INTERNATIONAL STANDARD **ISO/IEC 9318-3**

Draft edition
1994-00-00

DRAFT ANNEX

Information technology - Intelligent Peripheral Interface

Part 3:
Device generic command set for magnetic and optical disk drives

Preliminary Document

This is a working document and may be reproduced for the purposes of evaluation and review.

ANSI X3T9.3 Technical Editor:

Horst L Truestedt
IBM Corporation
Dept 2C3 / Bldg 114-1
3605 Highway 52 North
Rochester, MN 55901-7829
USA

Ph: 507-253-4101
Fx: 507-253-1000
In: truested@rchvmx.vnet.ibm.com

Reference number
ISO/IEC 9318-3 : 1994 (WD)
Version: 10Jan94

Annex A

(normative)

Overview: the IPI-3 command sets on various physical interfaces

This section provides information on how the IPI-3 command set operates on the various physical interfaces as follows:

Annex A.1 - Intelligent Peripheral Interface

Annex A.2 - High Performance Parallel Interface

Annex A.3 - Fibre Channel

A.1 Mapping to the Intelligent Peripheral Interface

A.1.1 Control of facilities by the master

The master performs Facility Selection and issues Individual commands. This is a master-oriented environment that typifies the Operation of most of today's host system I/O interfaces. The master prioritizes tasks and dispenses them to the various peripherals attached to it to optimize the system performance from the master's perspective.

Since the master controls all activity on the interface, it shall poll both the slave and facility interrupts to maximize the efficiency of data transfers. The master-oriented environment is typified by the following features of the interface:

- command execution sequenced by the master;
- individual commands;
- selection to the facility;
- master polling of both slaves and facilities;
- Bus Control established by the master;
- command queuing done by the master.

A.1.2 Shared Control of Facilities

In this environment, a slave has functional control of the facilities, but the master has explicit control over some aspects of facility management. Some tasks, such as command queuing, may be left to the discretion of the slave, but the master may choose to sequence the flow of data between specific facilities based on its own algorithms.

This environment is typified by the following features of the interface:

- command execution between facilities sequenced by the master;
- queued commands;
- selection to the facility;
- master polling of both slaves and facilities;
- Bus Control established by the master;
- command queuing done by the slave.

A.1.3 Control of Facilities by the Slave

In this environment, the slave has functional control of the facilities, and the master has limited control (if any) over facilvity management. The slave has the intelligence necessary to control operations on behalf of the master.

This is a slave-oriented environment because of the degree of control the slave exercises over system performance and optimization. This environment is typified by the following features of the interface:

- command execution between facilities sequenced by the slave;
- queued commands;
- selection to the slave;
- polling of the slave by the master;
- Bus Control established by the master or (optionally) by the slave;
- command queuing done by the slave.

A.1.4 Transfers on the Physical Interface

A Bus Exchange at the Physical Interface requires a Bus Control sequence and its associated Ending Status sequence to frame the Information Transfer of Commands, Responses, and data.

Before data is actually transferred, the slave activates the Class 2 Interrupt to inform the master that it is ready to transfer data.

In an operation between two different slaves, the master shall designate a dominant slave and a subservient slave. The dominant slave assumes the role of the master for the purpose of initiating Information Transfers to carry out slave-to-slave operations.

A.1.4.1 Interrupts

Physical Interface interrupts are used by the slave to request service from the master.

The master may poll any or all of the interrupt classes by setting the appropriate bits in the Request Interrupts byte of the Physical Interface. It is the responsibility of the master to prioritize the interrupts in the event that more than one slave is interrupting.

Optionally, the master may poll the interrupts from the facilities attached to a slave by the Request Facilities Interrupts byte of the Physical Interface if this feature is supported by the slave.

The Physical Interface ATTENTION IN signal is a logical OR of the interrupts from all but the addressee selected on the Physical interface. The interrupts may be enabled and disabled from generating Attention.

The interrupt classes for slaves and facilities are defined in descending order of priority. Slaves shall present interrupts in order of their priority, except when the response stack is full of lower order interrupt responses (i.e., if a Class 2 could be presented because data is available in a buffer, but the stack is full of Class 1 responses because the master has not requested them, the slave shall not respond to a poll for Class 2 interrupts).

An interrupt is cleared when the condition that caused it to be presented is no longer present.

The three classes of interrupts are as follows:

a) Class 3 (Critical Status Pending). This interrupt shall be used to alert the master to events or conditions existing in a slave or facility that require immediate attention from the master and that are not associated with command completion. An Asynchronous response shall be used to alert the master, and the conditions initiating same are defined by the Major Status, associated Substatus, and Extended Substatus (if any).

b) Class 2 (Transfer Status Pending). This interrupt shall be used by slaves to inform the master that data is ready to be transferred. A Transfer Notification to identify the command involved shall be generated by the slave unless the master is selecting facilities and executing Individual commands; or a Paused transfer implies the Transfer Notification.

c) Class 1 (Status Pending). This interrupt shall be used to identify command completion responses, successful or otherwise, of commands issued by the master, and to inform the master of Asynchronous responses that are not critical. The Major Status and associated Substatus, if any, identify either the conditions under which the command terminated or the cause of the Asynchronous response.

   An optional use of this interrupt is to report that the slave, which was previously reported Busy during a selection sequence, is no longer Busy. When the interrupt is used in this manner, the master is responsible for recognizing that there is not necessarily a Command Completion or Asynchronous Response packet already in the slave. If there is no Response packet pending in the slave, the interrupt shall be cleared by a successful selection sequence.

A.1.4.2 Response types

The response types do not directly correspond to the three interrupt classes. An Asynchronous response may be Class 3 if critical (e.g., power failure warning), or Class 1 if not critical (e.g., transition from Ready to Not Ready).

A.1.4.2.1 Command completion response

This response shall be generated by the slave when a command has been completed, unless the response has been disabled by Attributes being set to "Inhibit Operation Response on Success."

The Class 1 interrupt is activated to inform the master that a Command Completion Response is available.

A.1.4.2.2 Transfer Notification Response (optional)

This response shall be generated by a slave capable of stacking Individual or Queued commands. It is used to identify the command with which the data transfer to follow is associated.

Transfer Notification Responses are not required under the following conditions:

- When commands are Individual, and the command was preceded by a facility selection
- For continuation of paused data transfers

The Class 2 Interrupt is activated to inform the master that a Transfer Notification Response is available.

A.1.4.2.3 Asynchronous response

This response shall be generated by the slave to advise the master of an unanticipated event not associated with command completion.

Either the Class 3 or Class 1 Interrupt is activated to inform the master that this response is available (depending upon whether or not it is considered critical by the slave).

A.1.4.2.4 Imbedded data response (optional)

This response shall be generated by the slave to send small amounts of data in a parameter field of the Response packet. No more than 254 bytes can be transferred in this manner.

The objective of this response is to permit peripherals with very low transfer rates and small transfer needs to gather data in a manner that does not interfere with high-performance peripherals. This response shall not terminate the command, and many of them can be received in answer to a single command.

The Class 2 Interrupt is activated to inform the master that an Imbedded Data Response is available.

A.1.4.3 Response handling

When there is more than one Response packet to be transferred by the slave, it shall transfer Response packets to the master in the order of interrupt priority. Within an interrupt class, the responses shall be presented in order of command completion.

If facility selection precedes the request for a Response packet, the slave shall only transfer the highest priority response among those for the selected facility.

The master shall properly handle any Response packet that it receives, even if it is not necessarily the response type that it expected.

The master should not attempt to initiate the transfer of a Response packet before interrupts indicate the availability of a response ready to transfer. If the slave has no Response packet to transfer upon receiving a request for one, it shall terminate the attempted Information Transfer (without transferring any information) and not post the Successful bit in Slave Status.

When a slave that supports Control of Bus at the Physical Interface is given control, it may elect to transfer a Response packet or transfer data.

A.1.5 Physical interface error recovery considerations

Errors detected by the Physical Interface protocol require recovery action be taken via a Device-Generic protocol. Recovery procedures depend on the packet type (Command, Data, or Response), and on whether or not the error is detected prior to the Slave Status byte being presented.

A.1.5.1 Recovery from unsuccessful Slave Status byte

The Slave Status byte contains the Successful Information Transfer bit, which is set to 0 if any errors are detected by the slave prior to transmission of the byte. This includes not posting Successful if the slave recognized bad parity on the Master Status byte.

NOTE — The slave does not change its Slave Status byte contents based on the contents of the Master Status byte.

Following a command transfer, if the Success bit in either the Master or the Slave Status byte is set to 0, the slave shall ignore the command. The master shall retransmit the command.

Following a data transfer, if the Success bit in either the Master or the Slave Status byte is set to 0, the slave shall terminate the command in progress. A response indicating the failure shall be generated.

Following a response transfer, if the Success bit in either the Master or the Slave Status byte is set to 0, the slave shall retain the response in its buffer and attempt re-transmittal under the master's control. The slave shall assert the appropriate interrupt upon deselection.

A.1.5.2 Recovery from bad parity on the Slave Status byte

If the master detects a parity error or any other invalid condition on the Slave Status byte, the slave has no way to determine what has happened (as this is presented in the last state before deselection may occur).

Whenever an invalid condition is detected in the Slave Status byte following a command, data, or Transfer Notification packet transfer, the master shall use an ABORT command or a selective reset (either Logical Interface Reset or Slave Reset) to force the slave to discontinue execution of the command.

If an invalid condition is detected in the Slave Status byte following a Response packet other than Transfer Notification (typically Command Completion), the master shall retry execution of the entire command because the slave is not aware of its failure, and has released the buffer containing the Response packet.

NOTE — In the case of a slave that queues commands, the master may have to issue no more commands until it can match valid responses with all the issued commands in order to identify the one that was unsuccessful.

If an invalid condition is detected in the Slave Status byte following an Asynchronous Response packet, the master has no way of advising the slave that the Slave Status byte was not received correctly. Master-specific recovery procedures may choose whether or not to accept the contents of the Information Transfer. If it is a condition that keeps recurring, the slave shall generate another response.

NOTE — Under this circumstance the error recovery action cannot be certain to obtain the same information as was originally presented.

A.1.6 Operation sequences

Operation sequences of the logical interface are controlled by the master, since it establishes the Bus Exchanges that transfer information between the master and slave. Optionally, the master may give control of the bus to a slave; actual control of the Information Transfer at the Physical Interface would then be performed by the slave.

The sequence of actions taken to execute operations between master and slave(s) is done as a series of Information Transfers across the Physical Interface. Slave selection and deselection is done as necessary to communicate with the desired addressee.

A.1.6.1 Slave procedures

The following procedures are not necessarily supported by every slave. The slave's Attributes indicate those which are supported.

a) Facility Selection and Facility Interrupts. When the slave supports facility selection and interrupts, the master may make use of them to exert control of the facility. The use of the facility interrupts allows the master to determine the facility requiring service and the facility selection allows the master to direct an operation to the desired facility.

b) Slave Control. When the slave Control of Bus is supported (Bus Acknowledge byte), the master may give control of the subsequent Information Transfer to the slave to allow it to determine whether to transfer a Response packet or data. The slave shall not set the Operation Out bits in the Bus Acknowledge byte (i.e., attempt to cause a command to be transferred).

c) Queued Facility. When the slave's Attributes indicate that a facility can operate in a queued environment, the master may issue more than one command for a facility.

A.1.6.2 Basic steps

The basic steps performed by the master in carrying out operations are

a) transmit Command packet to slave;
b) poll interrupts;
c) receive response from slave;
d) transfer data between master and slave.

These steps shall be as described in A.1.6.2.1 through A.1.6.2.4.

A.1.6.2.1 Transmit Command packet to slave

The master establishes a Bus Exchange to transfer a single Command packet to the selected slave. The number of commands that can be stacked at an addressee is defined in the Attributes.

Commands consist of variable length transfers, and some slaves may terminate the transfer based on the Packet Length defined in the first two bytes. If slaves require master termination of transfers, the length of the transfer is decided by the master.

If the Master Status byte in the Ending Status sequence indicates an unsuccessful transfer, then the slave shall ignore the received command. No Response packet shall be generated.

If the Slave Status byte in the Ending Status sequence indicates a successful transfer, the master need not retain the entire Command packet.

A.1.6.2.2 Poll interrupts

The master polls interrupts to know when to perform data transfers or receive Responses. The polling of a specific interrupt from each of the slaves is performed by the Request Interrupt sequence of the Physical Interface. The presence of an interrupt in any of the slaves is typically provided by the ATTENTION IN signal at the Physical Interface. The master can choose to mask out interrupts by class from setting Attention (via the ATTENTION CONTROL command).

