Annex <$chapnum>

(normative)

SONET/SDH reconciliation sublayers

Editors' Notes: *To be removed prior to final publication.*

References:
None.

Definitions:
None.

Abbreviations:
None.

Revision History:

Draft 0.1, February 2002	*Initial draft document for RPR WG review.*
Draft 0.2, April 2002	*Revision for RPR WG review.*
Draft 0.3, June 2002	*Revised draft for TF review.*
Draft 1.0, August 2002	*Revised draft for TF review.*
Draft P1.2, Novemver 2002	*Rewrite for WG review.*

Editors' Notes: *To be removed prior to final publication.*

OIF references have been retained for draft 1.0. Editor-in-chief is working with the IEEE project editor to determine whether these references will be permitted for the standard.

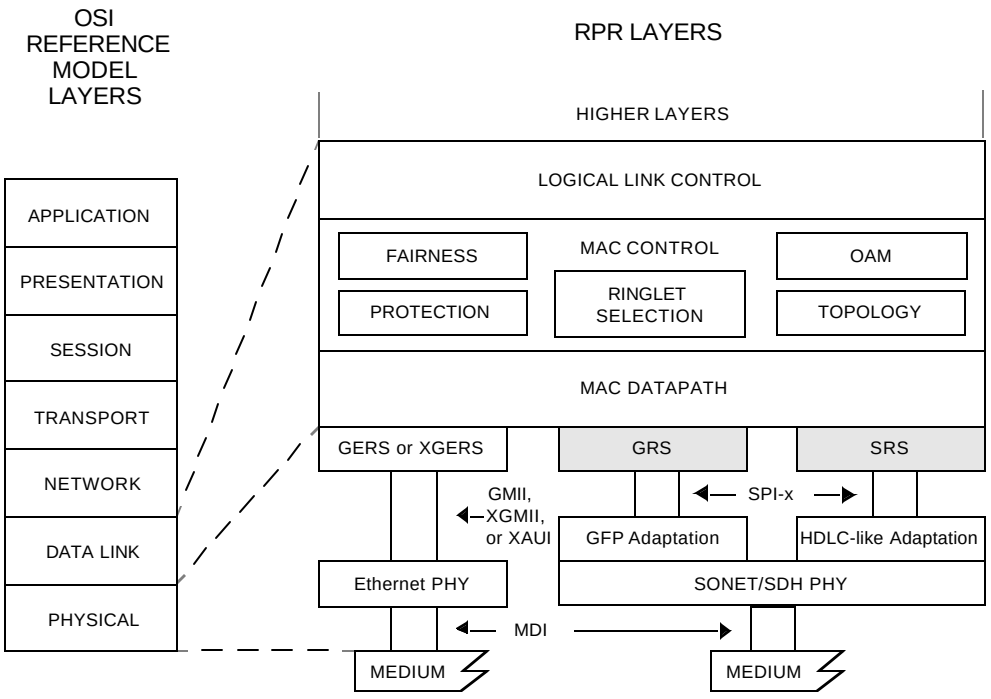
<$chapnum>.1 Overview

<$chapnum>.1.1 Scope

This annex defines two families of reconciliation sublayers for use with SONET/SDH physical layer entities (PHYs). The first is the SONET/SDH Reconciliation Sublayer (SRS), which maps the logical primitives at the RPR MAC physical layer service interface to 8-bit SPI-3, 32-bit SPI-3, SPI-4 Phase 1, and SPI-4 Phase 2 interfaces. The second is the GFP Reconciliation Sublayer (GRS), which also maps the logical primitives at the RPR MAC physical layer service interface to the same System Packet Interface (SPI) interfaces. The SRS is intended for use with GFP without core header insert or for byte-synchronous HDLC-like framing adaptation sublayers. The GRS is intended for use only with a GFP adaptation sublayer with core header insert. The SRS and GRS are identical, except that the GRS conveys packet length information to the GFP adaptation sublayer thus eliminates the need to calculate this parameter in the PHY device and incur a store and forward delay.

<$chapnum>.1.2 Relationship to other sublayers

The relationship of the SRS and GRS to RPR sublayers and to SONET/SDH physical layers is shown in Figure D.1.



GMII = GIGABIT MEDIA INDEPENDENT INTERFACE
GERS = GIGABIT ETHERNET RECONCILIATION SUBLAYER
GRS = GFP RECONCILIATION SUBLAYER
MAC = MEDIA ACCESS CONTROL
MDI = MEDIUM DEPENDENT INTERFACE
PHY = PHYSICAL LAYER ENTITY
SRS = SONET/SDH RECONCILIATION SUBLAYER
XAUI= 10 GIGABIT ATTACHMENT UNIT INTERFACE
XGMII = 10 GIGABIT MEDIA INDEPENDENT INTERFACE
XGERS = 10 GIGABIT ETHERNET RECONCILIATION SUBLAYER

Figure <$chapnum>.1—Reconciliation sublayer relationship to the ISO/IEC Open Systems Interconnection (OSI) reference model

The expected function in the GFP and HDLC like Adaptation sublayer in the PHY device is to perform Idle insertion in case of no packet to send on the medium in the transmit direction. Idle deletion and packet delineation in the receive direction.

Across the SPI interface only delineated packets are transmitted.

<$chapnum>.1.3 SRS and GRS Interfaces

This annex describes four SRS and GRS implementations using the following electrical interfaces specified by the Optical Interworking Forum (OIF):

a) SPI Level 3 (SPI-3) using 8-bit transmit and receive data paths and operating at 155 Mbps to 622 Mbps;
b) SPI Level 3 (SPI-3) using 32-bit transmit and receive data paths and operating at 155 Mbps to 2.5 Gbps;
c) SPI Level 4 Phase 1 (SPI-4.1) operating at 200 Mbps to 10 Gbps;
d) SPI Level 4 Phase 2 (SPI-4.2) operating at 622 Mbps to 10 Gbps.

The SRS and GRS for each of the interfaces are electrically identical, except that the GRS conveys packet information. The SRS does not convey packet information

The SPI interfaces support multi-PHY devices. In the context of a single MAC, each MAC datapath entity is connected to one instance of a physical device. Multi-PHY support is discussed in this clause and can be supported if the SPI interface can handle the aggregate BW of the attached ringlets.

The figure below shows designation of the SPI interface with each ringlet connected through a separate SPI interface to its matching MAC datapath entity.

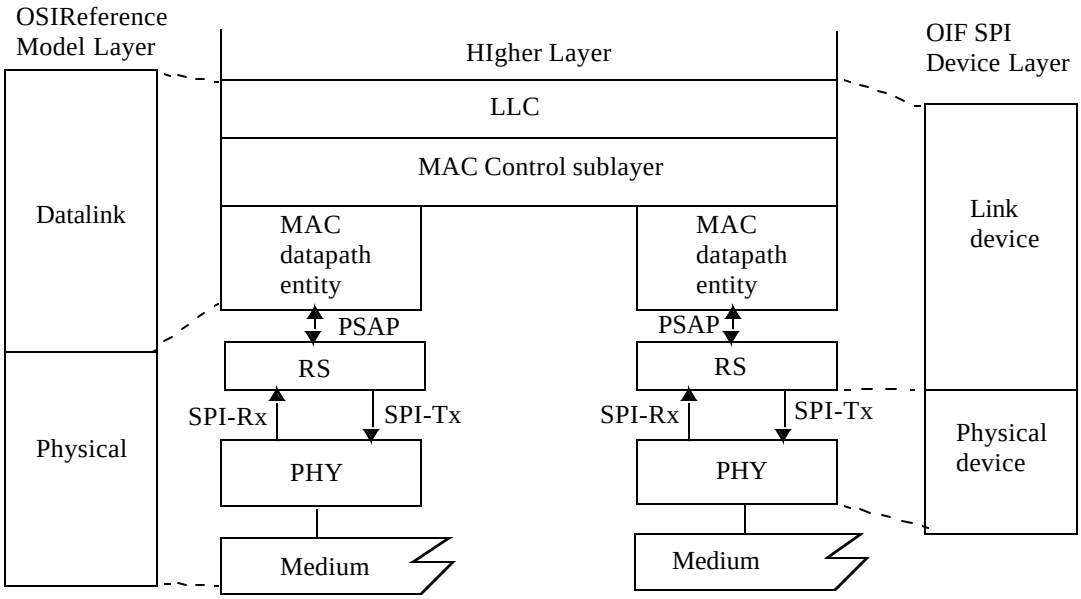


Figure <$chapnum>.2—OSI and OFI reference model layers

<$chapnum>.1.4 Link status signals

In addition to the SPI-3 interface, signal_fail and signal_degrade signals are used to convey link status from the PHY to the reconciliation sublayer.

The signal_fail and signal_degrade signals may not be directly available from the PHY. The RS provides the physical interface and map to physical layer interface primitives, PHY_LINK_STATUS.indicate. A MAC implementation is required to generate these signals based on the specific PHY implementation.

<$chapnum>.1.4.1 Signal_fail

The reconciliation sublayer shall include a signal_fail input. Signal_fail asserted indicates a PHY signal fail condition depending on the medium and the physical medium dependent transmission method signal fail is a loss of optical signal, or loss of carrier. These are just some attributes of signal_fail and it is not comprehensive.

<$chapnum>.1.4.2 Signal_degrade

The reconciliation sublayer shall include a signal_degrade input. Signal_degrade asserted indicates a PHY signal degrade condition. If the PHY provides a signal_degrade status this input is used otherwise, it is not connected and shall be tied to inactive.

<$chapnum>.1.4.3 Mapping of Link Status signals to service interface primitives

The reconciliation sublayer shall map the signals provided at the physical interface to the MAC physical layer service interface primitives defined in Clause 7. Mappings for the following primitives are defined:

— PHY_LINK_STATUS.indicate

The primitive is generated by the physical signal and through the layer management interface. The layer management interface takes precedence over the physical pins.The LINK_STATUS attribute takes on the values: OK, FAILED, and DEGRADED as describe in Clause 7.2.3.2.

Editors' Notes: *To be removed prior to final publication.*

change clause to a cross-reference.

<$chapnum>.1.4.3.1 Mapping of Layer Management MDSF and MDSD PHY_LINK_STATUS.indicate

PHY_LINK_STATUS.indicate (LINK_STATUS)

Editors' Notes: *To be removed prior to final publication.*

coordinate with Layer Management section.

Map the primitive PHY_LINK_STATUS.indicate to the STA signals MDSF and MDSD.

The RS generates the PHY_LINK_OK.indicate primitives whenever the LINK_STATUS parameter changes from one possible value (OK, FAIL, DEGRADE) to a different possible value.

The LINK_STATUS values are derived from the MDSF and MDSD signals received from the Layer Management asynchronously from the data transfer.

The LINK_STATUS parameter assumes the value OK when neither MDSF nor MDSD signal has been received.

It assumes the value FAIL when the MDSF signal is asserted.

The value DEGRADE is assumed when the MDSD signal is asserted and the MDSF has not been received.

<$chapnum>.1.4.3.2 Mapping Physical Signals to PHY_LINK_STATUS.indicate

PHY_LINK_STATUS.indicate (LINK_STATUS)

There are two physical pins that receives PHY link status if provided by the PHY. Thed two physical siganls are Signal_Fail and Siganl_Degrade.

The two physical link status input maps the signals at these pins to the LINK_STATUS attribute.

Table 0.1—

Signal_Fail	Signal Degrade	LINK_STATUS
inactive	inactive	OK
inactive	active	DEGRADE
active	inactive	FAIL
active	active	DEGARDE and FAIL

<$chapnum>.1.4.3.2.1 When generated

This primitive is invoked by the RS after it determined that the PHY is not performing properly. For example, the received RFCLK has failed. This would cause the SPI receive interface to not be able to receive data. The LINK_STATUS FAIL is also generated up on reception of signals from the PHY device which it has generated to indicate that the local physical sublayer has failed.

This signal is indicated as signal_fail and signal_degrade in Figure <$chapnum>.6—

<$chapnum>.1.4.3.3 Effect of receipt

The effect of receiving this primitive by the MAC sublayer is specified in clause 6.

<$chapnum>.1.5 Electrical specifications

The signal_degrade and signal_fail are intended to be DC status signals. The signal level is LVTTL with threshold dependent on the I/O operating voltage..

Editors' Notes: *To be removed prior to final publication.*

change clause to a cross-reference.

<$chapnum>.2 Physical Frame Format for SRS and GRS

The physical electrical interface for the GRS and the SRS is identical for both data and control signals. The difference in GFP adaptation and the HDLC like-adaptation layer is the delineation mechanism and hence different frame format, specifically the header prefix.

<$chapnum>.2.1 SRS Physical Frame Format

In the HDLC-like adaptation interface, this sublayer inserts the start of frame flag, $7E_{16}$, and at least one escape flag to mark the end of packet. For consecutive packet transmission the minimum requirement is one escape flag to separate the two packets.

