INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2017-2020

SG15-TD152R1/PLEN

STUDY GROUP 15

Original: English

Question(s):	11/15	Geneva, 29 January – 9 February

TD

Source:	Editor G.8023	
Title:	Draft new Recommendation G.8023 (for consent)	
Purpose:	Discussion	

Contact:	Tom Huber Coriant GmbH & Co KG Germany	Tel: +1 630 798 6625 Fax: Email: tom.huber@coriant.com
Contact:	Wei Su Huawei Technologies Co., Ltd. P.R. China	Tel: +86 28 62840979 Fax: Email: water.suwei@huawei.com

This document provides the draft text for new Recommendation G.8023. It is based on the material presented in C82 at the June 2017 plenary meeting, with some editorial modifications in clause 6 related to naming the Ethernet sublayers, simplifying the description of the adaptations to the Ethernet Coding sublayer, and more clearly specifying the server layer for the adaptations that are specific to 10GE (i.e., the one that is used with ODU2e mapping and the one that is used with the PP-OS mapping). It was further modified in the Q11-12 interim meeting and during correspondence.

Updated with general editorial updates to references, definitions, acronyms, changes from TD191/3, changes from C710, and further changes based on drafting sessions,

Recommendation ITU-T G.8023

Characteristics of equipment functional blocks supporting Ethernet physical layer and Flex Ethernet interfaces

Summary

Recommendation ITU-T G.8023 specifies both the functional components and the methodology that should be used in order to specify the Ethernet physical layer and Flex Ethernet interfaces.

Keywords

Ethernet, physical layer, functional blocks

Introduction

This Recommendation forms part of a suite of ITU-T Recommendations covering the full functionality of Ethernet transport network architecture and equipment (e.g., Recommendations ITU-T G.8010/Y.1306 and ITU-T G.8021/Y.1341) and follows the principles defined in Recommendation ITU-T G.806.

This Recommendation specifies a library of basic building blocks and a set of rules by which they may be combined in order to describe Ethernet physical layer and Flex Ethernet interfaces. The building blocks are based on atomic modelling functions and combination rules defined in Recommendations ITU-T G.806.

This Recommendation includes atomic functions that replace the atomic functions that were previously published in clause 10 of Recommendation ITU-T G.8021 (2016).

1 Scope

This Recommendation specifies the functions required to insert and extract information to/from an Ethernet PHY as defined in IEEE Std 802.3, including the FlexE shim as defined in the OIF-FLEXE implementation agreement, for the purposes of mapping into an ODU (as defined in Recommendation ITU-T G.798) or performing MAC layer operations on Ethernet frames (as defined in Recommendation ITU-T G.8021).

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[IEEE 802.1AB] IEEE Std. 802.1AB (2016), *IEEE Standard for Local and Metropolitan Area Networks: Station and Media Access Control Connectivity Discovery*.

[IEEE 802.1X] IEEE Std. 802.1X (2010), *IEEE Standard for Local and metropolitan area networks – Port-Based Network Access Control*.

[IEEE 802.3] IEEE Std. 802.3 (2015), *IEEE Standard for Ethernet*

[IEEE 802.3bs] IEEE Std 802.3bs (2017), *Media Access Control Parameters, Physical Layers and Management Parameters for 200 Gb/s and 400 Gb/s Operation*

[IEEE 802.3by] IEEE Std. 802.3by (2016), *IEEE Standard for Ethernet – Amendment 2: Media Access Control Parameters, Physical Layers, and Management Parameters for 25 Gb/s Operation*.

[ITU-T G.798] Recommendation ITU-T G.798 (2017), *Characteristics of optical transport network hierarchy equipment functional blocks*.

[ITU-T G.805] Recommendation ITU-T G.805 (2000), *Generic functional architecture of transport networks..*

[ITU-T G.806] Recommendation ITU-T G.806 (2012), *Generic functional architecture of transport networks.*

[ITU-T G.809] Recommendation ITU-T G.809 (2003), *Functional architecture of connectionless layer networks.*

[ITU-T G.959.1] Recommendation ITU-T G.959.1 (2016), O*ptical transport network physical layer interfaces*.

[ITU-T G.8021] Recommendation ITU-T G.8021 (2018), *Characteristics of Ethernet transport network equipment functional blocks*.

[OIF FLEXE IA] OIF, *Flex Ethernet Implementation Agreement 1.1* (2017).

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 1000BASE-X [IEEE 802.3]

3.1.2 100GBASE-R [IEEE 802.3]

3.1.3 200GBASE-R [IEEE 802.3]

3.1.4 25GBASE-R [IEEE 802.3]

3.1.5 40GBASE-R [IEEE 802.3]

3.1.6 400GBASE-R [IEEE 802.3]

3.1.7 access point [ITU-T G.805]

3.1.8 adaptation [ITU-T G.805]

3.1.9 adapted information [ITU-T G.805]

3.1.10 characteristic information [ITU-T G.805]

3.1.11 connection point [ITU-T G.805]

3.1.12 consequent actions [ITU-T G.805]

3.1.13 defect correlations [ITU-T G.805]

3.1.14 defects [ITU-T G.805]

3.1.15 flow point [ITU-T G.809]

3.1.16 optical tributary signal [ITU-T G.959,1]

3.1.17 reconciliation sublayer [IEEE 802.3]

3.1.18 termination flow point [ITU-T G.809]

3.1.19 traffic unit [ITU-T G.809]

3.1.20 trail termination [ITU-T G.805]

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 electrical tributary signal: Electrical signal that is placed within a network media channel for transport across an electrical link.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AI	Adapted Information
AP	Access Point
CBR	Constant Bit Rate
CI	Characteristic Information
CK	Clock
COMMS	Communications channel
CP	Connection Point
CRC	Cyclic Redundancy Check
D	Data
DE	Drop Eligibility
ETH	Ethernet MAC layer
ERSy	Ethernet Reconciliation sublayer for PHY y
ETCy	Ethernet Coding sublayer for PHY y
ETSi[G]	Electrical Tributary Signal or Electrical Tributary Signal Group
FCS	Frame Check Sequence
FlexEC	Flex-Ethernet Client
FlexE	Flex-Ethernet
FP	Flow Point
FS	Frame Start
MAC	Media Access Control
MFS	MultiFrame Start
MI	Management Information
MP	Management Point
OTSi[G]	Optical Tributary Signal or Optical Tributary Signal Group

P	Priority
PCS	Physical Coding Sublayer
PHY	Ethernet Physical layer
PMA	Physical Medium Attachment
PP	replication Point
RP	Remote Point
SSF	Server Signal Fail
TCP	Termination Connection Point
TFP	Termination Flow Point
TP	Timing Point
VLAN	Virtual Local Area Network

5 Conventions

For the basic methodology to describe transport network functionality of network elements, refer to clause 5 of [ITU-T G.806]. For Ethernet-specific extensions to the methodology, see clause 5 of [ITU-T G.8010].

The convention MI_[802.3] is used to indicate the MI_ input/output signals required to map the management attributes, defined in clause 30 of [IEEE 802.3], for the [IEEE 802.3] processes supported by this adaptation function. Their detailed definition is intentionally left outside the scope of this Recommendation.

The following conventions are used in naming layer networks:

The server layer for the adaptation functions is modelled as a group of one or more optical or electrical signals. xTSi[G] is used to indicate this generically. The ‘x’ takes the value O or E when naming specific server layers; the G is present for the case where a group of signals is used or absent when only a single signal is present.

y is used as a variable for the client layer in the adaptation functions, specifically:

CBRy is used to indicate a nominal rate of a constant bit rate signal, with ‘y’ indicating the rate using the form "unit value, unit, [fractional unit value]", for example 10G3 for a signal with a nominal rate of 10.3 Gbps.

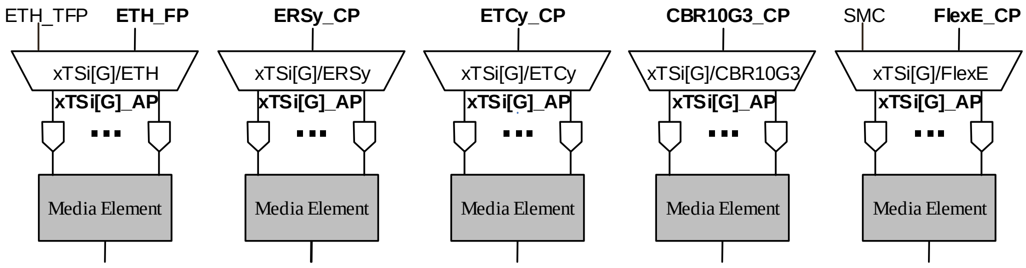
ERSy indicates a reconciliation sublayer of an Ethernet PHY, and ETCy indicates a coding sublayer of an Ethernet PHY, with ‘y’ describing the PHY using IEEE nomenclature, minus the word BASE. For example, the coding sublayer for 1000BASE-X PHY would be ETC1000X.

The label *[IEEE 802.3] processes* is used in multiple process diagrams to refer to processes performed by an adaptation function that are defined in [IEEE 802.3]. The processes referenced may encompass the entire 802.3 PHY, or only some of the sublayers of the PHY, depending on the nature of the specific adaptation function. The text that accompanies the process diagrams clarifies which portions of the PHY are being referenced.

6 xTSi[G] to client layer adaptation functions

The modelling of the Ethernet PHY relies heavily on existing description in [IEEE 802.3]. The physical lanes of the PHY are modelled as optical tributary signals (OTSi) or electrical tributary signals (ETSi) with modulators/demodulators, as is done with OTN signals in [ITU-T G.798].

This clause describes the adaptation functions for the OTSi[G] and ETSi[G] to the client. In addition to the MAC layer as a client, several different types of clients are defined based on the different types of mappings of Ethernet into OTN.



NOTE – In case of a single OTSi or ETSi, only one OTSi/ETSi modulator/demulator is applicable.

Figure 6-1 – OTSi[G] and ETSi[G] to client adaptation functions

6.1 xTSi[G] to ETH adaptation functions (xTSi[G]/ETH_A)

The xTSi[G] to ETH adaptation functions fully encapsulate a single Ethernet PHY layer with no flexE shim and adapts it to MAC layer frames. Adaptation of a flexE signal is discussed in clauses 6.5, 7 and 8. Figure 6-2 illustrates the xTSi[G] to ETH adaptation function (xTSi[G]/ETH_A). Information crossing the ETH flow point (ETH_FP) and ETH termination flow point (ETH_TFP) is referred to as ETH characteristic information (ETH_CI). Information crossing the xTSi[G] access point (xTSi[G]_AP) is referred to as xTSi[G] adapted information (xTSi[G]_AI).

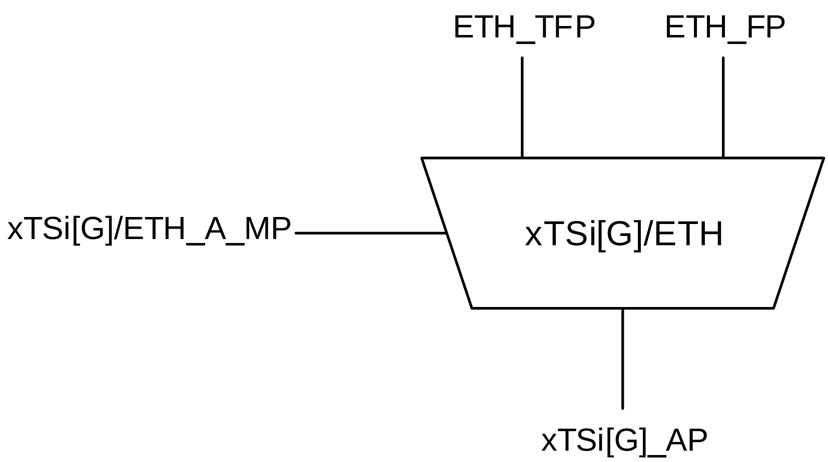


Figure 6-2 – xTSi[G] to ETH adaptation function

6.1.1 xTSi[G] to ETH adaptation source function (xTSi[G]/ETH_A_So)

Symbol

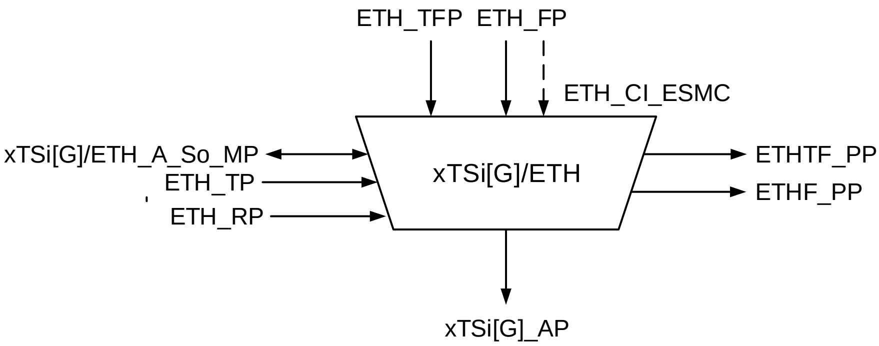


Figure 6-3 – xTSi[G]/ETH_A_So symbol

Interfaces

Table 6-1 – xTSi[G]/ETH_A_So interfaces

Inputs	Outputs
ETH_TFP: ETH_CI_D ETH_CI_P ETH_CI_DE **ETH_FP**: ETH_CI_D ETH_CI_P ETH_CI_DE ETH_CI_SSF ETH_CI_SSFrdi ETH_CI_SSFfdi ETH_CI_ESMC **ETH_TP**: ETH_TI_CK **xTSi[G]/ETH_A_RP:** xTSi[G]/ETH_A_RI_RSF **xTSi[G]/ETH_A_So_MP**: xTSi[G]/ETH_A_So_MI_[IEEE 802.3] xTSi[G]/ETH_A_So_MI_FTSEnable	**xTSi[G]_AP**: xTSi[G]_AI_D[1..N] (Note) **ETHTF_PP**: ETH_PI_D ETH_PI_P ETH_PI_DE **ETHF_PP**: ETH_PI_D ETH_PI_P ETH_PI_DE **xTSi[G]/ETH_A_RP:** xTSi[G]/ETH_A_RI_FTS **xTSi[G]/ETH_A_So_MP**: xTSi[G]/ETH_A_So_MI_[IEEE 802.3]
Note – when the PHY has only one OTSi, there is only one OTSi_AI_D signal	

Processes

A process diagram of this function is shown in Figure 6-4.

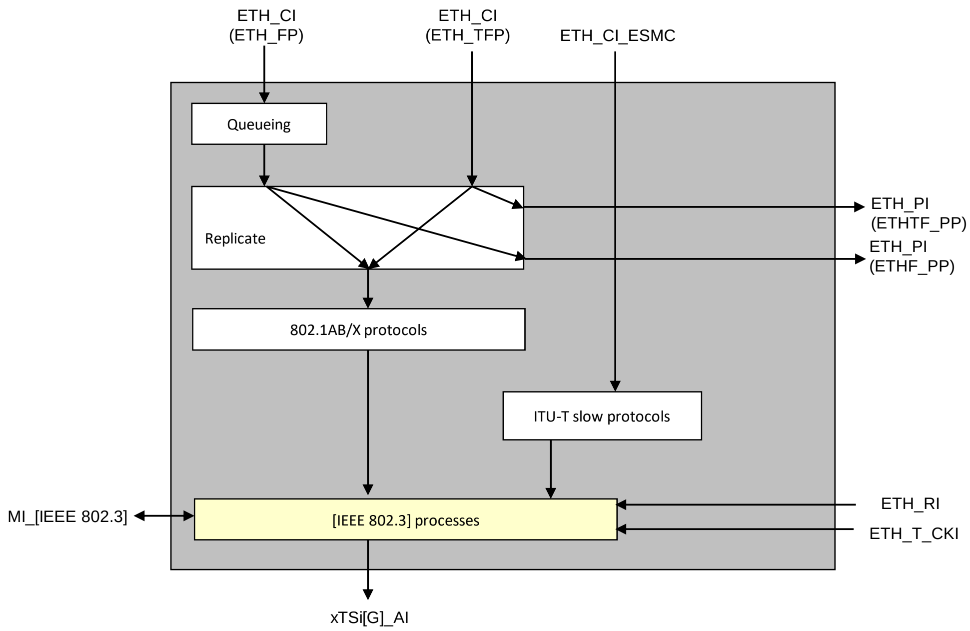


Figure 6-4 – xTSi[G]/ETH_A_So process

Processes

The queueing, replicate, and 802.1AB/X protocols processes are defined in clause 8 of [ITU-T G.8021].

ITU slow protocols

The ITU slow protocols use OUI=0x0019A7. The ITU-T slow protocol process allows for multiplexing multiple ITU defined protocols by using an ITU-T subtype.

The ITU slow protocols source process takes an incoming PDU and creates an ETH_CI traffic unit by adding the ITU-T subtype and encapsulating the resulting ITU-T slow protocol PDU within the Organization Specific Data (as defined in Annex 57B of [IEEE 802.3]):

DA=01-80-C2-00-00-02 (hex)

SA=Local MAC address

EtherType=88-09 (hex)

Slow Protocol Type=0A (hex)

OUI=0x0019A7

Organization Specific Data=ITU-T slow protocol PDU.

