X3T11/
X3T11/Project 818R/REV 3.0

# HIGH-PERFORMANCE PARALLEL INTERFACE - Physical Switch Control

# (HIPPI-SC)

draft revision of
American National Standard
for Information Systems
X3.222-1993

January 24, 1996

Secretariat:

Information Technology Industry Council (ITI)

ABSTRACT: This revision of ANSI X3.222-1993 provides a protocol for controlling physical layer switches which are based on the High-Performance Parallel Interface, a simple high-performance point-to-point interface for transmitting digital data at peak data rates of 800 or 1600 Mbit/s between data-processing equipment.

*NOTE:*

*This is an internal working document of X3T11, a Technical Committee of Accredited Standards Committee X3. As such, this is not a completed standard. The contents are actively being modified by X3T11. This document is made available for review and comment only. For current information on the status of this document contact the individuals shown below:*

POINTS OF CONTACT:

Roger Cummings (X3T11 Chairman)
Storage Technology Corporation
2270 South 88th Street
Louisville, CO 80028-0268
(303) 661-6357, FAX (303) 684-8196
E-mail: Roger_Cummings@Stortek.com

Carl Zeitler (X3T11 Vice-Chairman)
IBM Corporation, MS 9440
11400 Burnet Road
Austin, TX 78758
(512) 838-1797, FAX (512) 838-3822
E-mail: zeitler@ausvm6.vnet.ibm.com

Don Tolmie (HIPPI-SC Technical Editor)
Los Alamos National Laboratory
CIC-5, MS-B255
Los Alamos, NM 87545
(505) 667-5502, FAX: (505) 665-7793
E-mail: det@lanl.gov

**ANSI ®**
**X3.222-199x**
Revision of
ANSI X3.222-1993

American National Standard
for Information Systems –

# High-Performance Parallel Interface – Physical Switch Control (HIPPI-SC)

Secretariat

**Information Technology Industry Council (ITI)**

Approved , 199

**American National Standards Institute, Inc**

## Abstract

This revised standard provides a protocol for controlling physical layer switches which are based on the High-Performance Parallel Interface, a simple high-performance point-to-point interface for transmitting digital data at peak data rates of 800 or 1600 Mbit/s between data-processing equipment.

## American National Standard

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made towards their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give interpretation on any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

**CAUTION NOTICE:** This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

## Patent Statement

**CAUTION:** The developers of this standard have requested that holder's of patents that may be required for the implementation of this standard, disclose such patents to the publisher. However, neither the developers nor the publisher have undertaken a patent search in order to identify which, if any, patents may apply to this standard.

As of the date of publication of this standard and following calls for the identification of patents that may be required for the implementation of the standard, no such claims have been made. No further patent search is conducted by the developer or the publisher in respect to any standard in process. No representation is made or implied that licenses are not required to avoid infringement in the use of this standard.

Published by

**American National Standards Institute**
**11 West 42nd Street, New York, New York 10036**

Printed in the United States of America

## Contents

Page

### Annexes

### Figures

## Foreword (This Foreword is not part of American National Standard X3.222-1993.)

This High-Performance Parallel Interface, Physical Switch Control (HIPPI-SC) standard defines the control for HIPPI physical layer switches. HIPPI is an efficient simplex high-performance point-to-point interface. HIPPI physical layer switches may be used to give the equivalent of multi-drop capability, connecting together multiple data processing equipments. The HIPPI is designed for transmitting data at peak rates of 800 or 1600 Mbit/s between data processing equipment.

This document includes annexes which are informative and are not considered part of the standard.

This standard was developed by Task Group X3T9.3 of Accredited Standards Committee X3 during 1990 and 1991. The standards approval process started in 1991. X3T9.3 was reconstituted as X3T11 in 1993.

Requests for interpretation, suggestions for improvement or addenda, or defect reports are welcome. They should be sent to the X3 Secretariat, Computer and Business Equipment Manufacturers Association, 1250 Eye Street, NW, Suite 200, Washington, DC 20005.

This standard was processed and approved for submittal to ANSI by Accredited Standards Committee on Information Processing Systems, X3. Committee approval of the standard does not imply that all committee members voted for approval. At the time it approved this standard, the X3 Committee had the following members:

> *(List of X3 Committee members to be included in published standard by the ANSI Editor.)*

Subcommittee X3T11 on Device Level Interfaces, which developed this revised standard, had the following participants:

Roger Cummings, Chair
Carl Zeitler, Vice-Chair
Horst Truestedt, International Representative
Don Tolmie, HIPPI-SC R1 Technical Editor

I. Ahmed
D. Allan
D. Allen
B. Atkinson
R. Bailly
B. Bellino
C. Binford
P. Boulay
C. Brill
J. Bumblis
E. Cady
J. Calle
K. Chan
E. Chang
M. Chenery
K. Chennappan
H. Chin
T. Clay
R. Cook
J. Coomes
R. Cornelius
S. Darnell
J. Dedek
D. Deel
M. Dorsett
P. Duclos
B. Edge
S. Erler
F. Ferguson
D. Flanagan
M. Fogg
D. Ford
C. Foster
M. Foster
K. Fredericks
G. Freeman
E. Frymoyer
B. Gallagher
B. Galloway
E. Gardner
F. Gaullier
C. Grant
M. Griffin
E. Grivna
P. Haas
D. Hagerman
B. Ham
N. Harris
B. Harvie
V. Haydu
D. Hepner
J. Himes
M. Hoard
E. Jacques
T. Johnson
S. Joiner
D. Kachelmeyer
M. Karg
M. Kelley
R. Kembel
R. Kleckowski
O. Kornblum

B. Kowalski
L. Lamers
M. Leib
W. Leslie
B. Levin
J. Lohmeyer
P. Maher
K. Malavalli
W. Martin
J. Mathis
M. McGowen
V. Melendy
J. Merkwaz
S. Mindemann
J. Murdock
K. Nakamura
R. O'Connell
J. Oliver
T. Palkert
A. Parsons
R. Pedersen
R. Petersen
S. Poole
G. Porter
K. Powell
R. Prentice
S. Quan
G. Rara
J. Renwick
C. Rich
W. Rickard
R. Ronald
B. Rumer
P. Rupert
J. Sherfey
M. Shiflett
M. Shriki
K. Smith
R. Snively
T. Sprenkle
J. Stai
G. Stephens
B. Stoevhase
A. Stone
C. Stoops
R. Taborek
K. Teow
L. Thorsbakken
J. Toy
S. Ucci
H. Van Deusen
S. Van Doorn
T. Voss
R. Wagner
P. Walford
N. Wanamaker
G. Warden
L. Whitfield
S. Wilson
M. Wingard
K. Witte
M. Zandy

## Introduction

This High-Performance Parallel Interface, Switch Control (HIPPI-SC) standard defines the control for HIPPI physical layer switches. HIPPI by itself is an efficient simplex high-performance point-to-point interface. The physical switch control allows the interconnection of multiple HIPPI based equipments with HIPPI physical layer switches.

Characteristics of this HIPPI physical switch control protocol include:

- Support for both source routing and destination addresses;
- I-Fields and CCIs can span multiple physical layer switches within a fabric;
- When a Destination end-point receives a packet, it can easily manipulate the I-Field received to return a reply packet to the Source;
- Support for physical layer switches with differing numbers of ports, all within the same fabric.
- Specified reserved addresses to aid address self-discovery, switch management, and switch control.