The HDLC-like adaptation sublayer, thus has to perform escaping for any $7E_{16}$ occurrence in the packet. The packet transmitted across this interface is the 802.17 frame as specified in Clause 8.

Editors' Notes: *To be removed prior to final publication.*

Change Clause 8 to a Frame cross_reference variable.

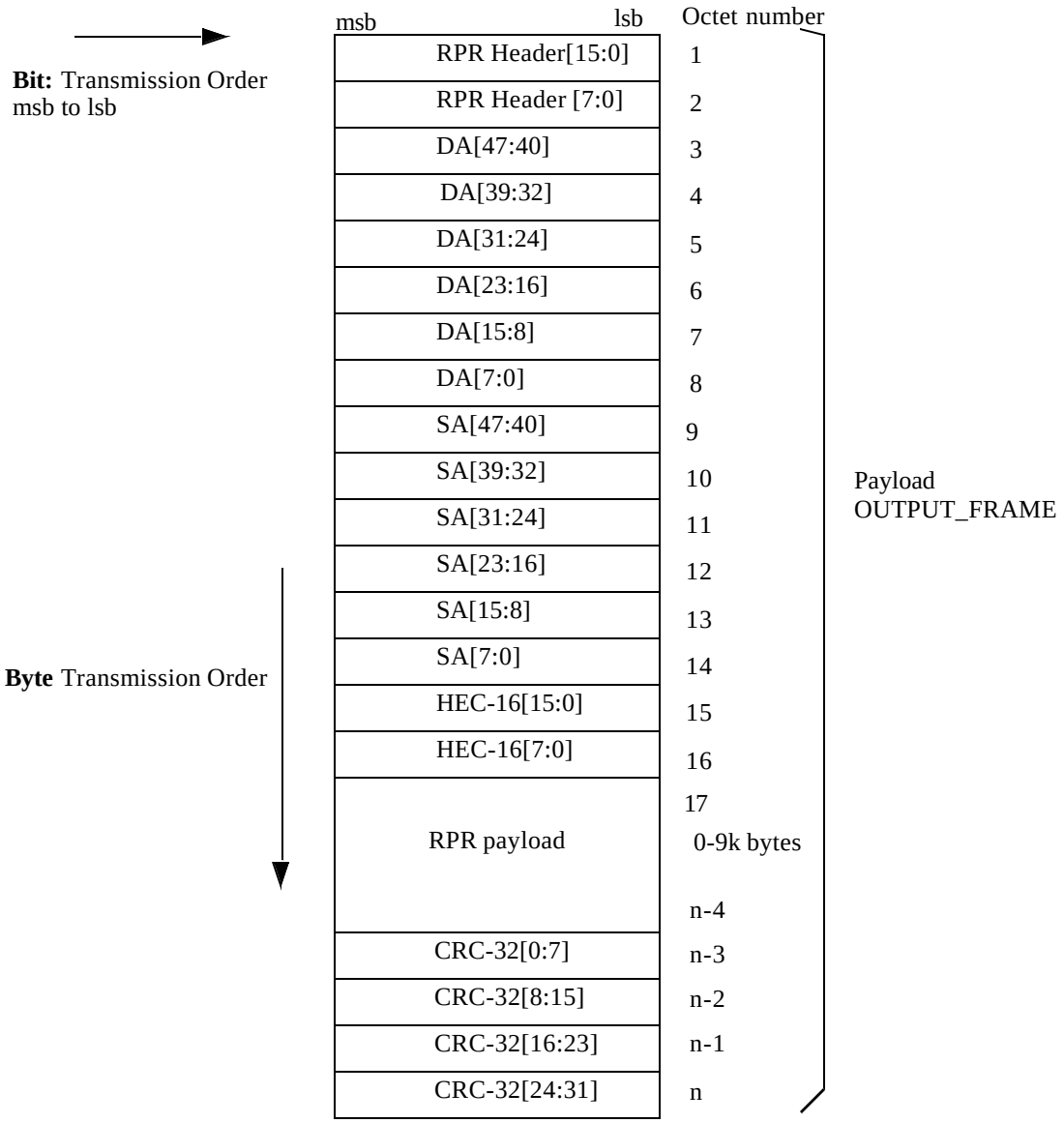


Figure <$chapnum>.3—SRS Frame Structure.

Editors' Notes: *To be removed prior to final publication.*

update frame format if changes in clause 8.

<$chapnum>.2.2 GRS Physical Frame Format

The GFP delineation method provides a deterministic inflation factor that is not packet content sensitive. It uses a length value protected by a cHEC to provide robust delineation under degrade channel condition. The HEC provides single error correction. In the case of GRS interface, the length and the HEC fields are part of the packet transmitted across the physical interface. The frame format is:

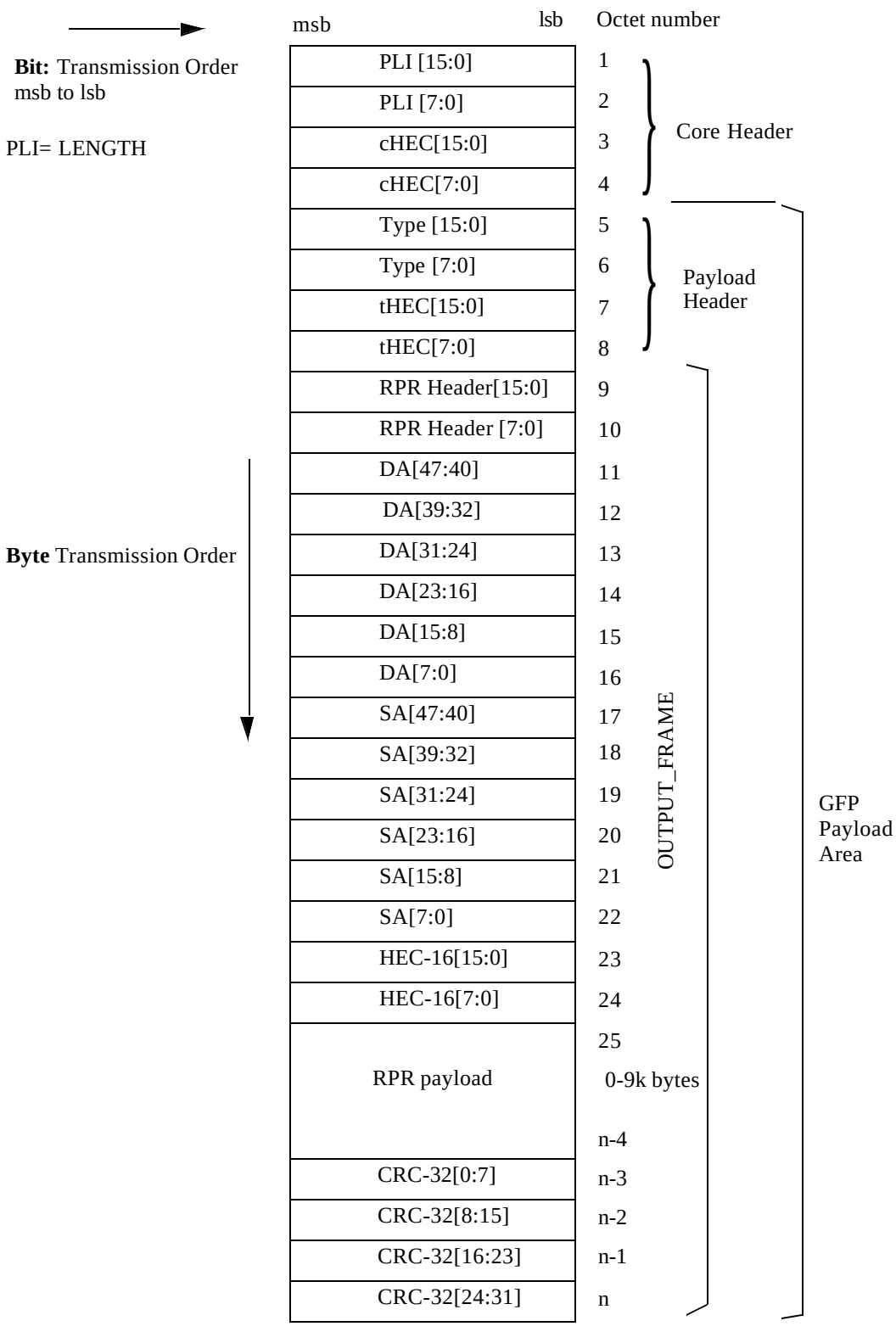


Figure <$chapnum>.4—GRS Frame structure. Showed without GFP payload FCS

The length field identifies the number of bytes from the RPR header to the end of CRC-32, inclusive.

The calculation of the cHEC is optional as the GFP adaptation sublayer can insert the actual cHEC value. If the GRS does not calculate the cHEC then it shall insert zero's.

The type field is constructed in the RS and prepends the OUTPUT_FRAME.

Operating in the GRS mode, if the length primitive is not received across the PSAP interface, a null value across the logical interface, the GRS shall generate the required GFP core header field. Alternatively, the SRS is used, then the GFP adaption sublayer generates the required GFP headers.

The calculation of the HEC uses the polynomial of:

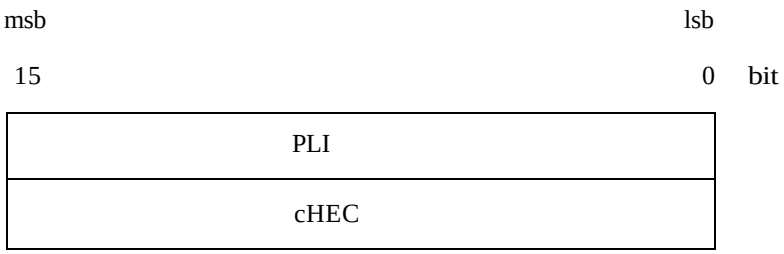
$$CRC - 16 = x^{16} + x^{12} + x^5 + 1$$

Depending on the filed the HEC protects over, it is called cHEC or tHEC.

<$chapnum>.2.2.1 Payload Length Identifier

The GFP core header consists of 2 fields: payload length indicator (PLI) and cHEC.

The core header supports frame delineation. The core header consists of two fields 1) payload length indicator and a cHEC. The PLI is a 16 bits binary representation of the decimal number in octets of the payload area. The cHEC is calculated over the PLI only.



<$chapnum>.2.2.2 GFP Payload Header

The basic GFP payload header consists of two fields: type and tHEC. For RPR application the non-extension type payload header format is used. The tHEC is calculated over the type field only.

The GFP type field consists of the following sub-fields: 1) a 3 bit payload type identifier (PTI), 2) a 1-bit payload FCS indicator (PFI), 3) a 4-bit extension header identifier and 4) an 8-bit user payload identifier (UPI).

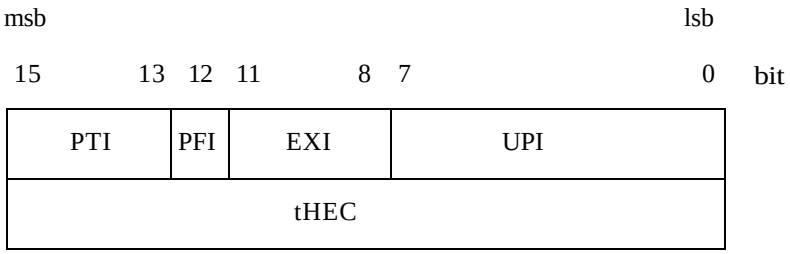


Figure <$chapnum>.5—

The definitions of the values are defined by the ITU-T G.7041 document. The recommend values for the above fields are:

Field	*Recommended value*	*Comments*
PTI	*000_2*	*Client data*
PFI	*0*	*absent of GFP FCS*
EXI	*0000_2*	*null extension header*
UPI	*See Note*	*RPR user payload type*

Note: The assignment of this number is by ITU-T. Refer to ITU-TG.7041.

Editors' Notes: *To be removed prior to final publication.*

At the time this draft is written, there is no value assigned for RPR user payload.

<$chapnum>.3 SRS and GRS using the 8-bit SPI-3 interface

The SONET/SDH Reconciliation Sublayer and GFP Reconciliation Sublayer (GRS) may be implemented with an 8-bit OIF SPI-3 interface.

<$chapnum>.3.1 General requirements

The clauses in this section provide a brief description of the signals for the SPI3-01.0 interface and their interaction with the MAC physical layer service interface primitive.

Any discrepancy in the signal description the OIF-SPI3-01.0 takes precedence.

