5. Backplane Physical Layer Specification

The backplane environment physical layer provides the interface from a physical device to the backplane media. It provides arbitration services to permit a node to gain access to the bus, and performs the signal translations required to drive the bus and receive information from the bus.

Unlike the specification for the cable physical layer, the backplane physical layer specification does not include a description of connectors or media. Such documentation is assumed to be part of a host backplane specification or to be included in the requirements for the application environment.

The term “application environment” refers to the physical environment of the bus, the nodes, and the system which contains them. This environment may be a standardized host backplane (e.g., a FutureBus+ profile) which provides the signal requirements, a detailed description of the transceivers, the mechanical arrangement of the modules, and the temperature range over which operation is guaranteed.

The backplane physical layer shares some commonality with the cable physical layer. Common functions include: bus state determination, bus access protocols, encoding and decoding functions, and synchronization of received data to a local clock.

5.1 Backplane Physical Layer Services

PHY Layer services are provided at the interface between the PHY Layer and higher layers; specifically, the Link Layer and the SBM Layer. The method by which these services are communicated between the layers is not defined by this standard. PHY Layer services may perform actions specified by the higher layer. PHY Layer services may also communicate parameters that may or may not be associated with an action.

5.1.1 Backplane Physical Layer Bus Management Services for the Management Layer

These services are used by the SBM Layer to control the bus level actions of the PHY Layer. The PHY Layer uses these services to communicate changes of state within the PHY Layer or on the bus.

5.1.1.1 PHY Control Request (PH_CONTROL.request)

The SBM Layer uses this service to request the PHY Layer to perform specific actions and to specify PHY Layer parameters. It may also be used to request status about the PHY Layer. The PHY Layer shall service the request immediately upon receipt by the PHY Layer. This service is confirmed.

The following actions shall be provided by this service:

- Bus Reset. The PHY Layer shall reset the bus and initialize itself.
- Disable Transmit. The PHY Layer shall set all bus outputs to a high impedance state. This bus output state shall be maintained until the SBM Layer requests an Enable Transmit action. Link Layer service actions that would require a change in bus output state shall not be performed.
- Enable Transmit. The PHY Layer shall allow Link Layer service actions to change the state of the bus outputs.
- Present Status. The PHY Layer shall return status to the SBM Layer. The PHY Layer shall return status via the PHY Control Confirmation service.

NOTE: This service is expected to be used to communicate new parameters without causing any other action.

The following parameters are communicated via this service:

- Arbitration Number. This parameter six-bit number used during the arbitration process. It is generated within the Serial Bus Management Layer and is unique for each node on the bus. Refer to clause 5.4.2.1.

NOTE: These parameters are changed remotely by Transaction Layer “write” operations to control registers (CSRs) defined in the SBM Layer.

5.1.1.2 PHY Control Confirmation (PH_CONTROL.confirmation)

The PHY Layer uses this service to confirm the results of a PHY Control Request service. The PHY Layer shall communicate this service to the SBM Layer upon completion of a PHY Control Request. There are no actions provided by this service. The following parameter is communicated via this service:

- Requested Status. This parameter shall contain the result of a PHY Control Request action. The following values are defined for this parameter:

  COMPLETE. The PHY Control Request action was completed successfully.

5.1.1.3 PHY State Indication (PH_STATE.indication)

The PHY Layer uses this service to indicate to the SBM Layer changes in the state of the PHY Layer. There are no actions provided by this service. No response is defined for this indication. The following parameters are communicated via this service:

- PHY State. This parameter shall contain the current state of the PHY Layer. The following values are defined for this parameter:
  - BUS_RESET_START. The PHY Layer has detected a bus reset.
  - BUS_RESET_COMPLETE. The PHY Layer has completed PHY Layer Initialization.

5.1.2 PHY Layer Arbitration Services for the Link Layer

These services are used to communicate arbitration requests between the PHY Layer and the Link Layer. See clause 5.4.1.

5.1.2.1 PHY Arbitration Request (PH_ARB.request)

The Link Layer uses this service to request the PHY Layer to start arbitration for the bus. The PHY Layer shall arbitrate for the bus using the method specified by the service parameters. The PHY Layer shall service the request immediately upon receipt from the Link Layer. This service shall be confirmed when the arbitration process is complete.

The following parameters are communicated via this service:

- Arbitration Class. This parameter shall contain the method of arbitration performed by the PHY Arbitration Request. The method of arbitration shall be one of the following:
  - FAIR. The Link Layer uses this arbitration class to send a fair asynchronous packet. The PHY Layer shall begin arbitration at the next subaction gap if its arbitration_enable flag is set, otherwise it shall start arbitration at the next arbitration reset gap. The lowest priority (all zeros) is reserved for this arbitration class. Refer to clause 5.4.1.2.
  - URGENT. The Link Layer uses this arbitration class to send an urgent asynchronous packet. The PHY Layer shall begin arbitration at the next subaction gap if its urgent_count is not zero. Otherwise it shall start arbitration at the next arbitration reset gap. The four-bit subparameter “pri” is used to set the priority level for the arbitration process, with the exception that the lowest priority and the highest priority are reserved (and can not be used for this arbitration class; refer to clause 5.4.2.1). This allows urgent requests to get priority access to the bus, but the number of requests that can be made each fairness interval is limited. Refer to clause 5.4.1.3 for a description of urgent arbitration.
  - CYCLE_MASTER. The Link Layer uses this arbitration class to send a cycle start packet. The PHY Layer shall begin arbitration at the next subaction gap. The highest priority (all ones) is reserved for this arbitration class.
  - ISOCHRONOUS. The Link Layer uses this arbitration class to send an isochronous packet. The PHY Layer shall begin arbitration as soon as an acknowledge gap is detected.
  - IMMEDIATE. The Link Layer uses this arbitration class to send an acknowledge packet. As soon as an acknowledge gap is detected, the PHY Layer shall begin transmission of a PH_DATA.request(DATA_PREFIX) for an acknowledge, and a PH_ARB.confirmation(WON) shall be communicated to the Link Layer.
- Priority. This parameter shall contain the priority associated with the Urgent Access Method. This parameter shall be ignored if the arbitration class is ISOCHRONOUS or IMMEDIATE.

