Contribution to IEEE 802.3 HSSG

Proposal for a 10 Gigabit Ethernet WAN PHY

November 10, 1999

Norival Figueira, Nortel Networks
Paul Bottorff, Nortel Networks
Tom Palkert, AMCC

Contents

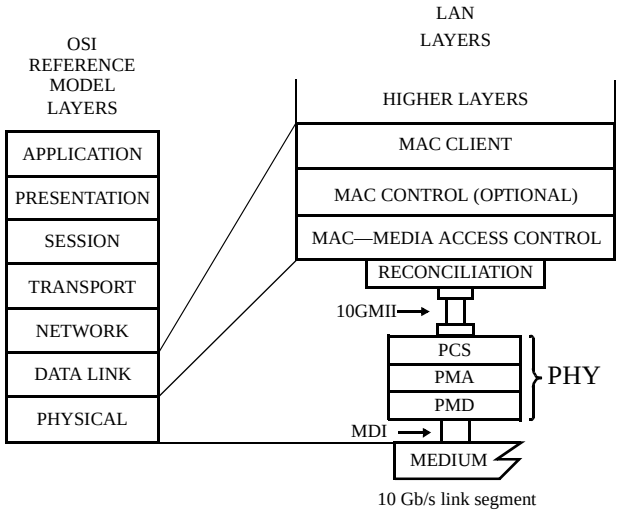
1. Summary

This document is a proposal for the IEEE 10 Gigabit Ethernet WAN Physical Layer (PHY). We refer to the WAN PHY solution as 10GBASE-WX (the “W” means WAN), which operates over a pair of optical fibers using 9.95328 Gb/s photonics. 10GBASE-WX seeks compatibility with the Physical Layer standards define in the following references:

a) Telcordia GR-253-CORE Revision 2, January 1999 (i.e., SONET), and GR-1377-CORE (i.e., SONET OC-192).
b) ITU-T G.707, March 1996, (i.e., SDH), G.783, April 1997, and G.957, July 1995.

This document uses some SONET terminology whenever absolutely required to simplify exposition.

The relationships among 10 Gigabit Ethernet, the existing IEEE 802.3 MAC, and the ISO/IEC Open System Interconnection (OSI) reference model is shown in Figure 1-1. The 10GBASE-WX Physical Layer (PHY) is composed of the Physical Coding Sublayer (PCS), the Physical Medium Attachment (PMA) sublayer, and the Physical Medium Dependent (PMD) sublayer.



MDI = MEDIUM DEPENDENT INTERFACE
10GMII = 10 GIGABIT INDEPENDENT INTERFACE
PCS = PHYSICAL CODING SUBLAYER
PMA = PHYSICAL MEDIUM ATTACHMENT
PHY = PHYSICAL LAYER DEVICE
PMD = PHYSICAL MEDIUM DEPENDENT

Figure 1-1—Architectural positioning of 10 Gigabit Ethernet

2. 10GMII data stream

Data frames transmitted through the 10GMII shall be transferred within the data stream shown in Figure 2-1.

<inter-frame><preamble><sfd><data><efd>

Figure 2-1—10GMII data stream

2.1 Inter-frame <inter-frame>

TBD

2.2 Preamble <preamble> and start of frame delimiter <sfd>

2.2.1 Transmit case

The preamble <preamble> and SFD <sfd> begin a frame transmission. The bit value of the preamble and SFD fields at the 10GMII shall consist of 8 octets with the bit values shown in Figure 2-2. The SFD (Start Frame Delimiter) <sfd> indicates the start of a frame and immediately follows the preamble. The bit value of the SFD at the 10GMII is unchanged from that specified in 4.2.6 (refers to IEEE 802.3 1998 Edition). As shown in Figure 2-2, the leftmost bit of the SFD field is the MSB of the octet and the rightmost bit is the LSB of the octet.

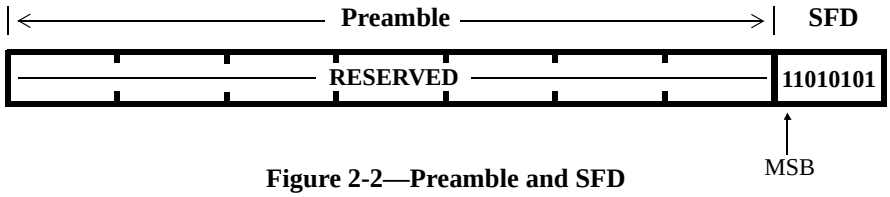


Figure 2-2—Preamble and SFD

2.2.2 Receive case

The operation of the 10 Gb/s PHYs shall not result in shrinkage of the preamble between transmission at the source 10GMII and reception at the destination 10GMII. The reception of the whole preamble is required.

2.3 Data <data>

The data <data> in a well-formed frame shall consist of a set of data octets.

2.4 End-of-Frame delimiter <efd>

TBD

2.5 Definition of Start of Packet and End of Packet Delimiters

TBD

3. Overview of the Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) sublayer, type 10GBASE-WX

3.1 Scope

This clause specifies the Physical Coding Sublayer (PCS) and the Physical Medium Attachment (PMA) sub-layer that are common to the family of 10 Gb/s WAN-compatible Physical Layer implementations, collectively called in this document the 10GBASE-WX (W as in WAN) family.

The 10GBASE-WX PHY is based on the same SONET framing and scrambler synchronization as defined in Telcordia GR-253-CORE Revision 2, January 1999, and Telcordia GR-1377-CORE Issue 5, December 1998, using 9.95328 Gb/s photonics, and seeks compatibility with ITU-T G.707, March 1996, and ITU-T G.783, April 1997.

The PCS and PMA sublayers map the interface characteristics of the PMD sublayer (including MDI) to the services expected by the Reconciliation sublayer.

3.2 Capabilities

The following are the capabilities of the 10GBASE-WX PHY:

a) Supports full duplex operation only;
b) Supports a proper subset of the MAC functionality (the multiple access – i.e., CSMA/CD – algorithms are unnecessary);
c) Provides an effective data rate of 9.58464 Gb/s using a 10GMII clocked at 10 Gb/s and flow controlled to 9.58464 Gb/s;
d) Allows a nominal network extent of up to 300m using multimode fiber and up to 40km using single mode fiber;
e) Preserves full duplex behavior of underlying PMD channels;
f) Supports a BER objective of 10^{-12};
g) Interworks with long haul OC-192 SONET regenerators;
h) Utilizes the OC-192 line frequency, frame format, and the minimum Line and Section Overheads to allow compatibility with existing transmission networks;
i) Utilizes the minimum Line and Path Overheads to allow 10GBASE-WX to operate as a tributary on future higher-capacity transport systems;
j) It does not require an isochronous clock;

3.3 Summary of 10GBASE-WX sublayers

The following provides an overview of the 10GBASE-WX PHY sublayers.[1]

3.3.1 Physical Coding Sublayer (PCS)

The 10GBASE-WX PCS provides the following services:

a) Scrambling (descrambling) of 10GMII data octets (excluding the first eight octets, i.e., preamble plus SFD) using the $x^{43} + 1$ self-synchronous scrambler to protect against potential security threats (see 4.4);

[1]The 10 Gb/s Ethernet WAN PHY consists of that portion of the Physical Layer between the MDI and 10GMII consisting of the PCS, PMA, and PMD sublayers.

b) 10 Gigabit Ethernet frame delineation using the header error control (HEC) check algorithm (see 4.5);
c) Communication with the underlying PMA;
d) Communication with the Reconciliation Sublayer.

3.3.2 Physical Medium Attachment (PMA) sublayer

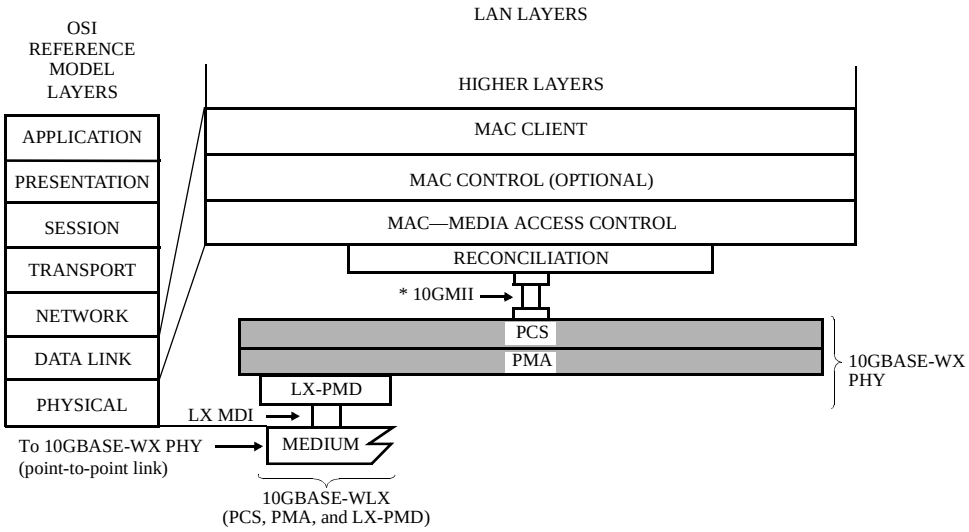
The 10GBASE-WX PMA performs the following functions:

a) Mapping of transmit and receive octets between the PCS and PMA via the PMA Service Interface;
b) Communication with the underlying serial PMD;
c) Mapping of transmit and receive bits between the PMA and PMD via the PMD Service Interface;
d) Data loopback at the PMD Service Interface.
e) PMA framing and octet delineation;
f) Scrambling (descrambling) of the PMA frame using the $x^7 + x^6 + 1$ frame-synchronous scrambler to ensure that the bit stream has plenty of transitions to allow the receiver to recover the sender's clock (see 5.9);
g) Scrambler synchronization;

3.3.3 Physical Medium Dependent (PMD) sublayer

10GBASE-WX physical layer signaling for fiber is adapted from Telcordia GR-253-CORE Revision 2, January 1999, Telcordia GR-1377-CORE Issue 5, December 1998, and seeks compatibility with ITU-T G.957, July 1995, running at STS-192c rate (i.e., 9.95328 Gb/s).

Figure 3-1 depicts the relationship between 10GBASE-WX and its associated PMD sublayers.



MDI = MEDIUM DEPENDENT INTERFACE
10GMII = 10 GIGABIT INDEPENDENT INTERFACE
PCS = PHYSICAL CODING SUBLAYER
PMA = PHYSICAL MEDIUM ATTACHMENT
PHY = PHYSICAL LAYER DEVICE
PMD = PHYSICAL MEDIUM DEPENDENT
LX-PMD = PMD for FIBER—LONG WAVELENGTH, Clause TBD

Note—The PMD sublayers are mutually independent.

* 10GMII is optional.

Figure 3-1—Relationship of 10GBASE-WX and the PMDs

3.4 Inter-sublayer interfaces

There are a number of interfaces employed by 10GBASE-WX. Some (such as the PMA Service Interface) use an abstract service model to define the operation of the interface. An optional physical instantiation of the PCS Interface is called the 10GMII (10 Gigabit Media Independent Interface). Figure 3-2 depicts the relationship and mapping of the services provided by all of the interfaces relevant to 10GBASE-WX.

It is important to note that, while this specification defines interfaces in terms of bits, octets, and 32-bit words, implementors may choose other data path widths for implementation convenience. The only exceptions are a) the 10GMII, which, when implemented at an observable interconnection port, uses a 32-bit-wide data path as specified in TDB, and b) the MDI, which uses a serial, physical interface.

3.5 Functional block diagram

Figure 3-2 provides a functional block diagram of the 10GBASE-WX PHY.