<$chapnum>.3.1.1 Summary of major concepts

a) The SRS and GRS map the signals provided at the 8-bit SPI-3 interface to the logical physical layer service interface primitives provided at the MAC;
b) Each direction of data transfer is independent, and serviced by 8-bit data, delimiter, error, and clock signals;
c) Data and delimiter are synchronous to clock references;

d) The SPI-3 interface supports logical channels with individual word-level or packet-level out of band flow control;
e) The SRS and GRS using the 8-bit SPI-3 interface support full-duplex operation only.

<$chapnum>.3.1.2 Rate of operation

The SRS and GRS using the 8-bit SPI-3 interface are capable of supporting data rates of 155 Mbps to 622 Mbps, with an effective operation granularity of 155 Mbps. This is not an interface limit but the result of physical coding which maps to increments of STS1 or STS-3 SONET rates and increments of STM-1 for SDH rates.

SONET/SDH PHYs that provide an SPI-3 with 8-bit data path shall support operations, at the SONET/SDH Path level, at the selected rate on the SPI-3. PHYs reports the rates at which they are operating via the management interface. The operation mode of the PHY is expected operate in streaming, flow through or cut through mode.

<$chapnum>.3.1.3 8-bit SPI-3 structure

The 8-bit SPI-3 interface is composed of independent transmit and receive paths. The MAC behaves as a link layer device.

SPI interfaces can operate in byte level or packet level mode. In byte-level transfer, the FIFO status information is presented on a cycle by cycle basis. A PHY device indicates the transmit packet available status via the selected or directed method signals, STPA and DTPA[3:0] respectively for multi-phy case. DTPA signals are provided for simpler implementation to reduce the need for addressing. Support for one or the other is optional but one is to be supported to ensure the PHY FIFO does not overrun.

In packet-level transfer, the FIFO status information applies to segments of packet data. The RS polls the status of the PHY for transmit ready status. The RS polls one of multiple available PHYs by asserting the TADR address of the PHY which response via a common PTPA signal.

Additional signals TENB, RENB, and RVAL are used by the protocol to indicate valid data on the bus.

SPI-3 in each direction uses 8 data signals (TDAT<7:0> and RDAT<7:0>), clocks (TFCLK and RFCLK), start of packet delimiter signals (TSOP and RSOP), and end of packet delimiter signals (TEOP and REOP). These signals are described in <$chapnum>.3.1.3.1 through <$chapnum>.3.1.3.20, and are fully defined in the OIF SPI-3 implementation agreement document.

Figure<$chapnum>.6 shows a schematic view of the SRS and GRS inputs and outputs using the 8-bit SPI-3 interface

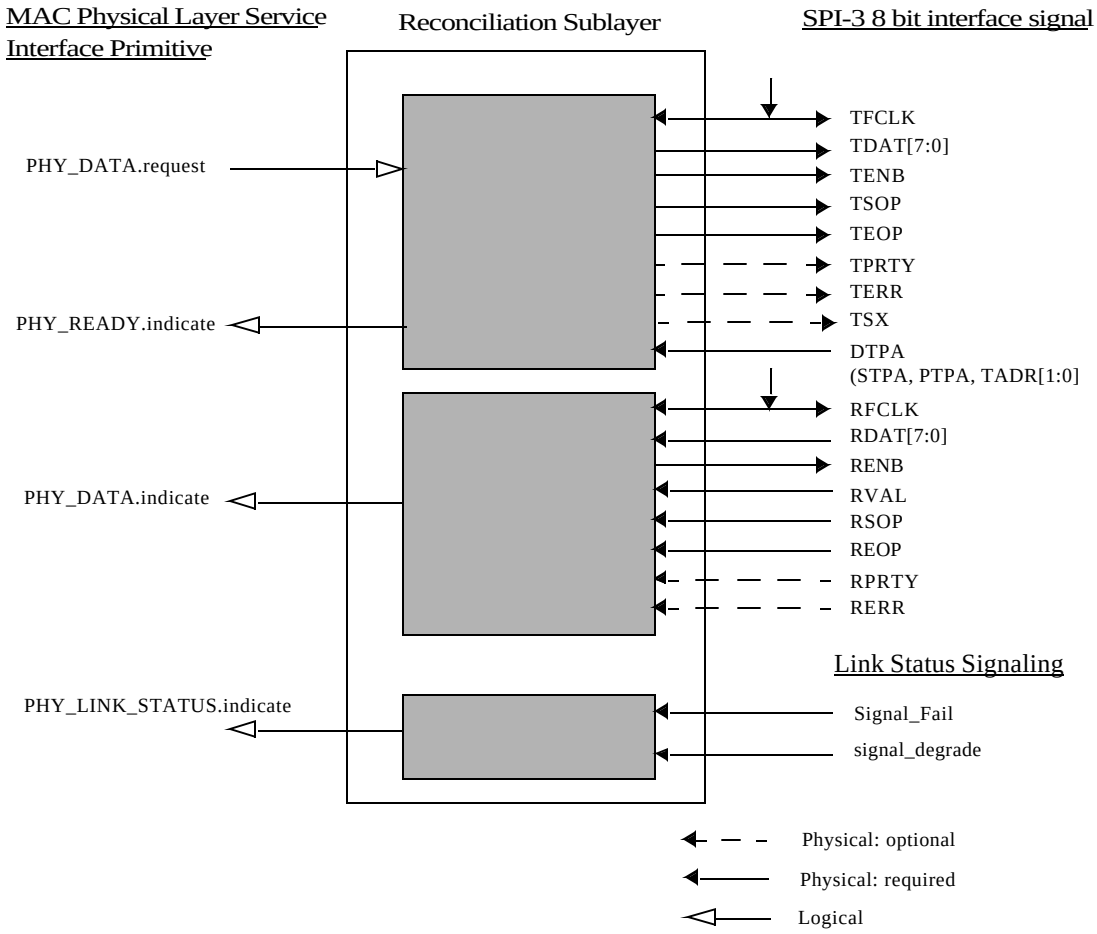


Figure <$chapnum>.6—SRS and GRS inputs and outputs using 8-bit SPI-3

The 8-bit SPI-3 interface signals are described in the following subclauses. All transmit signals are expected to be updated and sampled using the rising edge of the transmit clock, TFCLK.

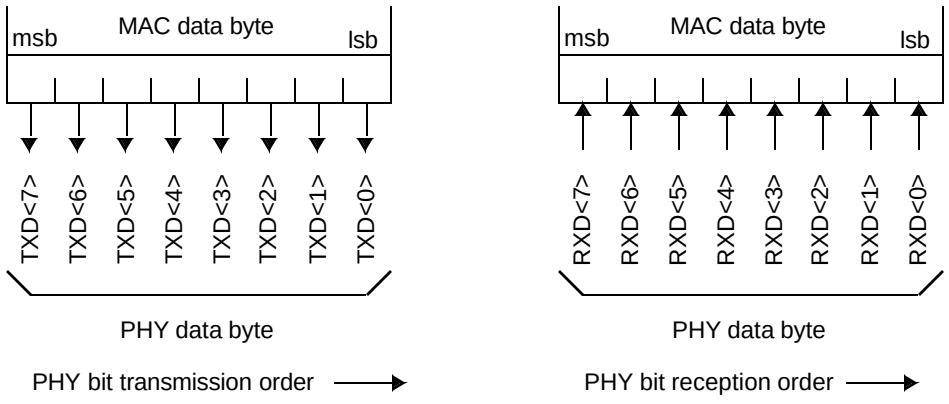


Figure <$chapnum>.7—8-bit SPI-3 transmit and receive signal mapping

<$chapnum>.3.1.3.1 TFCLK (transmit clock)

TFCLK is a continuous clock used to synchronize data transfer transactions between the RS and the PHY layer device. TFCLK may cycle at a rate up to 104 MHz. TFCLK is externally sourced to the RS and to the PHY.

<$chapnum>.3.1.3.2 TERR (transmit error Indicator)

The use of TERR is optional. There is no direct mapping from the PSAP primitives.

This is internal to the RS. When a packet is being transmitted and an internal error occurs then TERR is used to indicate that there is an error in the current packet. TERR should only be asserted when TEOP is asserted; it is considered valid only when TENB is asserted.

<$chapnum>.3.1.3.3 TENB (transmit write enable)

This is a physical interface protocol signal for indicate valid symbol transfer at the interface. The generation of this signal is as result the reception of a PHY_DATA.request and while valid data symbols are being transmitted across the physical interface.

When TENB is high the TDAT, TMOD, TSOP, TEOP and TERR signals are invalid and should be ignored by the PHY. The TSX signal is valid and is processed by the PHY when TENB is high. When TENB is low the TDAT, TMOD, TSOP, TEOP and TERR signals are valid and are processed by the PHY. The TSX signal is ignored by the PHY when TENB is low.

<$chapnum>.3.1.3.4 TDAT[7:0] (transmit packet data bus)

The generation of this signal is as result the reception of a PHY_DATA.request. The OUTPUT_FRAME and LENGTH primitives are formatted into the appropriate frames and the content of the frame are then transmitted across the physical interface.

The OUTPUT_FRAME is mapped into the payload for both SRS and GRS. GRS shall map valid LENGTH primitive into the PLI field of the GFP core header.

TDAT is an 8-bit bus which carries the packet bytes that are written to the selected transmit FIFO and the in-band port address to select the desired PHY transmit FIFO. The TDAT bus is considered valid only when TENB is asserted. Data is transmitted msb first. Mapping of TDAT signals is shown in Figure<$chapnum>.7.

<$chapnum>.3.1.3.5 TPRTY (transmit bus parity)

The use of TPRTY is optional and it is interface specific. There is no direct mapping to the PSAP primitives.

TPRTY indicates the parity calculated over the TDAT bus. TPRTY is considered valid only when TENB or TSX is asserted. The parity calculation is such that odd parity is indicated on this signal.

<$chapnum>.3.1.3.6 TSX (transmit start of transfer)

The use of TSX is optional and it is interface specific. There is no direct mapping from the PSAP primitives.

The use of TSX applies when multi-port PHY application.

TSX indicates when the in-band port address is present on the TDAT bus. When TSX is high and TENB is high, the value of TDAT is the address of the transmit FIFO to be selected. Subsequent data transfers on the

TDAT bus will fill the FIFO specified by this in-band address. For single channel PHY devices the TSX signal is optional. TSX is considered valid only when TENB is not asserted.

<$chapnum>.3.1.3.7 TSOP (transmit start of packet)

This is a physical interface protocol signal for flow control at the interface level. The generation of this signal is as result the reception of a PHY_DATA.request and while valid data symbols are being transmitted across the physical interface.

TSOP is used to delineate the packet boundaries on the TDAT bus. When TSOP is high the start of the packet is present on the TDAT bus. TSOP is required to be present at the beginning of every packet and is considered valid only when TENB is asserted.

<$chapnum>.3.1.3.8 TEOP (transmit end of packet)

This is a physical interface protocol signal for flow control at the interface level. The generation of this signal is as result the reception of a PHY_DATA.request and while the last valid data symbols are being transmitted across the physical interface.

TEOP is used to delineate the packet boundaries on the TDAT bus. When TEOP is high, the end of the packet is present on the TDAT bus. TEOP is required to be present at the end of every packet and is considered valid only when TENB is asserted.

<$chapnum>.3.1.3.9 TADR (transmit PHY address) and PTPA (polled-PHY transmit packet available)

The use of TADR is optional. It is for multi-port PHY applciation. TADR is used in the packet transfer mode. There is no direct mapping from the PSAP primitives.

TADR bus is used with the PTPA signal to poll the transmit FIFO's packet available status. When TADR is sampled on the rising edge of TFCLK by the PHY the polled packet available indication PTPA is updated with the status of the channel specified by the TADR address on the following rising edge of TFCLK.

The result of the polled status PTPA is used to generate PHY_READY.indicate.

<$chapnum>.3.1.3.10 DTPA (direct transmit packet available)

DTPA is used in the byte transfer mode.

The result of DPTA is used to generate PHY_READY.indicate.

DTPA bus provides direct status indication for the corresponding ports in the PHY device. DTPA transitions high when a predefined minimum number of bytes is available in its transmit FIFO. Once high, the DTPA signal indicates that its corresponding transmit FIFO is not full. When DTPA transitions low it indicates that its transmit FIFO is full or near full. DTPA is updated on the rising edge of TFCLK.

<$chapnum>.3.1.3.11 STPA (selected transmit packet available)

The use of STPA is optional. It is for multi-port PHY applciation. STPA is used in the byte transfer mode.

The result of DPTA is used to generate PHY_READY.indicate.

STPA provides status indication of the selected port of a PHY device in order to avoid PHY FIFO overflow while polling is performed. The port which STPA reports is updated on the following rising edge of TFCLK after the PHY address on the TDAT is sampled by the PHY.

<$chapnum>.3.1.3.12 RFCLK (receive FIFO write clock)

RFCLK is a continuous clock used to synchronize data transfer transactions between the RS and the PHY. RFCLK may cycle at a rate up to 104 MHz.