NOTE: This parameter could be used to distinguish FAIR and CYCLE MASTER requests from URGENT requests.

5.1.2.2 PHY Arbitration Confirmation (PH_ARB.confirmation)

The PHY Layer uses this service to confirm the results of a PHY Arbitration Request service. The PHY Layer shall communicate this service to the Link Layer upon completion of a PHY Arbitration Request. There are no actions provided by this service. The following parameter is communicated via this service:

- Arbitration Request Status. This parameter shall contain the result of a PHY Arbitration Request action. The following values are defined for this parameter:
  - WON. The PHY Arbitration Request action was completed successfully. The PHY Layer shall immediately begin sending PHY Clock Indications to the Link Layer.
  - LOST. The PHY Arbitration Request action was not successful.

5.1.3 PHY Layer Data Services for the Link Layer

These services are used to communicate data symbols and control information between the PHY Layer and the Link Layer.

5.1.3.1 PHY Clock Indication (PH_CLOCK.indication)

The PHY Layer uses this service to indicate to the Link Layer the data rate for the current transaction. Depending upon the technology used for the bus media, the frequency of this clock may be 49.152 MHz (±100 ppm) or 24.576 MHz (±100 ppm). There are no actions provided by this service. No response is defined for this indication. No parameters are communicated via this service.

The PHY layer shall begin communicating these indications to the Link Layer after it has communicated a PHY Arbitration Confirmation with an Arbitration Request Status of WON. The PHY Layer shall stop communicating these indications to the Link Layer after the Link Layer communicates a PHY Data Request with Data of DATA_END.

5.1.3.2 PHY Data Request (PH_DATA.request)

The Link Layer uses this service to control the PHY Layer's transmission of clocked data symbols. The Link Layer shall communicate one PHY Data Request for each PHY Clock Indication. The PHY Layer shall service the request immediately upon receipt by the PHY Layer. This service shall not be confirmed.

The following parameter is communicated via this service:

- Data. This parameter shall contain the symbol to be transmitted on the bus. See clause 5.2.2.1 for a definition of the logic states associated with the symbols. T he following values are defined for this parameter:

  DATA_ONE. A symbol representing a data bit of one shall be transmitted on the bus.

  DATA_ZERO. A symbol representing a data bit of zero shall be transmitted on the bus.

  DATA_PREFIX. The PHY Layer shall stop sending clocked data bits and leave the bus in the data_prefix state using the algorithm described in clause 5.4.2.1. This symbol shall be transmitted between acknowledge and response packets during concatenated subactions.

  DATA_END. The Link Layer shall stop communicating PHY Data Requests to the PHY Layer. The PHY Layer shall stop communicating PHY Clock Indications to the Link Layer. The PHY Layer shall set all bus outputs to the data_end state using the algorithm described in clause 5.4.2.1.

NOTE: A DATA_PREFIX can be initiated by the PHY Layer. This occurs: once a PHY Arbitration Request has been completed successfully and until transmission of the packet begins; and once an acknowledge gap is detected and until transmission of the acknowledge packet begins.

Once the Link Layer starts sending a data bit (DATA_ONE or DATA_ZERO), it must guarantee a minimum of at least 8 continuous data bits (no intervening DATA_PREFIX), and the total number of continuous data bits must be even.

Once the Link Layer starts sending a DATA_PREFIX or a DATA_END, it shall continue sending it for at least four base rate bit times (≈ 81.38 ns) before sending any other data symbols. This minimum value allows sufficient time for all nodes to detect that such a symbol has been transmitted. The duration of a DATA_PREFIX shall not exceed 160 base rate bit times (≈ 3255.2 ns), and the duration of a DATA_END shall not exceed 16 base rate bit times (≈ 325.52 ns). These maximum values ensure that a transaction will be completed within a certain amount of time.

5.1.3.3 PHY Data Indication (PH_DATA.indication)

The PHY Layer uses this service to indicate to the Link Layer changes in the state of the PHY Layer. These changes can include received data and other bus events. The PHY Layer shall communicate this indication to the Link Layer for each data bit received. If the node is not receiving data bits, the PHY Layer shall use this indication to communicate events needed by the Link Layer. No response is defined for this indication. The following parameters are communicated via this service:

- Data. This parameter shall contain the information decoded by the PHY which is needed by the Link Layer. The following values are defined for this parameter:

  DATA_START. The start of a packet has been detected on the bus.

  DATA_ONE. A data bit of one has been received on the bus.

  DATA_ZERO. A data bit of zero has been received on the bus.

  DATA_PREFIX. A node has control of the bus, and is preparing to transmit a packet.

  DATA_END. The end of a packet has been detected on the bus, and the transmitting node is releasing control.

ARBITRATION_RESET_GAP. An Arbitration Reset Gap event has been detected on the bus. This event is needed for Link Layers that implement the Retry A/B protocol.

5.2 Backplane Physical Connection Specification

Within the backplane environment the Serial Bus is implemented with a pair of signals: Sbus_Strb and Sbus_Data. The topology is a simple pair of bussed signals as shown in figure 5-1.