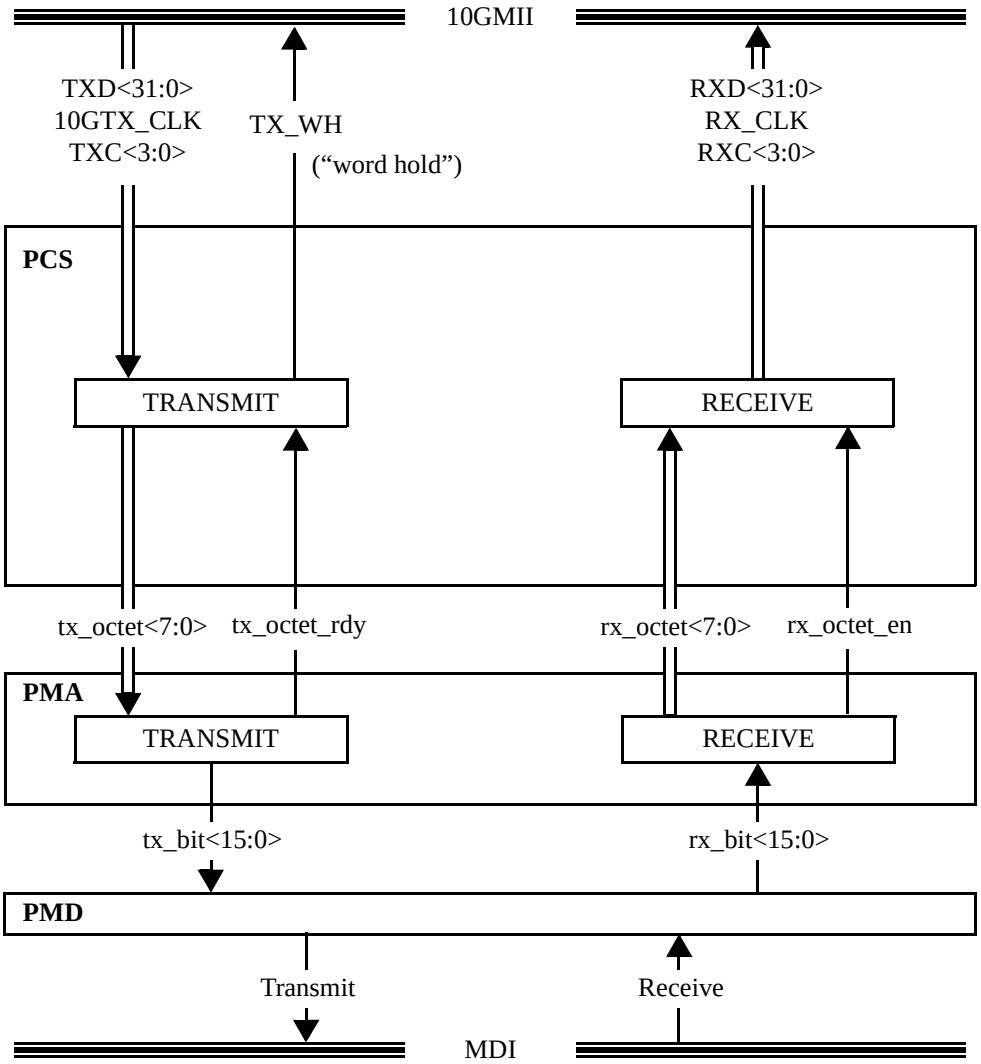


Figure 3-2—Functional block diagram

4. Physical Coding Sublayer (PCS), type 10GBASE-WX

4.1 PCS Interface (10GMII)

The PCS Service Interface allows the 10GBASE-WX PCS to transfer information to and from a PCS client.

4.2 Functions within the PCS

The PCS comprises the PCS Transmit and PCS Receive processes for 10GBASE-WX. The PCS shields the Reconciliation sublayer (and MAC) from the specific nature of the underlying channel. When communicating with the 10GMII, the PCS uses a 32-bit-wide, synchronous data path, with MAC packet delimiting being provided by control symbols, e.g., IDLE, SOP, and EOP. When communicating with the PMA, the PCS uses an octet-wide, synchronous data path.

At the PMA Service Interface, MAC packet delimiting is made possible by embedding a two-octet HEC field (see 4.5.2) after the SFD field of every MAC packet, and by using the HEC check algorithm (see 4.5.3). In addition, to protect against potential security threats (see 4.4), all octets between the HEC field and end of the MAC packet (i.e., excluding the HEC field) are scrambled using a self-synchronous scrambler (see 4.4.1). The PCS provides the functions necessary to map packets between the 10GMII format and the PMA Service Interface format (see Figure 4-1).

The PCS Transmit process generates descrambled/scrambled octets and HEC fields based upon the TXD<31:0> and TXC<3:0> signals on the 10GMII, sending them to the PMA Service Interface via the PMA_UNITDATA.request primitive. This process is not continuous because the PMA Transmit process paces the PCS through the implicit data flow control of the PMA_UNITDATA.request primitive (i.e., using the tx_octet_rdy signal) to allow for the encoding of the PMA framing overhead. The PCS Transmit process generates the 10GMII signal TX_WH to pace the MAC in response to the data flow control of the PMA_UNITDATA.request primitive, to allow for the embedding of the two-octet HEC field in the transmitted octet stream, and also to provide data rate matching between the 10GMII's transmit data rate (i.e., 10 Gb/s) and the PHY's transmit data rate, i.e., 9.95328 Gb/s. During the transmission of a MAC packet, the MAC sublayer responds to the TX_WH signal by transmitting a word of NULL control symbols within a pre-defined length of time (refer to the 10GMII definition in TBD).

The PCS Receive process accepts octets via the PMA_UNITDATA.indicate primitive. The PCS Receive process monitors these octets and generates RXD <31:0> and RXC<3:0> on the 10GMII. This process is not continuous because of the required data rate matching between the 10GMII's receive data rate (i.e., 10 Gb/s) and the PHY's receive data rate (i.e., 9.95328 Gb/s), and because the PMA Receive process paces the PCS through the implicit data flow control of the PMA_UNITDATA.indicate primitive (i.e., using the rx_octet_en signal) to allow for the skipping of the PMA framing overhead. During the reception of a MAC packet, the PCS Receive process generates words of 10GMII NULL control symbols whenever it does not have a complete word for transmission on the 10GMII. The PCS Receive process uses a HEC check algorithm to determine MAC packet delineation. Once a MAC packet is delineated, the PCS Receive process descrambles the octets between the HEC field and the end of the MAC packet.

4.3 Mapping between 10GMII, PCS, and PMA

The octets of MAC packets are transmitted and received sequentially. Figure 4-1 depicts the mapping of the data path of the 10GMII to the PCS, and the data path of the PCS to the PMA interface.

The PCS Transmit process embeds a two-octet HEC field after the SFD field, and scrambles all the octets between the HEC field and the end of the MAC packet (see 4.4). The PCS Receive process delineates MAC packets using the HEC field (see 4.5), and descrambles all the octets between the HEC field and the end of

the MAC packet. Since a MAC packet is delineated only after the HEC field is validated, the PCS Receive process (Figure 3-2) delays the transmission of MAC packets at the 10GMII by at least ten octets. Received HEC fields are deleted from received frames and are not transmitted to the MAC sublayer.

The bit transmission and reception order of octets is also illustrated in Figure 4-1. The bit order of octets is reversed at the PMA Service Interface to maintain the usual FCS error detection capabilities. Otherwise, burst errors at the medium could be seen as extended burst errors at the 10GMII since the PMD transmits and receives octets most significant bits first, while the FCS field is calculated least significant bit first.

4.4 Use of the $x^{43} + 1$ self-synchronous scrambler and descrambler

The PCS Transmit process scrambles all octets between the HEC field (excluded) and end of the MAC packet using the $x^{43} + 1$ self-synchronous scrambler. Conversely, the PCS Receive process descrambles octets between the HEC field (excluded) and end of the MAC packet using the $x^{43} + 1$ self-synchronous descrambler.

The use of the $x^{43} + 1$ self-synchronous scrambler/descrambler protects against potential security threats from malicious users. Without the $x^{43} + 1$ self-synchronous scrambler/descrambler, a malicious user could generate MAC packets with bit patterns that, after being processed by the $x^7 + x^6 + 1$ frame-synchronous scrambler at the PMA sublayer (see 5.9), could create a bit stream without enough transitions at the PMD Service Interface. Lack of enough bit transitions at the PMD Service Interface can degrade clock recovery at the receiver.

The $x^{43} + 1$ self-synchronous scrambler/descrambler improves security because predicting the output of the scrambler requires knowledge of its 43-bit state when the ninth octet of a known MAC packet begins to be scrambled (remember that the first eight octets of a MAC packet are never scrambled). The odds of guessing correctly are $1/2^{43}$.

The PCS functions SCRAMBLE and DESCRAMBLE are used to scramble and descramble octets, respectively, as provided by the following rules.

Scrambling is enabled starting at the first transmitted octet after the HEC field, and is disabled after the last transmitted octet of the MAC packet. The activation of the descrambler depends on the present state of the HEC check algorithm (see 4.5.3):

a) In the HUNT and PRESYNC states, the descrambler is disabled.
b) In the SYNC state, the descrambler is enabled only for the octets between the HEC field and the end of the assumed MAC packet.

The initial state (i.e., at power on or main reset) of the scrambler should be randomly selected to improve security. The initial state of the descrambler is irrelevant, since the descrambler will self-synchronize after 43 descrambled bits. When the scrambler/descrambler is disabled its state is retained.

At start up (i.e., at power on, main reset, or re-synchronization following a loss of signal – for loss of signal see 5.13), the first descrambled 43 bits of the first transmitted MAC packet is used to synchronize the self-synchronous scrambler and, consequently, are corrupted, resulting in the loss of the MAC packet.

Scrambled octets are unobservable and have no meaning outside the PCS.

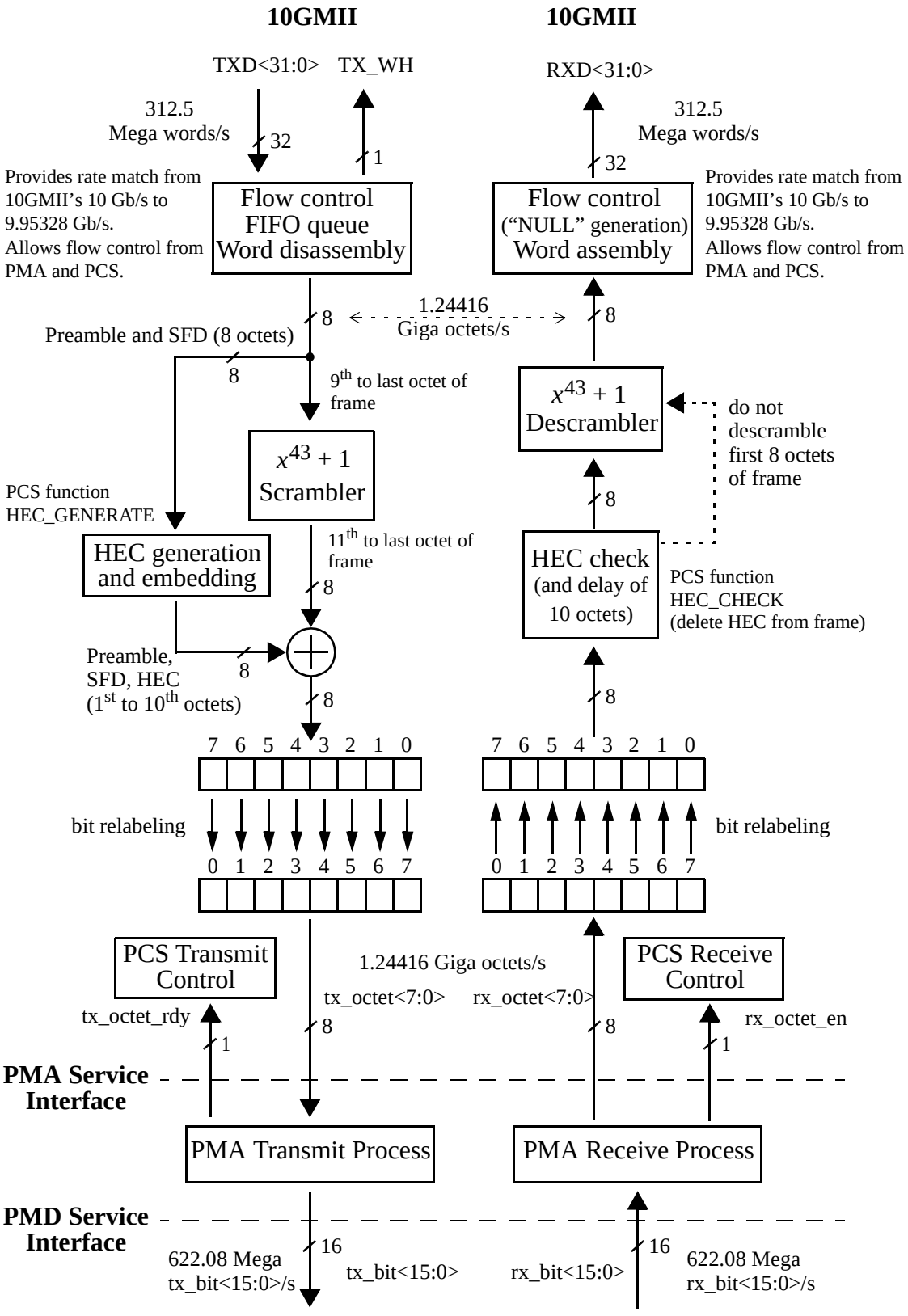
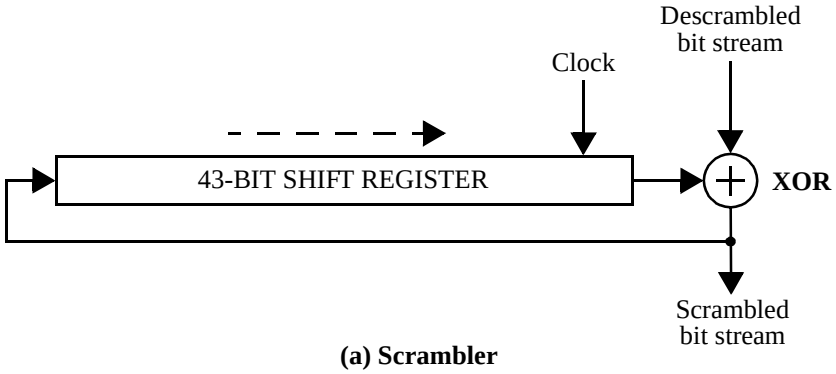