This document includes annexes which are informative and not considered part of the standard.

American National Standard
for Information Systems

# High-Performance Parallel Interface
# Physical Switch Control (HIPPI-SC)

## 1 Scope

This American National Standard provides switch control for physical layer switches using the High-Performance Parallel Interface (HIPPI), a high-performance point-to-point interface between data-processing equipment. This standard does not protect against errors introduced by intermediate devices interconnecting multiple HIPPI-PHs.

The purpose of this standard is to facilitate the development and use of the HIPPI in computer systems by providing common physical switch control. The standard provides switch control structures for physical layer switches interconnecting computers, high-performance display systems, and high-performance, intelligent block-transfer peripherals. This standard also applies to point-to-point HIPPI topologies.

## 2 Normative references

This American National Standard contains provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the edition indicated was valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

ANSI X3.183-1991, *High-Performance Parallel Interface, Mechanical, Electrical, and Signalling Protocol Specification (HIPPI-PH)* .

ANSI X3.218-1993, *High-Performance Parallel Interface - Encapsulation of ISO 8802-2 (IEEE Std 802.2) Logical Link Control Protocol Data Units (HIPPI-LE).*

ANSI X3.xxx-199x, *High-Performance Parallel Interface - Serial Specification (HIPPI-Serial).*

## 3 Definitions and conventions

### 3.1 Definitions

For the purposes of this standard, the following definitions apply.

**3.1.1 connection:** Condition of the HIPPI-PH when data transfers from a Source end-point to a Destination end-point are possible.

**3.1.2 connection control information (CCI):** A parameter sent as part of the sequence of operations establishing a connection from a Source to a Destination.

**3.1.3 end-point:** The equipment at either end of the fabric for a particular connection.

**3.1.4 Destination:** The equipment at the end of the interface that receives the data.

**3.1.5 Destination end-point:** The equipment at the end of the fabric that receives the data.

**3.1.6 fabric:** A group of one or more physical layer switches that can be traversed with one I-Field.

**3.1.7 I-Field:** A 32-bit field that is sent as part of the sequence of the physical layer operations establishing a connection from a Source to a Destination.

**3.1.8 interface:** The set of protocols and control signals used to connect a Source and Destination, as defined by HIPPI-PH. Within a fabric, an interface connects an end-point to a switch or a switch to a neighboring switch.

**3.1.9 Logical Address:** An address stored in an I-Field that uniquely identifies a Destination end-point or set of end-points.

**3.1.10 nibble:** A 4-bit entity.

**3.1.11 optional:** Features that are not required by the standard. However, if any optional feature defined by the standard is implemented, it shall be implemented according to the standard.

**3.1.12 packet:** A data set, as defined by HIPPI-PH, sent from Source to Destination. A packet is composed of one or more bursts.

**3.1.13 physical layer switch:** A device which allows a single HIPPI physical layer interface to switch between multiple HIPPI physical layer interfaces without involving protocols above the HIPPI Mechanical, Electrical, and Signalling Protocol Specification (HIPPI physical layer).

**3.1.14 Source**: The equipment at the end of the interface that transmits the data.

**3.1.15 Source Address:** An address stored in an I-Field that uniquely identifies a Source end-point or set of end-points.

**3.1.16 Source end-point**: The equipment at the end of the fabric that transmits the data.

**3.1.17 source routing**: A means of packet routing whereby the Source end-point specifies the action of each switch on the way to the Destination.

### 3.2 Editorial conventions

In this standard, certain terms that are proper names of signals, state mnemonics, or similar terms are printed in uppercase to avoid possible confusion with other uses of the same words (e.g., REQUEST). Any lowercase uses of these words have the normal technical English meaning.

A number of conditions, sequences, parameters, events, states, or similar terms are printed with the first letter of each word in uppercase and the rest lowercase (e.g., Source). Any lowercase uses of these words have the normal technical English meaning.

## 4 CCI and I-Field formats

### 4.1 Format

The connection control information (CCI) shall be used for controlling HIPPI physical layer switches. Within X3.183-1991 (HIPPI-PH) the CCI is used as the I-Field, and is asserted on the HIPPI-PH Data Bus during a connection sequence. The format of the CCI (I-Field) is shown in figure 1. Examples of CCI and I-Field usage for routing are contained in annex A.

L = Locally Administered (bit 31) = 0 designates that the I-Field is defined by this standard. L = 1 designates that the rest of the I-Field, bits 30 - 0, are locally administered and are not defined by this standard.

VU = Vendor_Unique (bits 30,29). The contents of the Vendor_Unique bits are not defined in this standard. Switches shall pass these bits unmodified to the Destination.

NOTE - These bits are available for providing signals to Destinations. Such signals can be used to modify the Destination's behavior or supply it with additional information on the purpose of the attempted connection.

W = Double-wide (bit 28) = 0 designates that the Source is using the 800 Mbit/s data rate option (DATA BUS is 32 bits wide as defined in HIPPI-PH); the switch shall connect through Cable-A. W = 1 designates that the Source is using the 1600 Mbit/s data rate option (DATA BUS is 64 bits wide); the switch shall connect through both Cable-A and Cable-B.

NOTE - The W bit is used in conjunction with the INTERCONNECT signals on Cable-A and Cable-B. The INTERCONNECT signals, as defined in HIPPI-PH, tell a switch or end-point that the cable is physically attached to an active HIPPI port. The W bit is used to tell the switch, or Destination end-point, whether or not Cable-B is being used in particular connection.

D = Direction (bit 27) = 0, designates that the right-hand end (least significant bits) of the Routing Control field shall be the current sub-field. D = 1 designates that the left-hand end (most significant bits) of the Routing Control field shall be the current sub-field.

NOTE - When a reverse path exists, a Destination end-point may return a reply to a received packet by simply using the same I-Field that it received with the D bit complemented. For this to work correctly with source routing (PS = 00) then the return path must be symmetrical with the forward path.

PS = Path Selection (bits 26,25). Used to select either (1) a source route, i.e., a specific route through the switches, with output port numbers specified for each switch, or (2) to specify the Logical Address.

00 = source routing: Source selects the route through the switches.

01 = Logical address: Switches select the first route from a list of possible routes.

10 = reserved

11 = logical address: Switches select a route.

C = Camp-on (bit 24) = 0 specifies that the switch shall reply with a connection reject sequence if unable to complete the connection. C = 1 specifies that the switch shall attempt to establish a connection until either the connection is completed or the Source aborts the connection request.

Camp-on is used by the Source to tell a switch to wait for the selected path (or paths) to become available, i.e., the switch should not generate a rejected connection sequence because the selected path is busy. The algorithm used by a switch to select among multiple Sources camped-on to a single Destination is implementation-specific and is not specified in this standard.

NOTE - A HIPPI rejected connection has a different set of meanings depending on whether or not the Camp-on feature is being used. See B .1 for details.

### 4.2 Source routing

When PS = 00, i.e., source routing, the Routing Control field shall be split into multiple sub-fields, with the size of each sub-field dependent upon the size of the switch that is using it. The number of bits in the sub-field is described as $\lceil \mathrm{Log}_2 N \rceil$ where N is the switch size. For example, a 16 by 16 switch would use a four-bit sub-field.