<$chapnum>.3.1.3.13 RVAL (receive data valid)

RVAL indicates the validity of the receive data. RVAL is low between transfers and when RSX is asserted. It is also low when the PHY pauses a transfer due to an empty PHY receive FIFO. When a transfer is paused by holding RENB high RVAL will hold its value unchanged although no new data will be present on RDAT until the transfer resumes. When RVAL is high the RDAT, RMOD, RSOP, REOP and RERR signals are valid. When RVAL is low, the RDAT, RMOD, RSOP, REOP and RERR signals are invalid and shall be disregarded. The RSX signal is valid when RVAL is low.

<$chapnum>.3.1.3.14 RENB (receive read enable)

The RENB signal is used to control the flow of data from the PHYs. During data transfer, RVAL shall be monitored as it will indicate if the RDAT, RPRTY, RMOD, RSOP, REOP, RERR and RSX are valid. The system may deassert RENB at anytime if it is unable to accept data from the PHY device. When RENB is sampled low by the PHY device a read is performed from the receive FIFO and the RDAT, RPRTY, RMOD, RSOP, REOP, RERR, RSX and RVAL signals are updated on the following rising edge of RFCLK. When RENB is sampled high by the PHY device a read is not performed and the RDAT, RPRTY, RMOD, RSOP, REOP, RERR, RSX and RVAL remain unchanged on the following rising edge of RFCLK.

The MAC is expected to receive data from the PHY without applying back pressure.

<$chapnum>.3.1.3.15 RDAT[7:0] (receive packet data bus)

RDAT is an 8-bit bus which carries the packet bytes that are read from the PHY and the in-band port address of the selected receive port. RDAT is considered valid only when RVAL is asserted. Mapping of RDAT signals is shown in Figure <$chapnum>.6.

Valid data received on the RDAT bus from RSOP to REOP are mapped into PHY_DATA.indicate.

<$chapnum>.3.1.3.16 RPRTY (receive parity)

The use of RPTRY is optional as it does not affect the flow of data from the PHY to the RS. This is used for interface integrity checking. Parity error detected does not cause the received packet to be discarded.

RPRTY signal indicates the odd parity calculated over the RDAT bus.

<$chapnum>.3.1.3.17 RSOP (receive start of packet)

RSOP is used to delineate the packet boundaries on the RDAT bus. When RSOP is high, the start of the packet is present on the RDAT bus. RSOP is required to be present at the beginning of every packet and is considered valid only when RVAL is asserted.

Valid data received on the RDAT bus from RSOP to REOP are mapped into PHY_DATA.indicate.

<$chapnum>.3.1.3.18 REOP (receive end of packet)

REOP is used to delineate the packet boundaries on the RDAT bus. When REOP is high, the end of the packet, the last byte is present on the RDAT bus. REOP is required to be present at the end of every packet and is considered valid only when RVAL is asserted.

Upon receiving an REOP, the data received from RSOP constitute a received frame. For SRS sublayer, the received frame formulates the INPUT_FRAME attribute of the PHY_DATA.indicate primitive. For a GRS sublayer, the first 2 bytes of the received frame is mapped into the LENGTH attribute and the GFP paylaod less the header is mapped into the INPUT_FRAME attribute of the PHY_DATA.indicate primitive.

<$chapnum>.3.1.3.19 RERR (receive error indicator)

The use of RERR is optional, as it does not interfere with the transfer of the packet.

RERR is used to indicate that the current packet is in error. RERR shall only be asserted when REOP is asserted. Conditions that can cause RERR to be set may be, but are not limit to, PHY FIFO overflow, abort sequence detection and PHY state machine error detected.

<$chapnum>.3.1.3.20 RSX (receive start of transfer)

The use of RSX is optional, when a multi-port PHY device and in-band addressing scheme is used.

RSX indicates when the in-band port address is present on the RDAT bus. When RSX is high, the value of RDAT[7:0] is the address of the receive FIFO to be selected by the PHY. Subsequent data transfers on the RDAT bus will be from the FIFO specified by this in-band address. For single channel PHY devices the RSX signal is optional. For multi-port PHY devices, RSX is asserted at the beginning of each transfer. When RSX is high RVAL shall be low.

<$chapnum>.3.1.4 Mapping of SPI-3 signals to service interface primitives

The reconciliation sublayer shall map the signals provided at the SPI-3 interface to the MAC physical layer service interface primitives defined in Clause 7. Mappings for the following primitives are defined:

— PHY_DATA.request
— PHY_DATA.indicate
— PHY_READY.indicate.

<$chapnum>.3.1.4.1 Mapping of PHY_DATA.request

PHY_DATA.request (OUTPUT_FRAME, LENGTH

There are two attributes: OUTPUT_FRAME and LENGTH

The OUTPUT_FRAME is mapped into the payload. The LENGTH applies to GRS only, it is mapped into the PLI field of the GFP core header.

<$chapnum>.3.1.4.1.1 When generated

This primitive is invoked by the MAC_datapath entity when it is performed its tasks after it had received a received a request from its client.

<$chapnum>.3.1.4.1.2 Effect of receipt

The receipt of this primitive shall cause the RS to generate the physical frame format required operating in GRS or SRS mode as described in <$chapnum>.2.1 and <$chapnum>.2.2 . The RS will start transferring the physical frame to the PHY in according to SPI-3 protocol: generating TSOP, TDAT, TMOD, and TEOP signals.

<$chapnum>.3.1.4.2 Mapping of PHY_DATA.indicate

PHY_DATA.indicate (INPUT_FRAME, STATUS, LENGTH)

There are three attributes: INPUT_FRAME, STATUS, and LENGTH

The INPUT_FRAME is mapped from the received payload. The LENGTH applies to GRS only. It is mapped from the PLI field in the GFP core header.

The STATUS parameter takes the value of OK or ERROR and is used to identify frames that are receive with error indications from the PHY, for example, due to assertion of a error signal on the PHY electrical interface. A value of OK indicates no error indication. A value of ERROR indicates that the PHY signaled an error during frame transmission to the reconciliation sublayer.

<$chapnum>.3.1.4.3 When generated

This primitive is invoked by the RS after it had received a complete frame from the physical interface. The INPUT_FRAME is derived from the signals RDAT[31:0], RMOD[1:0], as signalled by the status RSOP, REOP, RSX, in accordance to SPI-3 specification.

The STATUS generates OK unless an active error signal is received on RERR.

<$chapnum>.3.1.4.4 Effect of receipt

The effect of receiving this primitive by the MAC sublayer is specified in clause 6.

<$chapnum>.3.1.4.5 Mapping of PHY_READY.indicate

PHY_READY.indicate (READY_STATUS

The READY_STATUS attribute takes on the3 value of READY or NOT_READY.

<$chapnum>.3.1.4.5.1 When generated

Depending on single or multiple PHY mode, STPA, PTPA, or DPTA[], are indications from the PHY to the RS that the PHY is ready to receive data on the TDAT bus. Valid indication from the PHY is mapped to the PHY_READY.indicate primitive and send to RS client.

<$chapnum>.3.1.4.6 Effect of receipt

The effect of receiving this primitive by the MAC sublayer is specified in clause 6.

<$chapnum>.3.2 SRS and GRS 8-bit SPI datastream

There are no electrical difference and signalling protocol between the GRS and the SRS sublayer. The only difference is the physical frame format as specified in <$chapnum>.2.1 and <$chapnum>.2.2 .

<$chapnum>.3.3 Functional specifications

The SRS and GRS using 8-bit SPI-3 interfaces shall meet the functional requirements of paragraphs 8.4, 9, 10 and 11 of the OIF SPI-3 implementation agreement.

<$chapnum>.3.4 Electrical specifications

The SRS and GRS using 8-bit SPI-3 interfaces shall meet the transmit electrical timing requirements of paragraph 10.3, and the receive electrical timing requirements of paragraph 11.3, of the OIF SPI-3 implementation agreement.

The electrical characteristics for signal_degrade and signal_fail are LVTTL compatible with the supply voltage of the SPI-3 bus.

<$chapnum>.4 SRS and GRS using the 32-bit SPI-3 interface

The SONET/SDH Reconciliation Sublayer and GFP Reconciliation Sublayer (GRS) may be implemented with a 32-bit OIF SPI-3 interface.

<$chapnum>.4.1 General requirements

This clause provides a brief description of the signals for the SPI3-01.0 interface and their interaction with the MAC physical layer service interface primitive.

Any discrepancy in the signal description the OIF-SPI3-01.0 takes precedence.

<$chapnum>.4.1.1 Summary of major concepts

a) The SRS and GRS map the signals provided at the 32-bit SPI-3 interface to the logical physical layer service interface primitives provided at the MAC;
b) Each direction of data transfer is independent, and serviced by 32-bit data, delimiter, error, and clock signals;
c) Data and delimiter are synchronous to clock references;
d) The SPI-3 interface supports logical channels with individual word-level or packet-level out of band flow control;
e) The SRS and GRS using the 32-bit SPI-3 interface support full-duplex operation only.

<$chapnum>.4.1.2 Rate of operation

The SRS and GRS using the 32-bit SPI-3 interface are capable of supporting data rates of 155 Mbps to 2.5 Gbps. The granularity is dependent on the mapping of the SONET/SDH synchronous payload. The interface clock is set fast enough so as the PHY transmit FIFO does not starve in and the PHY receive FIFO does not overrun

SONET/SDH PHYs that provide an SPI-3 with 32-bit data path shall support operations, at the SONET/SDH Path level, at the selected rate on the SPI-3.PHYs reports the rates at which they are operating via the management interface. The operation mode of the PHY is expected operate in streaming, flow through or cut through mode.

<$chapnum>.4.1.3 32-bit SPI-3 structure

SPI interface can operate in byte level or packet level mode. In byte-level transfer, the FIFO status information is presented on a cycle by cycle basis. The PHY device indicates the transmit packet available status via the STPA or DTPA[3:0] signals. DTPA signals are provided for simpler implementation to reduce the need for addressing. Support for one or the other is optional but one is to be supported to ensure the PHY FIFO does not overrun.

In packet-level transfer, the FIFO status information applies to segments of packet data. The RS polls the status of the PHY for transmit ready status. The RS polls one of multiple available PHYs by asserting the TADR address of the PHY which response via a common PTPA signal.

Additional signals TENB, RENB, and RVAL are used by the protocol to indicate valid data on the bus.

The 32-bit SPI-3 interface is composed of independent transmit and receive paths. Each direction uses 32 data signals (TDAT<31:0> and RDAT<31:0>), word modulo signals (TMOD<1:0> and RMOD<1:0>), clocks (TFCLK and RFCLK), start of packet delimiter signals (TSOP and RSOP), and end of packet delimiter signals (TEOP and REOP). Mapping of the TDAT and RDAT signals is shown in These signals are fully defined in the OIF SPI-3 implementation agreement. Figure<$chapnum>.8 shows a schematic view of the SRS and GRS inputs and outputs using the 32-bit SPI-3 interface.

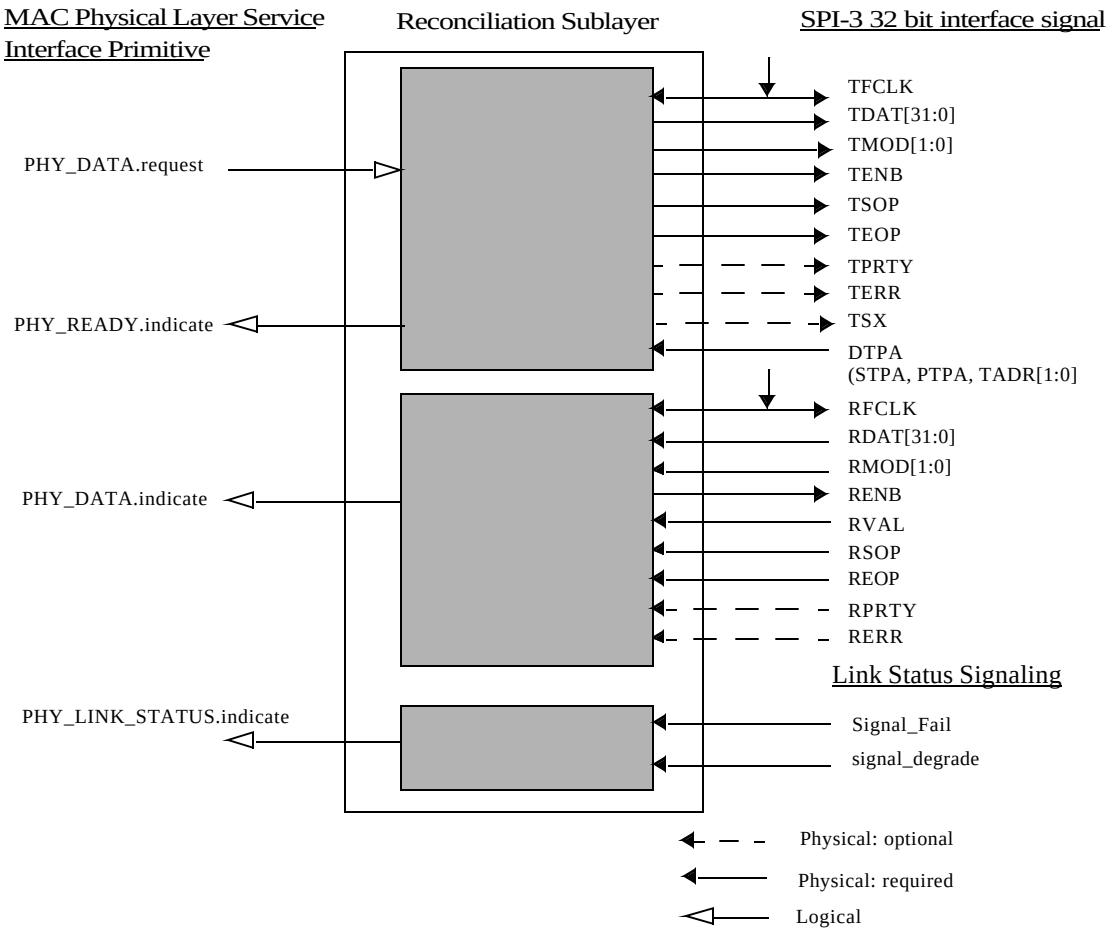


Figure <$chapnum>.8—SRS and GRS inputs and outputs using 32-bit SPI-3

The 32-bit SPI-3 interface signals are described in the following subclauses. All signals are expected to be updated and sampled using the rising edge of the transmit clock, TFCLK.

