not be identical to equivalent images from the current version of the software. As part of the update, we will be generating new images with better resolution. We apologize for any potential confusion.

If you have questions, concerns, or issues, please contact:

Dean Stevens
President, Commercial Systems Group
Foresight Systems M&S
deans@foresightsystems-mands.com
(408) 278-3983

FIBRE CHANNEL

A Functional Performance model of the Fibre Channel – Arbitrated Loop (FC-AL)

Foresight Systems, Inc.
8217 Shoal Creek Blvd.
Austin, TX 78757
512-420-9210
Fax: 512-420-9035
info@foresight-systems.com
www.foresight-systems.com

Introduction

In recent years several technical developments have converged to a bigger than ever need for extremely fast data links. High performance computers have become the focus of much attention in the data communications industry. Performance improvements have spawned increasingly data-intensive and high-speed networking applications, such as multimedia and scientific visualization. However, the existing network interconnects between computers and I/O devices are unable to run at the speeds needed.

The intention of the Fibre Channel (FC) is to develop practical, inexpensive, yet expendable means of quickly transferring data between workstations, mainframes, super computers, desktop computers, storage devices, displays and other peripherals. Fibre Channel is the general name of an integrated set of standards being developed by the American National Standards Institute (ANSI).

There are two basic types of data communication between processors and between processors and peripherals: channels and networks. A channel provides a direct or switched point-to-point connection between the communicating devices. A channel is typically hardware-intensive and transports data at high speed with low overhead. In contrast, a Network is an aggregation of distributed nodes (like workstations, file servers or peripherals) with it's own protocol that supports interaction among these nodes. A network has relatively high overhead since it is software-intensive, and consequently slower than a channel. Networks can handle a more extensive range of tasks than channels as they operate in an environment of unanticipated connections, while channels operate amongst only a few devices with pre-defined addresses. Fibre Channel attempts to combine the best of these two methods of communication into a new I/O interface that meets the needs of channel users and also network users.

Although it is called Fibre Channel, its architecture doesn't represent either a channel or a real network topology. It allows for an active intelligent interconnection scheme, called a Fabric, to connect devices. All a Fibre channel port has to do is to manage a simple point-to-point connection between itself and the Fabric.

Fibre channel is a high performance serial link supporting its own, as well as higher level protocols such as FDDI, SCSI, HIPPI and IPI. The Fibre Channel standard addresses the need for very fast transfers of large amounts of information. The fast (up to 1 Gbit/s) technology can be converted for Local Area Network technology by adding a switch specified in the Fibre Channel standard, that handles multipoint addressing. There is a perspective as an I/O technology and a Local Area Network technology as well. Another advantage of Fibre Channel is that it gives users one port that supports both channel and network interfaces, unburdening the computers from large number of I/O ports. FC provides control and complete error checking over the link.

Fibre Channel Topology

In Fibre Channel terms the switch connecting the devices is called the Fabric. The link is the two unidirectional fibres transmitting to opposite directions with their associated transmitter and receiver. Each fibre is attached to a transmitter of a port at one end and a receiver of another port at the other end. When a Fabric is present in the configuration, the fibre may attach to a node port (N_Port) and to a port of the Fabric (F_Port).

Since a Fibre channel system relies on ports logging in with each other and the Fabric, it is irrelevant whether the Fabric is a circuit switch, an active hub or a loop. The topology can be selected depending on system performance requirements or packaging options. Possible FC topologies include point-to-point, crosspoint switched or arbitrated loop.

FC operates at a wide variety of speeds (133 Mbit/s, 266 Mbit/s, 530 Mbit/s, and 1 Gbits/s) and on three types of both electrical and optical media. Transmission distances vary depending on the combination of speed and media. The single mode fibre optic media using long-wave laser light source gives the highest performance (10 km maximum distance at 1 Gbit/s).

The Fibre Channel Model

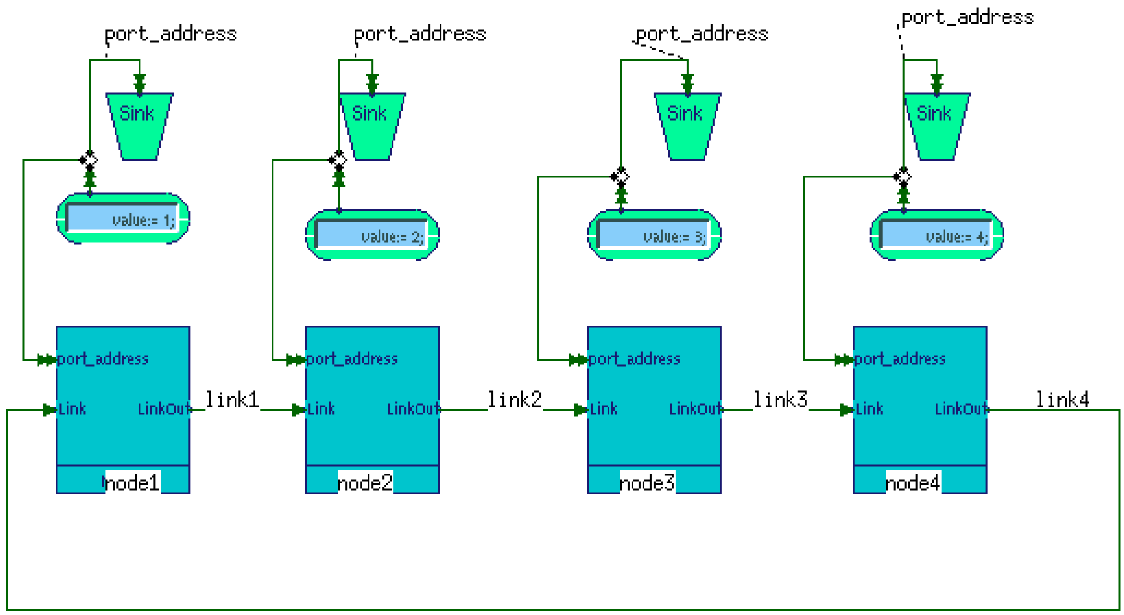

Figure 1. *4_node_net* DFD

The Fibre Channel model begins with the DFD labeled *4_node_net* shown in Figure 1. This arrangement has four nodes, labeled 1 through 4. Each is identical and comprise the detail of the model. Port address is an identifying integer and the "link" flows are comprised of "frames" amd "primitives" which pass between the nodes. Frames are packages of data with time parmeters (more about these later) and other parts that include a header, start and end of file markers. Primitives are associated with control functions, which will also be discussed further.

Within each node the layers of the Fibre Channel model are displayed. As shown in figure 2, the layers are: FC-4 (Application Program), FC-3 (Upper Level Protocal), FC-2, FC-1 and FC-0. Each layer performs specific functions. Between each layer the "IU" flows contain the destination id, the "payload" and time parameters. Each layer will be discussed in detail.

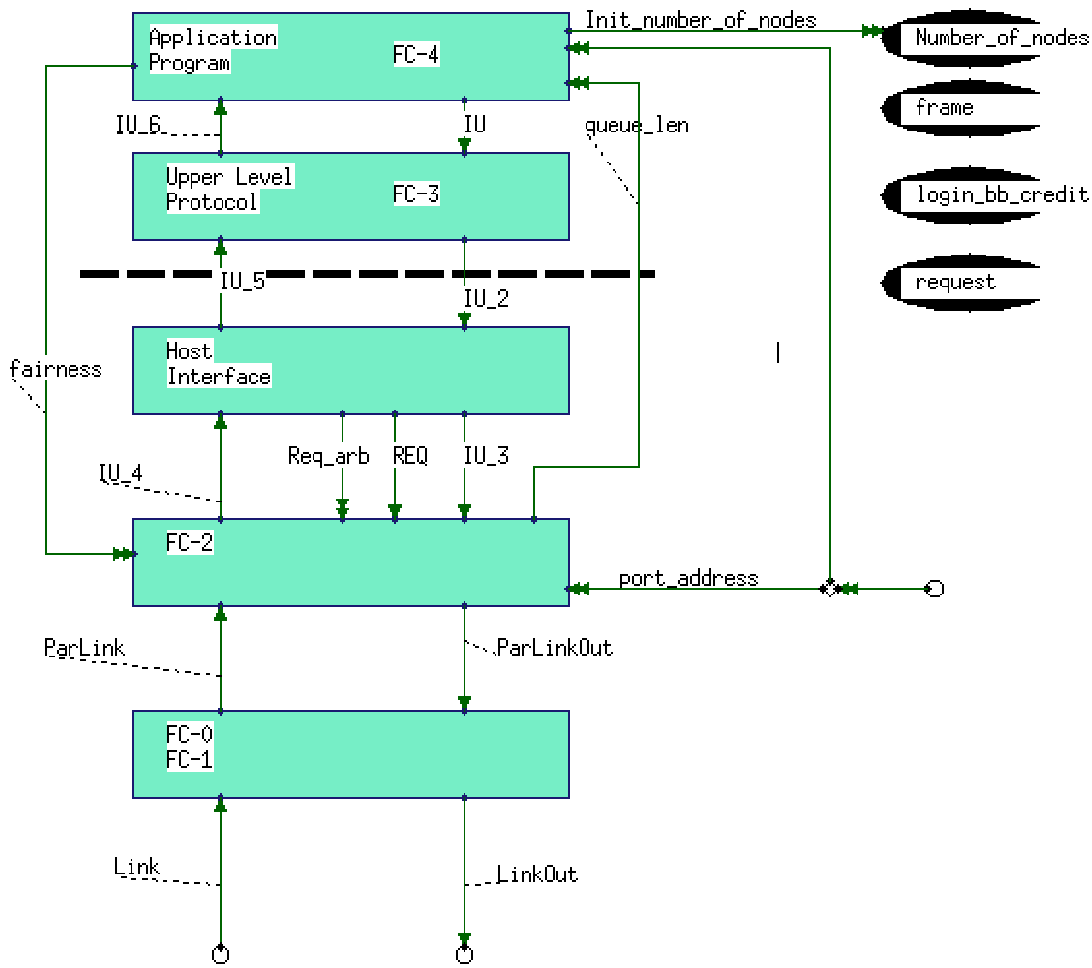

Figure 2. Node DFD

FC-4 Layer

FC-4, the highest level in the FC structure defines the application interfaces that can execute over Fibre Channel. It specifies the mapping rules of upper layer protocols using the FC levels below. Fibre Channel is equally adept at transporting both network and channel information and allows both protocol types to be concurrently transported over the same physical interface.

The following network and channel protocols are currently specified or proposed as FC-4s:

Small Computer System Interface (SCSI)
Intelligent Peripheral Interface (IPI)
High Performance Parallel Interface (HIPPI) Framing Protocol
Internet Protocol (IP)
ATM Adaptation Layer for computer data (AAL5)
Link Encapsulation (FC-LE)
Single Byte Command Code Set Mapping (SBCCS)
IEEE 802.2

Figure 3 depicts the FC-4 layer , Applications Program DFD of the model. At this level the user can control the simulation by choosing the fairness policy, periodic flows of data (size and frequency), the uploading of files of varied size, credit policy, the destination address and can select data collection to external files for later analysis.