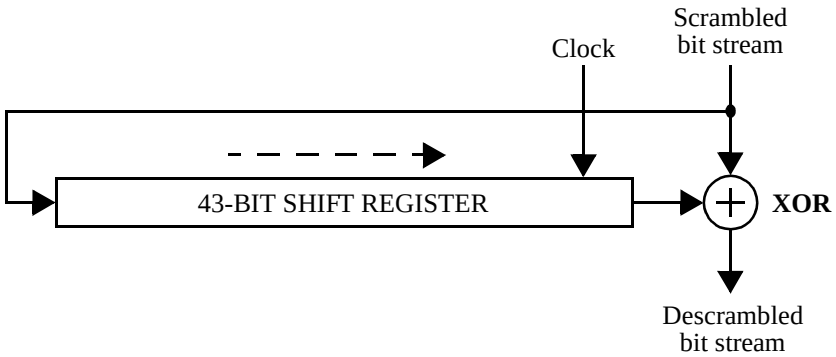
Figure 4-1—PCS reference diagram

4.4.1 Self-synchronous scrambler and descrambler

The $x^{43} + 1$ self-synchronous scrambler/descrambler is used by the PCS function SCRAMBLE and DESCRAMBLE. Figure 4-2 shows how the $x^{43} + 1$ self-synchronous scrambler and descrambler conceptually operate. At any given time, the state of the scrambler is the contents of the 43-bit shift register.



(a) Scrambler



(b) Descrambler

Figure 4-2—Self-synchronous scrambler (a) and descrambler (b) (functional diagram)

4.4.2 Bit order of scrambling and descrambling

Scrambling is done least significant bit first as shown in Figure 4-3. If scrambling and FCS calculation (which is done least significant bit first) were reversed with respect to each other, the short error burst detection capabilities of the FCS (i.e., CRC-32) would be degraded.

4.5 Use of header error control (HEC)

The header error control (HEC) field is embedded as a two-octet field after the SFD field of the MAC packet. The HEC field covers the Preamble and SFD fields (i.e., first eight octets) of the MAC packet, and is capable of single-error correction or multiple-error detection. A correct HEC field (see 4.5.1) means that no error was detected. An incorrect HEC field implies that there is at least one bit in error in the Preamble, SFD, or HEC fields.

The PCS Transmit process computes the two-octets of the HEC field (see 4.5.2) and embeds them into the octet stream of a MAC packet.

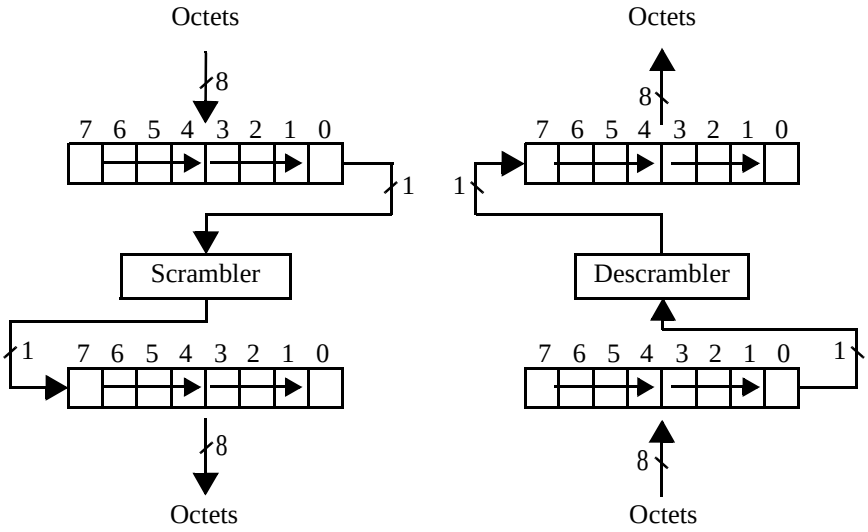


Figure 4-3—Bit order of scrambling and descrambling (functional diagram)

The PCS Receive process uses a modified version of the HEC check algorithm specified in ITU-T I.432, Clause 4.5.1.1, to provide MAC packet delineation. The HEC check algorithm (see 4.5.3) performs MAC packet delineation based on the correlation between the Preamble and SFD fields of the MAC packet and the embedded two-octet HEC field.

The HEC check algorithm defines a SYNC state where correct MAC packet delineation is assumed. In this state, incorrect HECs associated with single bit errors are corrected depending on the current HEC error handling mode, as shown in Figure 4-4. The CORRECTION mode provides for single-bit error correction and multi-bit error detection, while the DETECTION mode provides for single-bit and multi-bit error detection only. The HEC error handling mode is set to the CORRECTION whenever the HEC check algorithm moves from the PRESYNC to the SYNC state.

4.5.1 Header error control (HEC) field validation

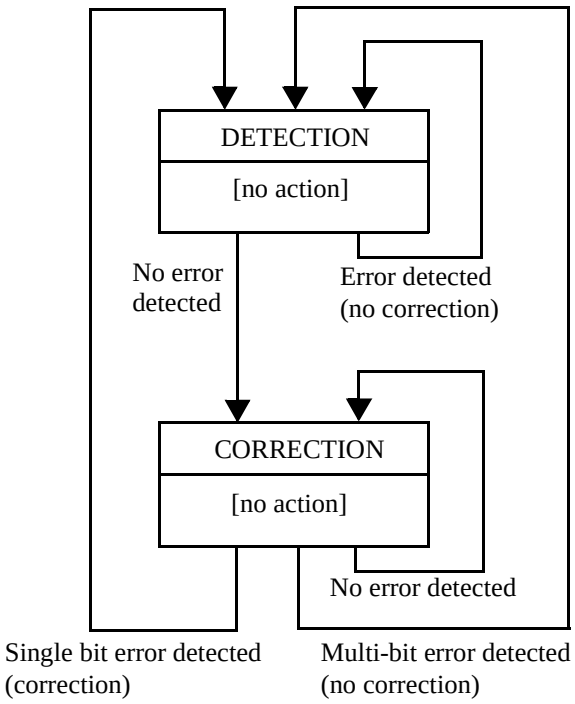
The PCS function HEC_VALIDATE, which is used by the PCS Receive process, receives octets and processes the last ten received octets to determine if the last two octets represent a correct HEC field value for the first eight octets. HEC field validation is essentially identical to HEC field generation (see 4.5.2). The last two octets of a ten-octet sequence is a correct HEC field if the bits of the first eight octets generate a HEC field value identical to the value of the last two octets of the sequence.

4.5.2 Header error control (HEC) field generation

The PCS function HEC_GENERATE calculates a HEC field value for the first eight octets of the MAC packet (i.e., Preamble and SFD fields) according to the following rules. The HEC field is the sum (module 2) of the 16-bit pattern “0101010101010101” (the left bit is the most significant bit) with a CRC-16 of the first eight octets of the MAC packet, as conceptually depicted in Figure 4-5. The CRC-16 is defined by the following generating polynomial.

$$G(x) = x^{16} + x^{12} + x^5 + 1$$

Mathematically, the CRC-16 value is defined by the following procedure:



Error detected = Incorrect HEC

Figure 4-4—HEC error handling modes

a) The first eight octets of the MAC packet are taken in order, least significant bit first, to form a 64-bit pattern representing the coefficients of a polynomial $M(x)$ of degree 63. The least significant bit of the first octet of the Preamble corresponds to the x^{63} term and the most significant bit of the SFD field corresponds to the x^{0} term.
b) $M(x)$ is multiplied by x^{16} and divided (module 2) by $G(x)$, producing a remainder $R(x)$ of degree ≤ 15.
c) The coefficients of $R(x)$ are considered to be a 16-bit sequence, where x^{15} is the most significant bit.
d) This 16-bit sequence is the CRC-16.

The PCS function HEC_GENERATE provides the HEC octets as depicted in Figure 4-5. The bit order of the two octets to be transmitted as the HEC field are reversed to agree with the bit order of the CRC-16 calculation.

4.5.3 The header error control (HEC) check algorithm

10GBASE-WX uses a modified version of the HEC check algorithm specified in ITU-T I.432, Clause 4.5.1.1, to provide MAC packet delineation. The HEC check algorithm used in 10 Gigabit Ethernet differs from that in ITU-T I.432 in two basic ways: 1- The algorithm uses the Length field of the Preamble[2] of the MAC packet to find the end of the packet; and 2- HEC field calculation uses a 16-bit polynomial and, consequently, generates a two-octet HEC field.

[2]We assume here that the length of the MAC frame (excluding the Preamble and SFD) is transmitted in the first two octets of the Preamble.

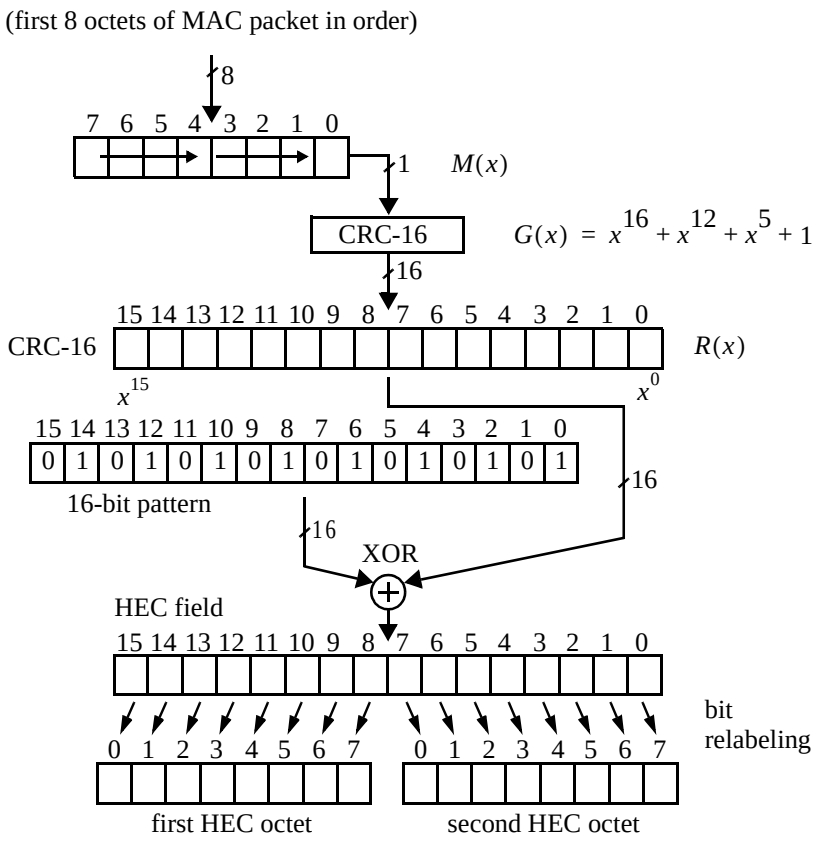


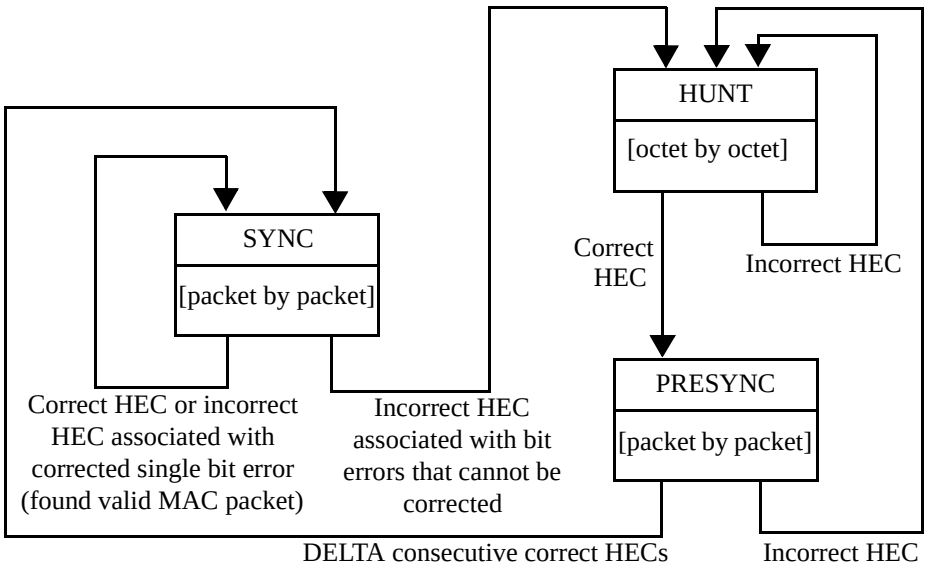
Figure 4-5—HEC octet generation (functional diagram)

The MAC packet delineation provided by the HEC check algorithm is completely transparent to the lower PHY sublayers. The scrambling of all octets after the HEC field (see 4.4) improves security and robustness of the MAC packet delineation algorithm.