When D = 0, a switch shall use the current sub-field (right most bits of the routing control field) to select the switch output port. The switch shall right shift the routing control field by the number of bits in the sub-field, and shall insert the switch input port number in the left most bits of the routing control field. See figure 2.

When D = 1, the same actions occur except that the current sub-field shall be at the left end of the Routing Control field. The Routing Control field shall be shifted left, and the input port number shall be inserted at the right end of the Routing Control field. See figure 3.

A switch shall not alter the I-Field except when PS=00, and then, only the Routing Control field shall be modified.

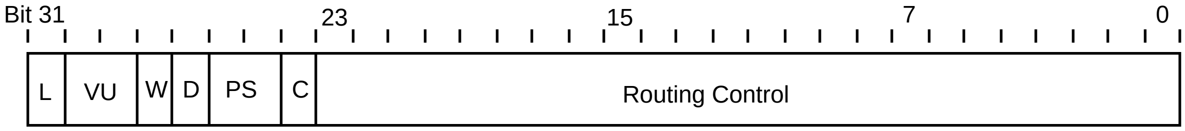

**Figure 1 – CCI and I-Field format**

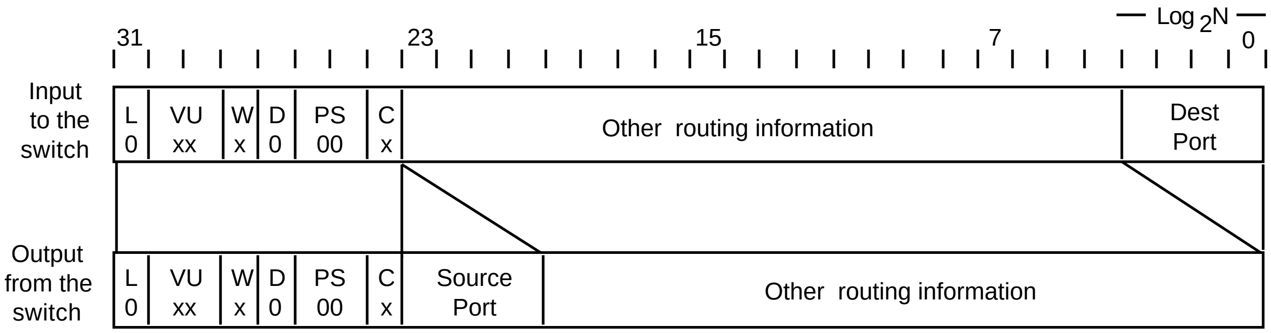

**Figure 2 – I-Field with source routing, D = 0, and 16 by 16 switch**

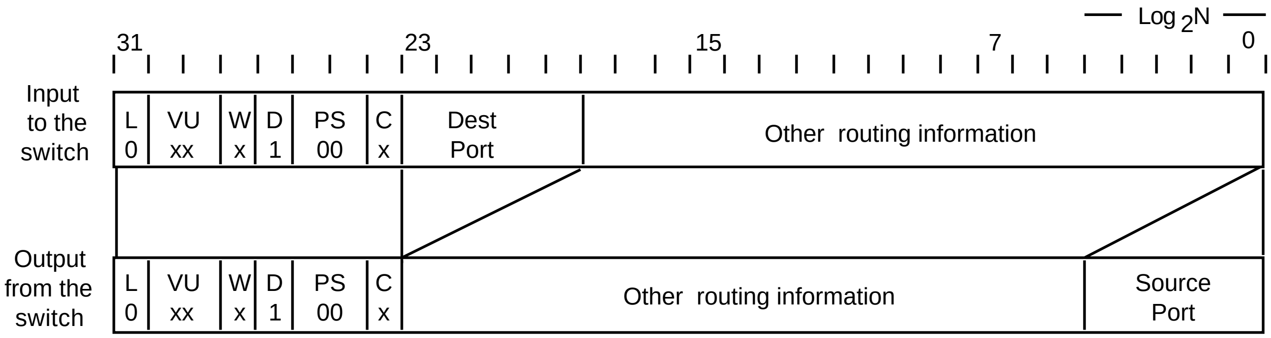

**Figure 3 – I-Field with source routing, D = 1, and 32 by 32 switch**

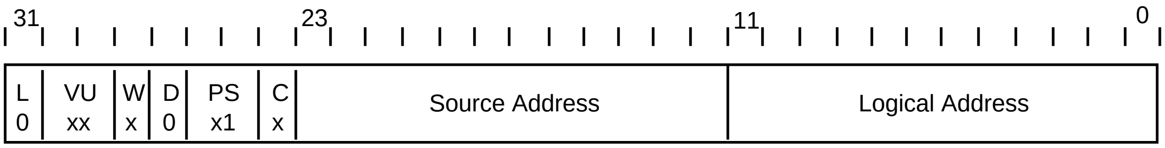

**Figure 4 – I-Field with logical addressing and D = 0**

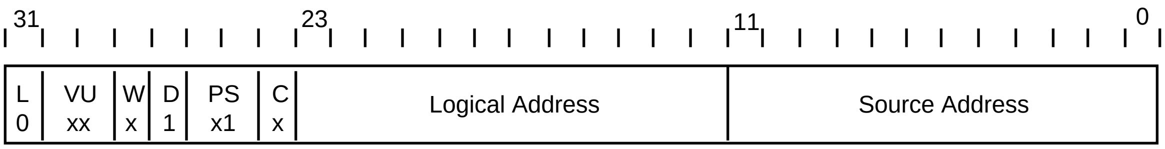

**Figure 5 – I-Field with logical addressing and D = 1**

### 4.3 Logical address

When PS = 01 or 11, i.e., logical address, the Routing Control field shall be split into two 12-bit fields. One 12-bit field specifies the address of the Destination end-point(s), the other specifies the address of the Source end-point. When the direction D bit = 0, the right-hand 12 bits shall specify the Destination end-points Logical Address and the left-hand 12 bits shall specify the Source end-points Logical Address (see figure 4). When D = 1, the opposite is true (see figure 5).

### 4.4 Reserved Logical Addresses

Part of the range of logical addresses are reserved to designate the addresses of network services whose location in the network may vary, and for other network management functions. All others are available for assignment to specific Destinations. The logical addresses assigned at the time this standard was approved include the following (shown in hexadecimal notation).

NOTE - Later registrations will be added as an addendum to this standard, or a revision of the standard.

F90 - FBF Trial self-discovery addresses. These addresses are reserved for an iterative address self-discovery algorithm. F9x tests the low-order 4-bit nibble, FAx tests the middle 4-bit nibble, and FBx tests the high-order 4-bit nibble, of the switch port's logical address. If the low-order 4-bit nibble of the trial self-discovery address, i.e., the "x" in Fnx, matches the selected nibble in the switch port's logical address, then the switch shall establish a connection back to a HIPPI Destination that is paired with the HIPPI port requesting the connection, i.e., a loopback. If the trial self-discovery nibble values are not equal, the switch shall reject the connection. For example, if the logical address for port "J" is "xyz" where "x", "y", and "z" are 4-bit nibbles, then loopback connections shall be made when trial logical addresses F9"z", FA"y", and FB"x" are used. (See B.3.3 and B.3.5.) Support of this feature is optional.

