International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.8023

Amendment 1
(02/2022)

SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

Packet over Transport aspects – Ethernet over Transport aspects

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

Amendment 1

Recommendation ITU-T G.8023 (2018) – Amendment 1

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TRANSMISSION MEDIA AND OPTICAL SYSTEMS CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
MULTIMEDIA QUALITY OF SERVICE AND PERFORMANCE – GENERIC AND USER-RELATED ASPECTS	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DATA OVER TRANSPORT – GENERIC ASPECTS	G.7000–G.7999
PACKET OVER TRANSPORT ASPECTS	G.8000–G.8999
Ethernet over Transport aspects	**G.8000–G.8099**
MPLS over Transport aspects	G.8100–G.8199
Synchronization, quality and availability targets	G.8200–G.8299
Mobile network transport aspects	G.8300–G.8399
Service Management	G.8600–G.8699
ACCESS NETWORKS	G.9000–G.9999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T G.8023

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

Amendment 1

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.

Amendment 1 contains text modifications:

– to update references in clauses 1, 2, and 6.3;
– to align terminology with OIF FLEXE IA;
– to align terminology with Recommendation ITU-T G.807;
– to add 50G PHYs from 802.3cd;
– to clarify that the use of the ESMC is optional;
– to update server/ETH_A functions to remove details of client-specific processes that are covered in Recommendation ITU-T G.8021;
– to correct errors in Tables 8-4 and 8-5;
– to add new clause 6.6 and update Annex A regarding FlexE aware mapping;
– to include common processes for FlexE that were formerly in Annex B of Recommendation ITU-T G.798.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T G.8023	2018-06-06	15	11.1002/1000/13544
1.1	ITU-T G.8023 (2018) Cor. 1	2018-11-29	15	11.1002/1000/13758
1.2	ITU-T G.8023 (2018) Amd. 1	2022-02-13	15	11.1002/1000/14901

Keywords

Ethernet, functional blocks, physical layer.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Introduction

Recommendation ITU-T G.8023 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 Recommendation 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).

Recommendation ITU-T G.8023

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

Amendment 1

Editorial note: This is a complete-text publication. Modifications introduced by this amendment are shown in revision marks relative to Recommendation ITU-T G.8023 (2018) plus its Corrigendum 1.

1 Scope

This Recommendation specifies the functions required to insert and extract information to/from an Ethernet physical layer (PHY) as defined in [IEEE 802.3], including the FlexE shim as defined in ~~the~~ [OIF FLEXE IA] ~~implementation agreement~~, for the purposes of mapping into an optical channel data unit (ODU) as defined in [ITU-T G.798], or performing media access control (MAC) layer operations on Ethernet frames as defined in [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.

[ITU-T G.709] Recommendation ITU-T G.709/Y.1331 (2016), *Interfaces for the optical transport network (OTN)*.

[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), *Characteristics of transport equipment – Description methodology and generic functionality*.

[ITU-T G.807] Recommendation ITU-T G.807 (2020), *Generic functional architecture of the optical media network.*

[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.7041] Recommendation ITU-T G.7041/Y.1303 (2016), *Generic framing procedure*.

[ITU-T G.8010] Recommendation ITU-T G.8010/Y.1306 (2004), *Architecture of Ethernet layer networks*.

[ITU-T G.8021] Recommendation ITU-T G.8021/Y.1341 (2018), *Characteristics of Ethernet transport network equipment functional blocks*.

[IEEE 802.1AB] IEEE 802.1AB (2016), *IEEE Standard for Local and metropolitan area networks: Station and Media Access Control Connectivity Discovery*.

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

[IEEE 802.3] IEEE 802.3 (~~2015~~2018), *IEEE Standard for Ethernet*.

~~[IEEE 802.3bs] IEEE 802.3bs (2017), *IEEE Standard for Ethernet – Amendment 10: Media Access Control Parameters, Physical Layers and Management Parameters for 200 Gb/s and 400 Gb/s Operation*.~~

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

[IEEE 802.3cd] IEEE 802.3cd (2018), *IEEE Standard for Ethernet – Amendment 3: Media Access Control Parameters for 50 Gb/s and Physical Layers and Management Parameters for 50 Gb/s, 100 Gb/s, and 200 Gb/s Operation.*

[OIF FLEXE IA] Optical Internetworking Forum, *Flex Ethernet Implementation Agreement ~~1.1~~2.2* (~~2017~~2021).

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 **50GBASE-R**: [IEEE 802.3cd]

3.1.~~7~~8 **access point**: [ITU-T G.805]

3.1.~~8~~9 **adaptation**: [ITU-T G.805]

3.1.~~9~~10 **adapted information**: [ITU-T G.805]

3.1.~~10~~11 **characteristic information**: [ITU-T G.805]

3.1.~~11~~12 **connection point**: [ITU-T G.805]

3.1.~~12~~13 **consequent actions**: [ITU-T G.805]

3.1.~~13~~14 **defect correlations**: [ITU-T G.805]

3.1.~~14~~15 **defects**: [ITU-T G.805]

3.1.16 **FlexE Client**: [OIF FLEXE IA]

3.1.17 **FlexE Group**: [OIF FLEXE IA]

3.1.18 **FlexE Instance**: [OIF FLEXE IA]

3.1.~~15~~19 **flow point**: [ITU-T G.809]

3.1.~~16~~20 **optical tributary signal**: [ITU-T G.959.1]

3.1.~~17~~21 **reconciliation sublayer**: [IEEE 802.3]

3.1.~~18~~22 **termination flow point**: [ITU-T G.809]

3.1.~~19~~23 **traffic unit**: [ITU-T G.809]

3.1.~~20~~24 **trail termination**: [ITU-T G.805]

3.2 Terms defined in this Recommendation

This Recommendation defines the following term:

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
EMF	Equipment Management Function
ERSy	Ethernet Reconciliation sublayer for PHY y
ESMC	Ethernet Synchronization Message Channel
ETCy	Ethernet Coding sublayer for PHY y
ETH	Ethernet MAC layer
ETSi[G]	Electrical Tributary Signal or Electrical Tributary Signal Group
FCS	Frame Check Sequence
FlexE	Flex-Ethernet
FlexEC	Flex-Ethernet Client
FP	Flow Point
FS	Frame Start
GID	Group Identification
IF	In-Frame
IID	Instance Identification
ILA	In-multiLane-Alignment

M-AI	Media layer Adapted Information
MAC	Media Access Control
MFS	MultiFrame Start
MI	Management Information
MP	Management Point
ODU	Optical channel Data Unit
OMF	Overhead Multi-Frame
OOF	Out-Of-Frame
OLA	Out-of-multilane-Alignment
OTN	Optical Transport Network
OTSi[G]	Optical Tributary Signal or Optical Tributary Signal Group
P	Priority
PCS	Physical Coding Sublayer
PDU	Protocol Data Unit
PHY	Ethernet Physical layer
PMA	Physical Medium Attachment
PMD	Physical Medium Dependent
PP	replication Point
RP	Remote Point
RPF	Remote PHY Fault
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_[IEEE 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.~~

Because [IEEE 802.3] provides very detailed description of Ethernet interfaces already, this Recommendation does not attempt to re-create all of the detailed information. Instead, the set of processes that represent an Ethernet interface is modelled as a single adaptation function between the media and the digital (sub)layer of interest. The server layer for the adaptation functions is modelled as a group of one or more optical or electrical signals. Following the convention of

[ITU-T G.807], the signals at the access point of the adaptation function are named media layer adapted information (M-AI).

The convention MI_[IEEE 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 an adaptation function. Their detailed definition is intentionally left outside the scope of this Recommendation.

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.

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 (CBR) 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]~~M-AI 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 optical transport network (OTN) signals in [ITU-T G.798]. Per the convention of [ITU-T G.807], these signals are described as media layer adapted information (M-AI).

This clause describes the adaptation functions for the ~~OTSi[G] and ETSi[G]~~M-AI 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. Figure 6-1 illustrates the ~~OTSi[G] and ETSi[G]~~M-AI to client adaptation functions.

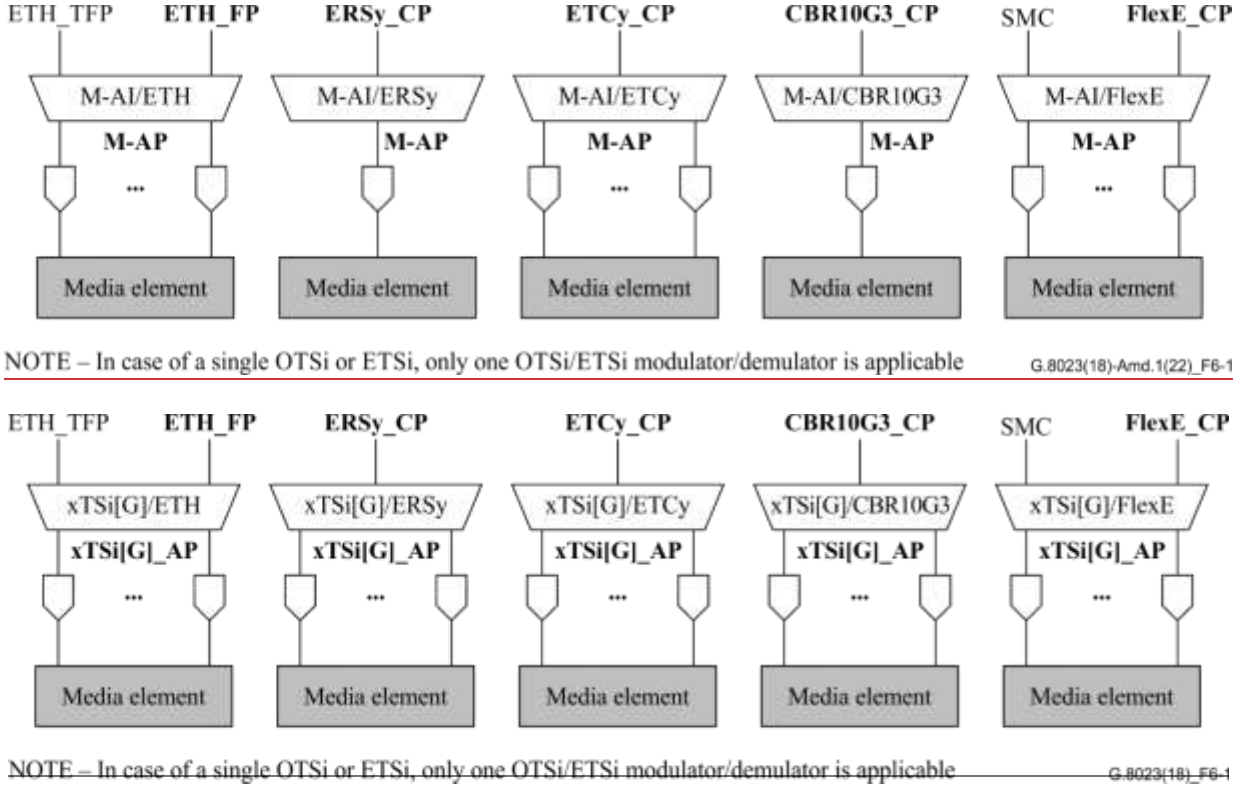


Figure 6-1 – ~~OTSi[G] and ETSi[G]~~M-AI to client adaptation functions

6.1 ~~xTSi[G]~~M-AI to ETH adaptation functions (~~xTSi[G]~~M-AI/ETH_A)

The ~~xTSi[G]~~M-AI to Ethernet MAC layer (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]~~M-AI to ETH adaptation function (~~xTSi[G]~~M-AI/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]~~M-AI access point (~~xTSi[G]_~~M-AP) is referred to as ~~xTSi[G]~~media layer adapted information (~~xTSi[G]_~~M-AI).