Even if a master has masked out all interrupts from setting Attention, it is still necessary to recognize that unsolicited Attentions could occur (e.g., if an attached slave executed a Power On reset, the mask information would be lost, and an Attention caused by powering on would be generated when its drivers were enabled).

If facility interrupts are supported by the slave, the master may poll facilities attached to a slave after receiving an interrupt from the slave.

NOTE — The slave generates interrupts on behalf of the attached facility or facilities.

A.1.6.2.3 Receive Response packet from slave

Upon receiving an interrupt from a slave, the master establishes a Bus Exchange to transfer the slave's Response. There is one exception to this action by the master: when the interrupt is Class 2 and the master is executing in a non-queued environment with facility selection.

The master may give control of the bus to the slave (if this function is supported) so that the slave can decide whether to transfer a Response packet or to transfer data.

If the Master Status in the Ending Status sequence indicates a successful transfer, the slave releases the Response packet and deactivates the interrupt; otherwise, it retains the packet until successfully transferred.

The Response packet may indicate

a) the successful or unsuccessful completion of a command;
b) a notice that data is to be transferred for a particular command;
c) the occurrence of an asynchronous event.

A.1.6.2.4 Transfer of data between master and slave

After receiving a Transfer Notification [or when there is a Class 2 interrupt in a non-queued environment with facility selection (see A.1.2.2.3)], the master initiates a Bus Exchange to transfer data to or from the slave as required by the command identified in the response. (If the slave is given Control of Bus, it sets up the direction of transfer.)

The amount of data intended to be transferred shall be determined by the slave (because it generates SYNC IN at the Physical Interface). Whether or not the ending Master Status indicates a successful input transfer, the slave may release its input data buffer. If the Slave Status indicates a successful output transfer, the master may release its output data buffer.

If the master terminates a data transfer, the characteristics of the slave shall determine whether or not it is an unsuccessful transfer. To a fixed block class of device such as disk, the transfer would be considered unsuccessful unless the transfer was being Paused (optional feature) by the master. To a variable block class of device such as tape, master termination would normally be considered successful and a short PhysicalBlock would be written.

If not Paused by the master, the Command Completion response shall identify the termination to the master (whether or not the transfer was successful) as "Unexpected Master Status" in Machine Exception Substatus.

If the transfer requested by the command was completed successfully, unless overridden by Attributes, the slave shall generate a Class 1 interrupt and a Command Completion response. If the transfer is unsuccessful, the slave shall generate a Class 1 interrupt and report the appropriate status in the Command Completion Response.

A.1.6.3 Operation sequence examples

The four steps described in the preceding subclauses may be used in a number of ways, and can best be illustrated by a set of examples executing the same scenario in different implementations. In these examples, the data transfer for Facility 1 requires two transmissions because the slave paused (either unanticipated, or multiplexed) during transfer.

The examples attempt to define only major steps in the flow (e.g., deselection is implicit). To clarify direction of transfer, the terms "transmit" are used for transfers out to the slave, and "receive" is used for transfers in from the slave. The term "recognize" is used for polling because the master has a choice of alternatives when polling (between slaves, facilities and masking).

A.1.6.3.1 Example of facility selection and individual commands

In this example, transfer commands are sent to facilities 0, 1 on slave 0.

a) Select Slave 0/Facility 0 and transmit command packet
b) Select Slave 0/Facility 1 and transmit Command packet
c) Poll interrupts and recognize Class 2 Interrupt from Slave 0
d) Poll Slave 0 and recognize Class 2 Interrupt from Facility 1
e) Select Slave 0/Facility 1 and continue to perform data transfer until paused by Slave 0
f) Poll interrupts and recognize Class 2 Interrupt from Slave 0
g) Poll Slave 0 and recognize Class 2 Interrupt from Facility 0
h) Select Slave 0/Facility 0 and perform data transfer until terminated by Slave 0
i) Poll interrupts and recognize Class 1 Interrupt from Slave 0
j) Poll Slave 0 and recognize Class 1 Interrupt from Facility 0
k) Select Slave 0/Facility 0 and receive Command Completion Response with "Successful" Status
l) Poll interrupts and recognize Class 2 Interrupt from Slave 0
m) Poll Slave 0 and recognize Class 2 Interrupt from Facility 1
n) Select Slave 0/Facility 1 and perform data transfer until terminated by Slave 0
o) Poll interrupts and recognize Class 1 Interrupt from Slave 0
p) Poll Slave 0 and recognize Class 1 Interrupt from Facility 1
q) Select Slave 0/Facility 1 and receive Command Completion Response with "Successful" Status

A.1.6.3.2 Example of facility selection and queued commands

In this example, Transfer commands are sent to Facilities 0, 1 on Slave 0.

a) Select Slave 0/Facility 0 and transmit Command packet
b) Select Slave 0/Facility 1 and transmit Command packet
c) Poll interrupts and recognize Class 2 Interrupt from Slave 0
d) Poll Slave 0 and recognize Class 2 Interrupt from Facility 1
e) Select Slave 0/Facility 1 and receive Transfer Notification identifying command
f) Perform data transfer until paused by Slave 0
g) Poll interrupts and recognize Class 2 Interrupt from Slave 0
h) Poll Slave 0 and recognize Class 2 Interrupt from Facility 0
i) Select Slave 0/Facility 0 and receive Transfer Notification identifying command
j) Perform data transfer until terminated by Slave 0
k) Poll interrupts and recognize Class 1 Interrupt from Slave 0
l) Poll Slave 0 and recognize Class 1 Interrupt from Facility 0
m) Select Slave 0/Facility 0 and receive Command Completion Response with "Successful" Status
n) Poll interrupts and recognize Class 2 Interrupt from Slave 0
o) Poll Slave 0 and recognize Class 2 Interrupt from Facility 1
p) Select Slave 0/Facility 1 and receive Transfer Notification identifying command
q) Perform data transfer until terminated by Slave 0
r) Poll interrupts and recognize Class 1 Interrupt from Slave 0
s) Poll Slave 0 and recognize Class 1 Interrupt from Facility 1
t) Select Slave 0/Facility 1 and receive Command Completion Response with "Successful" Status

NOTE — Applications that implement multiple commands per facility in a non-ordered sequence, although not included in this example, require the use of Transfer Notification Packets. This example represents the same scenario as the other examples in this clause.

A.1.6.3.3 Example of slave selection.

In this example, Transfer commands are sent to Facilities 0, 1 on Slave 0.

a) Select Slave 0 and transmit Command packet for Facility 0
b) Remain selected and transmit Command packet for Facility 1
c) Poll interrupts and recognize Class 2 Interrupt from Slave 0
d) Select Slave 0 and receive Transfer Notification identifying command for Facility 1
e) Perform data transfer until paused by Slave 0
f) Poll interrupts and recognize Class 2 Interrupt from Slave 0
g) Select Slave 0 and receive Transfer Notification identifying command for Facility 0
h) Perform data transfer until terminated by Slave 0
i) Poll interrupts and recognize Class 1 Interrupt from Slave 0
j) Select Slave 0 and receive Command Completion Response for Facility 0 with "Successful" Status
k) Poll interrupts and recognize Class 2 Interrupt from Slave 0
l) Select Slave 0 and receive Transfer Notification identifying command for Facility 1
m) Perform data transfer until terminated by Slave 0
n) Poll interrupts and recognize Class 1 Interrupt from Slave 0
o) Select Slave 0 and receive Command Completion Response for Facility 1 with "Successful" Status

A.1.6.3.4 Example of slave selection and slave control of Bus

In this example, Transfer commands are sent to Facilities 0 and 1 on Slaves 0 and 1, respectively. This introduces implicit Transfer Notification (i.e., only one data transfer is paused at a slave, so the Transfer Notification can be implied when the transfer is continued).

a) Select Slave 0 and transmit Command packet for Facility 0
b) Select Slave 1 and transmit Command packet for Facility 1
c) Poll interrupts and recognize Class 2 Interrupt from Slave 0
d) Select Slave 0 and with Bus Control give Control of Bus to slave
e) Receive Bus Acknowledge byte indicating Response (Operation In)
f) Receive Transfer Notification identifying command for Facility 0
g) Master sets Bus Control (Data Direction)
h) Perform data transfer until paused by Slave 0
i) Poll interrupts and recognize Class 2 Interrupt from Slave 1
j) Select Slave 1 and with Bus Control give Control of Bus to slave
k) Receive Bus Acknowledge byte indicating Response (Operation In)
l) Receive Transfer Notification identifying command for Facility 1
m) Master sets Bus Control (Data Direction)
n) Perform data transfer until terminated by Slave 1
o) Poll interrupts and recognize Class 2 Interrupt from Slave 0
p) Select Slave 0 and with Bus Control give Control to slave
q) Receive Bus Acknowledge byte with implicit Transfer Notification and Data Direction
r) Perform data transfer until terminated by Slave 0
s) Poll interrupts and recognize Class 1 Interrupt from Slave 0
t) Select Slave 0 and with Bus Control give Control of Bus to slave
u) Receive Bus Acknowledge byte indicating Response (Operation In)
v) Receive Command Completion Response for Facility 0 with "Successful" Status
w) Poll interrupts and recognize Class 1 Interrupt from Slave 1
x) Select Slave 1 and with Bus Control give Control of Bus to slave
y) Receive Bus Acknowledge byte indicating Response (Operation In)
z) Receive Command Completion Response for Facility 1 with "Successful" Status

A.1.7 Multiplexed data transfers (optional)

Different computer system architectures will utilize the IPI, and a wide range of products may be intermixed on a single cable. There are occasions in which a transfer may be terminated for either an unanticipated reason or a deliberate event. This is known as a "pause".

NOTE — Command, and Response packet transfers cannot be paused — only data transfers may be paused and continued.

It is necessary that a master be able to control some of the ways in which different products from different vendors coexist. One important area is in ensuring equitable use of the bandwidth. A deliberate event to pause a transfer occurs when a master wishes to prevent a particular peripheral from dominating the use of the interface during Information Transfers.

In an application in which the master requests large data transfers, it is preferred that the master be able to predict the use of the interface. The master can accomplish this by multiplexing transfers between addressees on a predictable basis. The master may define (in Attributes) the maximum number of bytes to be transferred in any one data transfer (a burst), even though any single command may itself define considerably more.

A command for a 20Kbytes data transfer to a slave with a defined 8Kbytes maximum burst size would require three data transfers in order to complete. The slave would pause after each 8Kbytes burst and complete after the remainder of 4K was transferred.

The different ways of multiplexing transfers are defined in Attributes. The various combinations shall be as shown in table 1.

Table 1 — Different Ways of Multiplexing Transfers

Who Paused	Who Continued	Attribute Settings ID/Octet/Bit (Pause)	Attribute Settings ID (Other)
M	M	6B/4/5	
S	M	6B/5/5, 6B/5/4	66, 67
S	S		6B/2/4, 6B/3/4

A.1.7.1 Physical interface Pause and Continue (optional)

The Physical Interface provides the ability to Pause and Continue transfers from both the master and the slave. The master and the slave may use this function to interrupt a continuous data transfer. The slave shall transfer data in bursts no longer than those defined or set in Attributes.

The master cannot indicate any time dependency to the slave, only whether or not the transfer is paused. However, the slave can indicate to the master some idea of the expected time period that the transfer shall be paused. This is done through use of the Time-Dependent Operation bit in the Slave Status byte.

The meaning of the bits shall be as follows:

Pause	Time-Dependent Operation	
0	0	No Delay
1	0	Short Delay
1	1	Long Delay
0	1	No More Data

If for some reason both the master and slave present the Pause bit in their respective ending status bytes, the slave Pause shall override the master Pause.

In the following descriptions (A.1.7.2 to A.1.7.9), the term P/TDO refers to the setting of the Pause and Time-Dependent Operation bits in the Slave Status byte.

A.1.7.2 Slave pause and master Continue

The master controls the slaves that receive service. The master shall use Attributes to establish the size of transfer, defined as a burst, to be multiplexed. During data transfers, when the burst defined by the master has been transmitted, the slave shall terminate. If information is immediately available to resume transfer with another burst, it shall so indicate by P/TDO=00. This condition can only occur on multiplexed burst boundaries.

If a slave Pauses with no information remaining in its buffer to be transferred, but it anticipates only a short delay before information shall be available, it shall set P/TDO=10. If, however, the delay is expected to exceed the value defined in Attributes, it shall set P/TDO=11.

If it is deselected, the slave shall generate a Class 2 interrupt when it has more information to transfer (immediately when P/TDO=00).

When a slave transferring information pauses at the end of a burst, the master may choose to deselect and poll the other slaves, or remain selected and continue the transfer.

To continue a paused data transfer, the master shall re-select (if necessary), and resumption of the data transfer may or may not be implicit, based on the mode of interface operation.