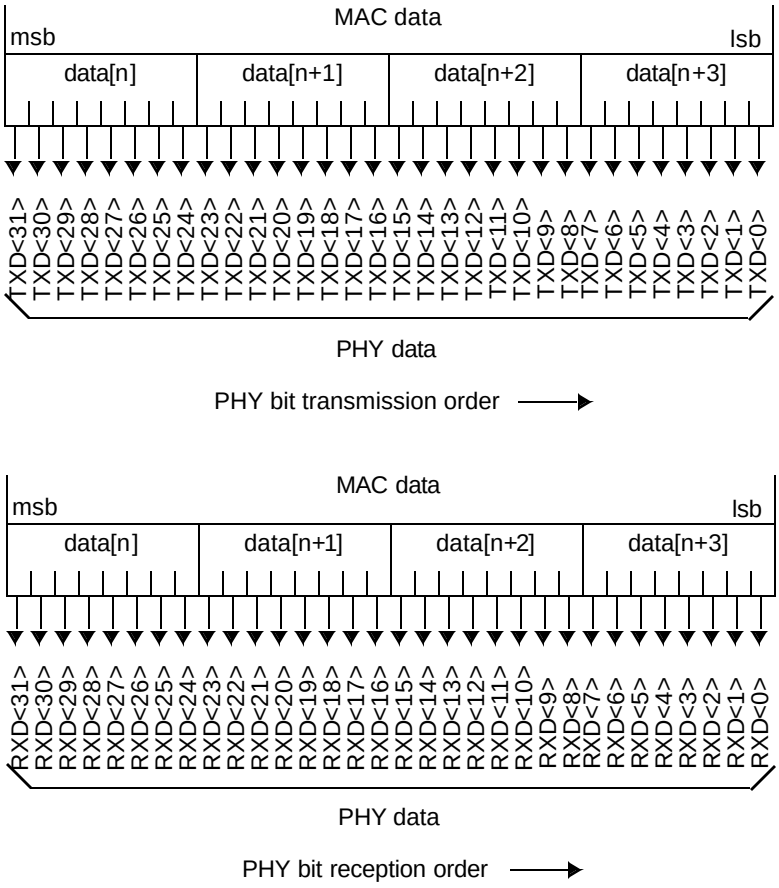


Figure <$chapnum>.9—32-bit SPI-3 transmit and receive signal mapping

<$chapnum>.4.1.3.1 TFCLK (transmit clock)

TFCLK is a continuous clock used to synchronize data transfer transactions between the RS and the PHY layer device. TFCLK may cycle at a rate up to 104 MHz. TFCLK is externally sourced to the RS and to the PHY.

<$chapnum>.4.1.3.2 TERR (transmit error Indicator)

The use of TERR is optional. There is no direct mapping from the PSAP primitives.

This is internal to the RS. When a packet is being transmitted and an internal error occurs then TERR is used to indicate that there is an error in the current packet. TERR should only be asserted when TEOP is asserted; it is considered valid only when TENB is asserted.

<$chapnum>.4.1.3.3 TENB (transmit write enable)

This is a physical interface protocol signal for flow control at the interface level. The generation of this signal is as result the reception of a PHY_DATA.request and while valid data symbols are being transmitted across the physical interface.

When TENB is high the TDAT, TMOD, TSOP, TEOP and TERR signals are invalid and should be ignored by the PHY. The TSX signal is valid and is processed by the PHY when TENB is high. When TENB is low the TDAT, TMOD, TSOP, TEOP and TERR signals are valid and are processed by the PHY. The TSX signal is ignored by the PHY when TENB is low.

<$chapnum>.4.1.3.4 TDAT[31:0] (transmit packet data bus)

TDAT is an 32-bit bus which carries the packet bytes that are written to the selected transmit FIFO and the in-band port address to select the desired transmit FIFO. The TDAT bus is considered valid only when TENB is asserted. Data is transmitted in big endian order on TDAT. Mapping of TDAT signals is shown in Figure<$chapnum>.9.

<$chapnum>.4.1.3.5 TMOD[1:0] (transmit word modulo)

TMOD is required only when TDAT is 32-bits. TMOD indicates the number of valid bytes of data in TDAT. The TMOD bus should always be zero except during the last word transfer of a packet on TDAT. When TEOP is asserted the number of valid packet data bytes on TDAT is decoded from TMOD as:

TMOD[1:0] = 00	TDAT[31:0] is valid
TMOD[1:0] = 01	TDAT[31:8] is valid
TMOD[1:0] = 10	TDAT[31:16] is valid
TMOD[1:0] = 11	TDAT[31:24] is valid

TMOD is considered valid only when TENB is asserted.

<$chapnum>.4.1.3.6 TPRTY (transmit bus parity)

TPRTY indicates the parity calculated over the TDAT bus. TPRTY is considered valid only when TENB or TSX is asserted. The parity calculation is such that odd parity is indicated on this signal. There is no direct mapping from the PSAP primitives.

<$chapnum>.4.1.3.7 TSX (transmit start of transfer)

The use of TSX is optional and it is interface specific. There is no direct mapping from the PSAP primitives.

TSX indicates when the in-band port address is present on the TDAT bus. When TSX is high and TENB is high, the value of TDAT is the address of the transmit FIFO to be selected. Subsequent data transfers on the TDAT bus will fill the FIFO specified by this in-band address. For single channel PHY devices the TSX signal is optional. TSX is considered valid only when TENB is not asserted.

<$chapnum>.4.1.3.8 TSOP (transmit start of packet)

TSOP is used to delineate the packet boundaries on the TDAT bus. When TSOP is high the start of the packet is present on the TDAT bus. TSOP is required to be present at the beginning of every packet and is considered valid only when TENB is asserted. The generation of this signal is as result of reception of a PHY_DATA.request.

<$chapnum>.4.1.3.9 TEOP (transmit end of packet)

TEOP is used to delineate the packet boundaries on the TDAT bus. When TEOP is high, the end of the packet is present on the TDAT bus. TEOP is required to be present at the end of every packets and is considered valid only when TENB is asserted. The generation of this signal is as result of reception of a PHY_DATA.request.

<$chapnum>.4.1.3.10 TADR (transmit PHY address) and PTPA (polled-PHY transmit packet available)

The use of TADR is optional. It is for multi-port PHY applciation. A RS communicates with one port of a PHY device. TADR is used in the packet transfer mode. There is no direct mapping from the PSAP primitives.

TADR bus is used with the PTPA signal to poll the transmit FIFO's packet available status. When TADR is sampled on the rising edge of TFCLK by the PHY the polled packet available indication PTPA is updated with the status of the channel specified by the TADR address on the following rising edge of TFCLK.

The result of the polled status PTPA is used to generate PHY_READY.indicate.

<$chapnum>.4.1.3.11 DTPA (direct transmit packet available)

DTPA is used in the byte transfer mode.

The result of DPTA is used to generate PHY_READY.indicate.

DTPA bus provides direct status indication for the corresponding ports in the PHY device. DTPA transitions high when a predefined minimum number of bytes is available in its transmit FIFO. Once high, the DTPA signal indicates that its corresponding transmit FIFO is not full. When DTPA transitions low it indicates that its transmit FIFO is full or near full. DTPA is updated on the rising edge of TFCLK.

<$chapnum>.4.1.3.12 STPA (selected transmit packet available)

The use of STPA is optional. It is for multi-port PHY applciation. STPA is used in the byte transfer mode.

The result of DPTA is used to generate PHY_READY.indicate.

STPA provides status indication of the selected port of a PHY device in order to avoid PHY FIFO overflow while polling is performed. The port which STPA reports is updated on the following rising edge of TFCLK after the PHY address on the TDAT is sampled by the PHY.

<$chapnum>.4.1.3.13 RFCLK (receive FIFO write clock)

RFCLK is a continuous clock used to synchronize data transfer transactions between the MAC and the PHY. RFCLK may cycle at a rate up to 104 MHz.

<$chapnum>.4.1.3.14 RVAL (receive data valid)

RVAL indicates the validity of the receive data. RVAL is low between transfers and when RSX is asserted. It is also low when the PHY pauses a transfer due to an empty receive FIFO. When a transfer is paused by holding RENB high RVAL will hold its value unchanged although no new data will be present on RDAT until the transfer resumes. When RVAL is high the RDAT, RMOD, RSOP, REOP and RERR signals are

valid. When RVAL is low, the RDAT, RMOD, RSOP, REOP and RERR signals are invalid and shall be disregarded. The RSX signal is valid when RVAL is low.

<$chapnum>.4.1.3.15 RENB (receive read enable)

The RENB signal is used to control the flow of data from the receive FIFOs. During data transfer, RVAL shall be monitored as it will indicate if the RDAT, RPRTY, RMOD, RSOP, REOP, RERR and RSX are valid. The system may deassert RENB at anytime if it is unable to accept data from the PHY device. When RENB is sampled low by the PHY device a read is performed from the receive FIFO and the RDAT, RPRTY, RMOD, RSOP, REOP, RERR, RSX and RVAL signals are updated on the following rising edge of RFCLK. When RENB is sampled high by the PHY device a read is not performed and the RDAT, RPRTY, RMOD, RSOP, REOP, RERR, RSX and RVAL remain unchanged on the following rising edge of RFCLK.

<$chapnum>.4.1.3.16 RDAT[31:0] (receive packet data bus)

RDAT is an 8-bit or 32-bit bus which carries the packet bytes that are read from the receive FIFO and the in-band port address of the selected receive FIFO. RDAT is considered valid only when RVAL is asserted. Mapping of RDAT signals is shown in Figure<$chapnum>.9

Valid data received on the RDAT bus from RSOP to REOP are mapped into PHY_DATA.indicate.

<$chapnum>.4.1.3.17 RPRTY (receive parity)

RPRTY signal indicates the odd parity calculated over the RDAT bus. There is no direct mapping to the PSAP primitives.

TPRTY indicates the parity calculated over the TDAT bus. TPRTY is considered valid only when TENB or TSX is asserted. The parity calculation is such that odd parity is indicated on this signal.

<$chapnum>.4.1.3.18 RMOD[1:0] (receive word modulo)

RMOD is required only when RDAT is a 32-bit bus. RMOD indicates the number of valid bytes of data in RDAT. The RMOD bus should always be zero except during the last word transfer of a packet on RDAT. When REOP is asserted the number of valid packet data bytes on RDAT is decoded from RMOD as:

RMOD[1:0] = 00 RDAT[31:0] is valid
RMOD[1:0] = 01 RDAT[31:8] is valid
RMOD[1:0] = 10 RDAT[31:16]is valid
RMOD[1:0] = 11 RDAT[31:24] is valid

RMOD is considered valid only when RVAL is asserted

<$chapnum>.4.1.3.19 RSOP (receive start of packet)

RSOP is used to delineate the packet boundaries on the RDAT bus. When RSOP is high, the start of the packet is present on the RDAT bus. RSOP is required to be present at the beginning of every packet and is considered valid only when RVAL is asserted.

Valid data received on the RDAT bus from RSOP to REOP are mapped into PHY_DATA.indicate.