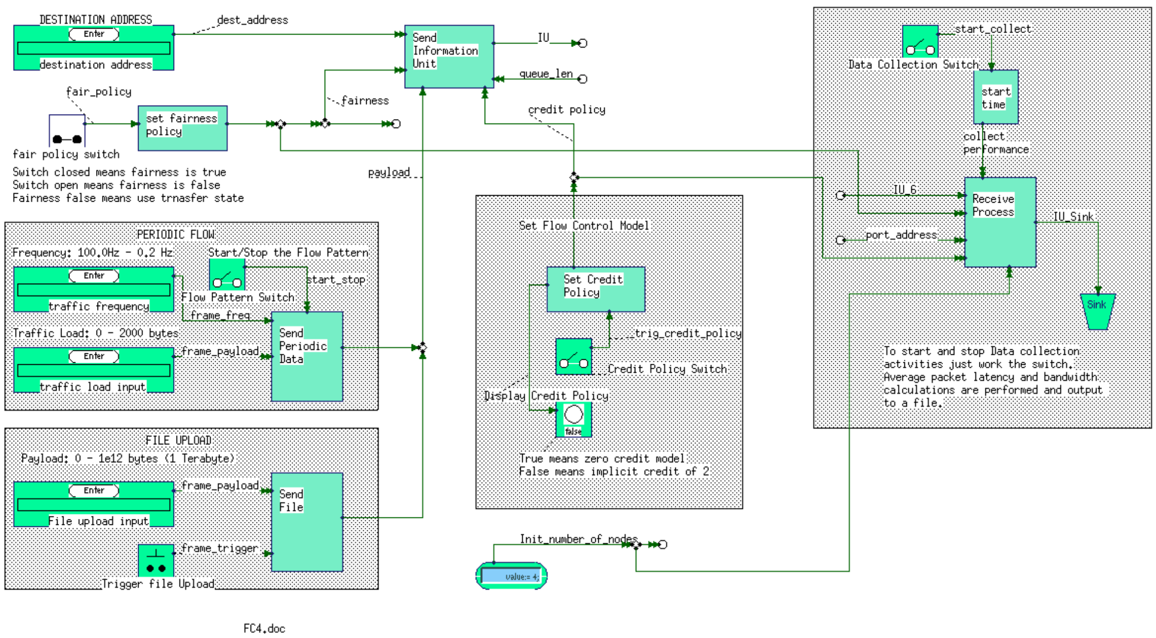

Figure 3. FC-4 (Application Program) DFD

FC-3 Layer

The FC-3 level of the FC standard is intended to provide the common services required for advanced features such as:

Striping -To multiply bandwidth using multiple N_ports in parallel to transmit a single information unit across multiple links.

Hunt groups - The ability for more than one Port to respond to the same alias address. This improves efficiency by decreasing the chance of reaching a busy N_Port.

Multicast - Multicast delivers a single transmission to multiple destination ports. This includes sending to all N_Ports on a Fabric (broadcast) or to only a subset of the N_Ports on a Fabric.

The present FC-3 layer of the model, Upper Level Protocol, simply passes data through at this point. It will be built in the near future.

FC-2 Layer (General)

The Signaling Protocol (FC-2) level serves as the transport mechanism of Fibre Channel. The framing rules of the data to be transferred between ports, the different mechanisms for controlling the three service classes and the means of managing the sequence of a data transfer are defined by FC-2. To aid in the transport of data across the link, the following building blocks are defined by the standard:

Ordered Set
Frame
Sequence
Exchange
Protocol

Ordered Set

The Ordered Sets are four byte transmission words containing data and special characters which have a special meaning. Ordered Sets provide the availability to obtain bit and word synchronization, which also establishes word boundary alignment. An Ordered Set always begins with the special character K28.5. Three major types of Ordered Sets are defined by the signaling protocol.

The Frame delimiters (the Start-of-Frame (SOF) and End-of-Frame (EOF) Ordered Sets) are Ordered Sets which immediately precede or follow the contents of a Frame. There are multiple SOF and EOF delimiters defined for the Fabric and N_Port Sequence control.

The two Primitive Signals: Idle and Receiver Ready (R_RDY) are Ordered Sets designated by the standard to have a special meaning. An Idle is a Primitive Signal transmitted on the link to indicate an operational Port facility ready for Frame transmission and reception. The R_RDY Primitive Signal indicates that the interface buffer is available for receiving further Frames.

A Primitive Sequence is an Ordered Set that is transmitted and repeated continuously to indicate specific conditions within a Port or conditions encountered by the receiver logic of a Port. When a Primitive Sequence is received and recognized, a corresponding Primitive Sequence or Idle is transmitted in response. Recognition of a Primitive Sequence requires consecutive detection of 3 instances of the same Ordered Set. The Primitive Sequences supported by the standard are Off-line (OLS), Not Operational (NOS), Link Reset (LR) and Link Reset Response (LRR).

Frame

The basic building blocks of an FC connection are the Frames. The Frames contain the information to be transmitted (Payload), the address of the source and destination ports and link control information. Frames are broadly categorized as Data frames and Link_control frames. Data frames may be used as Link_Data frames and Device_Data frames, link control frames are classified as Acknowledge (ACK) and Link_Response (Busy and Reject) frames. The primary function of the Fabric is, to receive the Frames from the source port and route them to the destination port. It is the FC-2 layer's responsibility to break the data to be transmitted into Frame size, and reassemble the Frames.

Each Frame begins and ends with a Frame Delimiter, The Frame Header immediately follows the SOF delimiter. The Frame Header is used to control link applications, control device protocol transfers, and detect missing or out of order Frames. An optional header may contain further link control information. A maximum 2112 byte long field (payload) contains the information to be transferred from a source N_Port to a destination N_Port. The 4 bytes Cyclic Redundancy Check (CRC) precedes the EOF delimiter. The CRC is used to detect transmission errors.

Sequence

A Sequence is formed by a set of one or more related Frames transmitted unidirectional from one N_Port to an other. Each Frame within a sequence is uniquely numbered with a Sequence Count. Error recovery, controlled by an upper protocol layer is usually performed at Sequence boundaries.

Exchange

An Exchange is composed of one or more non-concurrent sequences for a single operation. The Exchanges may be unidirectional or bi-directional between two N_Ports. Within a single Exchange, only one sequence may be active at any one time, but Sequences of different Exchanges may be concurrently active.

Protocol

The Protocols are related to the services offered by Fibre Channel. Protocols may be specific to higher-layer services, although Fibre Channel provides its own set of protocols to manage its operating environment for data transfer. The following Protocols are specified by the standard:

Primitive Sequence Protocols are based on Primitive Sequences and specified for link failure.
Fabric Login Protocol: The interchanging of Service Parameters of an N_Port with the fabric.
N_Port Login Protocol: Before performing data transfer, the N_Port interchanges its Service Parameters with another N_Port.
Data Transfer Protocol describes the methods of transferring Upper Layer Protocol (ULP) data using the Flow control management of Fibre Channel.
N_Port Logout Protocol is performed when an N_Port requests removal of its Service Parameters from the other N_Port. This may be used to free up resources at the connected N_Port.

Flow control

Flow control is the FC-2 control process to pace the flow of Frames between N_Ports and between an N_Port and the Fabric to prevent overrun at the receiver. Flow control is dependent upon the service classes. Class 1 Frames use end-to-end flow control, class 3 uses only buffer-to-buffer, class 2 Frames use both types of flow control.

Flow control is managed by the Sequence Initiator (source) and Sequence Recipient (destination) Ports using Credit and Credit_CNT. Credit is the number of buffers allocated to a transmitting Port. The Credit_CNT represents the number of data frames which have not been acknowledged by the Sequence Recipient.

The end-to-end flow control process paces the flow of Frames between N_Ports. In this case the Sequence Recipient is responsible for acknowledging the received valid data Frames by ACK Frames. When the number of receive buffers are insufficient for the incoming Frame, a "Busy", when a Frame with error is received a "Reject" Frame will be sent to the Initiator Port. The Sequence Initiator is responsible for managing EE_Credit_CNT. The N_Port login is used to establish EE_Credit.

The buffer-to-buffer flow control is managed between an N_Port and an F_Port or between N_Ports in point-to-point topology. Each port is responsible for managing BB_Credit_CNT. BB_Credit is established during the Fabric Login. The Sequence Recipient (destination) Port signals by sending a Receiver_Ready primitive signal to the transmitting Port whether it has free receive buffers for the incoming Frames.

Service Classes

To ensure efficient transmission of different types of traffic, FC defines three classes of service. Users select service classes based on the characteristics of their applications, like packet length and transmission duration, and allocate the services by the Fabric Login protocol.

Class 1 is a service which provides dedicated connections, in effect providing the equivalent of a dedicated physical connection. Once established, a Class 1 connection is retained and guaranteed by the Fabric. This service guarantees the maximum bandwidth between two N_Ports, so this is the best for sustained, high throughput transactions. In Class 1, Frames are delivered to the destination Port in the same order as they are transmitted.

Class 2 is a Frame-switched, connection-less service that allows bandwidth to be shared by multiplexing Frames from multiple sources onto the same channel or channels. The Fabric may not guarantee the order of the delivery and Frames may be delivered out of order. This service class can be used, when the connection setup time is greater than the latency of a short message. Both Class 1 and Class 2 send acknowledgment

Frames confirming Frame delivery. If delivery cannot be made due to congestion, a Busy frame is returned and the sender tries again.

Class 3 service is identical to Class 2, except that the Frame delivery is not confirmed. (Flow control is managed only on buffer level). This type of transfer, known as datagram provides the quickest transmission by not sending confirmation. This service is useful for real-time broadcasts, where timeliness is key and information not received in time is valueless. The FC standard also defines an optional service mode called intermix. Intermix is an option of Class 1 service, in which Class 1 Frames are guaranteed a special amount of bandwidth, but Class 2 and Class 3 Frames are multiplexed onto the channel, only when sufficient bandwidth is available to share the link.

The FC-2 layer DFD is shown in figure 4.