The slave shall always have a Transfer Notification packet prepared (if they are being used) so that if the master issues a Response request, the slave can respond with same.

A.1.7.2.1 Implicit continue

If the master is executing Individual commands with Facility Selection, it can reissue the same Bus Control byte as for the initial transfer.

If the master is executing Queued commands to a slave that can have more than one paused transfer, the master has to receive a Transfer Notification Response packet to know which one is being continued. To do so, the master may issue the Bus Control byte for a Response, or permit the slave Control of Bus and the slave shall set up the data transfer of a Response.

A.1.7.2.2 Explicit continue

If the master is executing Queued commands to a slave, but only one transfer can be paused per slave, the master may reissue the same Bus Control Octet as for the initial data transfer.

Alternatively, the master may provide the slave Control of Bus, and the slave shall respond with the Data, Direction, and Control of Bus Accepted bits set in the Bus Acknowledge byte. The explicit continue prevents a slave, that can have more than one paused transfer and has been given Control of Bus, from starting a second paused transfer.

NOTE — The paused slave is locked to the same transfer command until it is either successfully or unsuccessfully completed. It is possible for a master to issue commands or request responses from a paused slave, but based on internal design a slave may or may not be able to comply. The only way in which a master can free a paused slave is to continue the data transfer or abort the transfer command that is Paused.

A.1.7.3 Slave control of Pause and Continue

A slave may choose to pause data transfer due to an unanticipated delay encountered (e.g., repositioning, error correction). If so, the slave shall terminate the transfer and set P/TDO=11. The master can deselect the slave and await a Class 2 interrupt to advise that transfer can be continued.

If the master had deselected, when the slave is ready to continue the transfer, the slave shall generate a Class 2 interrupt. The master shall re-select and set the Control of Bus bit in the Bus Control byte. If the master did not deselect, the master can remain in a loop presenting the Bus Control byte, and being rejected by the slave until the slave is ready to continue.

If the master used the Control of Bus bit, the slave shall set the Data, Direction, and Control of Bus Accepted bits in the Bus Acknowledge byte, and transfer shall begin after the master asserts the MASTER OUT signal.

A.1.7.4 Master control of Pause and Continue

When a master encounters a temporary interruption, sufficient to stop a data transfer, it shall initiate termination of the transfer and set the Pause bit in the Master Status byte. The master may choose to deselect or remain selected to the slave.

To continue a Paused data transfer, the master shall re-select (if necessary), along with the same settings in the Bus Control byte used for the original command, then assert the MASTER OUT signal to restart the data transfer.

The slave shall not generate a Class 2 interrupt, but if it encounters any error while Paused, it may generate either a Class 3 or a Class 1 interrupt.

A.1.7.5 Uses of multiplexing

The three basic uses that are made of multiplexing between addressees are described in A.1.3.5.1 through A.1.3.5.3.

A.1.7.5.1 One paused transfer per facility

When Facility Selection is used by the master to issue Individual commands, a slave may have up to 16 concurrent data transfers being multiplexed. It is the master's responsibility to manage the multiplexing of information via selection and re-selection of the facility.

A.1.7.5.2 One paused transfer per slave

When Slave Selection is used by the master, a slave may be able to multiplex only one transfer (defined by Attributes). Continuing a paused transfer does not require the use of a Transfer Notification (unless enabled by Attributes) since the master can reissue the same Bus Control byte as for the initial transfer to indicate an implicit Transfer Notification.

A.1.7.5.3 Multiple paused transfers per slave

When Slave Selection is used by the master, a slave may be able to multiplex more than one transfer. This application requires the use of Transfer Notification packets unless a master remains selected. Upon selection, the slave shall present a Transfer Notification packet to advise the master which paused data transfer is going to be continued.

A.1.7.6 Anticipated pause

The master uses Attributes to advise the slave of the burst size to be transferred during a multiplexed transfer.

After the slave has terminated a burst data transfer, it posts Slave Status with the Successful and the requisite P/TDO bits.

a) P/TDO=00Immediate Continuation Possible. On Read, next burst is already buffered and ready to transfer. On Write, the buffer is free to accept next burst from master.

b) P/TDO=10Continuation Possible after Short Delay. The slave expects a delay of less than the period defined by the master in Attributes.

c) P/TDO=11Continuation Possible after Long Delay. The slave expects a delay longer than the period defined by the master in Attributes.

d) P/TDO=01No More Data. The entire data transfer is now complete.

The Time-Dependent delay posted by the slave is an indication, and not a guarantee, because many factors are involved (e.g., a Short Delay could actually be a long one if an unforeseen error condition occurred).

A.1.7.6.1 Master stays selected

If the master chooses to stay selected to the addressee, it continues by issuing a Bus Control byte with the same settings as on the original. If the slave cannot continue, it shall reject the Bus Control byte and use encoded Slave Status to advise why.

A.1.7.6.2 Master deselects

If the master deselects, the addressee shall assert its Class 2 Interrupt (and Attention if not overridden by ATTENTION CONTROL) when it is capable of continuing the burst transfer.

The actions to be taken are a function of the application (see A.1.3.2.1 and A.1.3.2.2).

A.1.7.7 Unanticipated pauses by the slave

If the slave encounters a condition that causes interruption of a data transfer (before reaching a burst boundary), it shall terminate the transfer and advise P/TDO=1x in the Slave Status byte (continuation possible after Delay).

NOTE — P/TDO=00 is invalid, because if it could continue immediately, it would not have paused in the first place.

The master may choose to remain selected or not. The same actions that are defined in A.1.3.6 for anticipated pauses shall be followed by the slave and the master in order to continue the transfer.

A.1.7.8 Unanticipated pauses by the master

The master cannot pause more than one data transfer to the slave. The next operation to the same addressee has to be the continuation of the paused data transfer.

If a slave cannot accept a master-initiated Pause, it shall set the Not Successful bit in the Slave Status byte.

NOTE — Close examination of a vendor's specification is recommended because there are many conditions possible that could cause unpredictable results, or require extensive error recovery.

A.1.7.9 Multiplexed transfer mode identification

The different ways of multiplexing transfers are defined in Attributes. The various combinations shall be as shown in table 1.

Table A.1 — Different Ways of Multiplexing Transfers

Who Paused	Who Continued	Attribute Settings ID/Octet/Bit (Pause)	ID (Other)
M	M	6B/4/5	
S	M	6B/5/5, 6B/5/4	66, 67
S	S		6B/2/4, 6B/3/4

A.1.8 Basic Command packet in Double Octet Mode

Clause 5 shows the Basic Command packet for Single Octet Mode. Figure A.1 shows the structure of this packet in Double Octet Mode.

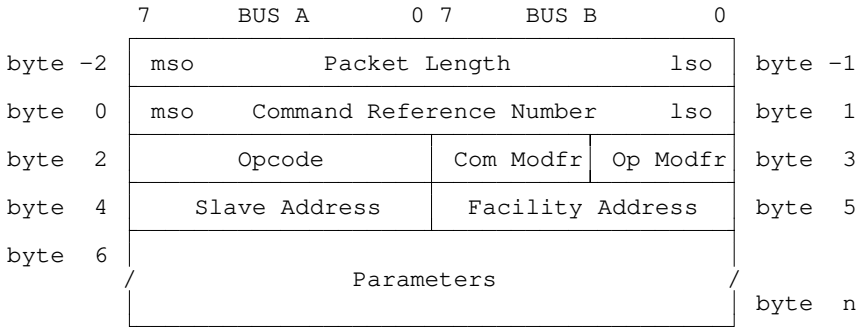


Figure A.1 — Basic Command packet in DOM Representation

If two parameters were present, in double byte mode, the first of five bytes (Parameter Length + Parameter ID + a three-byte field), and the second of three bytes (Parameter Length + Parameter ID + a one byte field), would be an 18-byte transfer, and the parameter list would appear as shown in figure 15.

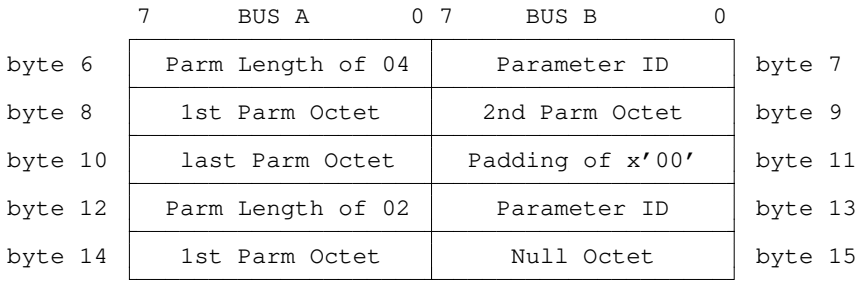


Figure A.2 — Parameter List for 18-Octet Transfer

A.2 Mapping to High Performance Parallel Interface

A.2.1 Introduction

This subclause defines how the Operations described in clause 4 shall be performed using the High Performance Parallel Interface (HIPPI) as a physical interface. This "mapping" does not require any changes to the Packet formats defined in clause 5.

A.2.1.1 HIPPI Characteristics

HIPPI supports a protocol for low-latency and variable size packet transfers. Once a connection has been established, multiple bursts may be transferred without a round-trip cable delay.

Signaling and control sequences are kept simple to allow average transfer rates for large transfers to approach the peak transfer rate. Look-ahead flow control is used to enable these same high data rates over long distances.

The HIPPI-PH provides the following physical interface:

— data rate of 100-200 MB/sec
— point-to-point
— 25 meter
— parallel copper

A.2.1.2 Configuration characteristics

HIPPI is a simplex channel, capable of data transfer in one direction only. The use of it in this standard requires two HIPPI channels: one for commands and write-data, the other for responses and read-data.

NOTE — for special applications, it is possible to use one HIPPI channel (i.e., a transfer command to a Frame Buffer may be set up to require no return path), but for the usual read/write Operations, two HIPPI channels are required.

The topologies targeted by this standard include the following:

Point-to-point: two end-nodes are connected (each node may be a master, a slave, or both a master and a slave).

Switched: multiple end-nodes are connected by using a switch (or switches) through which a Connection is established (i.e., for any given Connection, the identity of each end-node is statically implied).

NOTE — The use of switches is not described in this document. For further information on switches, see ANSI X3.222-1993.

A.2.2 Normative references

The following standards contain provisions that, through reference in this text, constitute provisions of this part of ISO/IEC 9318. At the time of publication, the editions shown were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 9318 are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

ANSI X3.183-1991 — *Information technology — High Performance Parallel Interface — PH: Physical*

ANSI X3.210-1992 — *Information technology — High Performance Parallel Interface — FP: Framing Protocol*

ANSI X3.222-1993 — *Information technology — High Performance Parallel Interface — SC: Switch Control*

A.2.3 Definitions

To use the IPI-3 command sets on HIPPI, the following definitions apply.

A.2.3.1 CCI: Connection Control Information used to make an HIPPI connection (see ANSI X3.222-1993).

A.2.3.2 connection: The condition of the HIPPI-PH when data transfers from Source to Destination are possible (see ANSI X3.222-1993).

A.2.3.3 DESTINATION: The equipment that receives a packet.

A.2.3.4 D1: Data set 1 is that portion of the HIPPI ULP Data Set that contains control information (either IPI-3 commands or responses).

A.2.3.5 D2: Data set 2 is that portion of the HIPPI ULP Data Set that contains one IPI-3 Burst (as defined in Attributes) of read or write data.

A.2.3.6 link: A link consists of two unidirectional physical HIPPI channels, one outbound and the other inbound.

A.2.3.7 node: Each end of a link.

A.2.3.8 SOURCE: The equipment that transmits a packet.

A.2.3.9 ULP Data Set: The Upper Level Protocol "payload" or data set which is transferred in an HIPPI packet. This data set consists of one of the following:

— one IPI-3 Command packet.
— one IPI-3 Response packet.
— one IPI-3 Command packet and its associated data.
— one IPI-3 Response packet and its associated data.
— more than one IPI-3 Command packet.
— more than one IPI-3 Response packet.