Figure 5-1 – Backplane topology

The backplane environment can be implemented with a number of different interface technologies. These include, but are not limited to: TTL for industry-standard transistor-transistor logic, BTL for backplane transceiver logic as defined by IEEE 1194.1, and ECL for emitter-coupled logic.

In addition to the requirements specified by the application environment, the physical media of the Serial Bus shall meet the requirements defined for media attachment, media signal interface, and media signal timing. Timing requirements must be met over the ranges specified in the application environment. These include temperature ranges, voltage ranges, and manufacturing tolerances.

5.2.1 Media Attachment

5.2.1.1 Distribution of Nodes

The maximum number of modules on a backplane shall not exceed the maximum number specified for the application environment. The number of nodes on the backplane shall not exceed 63, although there is no restriction on the distribution of these nodes throughout the modules on the bus. All nodes shall have unique numbers or node addresses. Refer to clause 5.4.2.1.

As long as the arbitration and bus synchronization timing requirements are met it is unnecessary to specify the number of modules on the backplane, the pitch of the modules, or the load that each module presents to the backplane. Nonetheless, these characteristics may still be defined within the specification (or profile) for a particular application environment.

5.2.1.2 Fault Detection and Isolation

It is recommended that modules be designed to support fault detection, and that single points of failure between the serial bus and the host backplane bus be kept to a minimum. Within a module, it is recommended that the serial bus transceiver circuitry be packaged separately from the parallel backplane circuitry. Within the backplane chassis, it is recommended that termination impedances and voltage supplies be packaged separately where possible.

A serial bus backplane shall be capable of operation with modules not powered or removed. A disabled or not powered module shall not affect the operation of the bus.

5.2.1.3 Live Insertion

Depending upon the application environment, modules implementing Serial Bus may be required to support live insertion. Modules may support live insertion, minimizing glitches that may occur upon insertion of a board into an active backplane. The implementation of such shall in no way introduce incompatibilities, under normal operating conditions, between interface circuits supporting live insertion and interface circuits that do not.

5.2.2 Media Signal Interface

The backplane media signal interface consists of two single-ended signal interfaces; one for Sbus_Data and one for Sbus_Strb.

If the Serial Bus is intended to accompany a standardized parallel bus, it is recommended that the media signal interface of the Serial Bus be similar to this bus. It is the intention of this specification that the transceivers, terminations, and other physical parameters be the same as those required for the control lines of the host bus (e.g., FutureBus+ profile characteristics or VMEbus electrical specifications). In this case, the specification for the host bus should be referenced for a description of the media signal interface.

If there is no host bus or the host bus does not adequately address the physical requirements for the serial bus, then the choice of interface technology is left to the implementor. Care should be taken with the engineering of the backplane to ensure proper performance of the bus. This document specifies parameters for the use of certain technologies (i.e., TTL, BTL, and ECL), but does not prohibit the use of other technologies. It is not appropriate for this document to specify a

particular technology for a Serial Bus application.

5.2.2.1 Definition of Logic States

Drivers shall assert the bus to indicate a “1” logic state, or release the bus to indicate a “0” logic state. To assert the bus, a driver sinks current, causing the bus signal to be brought to a low voltage level. To release the bus, drivers are tri-stated or turned off, allowing the bus signal to be pulled up to the termination voltage of the bus. This results in a logical inversion of signals levels on the bus.

All drivers shall operate in a “wire-OR” or “open-collector” mode during arbitration. Drivers may operate in a “totem pole” mode during data packet and acknowledge transfers. In this mode, a driver may “drive” the bus into its released state in order to decrease the rise time of the bus signal (referred to as a “rescinding release” with TTL technology).

5.2.2.2 Bit Rates

Data transmission and reception shall occur at 24.576 Mbit/sec (±100 ppm) for TTL, and 49.152 Mbit/sec (±100 ppm) for BTL and ECL. Regardless of the interface technology, arbitration occurs at the same rate (refer to clause 5.2.4.2).

5.2.2.3 Transition Times

The transition times of the transceivers shall comply with the values indicated within the media signal timing (table 5-1, table 5-2 and table 5-4). The minimum detectable pulse width of receivers shall be less than the edge separation specified in table 5-2.

5.2.2.4 Noise Rejection

Receivers may have noise filters which reject pulses. Such pulses are 3 ns or less for TTL and 1.5 ns or less for BTL, measured at the center of the receiver input threshold.

5.2.3 Media Signal Timing

5.2.3.1 Backplane Transmit Data Timing

The backplane bus signals, as they appear from the output of the transmitters and onto the backplane media, shall be within the constraints outlined in figure 5-2. Different data rates are supported for the different backplane technologies. Rise and fall times for the bus signals are measured from 10% to 90%. Slew rates are specified to insure minimum high frequency noise coupling during signal transitions. Edge separation is the minimum required time between any two consecutive transitions of the bus signals, whether they be transitions on the same signal or transitions on the two separate signals. Edges are measured at the center of the receiver input threshold. A minimum edge separation is required to ensure proper operation of the data-strobe bit level encoding mechanism.

Figure 5-2 – Backplane transmit data timing

Table 5-1 – Backplane transmit data timing

		TTL	**BTL**	**ECL**
	data rate	S25	S50	S50
t1	bit cell period	40.690 ns	20.345 ns	20.345 ns
t2	rise time	1.2 ns min 3 ns max (from 1.0 to 2.0 V*)	5 ns max	2 ns min 5 ns max
t3	fall time	1.2 ns min 3 ns max (from 1.0 to 2.0 V*)	5 ns max	2 ns min 5 ns max
	slew rate		0.5 V/ns max (from 1.3 to 1.8 V**)	
t4	TX edge separation	33 ns min	15 ns min	15 ns min

NOTE: * Measured with a 25 Ω ±1% load terminated to 1.5 V ±1%
** Measured with a 16.5 Ω load to 2.1V.