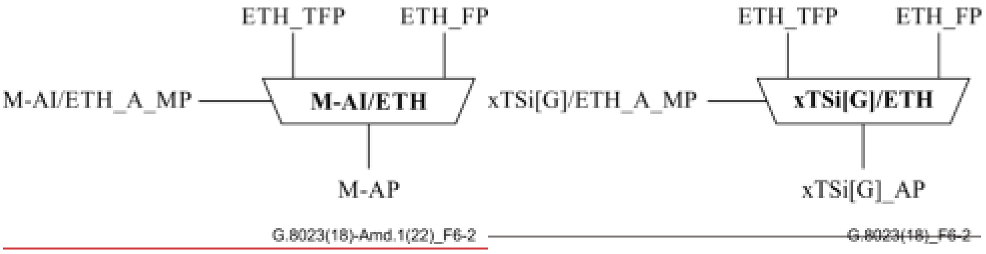


Figure 6-2 – ~~xTSi[G]~~M-AI to ETH adaptation function

6.1.1 ~~xTSi[G]~~M-AI to ETH adaptation source function (~~xTSi[G]~~M-AI/ETH_A_So)

Symbol

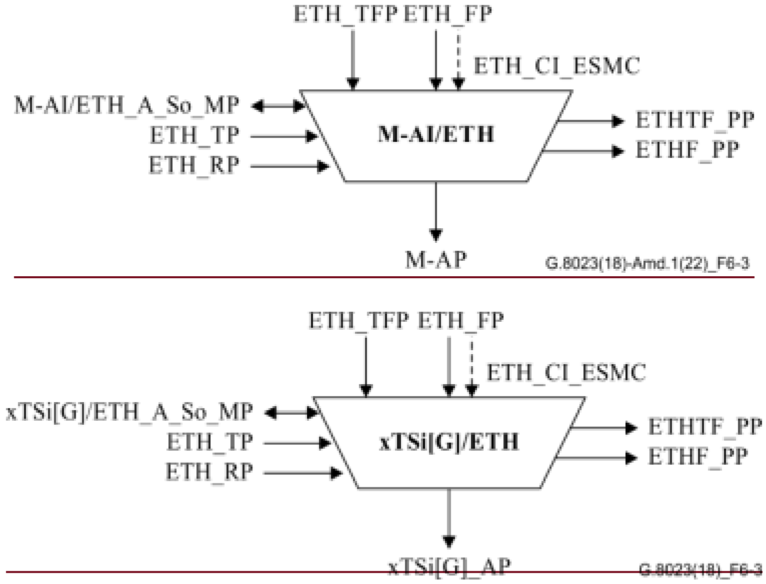


Figure 6-3 – ~~xTSi[G]~~M-AI/ETH_A_So symbol

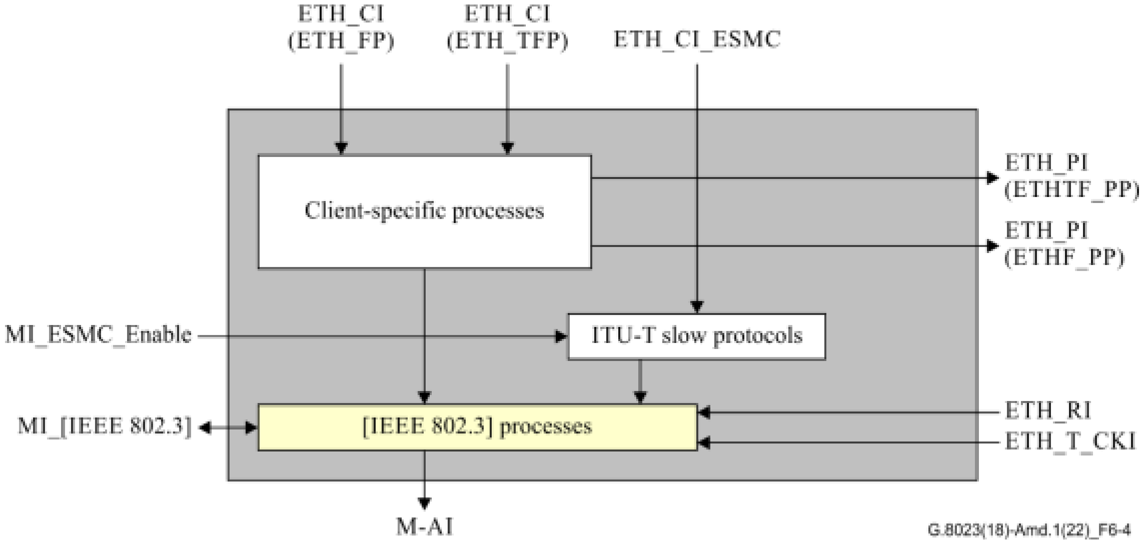
Interfaces

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

Inputs	Outputs
ETH_TFP: ETH_CI_D ETH_CI_P ETH_CI_DE	**~~xTSi[G]_~~M-AP**: ~~xTSi[G]_~~M-AI_D[1..N] (Note)
ETH_FP: ETH_CI_D ETH_CI_P ETH_CI_DE ETH_CI_SSF ETH_CI_SSFrdi ETH_CI_SSFfdi ETH_CI_ESMC	**ETHTF_PP**: ETH_PI_D ETH_PI_P ETH_PI_DE
ETH_TP: ETH_TI_CK	**ETHF_PP**: ETH_PI_D ETH_PI_P ETH_PI_DE
~~xTSi[G]~~M-AI/ETH_A_RP: ~~xTSi[G]~~M-AI/ETH_A_RI_RSF	**~~xTSi[G]~~M-AI/ETH_A_RP**: ~~xTSi[G]~~M-AI/ETH_A_RI_FTS
~~xTSi[G]~~M-AI/ETH_A_So_MP: ~~xTSi[G]~~M-AI/ETH_A_So_MI_[IEEE 802.3] ~~xTSi[G]~~M-AI/ETH_A_So_MI_FTSEnable M-AI/ETH_A_So_MI_ESMC_Enable	**~~xTSi[G]~~M-AI/ETH_A_So_MP**: ~~xTSi[G]~~M-AI/ETH_A_So_MI_[IEEE 802.3]
NOTE – When the PHY has only one OTSi, there is only one ~~OTSi_~~M-AI_D signal.	

Processes

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



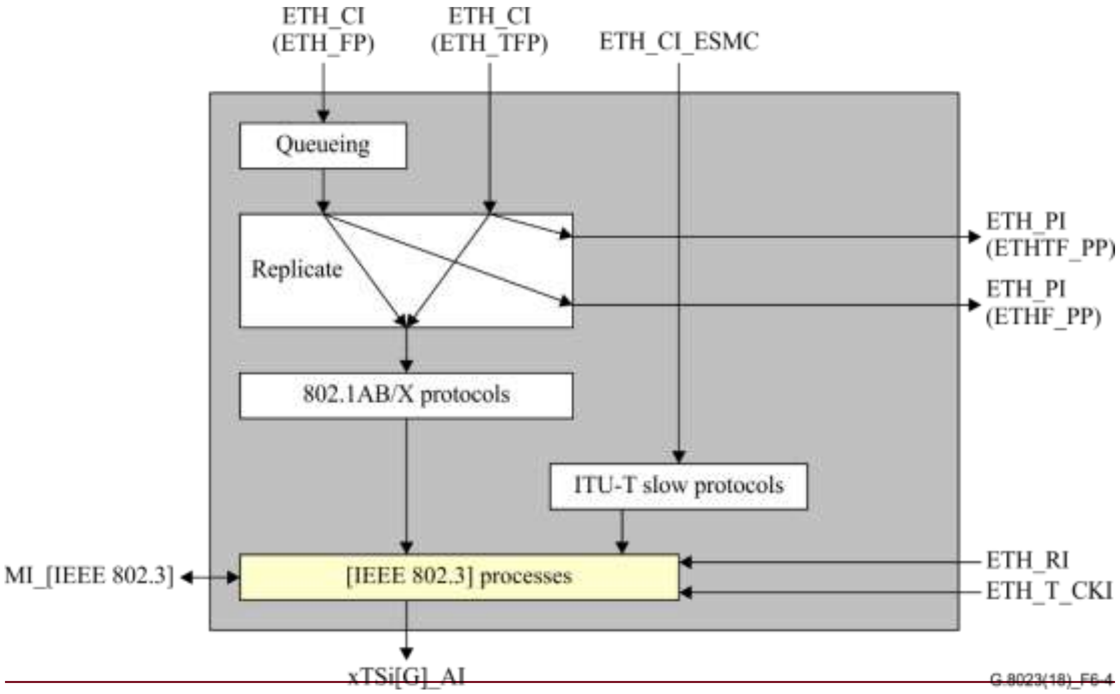


Figure 6-4 – ~~xTSi[G]~~M-AI/ETH_A_So process

Processes

The ~~queueing, replicate, and 802.1AB/X protocols~~client-specific processes, and associated MI and PI signals, are defined in clause ~~8~~ 9.5 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 protocol data unit (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. Use of the ESMC is configured via MI_ESMC_Enable.