A.2.4 Using IPI-3 with HIPPI

A.2.4.1 IPI-3 on HIPPI flow diagram

Figure A.1 shows the data flow from IPI-3 master/slaves/peer over the HIPPI-FP and HIPPI-PH:

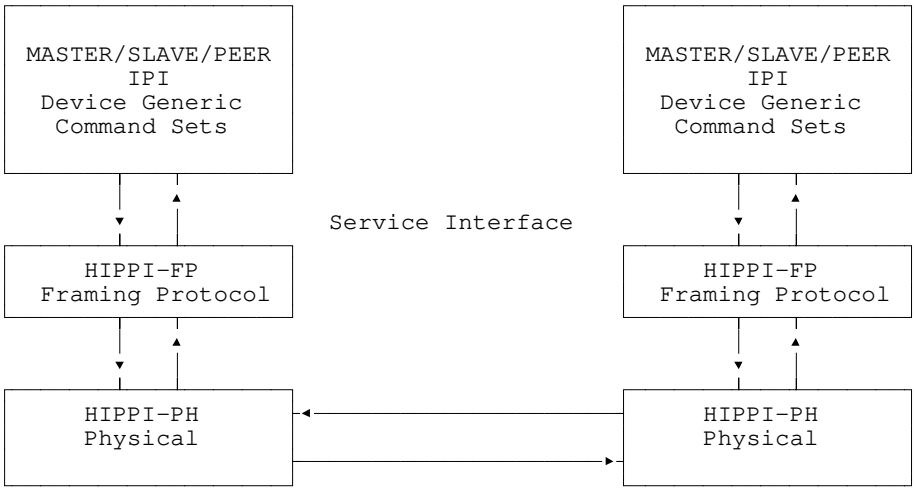


Figure A.1 — IPI-3 on HIPPI Flow Diagram

A.2.4.2 HIPPI-FP packet format

The basic HIPPI-FP packet, as defined in ANSI X3.210, HIPPI-FP, is used for reference. Figure A.2 shows the following components:

Header
ULP Data Set
D1 - IPI-3 Command/Response packet(s).
D2 - IPI-3 Burst of Read/Write Data.
Fill

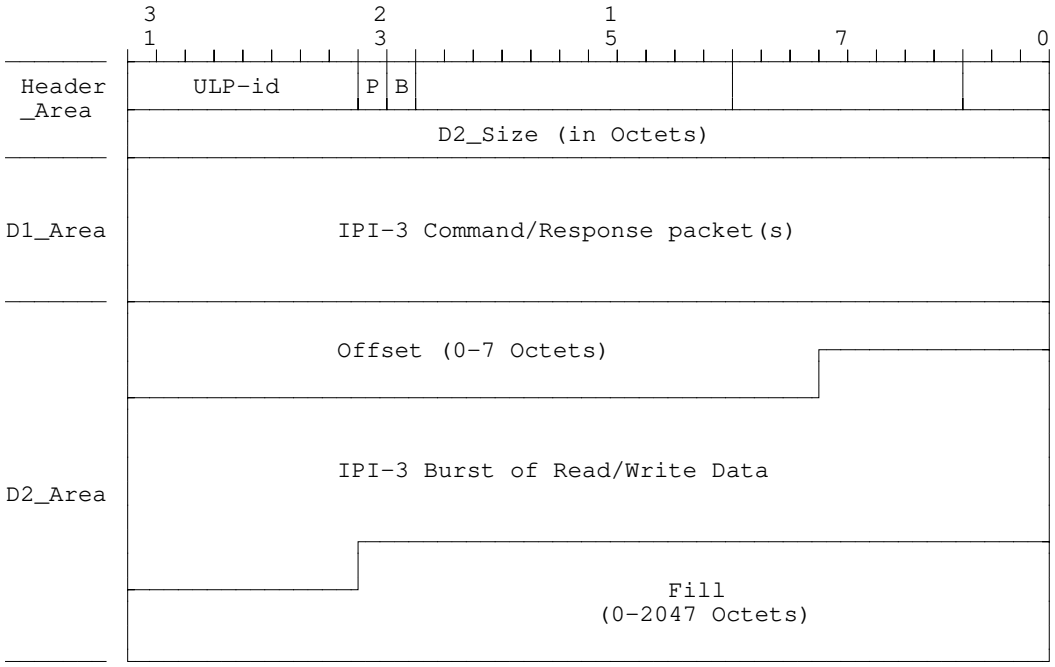


Figure A.2 — HIPPI-FP packet

All HIPPI-FP packets have the same general structure (i.e., they consist minimally of a D1_Area which contains all of the IPI-3 Command or Response packet(s) followed by D2). The number of words in D2 which contains the IPI-3 Data, is either implicitly specified by the command/response type or explicitly specified by a count field in the HIPPI-FP Header_Area.

D1 shall consist of one of the following:

- an IPI-3 Command (Command packet)
- an IPI-3 Response (Response packet)
- an IPI-3 Peer header
- an IPI-3 Write command followed by one or more bursts of write data.
- an IPI-3 Transfer Notification followed by one or more bursts of data.
- IPI-3 Commands (Command packets)
- IPI-3 Responses (Response packets)

D1 has a maximum length of 1016 bytes and shall end on a 64-bit boundary.

D2 consists of one IPI-3 Burst the size of which is defined in Attributes parameter 6F.

A.2.4.3 Multiple Command/Response packets in D1

D1 may consist of more than one IPI-3 Command or Response packet. The packets are self-defining since the Packet Length (first two bytes of each IPI-3 packet as defined in 5.2.1.1) defines the length (including all parameters). IPI-3 requires that all parameters use a padding of x'00' if the parameter would end on an odd byte. The Packet Length shall begin on a 32-bit boundary. To achieve this alignment, Null Octets are used (see 5.2 and 5.3). Figures A.5 and A.6 are examples to show these allignments.

NOTE — The order of these commands do not imply execution order (see 4.1.1.3).

A.2.5 HIPPI-FP Connections

A.2.5.1 Connection handling

This standard supports two methods of connection between a Source and Destination. The Attributes parameter 90 conveys which protocol is used.

NOTE — this is only a recommendation to the lower HIPPI levels (i.e., the lower levels may choose to ignore the request).

The example given shows an entire Operation (i.e., command, data, completion):

Protocol A - connect/disconnect: A connection is established at the beginning of each packet and remains in effect until all of the packet has been transferred. This method is useful when information is multiplexed between multiple end-nodes.

For the master to start an Operation, it tells its Source to initiate a connection and to transfer an HIPPI-FP packet (which contains the IPI-3 command and optionally, data). After the entire packet has been transferred, the Source shall initiate a disconnect sequence.

When the slave is ready to transfer to the master, it tells its Source to initiate a connection and to transfer an HIPPI-FP packet (which may contain a Transfer Notification and optionally data, or a Completion Response packet). After the entire packet has been transferred, the Source shall initiate a disconnect sequence.

Protocol B - always connected: Once a connection has been established, the connection remains until specifically broken. This may be a configuration where there is only one master and one slave.

For the master to start an Operation, it tells its Source to initiate a connection and to transfer an HIPPI-FP packet (which contains the IPI-3 command and optionally, data). After the entire packet has been transferred, the Source shall stay connected.

When the slave is ready to transfer to the master, it tells its Source to initiate a connection and to transfer an HIPPI-FP packet (which may contain a Transfer Notification and optionally data, or a Completion Response packet). After the entire packet has been transferred, the Source shall stay connected.

A.2.5.2 Rejected connections

If HIPPI cannot complete a connection to the destination (i.e., the Source detects a "Rejected Connection Sequence") it responds with a "Rejected" connection to the calling entity. The calling entity may retry the connection request.

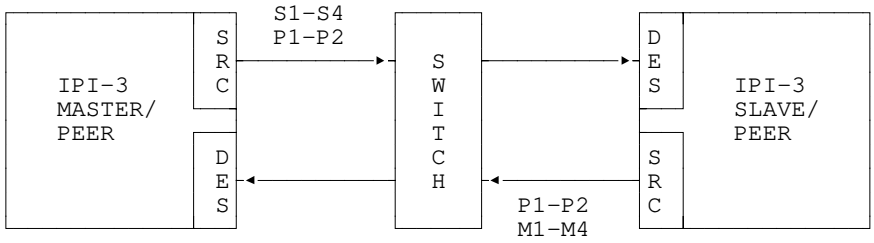
A.2.5.3 CCI handling

When the master connects with the slave, the Connection Control Information (CCI as defined in ANSI X3.222) is captured and saved by the slave. Thereafter, any commands sent during this connection are "tagged" with the contents of this CCI.

Whenever the slave connects to the master (e.g., send a Response packet), the CCI information (which was previously saved) shall be used according to Attribute parameter 90 (see A.2.6.2).

A.2.5.4 IPI-3 transfers on HIPPI

In the following discussion, a new HIPPI-FP packet is denoted whenever D1 is mentioned. Figure A.5 shows a model of transferring Information Units (IUs) between an IPI-3 Master (M), a Slave (S), and between Peers (P). IUs are either an IPI-3 Command packet, IPI-3 Response packet, IPI-3 Burst, or IPI-3 Peer header. The IPI-3 Burst size is controlled by Attributes parameter 6F (see 6.3.4.15).



NOTE — SWITCH is shown to suggest that multiple connections are possible. However, as far as the IPI-3 master or slave is concerned, the switch is transparent. "SRC" is an HIPPI Source and "DES" is an HIPPI destination.

Figure A.5 — Data transfer model

A.2.5.4.1 Information Units from master to slave

The IUs that are received by a slave are summarized in Table A.1.

Table A.1 — Information Units to a slave

IU NAME	OPERATION PHASE	D1_DATA_SET	D2_DATA_SET	M or O
		PAYLOAD	PAYLOAD	
S1	Command	One IPI-3 Command packet	-	M
S2	Data	IPI-3 Transfer Notification for Write data	One IPI-3 Burst	M
S3	Command	Multiple IPI-3 Command packets	-	O
S4	Command /Data	One IPI-3 Command packet for a Write command	One IPI-3 Burst	O

Key:
D1_DATA_SET — IPI-3 Command packet(s)
D2_DATA_SET — IPI-3 Burst
M or O is Mandatory or Optional

A.2.5.4.1.1 One IPI-3 command IU — S1

This IU shall be the normal method of transferring one IPI-3 Command packet.

A.2.5.4.1.2 Transfer Notification and one IPI-3 Burst IU — S2

This IU shall be the normal method of transferring an IPI-3 Burst of write data. The Transfer Notification contains the first 6 bytes of the original write command to identify the following data.

A.2.5.4.1.3 Multiple IPI-3 commands IU — S3

This IU shall be used if multiple IPI-3 Command packets are supported via Attributes parameter 90. Each IPI-3 Command packet is self describing (i.e., the length field allows parsing of each command).

NOTE — The order of these commands do not imply execution order (see 4.1.1.3).

A.2.5.4.1.4 One IPI-3 command and data IU — S4

This IU shall be used to send one command and the first or only data Burst of this command as described by Attributes parameter 90. Subsequent data Bursts for this command use S2.

A.2.5.4.2 Information Units from slave to master

The IUs that shall be routed to a master are summarized in Table A.2.

Table A.2 — Information Units to a master

IU NAME	OPERATION PHASE	D1_DATA_SET	D2_DATA_SET	M or O
		PAYLOAD	PAYLOAD	
M1	Response	One IPI-3 Response packet	-	M
M2	Data	IPI-3 Transfer Notification Response for Write	-	M
M3	Data	IPI-3 Transfer Notification Response for Read	One IPI-3 Burst	M
M4	Response	Multiple IPI-3 Response packets	-	O

Key:
D1_DATA_SET — IPI-3 Response packet(s)
D2_DATA_SET — IPI-3 Burst
M or O is Mandatory or Optional

A.2.5.4.2.1 IPI-3 response IU — M1

This IU shall be the normal method to transfer one IPI-3 Response packet.

A.2.5.4.2.2 IPI-3 Transfer Notification Response for Write data IU — M2

This IU shall be the normal method to transfer TNRs that requests write data which is identified by the Command Reference number in the TNR.

A.2.5.4.2.3 IPI-3 Transfer Notification Response and Read data IU — M3

This IU shall be the normal method to transfer an IPI-3 Burst of Read data. The IPI-3 TNR associates the data and the IPI-3 command.

A.2.5.4.2.4 Multiple IPI-3 Response packets IU — M4

This IU shall be the normal method to transfer multiple Command Completion responses. Note the responses are processed in the order that they are received.

A.2.5.4.3 Information Units routed to an IPI-3 Peer

The IUs that shall be routed to an IPI-3 Peer are summarized in Table A.3.

Table A.3 — Information Units to a peer

IU NAME	OPERATION PHASE	D1_DATA_SET	D2_DATA_SET	M or O
		PAYLOAD	PAYLOAD	
P1	Data	IPI-3 Peer header	-	M
P2	Data	IPI-3 Peer header	One IPI-3 Burst	M

Key:
D1_DATA_SET — IPI-3 Peer header
D2_DATA_SET — IPI-3 Burst
M or O is Mandatory or Optional

A.2.5.4.3.1 IPI-3 Peer header IU — P1

This IU shall be the normal method of requesting data from another IPI-3 Peer. The header identifies the requested data to the receiving IPI-3 Peer.