FC0 - FDF Reserved for local use

FE0 Messages pertaining to switch configuration, including HIPPI-LE Address Resolution requests as described in RFC 1374 "IP and ARP on HIPPI." [1]

FE1 All IP protocol traffic conventionally directed to the IEEE 802.1 broadcast address as described in RFC 1042 "Standard for IP transmission over 802 networks" [2]

FE2 RFC 1112 Host extensions for IP multicasting Class D addresses not assigned below. [3]

FE3 RFC1131 OSPF specification All Routers (Class D address 224.0.0.5) [4]

FE4 RFC1131 OSPF specification All Designated Routers (Class D address 224.0.0.6) [4]

FE5 - FE7 Reserved

FE8 ISO/IEC 9542:1988 CLNP ES-IS all ES's [5]

FE9 ISO/IEC 9542:1988 CLNP ES-IS all IS's [5]

FEA ISO/IEC 10589:1992 IS-IS all Level 1 IS's [6]

FEB ISO/IEC 10589:1992 IS-IS all Level 2 IS's [6]

FEC IEEE 802.1d MAC Bridging flooding. [7]

FED IEEE 802.1d MAC Bridging Spanning Tree Protocol. [7]

FEE Embedded switch management agent. Support of this feature is optional.

FEF - FFC Reserved

FFD Loopback logical address for switches to use when probing other switches. Support of this feature is optional.

FFE Loopback logical address for hosts to use when probing switches for the host's logical address. This value is reserved for the establishment of a connection back to a HIPPI Destination that is paired with the HIPPI Source requesting the connection. (See B.3.2 and B.3.5.) Support of this feature is optional.

FFF Unknown or unassigned address. This value should never be used to address a Destination or Destinations. It can be used to indicate that the Source is unaware of its Source Address in the CCI, or to signify an unknown Logical Address in higher layer protocols. A HIPPI-SC switch may alter the I-Field, substituting a valid Source Address for this value. Support for this address substitution is optional. Such substitution may be used to aid in discovery of a system's logical switch address by higher layer protocols. (See B.3.1, B.3.2, and B.3.5.)

The protocols used to access these services and the means whereby these services keep track of their configuration of the network are outside the scope of this standard.

## 5 Switch behavior

A HIPPI physical switch has input ports (attachments to HIPPI Sources) and output ports (attachments to HIPPI Destinations). These HIPPI ports shall conform to either the HIPPI-PH or HIPPI-Serial specifications. This clause defines how an HIPPI physical switch behaves with regards to the states of the HIPPI-PH control signals on the input and output ports for a particular connection operation.

### 5.1 Use of INTERCONNECT signals

As defined in HIPPI-PH, each switch input port and output port shall generate an INTERCONNECT signal when that port is "on-line"; i.e., powered on and enabled for HIPPI connections. Each switch input and output port shall monitor the received INTERCONNECT signal and shall use this signal to validate all other HIPPI control signals.

NOTE - A switch port may deassert the INTERCONNECT signal when that port is disabled for maintenance or diagnostics.

## 5.2 CLOCK signal

The HIPPI CLOCK signal generated by the switch output port (a HIPPI Source) shall be continuous and shall conform to the HIPPI-PH specification at all times.

## 5.3 Connection request successful

Once a connection is completed the switch shall be transparent, with the exception of switch induced latency, to the HIPPI signal sequences.

NOTE - The switch acts as a repeater under the constraints imposed by subclause 7.9 of ANSI X3.183, and can change the number of idle words between bursts or packets.

## 5.4 Breaking a connection

Either the Source end-point or the Destination end-point may break a connection.

### 5.4.1 Source deasserts REQUEST

When the Source end-point deasserts the REQUEST signal, the switch shall break the connection. The switch shall not wait for the Destination end-point to deassert the CONNECT signal to break the connection. The Source end-point will see the CONNECT signal go false regardless of Destination end-point actions.

NOTE - This immediate disconnection frees the associated input port for the next connection-request. This maximizes the efficiency of the HIPPI attachment by allowing the Source end-point to make a new connection without waiting for the propagation delay and the Destination end-point disconnect turn-around time.

### 5.4.2 Destination deasserts CONNECT

When the Destination end-point deasserts the CONNECT signal, the connection through the switch shall be broken. The switch shall not wait for the Source end-point to deassert the REQUEST signal to break the connection. The Destination end-point will see the REQUEST signal go false regardless of Source end-point actions.

### 5.4.3 INTERCONNECT false

If the INTERCONNECT signal, received by either the switch input port or the switch output port, goes false during any stage of a connection then the connection through the switch shall be broken.

## 5.5 Connection request unsuccessful

Connection requests can be unsuccessful due to

– Unavailable Destination end-points

– Unavailable fabric resources

– Errors

### 5.5.1 Down-stream connection reject

A rejected connection sequence can be initiated by either a down-stream switch or the Destination end-point. A rejected connection sequence shall be propagated through the switch without change. When the switch detects the CONNECT signal is false, at the end of the sequence, the connection through the switch shall be broken.

### 5.5.2 Switch-generated connection reject

The switch input port logic shall initiate a rejected connection sequence to the Source end-point, and shall not complete the connection through the switch, in the following situations

– When the selected output port(s), or Destination end-point, does not exist or is unavailable. For example, when the INTERCONNECT signal received by the selected output port is false.

– When the selected output port(s) is(are) busy, i.e., connected to some other port, and Camp-on = 0.

– When a data parity error is detected on the I-Field while connecting through this switch. The switch is not obligated to check the I-Field parity after the REQUEST signal has been propagated through the switch.

– When the I-Field Path Selection (bits 26,25) specifies an addressing mode not supported or disabled.

– When the I-Field Path Selection (bits 26,25) specifies logical addressing, and the Destination logical address is not mapped to any physical output port.

– When the I-Field Double-wide (bit 28) selection conflicts with the switch capabilities or the end-point capabilities. Clause B.2 discusses such conflicts.

– When the I-Field Locally Administered (bit 31) selection is not supported by the switch.

### 5.5.3 Connection request contention

When two or more input ports vie for the same output port only one input port can "win". The winner shall be connected through the switch. The loser(s) shall either wait for the output port to become available (i.e., Camp-on = 1), or the input port(s) shall generate a rejected connection sequence (i.e., Camp-on = 0).

# Annex A
(informative)

# Routing with the CCI and I-Field

## A.1 General example

The CCI and I-Field are use to control HIPPI physical layer switches, supporting the interconnection of many HIPPI devices. Figure A.1 is an example of a small general configuration that will be used to describe the operation of the CCI and I-Field as specified in clause 4. Three hosts, A, B, and C, are shown, but there will probably be many more hosts connected in an actual configuration. The switching fabric is the interconnection mechanism, in this example the four switches and the interconnecting HIPPI links.