MAC packet delineation is performed based on the correlation between the Preamble and SFD fields (i.e., first eight octets of the MAC packet) and the embedded two-octet HEC field. Figure 4-6 shows the state diagram of the MAC packet delineation method. The HEC check algorithm indicates octets that begin valid MAC packets.

The state diagram works as follows:

1) HEC field status, i.e., correct or incorrect, is determined by the PCS function HEC_VALIDATE (see 4.5.1).
2) In the HUNT state, the MAC packet delineation process is performed by checking octet by octet for a correct HEC for the sequence of the last ten octets, as defined in 4.5.1. Once a correct HEC is found, it is assumed that a MAC packet has been found, and the process enters the PRESYNC state.
3) In the PRESYNC state, the MAC packet delineation process is performed by checking packet by packet for a correct HEC. The process repeats until DELTA consecutive correct HECs are confirmed, at which point the process moves to the SYNC state. If an incorrect HEC is found, the process returns to the HUNT state. The total number of consecutive correct HECs required to move from the HUNT state to the SYNC state is therefore DELTA + 1. The Length field of



NOTE – The term "correct HEC" has the same meaning as the one defined in 4.5.1, i.e., the first eight octets of the ten-octet sequence generate a HEC value identical to the value of the last two octets. A correct HEC implies that no bit error was detected.

Figure 4-6—MAC packet delineation state diagram

the Preamble of the assumed MAC packet is used to find the beginning of the next assumed MAC packet.

4) In the SYNC state, MAC packet delineation is assumed lost if an incorrect HEC associated with bit errors that cannot be corrected is obtained (see Figure 4-4). In this case, the process enters the HUNT state. Correct HECs (or incorrect HECs associated with single bit errors that are corrected) that are processed while in the SYNC state shall indicate a valid MAC packet. The Length field of the Preamble of the MAC packet is used to find the beginning of the next assumed MAC packet.
5) Idle PHY packets (see 4.5.4) shall never indicate the beginning of MAC packets, i.e., idle PHY packets are silently discarded.

Robustness against false delineation in the re-synchronization process depends on the value of DELTA. The parameter DELTA is to be chosen to make the MAC packet delineation process as robust and secure as possible, and while satisfying the performance specified in 4.5.5. A value of DELTA = 1 is suggested.

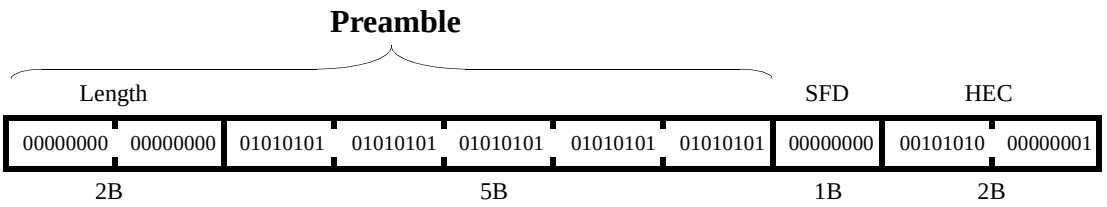
Idle PHY packets are generated by the PCS Transmit process whenever it needs to generate an octet on the PMA Service Interface and there is no MAC packet available for transmission (see 4.5.4).

4.5.4 Idle PHY packets

Idle PHY packets are generated by the PCS Transmit process whenever it needs to generate an octet and there is no MAC packet available for transmission. Idle PHY packets cause no action at the receiving PCS Receive process except for packet delineation including HEC verification. Idle PHY packets are identified by an SFD field with the bit sequence 00000000. Figure 4-7 shows the format of an idle PHY packet.

4.5.5 MAC packet delineation performance

This subclause is left for further study.



Note—As shown, the leftmost bit is the MSB of the octet and the rightmost bit is the LSB of the octet.

Idle PHY packet in hex = 00 00 55 55 55 55 55 00 2A 01

Figure 4-7—Idle PHY packet

4.6 Encapsulation

The 10GBASE-WX PCS accepts packets from the MAC through the Reconciliation sublayer and 10GMII, embeds the two-octet HEC field after the SFD field, and scrambles all the octets between the HEC field and the end of the MAC packet. The first two octets of the Preamble contain the Length subfield, which indicates the length in octets of the MAC packet beginning at the first octet of the DA field and ending at the last octet of the FCS field. Figure 4-8 depicts the PCS encapsulation of a MAC packet based on the octet stream transmitted by the MAC sublayer and PMA Service Interface signals.

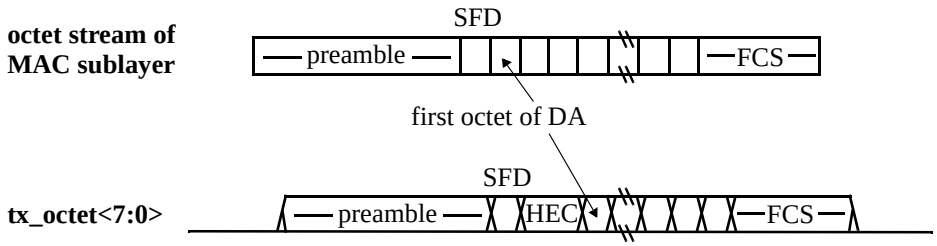


Figure 4-8—PCS encapsulation

4.7 PCS data delay

This subclause is left for further study.

4.8 Detailed functions and state diagrams

Most of this subclause is left for further study.

4.8.1 State variables

4.8.1.1 Notation conventions

TBD

4.8.1.2 Constants

TBD

4.8.1.3 Variables

rx_bit<15:0>
: A 16-bit vector of bits conveyed by the PMD_UNITDATA.indicate service primitive, as specified in TBD, to the PMA. The element rx_bit<15> is the most significant bit.

rx_octet<7:0>
: An 8-bit vector of bits represented by the most recently received octet from the PMA. This vector is conveyed to the PCS as the parameter of a PMA_UNITDATA.indicate(rx_octet<7:0>) service primitive.

rx_octet_en
: A signal used by the PMA sublayer to pace the transmission of octets to the PCS in order to allow the PMA to remove the octets of the STS-192 overhead. An octet is presented on rx_octet<7:0> only when rx_octet_en is High.

RXD<31:0>
: The RXD<31:0> signal of the 10GMII as specified in TBD. Set by the PCS Receive process.

RXC<3:0>
: The RXC<3:0> signal of the 10GMII as specified in TBD. Set by the PCS Receive process.

tx_bit<15:0>
: A 16-bit vector of bits used to convey data from the PMA to the PMD via the PMD_UNITDATA.request service primitive as specified in TBD. The element tx_bit<15> is the most significant bit.

tx_octet<7:0>
: An 8-bit vector of bits representing one octet that has been prepared for transmission by the PCS Transmit process. This vector is conveyed to the PMA as the parameter of a PMA_UNITDATA.request(tx_octet<7:0>) service primitive.

tx_octet_rdy
: A signal used by the PMA sublayer to pace the transmission of octets from the PCS in order to allow the PMA to add the octets of the STS-192 overhead. The PCS shall keep presenting the same octet on tx_octet<7:0> until it is sampled with tx_octet_rdy High.

TXD<31:0>
: The TXD<31:0> signal of the 10GMII as specified in TBD. Set by the Reconciliation Sublayer.

TXC<3:0>
: The TXC<3:0> signal of the 10GMII as specified in TBD. Set by the Reconciliation Sublayer.

TX_WH
: The TX_WH signal of the 10GMII as specified in TBD. Set by the PCS Transmit process.

More TBD

4.8.1.4 Functions

TBD

4.8.1.5 Counters

TBD

4.8.1.6 Message

TBD

4.8.1.7 Timer

TBD

4.8.2 State diagrams

TBD

4.8.2.1 Transmit

The PCS shall implement its Transmit process as depicted in Figures TBD, including compliance with the associated state variables as specified in 4.8.1.

More TBD

4.8.2.2 Receive

The PCS shall implement its Receive process as depicted in Figures TBD, including compliance with the associated state variables as specified in 4.8.1.

More TBD

5. Physical Medium Attachment (PMA) sublayer, type 10GBASE-WX

5.1 Service Interface

The PMA provides a Service Interface to the PCS. These services are described in an abstract manner and do not imply any particular implementation. The PMA Service Interface supports the exchange of octets between PCS entities. The PMA converts octets into 16-bit words and passes these to the PMD, and vice versa. It also generates additional status indication for use by its client.

The following primitives are defined:

PMA_UNITDATA.request(tx_octet<7:0>)
PMA_UNITDATA.indicate(rx_octet<7:0>)
PMA_FREQUENCY_STATUS.request(FREQUENCY_STATUS)

5.1.1 PMA_UNITDATA.request

This primitive defines the transfer of data (in the form of octets) from the PCS to the PMA. PMA_UNITDATA.request is generated by the PCS Transmit process.

5.1.1.1 Semantics of the service primitive

PMA_UNITDATA.request(tx_octet<7:0>)

The data conveyed by PMA_UNITDATA.request is the tx_octet<7:0> parameter. The tx_octet<7:0> is an 8-bit vector representing one octet that has been prepared for transmission by the PCS Transmit process, where tx_octet<7> is the most significant bit of the octet.

5.1.1.2 When generated

The PCS sends, at a nominal rate of 1.24416 GHz, as governed by the transmit clock of the PHY, i.e., 9.95328 GHz, tx_octet<7:0> to the PMA.

The transmission of tx_octet<7:0> to the PMA is only satisfied when tx_octet_rdy is asserted. If tx_octet_rdy is de-asserted, the PCS shall continuously send the same tx_octet<7:0> signal (i.e., octet) until tx_octet_rdy is asserted.

5.1.1.3 Effect of receipt

After the PMA acknowledges the receipt of this primitive (i.e., by the assertion of tx_octet_rdy – see 5.17), the PMA maps the received octet into the SPE. The PMA continuously generates PMD_UNITDATA.request primitives requesting transmission of tx_bit<15:0> to the PMD. The PMD transmits tx_bit<15:0> most significant bits first.

5.1.2 PMA_UNITDATA.indicate

This primitive defines the transfer of data (in the form of octets) from the PMA to the PCS. PMA_UNITDATA.indicate is used by the PCS Receive process.

5.1.2.1 Semantics of the service primitive

PMA_UNITDATA.indicate(rx_octet<7:0>)

The data conveyed by PMA_UNITDATA.indicate is the rx_octet<7:0> parameter. The rx_octet<7:0> parameter is an 8-bit vector representing the most recently received octet from the PMA, where rx_octet<7> is the most significant bit of the octet.