<$chapnum>.4.1.3.20 REOP (receive end of packet)

REOP is used to delineate the packet boundaries on the RDAT bus. When REOP is high, the end of the packet is present on the RDAT bus. REOP is required to be present at the end of every packet and is considered valid only when RVAL is asserted.

Upon receiving an REOP, the data received from RSOP constitute a received frame. For SRS sublayer, the received frame formulates the INPUT_FRAME attribute of the PHY_DATA.indicate primitive. For a GRS sublayer, the first 2 bytes of the received frame is mapped into the LENGTH attribute and the GFP paylaod less the header is mapped into the INPUT_FRAME attribute of the PHY_DATA.indicate primitive.

<$chapnum>.4.1.3.21 RERR (receive error indicator)

The use of RERR is optional, as it does not interfere with the transfer of the packet.

RERR is used to indicate that the current packet is in error. RERR shall only be asserted when REOP is asserted. Conditions that can cause RERR to be set may be, but are not limit to, FIFO overflow, abort sequence detection and L1 FCS error.

<$chapnum>.4.1.3.22 RSX (receive start of transfer)

RSX indicates when the in-band port address is present on the RDAT bus. When RSX is high, the value of RDAT[31:0] is the address of the receive FIFO to be selected by the PHY. Subsequent data transfers on the RDAT bus will be from the FIFO specified by this in-band address. For single channel PHY devices the RSX signal is optional. For multi-port PHY devices, RSX is asserted at the beginning of each transfer. When RSX is high RVAL shall be low.

<$chapnum>.4.1.4 Mapping of SPI-3 signals to service interface primitives

The reconciliation sublayer shall map the signals provided at the SPI-3 interface to the MAC physical layer service interface primitives defined in Clause 7. Mappings for the following primitives are defined:

— PHY_DATA.request
— PHY_DATA.indicate
— PHY_READY.indicate.

<$chapnum>.4.1.4.1 Mapping of PHY_DATA.request

PHY_DATA.request (OUTPUT_FRAME, LENGTH)

The OUTPUT_FRAME represents a complete RPR frame, and is conveyed to the PHY by the signals TDAT<31:0> and TMOD<1:0>.

On each TFCLK rising edge, 32 bits of data are transferred from the reconciliation sublayer to the PHY. Data is mapped to a TDAT signal in sequence (TDAT<0:7>, …, TDAT<24:31>, TDAT<0:7>), using the big endian coding as described in D.2.2.6, and with the parity information in the TPRTY signal corresponding to the value of the previous 32 bit data word. The TMOD<1:0> is always fixed to the “00” value except when the end-of-packet is transmitted.

When transmitting RPR MAC frames the TENB signal is low, so the TSX signal is ignored. The usage of the TSX and TENB signals for in-band addressing with multi-port PHY is an implementation choice and out of scope of this standard. If used, this feature should be compliant with the requirements in the OIF specification.

Following transmission of the complete frame, the reconciliation sublayer generates an end-of-packet on the SPI-3 interface. The RS requests transmission of 32 data bits by the PHY, together with the proper parity information in the TPRTY signal, containing the values of the previous PHY_DATA.request transactions not yet transmitted. When transmitting this TDAT<31:0> signal, the TEOP is also high and the TMOD<1:0> represents the number of valid data bytes on the TDAT<31:0> as defined in the OIF specification

The SRS and GRS do not generate preamble or interpacket gap since they are not needed for SONET/SDH PHYs.

<$chapnum>.4.1.4.1.1 When generated

This primitive is invoked by the MAC_datapath entity when it is performed its tasks after it had received a received a request from its client.

<$chapnum>.4.1.4.1.2 Effect of receipt

The receipt of this primitive shall cause the RS to generate the physical frame format required operating in GRS or SRS mode as described in <$chapnum>.2.1 and <$chapnum>.2.2 . The RS will start transferring the physical frame to the PHY in according to SPI-3 protocol: generating TSOP, TDAT, TMOD, and TEOP signals.

<$chapnum>.4.1.4.2 Mapping of PHY_DATA.indicate

PHY_DATA.indicate (INPUT_FRAME, STAUS, LENGTH)

The INPUT_FRAME is derived from the signals RFCLK, RVAL, RENB, RDAT<31:0>, RPRTY, RMOD<1:0>, RSOP, REOP, RERR, RSX.

When receiving frames, the RENB signal is low and the RVAL is high, so the RSX signal is ignored. The usage of the RSX and RENB signals for in-band addressing with multi-port PHY is an implementation choice and out of scope of this standard. If used, this feature should be compliant with the requirements of the OIF SPI-3 specification.

The INPUT_FRAME values are derived from the signals RMOD<1:0> and RDAT<31:0> received from the PHY on each rising edge of the RFCLK. Each primitive generated to the MAC sublayer entity corresponds to a PHY_DATA.request issued by the MAC at the other end of the link connecting two RPR stations.

For each RDAT<31:0> during frame reception, the RS receives four bytes of a frame until the end of frame (when the REOP is high), where one, tow, three or four bytes will be received from the RDAT<31:0> according to the value coded in the RMOD<1:0> as defined in the OIF SPI-3 specification.

During frame reception each RDAT signal shall be mapped in sequence into a received frame (RDAT<0:7>, …, RDAT<24:31>, RDAT<0:7>).

The STATUS parameter takes the value of OK or ERROR and is used to identify frames that are receive with error indications from the PHY, for example, due to assertion of a error signal on the PHY electrical interface. A value of OK indicates no error indication. A value of ERROR indicates that the PHY signaled an error during frame transmission to the reconciliation sublayer.

<$chapnum>.4.1.4.3 When generated

This primitive is invoked by the RS after it had received a complete frame from the physical interface. The INPUT_FRAME is derived from the signals RDAT[31:0], RMOD[1:0], as signalled by the status RSOP, REOP, RSX, in accordance to SPI-3 specification.

The STATUS generates OK unless an active error signal is received on RERR.

<$chapnum>.4.1.4.4 Effect of receipt

The effect of receiving this primitive by the MAC sublayer is specified in clause 6.

<$chapnum>.4.1.4.5 Mapping of PHY_READY.indicate

PHY_READY.indicate (READY_STATUS)

The READY_STATUS attribute takes on the3 value of READY or NOT_READY.

<$chapnum>.4.1.4.5.1 When generated

Depending on single or multiple PHY mode, STPA, PTPA, or DPTA[], are indications from the PHY to the RS that the PHY is ready to receive data on the TDAT bus. Valid indication from the PHY is mapped to the PHY_READY.indicate primitive and send to RS client.

<$chapnum>.4.1.4.6 Effect of receipt

The effect of receiving this primitive by the MAC sublayer is specified in clause 6.

<$chapnum>.4.2 SRS and GRS 32-bit SPI datastream

There are no electrical difference and signalling protocol between the GRS and the SRS sublayer. The only difference is the physical frame format as specified in <$chapnum>.2.1 and <$chapnum>.2.2 .

<$chapnum>.4.3 Functional specifications

The SRS and GRS using 32-bit SPI-3 interfaces shall meet the functional requirements of paragraphs 8., 9, 10 and 11 of the OIF SPI-3 implementation agreement.

<$chapnum>.4.4 Electrical specifications

The SRS and GRS using 32-bit SPI-3 interfaces shall meet the transmit electrical timing requirements of paragraph 10.3, and the receive electrical timing requirements of paragraph 11.3, of the OIF SPI-3 implementation agreement.

<$chapnum>.5 SRS and GRS using the SPI-4 Phase 1 interface

The SONET/SDH Reconciliation Sublayer and GFP Reconciliation Sublayer (GRS) may be implemented with an OIF SPI-4 Phase 1 interface.

<$chapnum>.5.1 General requirements

This clause provides a brief description of the signal for the SPI4-01.0 interface and its interaction with the MAC physical layer service interface primitive. Any discrepancy in the signal description is not intentional. The OIF-SPI4-01.0 takes precedence over the subclause below.

<$chapnum>.5.1.1 Summary of major concepts

a) The SRS and GRS map the signals provided at the SPI-4 Phase 1 interface to the logical physical layer service interface primitives provided at the MAC;
b) Each direction of data transfer is independent, and serviced by 64-bit data, delimiter, control, error, and clock signals;
c) Data, control, and delimiter are source-synchronous HSTL signals;
d) The SRS and GRS using the SPI-4 Phase 1 interface support full-duplex operation only.

<$chapnum>.5.1.2 Rate of operation

The SRS and GRS using the SPI-4 Phase 1 interface are capable of supporting data rates of 622 Mbps to 10 Gbps.