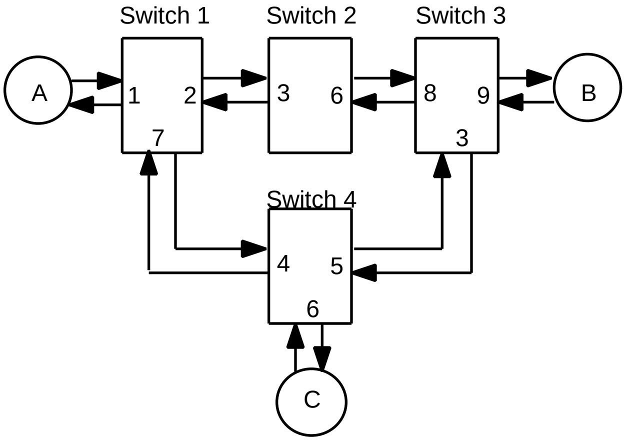

**Figure A.1 – Physical layer switch example**

Two types of operation are specified: (1) source routing, and (2) logical address. The direction of interpretation of the Routing Control field is also under user control, allowing a Destination end-point to return a reply by simply using the same I-Field that was received with the direction bit complemented.

## A.2 Source routing

With source routing, the Source end-point specifies the action of each switch on the way to the Destination. For example, to go from host-A to host-B, host-A could specify 2,6,9. This would result in switch-1 selecting output port-2, switch-2 selecting output port-6, and switch-3 selecting output port-9. Alternatively, host-A could have specified 7,5,9 to go through switch-4 instead of switch-2.

For connection or packet routing, this involves an end-point (the "originator" or "Source") having to know the physical route to a particular Destination before a connection can be established. When using multiple switches (where the switches have no intelligence with regards to network routing), the Source has to establish the entire physical route for a given connection. Source routing can often facilitate low latency connections because the switches have no burden of decision making during the connection process. However, source routing can be unattractive for a configuration with the following features: (1) large and/or dynamic configuration (the hosts must keep track of the interconnection configuration), (2) "blocking" configuration (there is a path to the Destination available, but it is not the one specified, and the one specified is unavailable), and (3) no mechanism that allows a node to "discover" the addresses and associated routes for other nodes

Source routing can be very valuable for diagnostics. By specifying a particular path, a diagnostic program can determine if that path is operable. Source routing can also be attractive when certain paths are more desirable to use than other paths based on criteria that the switches are unaware of, e.g., usage costs associated with particular links.

When using source routing with the CCI and I-Field as specified in clause 4, PS = 00 and the Routing Control contains the route information. The eight high-order bits of the I-Field will be called Ctl. If the switches are 16 x 16 switches using 4-bit addresses, and the direction bit (D) = 0, then to go from host-A to host-B through switch-2, host-A would set the I-Field = Ctl,x,x,x,9,6,2, (in hexadecimal) . When received at switch-1, switch-1 picks off the low order four bits (the low order because D=0, four bits because the switch size = 16 x 16) and connects input port-1 to output port-2. It also shifts the Routing Control field of the I-Field to the right and inserts switch-1's input port number in the left end of the Routing Control field. When received at switch-2, the I-Field will be Ctl,1,x,x,x,9,6. Likewise, when it reaches switch-3 it will be Ctl,3,1,x,x,x,9. When it reaches host-B, it will be Ctl,8,3,1,x,x,x.

In summary:

Ctl = L = 0 (not a locally administered I-Field)
VU = xx (Vendor Unique bits ignored)
W = x (this example is not concerned with width)
D = 0 (direction bit)
PS = 00 (source routing)
C = x (this example is not concerned with Camp-on)

| | |
|---|---|
| At A and input to switch-1: | I-Field = Ctl, x, x, x, 9, 6, 2 |
| At input to switch-2: | I-Field = Ctl, 1, x, x, x, 9, 6 |
| At input to switch-3: | I-Field = Ctl, 3, 1, x, x, x, 9 |
| At host-B: | I-Field = Ctl, 8, 3, 1, x, x, x |

Host-B can send a reply packet back to host-A by complementing the D bit (direction) in the Ctl field. Now the switches will use the left most bits of the Routing Control field as the output port selector, shift the Routing Control field to the left, and insert the input port on the right end. In summary, it would look like:

Ctl = L = 0 (not a locally administered I-Field)
VU = xx (Vendor Unique bits ignored)
W = x (this example is not concerned with width)
D = 1 (direction bit *Note that it changed*)
PS = 00 (source routing)
C = x (this example is not concerned with Camp-on)

| | |
|---|---|
| At B and input to switch-3: | I-Field = Ctl, 8, 3, 1, x, x, x |
| At input to switch-2: | I-Field = Ctl, 3, 1, x, x, x, 9 |
| At input to switch-1: | I-Field = Ctl, 1, x, x, x, 9, 6 |
| At host-A: | I-Field = Ctl, x, x, x, 9, 6, 2 |

Switches other than the 16 x 16 switches of the example can also be used. In that case the number of bits used by each switch will be determined by the switch size, For example, a 64 x 64 switch will use six bits for selection of its output port, shift off six bits, and add in six bits of input port number.

Mixing switches of different sizes in the same fabric also works. Each switch uses, and shifts, the appropriate number of bits.

This use of the Routing Control field assumes that all of the cabling is symmetrical. Because the backwards path is not guaranteed to be available, (a) host-B can use the received I-Field as described above with multiple tries or Camp-on, or (b) host-B can use an alternate path which must be determined by other means, e.g., a look-up table.

The reverse path problem is minimized with a system running under the following rules:

A fabric-wide limit for maximum connection time. This will minimize the time it takes for a particular path to become available.

The switches have the capability of allowing connection requests to "camp-on" a particular output port to wait for it to become available.

The output port contention algorithm is fair.

## A.3 Using a logical address

Logical addresses specify where the packet is to be delivered, not the route to take to get there. Source end-points use the same Logical Address to reach a particular host, no matter where the Source is located; note that this is not the case with source routing.

With logical addresses the intermediate switches are responsible for picking an appropriate route. The Path Selection bits of the I-Field tell the switches whether to (1) use only the first output port from a list of possible ports, or (2) to select any output port from the list of possible ports. Hence, the switches that use logical addresses must do the port selection as well as the port switching, where source routing switches only need to do the port switching.

A major advantage of using logical addresses is that only the switches need to know the fabric interconnection topology, and the hosts only need to know the logical addresses. Hence, if a link or switch fails, the other switches can route around it without the hosts having to know about it or do anything special.

Logical addresses are specified in clause 4; PS = 01 or 11 and the Routing Control field contains the address information. The Routing Control field contains two 12-bit fields, one for the Source Address and the other for the Logical Address. The direction bit (D) specifies which field is which.

The use of symmetrical cabling is not necessary with Logical addresses, just the need for a path back to the Source. Management is also easier since only the switches need to know how to route among themselves, the hosts do not need to know the fabric interconnection configuration, and there will probably be many fewer switches than hosts. The switches can also use alternate paths to avoid congestion without having to go back to the Source host and try another whole route.

Security is also enhanced by the fact that the switches control the routing and can exercise some control (via their look-up tables) over which paths are legitimate for certain hosts. One feature that is desirable is to have some way for the hosts to "discover" the Logical Address of the hosts, for example, via a name server.

The Logical Address may specify either the end-point of the fabric (i.e., fabric-specific), or the unique address of the host attached to the fabric end-point (i.e., host-specific). The difference between the two is seen when a host is moved from one fabric end-point to another. In this case, the fabric-specific address would change but the host-specific address would not change. There are advantages and disadvantages to both schemes.

It is envisioned that switches can be built to use look-up tables to house routing paths to Destinations. A look-up table can be indexed using a 12-bit Logical Address. The look-up table would be used to hold a possible route (or routes) for each Destination. The management of the tables can be based on either of the addressing schemes: fabric-specific or host-specific.