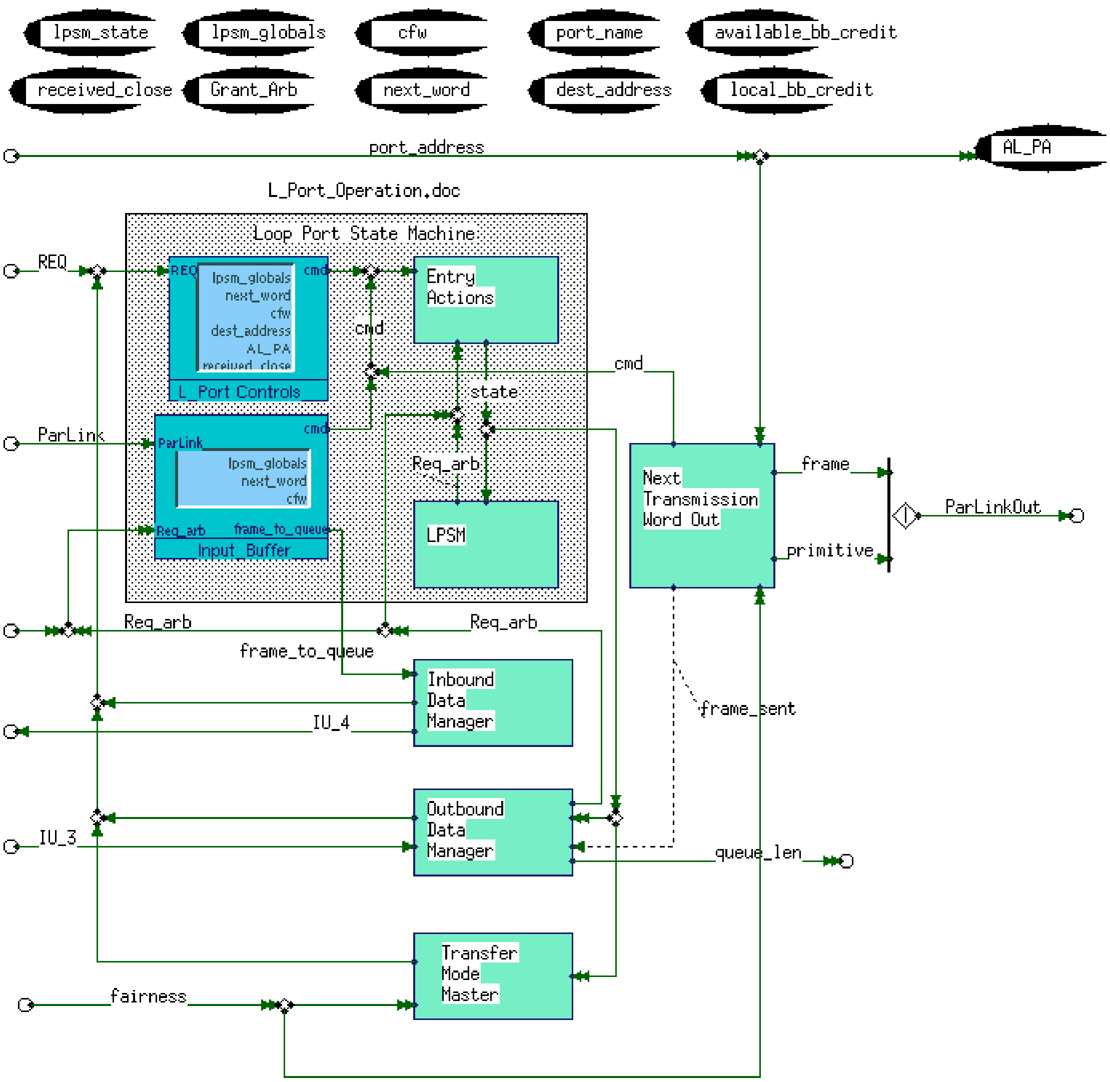

Figure 4. FC-2 DFD

The FC-2 layer is the most comprehnsive of the present model. Within this layer there are three parts with siginifciant detail, designated as the Loop Port State Machine. These three are: (see Figure 4) the L_Port_Controls re-useable, the Input_Buffer re-useable, and the LPSM -- the L Port State Machine STD. Each is discussed in turn below.

L_Port Operation

To simplify L_Port design and minimize Transmission Word propagation delay, the following general rules apply: all routing decisions by the LPSM (except during initialization) 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); there is no requirement for logging errors that are detected when re-transmitting Transmission Words and frame detection shall not be supported in the ARBITRATING and MONITORING states unless REPLICATE is set to TRUE(1). The maximum delay of a Transmission Word through an L_Port in the MONITORING or ARBITRATING state shall not exceed six (6) Transmission Word periods. The following steps show an example for an L_Port to transfer one or more ANSI X3.230, FC-PH frames on a Loop:

(1) The L_Port requests the LPSM to obtain access to the Loop.
(2) The LPSM transmits ARBx continuously until a matching ARBx is received. When ARBx is received, the L_Port opens the Loop (i.e., stops re-transmitting received Transmission Words).
(3) The LPSM transmits OPNy to establish a point-to-point circuit on the Loop. OPNy may be followed by ANSI X3.230, FC-PH frame(s). The number of frames that can immediately be transmitted is based on BB_Credit.
(4) Either L_Port may transmit a CLS when it finishes transmitting frames. The receiving L_Port completes transmitting its frame(s), re-transmits the CLS, and closes its end of the Loop. When the CLS returns to the L_Port which originated the CLS, this L_Port closes its end of the Loop.

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

History variables

Access fairness history

The access fairness algorithm requires three memory elements that shall be maintained and used by each L_Port:

(1) ACCESS the value of this variable is used by an L_Port to decide whether to use the fairness algorithm.
(2) ARB_WON the value of this variable is used by an L_Port to identify the L_Port which won arbitration.
(3) ARB_PEND the value of this variable is used by an L_Port with AL_PA =x to remember that it has transmitted one or more ARBx Primitive Signals.

Duplex mode history

The OPENED state requires one memory element, called DUPLEX, to determine how the L_Port was opened. If DUPLEX is FALSE(0), the circuit is operating in half-duplex mode; if DUPLEX is TRUE(1), the circuit is operating in full-duplex mode.

Replicate mode history

The MONITORING and ARBITRATING states require one memory element, called REPLICATE, to remember if an OPNr had been received. If REPLICATE is FALSE(0), the states operate normally; if REPLICATE is TRUE(1), all Transmission Words are received and re-transmitted and all frames are provided to FC-2 for further processing. The OPEN state requires REPLICATE to remember that OPNr was transmitted in the ARBITRATION WON state. If REPLICATE is FALSE(0), the state operates normally; if REPLICATE is TRUE(1), the L_Port shall discard all received frames. The ARBITRATION WON state requires REPLICATE to detect that the originator of the OPNr left the circuit without transmitting a CLS.

L_Port bypassed history

The MONITORING state requires one memory element, called LP_BYPASS, to determine whether the L_Port may originate Transmission Words on the Loop (e.g., ARBx, OPNy, frames, etc.). When LP_BYPASS is FALSE(0), the L_Port is "enabled"; if LP_BYPASS is TRUE(1), the L_Port is "bypassed." An L_Port is bypassed when it recognizes LPByx (where y is the AL_PA of the L_Port), LPBfx, or at the request of the L_Port (REQ(bypass L_Port)). When an L_Port is bypassed, it shall set the Bypass Circuit (if present) and shall set LP_BYPASS to TRUE(1). If an L_Port recognizes LIP while LP_BYPASS is TRUE(1), the L_Port shall relinquish its AL_PA and shall enter the non-participating mode. An L_Port is enabled when it recognizes LPEyx (where y is the AL_PA of the L_Port), LPEfx, or at the request of the L_Port (REQ(enable L_Port)). When an L_Port is enabled, it shall reset the Bypass Circuit (if present) and shall set LP_BYPASS to FALSE(0). When an L_Port is enabled, the L_Port shall operate normally (i.e., it shall be in the non-participating or participating mode, depending on whether it has a valid AL_PA).

DHD received history

The OPENED states requires one memory element, 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 B.)

Time-outs

FC-PH time-out values
Time-out values and related time-out procedures in ANSI X3.230, FC-PH, 29.2, shall be used.

Arbitrated Loop time-out value

The Arbitrated Loop time-out (AL_TIME) is specified as 15 ms (-0%+20%), 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 the implemented data rate of the L_Port and 134 times 10 km, the cable latency (3,5 ns/meter for electrical; 5 ns/meter for optical).

NOTE It is conceivable that the maximum round-trip delay of a loop configuration is greater than two (2) times the minimum AL_TIME. However, determining interoperability when using a different AL_TIME values is outside the scope of this standard.

Operational characteristics

Modes of operation

An L_Port is in one of two operational modes:

Participating mode: An L_Port is in participating mode when it has acquired an AL_PA (e.g., through the initialization process. Since there is no enforceable limit to the number of L_Ports that may be physically connected to a Loop, a maximum of 127 L_Ports (1 FL_Port and 126 NL_Ports) shall be in participating mode on the same Loop at the same time.

NOTE Some laser safety requirements may further limit the number of L_Ports that may be connected in a Loop because of propagation delays.

An L_Port that is in participating mode may voluntarily relinquish control of its AL_PA and enter non-participating mode. This allows another L_Port to reuse that AL_PA.

Non-participating mode: An L_Port is in non-participating mode when it does not have a valid AL_PA. Reasons for not having an AL_PA are: the L_Port was unable to obtain an AL_PA; the L_Port voluntarily does not participate, or the L_Port has been bypassed and has recognized a LIP. Non-participating mode is the default operational mode for an L_Port. An L_Port in non-participating mode shall not arbitrate for access to and shall not respond to any ARBx, OPNy, or OPNr received on the Loop.

NOTE A non-participating L_Port does not have a valid AL_PA and can therefore not be bypassed (if a Bypass Circuit is present) by another L_Port. To prevent a non-participating L_Port from causing a Loop failure, it is recommended that a non-participating L_Port requests a bypass (REQ(bypass L_Port)).

Invalid Transmission Word processing

An L_Port is capable of re-transmitting valid Transmission Words and making substitutions for invalid Transmission Words when the L_Port is in the MONITORING or ARBITRATING state and an invalid Transmission Character or a misplaced Special Character is detected, the L_Port shall substitute any valid Transmission Character (see ANSI X3.230, FC-PH, clause 17) or if an invalid Beginning Running Disparity condition is detected on an Ordered Set, the L_Port shall substitute the current Fill Word. when the L_Port is in any other state (see ANSI X3.230, FC-PH, 24.3.5, and clause 29).

Clock skew management

For clock skew management (i.e., frequency offset), when processing Transmission Words between frames, any ARBx shall be treated the same as Idle. Fill Words or any Ordered Set defined for use as a Primitive Sequence shall be treated equally. If an L_Port is required to remove one of these Transmission Words, the L_Port shall transmit at least two of these Transmission Words before removing the next one of these Transmission Words.

NOTE Although two Fill Words are transmitted between R_RDYs, an L_Port is not guaranteed to receive two Fill Words between each R_RDY. (SeeANSI X3.230, FC-PH, 16.3.2.)

If it becomes necessary to add a Fill Word for clock skew management, the L_Port shall normally transmit the current Fill Word. However, if the last received Transmission Word is LR or LRR (which is used during certain ANSI X3.230, FC-PH, 16.4 Primitive Sequences), LR or LRR shall be transmitted, respectively. See each state in LPSM for the procedure to determine the current Fill Word.

NOTE Because of clock skew management, when LPSM and clause 9 mention that the current Fill Word is to be transmitted, it may not actually be transmitted (i.e., it may be absorbed by the clock skew management circuit).

Primitive Signal and Sequence substitution

An L_Port shall be capable of FC-AL Primitive Signal substitution: when an L_Port is in the MONITORING or ARBITRATING state, FC-AL Primitive Signal substitution shall be used as specified in each state in LPSM or when an L_Port is in any other state, FC-AL Primitive Signals are either processed or discarded by the L_Port as specified in each state in LPSM. LIP, LPE, and LPB processing is specified independently in each state in LPSM.

Error detection and recovery

Each state in LPSM 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 ANSI X3.230, FC-PH, 24.3.5 and clause 29.

BB_Credit and Available_BB_Credit

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 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 state, respectively. During Login, BB_Credit shall be set to a value that represents the number of receive buffers that an L_Port shall guarantee to have available when a communication path (i.e., a circuit) is established.

When on a Loop, L_Ports have unique characteristics (unlike point-to-point or Fabric-attached N_Ports): communication paths are dynamic; an L_Port may have frames in the receive buffers from the previous communication when a new communications path 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 circuit;and, balancing BB_Credit at the end of a circuit may impede performance.