Supported ITU-T subtypes:

01: Ethernet synchronization message channel (ESMC) as defined in [ITU-T G.8264]. The ESMC PDUs are received from the CI_ESMC.

"Fault propagation" process

When the CI_SSF and the MI_FTSEnable (forced transmitter shutdown) are true and RI_RSF (remote signal fail) is false, this process forces the transmitter shutdown by either turning off the output transmitting device or inserting error codes (e.g., /V/, 10B_ERR for 1 GbE).

As soon as the transmitter shutdown is forced, the RI_FTS is asserted. The RI_FTS is reset after the forcing of transmitter shutdown is removed.

When the CI_SSFrdi is true and the PHY supports remote fault signalling, this process inserts the PHY-specific remote fault signal.

When the CI_SSFfdi is true and the PHY supports local fault signalling, this process inserts the PHY-specific local fault signal.

NOTE – Further details have been intentionally left out of this Recommendation.

[IEEE 802.3] Processes: The [IEEE 802.3] processes represent the entire physical layer in the IEEE 802.3 model, as well as the MAC frame counting, FCS generation, generation of 802.3 slow protocols, and multiplexing of organization-specific and 802.3 slow protocols.

NOTE – This Recommendation defines these processes by reference to [IEEE 802.3] and intentionally does not provide details, as this functionality is well understood from the IEEE work.

Defects	None.
Consequent actions	None.
Defect correlations	None.
Performance monitoring	For further study.

6.1.2 xTSi[G] to ETH adaptation sink function (xTSi[G]/ETH_A_Sk)

Symbol

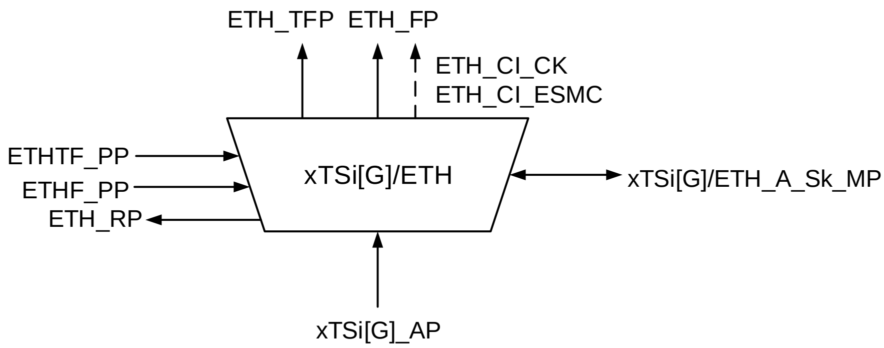


Figure 6-5 – xTSi[G]/ETH_A_Sk symbol

Interfaces

Table 6-2 – xTSi[G]/ETH_A_Sk interfaces

Inputs	Outputs
xTSi[G]_AP: xTSi[G]_AI_D[1..N] (Note) **ETHTF_PP**: ETH_PI_D ETH_PI_P ETH_PI_DE **ETHF_PP**: ETH_PI_D ETH_PI_P ETH_PI_DE **xTSi[G]/ETH_A_RP**: xTSi[G]/ETH_A_RI_FTS **xTSi[G]/ETH_A_Sk_MP**: xTSi[G]/ETH_A_Sk_MI_[IEEE 802.3] xTSi[G]/ETH_A_Sk_MI_FilterConfig	**ETH_TFP**: ETH_CI_D ETH_CI_P ETH_CI_DE **ETH_FP**: ETH_CI_D ETH_CI_P ETH_CI_DE ETH_CI_SSF ETH_CI_SSFrdi ETH_CI_SSFfdi ETH_CI_CK ETH_CI_ESMC **xTSi[G]/ETH_A_RP**: xTSi[G]/ETH_A_RI_RSF **xTSi[G]/ETH_A_Sk_MP**: xTSi[G]/ETH_A_Sk_MI_[IEEE 802.3]
Note – when the PHY has only one OTSi, there is only one OTSi_AI_D signal	

Processes

A process diagram of this function is shown in Figure 6-6.

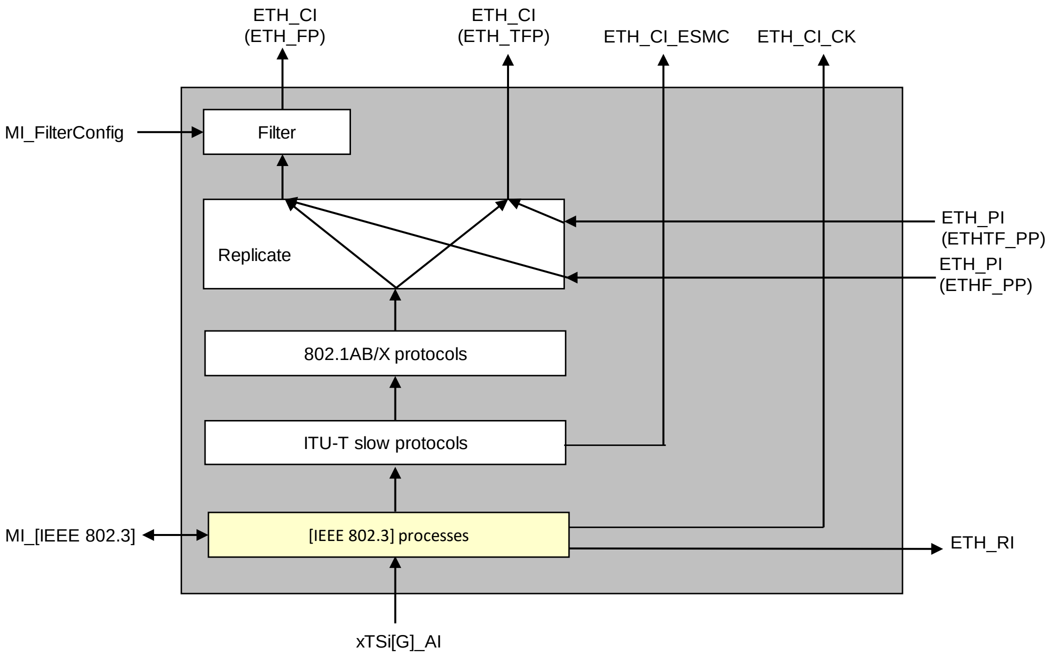


Figure 6-6 – xTSi[G]/ETH_A_Sk process

The filter, replicate, and 802.1AB/X protocols processes are defined in clause 8 of [ITU-T G.8021].

[IEEE 802.3] Processes: The [IEEE 802.3] processes represent the entire physical layer in the IEEE 802.3 model, plus the MAC length check, MAC frame check, MAC frame counting, slow protocols demultiplexing, and 802.3 slow protocols processing. Pause frames are always discarded.

NOTE – This Recommendation defines these processes by reference to [IEEE 802.3] and intentionally does not provide details, as this functionality is well understood from the IEEE work.

ITU slow protocols

The ITU slow protocols use OUI=0x0019A7. The ITU-T slow protocol process allows for multiplexing multiple ITU defined protocols by using an ITU-T subtype.

ITU slow protocols sink process separates the ETH_CI traffic units carrying ITU slow protocol PDUs with supported ITU-T subtypes from the rest of the ETH_CI traffic units. The former are passed to the corresponding processing function; the latter are passed through.

For the PDUs with supported ITU-T subtypes, this process extracts the ITU-T slow protocol PDU from the Organization Specific Data field of the received ETH_CI taffic units and removes the ITU-T subtype from it. The resulting PDU is forwarded to the protocol process identified by the removed ITU-T subtype.

Supported ITU-T subtypes:

01: Ethernet synchronization message channel (ESMC) as defined in [ITU-T G.8264].

Defects

The definition of defects used by xTSi[G]/ETH_A_Sk is outside the scope of this Recommendation.

Consequent actions

When an incoming signal failure is detected as specified in [IEEE 802.3], and the RI_FTS is false, the aSSF consequent action is triggered.

NOTE 1 – aRSF is generated and communicated to the paired xTSi[G]/ETH_A_So (RI_RSF) to prevent a forced transmitter shutdown in case of an incoming signal failure. This Recommendation does not specify the remote fault indication signalling.

When a PHY-specific remote fault signalling (as defined in [IEEE 802.3]) is detected, and the PHY supports remote fault signalling, the aSSFrdi consequent action is triggered.

When a PHY-specific local fault signalling (as defined in [IEEE 802.3]), and the PHY supports local fault signalling, the aSSFfdi consequent action is triggered.

NOTE 2 – Further details are intentionally left outside the scope of this Recommendation.

Defect correlations None.

Performance monitoring For further study.

6.2 OTSi to 10G Ethernet Reconciliation Sublayer adaptation functions (OTSi/ERS10G_A)

The OTSi to ERS10G adaptation functions perform the adaptation between the optical signal and the characteristic information of an ERS10G signal. They are used only in conjunction with the ODU2P/ERS10G_A functions defined in clause 14.3.3 of [ITU-T G.798].

ERS10G Characteristic Information

The ERS10G_CI is a stream of ERS10G_CI_D traffic units, complemented with the ERS10G_CI_SSF signal. The ERS10G_CI_D traffic units carry either an Ethernet data frame, including the preamble, or an ordered set, as defined in clause 7.9 of [ITU-T G.7041]. The ordered sets may carry local fault or remote fault indications. The ERS10G_CP reference point is located within the reconciliation sublayer (see Figure 44-1 of [IEEE 802.3]).

NOTE – There is no Ethernet MAC process in these adaptation functions. Consequently, since no error checking is performed on the Ethernet MAC frames, errored MAC frames are forwarded in both ingress and egress directions.

6.2.1 OTSi to 10G Ethernet Reconciliation Sublayer source adaptation function (OTSi/ERS10G_A_So)

Symbol

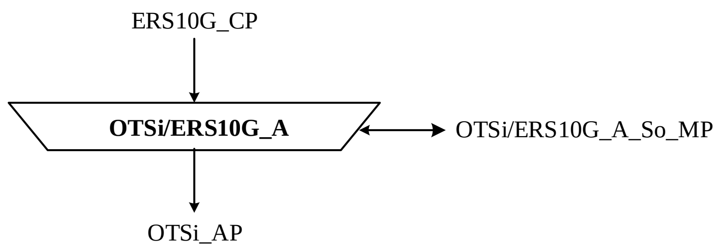


Figure 6-7 – OTSi/ERS10G_A_So function

Interfaces

Table 6-3 – OTSi/ERS10G_A_So inputs and outputs

Input(s)	Output(s)
ERS10G_CP: ERS10G_CI_D **OTSi/ERS10G_A_So_MP**: OTSi/ERS10G_A_So_MI_[IEEE 802.3]	**OTSi_AP**: OTSi_AI_D **OTSi/ERS10G_A_So_MP**: OTSi/ERS10G_A_So_MI_[IEEE 802.3]

Processes

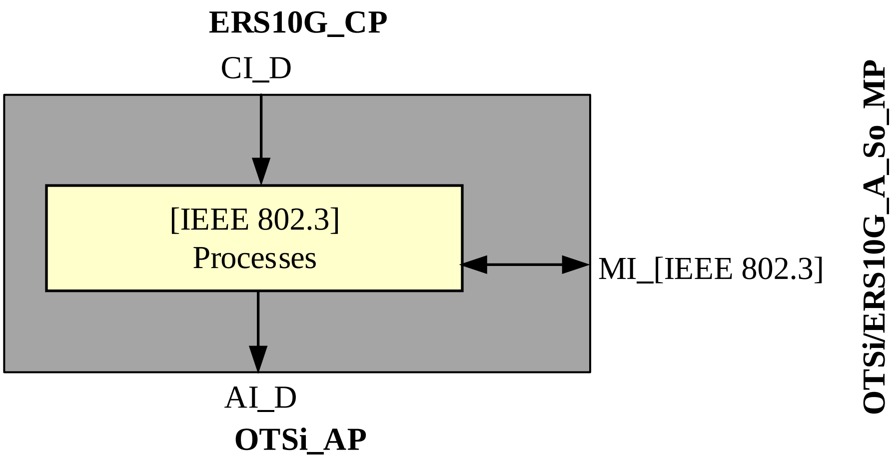


Figure 6-8 – xTSi/ERS10G_A_So processes

[IEEE 802.3] Processes: The [IEEE 802.3] processes represent the whole functionality of the physical layer in [IEEE 802.3] model below the ERS10G_CP reference point.

NOTE – This Recommendation defines these processes by reference to [IEEE 802.3] and intentionally does not provide details, as this functionality is well understood from the IEEE work.

Defects: None.

Consequent actions: None.

Defect correlations: None.

Performance monitoring: None.

6.2.2 OTSi to 10G Ethernet Reconciliation Sublayer sink adaptation function (OTSi/ERS10G_A_Sk)

Symbol

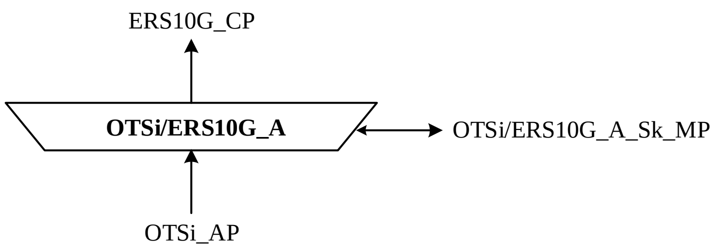


Figure 6-9 – OTSi/ERS10G_A_Sk function

Interfaces

Table 6-4 – OTSi/ERS10G_A_Sk inputs and outputs

Input(s)	Output(s)
OTSi_AP: OTSi_AI_D **OTSi/ERS10G_A_So_MP**: OTSi/ERS10G_A_Sk_MI_[IEEE 802.3]	**ERS10G_CP**: ERS10G_CI_D ERS10G_CI_SSF **OTSi/ERS10G_A_So_MP**: OTSi/ERS10G_A_Sk_MI_[IEEE 802.3]

Processes

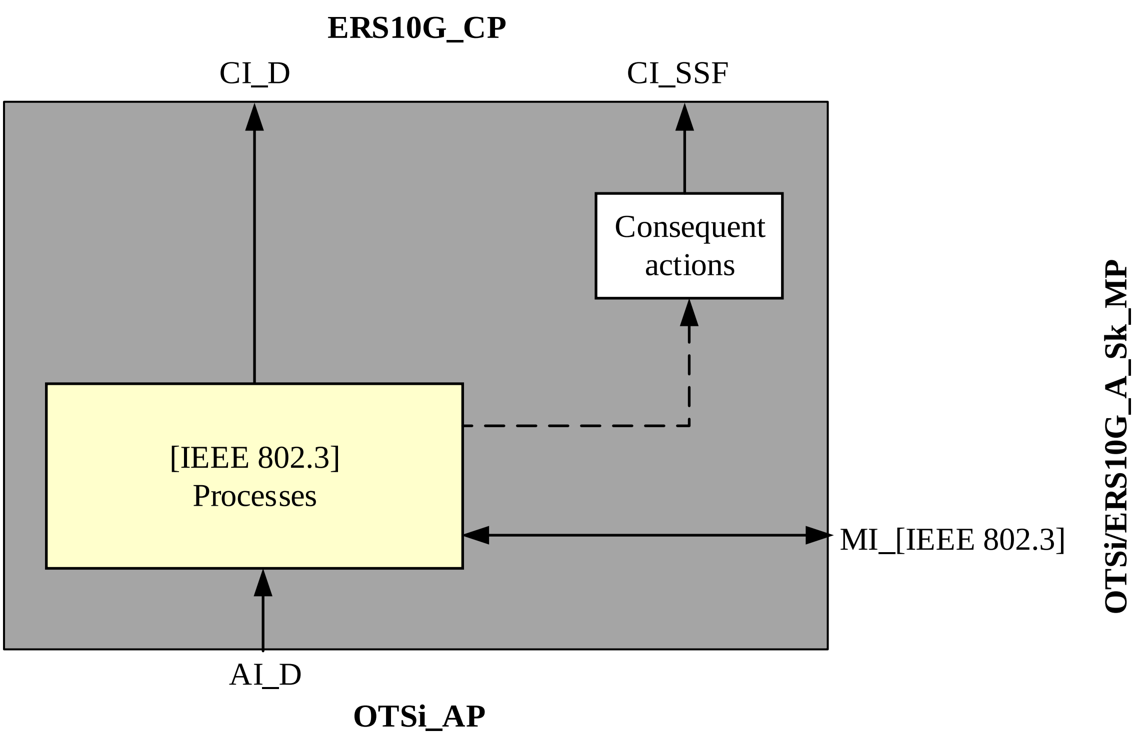


Figure 6-10 – OTSi/ERS10G_A_Sk processes

[IEEE 802.3] Processes: The [IEEE 802.3] processes represent the whole functionality of the physical layer in [IEEE 802.3] model below the ERS10G_CP reference point.

NOTE – This Recommendation defines these processes by reference to [IEEE 802.3] and intentionally does not provide details, as this functionality is well understood from the IEEE work.

Defects

The definition of the defects used by the OTSi/ERS10G_A_Sk is outside the scope of this Recommendation.