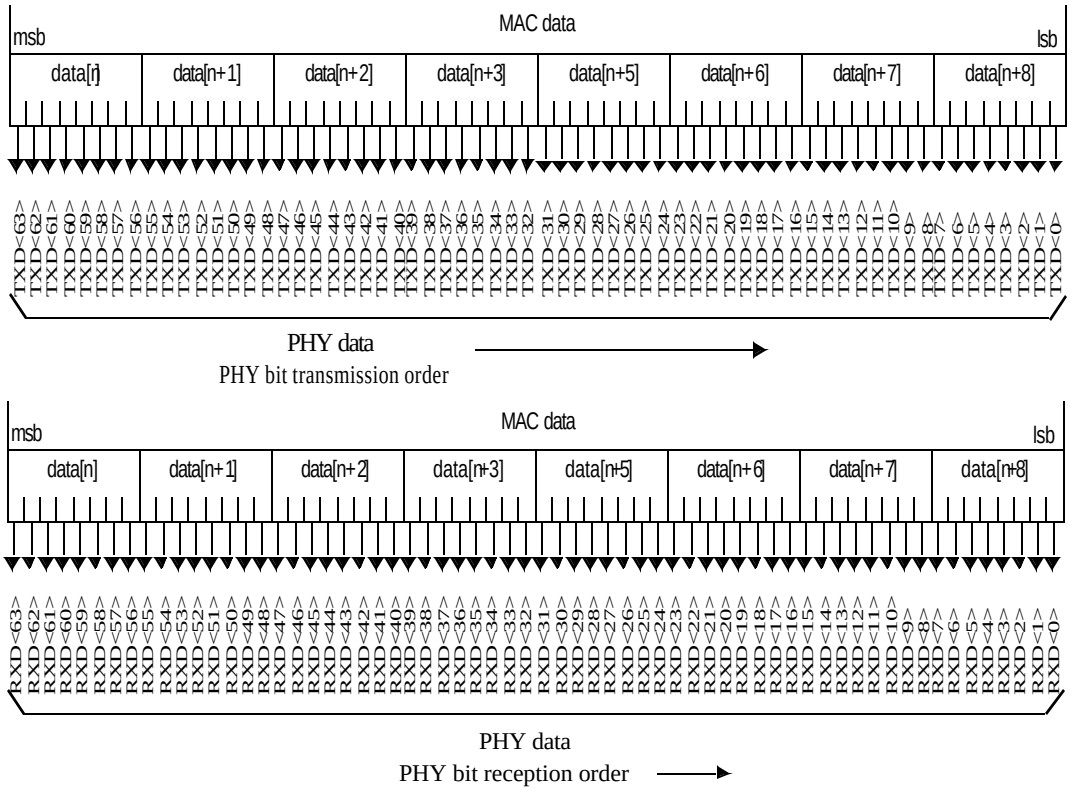


Figure <$chapnum>.10—64-bit SPI-4 phase 1 transmit and receive signal mapping

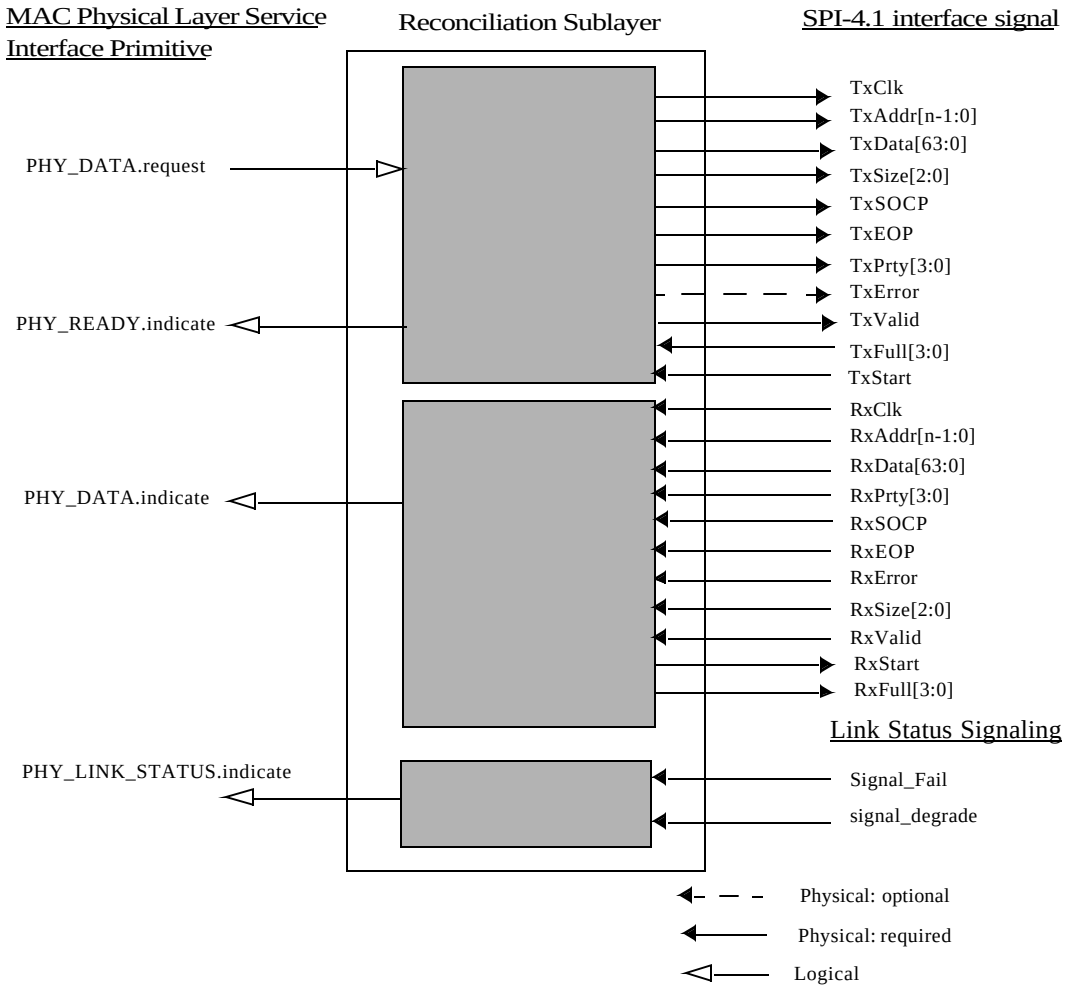
<$chapnum>.5.1.3 SPI-4 Phase 1 structure

SPI-4 phase 1 supports a single 64 bit interface and a 4x16 bit interface. The 4x16 mode operates as 4 independent 16 bit bus interface each with its full set control signals and clock. A RPR MAC is a single link layer device and it requires the PHY interface to operate a single PHY device.

The 4x16 mode is used where applicable. This clause describes the operation for single 10Gbps MAC operation.

Figure<$chapnum>.11 shows a schematic view of the SRS and GRS inputs and outputs using the 64-bit SPI-4 phase 1interface.

Figure <$chapnum>.11—SPI-4.1 single 16 bit interface Interface Signals



<$chapnum>.5.1.3.1 TxClk (transmit clock)

TxClk is a continuous clock used to synchronize data transfer transactions between the RS and the PHY layer device. TxClk may cycle at a rate up to 200 MHz. SPI-4.1 uses source synchronous clocking.

<$chapnum>.5.1.3.2 TxError (transmit error Indicator)

The use of TxError is optional. There is no direct mapping from the PSAP primitives.

When a packet is being transmitted and an internal error occurs then TxError is used to indicate that there is an error in the current packet. TxError should only be asserted when TxEOP is asserted; it is considered valid only when TxValid is asserted. The TxError does not interfere with the transfer of data in the PHY.

<$chapnum>.5.1.3.3 TxValid (transmit data valid)

This is a physical interface protocol signal for flow control at the interface level. The generation of this signal is as result the reception of a PHY_DATA.request and while valid data symbols are being transmitted across the physical interface.

When TxValid is deasserted the TxData, TxSize, TxSOCP, TxEOP and TxError signals are invalid and should be ignored by the PHY.

<$chapnum>.5.1.3.4 TxData[63:0] (transmit packet data bus)

TxData is an 64-bit bus which carries the packet bytes that are written to the selected transmit FIFO port whose address is select by the TxAddr siganls. The TxData bus is considered valid only when TxValid is asserted. Data is transmitted in big endian order on TxData. Mapping of TxData signals is shown in Figure<$chapnum>.10.

The generation of this signal is as result of reception of a PHY_DATA.request. The OUTPUT_FRAME and LENGTH primitives are formatted into the appropriate physical frame format and each

<$chapnum>.5.1.3.5 TxSize[2:0] (transmit word modulo)

TxSize is required for the 64-bits TxData bus. TxSize indicates the number of valid bytes of data in TxData. The TxSize bus should always be zero except during the last word transfer of a packet on TxData. When TxEOP is asserted the number of valid packet data bytes on TxData is decoded from TxSize as:

TxSize[2:0] = 001	TxData[63:56] is valid
TxSize[2:0] = 010	TxData[63:48] is valid
TxSize[2:0] = 011	TxData[63:40] is valid
TxSize[2:0] = 100	TxData[63:32] is valid
TxSize[2:0] = 101	TxData[63:24] is valid
TxSize[2:0] = 110	TxData[6316] is valid
TxSize[2:0] = 111	TxData[63:8] is valid

TxSize is considered valid only when TxValid is asserted.

<$chapnum>.5.1.3.6 TxPrty[3:0] (transmit bus parity)

TxPrty indicates the parity calculated over the TxData bus. TxPrty is valid even when TxValid is deasserted. The parity calculation is such that odd parity is indicated on this signal.

TxPrty[0] = odd parity over TxData[15:0]
TxPrty[1] = odd parity over TxData[31:16]
TxPrty[2] = odd parity over TxData[47:32]
TxPrty[3] = odd parity over TxData[63:48]

The parity is used to detect interface error. It does not interfere with the data transfer.

<$chapnum>.5.1.3.7 TxSOP (transmit start of packet)

TxSOP is used to delineate the packet boundaries on the TxData bus. When TxSOP is high the start of the packet is present on the TxData bus. TxSOP is required to be present at the beginning of every packet and is considered valid only when TxValid is asserted.

This is a physical interface protocol signal for flow control at the interface level. The generation of this signal is as result the reception of a PHY_DATA.request and for the first valid data symbols are being transmitted across the physical interface.

<$chapnum>.5.1.3.8 TxEOP (transmit end of packet)

TxEOP is used to delineate the packet boundaries on the TxData bus. When TxEOP is high, the end of the packet is present on the TxData bus. TxEOP is required to be present at the end of every packets and is considered valid only when TxValid is asserted.

This is a physical interface protocol signal for flow control at the interface level. The generation of this signal is as result the reception of a PHY_DATA.request and for the last valid data symbols are being transmitted across the physical interface.

<$chapnum>.5.1.3.9 TxStart (transmit flow control frame start)

TxStart indicates when the start of the TxFull frame boundary. When TxStart is high The TxFull indicate PHY transmit FIFO full status for the first 4 ports. The next clock cycle indicates the FIFO status for the next 4 ports. It is time multiplexed.

This signal is synchronous to the RxClk

<$chapnum>.5.1.3.10 TxFull[3:0] (transmit flow control full)

TxFull is used by the PHY to signal the RS that there is no room to receive data from the RS for a port of a multiport PHY device. This signal is synchronous to the RxClk.

For single PHY support, TxFull [0] is used and the round robin scheme is disable.

The result of the TxFull status is used to generate PHY_READY.indicate.

<$chapnum>.5.1.3.11 RxClk (receive FIFO write clock)

RxClk is a continuous clock used to synchronize data transfer transactions between the PHY and the RS. RxClk may cycle at a rate up to 200 MHz.

<$chapnum>.5.1.3.12 RxValid (receive data valid)

When RxValid is deasserted the following signals are undefined: RxAddr, RxData, RxSOCP, RxEOP, RxError, and RxSize.

<$chapnum>.5.1.3.13 RxData[63:0] (receive packet data bus)

RxData is a 64-bit bus which carries the packet bytes that are read from the PHY. RxData is considered valid only when RxValid is asserted. Mapping of RxData signals is shown in Figure <$chapnum>.11

<$chapnum>.5.1.3.14 RxPrty (receive parity)

RxPrty indicates the parity calculated over the RxData bus. RxPrty is valid even when RxValid is deasserted. The parity calculation is such that odd parity is indicated on this signal.

RxPrty[0] = odd parity over RxData[15:0]
RxPrty[1] = odd parity over RxData[31:16]
RxPrty[2] = odd parity over RxData[47:32]
RxPrty[3] = odd parity over RxData[63:48]

<$chapnum>.5.1.3.15 RxSize[1:0] (receive word modulo)

RxSize is required for 64 bit RxData bus. RxSize indicates the number of valid bytes of data in RxData. The RxSize bus should always be zero except during the last word transfer of a packet on Rxdata. When RxEOP is asserted the number of valid packet data bytes on RxData is decoded from RxSize as:

RxSize[2:0] = 001	RxData[63:56] is valid
RxSize[2:0] = 010	RxData[63:48] is valid
RxSize[2:0] = 011	RxData[63:40] is valid
RxSize[2:0] = 100	RxData[63:32] is valid
RxSize[2:0] = 101	RxData[63:24] is valid
RxSize[2:0] = 110	RxData[6316] is valid
RxSize[2:0] = 111	RxData[63:8] is valid

RxSize is considered valid only when RxValid is asserted

<$chapnum>.5.1.3.16 RxSOCP (receive start of packet)

RxSOCP is used to delineate the packet boundaries on the RxData bus. When RxSOCP is high, the start of the packet is present on the RxData bus. RxSOCP is required to be present at the beginning of every packet and is considered valid only when RxValid is asserted.

<$chapnum>.5.1.3.17 RxEOP (receive end of packet)

RxEOP is used to delineate the packet boundaries on the RxData bus. When RxEOP is high, the end of the packet is present on the RxData bus. RxEOP is required to be present at the end of every packet and is considered valid only when RxValid is asserted.

<$chapnum>.5.1.3.18 RxError (receive error indicator)

RxError is used to indicate that the current packet is in error. RxError is asserted when RxEOP is asserted. Conditions that can cause RxError to be set may be, but are not limit to, FIFO overflow, or abort sequence.

<$chapnum>.5.1.4 Mapping of SPI-4 signals to service interface primitives

The reconciliation sublayer shall map the signals provided at the SPI-4 interface to the MAC physical layer service interface primitives defined in Clause 7. Mappings for the following primitives are defined:

— PHY_DATA.request
— PHY_DATA.indicate
— PHY_READY.indicate.

<$chapnum>.5.1.4.1 Mapping of PHY_DATA.request

PHY_DATA.request (OUTPUT_FRAME, LENGTH)

The OUTPUT_FRAME is mapped into series of TxData octets. The PHY inserts Idle. The start of the frame is signaled by the TxSOP and the end alignes with the signaling of TxEOP and the correct value for TxSize.

The LENGTH is used by the GRS. It is mapped to the PLI field of the core header.

<$chapnum>.5.1.4.2 Mapping of PHY_DATA.indicate

PHY_DATA.indicate (INPUT_FRAME, STATUS, LENGTH)

For the GRS, upon receiving a complete frame from the PHY. All valid octets received from RxSOCP and RxEOP formulates the INPUT_FRAME primitive less the GFP core header and payload header. The core header field PLI is mapped into the LENGTH primitive.

For SRS, upon receiving a complete frame from the PHY. All valid octets received from RxSOP and RxEOP formulates the INPUT_FRAME primitive.

The STATUS generates OK unless an active error signal is received on RxError.

<$chapnum>.5.1.4.3 Mapping of PHY_READY.indicate

PHY_READY.indicate (READY_STATUS)

The READY_STATUS attribute takes on the value of READY or NOT_READY.

<$chapnum>.5.1.4.3.1 When generated

Depending on single or multiple PHY mode, TxStart and TxFull are indications from the PHY to the RS that the PHY is ready to receive data on the TxData bus. Valid indication from the PHY is mapped to the PHY_READY.indicate primitive and sent to RS client.

<$chapnum>.5.1.4.4 Effect of receipt

The effect of receiving this primitive by the MAC sublayer is specified in clause 6.

<$chapnum>.5.2 SRS and GRS 64-bit SPI datastream

There are no electrical difference and signaling protocol between the GRS and the SRS sublayer. The only difference is the physical frame format as specified in <$chapnum>.2.1 and <$chapnum>.2.2 .

<$chapnum>.5.3 Functional specifications

The SRS and GRS using SPI-4 Phase 1 interfaces shall meet the functional requirements of section 6 of the OIF SPI-4 Phase 1 implementation agreement.

<$chapnum>.5.4 Electrical specifications

The SRS and GRS using the SPI-4 Phase 1 interface shall meet the electrical timing requirements of sections 10, 11 and 12 of the OIF SPI-4 implementation agreement.