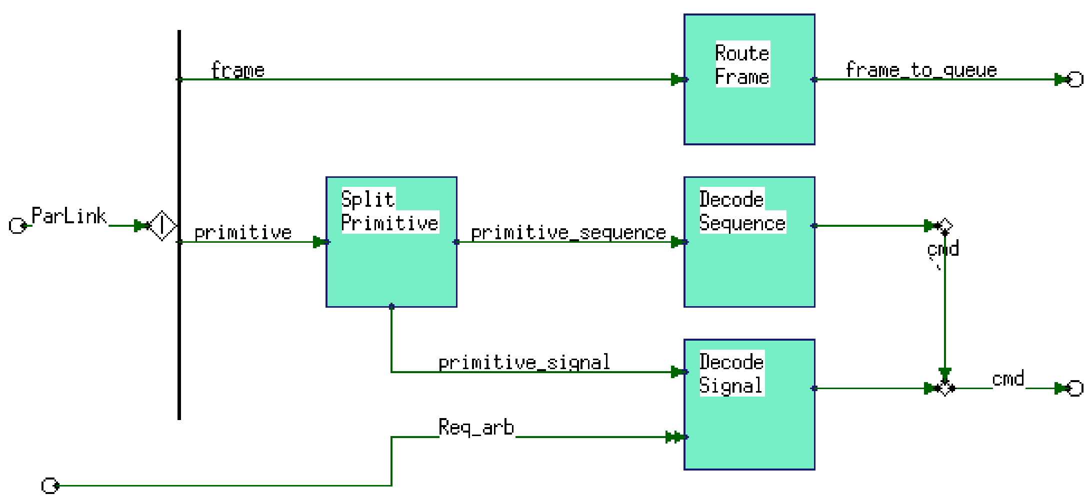

Figure 5. Input Buffer DFD

"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 circuit, L_Ports may take advantage of the BB_Credit Login value. Although balancing BB_Credit is not required (receive buffers may be emptied after the 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 circuit is established. Therefore, an L_Port shall not close a Loop unless the number of available receive buffers is at least equal to the largest BB_Credit Login value that the L_Port disseminated during Login. A positive BB_Credit allows the opening L_Port to follow OPNy with frames, without waiting for an R_RDY. Once the number of R_RDYs discarded equals the number of frames that have been transmitted, Available_BB_Credit is used to transmit subsequent frames. Annex F provides an example of how these credits may be used.

NOTE "Alternate BB_Credit Management" is written from the view of the L_Port that transmits the OPNy. The receiving L_Port may choose to identify the opening L_Port's BB_Credit, or immediately use Available_BB_Credit.

BB_Credit management per circuit

After transmitting OPNy, an L_Port may transmit one R_RDY for each free receive buffer before transmitting any frames. Since a minimum of six (6) Fill Words shall be transmitted between the OPNy and the first frame, if a receive buffer is available and the L_Port is ready to receive a frame from the L_Port that received the OPNy, at least one R_RDY shall be transmitted prior to the first frame. Subsequent R_RDYs may be transmitted to provide a balance between transmitting frames and transmittingR_RDYs.

After receiving OPNy, an L_Port shall transmit at least one R_RDY or the L_Port shall transmit CLS. CLS indicates that there are no free receive buffers or that the L_Port desires to close the circuit (i.e., has no frames to transmit).

The BB_Credit for each circuit is one of the following values: zero (0) (default value); the minimum value of the BB_Credit of all L_Ports that are currently logged-in; or, the specific Login value of the other L_Port.
NOTE If BB_Credit is zero (0), a Loop turn-around delay is required (i.e., an R_RDY must be received) before the opening L_Port is allowed to transmit the first frame.

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

Available_BB_Credit management per circuit

Once the L_Port has discarded the same number of R_RDYs as it has transmitted frames using the BB_Credit value, 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 an R_RDY is 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 is positive, the L_Port may transmit frames.

Loop Port State Machine (LPSM)

```
Inputs:   state;
Globals:  lpsm_globals,lpsm_state,cfw,port_name,next_word, available_bb_credit,
          dest_address, received_close;
Outputs:    Req_arb;

Initialize:
BEGIN
    lpsm_globals.lp_bypass := false;
    available_bb_credit := 0; lpsm_state := 8; cfw.pr_typ := idle;
    port_name := 1; received_close :=false;
    next_word.primitive.pr_typ := idle; dest_address := 1;
    lpsm_globals.participating := true;
    Req_arb := False;
END;
```

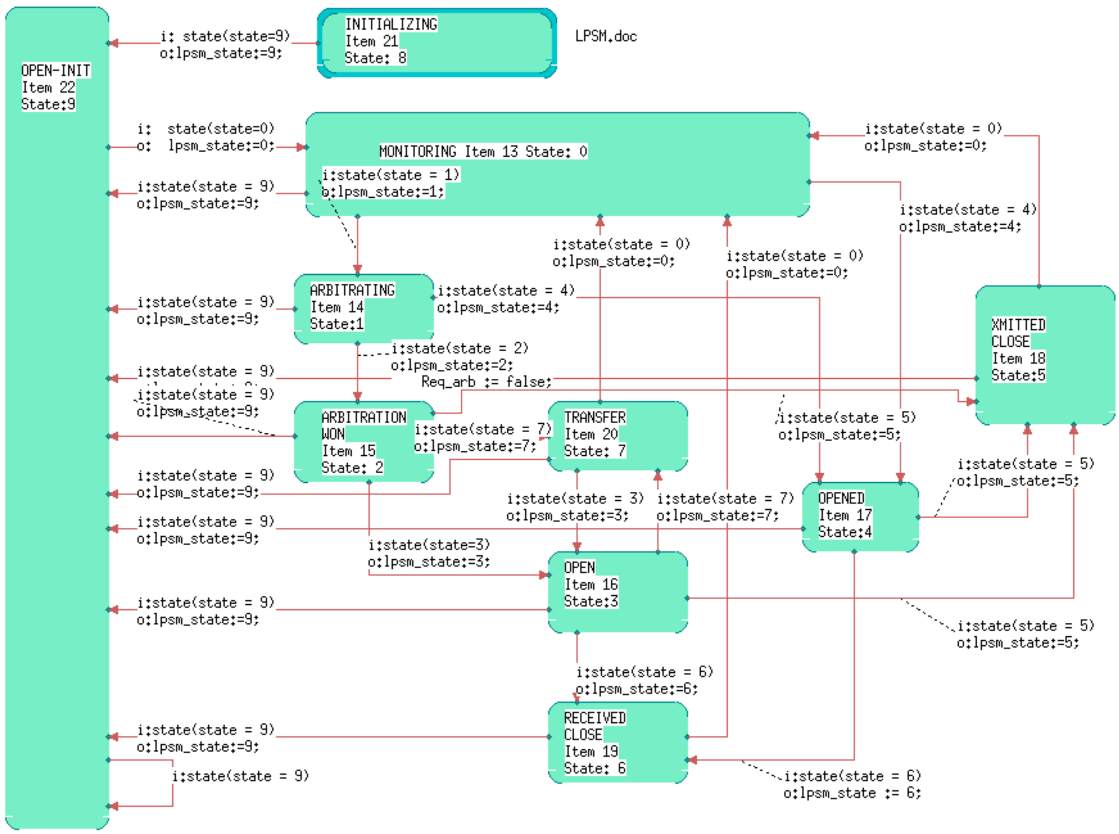

Figure 6. Loop Port State Machine STD

The Loop Port State Machine, State Transition Diagram, is shown in Figure 6. This is found in Figure 4 as the "LPSM" process block. In essence, it comprises the Loop Port State Machine functionality described below. A Loop Port State Machine (LPSM) is used to define the behavior of the L_Ports when they require access to and use of a Loop. The following sub clauses specify the state names, state diagram, and item references for the LPSM.

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.

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

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

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

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

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

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

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

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

INITIALIZING (8): The LPSM is initializing or re-initializing.

OPEN-INIT (9): The LPSM has recognized a LIP.

OLD-PORT (A): The LPSM has discovered that a non-L_Port is attached and the Arbitrated Loop protocol is not required.

State 0 (MONITORING) actions:

The LPSM shall set DUPLEX to FALSE(0), ARB_WON to FALSE(0), and REPLICATE to FALSE(0). If the L_Port is in non-participating mode, the LPSM shall re-transmit all received Transmission Words on the Loop. If the L_Port is in participating mode, the LPSM shall re-transmit all received Transmission Words unless specifically stated otherwise. If the Bpass Circuit (if present) is set, the LPSM shall respond only to LPEx (where = the AL_PA of the L_Port) or LPEfx. If ARB_PEND is TRUE(1) and LP_BYPASS is FALSE(0), the L_Port shall make the transition to the ARBITRATING state.
NOTE ARB_PEND is set to TRUE(1) when an L_Port has transmitted one or more ARBx Primitive Signals (where x = AL_PA of the L_Port). The L_Port ma have been opened b another L_Port while arbitrating. This flag forces the L_Port to finish arbitrating to assure that the fairness window is reset. If Idle is received, the current Fill Word shall be set to Idle and ACCESS shall be set to TRUE(1). To assure that a single Idle is not discarded for clock skew b another L_Port, at least two Idles shall be transmitted before changing the current Fill Word to another value.

If ARBx is received, the current Fill Word shall be modified as follows:
if x = hex 'F0' and the current Fill Word is Idle, the current Fill Word shall not be changed;
if x = hex 'F0' and the current Fill Word is not Idle, the current Fill Word shall be set to ARB(F0);
if x <> AL_PA of the L_Port, the current Fill Word shall be set to ARBx; or,
if x = AL_PA of the L_Port, the current Fill Word shall be changed to Idle.

If a Fill Word is to be transmitted, the current Fill Word shall be used.
If the L_Port is in participating mode with LP_BYPASS set to FALSE(0)
if REPLICATE is TRUE(1), the LPSM shall receive and re-transmit all Transmission Words;
if OPNfr is received, the LPSM shall set REPLICATE to TRUE(1) and re-transmit the received OPNfr;
if OPNr is received where = AL_PA of the L_Port, the LPSM shall set REPLICATE to TRUE(1) and re-transmit the received OPNr;

if OPN is received where = AL_PA of the L_Port, the LPSM shall make the transition to the OPENED state.
if an other OPN is received, it shall be re-transmitted;
if ARBx is received:
- if x = AL_PA of the L_Port, the LPSM shall transmit the current Fill Word; the ARBx is discarded;
- if x = hex 'F0', the LPSM shall transmit the current Fill Word; or,
- if x <> AL_PA of the L_Port, the LPSM shall re-transmit the ARBx.
if MRKtx is received:
- if x = AL_PA of the L_Port, the LPSM shall transmit the current Fill Word; the MRKtx is discarded;
- if the MK_TP and AL_PS match the expected values, synchronization shall be performed; or,
- if x <> AL_PA of the L_Port, the received MRKtx shall be re-transmitted.
if CLS is received while REPLICATE is TRUE(1), the LPSM shall set REPLICATE to FALSE(0) and re-transmit the received CLS; or, the LPSM shall make a transition to the ARBITRATING state when the L_Port requests arbitration (REQ(arbitrate as x)) and ACCESS is TRUE(1).
If LIP is received:
if LP_BYPASS is FALSE(0), the LPSM shall make the transition to the OPEN-INIT state. or,
if LP_BYPASS is TRUE(1), the L_Port shall relinquish its AL_PA (i.e., go to the non-participating mode) and remain in the MONITORING state.
If LPBx (= AL_PA of the L_Port) or LPBfx is recognized or the L_Port requests to be bypassed (REQ(bpass L_Port)), the LPSM shall set the Bpass Circuit (if present) and set LP_BYPASS to TRUE(1).
If LPEx (= AL_PA of the L_Port) or LPEfx is recognized or the L_Port requests to be enabled (REQ(enable L_Port)), the LPSM shall reset the Bpass Circuit (if present) and set LP_BYPASS to FALSE(0).