Consequent actions

When an incoming signal failure is detected, as specified in [IEEE 802.3], the aSSF consequent action is triggered.

NOTE 1 – Further details are intentionally left outside the scope of this Recommendation.

NOTE 2 – The replacement signal is generated in the subsequent adaptation source function ODU2P/ETHPP-OS_A_So defined in 14.3.3 of [ITU-T G.798].

Defect correlations: None.

Performance monitoring: None.

6.3 xTSi[G] to Ethernet Coding adaptation functions (xTSi[G]/ETCy_A)

The set of xTSi[G] to ETCy (y=1000X, 25GR, 40GR, 100GR, 200GR, 400GR) adaptation functions adapt between a group of one or more optical signals (depending on the client) and the characteristic information of the ETCy layer. These functions are used in conjunction with functions that provide codeword-transparent mapping into a non-Ethernet server layer.

The following specific functions are defined:

OTSi/ETC1000X_A

xTSi/ETC25GR_A

xTSi[G]/ETC40GR_A

xTSiG/ETC100GR_A

OTSiG/ETC200GR_A

OTSiG/ETC400GR_A

<u>ETCy Characteristic Information</u>

The ETCy_CI is a stream of ETCy_CI_D traffic units, complemented with the ETCy_CI_CK and the ETCy_CI_SSF signals. For the case of ETC200GR and ETC400GR, the CI is further complemented with ETCy_CI_AM_SF and ETCy_CI_FEC_DEG signals, corresponding to rx_am_sf<2:0> and tx_am_sf<2:0>, and FEC_degraded_SER, respectively, in [IEEE 802.3bs].

The ETC1000X_CI_D traffic units carry 8B/10B codewords of the 1000BASE-X PCS, as defined in clause 36 of [IEEE 802.3].

The ETC25GR_CI_D traffic units carry the 64b/66b codewords of the 25GBASE-R PCS, as defined in clauses 49 and clause 107 of [IEEE 802.3].

The ETC40GR_CI_D traffic units and ETC100GR_CI_D traffic units carry 64b/66b codewords of the 40GBASE-R or 100GBASE-R PCS, respectively, as defined in clause 82 of [IEEE 802.3].

The ETC200GR_CI_D and ETC400GR_CI_D traffic units carry 64b/66b codewords of the 200GBASE-R and 400GBASE-R PCS, respectively, as defined in clause 119 of [IEEE 802.3bs].

For 1000BASE-X interfaces, the ETC1000X_CP reference point is the interface between the PCS and PMA sublayers as shown in Figure 34-1 of [IEEE 802.3].

For 25GBASE-R interfaces, the ETC25GR_CP reference point is the interface between the PCS and PMA (or FEC) sublayers, as shown in Figure 107-2 of [IEEE 802.3by].

For 40GBASE-R interfaces, the ETC40GR_CP reference point is within the 40GBASE-R PCS, between the encode/decode and scramble/descramble processes; see figure 82-2 of [IEEE 802.3].

For 100GBASE-R interfaces, the ETC100GR_CP reference point is within the 100GBASE-R PCS, between the encode/decode and scramble/descramble processes; see figure 82-2 of [IEEE 802.3].

For 200GBASE-R interfaces, the ETC200GR_CP reference point is within the 200GBASE-R PCS, between the encode/decode and rate match process and the transcode/reverse transcode process, including the FEC_degraded_SER and rx_local_degraded bits; see figure 119-2 and clause 19.2.4.1 of [IEEE 802.3bs].

For 400GASE-R interfaces, the ETC400GR_CP reference point is within the 400GBASE-R PCS, between the encode/decode and rate match process and the transcode/reverse transcode process, including the FEC_degraded_SER and rx_local_degraded bits; see figure 119-2 of and clause 19.2.4.1 [IEEE 802.3bs].

6.3.1 xTSi[G] to Ethernet Coding adaptation source function (xTSi[G]/ETCy_A_So)

Symbol

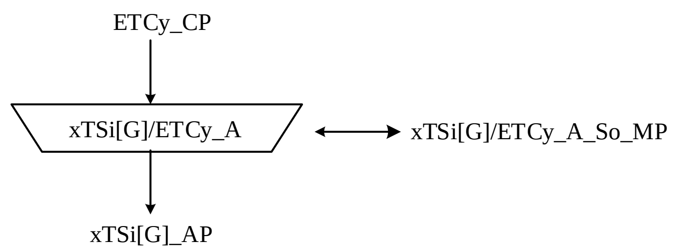


Figure 6-11 – xTSi[G]/ETCy_A_So function

Interfaces

Table 6-5 – xTSi[G]/ETCy_A_So inputs and outputs

Input(s)	Output(s)
ETCy_CP: ETCy_CI_CK ETCy_CI_D For y=200GR or 400GR: ETCy_CI_AM_SF[2:0] **xTSi[G]/ETCy_A_So_MP**: xTSi[G]/ETCy_A_So_MI_[IEEE 802.3]	**xTSi[G]_AP**: xTSi_AI_D[1..N] (Note) **xTSi[G]/ETCy_A_So_MP**: xTSi[G]/ETCy_A_So_MI_[IEEE 802.3]
Note – when the PHY has only one OTSi, there is only one OTSi_AI_D signal	

Processes

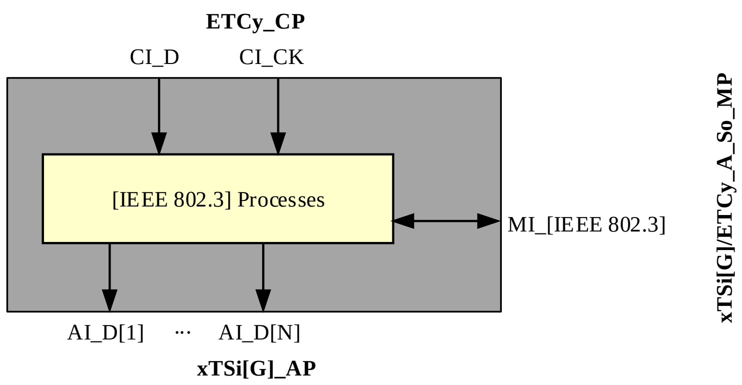


Figure 6-12 – xTSi[G]/ETCy_A_So processes

[IEEE 802.3] Processes: The [IEEE 802.3] processes represent the whole functionality of the physical layer in IEEE 802.3 model below the ETCy_CP reference point.

NOTE – This Recommendation defines these processes by reference to [IEEE 802.3] and intentionally does not provide details, as this functionality is well understood from the IEEE work.

Defects: None.

Consequent actions: None.

Defect correlations: None.

Performance monitoring: None.

6.3.2 xTSi[G] to Ethernet Coding adaptation sink function (xTSi[G]/ETCy_A_Sk)

Symbol

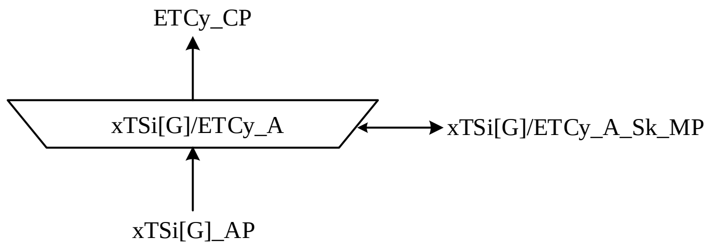


Figure 6-13 – xTSi[G]/ETCy_A_Sk function

Interfaces

Table 6-6 – xTSi[G]/ETCy_A_Sk inputs and outputs

Input(s)	Output(s)
xTSi[G]_AP: xTSi_AI_D[1..N] (Note) **xTSi[G]/ETCy_A_So_MP**: xTSi[G]/ETCy_A_Sk_MI_[IEEE 802.3]	**ETCy_CP**: ETCy_CI_CK ETCy_CI_D ETCy_CI_SSF For y=200GR or 400GR: ETCy_CI_AM_SF[2:0] ETCy_CI_FEC_DEG **xTSi[G]/ETCy_A_So_MP**: xTSi[G]/ETCy_A_Sk_MI_[IEEE 802.3]
Note – when the PHY has only one OTSi, there is only one OTSi_AI_D signal	

Processes

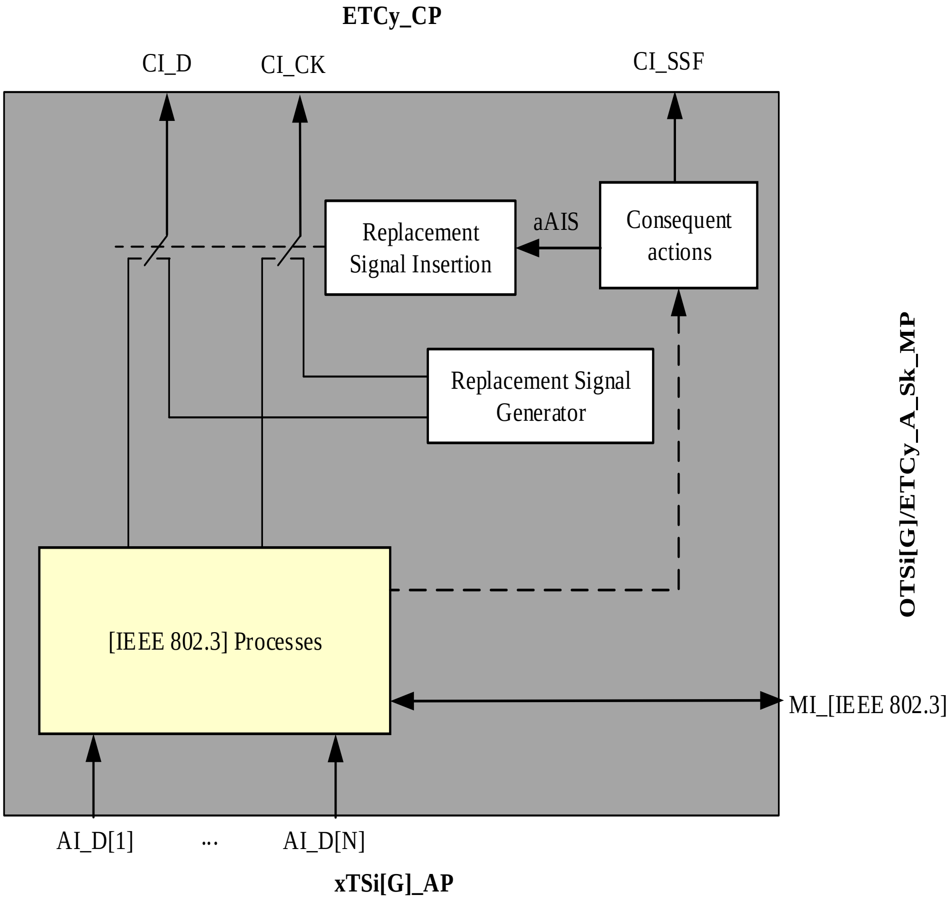


Figure 6-14 – xTSi[G]/ETCy_A_Sk processes

[IEEE 802.3] Processes: The [IEEE 802.3] processes represent the whole functionality of the physical layer in IEEE 802.3 model below the ETCy_CP reference point.

NOTE – This Recommendation defines these processes by reference to [IEEE 802.3] and intentionally does not provide details, as this functionality is well understood from the IEEE work.

Defects

The definition of the defects used by the xTSi[G]/ETCy_A_Sk is outside the scope of this Recommendation.

Consequent actions

When an incoming signal failure is detected, as specified in [IEEE 802.3], aSSF and aAIS consequent actions are triggered.

NOTE – Further details are intentionally left outside the scope of this Recommendation.

On declaration of aAIS, the function shall output a replacement signal.

For 1000BASE-X, the replacement signal is defined in clause 17.7.1.1 of [ITU-T G.709].

For 25GBASE-R, the replacement signal is defined in clause 17.13 of [ITU-T G.709].

For 40GBASE-R clients, the replacement signal is defined in clause 17.7.4 of [ITU-T G.709].

For 100GBASE-R clients, the replacement signal is defined in clause 17.7.5 of [ITU-T G.709].

For 200GBASE-R clients, the replacement signal is defined in clause 17.13 of [ITU-T G.709].

For 400GBASE-R clients, the replacement signal is defined in clause 17.13 of [ITU-T G.709].

Defect correlations: None.

Performance monitoring: None.

6.4 OTSi to CBR10G3 adaptation functions (OTSi/CBR10G3_A)

The OTSi to CBR10G3 adaptation functions perform the adaptation between the optical signal and the characteristic information of a CBR10G3 signal. It is used in conjunction with the ODU2eP/CBR10G3_A function as specified in clause 14.3.1 of [ITU-T G.798].

<u>CBR10G3 Characteristic Information</u>

The CBR10G3_CP reference point is located within the PMA sublayer as shown in figure 44-1 of [IEEE 802.3], at the point where conversion between the serial bit stream and the 16-bit data occurs.

6.4.1 OTSi to CBR10G3 adaptation source function (OTSi/CBR10G3_A_So)

Symbol

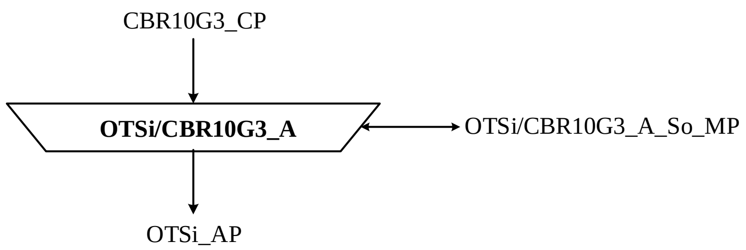


Figure 6-15 – OTSi/CBR10G3_A_So function

Interfaces

Table 6-7 – OTSi/CBR10G3_A_So inputs and outputs

Input(s)	Output(s)
CBR10G3_CP: CBR10G3_CI_CK CBR10G3_CI_D **OTSi/CBR10G3_A_So_MP:** OTSi.CBR10G3_A_So_MI_[IEEE 802.3]	**OTSi_AP**: OTSi_AI_D **OTSi/CBR10G3_A_So_MP:** OTSi.CBR10G3_A_So_MI_[IEEE 802.3]

Processes

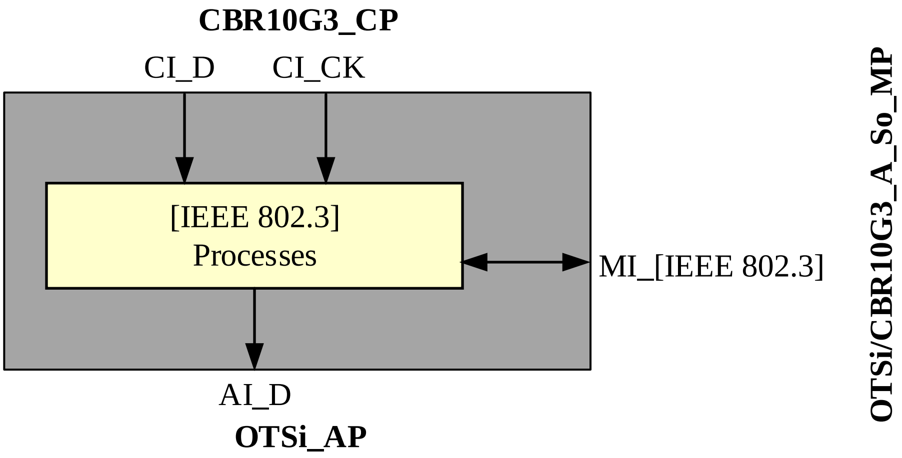


Figure 6-16 – xTSi(G)/CBR10G3_A_So processes

[IEEE 802.3] Processes: The [IEEE 802.3] processes represent the whole functionality of the physical layer in IEEE 802.3 model below the CBR10G3_CP reference point.

NOTE – This Recommendation defines these processes by reference to [IEEE 802.3] and intentionally does not provide details on it, as this functionality is well understood from the IEEE work.

Defects: None.

Consequent actions: None.

Defect correlations: None.

Performance monitoring: None.