"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 1 – 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, frame check sequence (FCS) generation, generation of 802.3 slow protocols, and multiplexing of organization-specific and 802.3 slow protocols.

NOTE 2 – 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]~~M-AI to ETH adaptation sink function (~~xTSi[G]~~M-AI/ETH_A_Sk)

Symbol

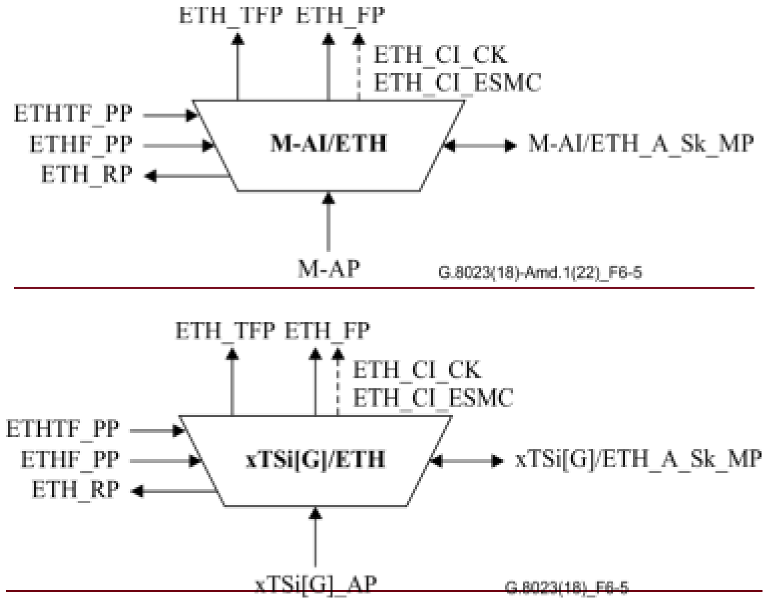


Figure 6-5 – M-AI~~xTSi[G]~~/ETH_A_Sk symbol

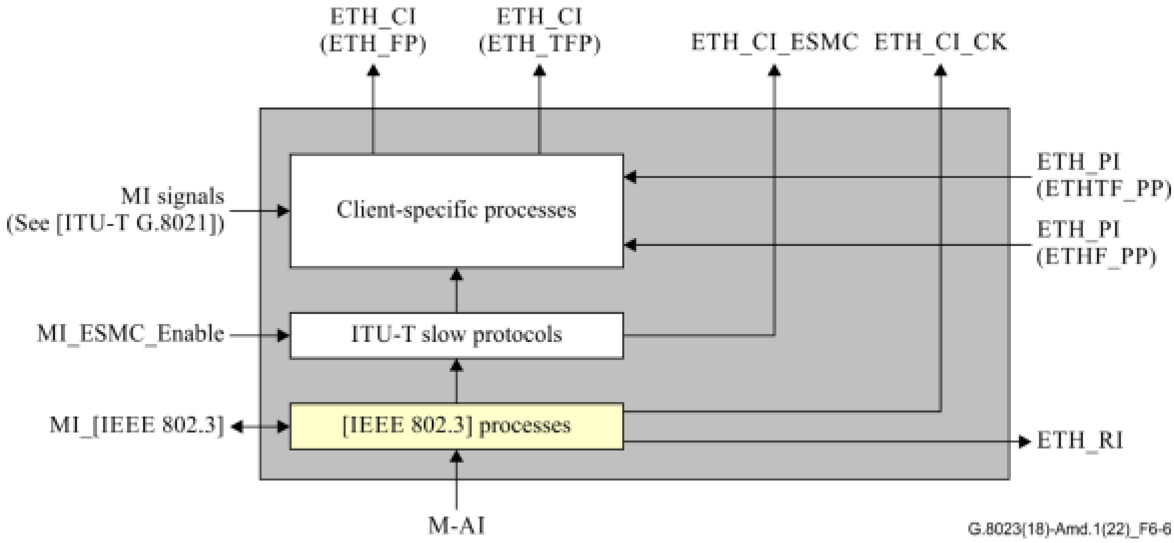
Interfaces

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

Inputs	Outputs
~~xTSi[G]_~~M-AP: ~~xTSi[G]_~~M-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]~~M-AI/ETH_A_RP**: ~~xTSi[G]~~M-AI/ETH_A_RI_FTS **~~xTSi[G]~~M-AI/ETH_A_Sk_MP**: ~~xTSi[G]~~M-AI/ETH_A_Sk_MI_[IEEE 802.3] ~~xTSi[G]/ETH_A_Sk_MI_FilterConfig~~ M-AI/ETH_A_Sk_MI_ESMC_Enable	**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]~~M-AI/ETH_A_RP**: ~~xTSi[G]~~M-AI/ETH_A_RI_RSF **~~xTSi[G]~~M-AI/ETH_A_Sk_MP**: ~~xTSi[G]~~M-AI/ETH_A_Sk_MI_[IEEE 802.3]
NOTE – When the PHY has only one OTSi, there is only one ~~OTSi_~~M-AI_D signal.	

Processes

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



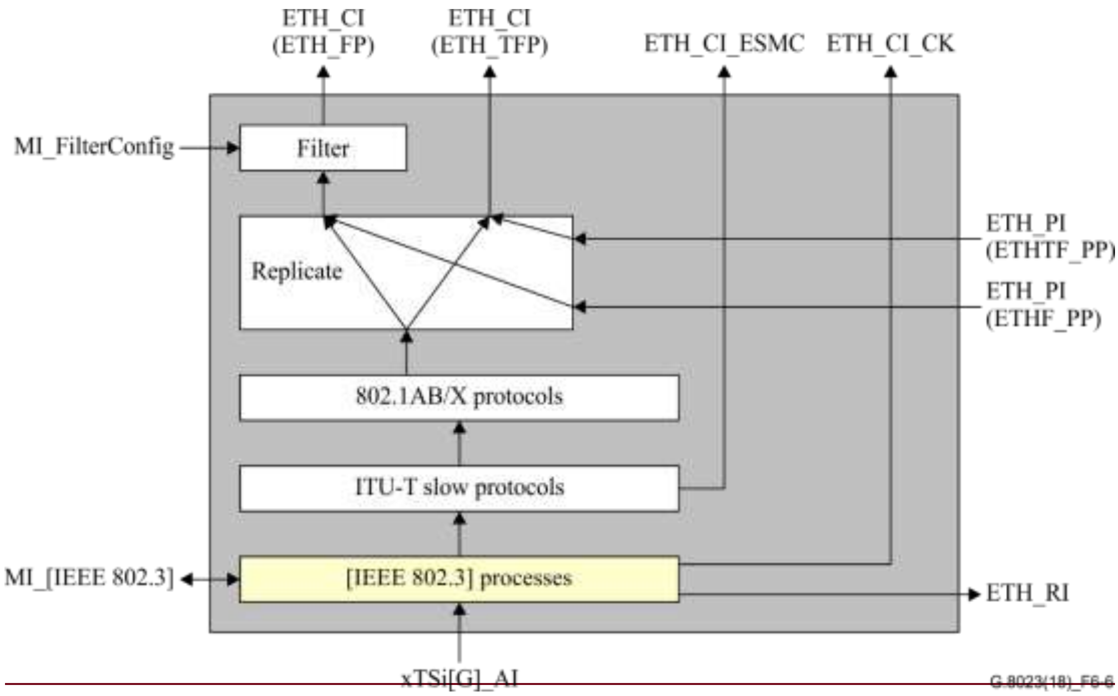


Figure 6-6 – ~~xTSi[G]~~M-AI/ETH_A_Sk process

The ~~filter, replicate, and 802.1AB/X protocols~~client-specific processes, and associated MI and PI signals, are defined in clause ~~8~~ 9.5 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 traffic 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: ESMC as defined in [ITU-T G.8264]. Use of the ESMC is configured via MI_ESMC_Enable.

Defects

The definition of defects used by xTSi[G]M-AI/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]M-AI/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 M-AI to 10G Ethernet reconciliation sublayer adaptation functions (OTSiM-AI/ERS10G_A)

The OTSi M-AI 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~~ M-AI to 10G Ethernet reconciliation sublayer source adaptation function (~~OTSi~~M-AI/ERS10G_A_So)

Symbol

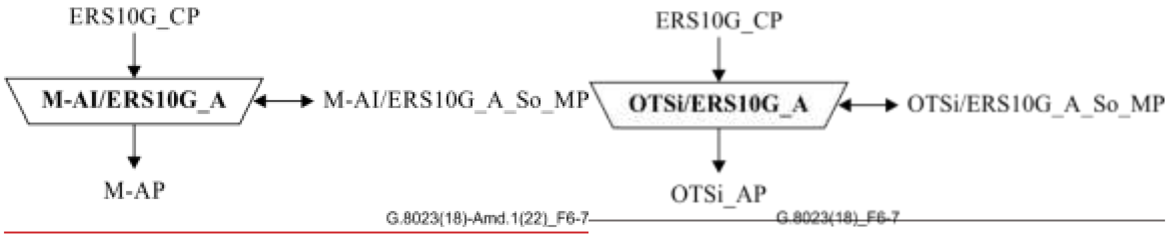


Figure 6-7 – ~~OTSi~~M-AI/ERS10G_A_So function

Interfaces

Table 6-3 – ~~OTSi~~M-AI/ERS10G_A_So inputs and outputs

Input(s)	Output(s)
ERS10G_CP: ERS10G_CI_D **~~OTSi~~M-AI/ERS10G_A_So_MP**: ~~OTSi~~M-AI/ERS10G_A_So_MI_[IEEE 802.3]	**~~OTSi_~~M-AP**: ~~OTSi_~~M-AI_D **~~OTSi~~M-AI/ERS10G_A_So_MP**: ~~OTSi~~M-AI/ERS10G_A_So_MI_[IEEE 802.3]