The LPSM shall re-transmit all other received Transmission Words on the Loop. Transmission Words shall be re-transmitted on the Loop.
If the LPSM detects a Loop failure on its inbound fibre or the L_Port requests initialization (REQ(initialize)), the LPSM shall make the transition to the INITIALIZING state.
If the L_Port requests not to participate on the Loop (REQ(non-participating.)), the LPSM shall transmit at least 12 LIPs (with the right-most two characters equal to hex 'F7F7') to invoke Loop Initialization. This allows another L_Port to acquire the relinquished AL_PA. The 12 LIPs are only transmitted once for each REQ(non-participating.) to allow this request to be active until the L_Port requests to participate (REQ(participating)). The L_Port shall not participate further in Loop Initialization until REQ(initialize) or REQ(participating) is set. If the L_Port requests to participate on the Loop (REQ(participating)), the LPSM shall make the transition to the INITIALIZING state.
If the L_Port requests to transmit a MRKtx (REQ(mark as tx)), the LPSM shall transmit one MRKtx at the next appropriate Fill Word, unless REQ(mark as tx) is removed before MRKtx is transmitted.

State 1 (ARBITRATING) actions:

The LPSM shall re-transmit all received Transmission Words unless specifically stated otherwise. The LPSM shall transmit an ARBx (where x equals the AL_PA of the L_Port) when either an Idle or a lower priority ARBx is received. Once the LPSM has transmitted its own ARBx, it shall set ARB_PEND to TRUE(1) and shall not transmit a lower-priority ARBx.If Idle is received, the current Fill Word shall be set to ARBx (where x equals the AL_PA of the L_Port). To assure that a single Idle is not discarded for clock skew b another L_Port, at least two Idles shall be transmitted before changing the current Fill Word to another value. If ARBx is received and x does not equal the AL_PA of the L_Port, the current Fill Word shall be modified as follows:
if x = hex 'F0' and AL_PA = hex 'F7' (i.e., the L_Port is attempting to initialize), the current Fill Word shall be changed to ARB(F7); if x > AL_PA, the current Fill Word shall be changed to ARBx (where x equals the AL_PA of the L_Port); or,
if x < AL_PA, the current Fill Word shall be changed to the received ARBx.

If a Fill Word is to be transmitted, the current Fill Word shall be used.
If ARBx is received and x equals the AL_PA of the L_Port, the LPSM shall make the transition to the ARBITRATIONWON state. (See items 5 and 15.If REPLICATE is TRUE(1), the LPSM shall receive all Transmission Words. If OPNfr is received, the LPSM shall set REPLICATE to TRUE(1) and re-transmit the received OPNfr.
If OPNr is received where = AL_PA of the L_Port, the LPSM shall set REPLICATE to TRUE(1) and re-transmit the received OPNr.
If OPN is received where = AL_PA of the L_Port, the LPSM shall make the transition to the OPENED state.
If another OPN or OPNr is received, it shall be re-transmitted.
If CLS is received and REPLICATE is TRUE(1), REPLICATE shall be set to FALSE(0). The CLS shall be re-transmitted.
If MRKtx is received:
if x = AL_PA of the L_Port, the LPSM shall transmit the current Fill Word; the MRKtx is discarded;
if the MK_TP and AL_PS match the expected values, synchronization shall be performed; or,
if x <> AL_PA of the L_Port, the received MRKtx shall be re-transmitted.
If LIP is recognized, the LPSM shall make the transition to the OPEN-INIT state.
If LPBx (= AL_PA of the L_Port) or LPBfx is recognized or the L_Port requests to be bypassed (REQ(bpass L_Port)), the LPSM shall set the Bpass Circuit (if present); set LP_BYPASS to TRUE(1); and, make the transition to the MONITORING state.
If the LPSM detects a Loop failure on its inbound fibre or the L_Port requests initialization (REQ(initialize)), the LPSM shall make the transition to the INITIALIZING state.
If the L_Port requests to transmit a MRKtx (REQ(mark as tx)), the LPSM shall transmit one MRKtx at the next appropriate Fill Word , unless REQ(mark as tx) is removed before MRKtx is transmitted.

State 2 (ARBITRATION WON) actions:

This is a transition state during which no Transmission Words are received. To identify this as the L_Port that won arbitration, the LPSM shall set ARB_WON to TRUE(1), ARB_PEND to FALSE(0), and the current Fill Word to ARB(F0).
If the LPSM used AL_PA = hex 'F7' to win arbitration, it shall make the transition to the INITIALIZING state.
If REPLICATE is TRUE(1), the L_Port shall transmit CLS and go to the TRANSFER state.
If the L_Port is using the fairness algorithm, ACCESS shall be set to FALSE(0);
if the L_Port is not using the fairness algorithm, ACCESS shall be set to TRUE(1). In this state, the L_Port shall make the decision to open the Loop or not to open the Loop:
if the L_Port still requires access to the Loop (REQ(open x), REQ(open), REQ(open fr), or REQ(open r)), the LPSM shall transmit OPN or the requested OPNr and shall make the transition to the OPEN state.
If OPNr is transmitted, REPLICATE shall be set to TRUE(1) or
if the L_Port no longer needs access to the Loop (REQ(close)), the LPSM shall transmit OPN (where = AL_PA of the L_Port) and shall make the transition to the OPEN state.
If the L_Port requests initialization (REQ(initialize)), the LPSM shall make the transition to the INITIALIZING state.

State 3 (OPEN) actions:

The LPSM shall transmit at least six (6) current Fill Words; interspersed among these shall be one R_RDY for each frame that the L_Port is willing to receive. The L_Port shall process, and shall not re-transmit subsequent Transmission Words received on its inbound fibre. The L_Port shall transmit Primitive Signals, Primitive Sequences, or frames as specified in ANSI X3.230, FC-PH.
If Idle is received, the current Fill Word shall be set to Idle and ACCESS shall be set to TRUE(1). To assure that a single Idle is not discarded for clock skew b another L_Port, at least two Idles shall be transmitted before changing the current Fill Word to another value.

If ARB(F0) is received, the current Fill Word shall be set to Idle. Receiving ARB(F0) indicates that no other L_Port is now arbitrating (i.e., no L_Port changed ARB(F0) to ARBx). If a Fill Word is to be transmitted, the current Fill Word shall be used.
If CLS is received, the LPSM shall make the transition to the RECEIVED CLOSE state. If MRKtx is received where the MK_TP and AL_PS match the expected values, synchronization shall be performed. Thereceived MRKtx shall not be re-transmitted. If REPLICATE is TRUE(1) and the L_Port requests a broadcast replicate (REQ(open fr) or another selective replicateREQ(open r)), the LPSM shall transmit OPN(fr) or one OPN(r) for each request at the next Fill Word, respectively.
If ACCESS is TRUE(1) and the L_Port requires communication with a different L_Port (REQ(transfer)), the LPSM shall transmit CLS instead of the next Fill Word and then shall make the transition to the TRANSFER state.
If ACCESS is FALSE(0), the request to transfer is ignored.
If a Class 1 connection exists, the L_Port shall remove the Class 1 connection before transmitting a CLS; only the L_Port which received EOFdt shall transmit CLS. The LPSM ma begin to close the Loop (REQ(close)) b transmitting either CLS or DHD instead of the next Fill Word.
If the login bit identifies that the L_Port in the OPENED state supports DHD, the LPSM may transmit DHD to indicate that it has finished transmitting Data frames.
IF DHD is transmitted, the LPSM shall remain in the OPEN state, however, it shall not transmit Data frames.
If CLS is transmitted, the LPSM shall make the transition to the XMITTED CLOSE state or the TRANSFER state.
If a Class 1 connection exists, the L_Port shall remove the Class 1 connection before transmitting a CLS; only the L_Port which received EOFdt shall transmit CLS.

NOTE Reasons for transmitting a CLS or DHD include, but are not limited to: ARBx was detected to indicate that another L_Port is arbitrating (the OPEN L_Port ma close the Loop at a convenient time); there are no additional Sequences to transmit to the other L_Port; or, the L_Port is making the transition to the non-participating mode.

If LIP is recognized, the LPSM shall make the transition to the OPEN-INIT state.
If LPB is recognized:
if x = AL_PA of the L_Port, the received LPBx shall be discarded or
if = AL_PA of the L_Port or xF, the LPSM shall end the current transmission; set the Bpass Circuit (if present); set LP_BYPASS to TRUE(1); and, make the transition to the MONITORING state.
If REPLICATE is TRUE(1), all received Transmission Words (except CLS, MRKtx, and an Primitive Sequence) shall be discarded.

NOTE This includes a frame(s) that traverses the Loop.

If the L_Port requests another L_Port to be either bypassed (REQ(bpass L_Port)) or enabled (REQ(enable L_Port) orREQ(enable all)), the LPSM shall begin to transmit LPB or LPE at the next Fill Word, until the Primitive Sequence is received
If the LPSM detects a Loop failure on its inbound fibre or the L_Port requests initialization (REQ(initialize)), the LPSM shall make the transition to the INITIALIZING state.
If the L_Port requests to transmit a MRKtx (REQ(mark as tx)), the LPSM shall transmit one MRKtx at the next appropriate Fill Word , unless REQ(mark as tx) is removed before MRKtx is transmitted.

State 4 (OPENED) actions:

The LPSM shall set ARB_WON to FALSE(0), REPLICATE shall be set to FALSE(0), and shall transmit the current Fill Word to replace the received OPN. The L_Port shall transmit at least six (6) current Fill Words (interspersed among these shall be one R_RDY for each frame that the L_Port is willing to receive) before transmitting a CLS. The L_Port shall process, and shall not re-transmit subsequent Transmission Words received on its inbound fibre. The L_Port shall transmit Primitive Signals, Primitive Sequences, or frames as

specified in ANSI X3.230, FC-PH. If OPNx was received, DUPLEX shall be set to TRUE(1); if OPN was received, DUPLEX shall be set toFALSE(0) and no Data frames shall be transmitted.
If this is a fair L_Port and ARB_PEND is TRUE(1), and
if the x value of the OPNx is equal to the AL_PA of the L_Port with which this NL_Port wished to communicate, and the LPSM was able to transmit frames to the other L_Port, then this NL_Port shall set ACCESS to FALSE(0) and ARB_PEND to FALSE(0); to indicate that it has met its arbitration requirement for the current access window.
If Idle is received and ARB_PEND is FALSE(0), the current Fill Word shall be set to Idle and ACCESS shall be set to TRUE(1). To assure that a single Idle is not discarded for clock skew b another L_Port, at least two Idles shall be transmitted before changing the current Fill Word to another value.
If Idle is received and ARB_PEND is TRUE(1), the current Fill Word shall be changed to ARBx (where x equals the AL_PA of the L_Port).
If ARBx is received and ARB_PEND is TRUE(1), the current Fill Word shall be modified as follows:
if x > AL_PA, the current Fill Word shall be changed to ARBx (where x equals the AL_PA of the L_Port);
if x = AL_PA, the current Fill Word shall be changed to ARB(F0); or,
if x < AL_PA, the current Fill Word shall be changed to the received ARBx.

If ARBx is received and ARB_PEND is FALSE(0), the current Fill Word shall be modified as follows:
if x = AL_PA, the current Fill Word shall be changed to ARB(F0) or
if x <> AL_PA, the current Fill Word shall be changed to the received ARBx.

If a Fill Word is to be transmitted, the current Fill Word shall be used. If OPNr or OPN are received, the shall be discarded. If DHD is received, the LPSM shall set DHD_RCV to TRUE(1). Receiving DHD is an indication to this L_Port that the LPSM in the OPEN state has no more Data frames to transmit. The L_Port shall respond to DHD b transmitting frames or CLS.