A.2.5.4.3.2 IPI-3 Peer header and Data Burst IU — P2

This IU shall be the normal method to transfer bursts of data to an IPI-3 Peer. The header identifies the following data to the receiving IPI-3 Peer.

A.2.5.5 Simple Read example

Transferring data from the slave to the master (S indicates to a slave; M indicates to a master):

Command: S1 - an HIPPI-FP packet containing an IPI-3 Read command in D1.

Data: the following IPI-3 packet(s) is sent if there is data to be transferred:

M3 - D1 containing the Transfer Notification (TN) to relate the command and the following data.
- D2 containing one IPI-3 Burst of the requested Read data.

If more than one IPI-3 Burst is to be transferred, M3 is repeated until all requested data have been transferred.

Completion: M1 - an HIPPI-FP packet containing an IPI-3 Completion Response packet in D1 to inform the master of completion status.

A.2.5.6 Simple Write example

Transferring data from the master to the slave (S indicates to a slave; M indicates to a master):

Command: S1 - an HIPPI-FP packet containing an IPI-3 Write command in D1.

Data: the following IPI-3 packet(s) is sent if there is data to be transferred:

M2 - D1 containing the Transfer Notification Response (TNR) to request the data associated with a command as identified by the CRN.

S2 - D1 containing the Transfer Notification (TN) for the requested data.
- D2 containing one IPI-3 Burst of the requested Write data.

If more than one IPI-3 Burst is to be transferred, S2 is repeated until all requested data have been transferred.

Completion: M1 - an HIPPI-FP packet containing an IPI-3 Completion Response packet in D1 is sent to inform the master of completion status.

If write immediate is supported (as defined by Attributes parameter 90), D2 may be included with one IPI-3 Burst in the command phase as shown in S4. Subsequent IPI-3 Bursts are then transferred via S2 in the data phase.

A.2.5.7 Complex Read/Write example

The HIPPI-FP D1 from the master may contain more than one command if the slave allows command queuing and multiple commands and responses in D1 (as defined by Attribute parameters 6F and 90, respectively).

For a multiple Read and Write commands in D1, the following sequence shall be used (S indicates to a slave; M indicates to a master):

Command: S3 - an HIPPI-FP packet containing IPI-3 Read/Write commands in D1. When multiple commands are used, no D2 shall be included in this phase (i.e., write immediate is not supported).

Data: the following IPI-3 packet(s) is sent if there is data to be transferred:

For Read operations:

M3 - D1 containing the Transfer Notification Response (TNR) for the following data.
- D2 containing one IPI-3 Burst of the requested Read data.

If more than one IPI-3 Burst is to be transferred, M3 is repeated until all data have been transferred.

For Write operations:

M2 - D1 containing the Transfer Notification Response (TNR) to request from the master the data for the command as identified by the CRN.

S2 - D1 containing the Transfer Notification (TN) for the requested data.
- D2 containing one IPI-3 Burst of the requested Write data.

If more than one IPI-3 Burst is to be transferred, S2 is repeated until all requested data have been transferred.

Completion: M1 - an HIPPI-FP packet containing an IPI-3 Completion Response packet for each command in D1 is sent to inform the master of completion status. Note that if IORS had been used on all commands except the last command and no errors were encountered, only one IPI-3 Completion Response packet (for the last command) could be sent in D1.

M4 - an HIPPI-FP packet containing all IPI-3 Completion Response packet for each command in D1 is sent to inform the master of completion status for all commands.

A.2.5.8 Error conditions

If either the master or slave HIPPI-PH Source cannot connect to the destination, it shall respond to the requesting entity with a "Reject" connection indication.

If the slave destination HIPPI-PH detects an error during HIPPI packet transfers and an error occurs in D1, the packet shall be discarded. If the error occurred in D2, a Response packet shall be sent to the master indicating a Machine Exception with Physical Interface Check substatus.

If the master destination HIPPI-PH detects an error during HIPPI Packet transfers, the contents of the HIPPI Packet are discarded and an indication is passed to the controlling entity. The ULP may then retry the Operation, however, it must recognize that any following HIPPI Packets may need to be discarded until the ULP has reached a known state.

A.2.6 Use of IPI-3 constructs

A.2.6.1 IPI-3 packets

All IPI-3 packets shall start on 32-bit boundaries and all IPI-3 parameters shall start on 16-bit boundaries.

NOTE — D1 appears the same whether a 32-bit or 64-bit HIPPI is used.

A.2.6.1.1 Multiple Command packets in D1

Figure A.3 shows an example of two IPI-3 commands with a padding of x'00' to place the next parameter on an even byte boundary.

Octet 0	Octet 1		Octet 2	Octet 3
Packet Length			Command Reference Number	
Opcode	Com Modfr	Op Modfr	Slave Address	Facility Address
Parm Length of 04	Parameter ID		1st Parm Octet	2nd Parm Octet
last Parm Octet	Padding of x'00'		Parm Length of 02	Parameter ID
1st Parm Octet	Null Octet		Null Octet	Null Octet
Packet Length			Command Reference Number	
Opcode	Com Modfr	Op Modfr	Slave Address	Facility Address
Parm Length of 03	Parameter ID		1st Parm Octet	last Parm Octet

Figure A.3 — Example of two IPI-3 commands in D1

A.2.6.1.2 Multiple Response packets in D1

Figure A.4 shows an example of two IPI-3 responses with the Null Octet to place the next parameter on an even byte boundary.

Octet 0	Octet 1		Octet 2	Octet 3
Packet Length			Command Reference Number	
Opcode	Com Modfr	Op Modfr	Slave Address	Facility Address
Major Status	Resp Type	Maj Stat	Parm Length of 04	Parameter ID
1st Parm Octet	2nd Parm Octet		last Parm Octet	Padding of x'00'
Parm Length of 02	Parameter ID		1st Parm Octet	Null Octet
Packet Length			Command Reference Number	
Opcode	Com Modfr	Op Modfr	Slave Address	Facility Address
Parm Length of 03	Parameter ID		1st Parm Octet	last Parm Octet

Figure A.4 — Example of two IPI-3 responses in D1

A.2.6.2 Command Reference Numbers

This mapping of IPI-3 assumes that the Command Reference Number is qualified by the Source Address from which it was received. The Source Address uniquely identifies the source of the command to a single process.

A.2.6.3 IPI-3 command sets and features

The command sets described in the IPI-3 for disk, tape, and communications are used to instruct the slave on the action that the master desires. This includes simple read and write commands, attributes, and diagnostic commands. Since the connection (once established) is a point-to-point connection between the master and the slave, the normal IPI-3 address byte for the slave is not used for addressing (see clause 5).

Three features of IPI-3 are:

Transfer Notification (TN): TN shall be used in every HIPPI Packet. This information is contained in D1 to associate the command and data. In the case of a Write operation, the master will respond with the write command followed by the write data. In the case of a Read operation, the slave will follow TN with the read data.

Inhibit Operation Response on Success (IORS): This attribute provides the option for the slave to report to the master that it can inhibit Completion Response packets and for the master to tell the slave that it should operate in this mode (i.e., both master and slave communicate that they can and want to operate without certain Response packets with each other).

If the slave is operating in the IORS mode, no Completion Response packet shall be sent to the master after successfully completing a command and the slave may start processing the next command (if there are any).

If an error is detected by the slave, the slave shall send a Completion Response packet describing the error after the last IPI-3 Burst has been transmitted.

Imbedded Data: Not supported over HIPPI.

Because HIPPI Packets are not self-identifying, parameter 3A shall be used with Transfer Notification Responses (TNRs) if there are more than one IPI-3 Burst to be transferred. Parameter 3A can either contain a Burst count or the relative address of the data and is used to identify the IPI-3 Burst that is being transferred.

A.2.6.4 Attributes

The following IPI-3 bits and fields of Attributes parameters are not used with HIPPI and are reserved:

Table A.4 — Attributes parameter 64, 66, 6B, 6E, and 6F

@	LTH	ID	OCTET	X/b	DEF	ATTRIBUTES PARAMETERS
S	15	64				PHYSICAL INTERFACE ATTRIBUTES PARAMETER
S	n+1	66				SLAVE CONFIGURATION (BIT SIGNIFICANT)
			01	3		Odd Octet Transfers
				2		Master Termination of Commands Required
			02	3		Master Throttling of Data Streaming
				1		Data Streaming Data Transfers
				0		Interlock Data Transfers
S	09	6B				MASKS OF OCTETS SUPPORTED
B	n+1	6E				SLAVE RECONFIGURATION (BIT SIGNIFICANT)
			01	4		Physical Selection of Synonyms
				5		Inhibit Unanticipated Pauses
				1		Data Streaming Data Transfers
				0		Interlock Data Transfers
			03	7		Response on P-Busy to Not P-Busy Suppressed
B	n+1	6F				SLAVE RECONFIGURATION (FIELDS)
			0B-0C			# of Unanswered SYNC INs during Data Streaming
			17-1A			Data Streaming SYNC timeout (µs)
			23-26			Implicit Release Delay (µs)

This mapping of IPI-3 is supported by a new Attributes parameter, which is defined below. Note that the parameter is intended to indicate the capabilities of the slave, and to allow the master to disable certain characteristics of the slave on a configuration basis. It is not intended that the master use this parameter to manage the Operation of the HIPPI port of the slave on a command by command basis. The slave only retains one Attribute parameter 90 at any time (i.e., if there are multiple masters, each may send a new Attributes parameter 90). Note that for peer-to-peer transfers, the IPI-3 Attributes are not used.

Table A.5 — Attributes parameter 90

@	LTH	ID	OCTET	X/b	DEF	ATTRIBUTES PARAMETERS
B	n+1	90				HIPPI-PH and HIPPI-FP (RE)CONFIGURATION
			01			Connection Usage
				7		Permanent Connection
				6-1		reserved
				0		Return CCI as defined in HIPPI-SC
			02			reserved
			03			HIPPI-FP Features supported
				7		D2 to start on HIPPI-FP Burst Boundary
				6-0		reserved
			04			Mapping features supported
				7		Write cmd+data in same HIPPI-FP packet supptd
				6		Multiple commands/responses in D1 supported
				5-0		reserved
			05-08			CCI to which Asynchronous Responses are sent
			09-0C			CCI master-to-slave | Repeated
			0D-10			Return CCI slave-to-master | as many
			n-7:4			CCI master-to-slave | times as
			n-3:n			Return CCI slave-to-master | needed

The following bytes shall be included in this parameter:

a) Permanent connection bit. If this bit set to 1, the service primitive to HIPPI-FP will include "Keep_Connection".

b) Return CCI as defined in HIPPI-SC bit. HIPPI-SC defines two CCI types. If this bit is set to 1, the end-node may use the received CCI and change the direction bit. If this bit is set to 0, the CCI pairs in byte 09-n in this parameter are used.

c) D2 to start on HIPPI Burst Boundary bit. If this bit is set to 1, then the service primitive to HIPPI-FP shall include "Start_D2_on_Burst_Boundary". This allows the IPI-3 Command/Response packets and data to be easily separated by the hardware.

NOTE — This bit implies that the slave sends and receives HIPPI packets in the same manner.

d) Write cmd+data in same HIPPI-FP packet supptd bit. If this bit is set to 1, it indicates that the IPI-3 Master may send a single IPI-3 Write Command packet in D1 with the write data in D2.

e) Multiple Command/Response packets in D1 supported bit. If this bit is set to 1, it indicates that the IPI-3 Master may transfer multiple IPI-3 Command packets in D1 and the IPI-3 Slave may transfer multiple IPI-3 Response packets in D1.

f) CCI to which Asynchronous Responses are sent. This CCI is the unique master in the system to which all IPI-3 Asynchronous Response packets are sent.

g) CCI master-to-slave and CCI slave-to-master. These words are used to find the corresponding return CCI if byte 1, bit 0 is set to 0. The CCI must uniquely identify one IPI-3 Master, one IPI-3 Slave, or one IPI-3 Peer.

NOTE — these are CCI pairs and may be repeated as many times as is necessary to describe all end-nodes with which this slave may communicate.

A.3 Mapping to Fibre Channel

A.3.1 Introduction

This subclause defines how the Operations defined in clause 4 shall be performed using Fibre Channel (FC) as a physical interface. This "mapping" does not require any changes to the IPI-3 Packet formats defined in clause 5.

Contained within this subclause are the following:

a) Information Units (IUs) which are the FC payloads.

b) Use of various Fibre Channel constructs in the context of IPI-3 Operations.

Taken together, these items define the mapping of the IPI-3 to Fibre Channel.