Processes

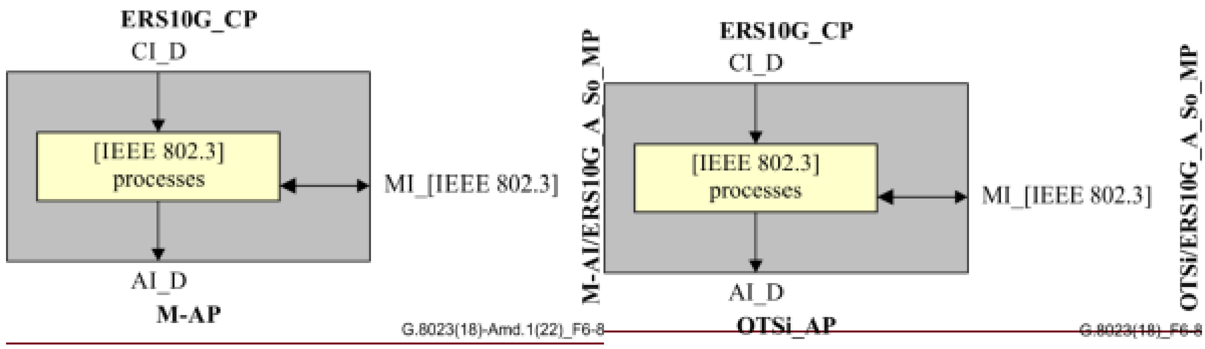


Figure 6-8 – ~~xTSi~~M-AI/ERS10G_A_So processes

[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~~ M-AI to 10G Ethernet reconciliation sublayer sink adaptation function (~~OTSi~~M-AI/ERS10G_A_Sk)

Symbol

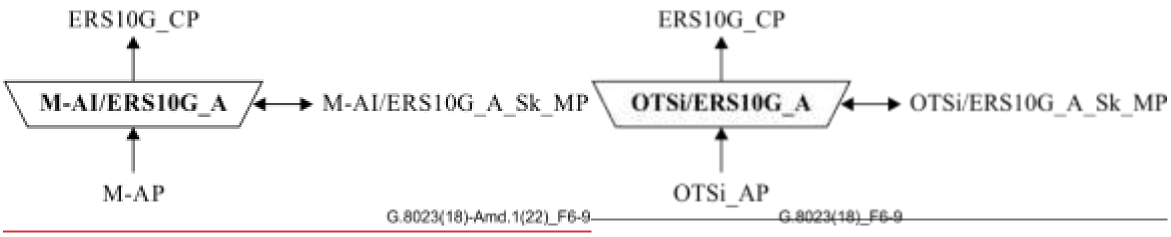


Figure 6-9 – ~~OTSi~~M-AI/ERS10G_A_Sk function

Interfaces

Table 6-4 – ~~OTSi~~M-AI/ERS10G_A_Sk inputs and outputs

Input(s)	Output(s)
~~OTSi~~ M-AP: ~~OTSi~~ M-AI_D	**ERS10G_CP**: ERS10G_CI_D ERS10G_CI_SSF
~~OTSi~~M-AI/ERS10G_A_Sk_MP: ~~OTSi~~M-AI/ERS10G_A_Sk_MI_[IEEE 802.3]	**~~OTSi~~M-AI/ERS10G_A_Sk_MP**: ~~OTSi~~M-AI/ERS10G_A_Sk_MI_[IEEE 802.3]

Processes

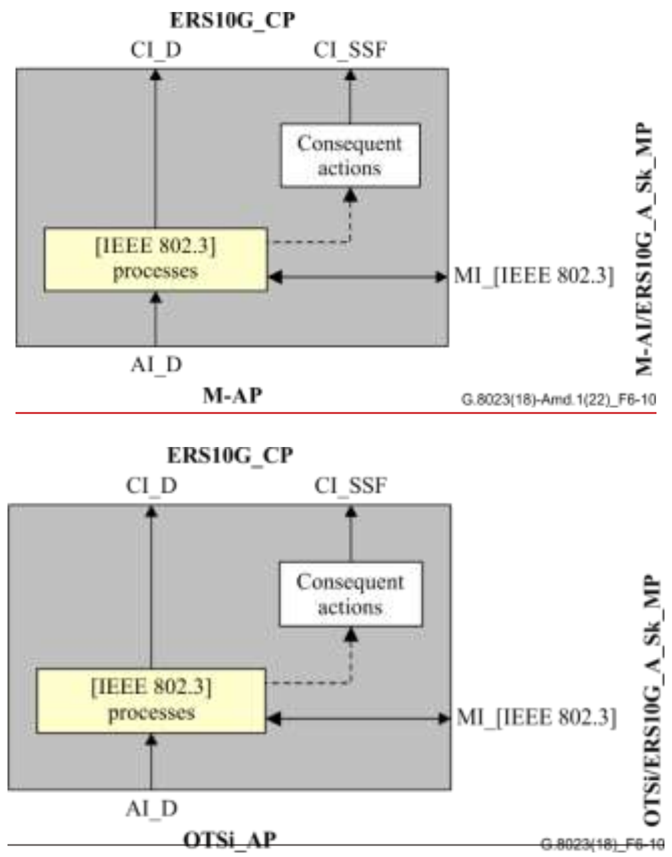


Figure 6-10 – ~~OTSi~~M-AI/ERS10G_A_Sk processes

[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~~M-AI/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~~ERS10G_A_So defined in 14.3.3 of [ITU-T G.798].

Defect correlations: None.

Performance monitoring: None.

6.3 ~~xTSi[G]~~M-AI to Ethernet coding adaptation functions (~~xTSi[G]~~M-AI/ETCy_A)

The set of ~~xTSi[G]~~M-AI to ETCy (y=1000X, 25GR, 40GR, 50GR, 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.

These functions are defined for the following Ethernet interfaces: 1000BASE-X, 25GBASE-R, 40GBASE-R, 50GBASE-R, 100GBASE-R, 200GBASE-R, 400GBASE-R. The value 'y' is formed by deleting "BASE-" from the name of the Ethernet interface (e.g., the coding sublayer for 100GBASE-R is named 100GR). ~~following specific functions are defined:~~

~~OTSi/ETC1000X_A~~

~~xTSi/ETC25GR_A~~

~~xTSi[G]/ETC40GR_A~~

~~xTSiG/ETC100GR_A~~

~~OTSiG/ETC200GR_A~~

~~OTSiG/ETC400GR_A~~

ETCy characteristic information

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 802.3~~bs~~].

~~The ETC1000X_CI_D traffic units carry 8B/10B codewords of the 1000BASE-X physical coding sublayer (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 ETC50GR_CI_D traffic units carry the 64B/66B codewords of the 50GBASE-R PCS, as defined in clause 133 of [IEEE 802.3cd].

~~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 physical medium attachment (PMA) sublayers as shown in Figure 34-1 of [IEEE 802.3]. The ETC1000X_CI_D traffic units carry 8B/10B codewords of the 1000BASE-X physical coding sublayer (PCS), as defined in clause 36 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 802.3]~~[IEEE 802.3by]~~. The ETC25GR_CI_D traffic units carry the 64B/66B codewords of the 25GBASE-R PCS, as defined in clause 49 and clause 107 of [IEEE 802.3].

For 40GBASE-R interfaces, the ETC40GR_CP reference point is within the 40GBASE-R PCS. For 100GBASE-R interfaces, the ETC100GR_CP reference point is within the 100GBASE-R PCS. With respect to Figure 82~~.~~-2 of [IEEE 802.3], in the source function, the reference point is below the Alignment insertion process; in the sink function, it is between the alignment lock/lane deskew and lane reorder processes. The ETC40GR_CI_D traffic units and ETC100GR_CI_D traffic units carry the 64B/66B codewords of the 40GBASE-R or 100GBASE-R PCS, respectively, as defined in clause 82 of [IEEE 802.3].

For 50GBASE-R interfaces, the ETC50GR_CP reference point is within the 50GBASE-R PCS. With respect to Figure 133-2 of [IEEE 802.3cd], in the source function, the reference point is below the alignment insertion process; in the sink function, it is between the alignment lock/lane deskew and lane reorder processes.

~~For 100GBASE-R interfaces, the ETC100GR_CP reference point is within the 100GBASE-R PCS. With respect to Figure 82.2 of [IEEE 802.3], in the source function, the reference point is below the alignment insertion process; in the sink function, it is between the alignment lock/lane deskew and lane reorder processes.~~

For 200GBASE-R interfaces, the ETC200GR_CP reference point is within the 200GBASE-R PCS, between the encode/decode and rate matching process and the transcode/reverse transcode process, including the FEC_degraded_SER and rx_local_degraded bits; see Figure 119-2 and clause 119.2.4.1 of [IEEE 802.3]~~[IEEE 802.3bs]~~.

For 400GBASE-R interfaces, the ETC400GR_CP reference point is within the 400GBASE-R PCS, between the encode/decode and rate matching process and the transcode/reverse transcode process, including the FEC_degraded_SER and rx_local_degraded bits; see Figure 119-2 ~~of~~ and clause 119.2.4.1 of [IEEE 802.3]~~[IEEE 802.3bs]~~. 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 802.3].

6.3.1 ~~xTSi[G]~~M-AI to Ethernet coding adaptation source function (~~xTSi[G]~~M-AI/ETCy_A_So)

Symbol

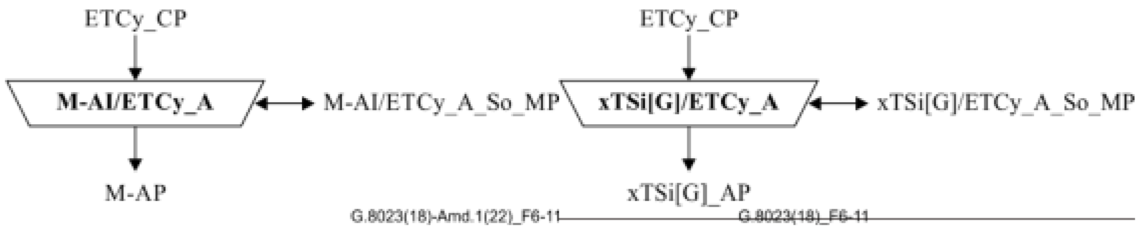


Figure 6-11 – ~~xTSi[G]~~M-AI/ETCy_A_So function

Interfaces

Table 6-5 – ~~xTSi[G]~~M-AI/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]~~M-AI/ETCy_A_So_MP**: ~~xTSi[G]~~M-AI/ETCy_A_So_MI_[IEEE 802.3]	**~~xTSi[G]_~~M-AP**: ~~xTSi_~~M-AI_D[1..N] (Note) **~~xTSi[G]~~M-AI/ETCy_A_So_MP**: ~~xTSi[G]~~M-AI/ETCy_A_So_MI_[IEEE 802.3]
NOTE – When the PHY has only one OTSi, there is only one ~~OTSi_~~M-AI_D signal.	