If DHD_RCV is TRUE(1) and the L_Port has completed all transfers (or it had nothing to transmit when it received DHD) to the L_Port in the OPEN state, it shall REQ(close) to begin closing the Loop.
If CLS is received, the LPSM shall make the transition to the RECEIVED CLOSE state.

If MRKtx is received where the MK_TP and AL_PS match the expected values, synchronization shall be performed. The received MRKtx shall not be re-transmitted. The LPSM ma begin to close the Loop (REQ(close)) b transmitting CLS instead of the next Fill Word and then shall make the transition to the XMITTED CLOSE state. If a Class 1 connection exists, the L_Port shall remove the Class 1 connection before transmitting a CLS; only the L_Port which received EOFdt shall transmit CLS.

NOTE Reasons for transmitting CLS include, but are not limited to:
ARBx has been detected to indicate that another L_Port is arbitrating (the OPENED L_Port ma close the Loop at a convenient time);frame transmission is required with a different L_Port; there are no more Sequences to process with the other L_Port in this circuit; or,the L_Port is making the transition to the non-participating mode.

If LIP is recognized, the LPSM shall make the transition to the OPEN-INIT state.
If LPBx (= AL_PA of the L_Port) or LPBfx is recognized or the L_Port requests to be bypassed (REQ(bpass L_Port)), the LPSM shall end the current transmission; set the Bpass Circuit (if present); set LP_BYPASS to TRUE(1); and, make the transition to the MONITORING state. If the LPSM detects a Loop failure on its inbound fibre or the L_Port requests initialization (REQ(initialize)), the LPSM shall make the transition to the INITIALIZING state. If the L_Port requests to transmit a MRKtx (REQ(mark as tx)), the LPSM shall transmit one MRKtx at the next appropriate Fill Word , unless REQ(mark as tx) is removed before MRKtx is transmitted.

State 5 (XMITTED CLOSE) actions:

The L_Port shall continue to operate on the Loop. The LPSM shall transmit only the current Fill Word (except MRKtx). The L_Port shall process, but shall not re-transmit subsequent Transmission Words received on its inbound fibre (except MRKtx).If Idle is received and ARB_PEND is FALSE(0), the current Fill Word shall be set to Idle and ACCESS shall be set to TRUE(1). To assure that a single Idle is not discarded for clock skew b another L_Port, at least two Idles shall be transmitted before changing the current Fill Word to another value. If Idle is received and ARB_PEND is TRUE(1), the current Fill Word shall be changed to ARBx (where x equals the AL_PA of the L_Port).
If ARB(F0) is received and ARB_WON is TRUE(1), the current Fill Word shall be set to Idle. Receiving ARB(F0) indicates that no other L_Port is now arbitrating (i.e., no L_Port changed ARB(F0) to ARBx).
If ARB(F0) is received and ARB_WON is FALSE(0), the current Fill Word shall be modified as follows:
if ARB_PEND is FALSE(0), the current Fill Word shall be changed to ARB(F0) or
if ARB_PEND is TRUE(1), the current Fill Word shall be changed to ARBx (where x equals the AL_PA of the L_Port).

If ARBx is received, ARB_WON is FALSE(0), and ARB_PEND is FALSE(0), the current Fill Word shall be modified as follows:
if x = AL_PA of the L_Port, the current Fill Word shall be changed to ARB(F0)or
if x <> AL_PA of the L_Port, the current Fill Word shall be set to ARBx.

If ARBx is received, ARB_WON is FALSE(0), and ARB_PEND is TRUE(1), the current Fill Word shall be modified as follows:
if x > AL_PA, the current Fill Word shall be changed to ARBx (where x equals the AL_PA of the L_Port);
if x = AL_PA, the current Fill Word shall be changed to ARBx (where x equals the AL_PA of the L_Port); or,
if x < AL_PA, the current Fill Word shall be changed to the received ARBx.

If a Fill Word is to be transmitted, the current Fill Word shall be used. If CLS is received, the LPSM shall transmit the current Fill Word and shall make the transition to the MONITORING state.

NOTE If the L_Port had advertised a Login BB_Credit > 0, in order to avoid an overruns, it is advisable that the number of available buffers at least equal the Login BB_Credit before making the transition to the MONITORING state.
If MRKtx is received:
if x = AL_PA of the L_Port, the LPSM shall transmit the current Fill Word; the MRKtx is discarded;
if the MK_TP and AL_PS match the expected values, synchronization shall be performed; or,
if x <> AL_PA of the L_Port, the received MRKtx shall be re-transmitted.
If LIP is recognized, the LPSM shall make the transition to the OPEN-INIT state.
If LPBx (= AL_PA of the L_Port) or LPBfx is recognized or the L_Port requests to be bypassed (REQ(bpass L_Port)), the LPSM shall set the Bpass Circuit (if present); set LP_BYPASS to TRUE(1); and, make the transition to the MONITORING state. If an other LPBx is received, it shall be replaced with the current Fill Word.

If the LPSM detects a Loop failure on its inbound fibre or the L_Port requests initialization (REQ(initialize)), the LPSM shall make the transition to the INITIALIZING state. If the L_Port requests to transmit a MRKtx (REQ(mark as tx)), the LPSM shall transmit one MRKtx at the next appropriate Fill Word unless REQ(mark as tx) is removed before MRKtx is transmitted.

State 6 (RECEIVED CLOSE) actions:

The LPSM shall set REPLICATE to FALSE(0). The L_Port may continue to transmit frames until Available_BB_Credit or EE_Credit is exhausted. An frame or R_RDY received from the other L_Portshall be discarded. The LPSM shall process, and shall not re-transmit subsequent Transmission Words received on its inbound fibre. The L_Port shall transmit Primitive Signals, Primitive Sequences, or frames as specified in

ANSI X3.230, FC-PH. When the LPSM transmits CLS (REQ(close)), the LPSM shall make the transition to the MONITORING state.

NOTE If the L_Port had advertised a Login BB_Credit > 0, in order to avoid an overruns, it is advisable that the number of available buffers at least equal the Login BB_Credit before making the transition to the MONITORING state. If Idle is received and ARB_PEND is FALSE(0), the current Fill Word shall be set to Idle and ACCESS shall be set to TRUE(1). To assure that a single Idle is not discarded for clock skew b another L_Port, at least two Idles shall be transmitted before changing the current Fill Word to another value. If Idle is received and ARB_PEND is TRUE(1), the current Fill Word shall be changed to ARBx (where x equals the AL_PA of the L_Port). If ARB(F0) is received and ARB_WON is TRUE(1), the current Fill Word shall be set to Idle. Receiving ARB(F0) indicates that no other L_Port is now arbitrating (i.e., no L_Port changed ARB(F0) to ARBx). If ARB(F0) is received and ARB_WON is FALSE(0), the current Fill Word shall be modified as follows:
if ARB_PEND is FALSE(0), the current Fill Word shall be changed to ARB(F0) or
if ARB_PEND is TRUE(1), the current Fill Word shall be changed to ARBx (where x equals the AL_PA of the L_Port).

If ARBx is received, ARB_WON is FALSE(0), and ARB_PEND is FALSE(0), the current Fill Word shall be modified as follows:
if x = AL_PA of the L_Port, the current Fill Word shall be changed to ARB(F0) or
if x <> AL_PA of the L_Port, the current Fill Word shall be set to ARBx.

If ARBx is received, ARB_WON is FALSE(0), and ARB_PEND is TRUE(1), the current Fill Word shall be modified as follows:
if x > AL_PA, the current Fill Word shall be changed to ARBx (where x equals the AL_PA of the L_Port);
if x = AL_PA, the current Fill Word shall be changed to ARBx (where x equals the AL_PA of the L_Port); or,
if x < AL_PA, the current Fill Word shall be changed to the received ARBx.

If a Fill Word is to be transmitted, the current Fill Word shall be used.
If MRKtx is received where the MK_TP and AL_PS match the expected values, synchronization shall be performed. The received MRKtx shall not be re-transmitted. If LIP is recognized, the LPSM shall make the transition to the OPEN-INIT state.

If LPBx (= AL_PA of the L_Port) or LPBfx is recognized or the L_Port requests to be bypassed (REQ(bpass L_Port)), the LPSM shall end the current transmission (e.g., frame); shall set the Bpass Circuit (if present); set LP_BYPASS to TRUE(1); and, make the transition to the MONITORING state.
If the LPSM detects a Loop failure on its inbound fibre or the L_Port requests initialization (REQ(initialize)), the LPSM shall make the transition to the INITIALIZING state. If the L_Port requests to transmit a MRKtx (REQ(mark as tx)), the LPSM shall transmit one MRKtx at the next appropriate Fill Word , unless REQ(mark as tx) is removed before MRKtx is transmitted.

State 7 (TRANSFER) actions:

The LPSM shall set REPLICATE to FALSE(0). The L_Port shall continue to operate on the Loop. The LPSM shall transmit only the current Fill Word (except MRKtx). The L_Port shall process, but shall not re-transmit subsequent Transmission Words received on its inbound fibre (except MRKtx).If Idle is received, the current Fill Word shall be set to Idle and ACCESS shall be set to TRUE(1). To assure that a single Idle is not discarded for clock skew b another L_Port, at least two Idles shall be transmitted before changing the current Fill Word to another value. If ARB(F0) is received, the current Fill Word shall be set to Idle. Receiving ARB(F0) indicates that no other L_Port is now arbitrating (i.e., no L_Port changed ARB(F0) to ARBx).
If a Fill Word is to be transmitted, the current Fill Word shall be used.

If CLS is received:
if the L_Port still requires access to the Loop (REQ(open x) or REQ(open), the LPSM shall transmit OPN to replace the received CLS and shall make the transition to the OPEN state. (See item 11 and 16);
if the L_Port still requires access to the Loop (REQ(open fr) or REQ(open r)), the LPSM shall transmit OPNr to replace the received CLS, shall set REPLICATE to TRUE(1), and shall make the transition to the OPEN state. or,
if the L_Port no longer needs access to the Loop (REQ(monitor)), the LPSM shall make the transition to the MONITORING state.
If MRKtx is received:
if x = AL_PA of the L_Port, the LPSM shall transmit the current Fill Word; the MRKtx is discarded;
if the MK_TP and AL_PS match the expected values, synchronization shall be performed; or,
if x <> AL_PA of the L_Port, the received MRKtx shall be re-transmitted.
If LIP is recognized, the LPSM shall make the transition to the OPEN-INIT state.
If LPBx (= AL_PA of the L_Port) or LPBfx is recognized or the L_Port requests to be bypassed (REQ(bpass L_Port)), the LPSM shall set the Bpass Circuit (if present); set LP_BYPASS to TRUE(1); and, make the transition to the MONITORING state. If an other LPBx is received, it shall be replaced b the current Fill Word. If the LPSM detects a Loop failure on its inbound fibre or the L_Port requests initialization (REQ(initialize)), the LPSM shall make the transition to the INITIALIZING state.
If the L_Port requests to transmit a MRKtx (REQ(mark as tx)), the LPSM shall transmit one MRKtx at the next appropriate Fill Word , unless REQ(mark as tx) is removed before MRKtx is transmitted.