A.3.2 Normative reference

The following standard contains provisions that, through reference in this text, constitute provisions of this part of ISO/IEC 9318. At the time of publication, the editions shown were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 9318 are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

ANSI X3.230-199x - *Information technology — Fiber Channel Physical and Signalling Interface (FC-PH)*

A.3.3 Definitions

For the purposes of this subclause, the following definitions shall apply. The definitions included herein are in addition to those defined in 3.1, and specifically relate to the Fibre Channel physical interface. Where a definition is included here as well as in 3.1, the definition given here shall have precedence for this subclause only.

A.3.3.1 Information_Set_1: first portion of an Information Unit (may contain one IPI-3 Command, one IPI-3 Response, one IPI-3 Burst, or one IPI-3 Peer header).

A.3.3.2 Information_Set_2: second portion of an Information Unit (contains only one IPI-3 Burst).

A.3.3.3 IPI-3 destination: the FC end-node to which a payload transfer is addressed. This may be either an IPI-3 Master, IPI-3 Slave, or IPI-3 Peer.

A.3.3.4 IPI-3 source: the FC end-node that originates a payload transfer. This may be either an IPI-3 Master, IPI-3 Slave, or IPI-3 Peer.

A.3.3.5 Payload: one IPI-3 Command, one IPI-3 Response, one IPI-3 Burst, or one IPI-3 Peer header.

A.3.3.6 Source Address: a unique identifier for the master.

A.3.4 Using IPI-3 with FC

A.3.4.1 IPI-3 on FC flow diagram

Figure A.6 shows the data flow from IPI-3 master/slaves over the FC-PH:

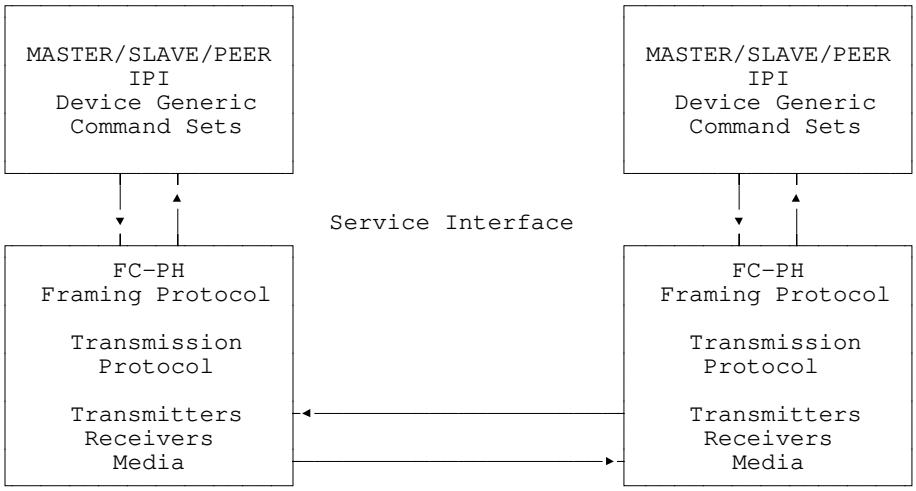


Figure A.6 — IPI-3 on FC-PH Flow Diagram

A.3.5 Information Units (IUs)

A.3.5.1 Usage

The IUs define the payloads carried over Fibre Channel to facilitate communication between a master and a slave. The IUs define not only the data which is transparent to Fibre Channel, but also the use of some Fibre Channel control information and constructs.

A.3.5.2 Format

The format of the IUs is described by the service interface of annex S in ANSI X3.230. This service interface defines a number of required and optional FC parameters that need to be supplied by an IPI-3 source or received by an IPI-3 destination. Some of these parameters are used in the same manner in all of the IUs, and for convenience they are defined in A.3.5.2.1. The parameters that differ between IUs are defined along with each IU.

A.3.5.2.1 FC common parameters

A.3.5.2.1.1 Common required FC parameters

Usage of the common required FC parameters shall be as follows:

a) Class of Service — the classes (1 or 2 only) which may be used to transport the IU

b) D_ID — the destination address identifier of the IU

c) S_ID — the source address identifier of the IU

d) Exchange reassembly — identifies if more than one N_Port is transmitting the data

e) Request_ID — identifies a specific IPI-3 Operation

NOTE — this provides a similar function to the Command Reference Number (CRN) in IPI-3.

A.3.5.2.1.2 Common optional FC parameters

A.3.5.2.1.2.1 Expiration/Security Header

The use of these parameters is beyond the scope of this document, and is both implementation and system dependent.

A.3.5.2.1.2.2 Network Header

The use of this parameter is beyond the scope of this document, and is both implementation and system dependent.

A.3.5.2.1.2.3 Association_Header

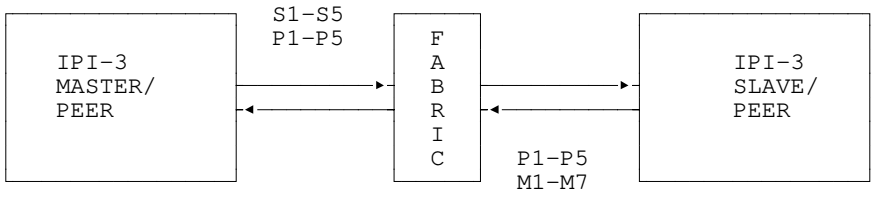
The use of this parameter is beyond the scope of this document, and is both implementation and system dependent.

A.3.5.2.1.2.4 Device_Header

Device_Headers are not used in this mapping.

A.3.5.3 IPI-3 transfers on FC

Figure A.7 shows a model of transferring Information Units (IUs) between an IPI-3 Master (M), a Slave (S), and between Peers (P). IUs are either an IPI-3 Command packet, IPI-3 Response packet, or IPI-3 Burst. The IPI-3 Burst size is controlled by Attributes parameter 6F (see 6.3.4.15).



NOTE — FABRIC is shown to indicate that multiple connections are possible. However, as far as the IPI-3 Master, IPI-3 Slave, or IPI-3 Peer is concerned, the FABRIC is transparent.

Figure A.7 — Data transfer model

A.3.5.3.1 Information Units from master to slave

The IUs that are received by a slave are summarized in Table A.6.

Table A.6 — Information Units to a slave

IU NAME	OPERATION PHASE	INFORMATION_SET_1 (I_S_1)		INFORMATION_SET_2 (I_S_2)		FML	SI	M or O
		CAT	PAYLOAD	CAT	PAYLOAD			
S1	Command	2	One Command packet	-	-	F	T	M
S2	Data	1	First or Intermediate IPI-3 Burst	-	-	M	H	M
S3	Data	1	Only or last IPI-3 Burst	-	-	M	T	M
S4	Command /Data	2	One Write Command packet	4	First IPI-3 Burst	F	H	O
S5	Command /Data	2	One Write Command packet	4	Only IPI-3 Burst	F	T	O

Key:
INFORMATION_SET CAT is the Information Category
INFORMATION_SET PAYLOAD is the IPI-3 construct
FML is First/Middle/Last IU of an Operation
SI is Sequence Initiative — Held or Transferred
M or O is Mandatory or Optional

A.3.5.3.1.1 Command IU — S1

This IU shall be the normal method of transferring commands.

A.3.5.3.1.2 Write first or intermediate IPI-3 Burst IU — S2

This IU shall be the normal method of transferring the first or intermediate IPI-3 Burst of Write data.

A.3.5.3.1.3 Write only or last IPI-3 Burst IU — S3

This IU shall be the normal method of transferring the only or last IPI-3 Burst of Write data.

A.3.5.3.1.4 Write command and IPI-3 Burst IU — S4

This IU shall provide an optional means of transferring a Write command and its associated first IPI-3 Burst as a single entity.

A.3.5.3.1.7 Write command and only IPI-3 Burst IU — S5

This IU shall provide an optional means of transferring a Write command and its associated IPI-3 Burst as a single entity.

A.3.5.3.2 Information Units routed slave to master

The IUs that shall be routed to a master are summarized in Table A.7.

Table A.7 — Information Units to a master

IU NAME	OPERATION PHASE	INFORMATION_SET_1 (I_S_1)		INFORMATION_SET_2 (I_S_2)		FML	SI	M or O
		CAT	PAYLOAD	CAT	PAYLOAD			
M1	Data	3	IPI-3 Transfer Notification Response for write	-	-	M	T	M
M2	Data	3	IPI-3 Transfer Notification Response for read	-	-	M	H	M
M3	Data	1	IPI-3 Burst	-	-	M	H	M
M4	Completion	3	One Command Completion Response packet	-	-	L	T	M
M5	Completion	2	One Asychronous Response packet	-	-	F & L	H	M
M6	Data	3	IPI-3 Transfer Notification Response for read	1	One IPI-3 Burst	M	H	O
M7	Data	1	Last IPI-3 Burst when IORS	-	-	L	T	O

Key:
INFORMATION_SET CAT is the Information Category
INFORMATION_SET PAYLOAD is the IPI-3 construct
FML is First/Middle/Last IU of an Operation
SI is Sequence Initiative — Held or Transferred
M or O is Mandatory or Optional

A.3.5.3.2.1 Transfer Notification Response for write IU — M1

This IU shall be the normal method to transfer TNRs for write-type commands.

A.3.5.3.2.2 Transfer Notification Response for read IU — M2

This IU shall be the normal method to transfer TNRs for read-type commands.

A.3.5.3.2.3 Read IPI-3 Burst IU — M3

This IU shall be the normal method to transfer bursts of Read data. When Relative Offset is supported, its initial value shall continue from the preceding IU of this Operation and shall continually increase, unless set otherwise by an Extent Offset parameter in the Read command.

A.3.5.3.2.4 Command completion IU — M4

This IU shall be the normal method of transferring Command Completion responses.

A.3.5.3.2.5 Asynchronous Response IU — M5

This IU shall be the normal method of Asynchronous Responses.

A.3.5.3.2.6 Transfer Notification Response and Read IPI-3 Burst IU — M6

This IU is used to transfer the TNR and one burst of Read data. When Relative Offset is supported, its initial value shall continue from the preceding IU of this Operation and shall continually increase, unless set otherwise by an Extent Offset parameter in the Read command.

A.3.5.3.2.7 Last IPI-3 Burst when IORS — M7

This IU is used when transferring the last IPI-3 Burst when IORS is used to suppress the Command Completion Response packet. When this IU is transferred, it indicates to the IPI-3 Master that this is the end of the Operation.

A.3.5.3.3 Information Units routed to an IPI-3 Peer

The IUs that shall be routed to an IPI-3 Peer are summarized in Table A.8.

Table A.8 — Information Units to an IPI-3 Peer

IU NAME	OPERATION PHASE	INFORMATION_SET_1 (I_S_1)		INFORMATION_SET_2 (I_S_2)		FML	SI	M or O
		CAT	PAYLOAD	CAT	PAYLOAD			
P1	Send Data	2	IPI-3 Peer header	-	-	F	H	M
P2	Send Data	1	One IPI-3 Burst	-	-	M	H	M
P3	Send Data	1	Last or only IPI-3 Burst	-	-	L	H	M
P4	Request Data	2	IPI-3 Peer header	-	-	F	T	M
P5	Send Data	2	IPI-3 Peer header	1	First or only IPI-3 Burst	F	H	O

Key:
INFORMATION_SET CAT is the Information Category
INFORMATION_SET PAYLOAD is the IPI-3 construct
FML is First/Middle/Last IU of an Operation
SI is Sequence Initiative — Held or Transferred
M or O is Mandatory or Optional

A.3.5.3.1 IPI-3 Peer header IU — P1

This IU shall be the method of transferring the IPI-3 Peer header when the FC-PH does not support multiple categories per sequence. The header identifies the following data transfer.

A.3.5.3.2 One IPI-3 Burst IU — P2

This IU shall be the normal method of transferring an IPI-3 Burst to an IPI-3 Peer.

A.3.5.3.3 Last or only IPI-3 Burst IU — P3

This IU shall be the normal method of transferring the last or only IPI-3 Burst of data to an IPI-3 Peer.

A.3.5.3.4 IPI-3 Peer header IU — P4

This IU shall be the normal method of requesting data from another IPI-3 Peer. The header identifies to the receiving IPI-3 Peer the requested data.

NOTE — Although this IU is optional, if the FC-PH supports multiple categories per sequence, this IU is the recommended data transfer.

A.3.5.3.5 IPI-3 Peer header and Data Burst IU — P5

This IU shall be the optional method of transferring data to an IPI-3 Peer. The header identifies the following data to the receiving IPI-3 Peer.

A.3.6 Use of Fibre Channel Constructs

A.3.6.1 Exchange

A Fibre Channel Exchange shall equal a single IPI-3 Operation between a master and a slave as defined in clause 4. All of the IUs that constitute an Operation shall be transferred in a single Exchange except for peer-to-peer transfers. A separate Fibre Channel Exchange shall be used for the peer transfer portion of the IPI-3 Operation.