5.1.2.2 When generated

The PMA sends one rx_octet<7:0> to the PCS corresponding to the receipt of each octet aligned set of eight bits (received from the PMD) that corresponds to received payload octets. The nominal rate of the PMA_UNITDATA.indicate primitive is 1.24416 GHz, as governed by the bit clock.

5.1.2.3 Effect of receipt

The effect of receipt of this primitive by the client is unspecified by the PMA sublayer.

5.1.3 PMA_FREQUENCY_STATUS.request

This primitive defines the transfer of a frequency justification request from a PMA client to the PMA sublayer.

Note—This primitive is provided to support "IEEE regenerators."

5.1.3.1 Semantics of the service primitive

TBD

5.1.3.2 When generated

TBD

5.1.3.3 Effect of receipt

TBD

5.2 Functions within the PMA

Figures 4-1 depicts the mapping of the 32-bit-wide data path of the 10GMII to the octet-wide[3] data path of the PMA Service Interface, and on to the 16-bit-wide PMD Service Interface. The PMA comprises the PMA Transmit and PMA Receive processes for 10GBASE-WX.

The PMA Transmit process performs the following functions (see Figure 5-1):

a) PMA framing of octets received from the PCS sublayer by way of tx_octet<7:0>.
b) Scrambling of PMA frames using the $x^7 + x^6 + 1$ frame-synchronous scrambler.
c) Assembly of octets into tx_bit<15:0> for transmission to the underlying PMD Service Interface.

Similarly, the PMA Receive process performs the following functions (see Figure 5-1):

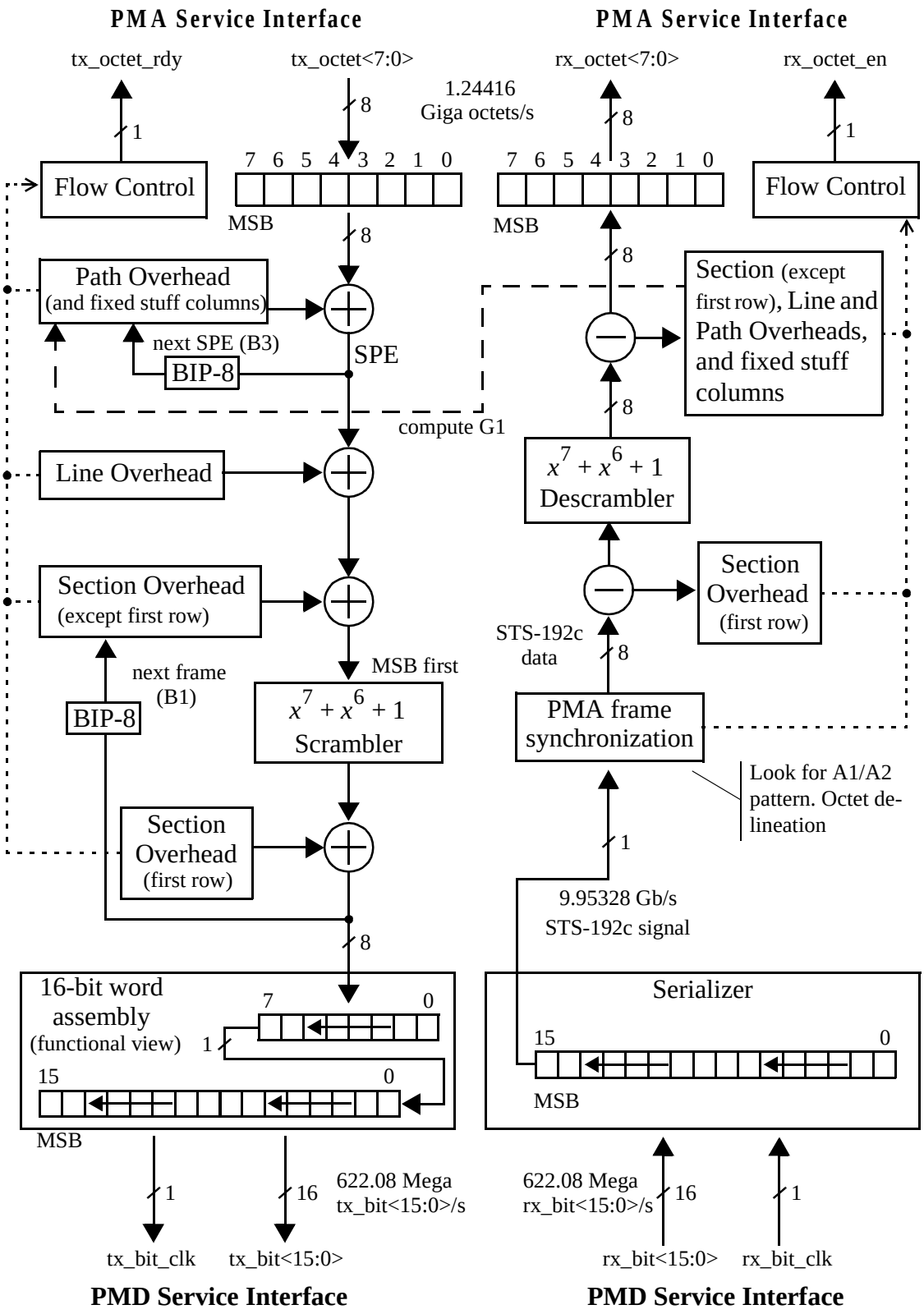
a) Serialization of rx_bit<15:0> received from the PMD sublayer.
b) PMA frame synchronization and octet delineation.
c) Descrambling of PMA frames using the $x^7 + x^6 + 1$ frame-synchronous scrambler.
d) Sending the octets of the payload of PMA frames to the PMA Service Interface by way of rx_octet<7:0>.

5.2.1 PMA transmit function

Figure 5-1 shows how the PMA forms the tx_bit<15:0> signal bundle (i.e., a 16-bit-wide STS-192 signal). This figure illustrates one possible set of stages in the formation of tx_bit<15:0>, and is not intended to depict a required, suggested, or preferred implementation.

The first stage involves the formation of an STS–192c SPE (see 5.3). The content of an STS–192c SPE is formed from the mapping of tx_octet<7:0> received from the PCS through the PMA_UNITDATA.request service primitive (see TBD), and from the addition of the STS Path Overhead (see 5.7 and Figure 5-10). Besides the STS Path Overhead, some octets of the STS-192c SPE are not used to transport payload. These octets are called "fixed stuff," as defined in 5.3 and depicted in Figure 5-4. The next stage involves frame aligning the STS–192 SPE (again, see 5.3), and the addition of the Line Overhead (see 5.6 and Figure 5-7).

[3]This specification defines the data path of the PMA in terms of octets for reference purposes only. Implementors may choose other data path widths for implementation convenience.



MSB = Most-significant bit

Figure 5-1—PMA reference diagram

The final part of the overhead, the Section Overhead, is added next (see 5.5 and Figure 5-7). Some octets of the Section Overhead are added before scrambling (see 5.9). The STS–192 frame is scrambled, and then the first row of the Section Overhead is added. The first row of the Section Overhead contains a special bit pattern (see 5.12) that is used by the PMA Receive process to determine the beginning of an STS-192 frame.

The STS–192 signal is then assembled into 16-bit-wide words for transmission to the PMD sublayer using the tx_bit<15:0> signal bundle. The resulting signal bundle has a rate of 622.08 Mega tx_bit<15:0>/s.

The PMA signal tx_octet_rdy is used to pace the PCS sublayer in order to allow for the insertion of the required overheads. An octet (by way of tx_octet<7:0>) is accepted for transmission by the PMA sublayer only when tx_octet_rdy is active.

5.2.2 PMA receive function

Figure 5-1 shows how the PMA sublayer processes the rx_bit<15:0> signal bundle (i.e., a 16-bit-wide STS-192 signal) received from the PMD sublayer. This figure illustrates one possible set of stages in the processing of the STS-192 signal, and is not intended to depict a required, suggested, or preferred implementation.

The PMA Receive function accepts a resulting 9.95328 Gb/s serial signal (the signal after the deserialization of the rx_bit<15:0> signal bundle, which has a rate of 622.08 Mega rx_bit<15:0>/s) either from the PMD sublayer or the PMA transmit function (the latter alternative is intended to provide a loopback function – this alternative is not shown in Figure 5-1).

The PMA frame synchronization unit (see 5.12) determines the beginning of the STS-192 frame and provides octet delineation using the framing pattern provided by the A1 and A2 octets (see 5.5.2) of the Section Overhead (see 5.5). Once the beginning of an STS-192 frame is determined, the PMA frame synchronization unit produces a stream of octets corresponding to the received STS-192 frame (shown as “STS-192c data” in Figure 5-1). If STS-192 frame synchronization is lost (as defined in 5.12), no STS-192c data is generated.

The first 576 octets of the STS-192c frame (i.e., the first row of the Section Overhead) are extracted from the octet stream before the descrambler unit (see 5.9). After the descrambler unit, the rest of the Section Overhead, the Line Overhead, the Path Overhead, and the fixed stuff columns are extracted from the octet stream. The remaining octets are mapped to rx_octet<7:0> and transmitted to the PCS using the PMA_UNITDATA.indicate service primitive. When active, the rx_octet_en signal indicates the transmission of an octet on the rx_octet<7:0> signal bundle.

5.3 Use of STS-192c frame structure

10GBASE-WX uses a (Synchronous Transport Signal – level 192) STS-192 frame structure, which has a bit rate of 9.95328 Gb/s. The optical counterpart of the STS-192 is the Optical Carrier – level 192 (OC-192) signal. An STS-192 signal is a sequence of 155,520 octets (1,244,160 bits), which include overhead octets (5,184 octets) and an envelop capacity for transporting payloads (150,336 octets). The frame format of the STS-192 signal can be seen as a 17,280 column by 9 row structure, as depicted in Figure 5-2, where the order of transmission of octets is top to bottom, row-by-row, left to right. Octets are transmitted most-significant bit (MSB) first, as shown in Figure 5-1.

The first 576 columns of the STS-192 frame are the Transport Overhead (Figure 5-3). The first three rows of the Transport Overhead are the Section Overhead (see 5.5), and the last six rows are the Line Overhead (see 5.6).

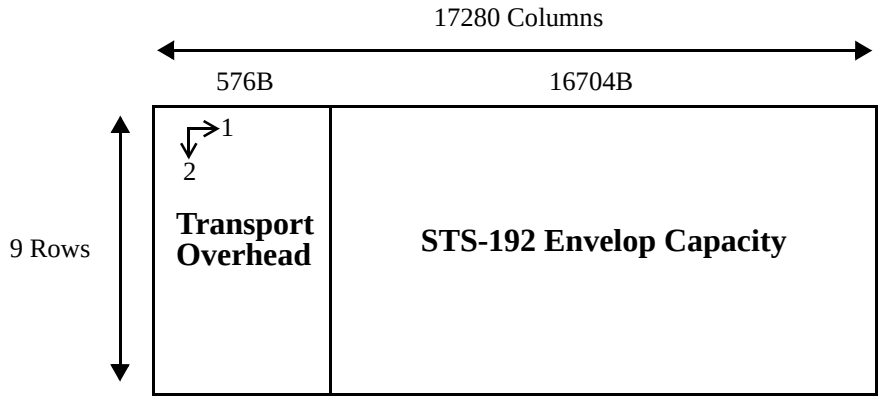


Figure 5-2—STS-192 frame format

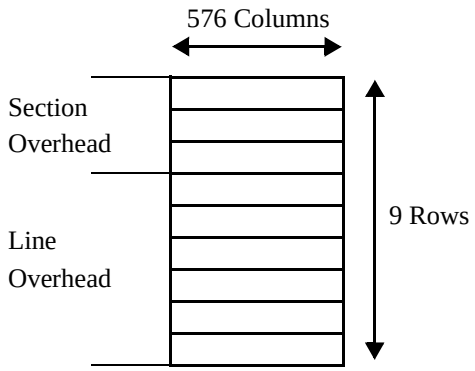


Figure 5-3—Transport overhead

After the columns of the Transport Overhead are the 16704 columns of the STS-192 Envelop Capacity. The Synchronous Payload Envelop (SPE) occupies the STS-192 Envelop Capacity. As depicted in Figure 5-4, the first column of the SPE contains the STS Path Overhead (POH), which is used to communicate various information from the point where payload is mapped into the STS-192 SPE to where it is delivered. After the STS POH, there are 63 columns of "fixed stuff," which represent unused octets that are set to an all-zeros pattern, i.e., 00000000. Subclause 5.7 contains details of the STS-192 POH. The resulting STS-192 SPE Payload Capacity is then 16640 columns by 9 rows (i.e., 149760 octets), which are used by the PMA sublayer to map payload octets (i.e., tx_octet<7:0> and rx_octet<7:0>).