Processes

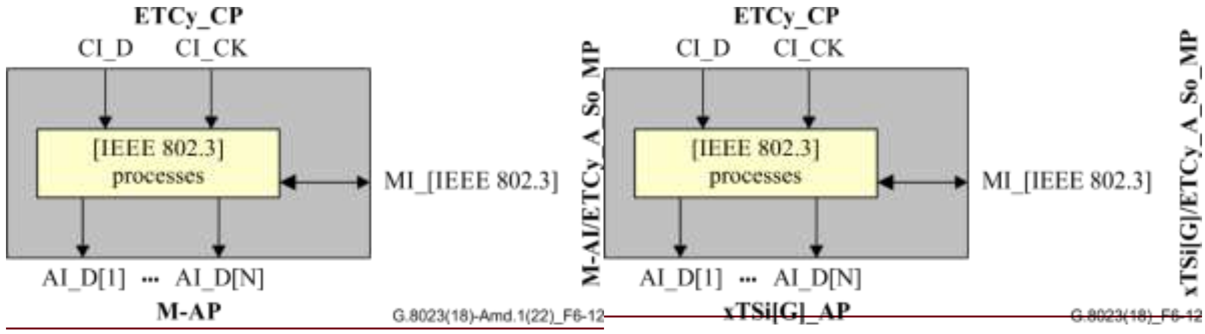


Figure 6-12 – ~~xTSi[G]~~M-AI/ETCy_A_So processes

[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]~~M-AI to Ethernet coding adaptation sink function (~~xTSi[G]~~M-AI/ETCy_A_Sk)

Symbol

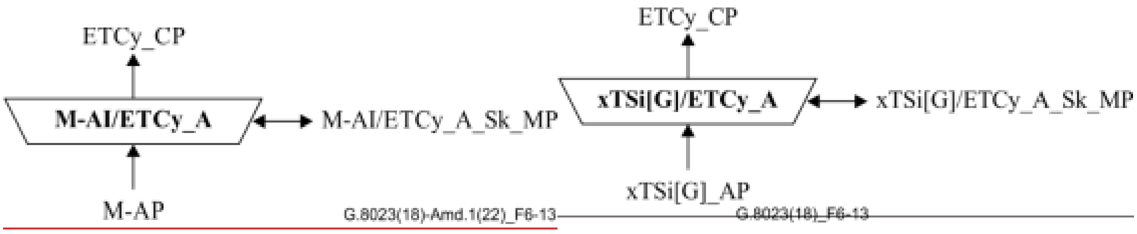


Figure 6-13 – ~~xTSi[G]~~M-AI/ETCy_A_Sk function

Interfaces

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

Input(s)	Output(s)
~~xTSi[G]~~_M-AP: ~~xTSi~~_M-AI_D[1..N] (Note) **~~xTSi[G]~~M-AI/ETCy_A_Sk_MP**: ~~xTSi[G]~~M-AI/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]~~M-AI/ETCy_A_Sk_MP**: ~~xTSi[G]~~M-AI/ETCy_A_Sk_MI_[IEEE 802.3]
NOTE – When the PHY has only one OTSi, there is only one ~~OTSi~~_M-AI_D signal.	

Processes

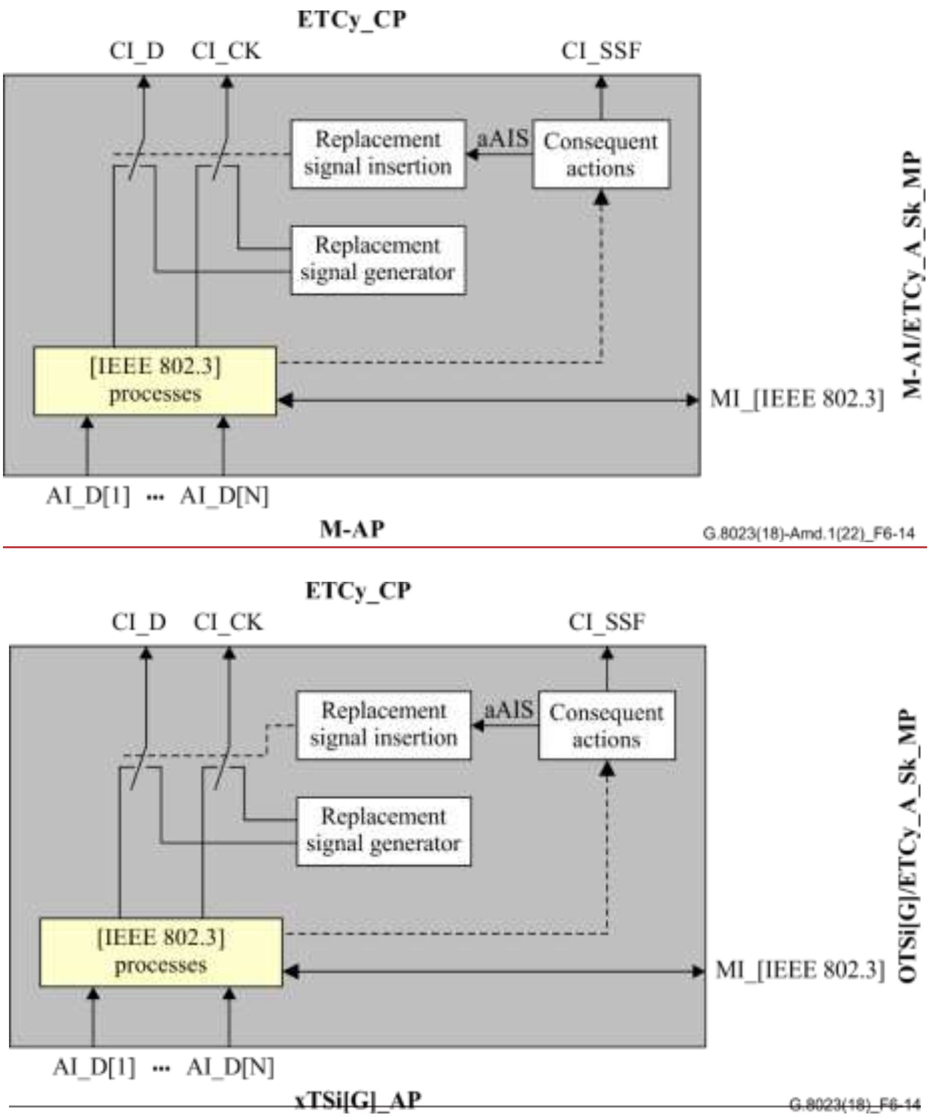


Figure 6-14 – ~~xTSi[G]~~M-AI/ETCy_A_Sk processes

[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 1 – 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]~~M-AI/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 2 – Further details are intentionally left outside the scope of this Recommendation.

On declaration of aAIS, the function shall output a replacement signal as specified in Table 6-7.

Table 6-7 – Replacement signals for ETCy clients

Client	Replacement signal defined in
1000BASE-X	Clause 17.7.1.1 of [ITU-T G.709]
25GBASE-R	Clause 17.13 of [ITU-T G.709]
40GBASE-R	Clause 17.7.4 of [ITU-T G.709]
50GBASE-R	Clause 17.13 of [ITU-T G.709]
100GBASE-R	Clause 17.7.5 of [ITU-T G.709]
200GBASE-R	Clause 17.13 of [ITU-T G.709]
400GBASE-R	Clause 17.13 of [ITU-T G.709]

~~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~~ M-AI to CBR10G3 adaptation functions (~~OTSi~~M-AI/CBR10G3_A)

The ~~OTSi~~ M-AI 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].

CBR10G3 characteristic information

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~~ M-AI to CBR10G3 adaptation source function (~~OTSi~~M-AI/CBR10G3_A_So)

Symbol

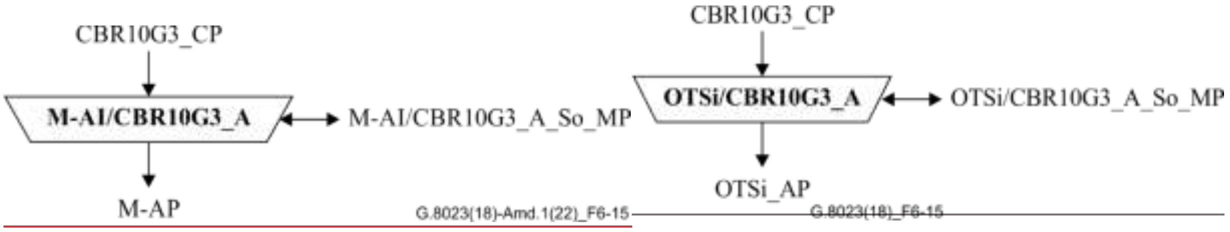


Figure 6-15 – ~~OTSi~~M-AI/CBR10G3_A_So function

Interfaces

Table 6-~~7~~ 8 – ~~OTSi~~M-AI/CBR10G3_A_So inputs and outputs

Input(s)	Output(s)
CBR10G3_CP: CBR10G3_CI_CK CBR10G3_CI_D **~~OTSi~~M-AI/CBR10G3_A_So_MP:** ~~OTSi~~M-AI/CBR10G3_A_So_MI_[IEEE 802.3]	**~~OTSi~~ M-AP**: ~~OTSi~~ M-AI_D **~~OTSi~~M-AI/CBR10G3_A_So_MP:** ~~OTSi~~M-AI/CBR10G3_A_So_MI_[IEEE 802.3]