### A.3.1 Fabric-specific addresses

In the following example, the fabric-specific addresses specify fabric end-points. Several schemes can be envisioned for fabric-specific addresses, but in this example the fabric-specific addresses are further broken into a switch number and port number. The fabric-specific address of host-A in figure A.1 is 1-1, host-B is 3-9, and host-C is 4-6. The eight high-order bits of the I-Field are called Ctl.

If the direction bit (D) = 0, then to go from host-A to host-B, host-A would set the I-Field = Ctl,1-1,3-9 (in hexadecimal) (3-9 is the Logical Address and 1-1 is the Source Address). When received at switch-1, switch-1 picks off the low order 12 bits (the low order because D=0) as the Logical Address. Even though these 12 bits contain a logical partition between the bits holding the "switch number" and the bits holding the "port number", all 12 bits can be used together as an index into the look-up table. Because this switch is not the switch specified in the Destination Logical Address, the look-up table will provide one or more possible routes to the Destination end-point (we will assume that it picks output port-2 on switch-1). The I-Field is then passed, unchanged, to switch-2. None of the switches change the I-Field as the packet is routed.

A similar action occurs at switch-2, passing the I-Field to switch-3. When switch-3 receives the I-Field, its look-up table yields the output port number specified in the Logical Address (the 9 of 3-9), and selects output port-9.

In summary:

Ctl = L = 0 (not a locally administered I-Field)
VU = xx (Vendor Unique bits ignored)
W = x (this example is not concerned with width)
D = 0 (direction bit)
PS = 01 or 11 (logical address)
C = x (this example is not concerned with Camp-on)

| | |
|---|---|
| At host-A and input to switch-1: | I-Field = Ctl, 1-1, 3-9 |
| At input to switch-2: | I-Field = Ctl, 1-1, 3-9 |
| At input to switch-3: | I-Field = Ctl, 1-1, 3-9 |
| At host-B: | I-Field = Ctl, 1-1, 3-9 |

Host-B can send a reply packet back to host-A by complementing the D bit (direction) in the Ctl field. Now the switches will use the left 12 bits of the Routing Control field as the Logical Address and the right 12 bits as the Source Address. (Note that the actions taken by the receiving node are the same for both source routing and logical addresses.) In summary, it would look like:

Ctl = L = 0 (not a locally administered I-Field)
VU = xx (Vendor Unique bits ignored)
W = x (this example is not concerned with width)
D = 1 (direction bit *Note that it changed*)
PS = 01 or 11 (logical address)
C = x (this example is not concerned with Camp-on)

| | |
|---|---|
| At host-B and input to switch-3: | I-Field = Ctl, 1-1, 3-9 |
| At input to switch-2: | I-Field = Ctl, 1-1, 3-9 |
| At input to switch-1: | I-Field = Ctl, 1-1, 3-9 |
| At host-A: | I-Field = Ctl, 1-1, 3-9 |

Note that the fabric-specific address does not need to be explicitly broken into switch number and port number fields, it could be all one field. Each switch still determines the path to take to the final Destination, i.e., which output port to use.

### A.3.2 Host-specific addresses

In the following example, the host-specific addresses specify unique hosts. For example, the 12-bit host-specific Logical Address for host-A in figure A.1 could be hexadecimal "AAA". Both host-B and host-C would use this address when establishing a connection with host-A. In the following example, the host-specific address for host-B is hexadecimal "BBB". Note that the host-specific addresses are independent of the fabric end-point addresses.

If hosts A and B have addresses "AAA" and "BBB", then to go from host-A to host-B, host-A would set the I-Field = Ctl,AAA,BBB (this assumes that the direction bit = 0). When received at switch-1, switch-1 uses the low order 12 bits (the low order because D=0) as an index into a look-up table to obtain the possible output ports associated with the Logical Address. Switch-1 could determine that output port-2 can be used to reach Destination BBB. Switch-2 would then use the same process to determine that it should use output port-9. Alternatively, switch-1 could have chosen to use output port-7; passing the connection through switch-4 to get to switch-3.

In summary:

Ctl = L = 0 (not a locally administered I-Field)
VU = xx (Vendor Unique bits ignored)
W = x (this example is not concerned with width)
D = 0 (direction bit)
PS = 01 or 11 (logical address)
C = x (this example is not concerned with Camp-on)

| | |
|---|---|
| At host-A and input to switch-1: | I-Field = Ctl, AAA, BBB |
| At input to switch-2: | I-Field = Ctl, AAA, BBB |
| At input to switch-3: | I-Field = Ctl, AAA, BBB |
| At host-B: | I-Field = Ctl, AAA, BBB |

Host-B can send a reply packet back to host-A by complementing the D bit (direction) in the Ctl field. Now the switches will use the left 12 bits of the Routing Control field as the Logical Address and the right 12 bits as the Source Address. (Note that the actions taken by the receiving node are the same for all of the routing schemes described.) In summary, it would look like:

Ctl = L = 0 (not a locally administered I-Field)
VU = xx (Vendor Unique bits ignored)
W = x (this example is not concerned with width)
D = 1 (direction bit *Note that it changed*)
PS = 01 or 11 (logical address)
C = x (this example is not concerned with Camp-on)

| | |
|---|---|
| At host-B and input to switch-3: | I-Field = Ctl, AAA, BBB |
| At input to switch-2: | I-Field = Ctl, AAA, BBB |
| At input to switch-1: | I-Field = Ctl, AAA, BBB |
| At host-A: | I-Field = Ctl, AAA, BBB |

Host-specific addresses have the advantage that when hosts are moved from one fabric end-point to another, the host-specific address does not change and hence the address tables in the other hosts do not need to be changed. Note that the look-up tables within the switches do need to be updated to reflect the new route to the host. Some disadvantages of host-specific addresses might be longer fabric reconfiguration times to update the switch tables, and problems with keeping the host-specific addresses unique (i.e., no duplicate addresses).

Fabric-specific addresses have the advantages of guaranteeing address uniqueness and quicker fabric reconfiguration.

By specifying the address sub-fields of the I-Field as unpartitioned, the sub-fields can be used for fabric-specific or host-specific addresses.

## A.4 Connections between fabrics

As defined in clause 4 the number of hosts available on a single fabric of HIPPI switches is limited to 4096-112, or 3984. Also the fabric is limited to a single type of Path Selection at one time, i.e., you cannot go part way through a fabric with a source route and then change over to a logical address.

To go beyond 3984 connections, or to connect dissimilar fabrics, a gateway will probably be required. The gateway would use higher layers in the HIPPI protocols to find an address, e.g., an IEEE 48-bit address (see HIPPI-LE), which would be translated to a new CCI and I-Field for the next fabric.

# Annex B
(informative)

# Implementation considerations

## B.1 Special considerations for Camp-on

The use of the Camp-on feature alters the possible reasons for a rejected connection sequence to occur.

### B.1.1 Camp-on not selected

When Camp-on = 0, a received rejected connection sequence tells the Source end-point that either a network resource is busy (e.g., the selected switch output port) or that the Destination end-point has rejected the connect request (e.g., out of buffer space). Because a switch output port being busy may be a fairly common occurrence, the Source end-point should not treat the rejected connection as an abnormal connection state, e.g., it should not log the rejected connection as an error.