6.4.2 OTSi to CBR10G3 adaptation sink function (OTSi/CBR10G3_A_Sk)

Symbol

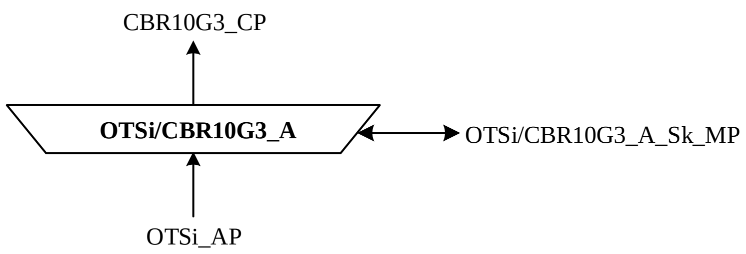


Figure 6-17 – OTSi/CBR10G3_A_Sk function

Interfaces

Table 6-8 – OTSi/CBR10G3_A_Sk inputs and outputs

Input(s)	Output(s)
OTSi_AP: OTSi_AI_D **OTSi/CBR10G3_A_Sk_MP:** OTSi.CBR10G3_A_Sk_MI_[IEEE 802.3]	**CBR10G3_CP**: CBR10G3_CI_CK CBR10G3_CI_D CBR10G3_CI_SSF **OTSi/CBR10G3_A_Sk_MP:** OTSi.CBR10G3_A_Sk_MI_[IEEE 802.3]

Processes

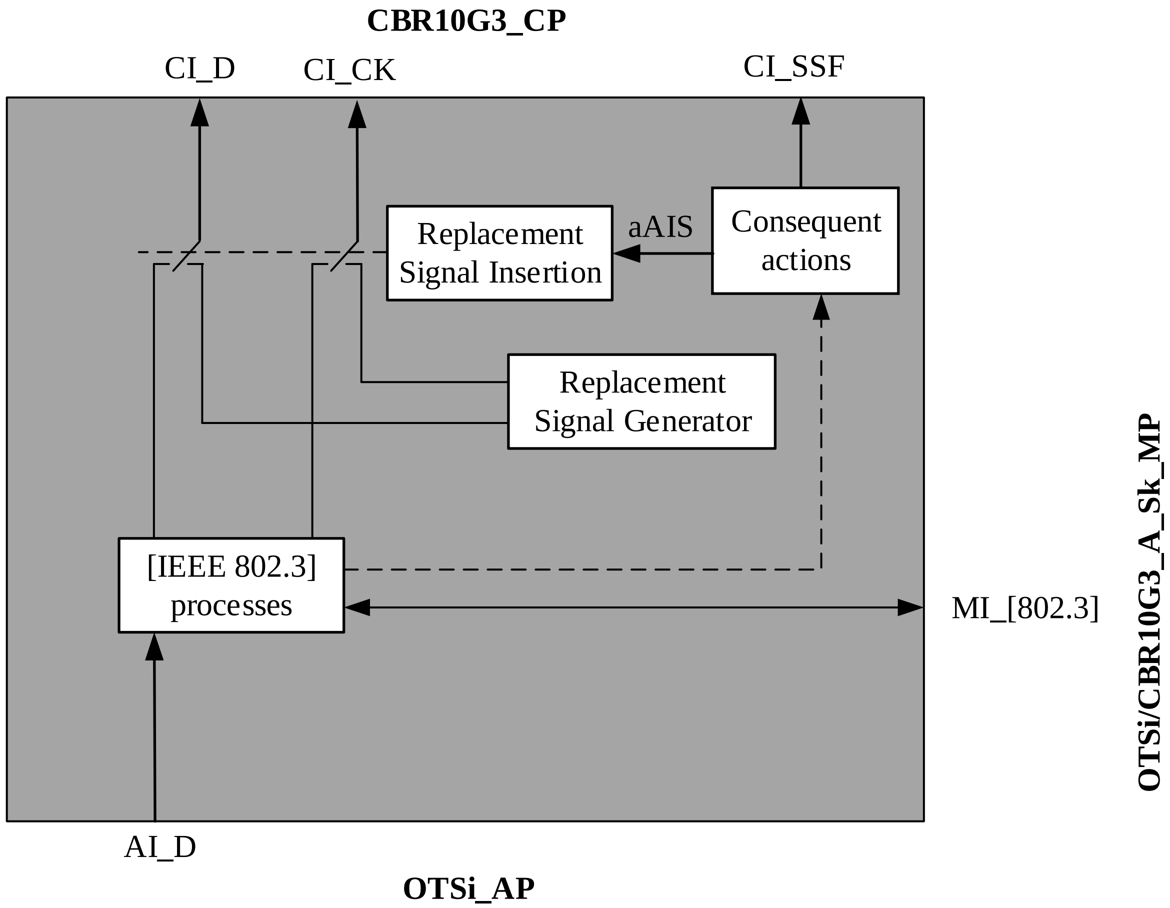


Figure 6-18 – OTSi/CBR10G3_A_Sk processes

[IEEE 802.3] Processes: The IEEE processes represent the whole functionality of the physical layer in IEEE 802.3 below the CBR10G3_CP reference point.

NOTE – This Recommendation defines these processes by reference to [802.3] and intentionally does not provide details, as this functionality is well understood from the IEEE work.

Defects

The definition of the defects used by the OTSi/CBR10G3_A_Sk is outside the scope of this Recommendation.

Consequent actions

When an incoming signal failure is detected, as specified in [IEEE 802.3], aSSF and aAIS consequent actions are triggered.

NOTE – Further details are intentionally left outside the scope of this Recommendation.

On declaration of aAIS, the function shall output a replacement signal, as defined in clause 17.2 of [ITU-T G.709] and clause 14.3.1.3 of [ITU-T G.798].

Defect correlations: None.

Performance monitoring: None.

6.5 OTSi[G] to Flex Ethernet (FlexE) adaptation functions (OTSi[G]/FlexE_A)

The OTSiG to FlexE adaptation functions perform the adaptation between the group of optical signals related with a PHY that is part of a flexE group and the characteristic information of the FlexE layer signal.

The FlexE characteristic information is defined in clause 7.

6.5.1 OTSiG to FlexE adaptation source function (OTSiG/FlexE_A_So)

The information flow and processing of the OTSiG/FlexE_A_So function is defined with reference to Figures 6-19 and 6-20.

Symbol

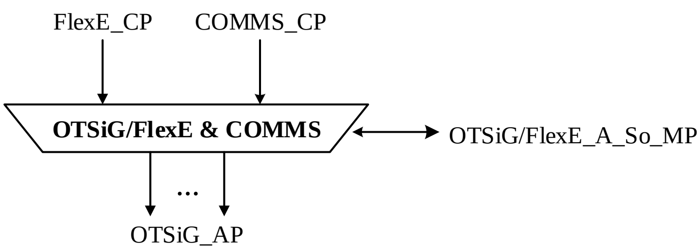


Figure 6-19 OTSiG/FlexE_A_So function

Interfaces

Table 6-9 – OTSiG/FlexE_A_So inputs and outputs

Input(s)	Output(s)
FlexE_CP: FlexE_CI_CK FlexE_CI_D FlexE_CI_FS FlexE_CI_MFS **COMMS_CP**: COMMS_CI_D	**OTSiG_AP**: OTSiG_AI_D[1..N] **COMMS_CP**: COMMS_CI_CK **OTSiG/FlexE_A_So_MP**: OTSiG/FlexE_A_So_MI_[IEEE 802.3]

Processes

The processes associated with the OTSiG/FlexE_A_So function are as depicted in Figure 6-20.

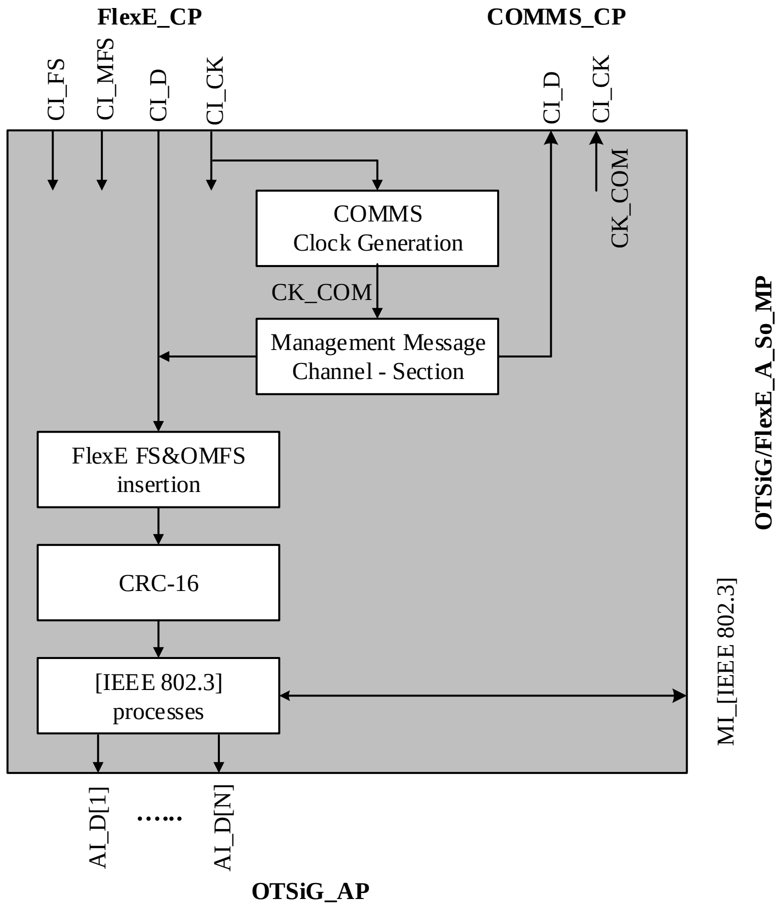


Figure 6-20 – OTSiG/FlexE_A_So processes

COMMS clock generation: The function shall generate the COMMS clock (CK_COM) by dividing the clock (CI_CK) by a factor of 81844.

Management Channel – section: The Management Channel is optional. When used, the incoming COMMS_CI_D data is inserted into the Management Channel – section field as described in clause 7.3.5 of [OIF FLEXE IA]. When it is not used, Ethernet idle control blocks are inserted as described in clause 7.3.5 of [OIF FLEXE IA].

FlexE FS&MFS insertion: The function shall insert the FlexE frame alignment signal (FS) and FlexE multi-frame alignment signal (MFS) as described in clause 7.3.1 of [OIF FLEXE IA]. The FlexE frame alignment signal is defined by the pattern "0x4B" in bits 0 to 7 and the pattern 0x5 in bits 32 to 35.

CRC-16: See clause 7.3.9 of [OIF FLEXE IA]. The function shall compute the CRC-16 and insert the calculated CRC-16 value into the CRC-16 byte of FlexE overhead field.

[IEEE 802.3] processes: The [IEEE 802.3] processes represent the whole functionality of the PCS sublayer below the flex-E shim, as well as the PMA and PMD sublayers in the IEEE 802.3 model.

NOTE – This Recommendation defines these processes by reference to [IEEE 802.3] and intentionally does not provide details, as this functionality is well understood from the IEEE work.

Defects: None.

Consequent actions: None.

Defect correlations: None.

Performance monitoring: None.

6.5.2 OTSiG to FlexE adaptation sink function (OTSiG/FlexE_A_Sk)

The information flow and processing of the OTSiG/FlexE_A_Sk function is defined with reference to Figures 6-21 and 6-22.

Symbol

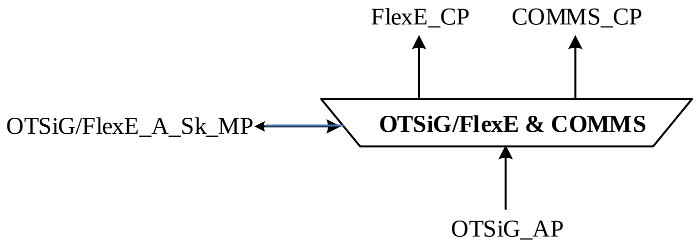


Figure 6-21 – OTSiG/FlexE_A_Sk function

Interfaces

Table 6-10 – OTSiG/FlexE_A_Sk inputs and outputs

Input(s)	Output(s)
OTSiG_AP: OTSiG_AI_D[1..N] **OTSiG/FlexE_A_Sk_MP**: OTSiG/FlexE_A_Sk_MI_1second	**FlexE_CP**: FlexE_CI_CK FlexE_CI_D FlexE_CI_FS FlexE_CI_MFS FlexE_CI_CRCerr FlexE_CI_SSF **COMMS_CP**: COMMS_CI_D COMMS_CI_CK COMMS_CI_SSF **OTSiG/FlexE_A_Sk_MP**: OTSiG/FlexE_A_Sk_MI_[IEEE 802.3] OTSiG/FlexE_A_Sk_MI_cLOF OTSiG/FlexE_A_Sk_MI_cLOM

Processes

The processes associated with the OTSiG/FlexE_A_Sk function are as depicted in Figure 6-22.

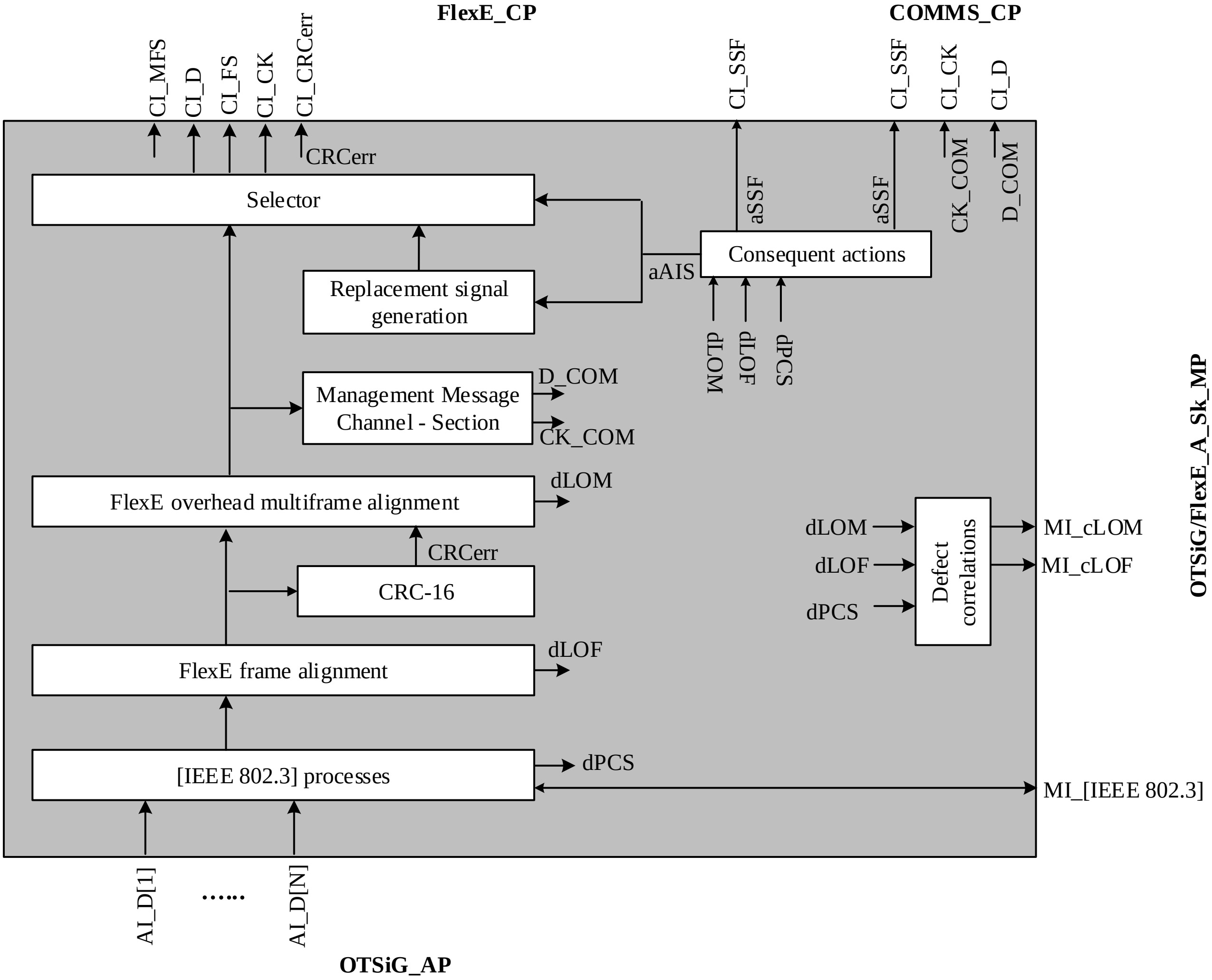


Figure 6-22 – OTSiG/FlexE_A_Sk processes

[IEEE 802.3 processes]: The [IEEE 802.3] processes represent the whole functionality of the PCS sublayer below the flex-E shim, as well as the PMA and PMD sublayers in the IEEE 802.3 model.

NOTE – This Recommendation defines these processes by reference to [IEEE 802.3] and intentionally does not provide details, as this functionality is well understood from the IEEE work.

FlexE Frame alignment: The function shall recover the FlexE frame start through recognizing the FlexE block 1 of the FlexE overhead frame as described in clause 7.3.1 of [OIF FLEXE IA]. The FlexE block 1 of the FlexE overhead frame is encoded as a special ordered set (the sync header is 10, the control block type is 0x4B (ordered set), and the "O" code is 0x5) and present once per (1023 × 20 + 1) × 8 blocks. The process has two states, out-of-frame (OOF) and in-frame (IF). The frame alignment start shall be maintained during the OOF state. In the OOF state, the frame alignment shall be assumed to be recovered and the IM state shall be entered, when a valid FlexE block 1 is found in two consecutive FlexO overhead frames. In the IF state, OOM shall be entered when FlexE block 1 of the FlexE overhead frame is lost if the sync header, control block type, or O code do not match at the expected position for 5 occurrences.

CRC-16: See clause 7.3.9 of [OIF FLEXE IA]. The CRC-16 is extracted from the CRC-16 field. If the extracted CRC-16 value can be divisibility by the expected polynomial, CRCerr is set to 0 (the default value); otherwise, CRCerr is set to 1.