Processes

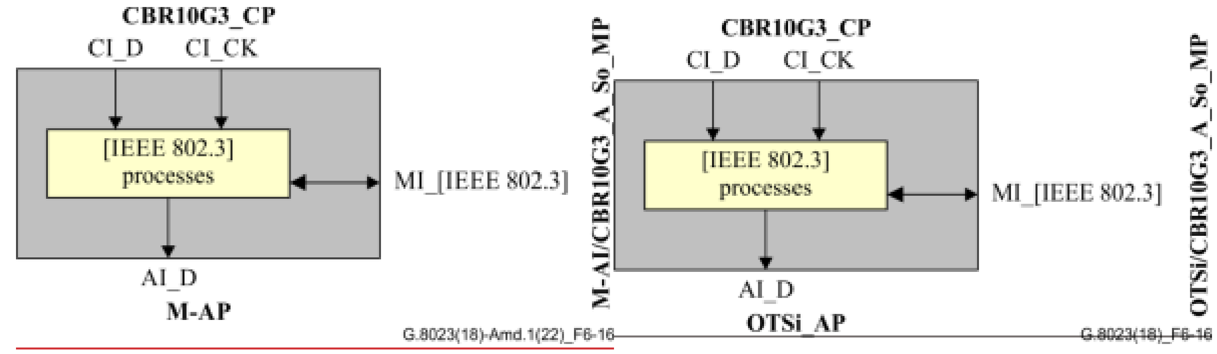


Figure 6-16 – ~~xTSi(G)~~M-AI/CBR10G3_A_So processes

[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~~ M-AI to CBR10G3 adaptation sink function (~~OTSi~~M-AI/CBR10G3_A_Sk)

Symbol

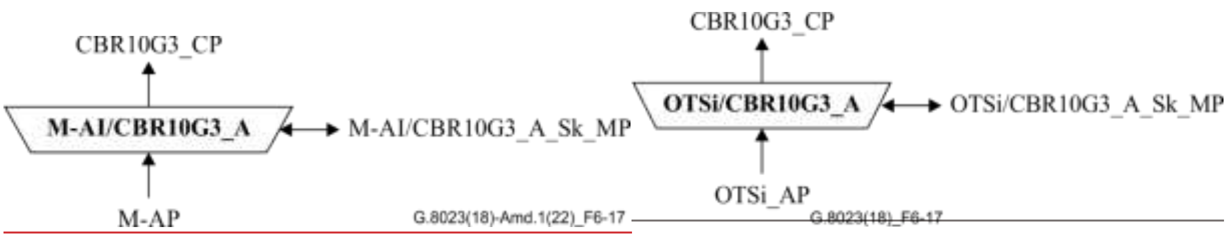


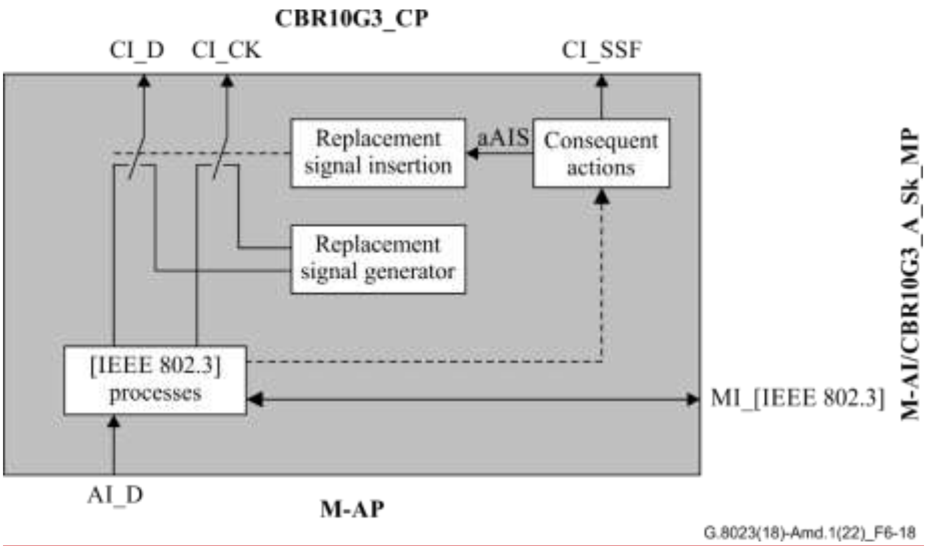
Figure 6-17 – ~~OTSi~~M-AI/CBR10G3_A_Sk function

Interfaces

Table 6-8 9 – OTSiM-AI/CBR10G3_A_Sk inputs and outputs

Input(s)	Output(s)
OTSi M-AP: OTSi M-AI_D	**CBR10G3_CP**: CBR10G3_CI_CK CBR10G3_CI_D CBR10G3_CI_SSF
OTSiM-AI/CBR10G3_A_Sk_MP: OTSiM-AI/CBR10G3_A_Sk_MI_[IEEE 802.3]	**OTSiM-AI/CBR10G3_A_Sk_MP:** OTSiM-AI/CBR10G3_A_Sk_MI_[IEEE 802.3]

Processes



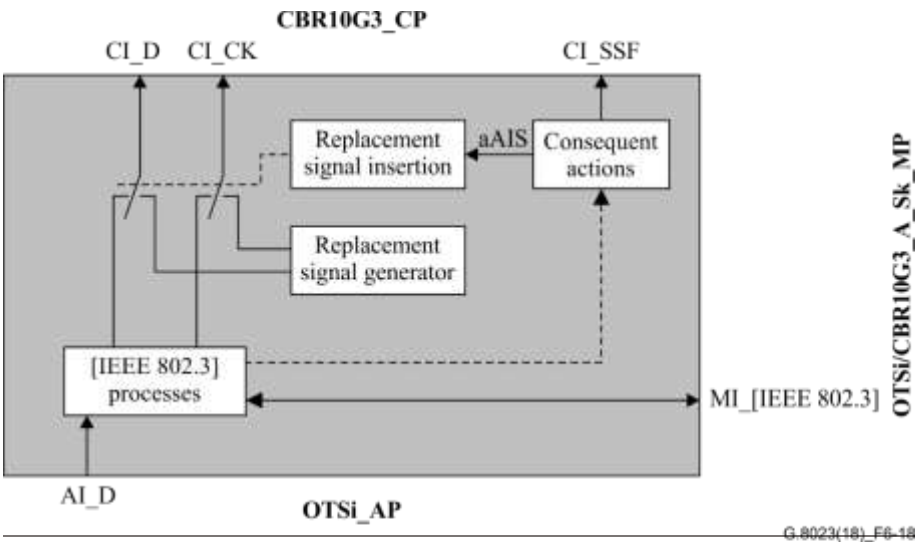


Figure 6-18 – OTSiM-AI/CBR10G3_A_Sk processes

[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 1 – 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~~M-AI/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 2 – 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]~~M-AI to Flex Ethernet (FlexE) adaptation functions (~~OTSi[G]~~M-AI/FlexE_A)

The ~~OTSiG~~ M-AI 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~~ M-AI to FlexE adaptation source function (~~OTSiG~~M-AI/FlexE_A_So)

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

Symbol

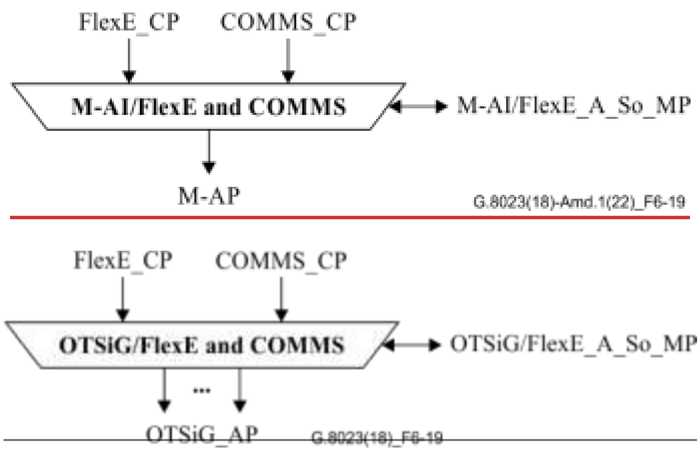


Figure 6-19 – ~~OTSiG~~M-AI/FlexE_A_So function

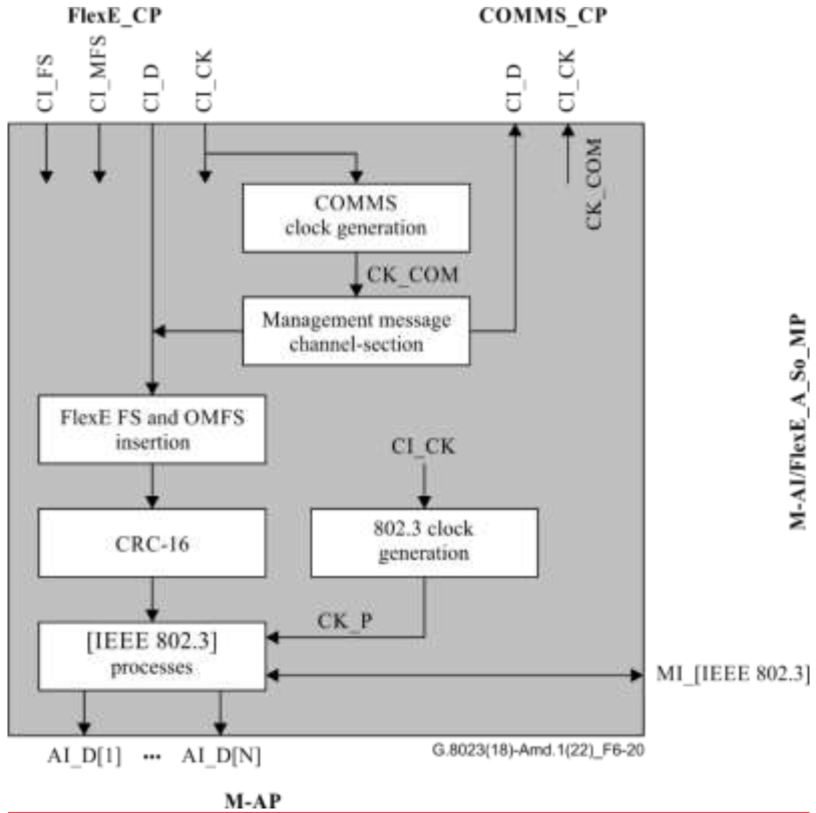
Interfaces

Table 6-9 10 – OTSiGM-AI/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 **M-AI/FlexE_A_So_MP**: M-AI/FlexE_A_So_MI_[IEEE 802.3]	**OTSiG_M-AP**: OTSiG_M-AI_D[1..N] **COMMS_CP**: COMMS_CI_CK **OTSiGM-AI/FlexE_A_So_MP**: OTSiGM-AI/FlexE_A_So_MI_[IEEE 802.3]