Note— The STS-192c SPE fixed stuff columns are required to provide compatibility with SDH's (Synchronous Digital Hierarchy) virtual concatenation.

The STS-192 SPE may begin anywhere in the STS-192 Envelop Capacity. The STS-192 SPE may begin in one STS-192 frame and end in the next, as shown in Figure 5-5. By default, the PMA sublayer shall generate SPEs wholly contained in STS-192 frames (i.e., the STS pointer shall be set to 522 – see 5.6.2.2). However, PMA clients (e.g., "IEEE regenerators") may request frequency justification using the PMA service primitive PMA_FREQUENCY_STATUS.request, which provides either positive or negative octet stuffing that shifts the location of the SPE by 192 octets either forward or backward in relation to the STS-192 frame, respectively. To allow for the use of "IEEE regenerators" and to interwork with long haul OC-192 networks, the receiver PMA sublayer shall accept SPEs with arbitrary STS pointer values.

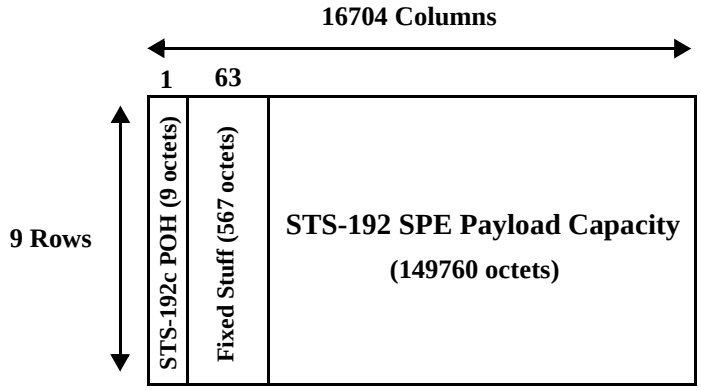


Figure 5-4—STS-192 SPE

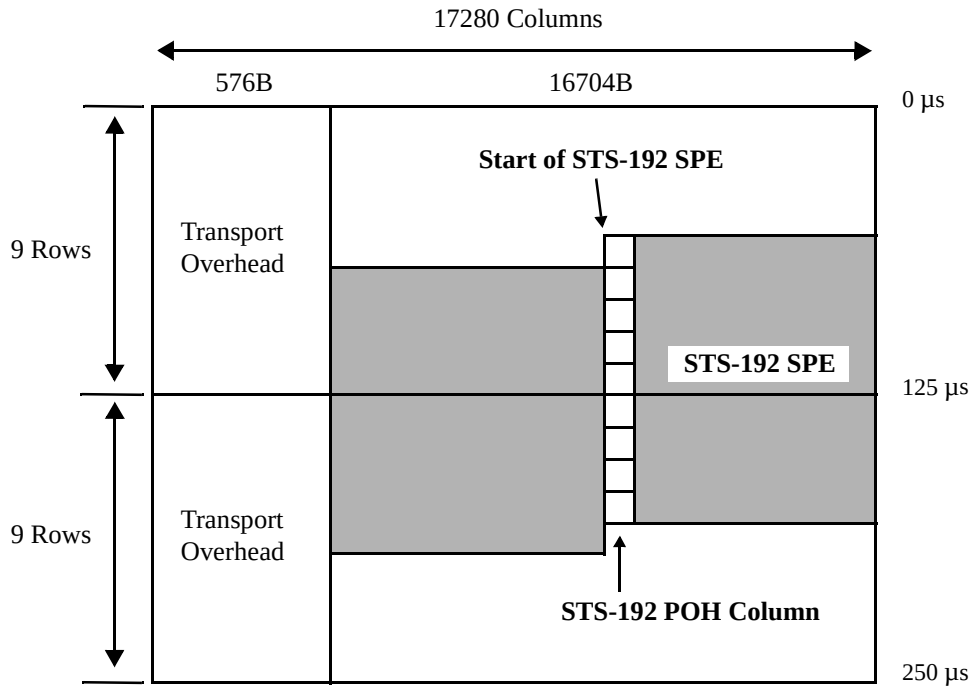


Figure 5-5—STS-192 SPE position in STS-192 frames

5.4 Use of Layered Overheads

The PMA frame is composed of the following hierarchy of overhead layers: Section Overhead, Line Overhead, and Path Overhead. Each overhead layer is logically built on top of the lower-level overhead layers, as depicted in Figure 5-6.

As explained in 5.3, the Transport Overhead is composed of the Section Overhead (see 5.5) and the Line Overhead (see 5.6). Figure 5-7 shows the Transport Overhead octet designations of the overhead octets used in 10GBASE-WX. The Path Overhead is addressed in 5.7.

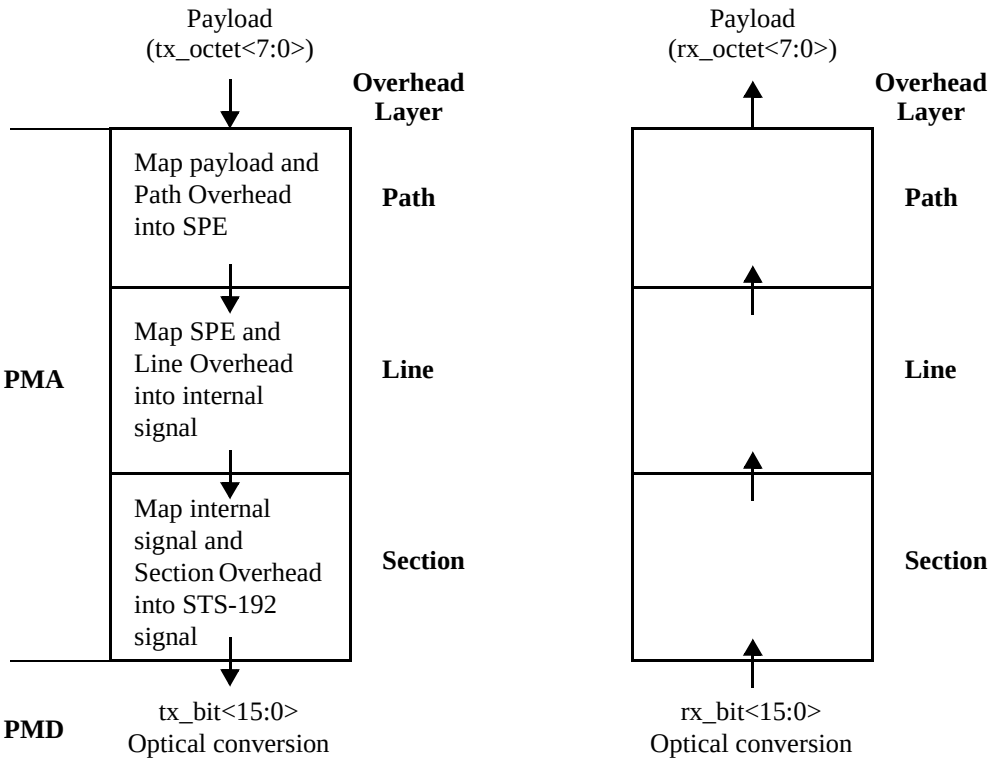


Figure 5-6—PMA overhead layers

5.5 Use of PMA Section Overhead

This subclause defines the Section Overhead octets (Figure 5-7) used by 10GBASE-WX.

The leftmost bit of all octets shown as bit patterns in this subclause is the MSB of the octet.

5.5.1 Undefined and unused Section Overheads

All undefined and unused octets of the Section Overhead shall be set to 00000000.

5.5.2 A1 and A2 (PMA Framing)

The A1 and A2 octets are used by the PMA frame synchronization procedure (see 5.12). The A1 octets shall be set to the bit sequence 11110110 (i.e., F6 hexadecimal) and all the A2 octets shall be set to the bit sequence 00101000 (i.e., 28 in hexadecimal).

5.5.3 J0 (Section Trace) and Z0 (Section Growth)

The J0 octet can be used to repetitively transmit a fixed value. This provides 256 possible values for the J0 octet. The purpose of this is to allow a receiver to verify its continued connection to the intended transmitter. The content of the J0 octet shall be user provisionable at both the transmit and receive equipment. When no value has been provisioned, then 00000010 shall be transmitted in the J0 byte.

Each Z0 octet shall contain the fixed pattern 11001100.

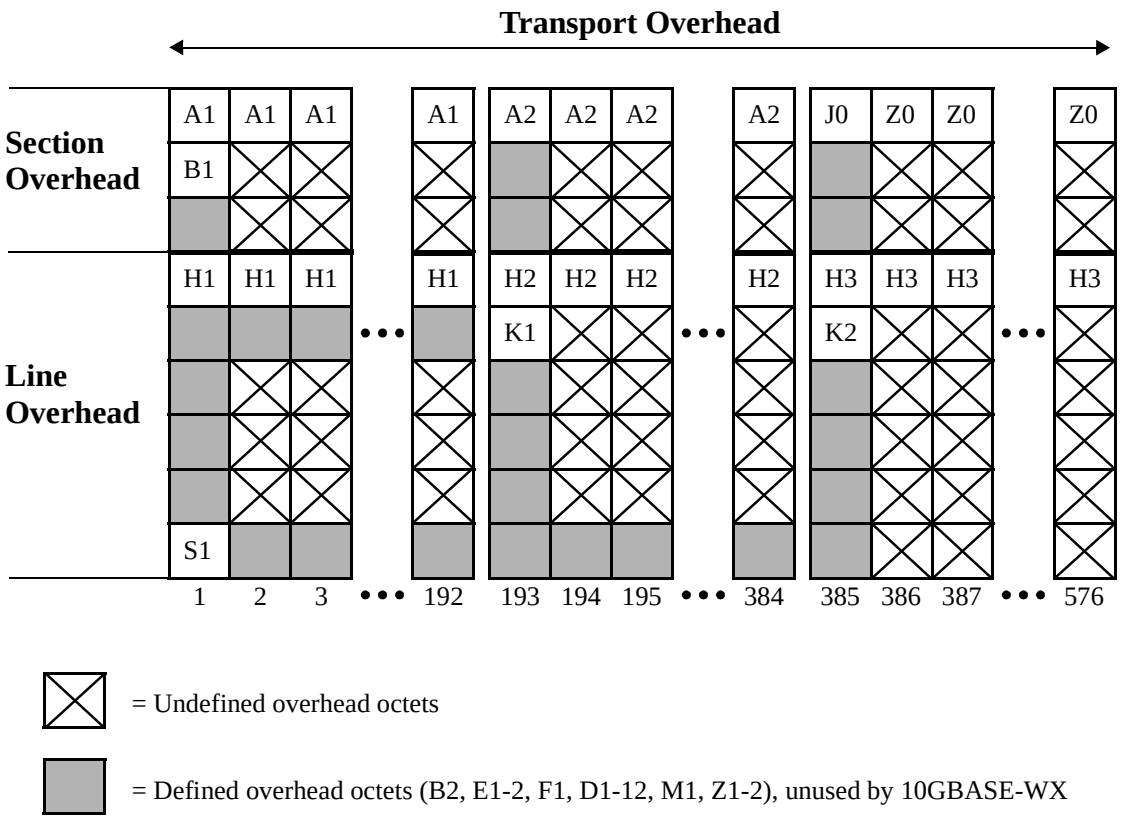


Figure 5-7—Transport overhead octet designations

5.5.4 B1 (Section BIP-8)

The B1 octet is used for a Section error monitoring function. A BIP-8 code is calculated over all bits of the last transmitted STS-192 frame after scrambling and placed in the B1 octet of the current STS-192 frame before scrambling. BIP-8 code calculation is defined in 5.8.

The use of received B1 octets by the PMA sublayer is for further study.

5.6 Use of PMA Line Overhead

This subclause defines the Line Overhead octets (Figure 5-7) used by 10GBASE-WX.

The leftmost bit of all octets shown as bit patterns in this subclause is the MSB of the octet.

5.6.1 Undefined and unused Line Overheads

All undefined and unused octets of the Line Overhead shall be set to 00000000.

5.6.2 H1 and H2 (STS Payload Pointer)

These octets allow the STS SPE to be dynamically aligned within the STS Envelop Capacity. These octets are also used to indicate concatenation. All H1 octets after the first one shall be set to 10010011. All H2 octets after the first one shall be set to 11111111. These octets indicate a concatenated STS-192c.