<$chapnum>.6 SRS and GRS using SPI-4 Level 2 interface

The SONET/SDH Reconciliation Sublayer and GFP Reconciliation Sublayer (GRS) may be implemented with an OIF SPI-4 Phase 2 interface.

SPI-4 is an interface for packet and cell transfer between a physical layer (PHY) device and a link layer device, for aggregate bandwidths of OC-192 rate.

On both the transmit and receive interfaces, FIFO status information is sent separately from the corresponding data path. By taking FIFO status information out-of-band, it is possible to decouple the transmit and receive interfaces so that each operates independently of the other. Such an arrangement makes SPI-4 suitable not only for bidirectional but also for unidirectional link layer devices.

In both the transmit and receive interfaces, the packet's address, delineation information and error control coding is sent in-band with the data.

<$chapnum>.6.1 General requirements

This clause provides a brief description of the signal for the SPI4-02.0 interface and its interaction with the MAC physical layer service interface primitive. Any discrepancy in the signal description is not intentional. The OIF-SPI4-02.0 takes precedence over the subclause below.

<$chapnum>.6.1.1 Summary of major concepts

a) The SRS and GRS map the signals provided at the SPI-4 Phase 2 interface to the logical physical layer service interface primitives provided at the MAC;
b) Each direction of data transfer is independent, and serviced by 16-bit data, control, and clock signals;
c) Each direction of data transfer includes separate FIFO status channels consisting of status signals and status clocks;
d) The SRS and GRS using the SPI-4 Phase 2 interface support full-duplex operation only.

Figure<$chapnum>.12 shows a schematic view of the SRS and GRS inputs and outputs using the 64-bit SPI-4 phase 2 interface.

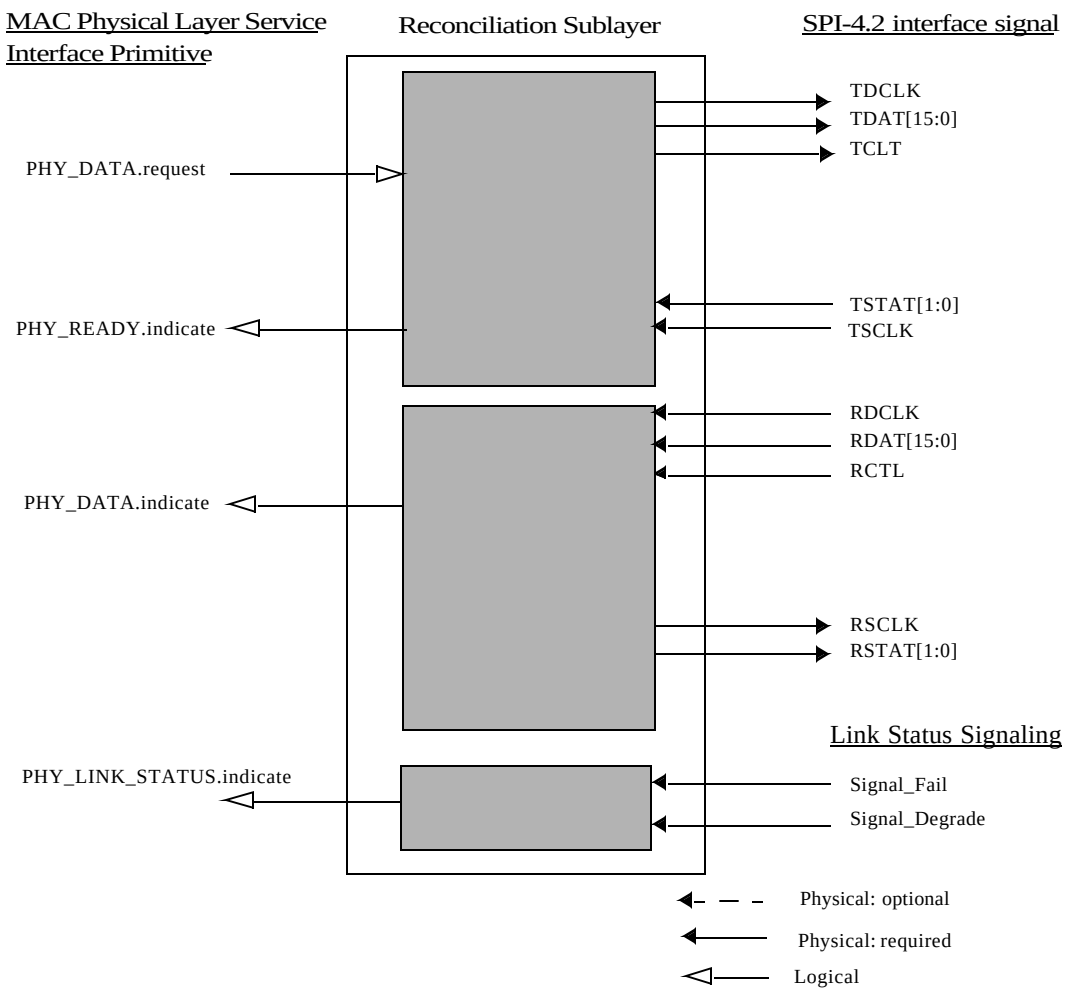


Figure <$chapnum>.12—SPI-4.2 Interface Signals

<$chapnum>.6.1.2 Rate of operation

The SRS and GRS using the SPI-4 Phase 2 interface are capable of supporting data rates of 622 Mbps to 10 Gbps.

.

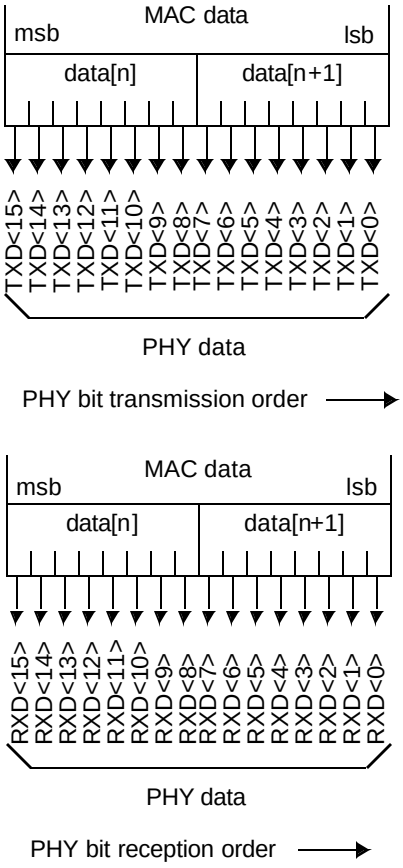


Figure <$chapnum>.13—SPI-4 Phase 2 transmit and receive signal mapping

<$chapnum>.6.1.3 SPI-4 Phase 2 structure

The SPI-4 Phase 2 interface signals are described in the following subclauses. Data and control lines are driven from the rising and falling edges of the clock (TDCLK).

<$chapnum>.6.1.3.1 TDCLK (transmit data clock)

TDCLK is a clock associated with TDAT and TCTL. TDCLK provides the datapath source-synchronous double-edge clocking with a minimum frequency of 311 MHz. Data and control lines are driven off the rising and falling edges of the clock. TDCLK is sourced by the MAC to the PHY.

<$chapnum>.6.1.3.2 TDAT[15:0] (transmit data)

TDAT is a 16-bit bus used to carry payload data and in-band control words from the Link Layer to the PHY device. A control word is present on TDAT when TCTL is high. The minimum data rate for TDAT is 622 Mb/s. Mapping of TDAT is shown in Figure <$chapnum>.13.

<$chapnum>.6.1.3.3 TCTL (transmit control)

TCTL is high when a control word is present on TDAT, otherwise it is low. TCTL is sourced by the MAC to the PHY.

<$chapnum>.6.1.3.4 TSCLK (transmit status clock)

TSCLK is a clock associated with TSTAT providing source-synchronous clocking. For LVTTL I/O a maximum clock rate restraint is ¼ that of the data path clock rate. LVDS I/O allows a maximum of that equal to the data path clock (double-edge clocking).

<$chapnum>.6.1.3.5 TSTAT[1:0] (transmit FIFO status)

TSTAT is a 2-bit bus used to carry round-robin FIFO status information, along with associated error detection and framing. The maximum data rate for TSTAT is dependent on the I/O type, either LVDS or LVTTL, and is limited to its respective TSCLK restraints. TSTAT is sourced by the PHY to the MAC. The FIFO status formats are:

TSTAT[1:0] = 11	Reserved for framing or to indicate a disabled status link.
TSTAT[1:0] = 10	SATISFIED
TSTAT[1:0] = 01	HUNGRY
TSTAT[1:0] = 00	STARVING

<$chapnum>.6.1.3.6 RDCLK (receive data clock)

RDCLK is a clock associated with RDAT and RCTL. RDCLK provides the datapath source-synchronous double-edge clocking with a minimum frequency of 311 MHz. Data and control lines are driven off the rising and falling edges of the clock. RDCLK is sourced by the PHY to the MAC.

<$chapnum>.6.1.3.7 RDAT[15:0] (receive data)

RDAT is a 16-bit bus which carries payload data and in-band control from the PHY to the Link Layer device. A control word is present on RDAT when RCTL is high. The minimum data rate for RDAT is 622 Mb/s. Mapping of RDAT is shown in Figure<$chapnum>.13

<$chapnum>.6.1.3.8 RCTL (receive control)

RCTL is high when a control word is present on RDAT, otherwise it is low. RCTL is sourced by the PHY to the MAC.

<$chapnum>.6.1.3.9 RSCLK (receive status clock)

RSCLK is a clock associated with RSTAT providing source-synchronous clocking. RSCLK is sourced by the Mac to the PHY. LVDS I/O allows a maximum of that equal to the data path clock (double-edge clocking).

<$chapnum>.6.1.3.10 RSTAT[1:0] (receive FIFO status)

RSTAT is a 2-bit bus used to carry round-robin FIFO status information, along with associated error detection and framing. The maximum data rate for RSTAT is dependent on the I/O type, either LVDS or LVTTL, and is limited to its respective RSCLK restraints. RSTAT is sourced by the Mac to the PHY. The FIFO status formats are:

RSTAT[1:0] = 11	Reserved for framing or to indicate a disabled status link.
RSTAT[1:0] = 10	SATISFIED
RSTAT[1:0] = 01	HUNGRY
RSTAT[1:0] = 00	STARVING

<$chapnum>.6.1.4 Mapping of SPI-4 signals to service interface primitives

The reconciliation sublayer shall map the signals provided at the SPI-4 interface to the MAC physical layer service interface primitives defined in Clause 7. Mappings for the following primitives are defined:

— PHY_DATA.request
— PHY_DATA.indicate
— PHY_READY.indicate.

<$chapnum>.6.1.4.1 Mapping of PHY_DATA.request

PHY_DATA.request (OUTPUT_FRAME, LENGTH)

There are two attributes: INPUT_FRAME and LENGTH

The INPUT_FRAME is mapped from the received payload. The LENGTH applies to GRS only. It is mapped from the PLI field in the GFP core header.

<$chapnum>.6.1.4.2 When generated

This primitive is invoked by the RS after it had received a complete frame from the physical interface. The INPUT_FRAME is derived from the signals TDAT and TCTL, in accordance to SPI-4.2 specification.

<$chapnum>.6.1.4.3 Effect of receipt

The effect of receiving this primitive by the MAC sublayer is specified in clause 6.

<$chapnum>.6.1.4.4 Mapping of PHY_DATA.indicate

PHY_DATA.indicate (INPUT_FRAME, STATUS, LENGTH)

PHY_DATA.indicate (INPUT_FRAME, STATUS,LENGTH)

There are three attributes: INPUT_FRAME, STATUS, and LENGTH

The INPUT_FRAME is mapped from the received payload. The LENGTH applies to GRS only. It is mapped from the PLI field in the GFP core header.

<$chapnum>.6.1.4.5 When generated

This primitive is invoked by the RS after it had received a complete frame from the physical interface. The INPUT_FRAME is derived from the signals RDAT[31:0], RSCLK, and RSTAT in accordance to SPI-4.2 specification.

<$chapnum>.6.1.4.6 Effect of receipt

The effect of receiving this primitive by the MAC sublayer is specified in clause 6.

<$chapnum>.6.2 SRS and GRS SPI-4 Phase 2 datastream

There are no electrical difference and signaling protocol between the GRS and the SRS sublayer. The only difference is the physical frame format as specified in

<$chapnum>.6.3 Functional specifications

The SRS and GRS using SPI-4 Phase 2 interfaces shall meet the functional requirements of section 6 of the OIF SPI-4 Phase 2 implementation agreement

<$chapnum>.6.4 Electrical specifications

The SRS and GRS using the SPI-4 Phase 2 interface shall meet the electrical timing requirements of sections 6.4 and 6.5 of the OIF SPI-4 implementation agreement.