Processes

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



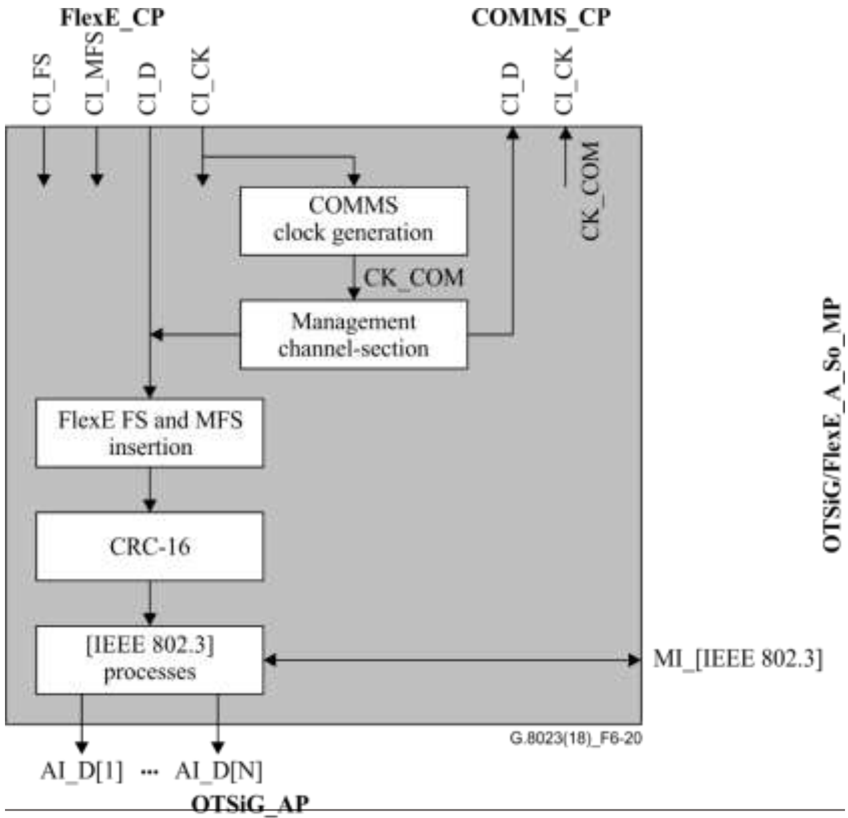


Figure 6-20 – ~~OTSiG~~M-AI/FlexE_A_So processes

Communications channel (COMMS) clock (CK) generation: The ~~function shall generate the~~ COMMS clock (CK_COM) is generated 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 and OMFS insertion: The function shall insert the FlexE frame alignment signal (i.e., frame start (FS)) and FlexE overhead multiframe alignment signal (i.e., multiframe start (MFS)) is inserted 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.**OMFS insertion**: The FlexE overhead multiframe alignment signal (i.e., multiframe start (MFS)) is inserted as described in clause 7.3.1 of [OIF FLEXE IA].

Cyclic redundancy check (CRC)-16: See clause 7.3.9 of [OIF FLEXE IA]. The function shall compute the CRC-16 is computed and insert the calculated CRC-16 valuethe value is inserted into the CRC-16 byte of FlexE overhead field.

FlexE FS insertion: The FlexE frame alignment signal (i.e., frame start (FS)) is inserted as described in clause 7.3.1 of [OIF FLEXE IA]. The FlexE frame alignment signal is contained in the first block of the FlexE overhead, and is defined by the pattern "0x01" in bits 0 to 1 (i.e., the sync header), the pattern "0x4B" in bits 0 2 to 7 9 (i.e., the control block type) and the pattern 0x5 in bits 32 34 to 3537 (i.e., the "O" code).**FlexE FS insertion**: The FlexE frame alignment signal (i.e., frame start (FS)) is inserted as described in clause 7.3.1 of [OIF FLEXE IA]. The FlexE frame alignment signal is contained in the first block of the FlexE overhead, and is defined by the pattern "0x01" in bits 0 to 1 (i.e. the sync header), the pattern "0x4B" in bits 2 to 9 (i.e. the control block type) and the pattern 0x5 in bits 34 to 37 (i.e., the "O" code).

802.3 clock generation: The 802.3 clock is generated from the FlexE clock.

[IEEE 802.3] processes: The [IEEE 802.3] processes represent the whole functionality of the PCS sublayer below the FlexE 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 M-AI to FlexE adaptation sink function (OTSiGM-AI/FlexE_A_Sk)

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

Symbol

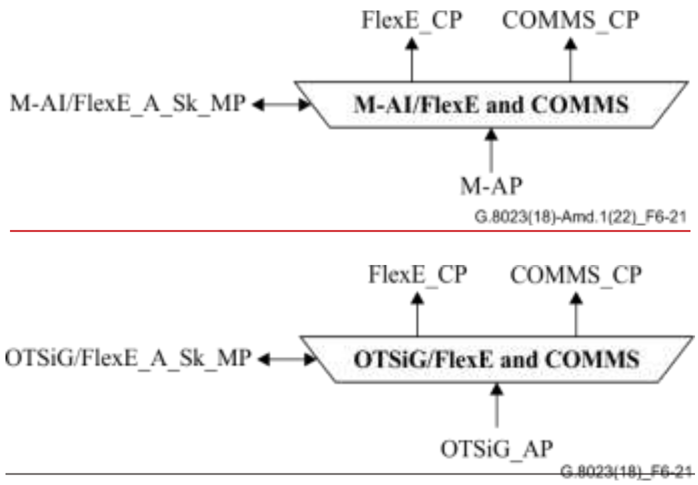


Figure 6-21 – ~~OTSiG~~M-AI/FlexE_A_Sk function

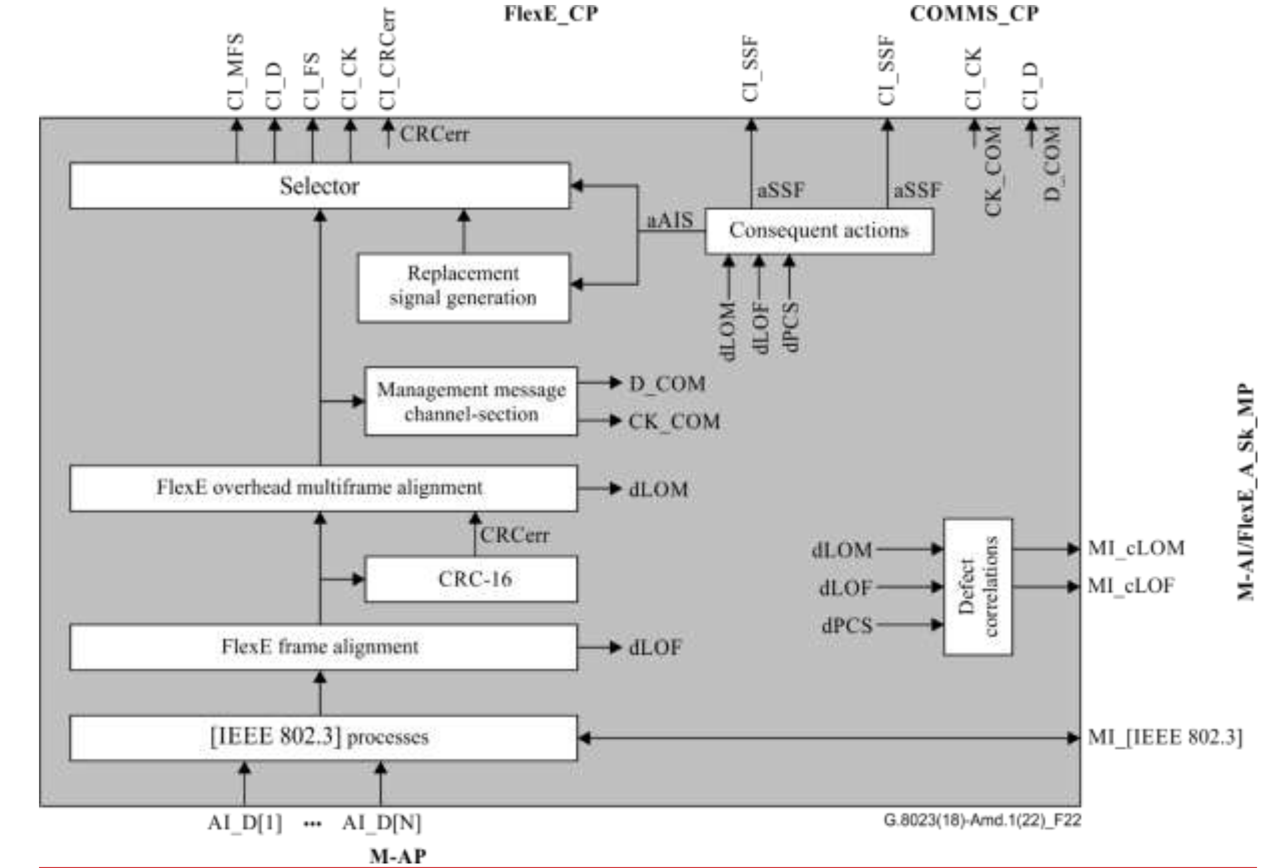
Interfaces

Table 6-~~10~~ 11 – ~~OTSiG~~M-AI/FlexE_A_Sk inputs and outputs

Input(s)	Output(s)
~~OTSiG~~ M-AP: ~~OTSiG~~ M-AI_D[1..N] **~~OTSiG~~M-AI/FlexE-A_Sk_MP**: M-AI/FlexE_A_Sk_MI_[IEEE 802.3] ~~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~~M-AI/FlexE_A_Sk_MP**: ~~OTSiG~~M-AI/FlexE_A_Sk_MI_[IEEE 802.3] ~~OTSiG~~M-AI/FlexE_A_Sk_MI_cLOF ~~OTSiG~~M-AI/FlexE_A_Sk_MI_cLOM