State 8 (INITIALIZING) actions:

The LPSM shall set LP_BYPASS to FALSE(0); shall transmit twelve (12) LIPs; and shall not re-transmit received Transmission Words except LPB and LPE. The L_Port shall continue to transmit LIP for up to two (2) maximum AL_TIMEs and monitor only for LIP, LPB, and LPE. During normal initialization the L_Port shall react as follows:

if an LIP is recognized, the LPSM shall make the transition to the OPEN-INIT state.

if LPBx (= AL_PA of the L_Port) or LPBfx is recognized, the LPSM shall set the Bpass Circuit (if present); set LP_BYPASS to TRUE(1); and, make the transition to the MONITORING state. An other LPBx shall be re-transmitted;

if LPEx or LPEfx is recognized, the received LPE shall be re-transmitted; or,
if LIP, LPBx, or LPBfx has not been recognized within two (2) maximum AL_TIMEs, either there is a non-L_Port on the Loop or a Loop failure has been detected.
If a non-L_Port is suspected, the L_Port shall go to the OLD-PORT state.
If a Loop failure is suspected, the L_Port shall attempt to recover the Loop b continuing in the INITIALIZING state and ma use the procedure outlined in annex I.2.2 to bpass a failing L_Port.

During Loop recover the L_Port shall react as follows:
if LIP(,x) is recognized and: = hex 'F8', then the LPSM shall set the received LIP to LIP(F7,x) and shall make the transition to the OPEN-INIT state.

if LIP(,x) <> hex F8', the LPSM shall make the transition to the OPEN-INIT state.
if LPBx is recognized, where x = AL_PA of the L_Port, the LBPx shall be replaced b LIP. Since the transmitted LPBx was received, presumably, the Loop is operational. The L_Port shall reenter the INITIALIZING state.
if LPBx (= AL_PA of the L_Port) or LPBfx is recognized, the LPSM shall set the Bpass Circuit (if present); set LP_BYPASS to TRUE(1); and, make the transition to the MONITORING state.
if an other LPBx is received, it shall be re-transmitted;

if LPEx or LPEfx is recognized, where x = AL_PA of the L_Port, the LPE shall be replaced b LIP. Since the transmitted LPE was received, presumably, the Loop is operational. The L_Port shall reenter the INITIALIZING state;
if an other LPE is received, it shall be re-transmitted; or,
if the L_Port is unsuccessful in its attempt to recover the Loop (i.e., LIP or the transmitted LPB was not received during two (2) AL_TIMEs of each bpass attempt), the L_Port shall go to the OLD-PORT state.

If at an time in the INITIALIZING state the L_Port requests to be bypassed (REQ(bpass L_Port)), the LPSM shall set the Bpass Circuit (if present); set LP_BYPASS to TRUE(1); and, make the transition to the MONITORING state.

State 9 (OPEN-INIT) actions:

The L_Port shall set ACCESS to TRUE(1), shall set the current Fill Word to Idle, shall set the "Alternate BB_Credit Management" bit to 1, shall set BB_Credit to zero (0), shall transmit at least twelve (12) LIPs of the same type as the last LIP received, and shall continue with the initialization procedure b transmitting the LISM sequence as described in 10.4. During the LISM sequence transmission, the L_Port shall ignore all other Transmission Words. During initialization only, two types of L_Ports are identified in the OPEN-INIT state:
(1) Loop master: This L_Port shall transmit and receive the Loop Initialization Sequences identified in 10.4.
(2) Non-Loop master: This L_Port shall receive and re-transmit the Loop Initialization Sequences as identified in 10.4. The Loop master selection is determined as follows:
(1) if a LISM sequence is received where the D_ID and Port_Name is equal to the D_ID and Port_Name of the L_Port, this L_Port becomes the Loop master.
(2) if ARB(F0) is received, another L_Port has become the Loop master.
(3) if a 10 second timer has elapsed before either condition (1) or (2) are met, the L_Port shall restart the initialization process b going to the INITIALIZATION state.
if CLS is recognized, initialization has completed. Before making the transition to the MONITORING state, the Loop master shall discard the CLS; all other L_Ports shall re-transmit the CLS;
if LIP is recognized, the LPSM shall reenter the OPEN-INIT state (i.e., transmit at least twelve (12) LIPS of the same type as the last LIP received, etc.);
if LPB is recognized:
- if x = AL_PA of the L_Port, the received LPB shall be discarded;
- if = AL_PA of the L_Port or = xF, the LPSM shall set the Bpass Circuit (if present); set LP_BYPASS to TRUE(1); and, make the transition to the MONITORING state; or,
- if <> AL_PA of the L_Port, the received LPBx shall be re-transmitted.
if LPE is recognized:
- if x = AL_PA of the L_Port, the received LPE shall be discarded or
- if x <> AL_PA of the L_Port, the received LPE shall be re-transmitted.
If the L_Port requests to be bypassed (REQ(bpass L_Port)), the LPSM shall set the Bpass Circuit (if present); set LP_BYPASS to TRUE(1); and, make the transition to the MONITORING state.
If the L_Port requests another L_Port to be either bypassed or enabled (REQ(bpass L_Port) or REQ(enable L_Port) or REQ(enable all)), the LPSM shall transmit LPB or LPE until the Primitive Sequence is received.
If the L_Port no longer requires access to the Loop (REQ(non-participant.)), the LPSM shall make the transition to the MONITORING state. If the LPSM detects a Loop failure on its inbound fibre or the L_Port requests initialization (REQ(initialize)), the LPSM shall make the transition to the INITIALIZING state.

State A (OLD-PORT) actions:

The LPSM shall set the current Fill Word to Idle; the L_Port shall process, but the LPSM shall not re-transmit Transmission Words received on its inbound fibre. The L_Port shall transmit Primitive Signals, Primitive Sequences, or frames as specified in ANSI X3.230, FC-PH. Before Login 1 , the "Alternate BB_Credit Management" bit shall be set to 0 and the BB_Credit shall be set to one (1). The LPSM shall monitor for LIP. If LIP is recognized, the L_Port shall implicitly Logout with the other non-L_Port and shall make the transition to the OPEN-INIT state. If the LPSM detects a Loop failure on its inbound fibre or the L_Port requests initialization (REQ(initialize)), the LPSM shall make the transition to the INITIALIZING state.

FC-AL Primitive Signals and Sequences

The re-useable element of Figure 4, labeled L_Port_Controls, is shown below as Figure 7.

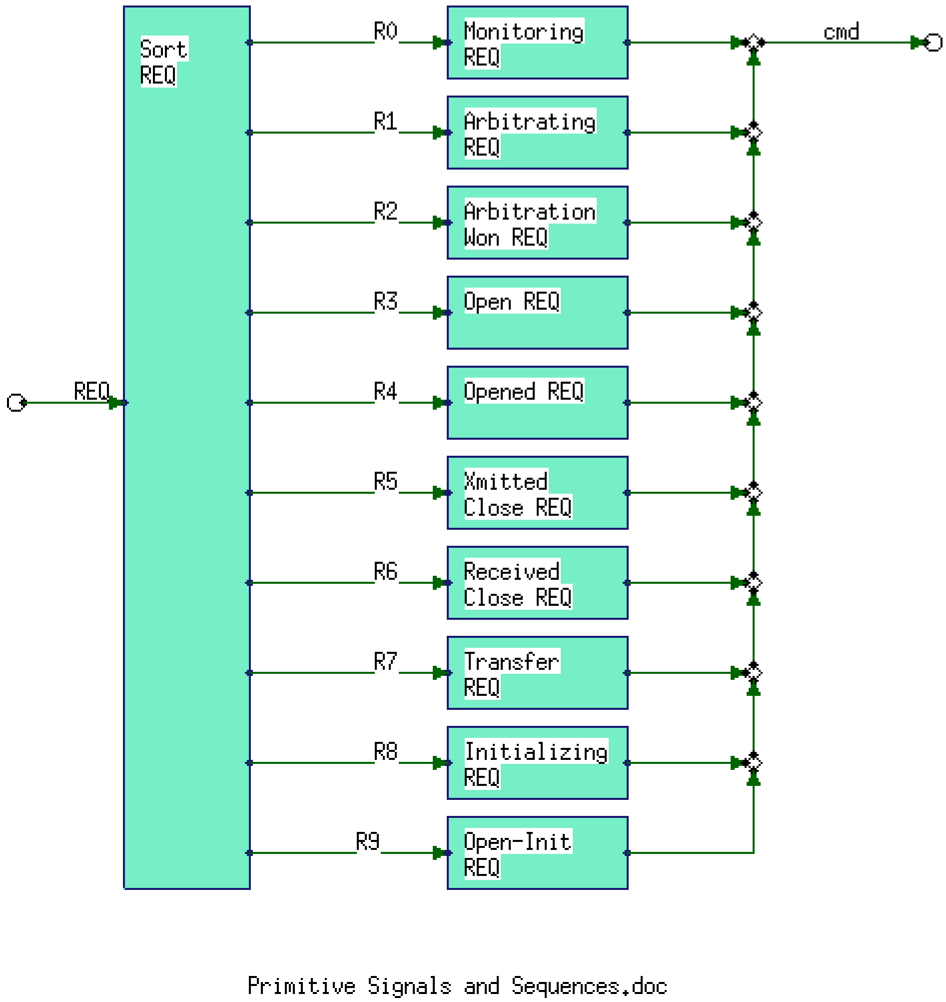

Figure 7. L_Port_Controls DFD

The Arbitrate and Mark Primitive Signals may be transmitted in place of an Idle and therefore, may be removed for clock skew management. All other Primitive Signals 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 ANSI X3.230, FC-PH, 16.3.2 and clause 6 for a specification of the following Ordered Sets.)

Arbitrate Primitive Signals (ARB)

Arbitrate (ARBx)

Arbitrate (ARBx) is transmitted on a Loop by a participating L_Port to request access to the Loop. Each ARBx shall contain the AL_PA (x value) of the L_Port making the request.

Arbitrate (ARB(F0))

Arbitrate (ARB(F0)) is transmitted on a Loop to manage access fairness. Since this is a low-priority ARB, any arbitrating L_Port may replace the ARB(F0) with its ARBx. ARB(F0) is also used while selecting a temporary Loop master during Loop initialization.

Arbitrate (ARB(F7))

Arbitrate (ARB(F7) is transmitted on a Loop by an L_Port which does not have a valid AL_PA. This arbitration character is used to quiesce the Loop prior to transmitting a LIP when initializing to obtain a valid AL_PA. Since this is a low-priority ARB, any arbitrating L_Port may replace the ARB(F7) with its ARBx, thereby forcing the initializing L_Port to the end of the fairness window. The L_Port shall treat the ARB(F7) as a higher priority arbitration character than ARB(F0) (i.e., any ARB(F0) shall be replaced by ARB(F7)).

Open Primitive Signals (OPNy)

An originating L_Port determines 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 or the left-most two bytes of the D_ID are hex '0000', 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.

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 ANSI X3.230, FC-PH, 4.6, model 2.) 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 an L_Port in the correct state indicates that another participating L_Port desires to communicate in full-duplex mode with the L_Port that received OPNyx.

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 ANSI X3.230, FC-PH, 4.6, model1.) The OPNyy shall contain the AL_PD (destination) y value) of the L_Port to be opened. OPNyy that is received by an L_Port in the correct state indicates that another participating L_Port desires to communicate in half-duplex mode with the L_Port that received OPNyy. The opened L_Port shall not transmit Data frames.

Open Replicate Primitive Signals (OPNr)

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 be ignored. 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 an L_Port is in the MONITORING or ARBITRATING state and recognizes 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 frame shall be re-transmitted to the next L_Port on the Loop. NL_Ports shall provide all frames to FC-2 for further processing, however, the FL_Port 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).