FlexE overhead multi-frame alignment: The function shall recover the FlexE overhead multi frame start of 32 frames through recognizing the OMF overhead with a good CRC (CRCerr = 0). The OMF (overhead multiframe) bit has a value of "0" for the first sixteen overhead frames of the overhead multiframe, and a value of "1" for the last sixteen overhead frames of the overhead multiframe, as described in clause 7.3.1 of [OIF FLEXE IA]. The specific process is described in annex B.2.1.2 of G.798.

Management Channel: The Management Channel is for user specific and the corresponding COMMS data (D_COM) and COMMS clock (CK_COM) is extracted from the Management Channel – section field described in clause 7.3.5 of [OIF FLEXE IA].

Replacement signal generation: When aAIS is true, the function shall provide for a FlexE replacement signal and clock generation process that generates a stream of local fault sequence ordered sets as specified in clause 5.2.2.3 of [OIF FLEXE IA].

Selector: The function shall select the normal FlexE signal or the replacement signal. When aAIS is true, it shall select the replacement signal. Otherwise, it shall select the normal FlexE signal.

Defects

The function shall detect dPCS, dLOF and dLOM.

dPCS: This defect represents any defect detected by the [IEEE 802.3] processes in the portion of the PCS below the flex-E shim, or in the PMA or PMD sublayers.

dLOF: The loss of FlexE frame defect dLOF is generated based on the state of the FlexE Frame alignment process. See dLOF described in annex B.1.1.1.1 of G.798.

dLOM: The loss of FlexE overhead multiframe defect dLOM is generated based on the state of the FlexE overhead multiframe alignment process. See dLOM described in annex B.1.1.1.2 of G.798.

Consequent actions

aSSF ← dLOM or dLOF or dPCS

aAIS ← dLOM or dLOF or dPCS

On declaration of aAIS, the function shall output an FlexE replacement signal, which consists of a continuous stream of Local Fault ordered sets at a bit rate of $103.125\ \text{Gb/s} \times \frac{16383}{16384} \pm 100\text{ppm}$, within two FlexE frames. On clearing aAIS, the FlexE replacement signal shall be removed within two FlexE frames, with normal data being output. The FlexE replacement signal clock, frame start and multiframe start shall be independent from the incoming clock, frame start and multiframe start. The FlexE replacement signal clock has to be within the frequency range $103.125\ \text{Gb/s} \times \frac{16383}{16384} \pm 100\text{ppm}$.

Defect correlations

cLOF ← dLOF and (not dPCS)

cLOM ← dLOM and (not dLOF) and (not dPCS)

Performance monitoring

Performance monitoring is part of the [IEEE 802.3] processes. There is no additional performance monitoring for the flexE-related processes.

7 Flex Ethernet (FlexE) functions

Figure 7-1 illustrates the FlexE and FlexE client adaptation functions. While these are modelled as multiple layer networks, the intent of FlexE technology is to provide a mechanism for bonding PCS lanes from one or more PHYs into a larger PHY, which is traditionally an adaptation function. The FlexE model is not intended to imply that connection functions exist for the FlexE or FlexEC information.

The information crossing the FlexE connection point (FlexE_CP) is referred to as the FlexE characteristic information (FlexE_CI). The information crossing the FlexE access point (FlexE_AP) is referred to as the FlexE adapted information (FlexE_AI). The information crossing the FlexEC connection point (FlexEC_CP) is referred to as the FlexEC characteristic information (FlexEC_CI).

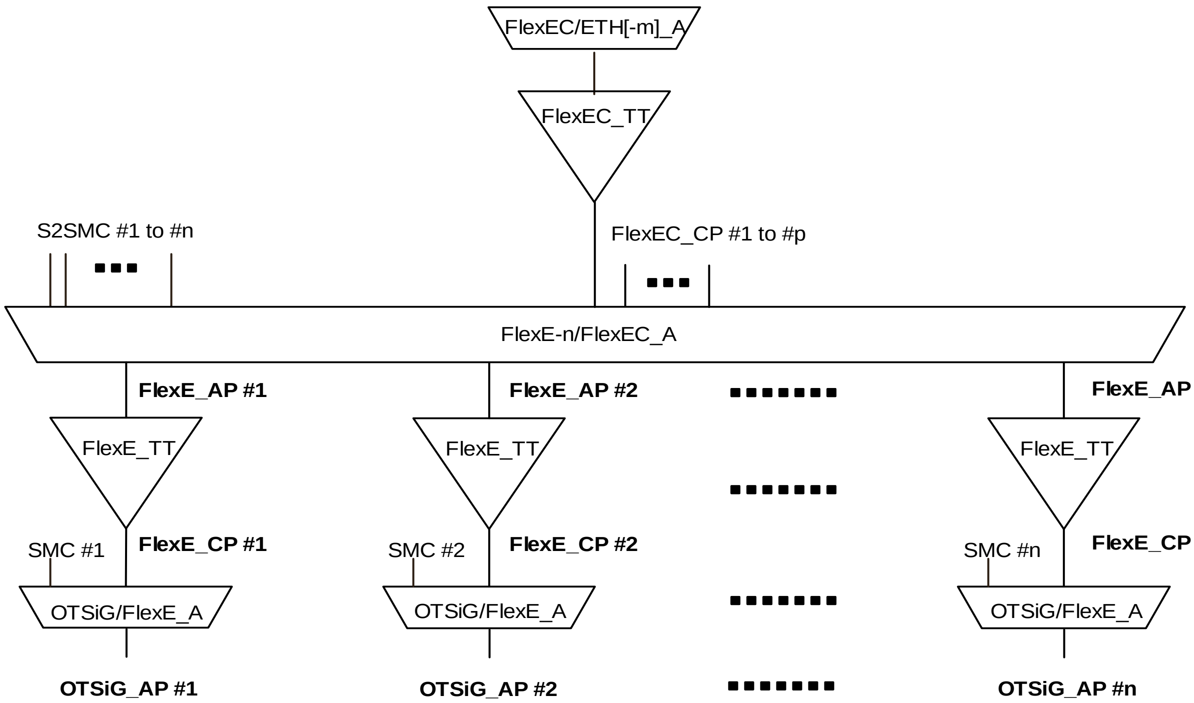


Figure 7-1 – FlexE and FlexE client adaptation functions

The FlexE characteristic information (FlexE_CI) is a stream of FlexE_CI traffic units complemented with FlexE_CI_CK, FlexE_CI_FS, FlexE_CI_MFS, FlexE_CI_CRCerr, and FlexE_CI_SSF signals. The FlexE_CI traffic unit defines the FlexE_CI_D signal as the FlexE frame defined in Figure 7-2, with FlexE overhead per figure 11 of [OIF FLEXE IA] except for the section management channel, CRC-16, frame alignment, and multiframe alignment fields. The shim-to-shim management overhead is optional. If it is not used, it is filled with Ethernet Idle control blocks, per clause 7.3.5 of [OIF FLEXE IA]. The RES overhead is set to all-ZEROs.

NOTE – the section management overhead, CRC-16, frame alignment, and multiframe alignment overhead fields of the FlexE overhead are processed in the OTSiG/FlexE_A function; see clause 6.5.

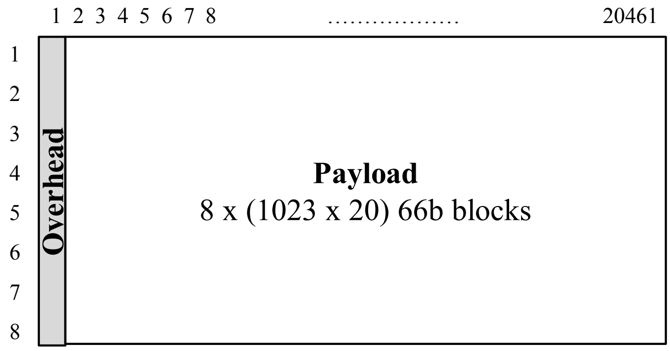


Figure 7-2 – FlexE frame

The FlexE adapted information (FlexE_AI) is a stream of FlexE_AI traffic units complemented with FlexE_AI_CK, FlexE_AI_FS, FlexE_AI_MFS, FlexE_AI_CRCerr and FlexE_AI_TSF signals. The FlexE_AI traffic unit defines the FlexE_AI_D signal, which consists of the FlexE_CI without the RPF overhead. The mapping-specific overhead depends on the client mapping scheme. In case of COMMS access at the FlexE_AP, it also includes the FlexE Shim-to-Shim Management Channel overhead (S2SMC).

The FlexEC characteristic information (FlexEC_CI) is a stream of FlexEC_CI traffic units complemented with FlexEC_CI_CK and FlexEC_CI_SSF signals. The FlexEC_CI traffic unit defines the FlexEC_CI_D signal as the FlexEC 66b block stream with the bit rates of 10, 40 and n x 25 Gbit/s ($n \geq 1$) defined in Table 7-1.

Table 7-1 – FlexEC clients

FlexEC bit rate	Bit rate	Clock tolerance	Encoding	AM
10G	10 312 500 (kbit/s)	± 100 ppm	Clause 82 64b/66b	No
40G	41 250 000 (kbit/s)	± 100 ppm	Clause 82 64b/66b	No
n × 25G	n × 25 781 250 (kbit/s)	± 100 ppm	Clause 82 64b/66b	No

7.1 FlexE trail termination function (FlexE_TT)

The FlexE_TT function terminates the section monitoring overhead of the FlexE overhead to determine the status of the FlexE trail. Figure 7-3 shows the combination of the unidirectional sink and source functions to form a bidirectional function.

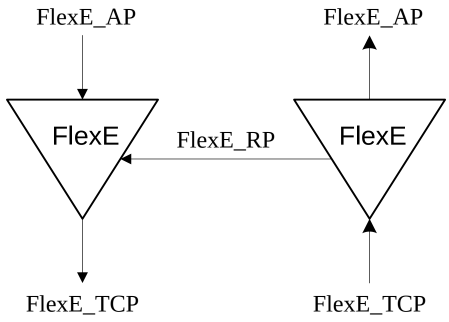


Figure 7-3 – FlexE_TT

7.1.1 FlexE trail termination source function (FlexE_TT_So)

The FlexE_TT_So function adds section monitoring overhead – including the RPF signal – to the FlexE signal at its FlexE_AP.

The information flow and processing of the FlexE_TT_So function is defined with reference to Figures 7-4 and 7-5.

Symbol

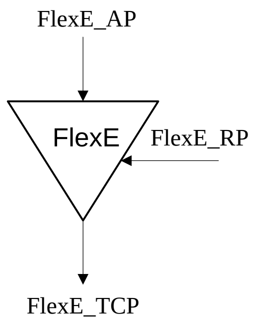


Figure 7-4 – FlexE_TT_So function

Interfaces

Table 7-2 – FlexE_TT_So inputs and outputs

Input(s)	Output(s)
FlexE_AP: FlexE_AI_CK FlexE_AI_D FlexE_AI_FS FlexE_AI_MFS **FlexE_RP**: FlexE_RI_RPF	**FlexE_TCP**: FlexE_CI_CK FlexE_CI_D FlexE_CI_FS FlexE_CI_MFS

Processes

The processes associated with the FlexE_TT_So function are as depicted in Figure 7-5.

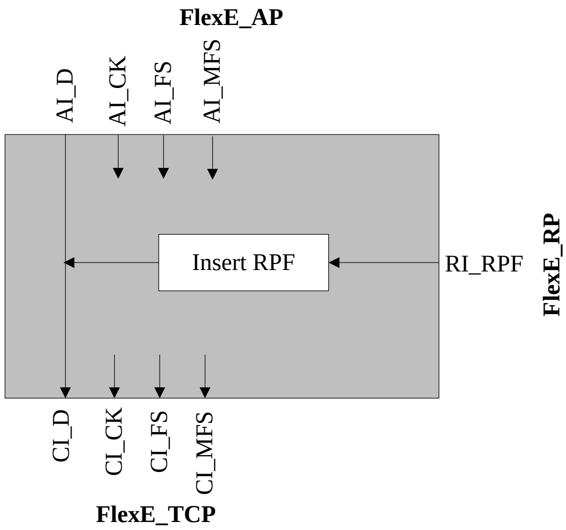


Figure 7-5 – FlexE_TT_So processes

RPF: The remote PHY fault indication is inserted in the RPF bit position as described in clause 7.3.8 of [OIF FLEXE IA]. Its value is derived from reference point FlexE_RP. Upon the declaration/clearing of aRPF at the termination sink function, the trail termination source function shall have inserted/removed the RPF indication within 50 ms.

Defects: None.

Consequent actions: None.

Defect correlations: None.

Performance monitoring: None.

7.1.2 FlexE trail termination sink function (FlexE_TT_Sk)

The FlexE_TT_Sk function reports the state of the FlexE trail. It extracts FlexE monitoring overhead – including the RPF signal – from the FlexE signal at its FlexE_TCP, detects for the RPF defect, and forwards the error and defect information as backward indications to the companion FlexE_TT_So function.

The information flow and processing of the FlexE_TT_Sk function is defined with reference to Figures 7-6 and 7-7.

Symbol

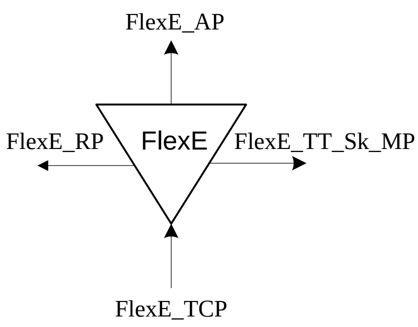


Figure 7-6 – FlexE_TT_Sk function

Interfaces

Table 7-3 – FlexE_TT_Sk inputs and outputs

Input(s)	Output(s)
FlexE_TCP: FlexE_CI_CK FlexE_CI_D FlexE_CI_FS FlexE_CI_MFS FlexE_CI_SSF FlexE_CI_CRCerr	**FlexE_AP**: FlexE_AI_CK FlexE_AI_D FlexE_AI_FS FlexE_AI_MFS FlexE_AI_CRCerr FlexE_AI_TSF **FlexE_RP**: FlexE_RI_RPF **FlexE_TT_Sk_MP**: FlexE_TT_Sk_MI_cRDI FlexE_TT_Sk_MI_cSSF

Processes

The processes associated with the FlexE_TT_Sk function are as depicted in Figure 7-7.

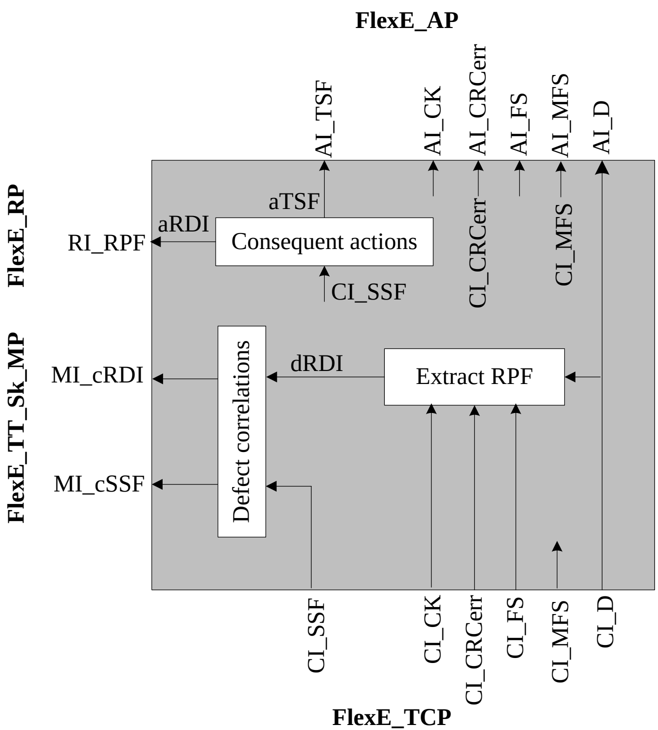


Figure 7-7 – FlexE_TT_Sk processes

RPF: The remote PHY fault indication shall be recovered from the RPF bit position of FlexE overhead field with the good CRC (CRCerr = 0) as described in clause 7.3.8 of [OIF FLEXE IA]. It shall be used for RPF defect detection.

Defects

The function shall detect the dRDI defect.

dRDI: If the extracted RPF is “1” with the good CRC, dRDI shall be set to true; Otherwise, dRDI shall be set to false; dRDI shall be set to false during CI_SSF.

Consequent actions

The function shall perform the following consequent actions:

aRDI ← CI_SSF

aTSF ← CI_SSF

Defect correlations

The function shall perform the following defect correlations to determine the most probable fault cause. This fault cause shall be reported to the EMF.

cSSF ← CI_SSF

cRDI ← dRDI and (not CI_SSF)

Performance monitoring: None.

7.2 FlexE-n to FlexE client adaptation function (FlexE-n/FlexEC_A)

The FlexE-n to FlexEC adaptation functions perform the adaptation between the FlexE-n layer adapted information and the characteristic information of the FlexE client signals.

7.2.1 FlexE-n to FlexE client adaptation source function (FlexE-n/FlexEC_A_So)

The information flow and processing of the FlexE-n/FlexEC_A_So function is defined with reference to Figures 7-8 and 7-9.