5.2.3.2 Backplane Receive Data Timing

The receiver typically uses the transitions on the incoming bus signals Data_Rx and Strb_Rx to derive a clock at the code bit frequency to extract the NRZ signal on Data_Rx. This clock may be derived by performing an exclusive-OR of Data_Rx and Strb_Rx.

The bus signals, as they appear from the backplane media and into the receivers, shall be within the constraints outlined by figure 5-3. The minimum required edge separation is reduced to allow for backplane and receiver skew.

Figure 5-3 – Backplane receive data timing

Table 5-2 – Backplane receive data timing

		TTL	BTL	ECL
	data rate	S25	S50	S50
t1	bit cell period	40.690 ns	20.345 ns	20.345 ns
t2	rise time	1.2 ns min 3 ns max (from 1.0 to 2.0 V*)	5 ns max	2 ns min 5 ns max
t3	fall time	1.2 ns min 3 ns max (from 1.0 to 2.0 V*)	5 ns max	2 ns min 5 ns max
	slew rate		0.5 V/ns max (from 1.3 to 1.8 V**)	
t4	Rx edge separation	26 ns min	9 ns min	9 ns min

NOTE: * Measured with a 25 Ω ±1% load terminated to 1.5 V ±1%
** Measured with a 16.5 Ω load to 2.1V.

5.2.3.3 Backplane and Transceiver Skew

The skew introduced between Data_Rx and Strb_Rx by the Serial Bus backplane and transceiver packages shall be no greater than the values mentioned in table 5-3. This table gives the maximum allowable skew for each technology. Conformance to these requirements is necessary to ensure Tx and Rx edge separations. Calculations for edge separation and skew margin are contained in the backplane timing annex to this document.

Table 5-3 – Maximum Transceiver Package and Bus Skew

	TTL	BTL	ECL
Tx package skew	5 ns	3 ns	3 ns
Rx package skew	5 ns	3 ns	3 ns
backplane skew	7 ns	6 ns	6 ns

5.2.4 Backplane Arbitration Timing

5.2.4.1 Bus Synchronization and Propagation Delay

To ensure proper operation of the arbitration mechanism, all nodes participating in arbitration must be synchronized to within a specified time period. This requirement allows all nodes to arbitrate at approximately the same time.

To achieve synchronization, nodes preparing to arbitrate shall sample the bus until an idle condition is detected. The bus

becomes idle once 4 base rate bit times (≈ 81.38 ns) have occurred without Data_Rx or Strb_Rx being asserted.

A node waiting to arbitrate for the bus shall detect an idle bus within 43.345 ns of the idle condition occurring at the receiver inputs. This time assumes a maximum propagation delay through the receiver (8 ns), a maximum input delay and setup time for the decision logic (15 ns), and a maximum synchronization delay of 1 base rate clock period (20.345 ns). If Sbus_Strb has already been asserted by another node beginning to arbitrate, all other nodes shall detect that the bus has been asserted within 43.345 ns of the arrival of the asserted signal at their receiver inputs.

Once a node which is waiting to arbitrate detects that the appropriate gap has occurred (clause 5.3.3), it must assert Strb_Tx within 43.345 ns. This time assumes a maximum output delay for the decision logic (15 ns), a maximum propagation delay through the driver (8 ns), and one state machine clock period (20.345 ns).

In order to guarantee that proper arbitration timing, it is necessary to specify a maximum bus propagation delay. The maximum time required for a signal to propagate from one end of the backplane serial bus to the other end (a one-way propagation delay) shall be 18 ns or less.

5.2.4.2 Arbitration Bit Timing

A node gains access to the serial bus by using the arbitration process. This is done in response to a PH_ARB.request from the Link Layer. To arbitrate for the bus, a node samples Strb_Rx and Data_Rx to determine if the bus is active. If the bus is inactive for a subaction_gap or an arbitration_reset_gap (for an asynchronous transfer) or an acknowledge_gap (for an isochronous transfer) the node can begin the arbitration process. It asserts Strb_Tx to indicate the beginning of the arbitration process. The assertion of Strb_Tx also ensures that long strings of “0” arbitration bits are not interpreted as gaps between packets.

At the same time that Strb_Tx is asserted, the node begins to transmit its arbitration sequence on Data_Tx. Each symbol in the sequence has a duration of 16 base rate bit times (≈ 325.6 ns). If the symbol to be transmitted is a “1” the node asserts Data_Tx. If the symbol to be transmitted is a “0” the node releases the Data_Tx, waits 10 base rate bit times (≈ 203.4 ns), and then samples Data_Rx. If the bus is asserted at this time, the node has lost the arbitration contest. It sends a PH_ARB.confirmation(LOST) to the Link Layer, and then must drop out of the arbitration process and wait for the next appropriate gap to compete for the bus. Regardless of whether arbitration is won or lost, the node must not cause a transition on Sbus_Strb or Sbus_Data (i.e., release or assert the bus) until 6 more base rate bit times have passed. This ensures that all participants in the arbitration process have had a chance to sample that symbol in the arbitration sequence.

An arbitrating node continues to transmit its arbitration sequence until it loses arbitration to a higher priority node or it successfully completes its arbitration sequence. If a PHY Layer has successfully transmitted each of its 10 arbitration symbols and still holds the bus, it sends a PH_ARB.confirmation(WON) and a PH_CLOCK.indication to the Link Layer. The PHY Layer also initiates the transmission a PH_DATA.request(DATA_PREFIX) on the bus (using the transitions in clause 5.4.2.1) until PH_DATA.requests are received from the Link Layer.