If an L_Port wins arbitration while REPLICATE is TRUE(1) (e.g., the L_Port which originated the OPNr was removed from the Loop before transmitting CLS), the L_Port in the Arbitration Won state shall transmit CLS (i.e. causes all L_Ports to set REPLICATE to FALSE(0)) and shall go to the TRANSFER state.

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.

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.

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

Close Primitive Signal (CLS)

Close (CLS) is transmitted on a Loop by a participating L_Port. Once an L_Port has transmitted CLS, the L_Port shall not transmit frames or R_RDYs in the current circuit. CLS indicates that the transmitting L_Port is prepared to or has relinquished control of the Loop for the current circuit.

Dynamic Half-Duplex Primitive Signal (DHD)

Dynamic Half-Duplex (DHD) is transmitted on a Loop by the L_Port in the OPEN state (in lieu of CLS) to indicate to the L_Port in the OPENED state that it has no more Data frames to transmit. DHD shall only be transmitted if the L_Port in the OPENED state has indicated via Login, support of the DHD feature. The DHD supported login bit is found in FC-PH-3 (see ANSI X3.303- 199x, FC-PH-3, 23.6.2.3). DHD allows L_Ports to make more efficient use of the established circuit by:

1. allowing an L_Port which is only capable of half-duplex data transfers, to transfer Data frames in the opposite direction without re-arbitrating.

2. 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.

Transmitting DHD only affects Data frames (i.e., Link_Control frames and R_RDYs may still be transmitted) just as in the definition of an 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, barring unusual circumstances, an L_Port in the OPEN state would normally not transmit CLS, if it has transmitted DHD.

Mark Primitive Signal (MRKtx)

Mark (MRKtx) is transmitted on a Loop by a master control point to synchronize other Nodes. 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. Since MRKtx shall only replace a Fill Word, it is possible that the mark window is exceeded (i.e., REQ(mark as tx) is withdrawn) before the MRKtx can be transmitted (i.e., no MRKtx is transmitted). The type of synchronization (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), the MRKtx shall be replaced with the current Fill Word. All other L_Ports which are in the MONITORING, ARBITRATING, XMITTED CLOSE, or TRANSFER state shall re-transmit the received MRKtx.

NOTE Since not all states re-transmit MRKtx, in order to guarantee that all L_Ports receive MRKtx, the originator has to 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).

Loop Port Bypass/Enable Primitive Sequences

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 the Bypass Circuit (if present). The Bypass Circuit may be used to physically bypass an L_Port, however, the L_Port is also logically bypassed (i.e., the L_Port cannot originate Transmission Words on the Loop).

Loop Port Bypass (LPByx)

Loop Port Bypass (LPByx) is transmitted on a Loop to set the Bypass Circuit (if present) and to bypass an L_Port. 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. When LPByx is recognized and the Bypass Circuit (if present) has been set, the L_Port shall not originate Transmission words (except for clock skew). The L_Port shall only monitor the Loop (as in non-participating mode), but shall keep its AL_PA untilit recognizes LIP. When LIP is received, the L_Port shall assume that its AL_PA is being used by another L_Port and it shall enter the non-participating mode. LPByx is primarily used to diagnose the Bypass Circuit and for error recovery. Each L_Port in the MONITORING, ARBITRATING, INITIALIZING, or OPEN-INIT state shall re-transmit the received LPByx. When LPByx is received by the originator (i.e., x = AL_PA), the LPByx shall be replaced with the current Fill Word. Although LPByx may be transmitted in a number of states, not all states re-transmit LPByx. To guarantee that the designated L_Port (as identified by the y value) receives LPByx, 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 OPEN-INIT state.Once an L_Port is bypassed and the Bypass Circuit (if present) has been set, the L_Port shall only monitor the Loop for a LPEyx (where y = AL_PA) or LPEfx and LIP. LIP is only used as a signal to relinquish its AL_PA; the L_Port shall not go to the OPEN-INIT state.

Loop Port Bypass all (LPBfx)

Loop Port Bypass all (LPBfx where f = hex 'FF') is transmitted on a Loop to set all Bypass Circuit(s) (if present) except for the L_Port at x. The originator of the LPBfx is identified by the AL_PS in character 4 (x value). The main purposed of this primitive is to verify that an operating Loop is possible, however, it is also useful to bypass a non-participating L_Port (i.e., the L_Port does not have an AL_PA).

When LPBfx is recognized, all L_Ports on the Loop (participating or non-participating), shall set the Bypass Circuit (if present).

Each L_Port in the MONITORING, ARBITRATING, INITIALIZING, or OPEN-INIT state shall re-transmit the received LPBfx. When LPBfx is received by the originator (i.e., x = AL_PA), the LPBfx shall be replaced with the current Fill Word.

Although LPBfx may be transmitted at any time, not all states re-transmit LPBfx. To guarantee that all L_Ports receive LPBfx, 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 OPEN-INIT state.

Loop Port Enable (LPEyx)

Loop Port Enable (LPEyx) is transmitted on a Loop to reset the Bypass Circuit (if present) and to enable an L_Port that had been previously bypassed 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). When LPEyx is recognized, the previously bypassed L_Port may participate on the Loop (e.g., originate frames). LPEyx is primarily used to detect if a Bypass Circuit is present and operational and for error recovery. Each L_Port in the MONITORING, ARBITRATING, INITIALIZING, or OPEN-INIT state shall re-transmit the received LPEyx. When LPEyx is received by the originator (i.e., x = AL_PA), the LPEyx shall be replaced with the current Fill Word. Although LPEyx may be transmitted at any time, not all states re-transmit LPEyx. To guarantee that the designated L_Port (as identified by the y value) receives LPEyx, the originator shall be in the OPEN state and all other L_Ports shall be in theMONITORING or ARBITRATING state or all L_Ports shall be in the OPEN-INIT state.

Loop Port Enable all (LPEfx)

Loop Port Enable all (LPEfx where f = hex 'FF') is transmitted on a Loop to reset all Bypass Circuit(s) (if present) that may have been previously set and to enable all L_Ports to participate on the Loop (e.g., originate frames). 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 a LIP). Therefore, LPEfx is very useful to allow these L_Ports(which no longer have an AL_PA) to be enabled on the Loop. When LPEfx is recognized, a previously bypassed participating L_Port may participate on the Loop; a previously non-participating L_Port may perform Loop Initialization. LPEfx is primarily used to detect if a Bypass Circuit is present and operational and for error recovery. Each L_Port in the MONITORING, ARBITRATING, INITIALIZING, or OPEN-INIT state shall re-transmit the received LPEfx. When LPEfx is received by the originator (i.e., x = AL_PA), the LPEfx shall be replaced with the current Fill Word. Although LPEfx may be transmitted at any time, not all states re-transmit LPEfx. To guarantee that all L_Ports receive LPEfx, the originator shall be in the OPEN state and all other L_Ports shall be in the MONITORING or ARBITRATING state or allL_Ports shall be in the OPEN-INIT state.

Loop Initialization Primitive Sequences (LIP)

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. The LIP carries with it information on why the LIP was transmitted in the right-most two characters. Other L_Ports may make decisions based on this information (e.g., inform an operator of a Loop failure).

Loop Initialization no valid AL_PA

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

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. The L_Port has not completed initialization or is bypassed, therefore, the hex 'F7' is used instead of a valid AL_PA.

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.

Loop Initialization Loop failure; valid AL_PA

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.

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 may also perform a vendor specific reset. If AL_PD = hex 'FF', a vendor specific reset shall be performed by all L_Ports (except the one at AL_PS). All L_Ports (including those which do not have an AL_PA), shall treat this as an L_Port reset.

FC-1 layer

FC-1 defines the transmission protocol including serial encoding and decoding rules, special characters and error control. The information transmitted over a fibre is encoded 8 bits at a time into a 10 bit Transmission Character. The primary rationale for use of a transmission code is to improve the transmission characteristic of information across a fibre. The transmission code must be DC balanced to support the electrical requirements of the receiving units. The Transmission Characters ensure, that short run lengths and enough transitions are present in the serial bit stream to make clock recovery possible.

FC-1 character conversion

An unencoded information byte is composed of eight information bits A,B,C,D,E,F,G,H and the control variable Z. This information is encoded by FC-1 into the bits a,b,c,d,e,i,f,g,h,j of a 10-bit Transmission Character. The control variable has either the value D (D-type) for Data characters or the value K (K-type) for special characters. Each valid Transmission Character has been given a name using the following convention: Zxx.y, where Z is the control variable of the unencoded FC-1 information byte, xx is the decimal value of the binary number composed of the bits E, D, C, B, and A, and y is the decimal value of the binary number composed of the bits H,G of the unencoded FC-1 information byte in that order. For example the name of the FC-1 Transmission Character composed of the hexadecimal "BC" special (K-type) code is K28.5.

The information received is recovered 10 bits at a time and those Transmission Characters used for data (D-type) are decoded into the one of the 256 8-bit combinations. Some of the remaining Transmission Characters (K-type) referred to as special characters, are used for protocol management functions. Codes detected at the receiver that are not D- or K- type are signaled as code violation errors.

Coding rules

Each data byte or special character has two (not necessarily different) transmission codes. The data bytes and special characters are encoded into these codes respectively, depending on the initial Running Disparity (RD). The RD is a binary parameter, which is calculated upon the balance of ones and zeros in the sub-blocks (the first six bits and the last four bits) of a transmission character. A new RD is calculated from the transmitted character at both the transmitter and the receiver. If the detected character has opposite RD which the transmitter should have sent, (depending on the RD of the previous bit stream) the receiver indicates a disparity violation condition. A Transmission Word is composed of four contiguous transmission characters.

FC-0 layer

FC is structured as a set of hierarchical functions.The lowest level (FC-0) defines the physical link in the system, including the fibre, connectors, optical and electrical parameters for a variety of data rates.

The system bit error rate (BER) at the supported media and speeds is less than 10exp-12. The physical level is designed for the use of large number of technologies to meet the widest range of system requirements. An end-to-end communicating route may consist of different link technologies to achieve the maximal performance and price efficiency.

Open Fibre Control

The FC-0 specifies a safety system - the Open Fibre Control system (OFC) - for SW laser data links, since the optical power levels exceed the limits defined by the laser safety standards. If an open fibre condition occurs in the link, the receiver of the Port the fibre is connected detects it and pulses its laser at a low duty cycle that meets the safety requirements. The receiver of the other port (at the other end of the fibre) detects this pulsing signal and also pulses it's transmitter at a low duty cycle. When the open fibre path is restored both ports receive the pulsing signals, and after a double handshaking procedure the connection is automatically restored within a few seconds.

NL_Port Node Hierarchy of Processes

- Node
  - FC-0 FC-1
  - FC-2
  - Input Buffer
    - Decode Sequence
      - Arbitrating Seq
      - Arbitrating Won Seq
      - Initializing Seq
      - Monitoring Seq
      - Open seq
      - Open-Init seq
      - Opened Seq
      - Received Close Seq
      - Sort Sequence
      - Transfer Seq
      - Xmitted Close Seq
    - Decode Signal
      - Arbitrating
      - Arbitration Won
      - Initializing
      - Monitoring

        - Open
        - Open-Init
        - Opened
        - Received Close
        - Sort State
        - Transfer
        - Xmitted Close
    - Route Frame
    - Split Primitive
- L_Port Controls
- Entry Actions
- LPSM
- Next Transmission Word Out
- Node Port Transmit Buffer
- Set Frame
- Host Interface
- Send Frame