Symbol

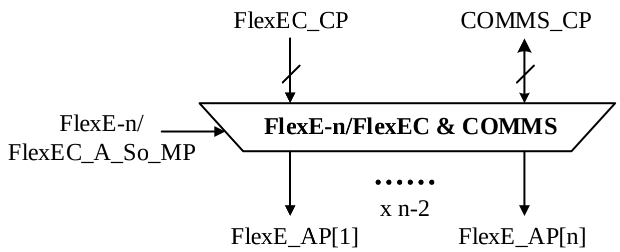


Figure 7-8 – FlexE-n/FlexEC_A_So function

Interfaces

Table 7-4 – FlexE-n/FlexEC_A_So inputs and outputs

Input(s)	Output(s)
p × FlexEC_CP: FlexEC_CI_CK FlexEC_CI_D **n × COMMS_CP**: COMMS_CI_D **FlexE-n/FlexEC_A_So_MP**: FlexE-n/FlexEC_A_So_MI_TxCC FlexE-n/FlexEC_A_So_MI_TxCCA FlexE-n/FlexEC_A_So_MI_TxCCB FlexE-n/FlexEC_A_So_MI_TxCR FlexE-n/FlexEC_A_So_MI_TxCA FlexE-n/FlexEC_A_So_MI_TxGID FlexE-n/FlexEC_A_So_MI_TxPHYMAP	**n x FlexE_AP**: FlexE_AI_D FlexE_AI_CK FlexE_AI_FS FlexE_AI_MFS **n × COMMS_CP**: COMMS_CI_CK

Processes

The processes associated with the FlexE-n/FlexEC_A_So function are as depicted in Figure 7-9

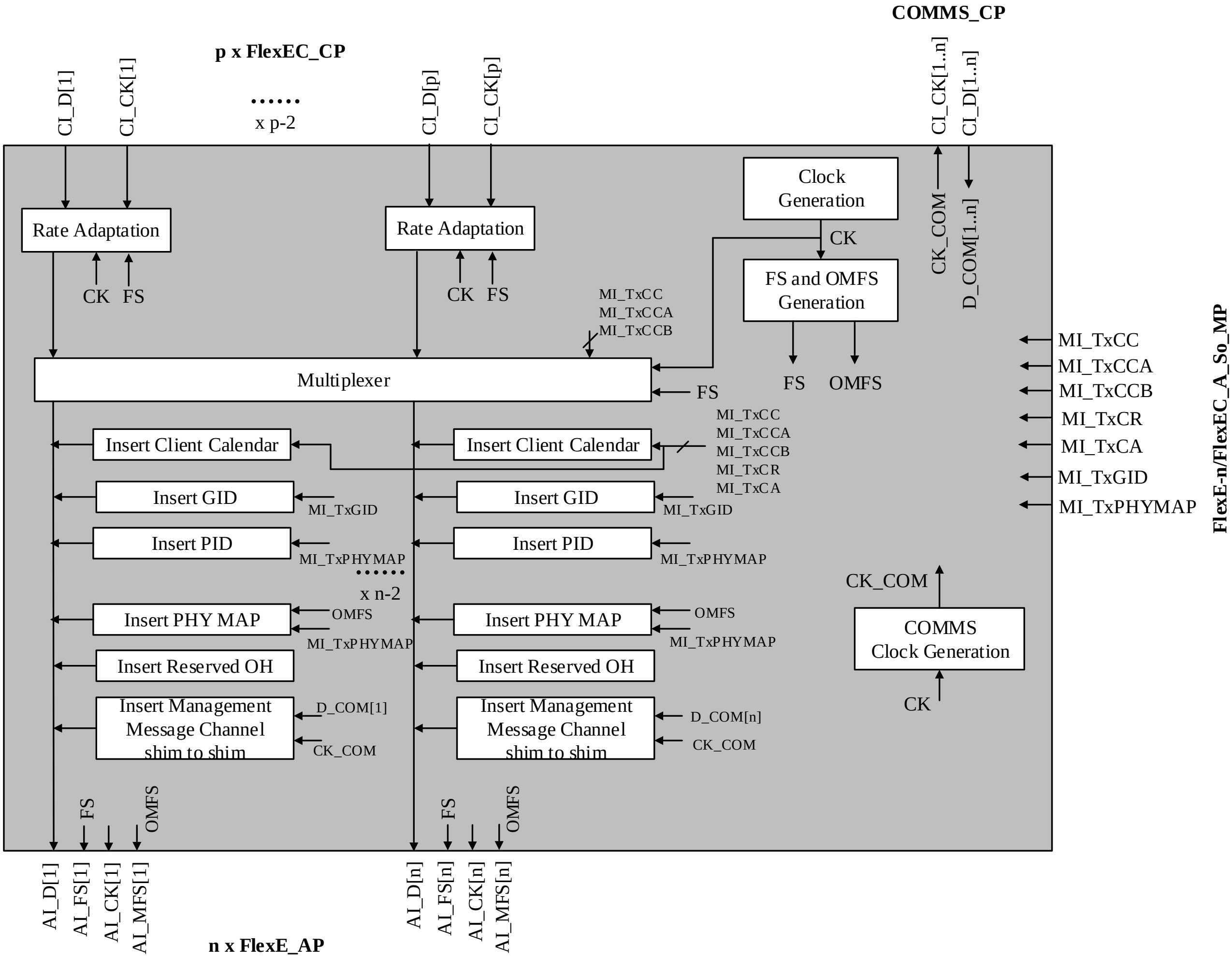


Figure 7-9 – FlexE-n/FlexEC_A_So processes

Clock generation: The function shall generate a local FlexE clock (FlexE_AI_CK) of "16383/16384 × 103 125 000 kHz ± 100 ppm" from a free-running oscillator.

FS and OMFS generation: The function shall generate FlexE frame start (FS) and FlexE overhead multi-frame start (OMFS) signals as described in clause 7.3.1 of [OIF FLEXE IA].

Rate adaptation: The function shall adjust the FlexEC signals applied at the FlexEC_CP input ports to the required rate through insertion of IDLE blocks and deletion of IDLE and Ordered Set blocks as described in clause 5.2.1.2 of [OIF FLEXE IA].

Multiplexing: The function assigns the individual FlexEC to specific calendar slots of the FlexE-n payload area as defined by the client calendar configuration in use (see clauses 7.4 of [OIF FLEXE IA]). The calendar configuration in use is configured either by the MI_TxCCA or the MI_TxCCB as indicated by the MI_TxCC, described in clauses 7.3.2 and 7.3.4 of [OIF FLEXE IA].

Note – It is assumed that the optional client calendar configuration protocol, as defined in clause 7.3.4 of [OIF FLEXE IA], may be implemented by the EMF.

Insert Client Calendar: The function shall insert the calendar configuration information into the FlexE overhead fields as defined in clauses 6.3, 7.3.2 and 7.3.4 of [OIF FLEXE IA]. The “calendar configuration in use” overhead bits are set as configured by the MI_TxCC; the “client carried calendar A/B" overhead fields are set extracting the sub-calendar information from the configured MI_TxCCA and MI_TxCCB. The "client switch request" and "client switch acknowledged" overhead bits are set as configure by the MI_TxCR and MI_TxCA.

FlexE GID: The FlexE group identifier is inserted in the FlexE Group Number field. Its value is configured via the MI_TxGID. The GID format is described in clause 7.3.6 of [OIF FLEXE IA].

FlexE PID: The FlexE physical interface identifier is inserted in the PHY Number field. Its value for each PHY member of FlexE group is a unique one and can be configured via the MI_PHYMAP. The PID format is described in clause 7.3.3 of [OIF FLEXE IA].

FlexE PHY MAP: The FlexE PHY MAP is inserted in the OMF and PHY Map fields using the multi-frame structure shown in Figure 11 of [OIF FLEXE IA]. Its value is configured via the MI_PHYMAP. The PHY MAP format is described in clause 7.3.3 of [OIF FLEXE IA].

Reserved OH: The function shall insert "all zero" in the reserved overhead field and in bit 36 to 63 of the FlexE overhead in row 1 as described in clause 7.3.7 of [OIF FLEXE IA].

Management Message Channel – shim to shim: The Management Message Channel is for user specific and inserted the incoming COMMS (CI_D) data into the Management Message Channel – shim to shim field described in clause 7.3.5 of [OIF FLEXE IA].

COMMS clock generation: The function shall generate the COMMS clock (CI_CK_COM) by dividing the local FlexE clock (CK) by a factor of 3/163688.

Defects: None.

Consequent actions: None.

Defect correlations: None.

Performance monitoring: None.

7.2.2 FlexE-n to FlexE client adaptation sink function (FlexE-n/FlexEC_A_Sk)

The FlexE-n to FlexEC adaptation sink function is defined for FlexE client signals (10G, 40G, and n × 25G FlexE client signals).

The information flow and processing of the FlexE-n/FlexEC_A_Sk function is defined with reference to Figures 7-10 and 7-11.

Symbol

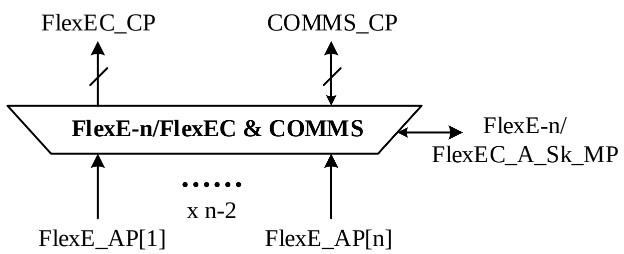


Figure 7-10 – FlexE-n/FlexEC_A_Sk function

Interfaces

Table 7-5 – FlexE-n/FlexEC_A_Sk inputs and outputs

Input(s)	Output(s)
n x FlexE_AP: FlexE_AI_D FlexE_AI_CK(Note) FlexE_AI_FS FlexE_AI_MFS FlexE_AI_CRCerr FlexE_AI_TSF **FlexE-n/FlexEC_A_Sk_MP**: FlexE/FlexEC_A_Sk_MI_ExCC FlexE/FlexEC_A_Sk_MI_ExCCA FlexE/FlexEC_A_Sk_MI_ExCCB FlexE/FlexEC_A_Sk_MI_ExGID FlexE/FlexEC_A_Sk_MI_ExPHYMAP	**p × FlexEC_CP**: FlexEC_CI_CK FlexEC_CI_D FlexEC_CI_SSF **n x COMMS_CP**: COMMS_CI_D COMMS_CI_CK COMMS_CI_SSF **FlexE-n/FlexEC_A_Sk_MP**: FlexE-n/FlexEC_A_Sk_MI_AcCC FlexE-n/FlexEC_A_Sk_MI_AcCR FlexE-n/FlexEC_A_Sk_MI_AcCA FlexE-n/FlexEC_A_Sk_MI_cCCM[1..p] FlexE-n/FlexEC_A_Sk_MI_cLOL FlexE-n/FlexEC_A_Sk_MI_cPMM FlexE-n/FlexEC_A_Sk_MI_cGIDM
Note: It needs only to select one clock of them to use, e.g. FlexE_AI_CK[1].	

Processes

The processes associated with the FlexE-n/FlexEC_A_Sk function are as depicted in Figure 7-11.

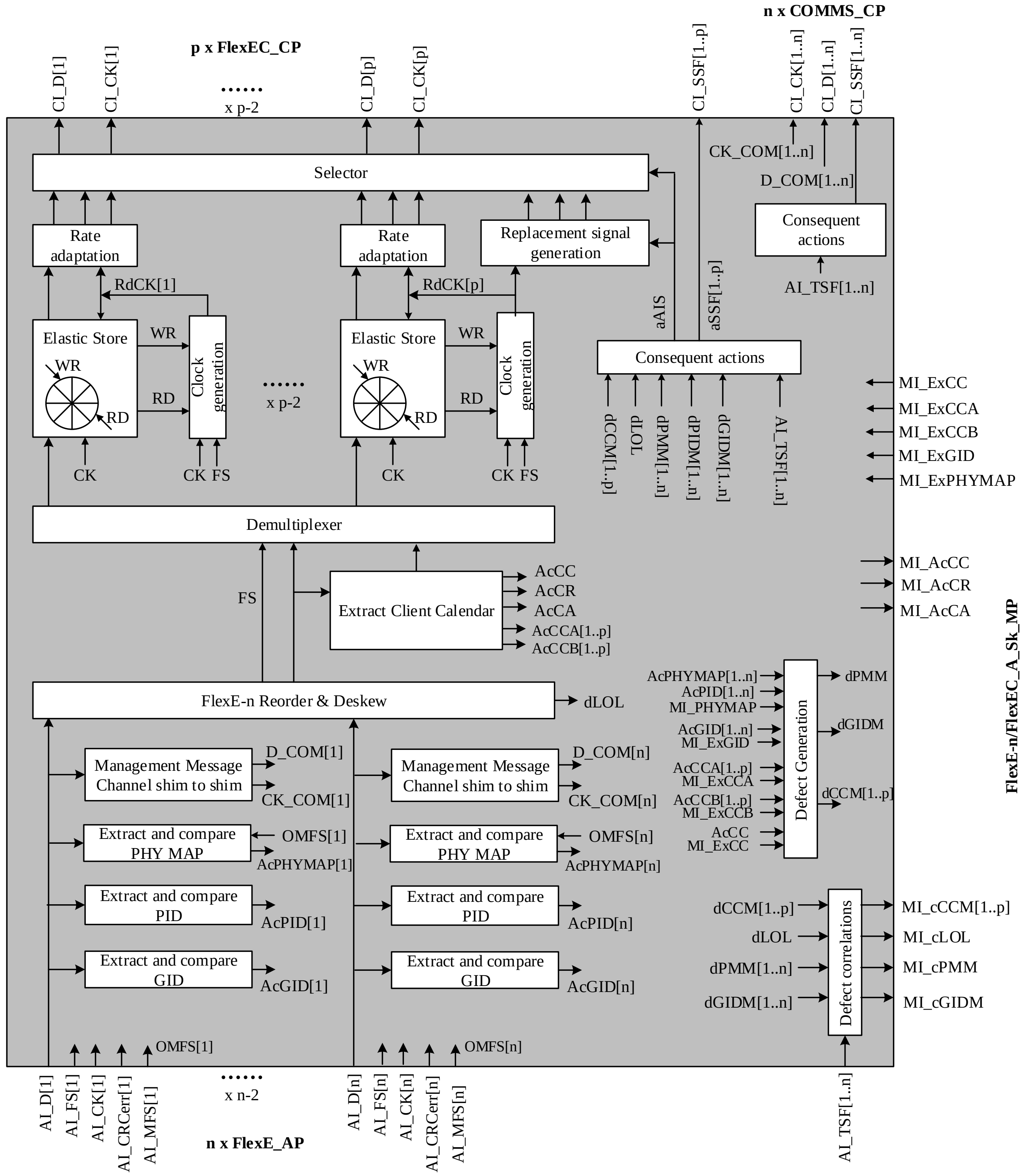


Figure 7-11 – FlexE-n/FlexEC_A_Sk processes

FlexE OH Extration: The function shall extract the overhead of FlexE group interface (GID, PID, PHY MAP and Client Calendar) from each of the n FlexE_AI signals as defined in clause 7.3 of [OIF FLEXE IA].

FlexE GID: The GID fields shall be read from the FlexE overhead and processed as specified in Annex B.2.2.1 of [ITU-T G.798]. The accepted GID values are available at the MP (MI_AcGID[i]) and are used for dGIDM defect detection.

FlexE PID: The PID fields shall be read from the FlexE overhead and processed as specified in Annex B.2.2.2.2 of [ITU-T G.798]. The accepted PID values are available at the MP (MI_AcPID[i]) and are used for dPMM defect detection.

FlexE PHY MAP: The MAP fields shall be read from the FlexE overhead and processed as specified in Annex B.2.2.3.2 of [ITU-T G.798]. The accepted PHYMAP values are available at the MP (MI_AcPHYMAP[i]) and are used for dPMM defect detection.

Management Message Channel: The Management Message Channel is for user specific and the corresponding COMMS data (D_COM) and COMMS clock (CK_COM) is extracted from the Management Message Channel – shim to shim field described in clause 7.3.5 of [OIF FLEXE IA].

FlexE-n Reorder & Deskew: The function shall include reordering and deskewing processes.

Reordering: The process shall reorder n lanes of FlexE PHY based on PID clause 7.3.3 of [OIF FLEXE IA].

Deskewing: The process shall compensate the skew between n FlexE PHY based on FlexE frame start indication (alignment markers) as described in clause 7.3.1 of [OIF FLEXE IA]. The alignment process shall establish the delay compensation, compensating the differential delay between the PHY lane signals as given in clause 6.4 of [OIF FLEXE IA]. The compensation between the PHY lanes is achieved by an elastic store per PHY lane. Each PHY lane signal shall be written into an elastic store with the FlexE frame start indication. Each elastic store shall be capable of compensating at least 300 ns of absolute differential delay between the PHY lanes. The process has two states, out-of-multilane-alignment (OLA) and in-multilane-alignment (ILA). The alignment start shall be maintained during the OLA state. In the OLA state, if the bytes of the PHY lane signals can be written consistently into the elastic store in the presence of a differential delay in line without exceeding the buffering time, the ILA state shall be entered. In this case, the differential delay can be compensated. In the ILA state, if the differential delay between two PHY lanes exceeds the maximum delay that can be compensated, the OLA state shall be entered.