A.3.6.2 Information categories

The following Information categories shall be supported:

1 solicited data
2 solicited control
3 unsolicited control
4 unsolicited data

A.3.6.3 Relative offset

Information categories solicited and unsolicited control shall not use Relative Offset.

Information categories solicited and unsolicited data in the first IU routed to an IPI-3 Slave shall transfer with an initial Relative Offset of zero and a Relative Offset that subsequently continually increases across all IUs.

Information category solicited data in the IUs routed to a master shall have an Initial Relative Offset defined by an Extent Offset Parameter (if the parameter is supplied in the IPI-3 Read Command packet). This parameter defines a non-continually increasing Relative Offset, if that is supported by the port which is the destination of the Read data.

If the source port of the read data does not support the creation of non-continually increasing Relative Offsets in a Sequence, then the command shall be rejected with Command Exception Status and an Invalid Parm parameter indicating the field in error.

If the IPI-3 Read Command packet does not include an Extent Offset parameter, then an initial Relative Offset of zero and a Relative offset that subsequently continually increases will also be used for Information Category solicited data.

A.3.6.4 Streaming

IUs within a single phase of an IPI-3 Operation may be streamed regardless of in-order support within the Fabric or in the communicating N_Ports.

A.3.7 Use of IPI-3 constructs

A.3.7.1 Transfer Notifications

IPI-3 Transfer Notifications Response packets are not mandatory when IPI-3 is mapped to Fibre Channel if no parameters would be included. Where Transfer Notifications are used, a Transfer Notification shall only be transferred in advance of the first IPI-3 Burst for the Operation. When the Transfer Notification contains no parameters, the X_ID of the received IU serves as a synonym for the Command Reference Number and no Transfer Notification is required. If there is no Transfer Notification to transfer the initiative to the other port, a null Information_Set_1 shall be used for this purpose.

A.3.7.2 Command Reference Numbers

This mapping of IPI-3 assumes that the Command Reference Number is qualified by the Source Address from which it was received. The Source Address uniquely identifies the source of the command to a single process. A combination of the S_ID, D_ID, and Process Associator may be used to establish the Source Address.

A.3.7.3 IPI-3 Bursts

The maximum size of an IPI-3 Burst is defined in Attribute parameter 6E. Separate maximum Data Burst sizes for Read and Write commands may be defined in Attributes parameter 81. When an Extent Offset parameter is used with a read command, the size of the read IPI-3 Bursts may be further restricted if the size of the maximum region of contiguous offset is less that the maximum burst size defined by Attribute parameter 6E.

A.3.7.4 Attributes

The following IPI-3 bits and fields of Attributes parameters are not used with Fibre Channel and are reserved (i.e., transmitted as zeros):

Table A.9 — Attributes parameters 64, 66, 6B, 6E, 6F

@	LTH	ID	BYTE	X/b	DEF	ATTRIBUTES PARAMETERS
S	15	64				PHYSICAL INTERFACE ATTRIBUTES PARAMETER
S	n+1	66				SLAVE CONFIGURATION (BIT SIGNIFICANT)
			01	3		Odd Byte Transfers
				2		Master Termination of Commands Required
			02	3		Master Throttling of Data Streaming
				1		Data Streaming Data Transfers
				0		Interlock Data Transfers
S	09	6B				MASKS OF BYTES SUPPORTED
B	n+1	6E				SLAVE RECONFIGURATION (BIT SIGNIFICANT)
			01	4		Physical Selection of Synonyms
				5		Inhibit Unanticipated Pauses
				1		Data Streaming Data Transfers
				0		Interlock Data Transfers
			03	7		Response on P-Busy to Not P-Busy Suppressed
B	n+1	6F				SLAVE RECONFIGURATION (FIELDS)
			0B-0C			# of Unanswered SYNC INs during Data Streaming
			17-1A			Data Streaming SYNC timeout (µs)
			23-26			Implicit Release Delay (µs)

IPI-3 is supported by a new Attribute parameter 91, which is defined below. The parameter is intended to indicate the capabilities of the slave, and to allow the master to disable certain characteristics of the Slave on a configuration basis. It is not intended that the master use this parameter to manage the Operation of the Fiber Channel port of the slave on a command by command basis. Note that the IPI-3 Attributes are not used in a peer-to-peer transfer.

Table A.10 — Attributes parameter 91

@	LTH	ID	BYTE	X/b	DEF	ATTRIBUTES PARAMETERS
B	n+1	91				FIBRE CHANNEL (RE)CONFIGURATION
			01			Class 1 usage
				7		Class 1 supported/enabled
				6		Permanent Connection
				5-0		reserved
			02			Class 2 usage
				7		Class 2 supported/enabled
				6-0		reserved
			03			reserved
			04			Relative Offset usage
				7		Single Relative Offset space supported for data in an Operation
				6		Initial Relative Offset for solicited data may be non-zero
				5-0		reserved
			05-12			Source Address to which Asynchronous Responses are to be sent

The following bytes shall be included in this parameter:

a) Class 1 supported / enabled bit: When this bit is set to 1, the IPI-3 Slave may use Class 1 for information transfers.

b) Permanent connection bit: When this bit is set to 1, the IPI-3 Slave may request a permanent connection.

c) Class 2 supported / enabled bit: When this bit is set to 1, the IPI-3 Slave may use Class 2 for information transfers.

d) Single Relative Offset space supported for data in an Operation: When this bit is set to 1, the Relative Offset shall start at zero and continually increase.

e) Initial Relative Offset for solicited data may be non-zero bit: When this bit is set to 1, the IPI-3 Slave may begin data transfers at any locations. This may be useful to transfer cache data and/or current position data that is not the beginning of the requested data space.

f) Source Address to which Asynchronous Responses are to be sent: This field identifies the Source Address to which all IPI-3 Asynchronous Response packets are sent.

A.3.7.5 Inhibit Operation Response on Success

When the master has set the Inhibit Operation Response on Success bit (Attributes parameter 6E), and an Operation is successful, the Operation shall be completed by the slave returning:

Write: Command Completion (M4) to the master, but Information_Set_1 shall be null. The transfer of this IU is necessary to transfer initiative to the port that originated the Exchange.

Read: Command Completion (M6) to the master, with Information_Set _1 containing the last IPI-3 Burst of the Operation.

Annex B

(normative)

IPI-3 Command striping

B.1 Introduction

Even though IPI, HIPPI, and FC provide data transfer rates of 100 MB/s to 200 MB/s, there continues to be a need for IPI-3 devices that have an aggregate bandwidth that exceeds that of a single physical channel. To address this need, IPI-3 Command striping is explained in this clause.

B.2 Striping model

By using IPI-3 Command striping, a master may issue IPI-3 Commands to a slave over multiple physical channels. The slave is responsible to reassemble the logical command stream and to execute the commands in the intended order. Figure B.1 illustrates a case of three-way striping, where each arrow represents an individual IPI-3 Command and its associated data, and the shaded connections represent the three physical channels. An important point to note is that this model proposes what has been called "command striping," as opposed to hardware striping. With command striping, each command, its associated data, and the response is carried entirely on a single physical channel. With FC striping, the lower logical layers implement the striping service; with hardware striping (as supported by 64-bit IPI or HIPPI), the hardware simply provides a wider path. In each of these alternate models, individual IPI-3 Commands and data use all of the physical channels concurrently.

Some advantages of command striping are:

- It can be implemented entirely at the IPI-3 level, and should operate equally well over any of the defined physical channels (IPI, HIPPI, FC, etc.).

- Since it is a logical level construct it does not necessarily require changes to master, slave, or switch fabric hardware, nor to the lower logical layers. For this reason, command striping may have a lower implementation cost.

- The striping can take advantage of varying numbers of available physical channels, and can potentially change its stripe width on the fly. For example, if multiple physical channels are available, the system can use all of them. However, if one or more of the channels become unavailable, due either to hardware failures or resource contention, the system can continue to operate (even if more slowly) by using the remaining channels.

Some disadvantages of command striping, relative to hardware striping are:

- Since commands and the associated data are carried entirely by one channel, start-up latencies are larger.

- The burden of implementing the striping service is shifted to the drivers that format the IPI-3 Command packets. This may lead to increased complexity and the need to replicate the striping support in each driver.

- More buffer space may be required in masters and slaves to de-skew commands and data.

B.3 Implementation requirements

In order to implement a reliable command striping service, two issues need to be addressed.

The first requirement is to provide a means for the slave to recognize the intended command order. This is especially important for physical layers, such as switched HIPPI, which have the potential for introducing significant amounts of skew between commands sent over different physical channels. This skew may prevent the intended command sequence from being inferred from the order in which commands are received.

The second requirement is to introduce the notion of a "master path group" so that commands received from any of a number of ports (e.g., Source CCI values in HIPPI) can be recognized as coming from the same master for the purpose of command sequencing. Command striping is not alone in needing this capability; it is also required for device assignment and command chaining when a single master may have multiple connections to a slave.

Meeting the first requirement mentioned above is accomplished by redefining the slave address field in the IPI-3 command header to allow it to provide command sequence information.

For unsequenced commands, which the slave is free to process in any order, this field would contain a value of 0 to 7. For IPI physical channels this is consistent with the existing use of this field for the hardware slave address. For HIPPI and FC physical channels, the mapping of IPI-3 does not require the use of this field and specifies that it should be zero.

For sequenced commands the field would contain a value of x'80' to x'FF'. This is the command succession number (modulo 128) biased by x'80'. Thus the first command in a sequence would have a value of x'80', the next x'81', and so on. After command x'FF', the next number would begin again with x'80'.

NOTE — this implies a maximum command window size of 128.

A special value of x'08' is defined to be an unsequenced command, and additionally causes the slave to reset its expected succession number to 0 (x'80' in the slave address field) as an aid in error recovery. Slave address/succession values of x'09' through x'7F' are reserved for future use.

For the commands in a logical stream, the succession number would increment by one (modulo 128) for each successive command. Commands received with unsequenced values (0-8) would be executed in a slave-determined order (maintaining incompatibility with existing implementations), while commands containing succession values (x'80'-x'FF') would be executed in strict numerical sequence.

A new slave attribute parameter is used to specify the maximum "window size" for command succession numbers. When the difference between the lowest command succession number expected and the largest actually received exceeds the window value, all the received commands will fail with a command exception error. This allows slaves to place an upper bound on the amount of buffer space required for de-skewing commands, and also protects against lost commands.

B.4 Existing example

To better appreciate the need for IPI-3 command striping, it may be instructive to look at an existing implementation that does not use them.

In this implementation, a high speed slave is driven by a pair of IPI channels. Without the use of command striping, a number of constraints have to be placed on the system.

The problem of command execution order may be sidestepped by requiring that successive commands arrive over alternating channels. In this way, a command sequence number (modulo 2) is implied by the port over which the command is received.

However, in a more flexible implementation (e.g., using switched HIPPI channels), these constraints would be burdensome. The requirement for strict port alternation eliminates the ability to work with reduced stripe widths, or to dynamically allocate channels

based on availability. Also, the mapping of ports (i.e., Source CCI values) to masters would probably require that a static configuration be loaded into the slave by some external means.

Annex C

(normative)

IPI-3 Third-Party Data Transfer Models

C.1 Introduction

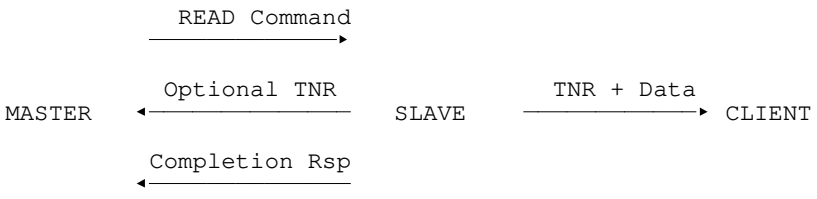
This clause describes several models that the IPI-3 command sets support to provide a third-party transfer capability. This facilitates the use of separate command and data channels. The basic model allows a master to send IPI-3 transfer commands to a slave and have the slave transfer data to a third party over a switched data path such as HIPPI or FC.

As shown in figure C.1, multiple Clients, File Servers, and Storage Subsystems may be connected with a control channel (commands and responses) and a data channel. Typical control channels may be IPI-PH, Ethernet, Token-Ring, and FDDI; typical data channels may be IPI-PH, HIPPI, and FC. Although, control channels may be separate, they may also be viewed as "logical" channels which reside on the same channel as the data.

Ports may be configured as IPI-3 Masters, IPI-3 Slaves, or IPI-3 Peer and each port may have one or more of these configurations at any time. The IPI-3 master sends IPI-3 commands and receives IPI-3 responses; the IPI-3 slave receives IPI-3 commands and sends IPI-3 responses; the IPI-3 peer sends or receives data only. The IPI-3 master and IPI-3 slave may also send and receive data.