The backplane bus signals, as they appear from the output of the transmitters as well as on the inputs of the receivers, shall be within the constraints outlined in figure 5-4. The arbitration symbol period is the same regardless of the backplane interface technology. The symbol period (t1) is divided into the sample time (t4) and hold time (t6). The sample time allows enough time for the bus to settle after arbitrating nodes have placed a symbol on the bus. The hold time enables all nodes on the bus to sample the symbol before transitions or glitches can occur on the bus. Rise and fall times for the bus signals are measured from 10% to 90%. Slew rates, if specified, insure minimum high frequency noise coupling during signal transitions.

NOTE: All signals are displayed with a “wired or” inversion.

Figure 5-4 – Arbitration symbol timing

Table 5-4 – Arbitration symbol times

		TTL	BTL	ECL
	data rate	S25	S50	S50
t1	arbitration symbol period	16 base rate code bits (≈ 325.6 ns)	16 base rate code bits (≈ 325.6 ns)	16 base rate code bits (≈ 325.6 ns)
t2	rise time	4 ns min 30 ns max	5 ns max	2 ns min 5 ns max
t3	fall time	4 ns min 10 ns max	5 ns max	2 ns min 5 ns max
	slew rate		0.5 V/ns (from 1.3 to 1.8 V*)	
t4	sample time	10 base rate code bits (≈ 203.4 ns)	10 base rate code bits (≈ 203.4 ns)	10 base rate code bits (≈ 203.4 ns)
t5	setup time	5 ns min	5 ns min	5 ns min
t6	hold time	6 base rate code bits (≈ 122.0 ns)	6 base rate code bits (≈ 122.0 ns)	6 base rate code bits (≈ 122.0 ns)

NOTE: * Measured with a 16.5 Ω load to 2.1V.

More information on arbitration symbol timing is contained in the backplane timing annex to this document. Refer to clause 5.3.4 for a description of the arbitration sequence.

5.3 Backplane Physical Layer Facilities

5.3.1 Coding

Peer Physical Layer entities on the bus communicate via NRZ data. The NRZ data is transmitted and received as Sbus_Data and is accompanied by a strobe signal, Sbus_Strb. This strobe signal changes state whenever two consecutive NRZ data bits are the same, ensuring that a transition occurs on either Sbus_Data or Sbus_Strb. A clock which transitions every bit period can be extracted by performing an exclusive OR of Data_Rx and Strb_Rx, as is shown in figure 5-5.

Figure 5-5 – Data-strobe coding

NOTE: The use of “data-strobe bit-level encoding” is based on (and is in basic agreement with) the following reference: U.K. patent number 9011700.3, claim 16 (DS-Link bit-level encoding) held by INMOS Limited, 1000 Aztec West, Almondsbury, Bristol BS12 4SQ, UK.

The primary rationale for use of this transmission code is to improve the transmission characteristics of information to be transferred across the serial bus. In particular, the code ensures that transitions occurring on Data_Rx and Strb_Rx are approximately one bit period apart. This results in an increase in skew tolerance which could not be obtained with a clocked NRZ format.

The procedure for encoding data during transmission of a packet is described in clause 5.4.2.1, and decoding during reception of a packet is described in clause 5.4.2.2.

5.3.2 Bus Signals

Backplane PHYs can send the following signals:

BUS_RESET	Both Data_Tx and Strb_Tx are asserted for more than 320 base rate bit times (6510.4 ns). This signal is sent in response to a PH_CONTROL.request(BUS_RESET) from the Link Layer (clause 5.1.1.1).
ARBITRATE	An arbitrating node asserts Strb_Tx while transmitting the arbitration sequence on Data_Tx. Arbitration bit timing is described in clause 5.2.4.2. Arbitration occurs in response to a PH_ARB.request from the Link Layer (clause 5.1.2.1).
PACKET	A node transfers a request or a response packet by transmitting NRZ Data on Data_Tx and a strobe signal on Strb_Tx. This strobe signal changes state whenever two consecutive NRZ data bits are the same, ensuring that a transition occurs on either Data_Tx or Strb_Tx. This bit level encoding method is described in clause 5.3.1. Data bits occur in response to a PH_DATA.request(DATA_ONE or DATA_ZERO) from the Link Layer (clause 5.1.3.2).
DATA_PREFIX	Before it transmits a packet, a node can hold the bus by asserting Data_Tx and deasserting (or driving into the unasserted state) Strb_Tx for at least four base rate bit times (81.38 ns), and at most 160 base rate bit times (3255.2 ns). The signal transitions required for this are defined in clause 5.4.2.1. This signal occurs in response to a PH_DATA.request(DATA_PREFIX) from the Link Layer (clause 5.1.3.2). This signal can also be initiated by the PHY Layer after arbitration is successfully completed (clause 5.2.4.2).
DATA_END	After it transmits a packet, a node signals the release of the bus by asserting Strb_Tx and deasserting (or driving into the unasserted state) Data_Tx for at least four base rate bit times (81.38 ns), and at most 16 base rate bit times (325.52 ns). The signal transitions required for this are defined in clause 5.4.2.1. This signal occurs in response to a PH_DATA.request(DATA_END) from the Link Layer (clause 5.1.3.2)

Backplane PHYs can receive the following signals:

BUS_RESET	Both Data_Rx and Strb_Rx are asserted for more than 320 base rate bit times (6510.4 ns). The duration of the reset signal distinguishes it from a sequence of zeroes transmitted during arbitration.
BUS_IDLE	Both Data_Rx and Strb_Rx are unasserted for at least 4 base rate bit times (≈ 81.4 ns). The amount of time that the bus is unasserted (idle) distinguishes an acknowledge_gap, a subaction_gap and an arb_reset_gap.
ARBITRATE	Strb_Rx is asserted as the arbitration sequence is received on Data_Rx. Arbitration bit timing is described in clause 5.2.4.2.
PACKET	NRZ Data is received on Data_Rx and is accompanied by a strobe signal on Strb_Rx. This strobe signal changes state whenever two consecutive NRZ data bits are the same, ensuring that a transition occurs on either Data_Rx or Strb_Rx. This bit level encoding method is described in clause 5.3.1. Each bit in the packet results in a PH_DATA.indication(DATA_ONE or DATA_ZERO) to the Link Layer (clause 5.1.3.3).
DATA_PREFIX	Data_Rx is asserted and Strb_Rx is unasserted for at least four base rate bit times. This indicates that another node is holding the bus before transmitting a packet. The receipt of this signal results in a PH_DATA.indication(DATA_PREFIX) to the Link Layer (clause 5.1.3.3).
DATA_END	Strb_Rx is asserted and Data_Rx is unasserted for at least four base rate bit times. This indicates that another node has released the bus after the transmission of a packet. The receipt of this signal results in a PH_DATA.indication(DATA_END) to the Link Layer (clause 5.1.3.3).

5.3.3 Gap Flags

A gap is a period of time during which the bus is idle (Data_Rx and Strb_Rx are unasserted). There are three types of gaps:

a) Acknowledge_gap — Appears between the end of a packet and an acknowledge, as well as between isochronous transfers. The bus has been in an unasserted state for at least 8 base rate bit times (162.76 ns). This includes the minimum time required to detect a BUS_IDLE, as well as the maximum delay between the arbitration state machines within any two nodes on the bus.

b) Subaction_gap — Appears before asynchronous transfers within a fairness interval. The bus has been in an unasserted state for at least 16 base rate bit times (325.52 ns). This ensures that an any other nodes asserting the bus after an acknowledge_gap will have been detected by this time.

c) Arb_reset_gap — Appears before asynchronous transfers when the fairness interval starts. The bus has been in an unasserted state for at least 24 base rate bit times (488.28 ns). This ensures that any other nodes asserting the bus after a subaction_gap or an acknowledge_gap will have been detected by this time.

If a node is waiting to arbitrate for the bus, and the bus becomes idle for an amount of time equal to the appropriate gap, the node must detect that the gap has occurred and begin the arbitration process within the time constraints described in clause 5.2.4.1.

More information on gap timing is available in the backplane timing annex to this document.

5.3.4 Arbitration Sequence

5.3.4.1 Arbitration Number

The arbitration sequence uses a unique arbitration number for each module. This 6-bit number is the same as the node's offset_ID, and may be set by the host backplane (e.g., with a built-in slot identifier or configuration mechanism). It is recommended that this number be software programmable to facilitate testing and to allow for consistent system operation and repeatability.

5.3.4.2 Priority

Within the arbitration sequence, the arbitration number is preceded by four bits which define a priority level. The method by which priority is assigned is to be determined by the system integrator, with two exceptions: the lowest priority (all zeros) is reserved for fair arbitration and the highest priority (all ones) is reserved for cycle start requests. This allows 14

priority levels to be used for the urgent arbitration process.

The use of an urgent priority class allows nodes to be granted a larger portion of the bandwidth on the bus. High priority nodes are granted the bus before lower priority nodes during urgent allocation of the bus, allowing such nodes to be granted more bandwidth. If used properly, deterministic latency algorithms (e.g., rate monotonic scheduling) could be used to assign priority to transfers so that they would be made within a specified amount of time.

In order to guarantee forward progress, the lowest priority level must be reserved for fair arbitration. This allows all nodes arbitrating with this priority level to be allowed one fair access to the bus each fairness interval. For fair arbitration, the arbitration number has a minimal impact on the allocation of the bus. Although nodes with higher arbitration numbers will be granted the bus sooner, there is only a small decrease in latency.

The four bit priority field is not used in isochronous arbitration. When arbitrating for an isochronous transfer, the priority field is zero-filled.

5.3.4.3 Format of Arbitration Sequence

The following format for the arbitration sequence shall be used:

a) Each module on the backplane shall have a unique 6-bit arbitration number which is equal to the node's offset_ID.

b) The arbitration number shall be preceded by four bits of priority (refer to clause 5.4.2.1).

c) Dynamic assignment of priority must be accommodated.

d) The lowest priority level (all zeroes) shall be reserved for fair arbitration, and the highest priority level (all ones) shall be reserved for the identification of the cycle start packet.

5.4 Backplane Physical Layer Operation

The operation of the backplane physical layer can best be understood with reference to the architectural diagram shown in figure 5-6. This diagram shows the two primary functions performed by the physical layer: arbitration and bit-level data transmission and reception.

Figure 5-6 – Backplane physical layer architecture

The main controller of the backplane physical layer is the block labeled "arbitration control," which responds to arbitration requests from the Link Layer (PHY_ARB.request) and indicates changes in the state of the bus. It provides the management and timing signals for transmitting and receiving packets. It also provides the bus reset and configuration functions. The operation of this block is described in clause 5.4.1.

The "encode" block generates the appropriate strobe signal for the transmitted data. Its operation is described in clause 5.4.2.1.

The "data resynch" block decodes the data-strobe signal and retimes the received data to a local fixed frequency clock provided by the "local clock" block. Since the clocks of receiving and transmitting nodes can be up to 100 ppm different from the nominal, the data resynch function must be able to compensate for a difference of 200 ppm over the maximum packet length of 87.24 µsec (256 byte isochronous packet at 24.576 Mbit/sec). The operation of this block is described in clause 5.4.2.2.