The coding of the first H1 and H2 octets is shown in Figure 5-8. The first H1 and H2 octets are seen as a single 16-bit word composed of the New Data Flag (NDF) field and the STS pointer field.

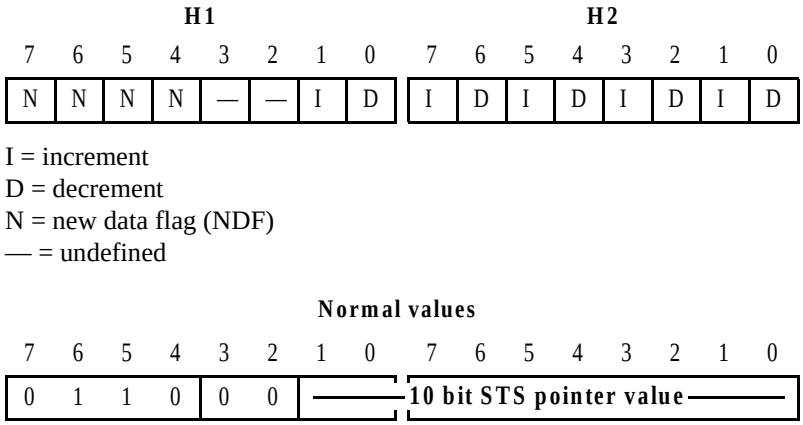


Figure 5-8—STS Payload pointer coding (first H1 and H2 octets)

5.6.2.1 New Data Flag (NDF) field

The NDF field[4] is used to set the STS SPE offset to an arbitrary value. To set the offset to the value indicated by the STS pointer field, the NDF field is set to 1001. Normal operation is indicated by a 0110 code (see Figure 5-8). The decoding of the NDF field shall be performed by majority voting, e.g., the NDF field shall be detected as being in the “normal” mode if three or four of the N-bits match the 0110 code.

5.6.2.2 STS Pointer

The STS pointer[5] value is a binary number in the range 0 to 782 (see Figure 5-9). The STS pointer value multiplied by 192 indicates the offset in octets between the last H3 octet of the Transport Overhead and the first octet of the STS-192c SPE (i.e., the J1 octet – see 5.7.2). The offset is taken without counting the Transport Overhead octets. For example, an STS pointer value of 0 indicates that the SPE starts in the octet location that immediately follows the last H3 octet, while an offset of 522 indicates that the SPE starts in the octet location that immediately follows the Z0 octet, i.e., the STS-192c SPE is wholly contained in the next STS-192 frame.

The STS pointer is divided into two sets of bits: the increment bits, which are labeled as I-bits in Figure 5-8, and the decrement bits, which are labeled as D-bits in Figure 5-8. An STS SPE offset increment operation shall be indicated by inverting the I-bits of the STS pointer. An STS SPE offset decrementing operation shall

[4]The NDF field is part of the first H1 octet only.
[5]The STS pointer is part of the first H1 and H2 octets only.

be indicated by inverting the D-bits of the STS pointer. These operations are used for STS SPE frequency justification.

The following criteria shall be applicable to the receiving of the STS pointer to provide protection against errors on the I- and D-bits. The current value of the STS SPE offset (which was acquired from a previous STS frame) is compared with the value of the received STS pointer. The increment/decrement decision should be made at the receiver by a match of 8 or more of the 10 bits of the STS pointer to either an increment or decrement indication. If the previous “8 of 10” objective is not met, then the increment decision shall be made by a majority vote of the I-bits, and the decrement decision shall be made by a majority vote of the D-bits. In this case, if a majority of the I-bits and a majority of the D-bits are detected to be inverted in the same STS pointer word, neither an increment nor a decrement decision shall be made.

As an example of increment and decrement indications that are used in the “8 of 10” objective, suppose that an STS pointer is received with three of the I-bits and two of the D-bits inverted (due to transmission errors) from the last known STS SPE offset value. In this case, six of the 10 I- and D-bits are correct for an increment operation, i.e., the three inverted I-bits plus the three non-inverted D-bits, while four of the 10 bits are correct for a decrement operation, i.e., the two non-inverted I-bits plus the two inverted D-bits. In this case, the “8 of 10” objective is not met.

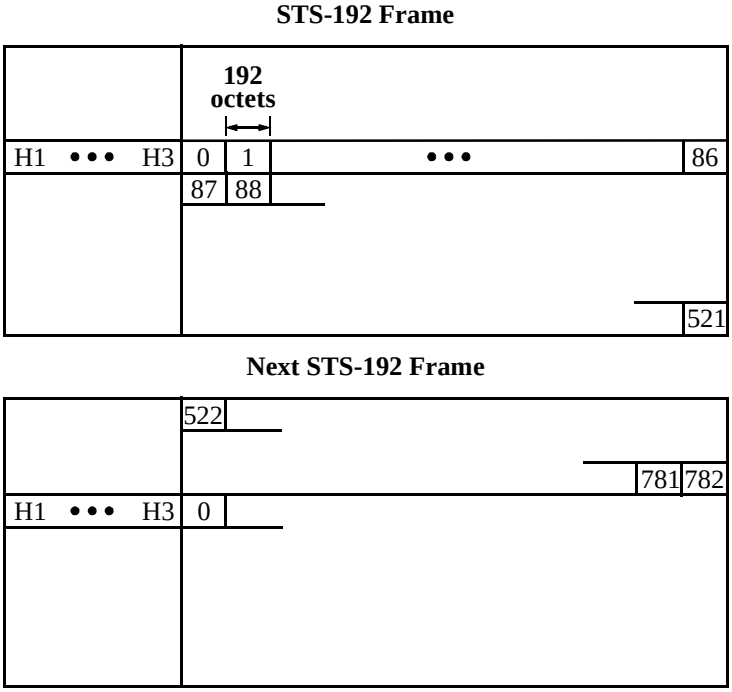


Figure 5-9—STS-192 pointer offset numbering

5.6.3 H3 (Pointer Action Byte)

These octets are used for STS SPE frequency justification purposes. These octets carry 192 extra SPE octets in the event of a negative pointer adjustment. The value contained in these octets when they are not used to carry SPE octets is undefined. The bit sequence 00000000 is suggested for all H3 octets in this case.

5.6.4 K1 and K2 (APS Channel)

The K1 octet shall be set to 00000001 and the K2 octet shall be set to 00010000.

Note—The K1 and K2 octets are used on the protection line for Automatic Protection Switching (APS) signaling between Line Terminating Equipment (LTE) that uses line protection switching. The K2 octet is also used to detect Line Alarm Indicator Signal (AIS-L) and Line Remote Defect Indication (RDI-L) signals. The above settings indicate a working channel rather than the protection channel.

5.6.5 S1 (Synchronization Status)

S1 shall be set to 00001111.

Note—The S1 octet is used to convey PMA clock quality information to allow receivers to select the most suitable synchronization reference from the set of available references. The above setting indicates the "DON'T USE for synchronization" (DUS) message.

5.7 Use of PMA Path Overhead

This subclause defines the Path Overhead octets (Figure 5-10) used by 10GBASE-WX.

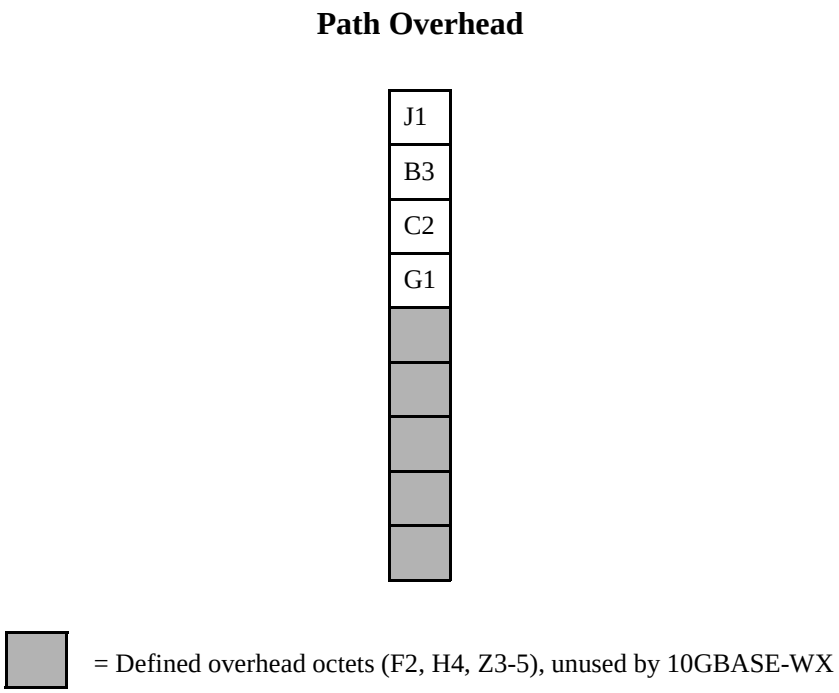


Figure 5-10—Path overhead octet designations

The leftmost bit of all octets shown as bit patterns in this subclause is the MSB of the octet.

5.7.1 Undefined and unused Path Overheads

All unused octets of the Path Overhead shall be set to 00000000.

5.7.2 J1 (STS Path Trace)

The J1 octet shall be set to 00000000.

Note—This octet is used to transport a repetitive 64-octet message so that the receiver can verify its continued connection to the intended transmitter. The use of this facility with values different from 00000000 is left for further study.

5.7.3 B3 (STS Path BIP-8)

The B3 octet is used for an STS Path error monitoring function. A BIP-8 code (using even parity) shall be calculated over all bits of the last transmitted STS-192 SPE before scrambling and placed in the B3 octet of the current STS POH before scrambling. BIP-8 calculation is described in 5.8.

5.7.4 C2 (STS Path Signal Label)

The C2 octet is used to indicate the content of the STS SPE. For 10GBASE-WX, C2 shall be set to TBD.

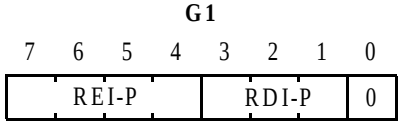
Note—We need to obtain a standard value from ITU.

5.7.5 G1 (Path Status)

The G1 octet is used for conveying the Path terminating status and performance back to the transmitter. This feature allows the status and performance of the complete duplex path to be monitored at either end, or at any point along the path.

The G1 octet shall be set as indicated in Figure 5-11. The Path Remote Error Indication (REI-P) field contains the count of bit errors that is detected based on the STS Path BIP-8 octet, i.e., B3, of the last received STS SPE. The error count shall be a binary number from zero (i.e., 0000) to eight (i.e., 1000). The count values 1001 to 1111 shall not be transmitted, and shall be interpreted by receivers as zero errors.

The Path Remote Defect Indication (RDI–P) field shall be calculated (appropriate values are for further study). Bit 0 (LSB) of the G1 octet shall be set to 0.



REI-P Coding:

0000	0 errors
0001	1 error
0010	2 errors
0011	3 errors
0100	4 errors
0101	5 errors
0110	6 errors
0111	7 errors
1000	8 errors
1001 to 1111	0 errors

Figure 5-11—STS Path Status Octet (G1)

5.8 Use of bit-interleaved parity 8 (BIP-8)

Bit-Interleaved Parity 8 (BIP–8) is a method of error monitoring based on a parity check. BIP-8 with even parity is an 8-bit code where the i^{th} bit of the code provides even parity over the i^{th} bit of all 8-bit sequences in the covered portion of the signal. Even parity is generated by setting the BIP–8 bits so that there are an even number of ones in each of all 8-bit sequences, including the BIP–8. For example, the BIP-8 of the bit sequence 11110000 00001111 is 11111111.

5.9 Use of the $x^7 + x^6 + 1$ frame synchronous scrambler/descrambler

To assure that the optical interface signal has an adequate number of transitions for line rate clock recovery at the receiver, the optical interface signal is scrambled using a frame synchronous scrambler, which is applied identically at the transmitter and the receiver. The scrambler uses a 7-bit linear feedback shift register (LFSR) operating at the line rate with the generating polynomial $x^7 + x^6 + 1$ as shown in Figure 5-12.

The shift register of the scrambler shall be reset to 1111111 on the most-significant bit of the octet following the last Z0 octet of the Section Overhead. That bit and all subsequent bits to be scrambled shall be added, modulo 2, to the output from the x^7 position of the scrambler, as shown in Figure 5-12. The scrambler shall run continuously from that bit on throughout the remainder of the STS–192 frame. Scrambling is done most significant bit first.