### B.1.2 Camp-on selected

When Camp-on = 1, a received rejected connection sequence tells the Source end-point that the switch fabric was available, but the Destination end-point is overloaded (e.g., out of buffer space). Because an overloaded Destination end-point may be an uncommon and/or an undesirable occurrence, the Source end-point may want to treat this connection request as an exception, e.g., log as error.

Note that intermediate fabric resources may be tied up indefinitely by long-duration successful connections, or pending connection requests that are using the Camp-on feature.

## B.2 Interpretation of the W bit and INTERCONNECT signals

HIPPI-PH defines a 1600 Mbit/s option for the use of HIPPI which uses a second set of HIPPI signals (Cable-B) to transfer additional data and parity while connection control is operating on the primary Cable-A. HIPPI-PH discusses proper interpretation of the INTERCONNECT signals on Cable-A and Cable-B to determine if a HIPPI interface between a Source and Destination is capable of supporting double width operations.

The W bit within the CCI field is used to specify the Source end-point's intent to use Cable-B for the duration of the connection to the Destination end-point. The Destination should ignore all data and parity signals on Cable-B until BURST is asserted by the Source for an accepted 64 bit connection.

Since the value of the W bit can conflict with the state of the INTERCONNECT lines on Cable-A and Cable-B, the following table provides the actions which should be taken upon receipt of the CCI by Destinations which are either Destination end-points or the input ports of intermediate HIPPI switches.

| SOURCE | | | DESTINATION | |
|---|---|---|---|---|
| I-A | I-B | W bit | Type | Action taken |
| 0 | N/A | any | 32 | ignore |
| 0 | any | any | 64 | ignore |
| 1 | N/A | 0 | 32 | accept (32 bits) |
| 1 | 0 | 0 | 64 | accept (32 bits) |
| 1 | N/A | 1 | 32 | reject |
| 1 | 0 | 1 | 64 | reject |
| 1 | 1 | 0 | 64 | accept (32 bits) |
| 1 | 1 | 1 | 64 | accept (64 bits) |

Where:

I-A is the logical AND of the two Cable-A INTERCONNECT signals exchanged between Source and Destination.

I-B is the logical AND of the two Cable-B INTERCONNECT signals exchanged between Source and Destination. Certain states are not applicable as a 32 bit Destination is unable to determine the state of a Sources Cable-B.

W bit is the value of the W bit provided by the Source end-point.

Type is the Destinations HIPPI capability: 32 if only Cable-A is installed, 64 if a Cable-B interface is present.

Action Taken is the Destinations response:

ignore indicates that no action should be taken in the absence of both INTERCONNECT signals on Cable-A.

accept indicates that the connection is acceptable pending examination of other CCI fields.

reject the Destination should reject the connection as the W bit value contradicts the state of the INTERCONNECT lines.

## B.3 Logical address self-discovery by a host connected to a HIPPI-SC switch

Some applications require that a HIPPI host connected to a HIPPI switch be able to discover what the switch thinks that HIPPI host's Source logical address is. This section discusses methods available when a host is connected by a HIPPI Source/Destination pair with both cables connected to the same HIPPI-SC switch port. All of the addresses in this section are in hexadecimal notation. The precedence for address self-discovery procedures is:

a) unknown or unassigned address plus loopback logical address;

b) trial self-discovery addresses; and

c) received logical address.

Some reserved logical address, as defined in 4.4 and shown below, can be used for logical address self-discovery.

| | |
|---|---|
| F90 - FBF | Trial self-discovery addresses |
| FFE | Loopback logical address |
| FFF | Unknown or unassigned address |

### B.3.1 Substitution for the unknown address

This feature is optional in HIPPI-SC compliant switches. When a switch supporting this feature receives an I-Field with a Source Logical Address = FFF, i.e., unknown or unassigned address, the switch substitutes a pre-configured logical address for the Source Logical Address before transmitting the I-Field to the selected output port. The method by which this logical address is configured is outside the scope of this document. The effect of this action is to inform the Destination of the Source's logical address, even when the Source does not know it.

### B.3.2 Loopback logical address

This feature is optional in HIPPI-SC compliant switches. When a switch supporting this feature receives an I-Field with a Destination Logical Address = FFE, i.e., loopback logical address, it connects the input port to the output port with the same number as the input port. This may be used as an aid in a loopback diagnostic test, or in address self-discovery.

Using the unknown logical address, and loopback logical addresses, with a switch that supports them, a host can discover its own logical address by asserting the REQUEST signal with the I-Field = 03FFFFFE. This value uses a PS field of 01 to select logical addressing, and Camp-on. The host will receive the I-Field value of 03xyzFFE, where "xyz" is its own logical address as long as "xyz" ≠ FFF.

### B.3.3 Using trial self-discovery addresses

Trial self-discovery addresses, as specified in 4.4, can be used in an iterative fashion to determine a host's logical address after it has been determined that unknown and loopback logical addresses are not supported. Each group of connection trials, with 16 possible connection trials per group, can determine one 4-bit nibble of the logical address. Hence, determining a 12-bit logical address requires three groups, or a maximum of 48 total trials.

### B.3.4 Using the received Logical Address

A less direct, and possibly less reliable, method of address self-discovery is to simply use the Destination logical address from any logical address I-Field that the host receives. This method depends upon some other network device doing a connection to the host.

### B.3.5 Host's address self-discovery algorithm

An example algorithm, in pseudo-code for a HIPPI host, that combines the self-discovery techniques, and also will work with switches that do not support the loopback address and unknown address substitution, is detailed below. Note that a HIPPI host is not required to implement this pseudo-code, but if host Logical Address self-discovery is desired, then this pseudo-code can be used as a guide. The notation ASD stands for address self-discovery. The host's Logical Address is called xyz, where each letter corresponds to a 4-bit nibble.

#### B.3.5.1 ASD1000

Initialize the trial address value, and D_Adrs (storing the received Logical Address from another host's unexpected connection). If this host's Destination side is already connected, then wait for it to finish before starting, i.e., you cannot do a loopback if this host's Destination is already in use.

```
Set Trial_Address = F90, i.e., Initialize Trial_Address
Set D_Adrs = FFF, i.e., Initialize to "Unknown"
IF the host's HIPPI Destination is not connected
    THEN Goto ASD1010
    ELSE wait for host's HIPPI Destination to finish
```

#### B.3.5.2 ASD1010

Try using the loopback and logical address substitution features to determine the host's Logical Address as described in B.3.1 and B.3.2.

```
Issue Connection Request with I-Field = 03FFFFFE
        PS = 01, i.e., Logical Addresses
        Camp-on = 1, i.e., wait if already busy
        Source Address = FFF, i.e., Unknown
        Destination Address = FFE, i.e., Loopback
Goto ASD1020
```

#### B.3.5.3 ASD1020

Wait for the host's Destination side to receive a connection request. If the switch supports the loopback address, but not the address substitution, the Source Address in the received I-Field will be FFF, i.e., unchanged. Since address substitution did not work, try the trial self-discovery address procedure next.

```
IF Connect with I-Field = 03FFFFFE
    THEN Break the connection on the Source side
        Goto ASD1030
```