The "data & signal decode" block provides a common interface to the Link Layer for both packet data and arbitration signals (data, gaps and bus reset indicators). The operation of this block is also described in clause 5.4.2.2

5.4.1 Arbitration

Unless a node is using immediate arbitration to access the bus (in which case there is no contention for the bus), it is possible that more than one node may attempt to access the bus at a given time. Consequently, it is necessary to arbitrate for the bus in order to gain access to the bus.

NOTE: A node uses immediate arbitration to send an acknowledge. Since there is no contention for the bus in this case, arbitration is not necessary (a node that is transmitting an acknowledge does not arbitrate for the bus, but merely waits for an acknowledge gap to be detected before it begins transmission). If a node is attempting to gain access to the bus without using immediate access, it must first arbitrate for the bus.

Arbitration occurs in response to a PHY Arbitration Request from the Link Layer. Nodes begin arbitrating once the bus has become idle for a predetermined amount of time (the appropriate gap indication occurs). Once this happens, nodes

begin a bit-by-bit transmission of their arbitration sequence. Refer to clause 5.3.4 for a description of the arbitration sequence. Refer to clause 5.2.4 for a description of the bit-by-bit process of asserting and sampling the bus during arbitration.

A node can obtain access to the bus in a limited number of ways. Because some arbitration classes allow nodes to begin arbitration before others, nodes arbitrating with certain arbitration classes may detect that the bus is busy before they can begin to arbitrate. In this way, certain arbitration classes can be bypassed (e.g., fair and urgent nodes will not get a chance to arbitrate if another node is sending an acknowledge or if it is arbitrating for an isochronous transfer).

The backplane environment supports the fair, urgent, cycle_master, isochronous and immediate arbitration classes. These are described in the following clauses. Refer to clause 5.1.2.1 for a description of the PH_ARB.request.

5.4.1.1 Fairness Intervals

The fairness protocol is based on the concept of a fairness interval. A fairness interval consists of one or more periods of bus activity separated by short idle periods called subaction_gaps and is followed by a longer idle period known as an arb_reset_gap. At the end of each gap, bus arbitration is used to determine the next bus owner. This concept is shown in figure 5-7.

Figure 5-7 – Fairness interval

The implementation of the fair arbitration protocol is defined in terms of these fairness intervals, as is discussed in the following clauses.

5.4.1.2 Fair Arbitration

When using this arbitration class, an active node becomes a bus owner exactly once each fairness interval. Once a subaction_gap is detected, a node can begin arbitration if its arb_enable signal is set. The arb_enable signal is set at the beginning of the fairness interval and is cleared when the node becomes a bus owner. In the absence of urgent nodes, a fairness interval ends when all of the nodes attempting fair arbitration have successfully accessed the bus. At this time, all of the fair nodes have their arb_enable signals reset and cannot arbitrate for the bus. The bus remains idle until an arb_reset_gap occurs. Once this happens, the next fairness interval begins. All of the nodes set their arb_enable signal and can begin to arbitrate for the bus. This process is illustrated in figure 5-8.

Figure 5-8 – Fair arbitration

5.4.1.3 Urgent Arbitration

When using urgent arbitration, a node may be a bus owner more than once in each fairness interval, with the constraint that the number of urgent packets (Nu) is related to the number of previously-sent fair packets in the same interval (Nf), as follows:

$Nu \leq 3 + 3*(Nf)$

Once a subaction_gap is detected, a node can begin urgent arbitration if it has an urgent_count which is not zero. When an fairness interval begins, all nodes have an urgent_count which is set to three. After any node successfully accesses the bus using urgent arbitration, all of the nodes on the bus decrement their urgent_count by one. After three urgent accesses have occurred, all nodes shall have their urgent_count decremented to zero, allowing fair nodes to successfully access the bus. After a fair access has occurred, all nodes set their urgent_count to three once again. This ensures that urgent packets are able to get most (up to three fourths) of the accesses, while reserving at least one fourth of the accesses for fair packets.

In the presence of urgent nodes, a fairness interval ends after the final fair node and up to three remaining urgent nodes have successfully accessed the bus. Since all fair nodes now have their arb_enable signals reset and all urgent nodes have their urgent_count decremented to zero, none of the nodes can access the bus. The bus remains idle until an arb_reset_gap has occurred, re-enabling arbitration on all nodes and starting the next fairness interval. This process is illustrated in figure 5-9:

Figure 5-9 – Urgent arbitration

Note that nodes using the urgent protocol have a higher priority than those using the fair protocol. If any nodes on a backplane are sending urgent packets, then those packets will be transmitted first.

5.4.1.4 Cycle_Master Arbitration

This arbitration class is used by the cycle_master when it needs to arbitrate for the transmission of a cycle_start packet. It is similar to the urgent arbitration class, except that the priority field is defined to be all ones (clause 5.4.2.1). Arbitration begins once a subaction_gap is detected, regardless of the state of the arb_enable signal or the urgent_count.

5.4.1.5 Isochronous Arbitration

This arbitration class is used by nodes arbitrating to send isochronous packets. Arbitration begins once an acknowledge_gap is detected, regardless of the state of the arb_enable signal or the urgent_count. Because an acknowledge_gap is shorter than an arb_reset_gap and a subaction_gap, nodes arbitrating with this class will win the bus before nodes waiting to send fair, urgent, or cycle_master packets. The priority field is not used in this arbitration class (clause 5.4.2.1).

5.4.1.6 Immediate Arbitration

This arbitration class is used by nodes sending an acknowledge to a received packet. Transmission of the acknowledge (beginning with a DATA_PREFIX, refer to clause 5.4.2.1) occurs as soon as an acknowledge_gap is detected. This arbitration class is referred to as “immediate” because an arbitration sequence is not transmitted to obtain access to the bus.