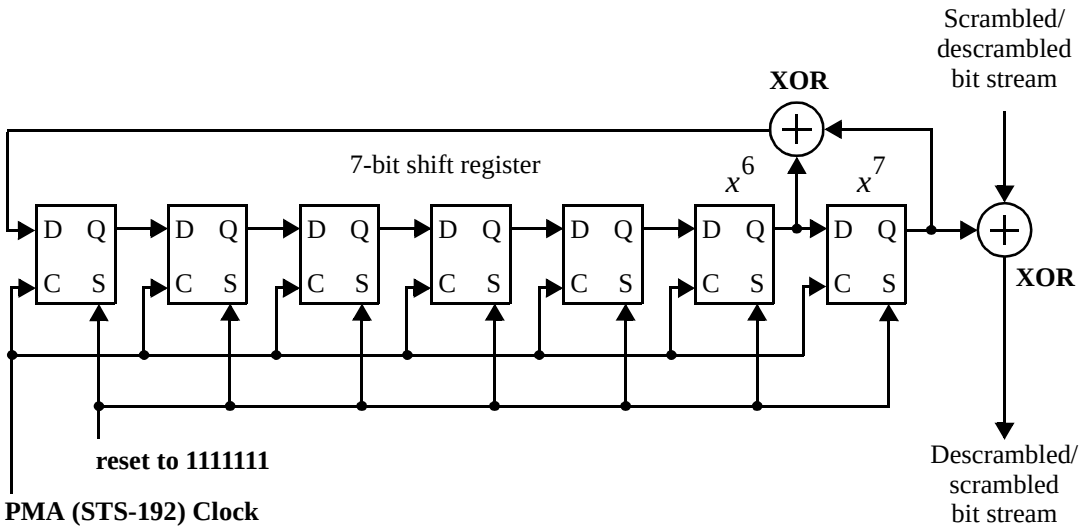


Figure 5-12—Frame synchronous scrambler/descrambler (functional diagram)

Note—The A1, A2, J0, and Z0 octets are not scrambled.

5.10 Example of STS-192 signal composition

Figure 5-1 shows how the PMA forms the STS–192 signal. Figure 5-1 illustrates one possible set of stages in the formation of the STS-192 signal.

The content of an STS–192c SPE is formed from the mapping of tx_octet<7:0> received from the PCS through the PMA_UNITDATA.request service primitive (see 5.1.1). Included within the STS–192c SPE is the STS Path Overhead (see 5.7 and Figure 5-10), which includes the J1, B3, C2 and G1 octets. The remaining octets of the Path Overhead are set to an all-zeros pattern (i.e., 00000000). Also included within the STS-192-c SPE are the fixed stuff columns (see Figure 5-4), which are all set to an all-zeros pattern. The BIP–8 error check octet is calculated over the entire STS SPE and the result is placed in the B3 octet of the following STS SPE. The G1 octet is calculated based on the STS Path BIP-8 octet, i.e., B3, of the last received STS SPE, and other information defined in 5.7.5.

The next stage involves frame aligning the STS–192 SPE, and the addition of the Line Overhead (see 5.6 and Figure 5-7), which includes the following octets: Payload Pointer/Pointer Action (H1, H2, H3), K1, K2, and S1. The remaining octets of the Line Overhead are set to an all-zeros pattern.

The final part of the overhead, the Section Overhead, is added next (see 5.5 and Figure 5-7). The Section BIP–8 (B1) is added before scrambling. The B1 octet is calculated based on the last transmitted STS-192 frame after scrambling. An all-zeros pattern is added in the undefined and unused octets. The STS–192 frame is scrambled, and then the framing (A1, A2) octets, Section Trace (J0) octet, and Growth (Z0) octets are added.

The STS–192 signal is then assembled into 16-bit words for transmission to the PMD sublayer using the tx_bit<15:0> signal bundle. The resulting signal has a bit rate of 9.95328 Gb/s.

5.11 Clock recovery

Clock recovery is provided by the PMD (as implied in Figure 5-1 by the rx_bit_clk signal). The clock recovery unit of the PMD shall receive and process incoming STS-192 signals with bit rates that are, at a minimum, anywhere in the range ±100 ppm off frequency from the nominal bit rates for those signals, i.e., 9.95328 Gb/s.

5.12 PMA frame synchronization

The A1 and A2 octets of the Section Overhead contain a special bit pattern (i.e., a framing pattern) that is used by the PMA frame synchronization unit to determine where the STS-192 frame starts. The PMA frame synchronization unit looks for the framing pattern consistently, expecting to see it appear once every 150,520 octets (i.e., the length of the STS-192 frame). When the framing pattern appears in the right place enough times, the PMA frame synchronization unit assumes that it has achieved frame synchronization. In this case, the PMA state variable frame_in_sync is asserted, and an octet-aligned STS-192 frame (i.e., STS-192 data) is passed to the next unit, as shown in Figure 5-1. If frame_in_sync is de-asserted, no STS-192 data is passed to the next unit.

This subclause does not provide specific details on how a frame synchronization algorithm works. In addition, it does not imply any specific implementation. Any implementation of the frame synchronization procedure that complies with the following rules is acceptable:

1) Frame synchronization (or alignment) shall be detected by searching for a subset of the A1 and A2 octets contained in the incoming STS-192c signal. The framing pattern does not necessarily have to consist of full octets. For example, the frame synchronization unit could use certain bits of a particular A1 octet, along with all or parts of one or more other A1 and A2 octets as the framing pattern.
2) At power on or main reset, the state variable frame_in_sync is de-asserted and the state variable framing_error is asserted.
3) When looking for the framing pattern, the PMA state variable framing_error shall be asserted if a framing pattern is not found in 625 µs. In addition, framing_error shall be asserted if the framing pattern is not found where it was expected to be for four or more consecutive times.
4) The frame synchronization algorithm used to check the frame alignment shall be such that framing_error is not asserted more than an average of once every 6 minutes while the bit error rate (BER) of the STS-192c signal is 10^{-3}, where Poisson distribution of bit errors is assumed.
5) Once framing_error is asserted, it is only de-asserted upon detecting two consecutive error-free framing patterns. Any implementation that de-asserts framing-error within the 250 µs interval implied by the preceding requirement is acceptable.
6) Rules 3 (i.e., asserting framing_error) and 5 (i.e., de-asserting framing_error) may use different subsets of the A1 and A2 octets as the framing pattern.
7) The PMA state variable frame_in_sync shall be de-asserted when framing_error is persistently in the asserted state for 3 ms or more.
8) Except when the optional 3 ms integration timer described in rule 9 is used, the PMA state variable frame_in_sync shall be asserted within 1 ms to 3 ms after framing_error is de-asserted, if

framing_error is not asserted before frame_in_sync is asserted. That is, frame synchronization requires a minimum of eight (8) and a maximum of twenty-four (24) consecutive error-free framing patterns after framing_error is de-asserted to assert frame_in_sync.

9) The following optional 3 ms integration timer may be used to deal with intermittent framing errors:
   i) A framing_error_integration timer is activated (i.e., accumulates) whenever framing_error is in the asserted state, and is deactivated (i.e., stops accumulating) whenever framing_error is in the de-asserted state. The timer is reset to zero whenever framing_error is continuously in the de-asserted state for 3 ms.
   ii) The variable frame_in_sync is de-asserted when framing_error_timer reaches 3 ms.
   iii) Once de-asserted, frame_in_sync is asserted when framing_error is continuously in the de-asserted state for 3 ms.
10) If the optional 3 ms integration timer described in rule 9 is used, the supplier shall clearly describe its use in the product documentation.

5.13 Loss of STS-192c signal

To detect failures that occur at the transmitter (such as laser failure) or within the transmission facility (such as a fiber cut), the PMA shall monitor the incoming STS-192 signal for an “all-zeros pattern,” i.e., no voltage transitions. The state variable loss_of_signal shall be asserted when an all-zeros pattern on the incoming STS-192c signal is detected. How an all-zeros pattern is detected is not specified and left to the choice of the equipment designer. However, an all-zeros pattern shall be detected in less than 3 ms.

The state variable loss_of_signal shall be de-asserted when two consecutive error-free framing patterns (see 5.12) are found in the incoming signal, if no all-zeros pattern is detected in the intervening time.

Note—The loss_of_signal signal should be mapped to a PHY register that is TBD.

5.14 STS Payload pointer generation and interpretation rules

This clause summarizes the STS payload pointer generation and interpretation rules.

The STS Payload Pointer shall be generated according to the following rules:

1) During normal operation, a “normal NDF” is sent, i.e., the N–bits of the first H1 octet are set to 0110 (see Figure 5-8), and the pointer value locates the start of the STS SPE within the STS Envelope Capacity.
2) The pointer value shall only be changed by the operations in rules 4, 5, or 6.
3) An STS SPE pointer word is generated for the first H1/H2 octets. The concatenation indicator is generated in the other H/H2 octets, as described in 5.6.2.
4) If a positive stuff is needed, the current STS pointer value is sent with the I–bits inverted, and the subsequent 192 positive stuff opportunities (i.e, the 192 octets following the last H3 octet) are considered undefined octets, and should be set to an all-zeros pattern. Subsequent STS pointers contain the previous pointer value incremented by one.
5) If a negative stuff is needed, the current STS pointer value is sent with the D–bits inverted, and the subsequent 192 negative stuff opportunities (i.e., the 192 H3 octets) are overwritten with SPE octets. Subsequent STS pointers contain the previous pointer value decremented by one.
6) If the alignment of the SPE changes for any reason other than rules 4 or 5, the new STS pointer value shall be sent accompanied by a “set NDF,” i.e., the N–bits of the first H1 octet are set to 1001 (see Figure 5-8). The set NDF only appears in the first STS frame that contains the new value. The new SPE begins at the first occurrence of the offset indicated by the new STS pointer value.

7) No increment or decrement operation shall be performed for three STS frames following any of the operations in rules 4, 5, and 6.

Note—In the pointer interpretation rules that follows, there is no rule or requirement equivalent to pointer generation rule 7. If a pointer processor detects an increment or decrement operation within three frames after another pointer change operation (e.g., due to transmission errors), it can either ignore that operation or interpret it as a valid operation.

The STS Payload Pointer shall be interpreted according to the following rules:

1) During normal operation, the STS pointer value locates the start of the STS SPE within the STS Envelope Capacity.
2) Any variation from the current STS pointer value shall be ignored unless a consistent new value is received three times consecutively, or the variation is one of the operations in rules 3, 4, or 5 below. Any consistent new value received three times in succession shall replace the current value at the offset indicated by the new pointer value.
3) If an increment is detected, then the 192 octets following the last H3 octet shall be considered positive stuff octets, and the current STS SPE offset value shall be incremented by one.
4) If a decrement is detected, then the 192 H3 octets shall be considered negative stuff octets, and the current STS SPE offset value shall be decremented by one.
5) If a “set NDF” is detected, i.e., the N–bits of the first H1 octet are set to 1001 (see Figure 5-8), then the STS pointer value replaces the current STS SPE offset value.

5.15 PMA Data delay

The PMA maps a nonaligned 16-bit data path from the PMD to an aligned octet-wide data path to the PCS on the receive side. Logically, received bits must be buffered to facilitate proper octet alignment and STS frame synchronization. Octet alignment necessitates an internal PMA delay of at least eight bit times. STS frame synchronization may necessitate even longer delays of the incoming rx_bit<15:0> stream.

5.16 Detailed functions and state diagrams

Most of this subclause is left for further study.

5.16.1 State variables

5.16.1.1 Notation conventions

TBD

5.16.1.2 Constants

TBD

5.16.1.3 Variables

TBD

5.16.1.4 Functions

TBD

5.16.1.5 Counters

TBD

5.16.1.6 Message

TBD

5.16.1.7 Timer

TBD

5.16.2 State diagrams

TBD

5.16.2.1 Transmit

The PMA sublayer shall implement its Transmit process as depicted in Figures TBD, including compliance with the associated state variables as specified in 5.16.1.

More TBD

5.16.2.2 Receive

The PMA sublayer shall implement its Receive process as depicted in Figures TBD, including compliance with the associated state variables as specified in 5.16.1.

More TBD

5.17 A physical instantiation of the PMA Service Interface

This subclause provides an example of a physical instantiation of the PMA Service Interface. There is no requirement for a compliant device to implement or expose this proposed physical instantiation.

The rest of this clause is for further study.