If everything works OK, then the host will receive a looped back connection request with the host's Logical Address in the Source Address field of the received I-Field.

```
ELSE IF Connect with I-Field = 03xyzFFE
    THEN xyz is the Host's Logical Address
        Set xyz = I-Field's xyz value
        Break the connection on the Source side
        EXIT; the Logical Address = xyz
```

If a connection from another host occurs while trying this operation, i.e., ships passing in the night, then check to see if it used a source route or logical addresses. If it is a source route, then ignore the I-Field received. Otherwise, the I-Field must contain logical addresses; save the Destination Address of the received I-Field for use as the host's Logical Address if it is not found another way. After processing this unexpected connection, wait for the queued (with Camp-on set) Connection Request issued by ASD1010.

```
ELSE IF Connect with other I-Field
    THEN another host is trying to connect
        IF PS = 00 in received I-Field, i.e., source route
            THEN Goto ASD1020
        ELSE Set D_Adrs = I-Field Destination Address
            Goto ASD1020
```

If the switch does not support loopback addresses, then it may reject the connection request. If so, then try the trial self-discovery address procedure next.

```
ELSE IF Connection Reject
    THEN Goto ASD1030
```

If the host's Source-side connection request is completed, without having first completed the host's Destination-side connection, then the Source is not in loopback and must be directly connected to another host rather than to a switch.

```
ELSE IF Source-side connection complete AND
            Destination-side connection not complete
    THEN host is connected to another host
        EXIT; the Logical Address = don't care
```

If it is not connected to another host, and does not support loopback addresses, then it should time out; try the trial self-discovery address procedure next.

```
ELSE IF Connection  not complete AND Timeout
    THEN Goto ASD1030
```

### B.3.5.4 ASD1030

Try a series of trial self-discovery addresses to determine the host's logical address, as described in 4.4 and B.3.3.

```
Issue Connection Request with:
        PS = 01, i.e., Logical Addresses
        Camp-on = 1, i.e., wait if already busy
        Source address = FFF, i.e., Unknown
        Destination Address = Trial_Address
Goto ASD1040
```

### B.3.5.5 ASD1040

Wait for the host's Destination side to receive a connection request. If the low-order nibble of this self-discovery address is the same as the selected nibble of the switch port's Logical Address, then the connection should be completed.

```
IF Connection with I-Field = 03FFFF9x
    THEN we have found the low-order nibble value
        Set z = low-order nibble of Trial_Address
        Set Trial_Address = FA0
        Break the connection on the Source side
        Goto ASD1030
ELSE IF Connection with I-Field = 03FFFFAx
    THEN we have found the middle nibble value
        Set y = low-order nibble of Trial_Address
        Set Trial_Address = FB0
        Break the connection on the Source side
        Goto ASD1030
ELSE IF Connection with I-Field = 03FFFFBx
    THEN we have found the high-order nibble value
        Set x = low-order nibble of Trial_Address
        Break the connection on the Source side
        EXIT; the Logical Address = xyz
```

If a connection from another host occurs while trying this operation, i.e., ships passing in the night, then check to see if it used a source route or logical addresses. If it is a source route, then ignore the I-Field received. Otherwise, the I-Field must contain logical addresses; save the Destination Address of the received I-Field for use as the host's Logical Address if it is not found another way. After processing this unexpected connection, wait for the queued (with Camp-on set) Connection Request issued by ASD1030.

```
ELSE IF Connect with other I-Field
    THEN another host is trying to connect
        IF PS = 00 in received I-Field, i.e., source route
            THEN Goto ASD 1040
        ELSE Set D_Adrs = I-Field Destination Address
            Goto ASD1040
```

If the low-order nibble of the self-discovery address does not match the selected nibble of the switch port's Logical Address, or the switch does not support the trial self-discovery address feature, then the connection will be rejected, or will time out.

```
ELSE IF Connection Reject or Timeout
    THEN Goto ASD1050
```

### B.3.5.6 ASD1050

Increment the trial self-discovery address. Check to see if we have tried all 16 possible values for this nibble. If not, try the next value. If all 16 have been tried with no success, then the switch is not responding or does not support the trial self-discovery procedure.

```
Increment Trial_Address
IF low-order nibble of Trial_Address ≠ 0
    THEN Goto ASD1030
```

As a last resort, use the Destination Address value contained in the I-Field of a connection from another host, if one was received while executing this address self-discovery algorithm. Note that if the substitution for the unknown address, or the trial self-discovery address technique, did not work, and no connection was received from another host, then the Logical Address value is set to unknown, i.e., FFF (the initial value set in ASD1000).

```
ELSE Set xyz = D_Adrs
    EXIT; the Logical Address = xyz
```

## B.4 CONNECT and READY signals

Since a clock signal does not accompany the CONNECT and READY signals, using the internal clock of a HIPPI switch implementation to strobe the CONNECT and READY signals may result in internal signals wider or narrower than expected due to sampling as the signals are changing. ANSI X3.183, HIPPI-PH, specifies that the CONNECT and READY signals are asserted, and deasserted, for a minimum of four HIPPI-PH CLOCK periods.

If the HIPPI Destination has a faster CLOCK than the HIPPI-SC implementation, and the Destination sends long sequences of minimum width READY signals, i.e., continuous back-to-back eight-CLOCK-period READY indications, then there is the possibility of losing one in 5000 READY indications. Lost READY indications should not occur unless both conditions are met, i.e., a pathological case. HIPPI-PH, clause 5.3.6, describes the possibility of lost READY indications. HIPPI-PH does not require Sources to accept more than 63 READY indications, but HIPPI-SC implementations that can accept up to 255 READY indications without loss are desirable. The onset of the first lost READY indication can be delayed through various means, including but not limited to:

– a design that tolerates receiving a few short READY indications before dropping an output READY indication, or;

– using a flag bit or special states to indicate that the forwarded READY indications are behind and need to catch up, when able, e.g., when receiving more than four contiguous CLOCK periods of READY asserted or READY deasserted, or;

– using a counter, counting up for each READY indication received, and counting down for each READY indication forwarded.

# Annex C
(informative)

# Bibliography

The following documents are the basis for assignment of specific logical addresses for certain network services:

[1] RFC 1042, *Standard for the transmission of IP datagrams over IEEE 802 networks*. (Provides the general techniques that the Internet Protocol uses to build media packet headers on IEEE 802 (IS 8802) networks.)

[2] RFC 1374, *IP and ARP on HIPPI.* (Describes a technique whereby Internet Protocol equipped hosts using a HIPPI compliant interface may discover the media address of a destination host with a known IP address.)

[3] RFC 1112, *Host extensions for IP multicasting*. (Provides a technique whereby network and transport layer Internet protocol applications may use the multicasting capabilities defined for IS 8802 networks.

[4] RFC 1131, *OSPF specification*. (Describes the open shortest path first IP protocol which permits network layer IP routers to discover the best route to remote IP addresses and networks.)

[5] ISO/IEC 9542:1988, *Telecommunications and information exchange between systems – End system to intermediate system routeing exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)*

[6] ISO/IEC 10589:1992, *Telecommunications and information exchange between systems – Intermediate system to intermediate system intra-domain routeing exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)*

[7] ANSI/IEEE 802.1D-1990, *Media access control (MAC) bridges,* (Specifies the operation of transparent bridges between IEEE 802 conferment networks.)