FlexE OH client calendar: The calendar information shall be read from the calendar configuration in use (C), client calendar A and B, calendar switch request (CR) and calendar switch acknowledge (CA) overheads as defined in clauses 6.4, 7.3.2 and 7.3.4 of [OIF FLEXE IA].

Calendar configuration in use overhead (C): The "calendar configuration in use" overhead from each member shall be accepted (AcCC[*i*]) by majority vote of the 3 C overhead bits. Furthermore, it shall confirm the accepted "AcCC" by unanimity of n PHY members of FlexE group.

Client calendar A and B overheads: The "client calendar A" and "client calendar B" overhead fields from each member shall be read and the calendar slot information shall be accepted in overhead frames with good CRC (AcCCA[i] and AcCCB[i]). The client calendar mismatch process reports the client calendar mismatch defect (dCCM[1..p]) for p FlexEC clients.

Calendar switch request overhead (CR): The "calendar switch request" overhead from each member shall be read and the calendar switch request information shall be accepted in overhead frames with good CRC (AcCR[*i*]). Furthermore, it shall confirm the accepted "AcCR" value by unanimity of n PHY members of FlexE group.

Calendar switch acknowledge overhead (CA): The "calendar switch acknowledge" overhead from each member shall be read and the calendar switch acknowledge information shall be accepted in overhead frames with good CRC (AcCA[*i*]). Furthermore, it shall confirm the accepted “AcCA” value by unanimity of n PHY members of FlexE group.

Demultiplexing: The function activates the FlexEC and assigns the calendar slots of the FlexE-n payload area to the individual FlexEC as defined by the client calendar in use (see clause 7.5 of [OIF FLEXE IA]). The calendar configuration in use is derived by the configuration provided by either the MI_ExCCA[1..p] or the MI_ExCCB[1…p] as indicated by the MI_ExCC or, when the MI_ExCC is set to any, by the accepted client calendar (AcCCA[1…p] or AcCCB[1…p]), as described in clauses 7.3.2 and 7.3.4 of [OIF FLEXE IA].

Note – It is assumed that the optional client calendar configuration protocol, as defined in clause7.3.4 of [OIF FLEXE IA], may be implemented by the EMF.

Clock generation: The function shall generate an FlexEC clock (FlexEC_CI_CK[1..p]) from a free-running oscillator or the FlexE-n group clock with a bit rate as specified in Table 7-1.

Rate adaptation: The function shall adjust the demapped FlexEC to the required rate through insertion of IDLE blocks or deletion of IDLE or ordered set blocks as described in clauses 5.2.1.1 and 5.2.1.2 of [OIF FLEXE IA].

Replacement signal generation: When aAIS[*j*] is true, the function shall provide for a replacement signal of FlexEC and clock generation process that generates a stream of local fault sequence ordered sets as specified in clause 5.2.2.3 of [OIF FLEXE IA].

Selector: The function shall select the normal FlexEC signal or the replacement signal. When aAIS[*j*] is true, it shall select the replacement signal. Otherwise, it shall select the normal FlexEC signal.

Defects

The function shall detect dGIDM, dPMM, dLOL and dCCM[*j*].

dGIDM: See Annex B.1.1.2.1 of [ITU-T G.798].

dPMM: See Annex B.1.1.2.2 of [ITU-T G.798].

dLOL: If the alignment process is in the out-of-alignment state, dLOL shall be set to true. dLOL shall be set to false when the alignment process is in the in-multilane-alignment state.

dCCM[*j*] (Client Calendar Mismatch): The dCCM defect for FlexEC tributaries is based on comparing the AcCC/AcCCA/AcCCB and ExCC/ExCCA/ExCCB of all FlexE PHY instances. Each FlexEC shall be corresponding to an dCCM, e.g. dCCM[*j*] where *j* = 1 to p denotes FlexEC tributary port.

Upon accepting a new AcCC/AcCCA/AcCCB value for a FlexE PHY instances the nCCM anomalies for all tributary ports are initialized to false. The AcCC/AcCCA/AcCCB are then compared to the ExCC/ExCCA/ExCCB for all FlexE PHY instances.

When AcCC is equalling to ExCC and both them are equalling ‘1’, for each case where the AcCCA/and ExCCA values pertaining to the same tributary slot are not equal,

a) if the tributary slot is allocated in the ExCCA value, the nCCManomaly of the tributary port subfield of the ExCCA value is set,
b) if the tributary slot is allocated in the AcCCA value, the nCCManomaly of the tributary port subfield of the AcCCA value is set.

When AcCC is equalling to ExCC and both them are equalling ‘0’, for each case where the AcCCB/and ExCCB values pertaining to the same tributary slot are not equal,

a) if the tributary slot is allocated in the ExCCB value, the nCCManomaly of the tributary port subfield of the ExCCB value is set,
b) if the tributary slot is allocated in the AcCCB value, the nCCManomaly of the tributary port subfield of the AcCCB value is set.

The dCCM defect of each tributary port shall be set to the corresponding nCCM anomaly.

```
for trib_port = 1 to number_of_trib_ports
    nCCM[trib_port] = false


for FlexE = 1 to number_of_instances
 for trib_slot = 1 to number_of_trib_slots
     if AcCC == '1' and ExCC == '1'
              if AcCCA[FlexE][trib_slot] != ExCCA[FlexE][trib_slot]
                    if ExCCA[FlexE][trib_slot].allocated
                         nCCM[ ExCCA[trib_slot].trib_port ] = true
                    if AcCCA[FlexE][trib_slot].allocated
                         nCCM[ AcCCA[trib_slot].trib_port ] = true
     else if AcCC == '0' and ExCC == '0'
              if AcCCB[FlexE][trib_slot] != ExCCB[FlexE][trib_slot]
                    if ExCCB[FlexE][trib_slot].allocated
                         nCCM[ ExCCB[trib_slot].trib_port ] = true
                    if AcCCB[FlexE][trib_slot].allocated
                         nCCM[ AcCCB[trib_slot].trib_port ] = true


for trib_port = 1 to number_of_trib_ports
    dCCM[trib_port] = nCCM[trib_port]
```

Consequent actions

aSSF[*j*] ← dCCM[*j*] or dLOL or dPMM or dGIDM or ∑AI_TSF[1..n] , where *j* = 1 to p.

aAIS[*j*] ← dCCM[*j*] or dLOL or dPMM or dGIDM or ∑AI_TSF[1..n] , where *j* = 1 to p.

Defect correlations

cGIDM ← dGIDM and (not ∑AI_TSF[1..n])

cPMM ← dPMM and (not dGIDM) and (not ∑AI_TSF[1..n])

cLOL ← dLOL and (not dPMM) and (not dGIDM) and (not ∑AI_TSF[1..n])

cCCM[*j*] ← dCCM[*j*] and (not dLOL) and (not dPMM) and (not dGIDM) and (not ∑AI_TSF[1..n]), where *j* = 1 to p.

Performance monitoring: None.

7.3 Interworking function between FlexE Client and ETCy (FlexEC<>ETCy)

The FlexEC<>ETCy interworking functions perform the conversion between FlexEC_CI signals and ETCy_CI signals. It shall be used between FlexE-n/FlexEC_A and ODUkP/ETCy_A as shown in Figure 7-12.

Table 7-6 – ETCy_CI and FlexEC_CI

y	ETCy_CI				FlexEC_CI			
	Bit rate (kbit/s)	**Clock tolerance (ppm)**	**Encoding**	**AM**	**Bit rate (kbit/s)**	**Clock toleranc e(ppm)**	**Encoding**	**AM**
10GR	10 312 500	± 100	Clause 49 64b/66b	No	10 312 500	± 100	Clause 82 64b/66b	No
25GR	25 781 250	± 100	Clause 49 64b/66b	No	25 781 250	± 100	Clause 82 64b/66b	No
40GR	41 250 000	± 100	Clause 82 64b/66b	Yes	41 250 000	± 100	Clause 82 64b/66b	No
100GR	103 125 000	± 100	Clause 82 64b/66b	Yes	103 125 000	± 100	Clause 82 64b/66b	No

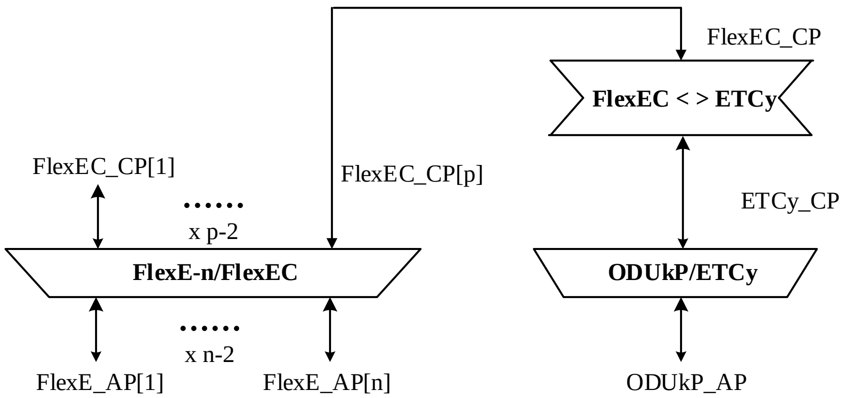


Figure 7-12 – FlexEC<>ETCy function position

7.3.1 FlexEC to ETCy interworking function (FlexEC> ETCy)

The information flow and processing of the FlexEC to ETCx interworking function is defined with reference to Figures 7-13, 7-14 and 7-15.

Symbol

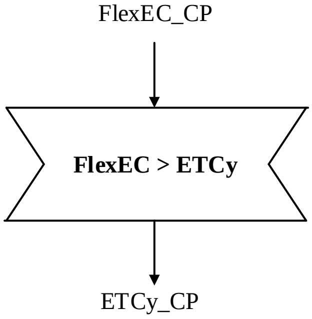


Figure 7-13 – FlexEC>ETCy function

Interfaces

Table 7-7 – FlexEC>ETCy inputs and outputs

Input(s)	Output(s)
FlexEC_CP: FlexEC_CI_CK FlexEC_CI_D FlexEC_CI_SSF	**ETCy_CP**: ETCy_CI_CK ETCy_CI_D ETCy_CI_SSF

Processes

The processes associated with the FlexEC>ETCy function are as depicted in Figures 7-14 (y=10GR or 25GR) and 7-15 (y=40GR or 100GR).

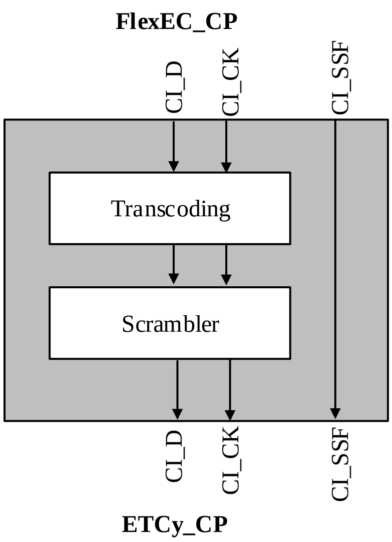


Figure 7-14 – FlexEC>ETCy processes (y = 10GR, 25GR)

Transcoding: The function shall transcode 66B blocks based on clause 82 of [IEEE 802.3] to 66B blocks based on clause 49 of [IEEE 802.3].

Scrambler: The function shall scramble the 66B blocks as described in clause 49.2.6 of [IEEE 802.3].

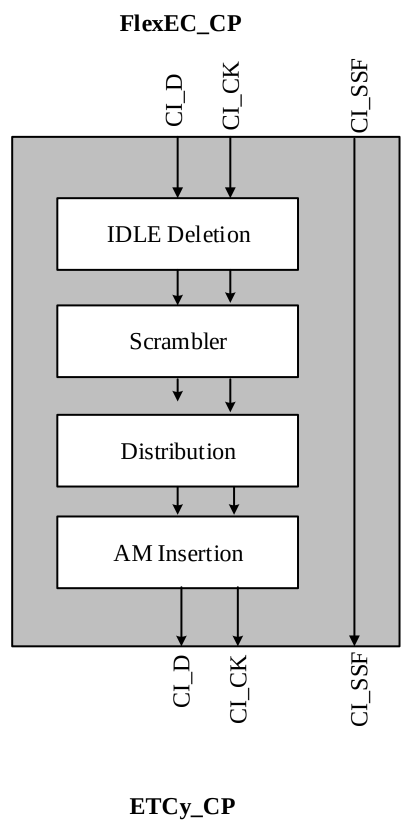


Figure 7-15 – FlexEC>ETCy processes (y = 40GR, 100GR)

IDLE deletion: The function shall delete idle control characters or sequence ordered sets in the data stream of 66b blocks to accommodate the transmission of alignment markers as described in clause 82.2.4 of [IEEE 802.3].

Scrambler: The function shall scramble the input 66B blocks as described in clause 82.2.5 of [IEEE 802.3].

Distribution: The function shall distribute the scrambled 66B blocks into 4 (for ETC40GR) or 20 (ETC100GR) PCS lanes in a round-robin manner as described in clause 82.2.6 of [IEEE 802.3].

Alignment Marker insertion: The function shall insert the alignment markers into 4 or 20 PCS lanes after every 16383 66-bit blocks on each PCS lane as described in clause 82.2.7 of [IEEE 802.3].

Defects: None.

Consequent actions: None.

Defect correlations: None.

Performance monitoring: None.

7.3.2 ETCy to FlexEC interworking function (ETCy>FlexEC)

The information flow and processing of the ETCy>FlexEC function is defined with reference to Figures 7-16, 7-17 and 7-18.

Symbol

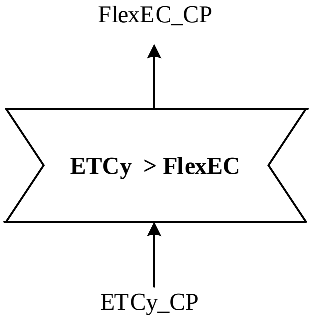


Figure 7-16 – ETCy>FlexEC function

Interfaces

Table 7-8 – ETCy>FlexEC inputs and outputs

Input(s)	Output(s)
ETCy_CP: ETCy_CI_CK ETCy_CI_D ETCy_CI_SSF	**FlexEC_CP**: FlexEC_CI_CK FlexEC_CI_D FlexEC_CI_SSF

Processes

The processes associated with the ETCy>FlexEC function are as depicted in Figures 7-17 (y=10GR or 25GR) and 7-18 (y=40GR or 100GR).

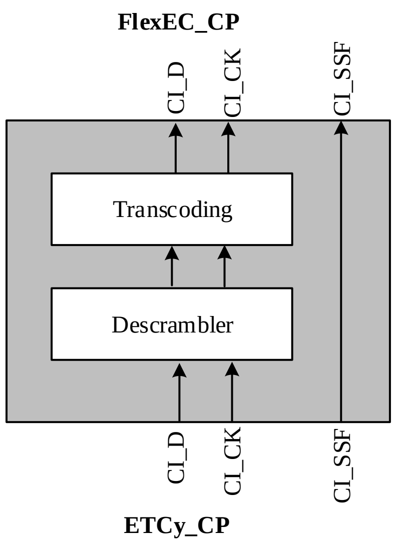


Figure 7-17 – ETCy>FlexEC processes (y = 10GR, 25GR)

Transcoding: The function shall transcode 66B blocks based on clause 49 of [IEEE 802.3] to 66B blocks based on clause 82 of [IEEE 802.3].

Descrambler: The function shall descramble the input 66B blocks as described in clause 49.2.10 of [IEEE 802.3].

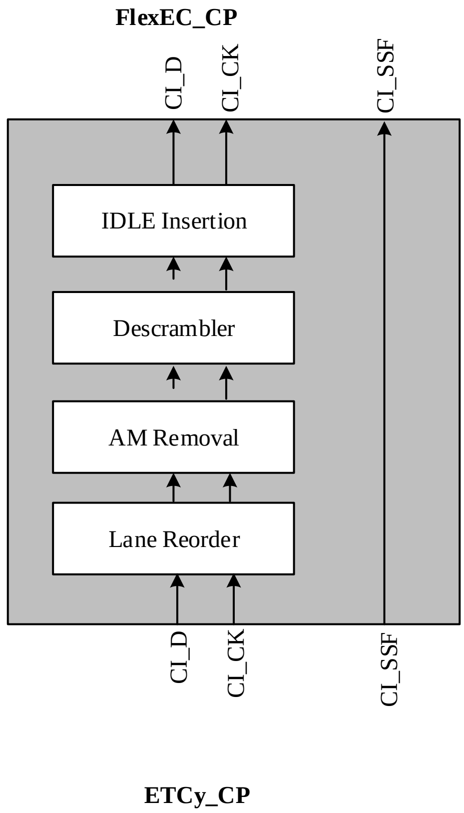


Figure 7-18 – ETCy>FlexEC processes (y = 40GR, 100GR)

Lane reorder: The function shall order the 4 (ETC40GR) or 20 PCS (ETC100GR) lanes according to their PCS lane number as described in the clause 82.2.14 of [IEEE 802.3].