5.4.2 Backplane Environment Packet Transmission and Reception

Packet transmission and reception are synchronized to a local clock that must be accurate within 100 ppm. The nominal data rates are either 49.152 Mbit/sec or 24.576 Mbit/sec for the backplane environment, depending on the particular backplane technology.

5.4.2.1 **Backplane Environmen**t Packet Transmission

Protocols and timing for packet transmission are different than those for arbitration. Packet transmission uses the data-strobe bit level encoding method described in clause 5.3.1, and is delimited by PH_DATA.request(DATA_PREFIX and DATA_END) symbols. Timing of the signals used to implement the encoding method is described in clause 5.2.3.1.

Once a PHY Layer completes an arbitration sequence, it signals the beginning of packet transmission by putting the bus into the PH_DATA.request(DATA_PREFIX) state. Because Sbus_Data can be either asserted or unasserted at the end of the arbitration sequence, the PHY layer first asserts Data_Tx for one base rate bit time (≈ 20.35 ns) to put the bus into a known state (Strb_Tx is already in a known state because it is asserted during the arbitration sequence). Strb_Tx is then deasserted to transmit the DATA_PREFIX.

If the PHY Layer uses immediate arbitration to access the bus, the DATA_PREFIX does not follow an arbitration sequence (the bus has been idle for an acknowledge_gap). In this case, Strb_Tx remains unasserted while Data_Tx is asserted to transmit the DATA_PREFIX.

Once the PHY Layer begins transmitting a DATA_PREFIX, it shall continue its transmission for at least four base rate bit times (≈ 81.38), and it shall begin transmission of a packet within 160 base rate bit times (≈ 3255.2 ns).

The transmission of data within a packet occurs in response to PHY Data Requests. The Link Layer transmits each data bit in the packet as a PH_DATA.request(DATA_ONE or DATA_ZERO) in response to PH_CLOCK.indications from the PHY Layer. The PHY Layer sends PH_CLOCK.indications to the Link Layer for each bit to indicate the rate at which data bits are to be transmitted. The PHY Layer encodes the data bits and produces the transmit signals Data_Tx and Strb_Tx. In order to provide a transition from the DATA_PREFIX state once the first bit of data is transmitted, Strb_Tx is asserted if the first bit is a DATA_ONE, or Data_Tx is deasserted if the first bit is a DATA_ZERO. This ensures that a transition will occur on either Sbus_Strb or Sbus_Data, and that a clock indication will be generated for the first bit of data once it is received by other nodes.

The completion of a packet transmission requires two additional bits after the last bit in the packet. The first bit is required to flush the last bit in the packet through the receiving circuit. The second bit is required to put Data_Tx and Strb_Tx into the proper state to transmit a PH_DATA.request(DATA_PREFIX or DATA_END). A node transmits a DATA_PREFIX at the end of a packet in order to hold onto the bus (with a concatenated response, this would happen after a node transmits the acknowledge and before it transmits the response packet). A node transmits a DATA_END at the end of a packet in order to signal the release of the bus. Refer to clause 5.3.3 for further description of these signals.

In order to remain in accordance with the data-strobe bit level encoding algorithm (clause 5.3.1), only one of the two

signals Data_Tx and Strb_Tx can make a transition at one time. The following tables indicate the proper series of transitions required to: provide the extra transition at the end of a packet; leave the bus in the proper state (DATA_PREFIX or DATA_END); and comply with the data-strobe bit level encoding algorithm. Note that the Data_Tx and Strb_Tx cannot have the same values at the end of the packet since: 1) the starting state of the bus is DATA_PREFIX (Strb is 1 and Data is 0), and 2) the number of bits in a packet **must** be even.

Table 5-5 – DATA_PREFIX signal transitions after packet transmission

	signal state after last bit in packet	**first transition**	**second transition**
Data_Tx Strb_Tx	0 1	1 1	1 0
Data_Tx Strb_Tx	1 0	1 1	1 0

NOTE: These signals undergo a “wire or” inversion at the output of the bus transceiver.

Table 5-6 – DATA_END signal transitions after packet transmission

	signal state after last bit in packet	**first transition**	**second transition**
Data_Tx Strb_Tx	0 1	1 1	0 1
Data_Tx Strb_Tx	1 0	1 1	0 1

NOTE: These signals undergo a “wire or” inversion at the output of the bus transceiver.

Once the transmission of a DATA_PREFIX or a DATA_END begins, it shall continue for at least four base rate bit times (≈ 81.38 ns). This allows sufficient time for all nodes to detect that such a symbol has been transmitted. The duration of a DATA_PREFIX shall not exceed 160 base rate bit times (≈ 3255.2 ns), and the duration of a DATA_END shall not exceed 16 base rate bit times (≈ 325.52 ns). Within this time, the PHY Layer must either begin transmission of a packet or release the bus. This ensures that transactions are completed within a certain amount of time.

5.4.2.2 Backplane Environment Packet Reception

Upon receipt of a packet, the receive signals Data_Rx and Strb_Rx are decoded using the method described in clause 5.3.1. The timing of these signals shall meet the requirements described in clause 5.2.3.2.

Each recovered data bit is sent to the Link Layer as a PHY Data Indication. Data bits within the packet are indicated as a PH_DATA.indication(DATA_ONE or DATA_ZERO). DATA_PREFIX and DATA_END indications are also communicated to the Link Layer, as well as DATA_START indications which are generated upon the reception of a packet. Indications of gap events may be communicated to the Link Layer, but they are used primarily within the PHY Layer.

5.5 Backplane Bus Initialization

Upon entering the bus (e.g., power up, live insertion, or bus reset), a module shall wait for an arbitration_reset_gap before arbitrating for the bus.