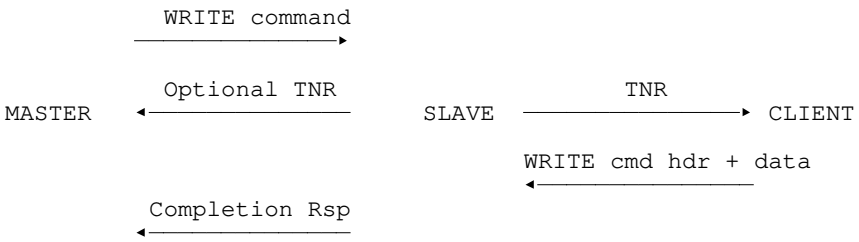
The Extended Port Mask parameter, is used to indicate the port to use for the data transfer, and the address of the third party.

The following is a scenario for a third-party Read Operation. The master is the sender of the IPI-3 commands, the slave is the device, and the client is the intended recipient of the data transfer. The communication between the master and the client in order to set up this transfer is outside the scope of this proposal. It is intended that the Transfer Notification parameter will be used to supply the client with information concerning the context of the data transfer.



The transfer begins with a READ command from the master to the slave requesting a third party transfer. The Extended Port Mask Parameters indicates to the slave which port to use for the transfer, and the address of the client. The slave sends the data to the client in a packet containing a transfer notification response followed by data. The master may optionally request that a transfer notification response be sent over the command path. The TNR back to the master will not contain any data. Once the transfer is complete, the slave sends a completion response back to the master.

The following scenario describes a third-party transfer for a Write Operation.



The transfer begins with an IPI-3 WRITE command from the master to the slave requesting a third party transfer. The slave sends a transfer notification response to the client when it is ready to transfer the data. The master can optionally request to receive the transfer notification response as well. The client responds to the TNR from the slave by sending back the WRITE command header from the original command (derived from the TNR with bit 7 of the command modifiers set) followed by the data. As in reads, the communication between the master and the client to set up the transfer is outside the scope of this proposal. Once the data transfer is complete, the slave sends a completion response back to the master.

C.1.1 Attributes

The Extended Port Mask parameter specifies a port to use for a third-party transfer and supplies the network address of the third-party client.

Table C.1 — Attributes parameter 43

@	LTH	ID	OCTET	x/b	DEF	ATTRIBUTES PARAMETER
M	n+1	43				EXTENDED PORT MASK PARAMETER
			01			WWN of physical ID
				7		All Addresses
				6-4		reserved
				3-0		Physical interface type
						0000 - HIPPI
						0001 - FC
						0010 - TCP
			02-09			Physical ID (address)
			0A			Facility Address
			0B- n			Port Mask

Common parameter 44 is used to transfer the "token" for IPI-3 Peer that is meaningful only to the peer entities.

Table C.2 — Peer header parameter 44

@	LTH	ID	OCTET	x/b	DEF	PEER HEADER PARAMETER
M	n+1	44				PEER HEADER PARAMETER
			01			Version number
			02			Flags
				7		Data requested
				6		Data present
				5		0 - Length fixed
						1 - Length negotiated
				4		End of transfer
				3-0		reserved
			03-08			reserved
			09-24			Transfer-ID
			25-32			Authentication
			33-48			Routing Information
			49-56			Offset
			57-64			Length

C.2 Third-party transfer models

C.2.1 Using IPI-3 COPY command

To be added...

C.2.2 Using a file server with IPI-3 Peer transfers

C.2.2.1 Storage subsystem to client data transfers

In this model, a client uses a high-level command across the control channel to request the file server to either get or put data onto a storage subsystem.

NOTE — this high-level language is outside the scope of this document, however, the high-level language could be the IPI-3 command set.

With the high-level language, the "token" or IPI-3 peer header (see parameter 44) is passed to the file server. The file server then becomes an IPI-3 master and sends the appropriate IPI-3 Read or Write command (including parameters 42 and 44) to the storage subsystem (IPI-3 slave). The IPI-3 slave executes the command as follows:

- if the command is an IPI-3 Read command — sets up the read command, connects to the client via parameter 42 (third-party address), and transfers the "token" (i.e., parameter 44 with bytes 1-64 only) followed by the read data.

- if the command is an IPI-3 Write command — sets up the write command, connects to the client via parameter 42 (third-party address), transfers the "token" (i.e., parameter 44 with bytes 1-64 only) to identify to the client the data that is being requested, receives the "token" from the client followed by the data to be written to the storage subsystem.

Once the IPI-3 command has been completed, the IPI-3 slave notifies the IPI-3 master (file server) with a Response packet (unless suppressed by Inhibit Operation Response on Success). The file server can then report to the client that the request has been completed.

NOTE — this high-level response is outside the scope of this document, however, the high-level language could be the IPI-3 command set.

C.2.2.2 Storage subsystem to storage subsystem

In this model, a client uses a high-level command across the control channel to request the file server to transfer data from one storage subsystem to another (see figure C.2).

NOTE — this high-level language is outside the scope of this document, however, the high-level language could be the IPI-3 command set.

With the high-level language, the "token" or IPI-3 peer header (see parameter 44) is passed to the file server. The file server then becomes an IPI-3 master and sends the appropriate IPI-3 Read command to one storage subsystem (IPI-3 slave 1) and the Write command to the other storage subsystem (IPI-3 slave 2). Included with the command to slave 1 is parameter 42 and to both slaves, parameter 44. The storage subsystems do the following:

- IPI-3 slave 1 sets up the read command and when the data is ready to be transferred, connects to the client (the second storage subsystem) via parameter 42 (third-party address), and transfers the "token" (i.e., parameter 44 with bytes 1-64 only) followed by the read data.

- IPI-3 slave 2 sets up its peer port with the appropriate "token"; prepares the device for write data; and waits for the data. When the "token" arrives at the peer port, the following data is written as had previously been specified in the IPI-3 Write command.

[The rest still to be completed — HLT]

Annex D

(informative)

Defect List Management

D.1 FORMAT command

Operation Mode		Parm 56, 57, or 5A attached?	Slave Action
Initial Format set?	Inhibit Realloc set?		
no	no	no	- Append contents of Suspect lists to corresponding Working lists; - Clear Suspect lists; - Insert new defects encountered during Format into Working Temporary list. Reallocate base on Working lists.
yes	no	no	- Clear both Suspect lists; - Clear Working Temporary list; - Initialize Working Permanent list with Manufacturer's defect list; - Insert new defects encountered during Format into Working Temporary list. Reallocate based on Working lists.
no	yes	no	Reject command.
yes	yes	no	- No modification of Suspect lists; - Clear Working Temporary list; - Initialize Working Permanent list with Manufacturer's defect list. Reallocate based on Working lists.

Operation Mode		Parm 56, 57, or 5A attached?	Slave Action
Initial Format set?	Inhibit Realloc set?		
no	no	yes	- Append parm list to Working Temporary list; - Append contents of Suspect lists to corresponding Working lists; - Clear Suspect lists; - Insert new defects encountered during Format into Working Temporary list. Reallocate based on Working lists.
yes	no	yes	- Clear both Suspect lists; - Initialize Working Temporary list with parm list; - Initialize Working Permanent list with Manufacturer's defect list; - Insert new defects encountered during Format into Working Temporary list. Reallocate base on Working lists.
no	yes	yes	Reject command.
yes	yes	yes	Reject command.

D.2 REALLOCATE command

Operation Mode Relocate set?	Parm 31, 3A, 50, or 55 attached?	Slave Action
no	no	- Reallocate based on Suspect lists; - Append contests on Suspect lists to corresponding Working lists; - Clear Suspect lists.
yes	no	- Reallocate based on Suspect lists; - Append contents of Suspect lists to corresponding Working list; - Clear Suspect lists.
no	yes	- Reallocate based on Parm list; - Append Parm list to Working Temporary list.
yes	yes	- Reallocate based on Parm list; - Append Parm list to Working Temporary list.

D.3 ALLOCATE RESTORE command

Operation Mode Relocate set?	Parm 31, 3A, 50, or 55 attached?	Slave Action
no	no	- Previously reallocated defects are discarded from the Working lists; - Reassign based on discarded defects.
yes	no	- Previously reallocated defects are discarded from the Working lists; - Restore based on discarded defects.
no	yes	- Selectively discard from Working lists based on Parm list; - Reassign based on Parm list.
yes	yes	- Selectively discard from Working lists based on Parm list; - Restore based on Parm list.

Annex E

(informative)

IPI-3 commands

This clause provides a representative list (not an exhaustive list) of commands that should be considered when implementing the IPI-3 command sets for magnetic and optical disk drives and tape drives.

E.1 Magnetic and optical disk commands

The following disk commands are a useful subset of the IPI-3 command set.

Basic commands are:

Attributes, Read, Write

For support of dual-ports, the following additional commands are:

Report Addressee Status, Port Address, Path Control

To support enhanced features, the following additional commands are:

Report Discontinuity, Format, Reallocate, Allocate Restore, Load Slave IML

For diagnostics, the following additional commands are:

NOP, Attention Control, Read from Buffer, Write to Buffer, Perform Slave Diagnostics, Perform Facility Diagnostics

E.2 Tape commands

Basic commands are:

Attributes, Read, Write, Operating Mode, Space Block/File Mark, Position Control, Record Position, Erase

For support of dual-ports, the following additional commands are:

Report Addressee Status, Port Address, Path Control

To support buffered tape drives, the following additional commands are:

Operator Display, Report Position, Read from Buffer

For diagnostics, the following additional commands are:

NOP, Attention Control, Write to Buffer, Perform Slave Diagnostics, Perform Facility Diagnostics

Annex F

(informative)

Bibliography

F.1 Informative references

The following parts of ISO/IEC 9318 are not essential for the completion of the requirements of this part of ISO/IEC 9318. They are intended to be used solely for explanation or clarification.

ISO/IEC 9318-2:1990 - *Information technology — Intelligent Peripheral Interface*
- *Part 2: Device-Specific Command Set for Magnetic Disk Drives*

ISO/IEC 9318-4:1990 - *Information technology — Intelligent Peripheral Interface*
- *Part 4: Device-Generic Command Set for Magnetic Tapes*

F.2 Equivalent ANSI standards

The following American National Standards Institute (ANSI) standards make up the entire package of standards referred to as the *Intelligent Peripheral Interface*.

ANSI X3.129-1986 - *Intelligent Peripheral Interface — IPI-0/1*
- *ISO/IEC 9318-1:1990 Physical Level*

ANSI X3.130-1986 - *Intelligent Peripheral Interface — IPI-2*
- *ISO/IEC 9318-2:1990 Device-Specific Command Set for Magnetic Disk Drives*

ANSI X3.132-1987 - *Intelligent Peripheral Interface — IPI-3*
- *ISO/IEC 9318-3:1990 Device-Generic Command Set for Magnetic and Optical Disk Drives*

ANSI X3.147-1988 - *Intelligent Peripheral Interface — IPI-3*
- *ISO/IEC 9318-4:1990 Device-Generic Command Set for Magnetic Tapes*

ANSI X3.176-1990 - *Intelligent Peripheral Interface — IPI-2*
- *ISO/IEC 9318-5:199x Device-Specific Command Set for Magnetic Tapes*

ANSI X3.201-199x - *Intelligent Peripheral Interface — IPI-0/1*
- *ISO/IEC 9318-6:1993 Enhanced Physical Level*

ANSI X3.177-1990 - *Intelligent Peripheral Interface — IPI-3*
- *Device-Generic Command Set for Communications*

Index:

addressee 5, 8, 15, 51
attributes 6, 8, 9, 12-15, 18, 20-23, 25, 28, 29, 33, 38-40, 45, 51
burst 12-15, 18, 20, 22-26, 28-31, 33-38, 40
Command packet 8, 10, 11, 16, 18, 20, 22, 23, 30, 33, 34, 37
data 4-15, 17-26, 28-30, 32-41, 44-47
device-generic 7, 52
device-specific 52
extent 35-38
facility address 16, 27, 45
information transfer 5, 7, 8
information unit 31
interrupts 4, 5, 7-11
logical interface 7, 8
mandatory 22-24, 34-36, 38
multiplex 14
optional 5, 6, 9, 12, 22-24, 32-37, 44, 45
P-busy 29, 38
physical interface 5, 7-9, 12, 17, 26, 29, 31, 38, 45
PhysicalBlock 9
queued 4, 6, 8-10, 13
ready 5-7, 14, 21, 45, 47
Response packet 5-9, 12, 13, 18, 20-23, 25, 26, 28, 33, 35, 36, 47
synonym 38

End of Document

Printed: Monday, January 31, 1994 at 14:43