Alignment Marker removal: The function shall remove the alignment markers from the data stream of 66b blocks. 4 alignment markers are removed per 4*16383 66-bit blocks for ETC40GR and 20 alignment markers are removed per 20*16383 66-bit blocks for ETC100GR.

Descrambler: The function shall descramble the input 66B blocks as described in clause 82.2.16 of [IEEE 802.3].

IDLE insertion: The function shall insert idle control characters in the data stream of 66b blocks to compensate for the removal of alignment markers as described in clause 82.2.15 of [IEEE 802.3].

Defects: None.

Consequent actions: None.

Defect correlations: None.

Performance monitoring: None.

8. FlexE client functions

8.1 FlexE client trail termination function (FlexEC_TT)

The FlexEC_TT function terminates the flexE client layer. Figure 8-1 shows the combination of the unidirectional sink and source functions to form a bidirectional function. As there is no overhead defined for monitoring a flexE client, this is a null function.

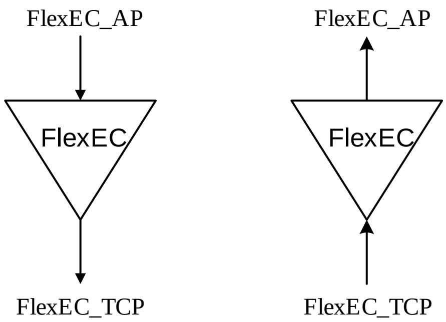


Figure 8-1 – FlexEC_TT

8.1.1 FlexEC trail termination source function (FlexEC_TT_So)

The FlexEC_TT_So function is a null function. It passes the signal from the FlexEC_AP to the FlexEC_TCP

Symbol

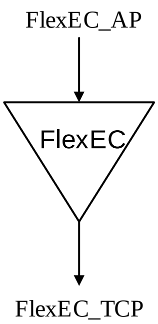


Figure 8-2 – FlexEC_TT_So function

Interfaces

Table 8-1 – FlexEC_TT_So inputs and outputs

Input(s)	Output(s)
FlexEC_AP: FlexEC_AI_CK FlexEC_AI_D	**FlexEC_TCP**: FlexEC_CI_CK FlexEC_CI_D

Processes

There are no processes performed by this function.

Defects: None.

Consequent actions: None.

Defect correlations: None.

Performance monitoring: None.

8.1.2 FlexEC trail termination sink function (FlexEC_TT_Sk)

The FlexEC_TT_Sk function is a null function. It passes the signal from the flexEC_TCP to the flexEC_AP

The information flow and processing of the FlexEC_TT_Sk function is defined with reference to Figures 8-3 and 8-4.

Symbol

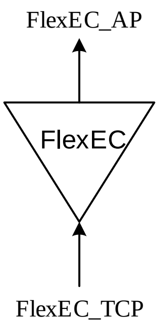


Figure 8-3 – FlexEC_TT_Sk function

Interfaces

Table 8-2 – FlexEC_TT_Sk inputs and outputs

Input(s)	Output(s)
FlexEC_TCP: FlexEC_CI_CK FlexEC_CI_D FlexEC_CI_SSF	**FlexEC_AP**: FlexCE_AI_CK FlexEC_AI_D FlexEC_AI_TSF

Processes

The processes associated with the FlexEC_TT_Sk function are as depicted in Figure 8-4.

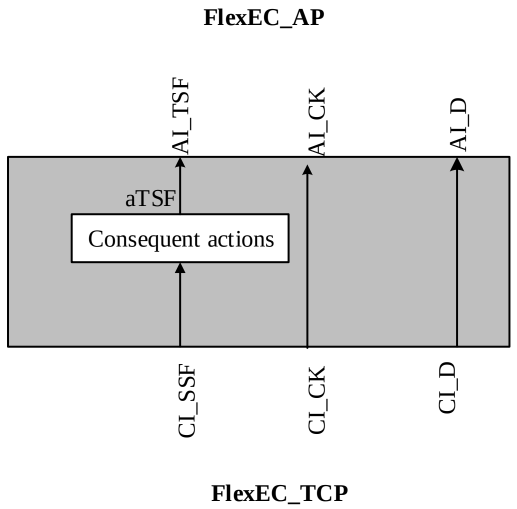


Figure 8-4 – FlexEC_TT_Sk processes

Defects

None

Consequent actions

The function shall perform the following consequent actions:

aTSF ← CI_SSF

Defect correlations

None

Performance monitoring: None.

8.2 FlexE client to Ethernet MAC layer adaptation function (FlexEC/ETH_A)

The FlexEC to ETH adaptation functions perform the adaptation between a flexE client signal and the Ethernet MAC layer.

8.2.1 FlexEC to ETH adaptation source function (FlexEC/ETH_A_So)

The information flow and processing of the FlexEC/ETH_A_So function is defined with reference to Figures 8-5 and 8-6.

Symbol

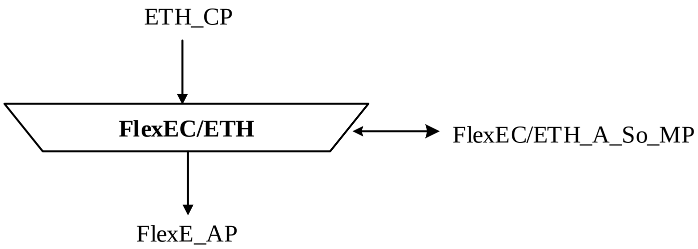


Figure 8-5 – FlexEC/ETH_A_So function

Interfaces

Table 8-3 – FlexEC/ETH_A_So inputs and outputs

Input(s)	Output(s)
ETH_TFP: ETH_CI_D ETH_CI_P ETH_CI_DE **ETH_FP:** ETH_CI_D ETH_CI_P ETH_CI_DE **FlexEC/ETH_A_So_MP:** FlexEC.ETH_A_SO_MI_[IEEE 802.3]	**FlexEC_AP**: FlexEC_AI_D FlexEC_AI_CK **ETHTF_PP:** ETH_CI_D ETH_CI_P ETH_CI_DE **ETH_PP:** ETH_CI_D ETH_CI_P ETH_CI_DE **FlexEC/ETH_A_So_MP**: FlexEC/ETH_A_So_MI_[IEEE 802.3]

Processes

The processes associated with the FlexEC/ETH_A_So function are as depicted in Figure 8-6.

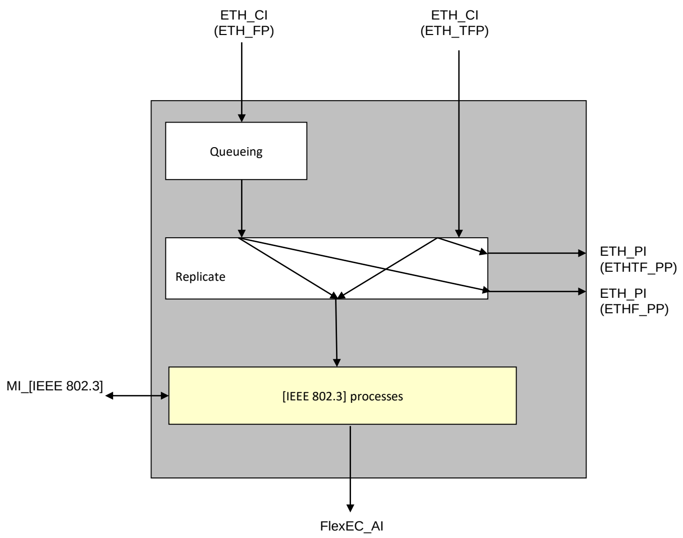


Figure 8-6 – FlexEC/ETH_A_So processes

The queueing and replicate processes are defined in clause 8 of [ITU-T G.8021].

[IEEE 802.3] processes: The [IEEE 802.3] processes represent the whole functionality of the PHY above the flex-E shim in the IEEE 802.3 model. This includes the reconciliation sublayer and the process of 64b/66b encoding, as well as MAC FCS generation and frame counting.

NOTE – This Recommendation defines these processes by reference to [IEEE 802.3] and intentionally does not provide details, as this functionality is well understood from the IEEE work.

Defects: None.

Consequent actions: None.

Defect correlations: None.

Performance monitoring: None.

8.2.2 FlexEC to ETH adaptation sink function (FlexEC/ETH_A_Sk)

The information flow and processing of the FlexEC/ETH_A_Sk function is defined with reference to Figures 8-7 and 8-8.

Symbol

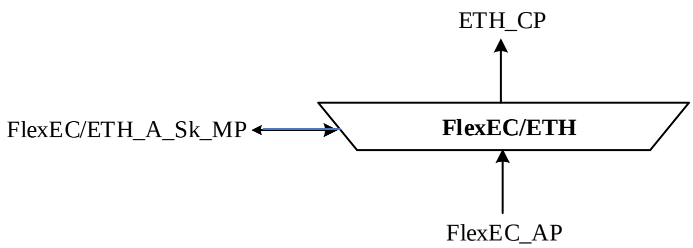


Figure 8-7 – FlexEC/ETH_A_Sk function

Interfaces

Table 8-4 – FlexEC/ETH_A_Sk inputs and outputs

Input(s)	Output(s)
FlexEC_AP: FlexEC_AI_D FlexEC_AI_CK FlexEC_AI_TSF **ETHTF_PP:** ETH_CI_D ETH_CI_P ETH_CI_DE **ETH_PP:** ETH_CI_D ETH_CI_P ETH_CI_DE **FlexEC/ETH_A_Sk_MP**: FlexEC/ETH_A_Sk_MI_[IEEE 802.3] FlexEC/ETH_A_Sk_MI_FilterConfig	**ETH_TFP**: ETH_CI_D ETH_CI_P ETH_CI_DE ETH_CI_SSF **ETH_FP:** ETH_CI_D ETH_CI_P ETH_CI_DE ETH_CI_SSF **FlexEC/ETH_A_Sk_MP**: FlexEC/ETH_A_Sk_MI_[IEEE 802.3]

Processes

The processes associated with the FlexEC/ETH_A_Sk function are as depicted in Figure 8-8.

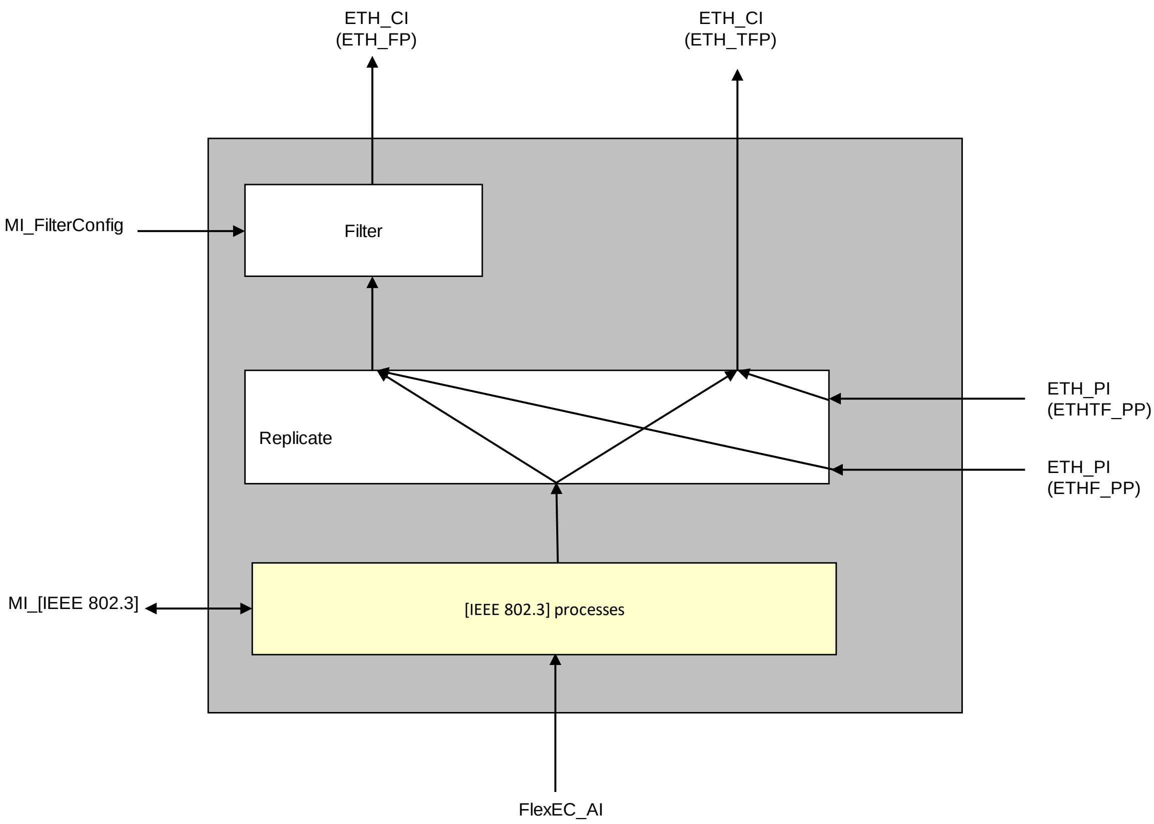


Figure 8-8 – FlexEC/ETH_A_Sk processes

The filter and replicate processes are defined in clause 8 of [ITU-T G.8021].

[IEEE 802.3 processes]: The [IEEE 802.3] processes represent the whole functionality of the PHY layer above the flex-E shim; this includes 64b/66b decoding and the reconciliation sublayer, as well as the MAC length check, MAC frame check and frame counting.

NOTE – This Recommendation defines these processes by reference to [IEEE 802.3] and intentionally does not provide details, as this functionality is well understood from the IEEE work.

Defects

The definition of the defects used by the flexEC/ETH_A_Sk is outside the scope of this Recommendation.

Consequent actions aSSF ← AI_TSF

Defect correlations None.

Performance monitoring For further study.

Annex A

Overview related of FlexE atomic functions

(This annex forms an integral part of this Recommendation.)

This annex describes an overview related of FlexE atomic functions. The left side of Figure A-1 shows the atomic functions of FlexE from the OTSiG up through the ETH layer, including those functions necessary to support FlexE-aware mapping to OTN and FlexE termination with mapping of FlexE client to OTN or switching in the ETH layer. The right side of Figure A-1 shows the atomic functions related to mapping or interconnecting FlexE into OTN. It totally includes 3 cases. In the case 1, when mapping FlexE subrating signals into ODUflex (i.e., FlexE-aware mapping), it connects FlexE CI at FlexE_CP into ODUflexP/FlexESG_A (see clause 14.3.18 of G.798); in the case 2, when mapping FlexE client signals into ODUflex, it connects FlexEC CI at FlexEC_CP into ODUflexP/FlexEC_A (see clause 14.3.17 of G.798); in the case 3, when interconnecting FlexE client signal into the legacy ODUk mapping path, it needs first through the interworking atomic function of FlexEC to ETCy to perform the conversion between FlexEC_CI signals and ETCy_CI signals and then connects the conversed ETCy_CI at ETCy_CP into ODUkP/CBRx-g_A (see clause 14.3.8 of G.798).

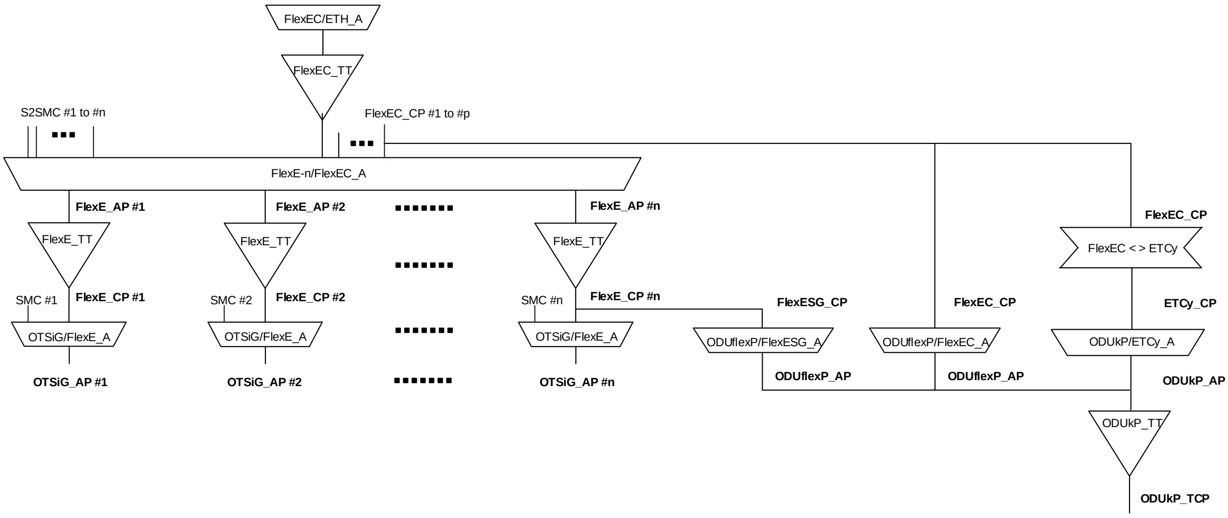


Figure A-1 – Overview of FlexE-related atomic functions