Processes

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



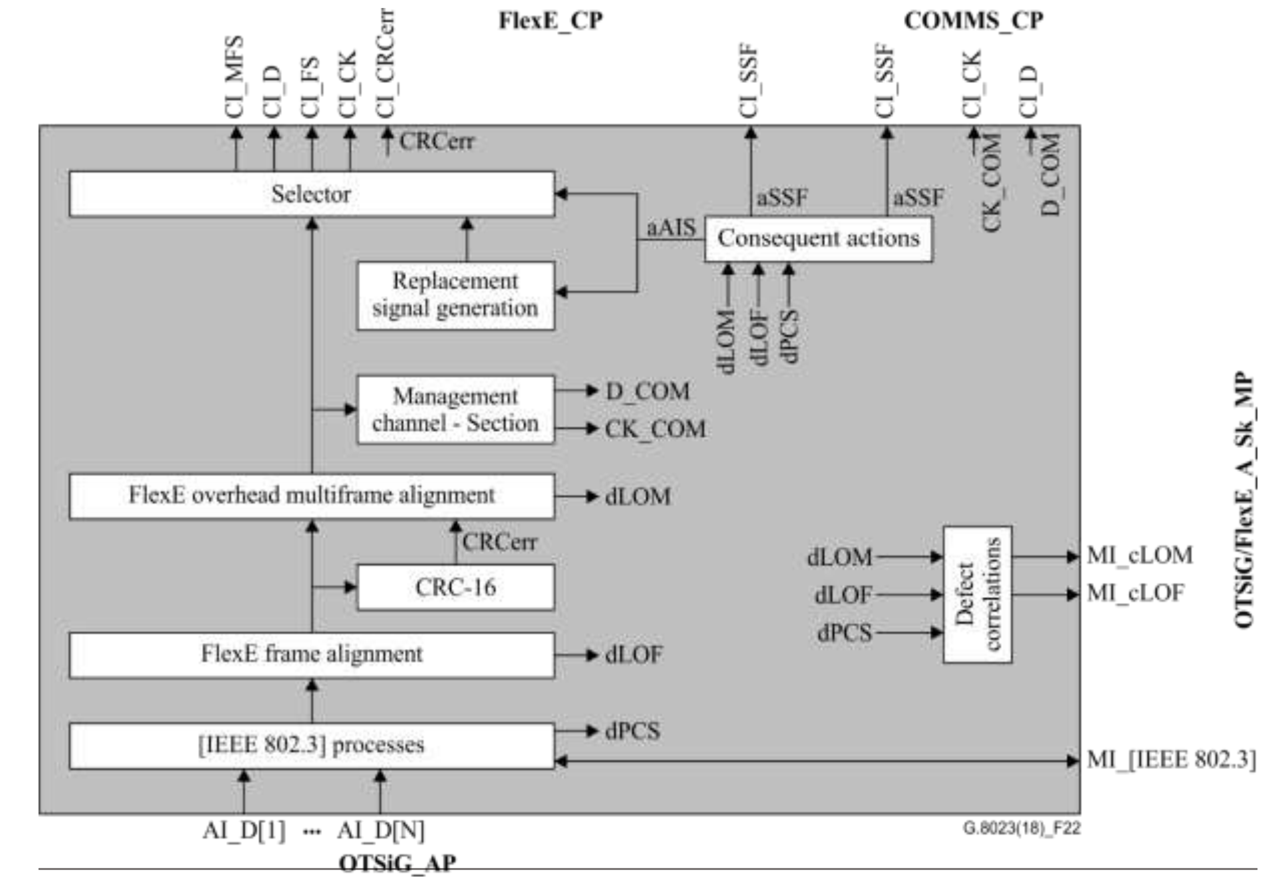


Figure 6-22 – ~~OTSiG~~M-AI/FlexE_A_Sk processes

[IEEE 802.3] processes~~[IEEE 802.3 processes]~~: The [IEEE 802.3] processes represent the whole functionality of the PCS sublayer below the FlexE 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 is recovered through recognizing ~~the FlexE~~ block 1 of the FlexE overhead frame as described in clause 7.3.1 of [OIF FLEXE IA]. ~~The FlexE b~~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 is present once per (1023 × 20 + 1) × 8 blocks. The specific process is described in Annex B.2.1.1. ~~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, out-of-multiframe (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 five 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~~is divisible by the expected polynomial, CRCerr is set to 0 ~~(the default value)~~; otherwise, CRCerr is set to 1.

FlexE overhead multiframe alignment: The ~~function shall recover the~~ FlexE overhead multi-frame alignment is found based on the overhead multi-frame (OMF) bit in the FlexE overhead. ~~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 clause B.2.1.2 ~~of [ITU-T G.798]~~.

Management channel – section: The management channel is ~~for user specific and~~optional. When used, 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~~A FlexE replacement signal (including clock, frame start, and multi-frame start) ~~and clock~~ generation process is provided that generates a stream of local fault sequence ordered sets as specified in clause 5.2.2.3 of [OIF FLEXE IA]. The FlexE replacement signal clock, frame start, and multi-frame start are independent from the incoming clock, frame start, and multi-frame start. The FlexE replacement signal clock must be within the frequency range $103.125\ \text{Gb/s} \times \frac{16383}{16384} \pm 100$.

Selector: The ~~function shall select the~~ normal FlexE signal or the replacement signal is selected. When aAIS is true, it shall select the replacement signals within two FlexE frames. Otherwise, it shall select the normal FlexE signals. On clearing aAIS, the FlexE replacement signal shall be removed within two FlexE frames, with normal data being output.

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 FlexE 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 clause B.1.1.1.1 ~~of [ITU-T 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 clause B.1.1.1.2 of [ITU-T 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 Gb/s $\times \frac{16383}{16384}$ ± 100 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 Gb/s $\times \frac{16383}{16384}$ ± 100 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.

6.6 M-AI to Flex Ethernet subgroup member (FlexESGM) function (M-AI/FlexESGM_A)

The M-AI to FlexE subgroup member adaptation function performs the adaptation between the media layer adapted information related with a PHY that is part of a FlexE group and the characteristic information of the FlexESGM signal that is used to support aware mapping over transport.

The FlexESGM characteristic information is the same as the FlexE_CI defined in clause 7, except that the CRC, multiframe (OMFS), and frame alignment are part of the CI.

6.6.1 M-AI to FlexE aware adaptation source function (M-AI/FlexESGM_A_So)

The information flow and processing of the M-AI/FlexESGM_A_So function is defined with reference to Figures 6-23 and 6-24.

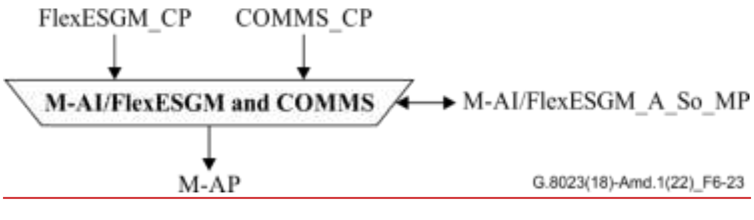


Figure 6-23 – M-AI/FlexESGM_A_So function

Interfaces

Table 6-12 – M-AI/FlexESGM_A_So inputs and outputs

Input(s)	Output(s)
FlexESGM_CP: FlexESGM_CI_CK FlexESGM_CI_D FlexESGM_CI_FS FlexESGM_CI_SSF **COMMS_CP**: COMMS_CI_D **M-AI/FlexESGM_A_So_MP**: M-AI/FlexESGM_A_So_MI_[IEEE 802.3]	**M-AP**: M-AI_D[1..N] **COMMS_CP**: COMMS_CI_CK **M-AI/FlexESGM_A_So_MP**: M-AI/FlexESGM_A_So_MI_[IEEE 802.3]

Processes

The processes associated with the M-AI/FlexESGM_A_So function are as depicted in Figure 6-24.

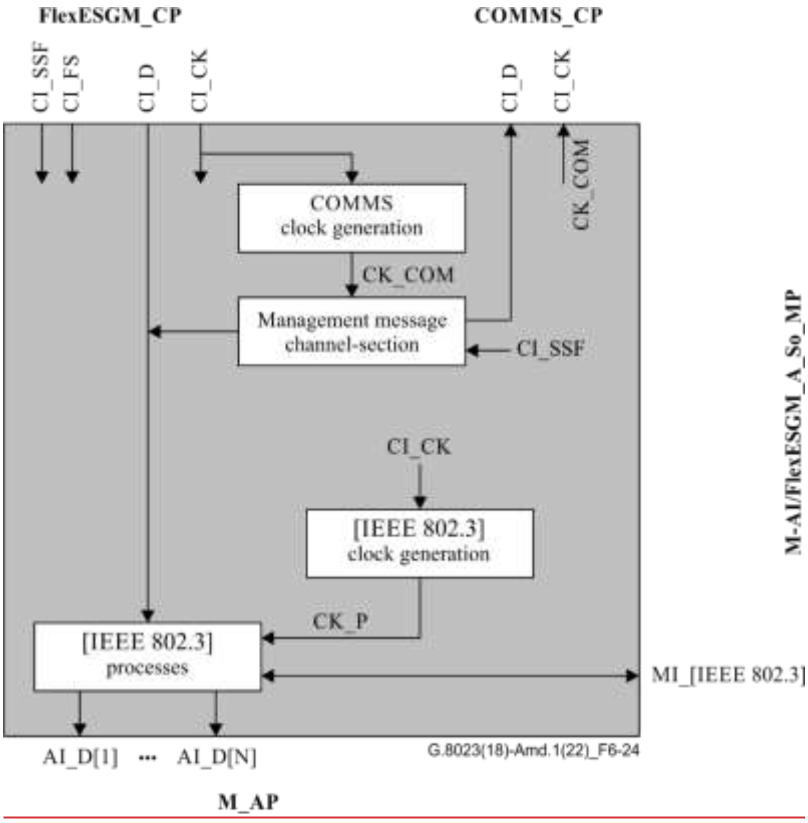


Figure 6-24 – M-AI/FlexESG_A_So processes

Communications channel (COMMS) clock (CK) generation: The COMMS clock (CK_COM) is generated by dividing the clock (CI_CK) by a factor of 81844.

Management channel – section: The management channel is optional. When used, if CI_SSF is false, 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 insertion: The FlexE frame alignment signal (i.e., frame start (FS)) is inserted as described in clause 7.3.1 of [OIF FLEXE IA]. The FlexE frame alignment signal is contained in the first block of the FlexE overhead, and is defined by the pattern 0x01 in bits 0 to 1 (i.e., the sync header), the pattern 0x4B in bits 2 to 9 (i.e., the control block type) and the pattern 0x5 in bits 34 to 37 (i.e., the "O" code).

802.3 clock generation: The 802.3 clock is generated from the FlexE clock.

[IEEE 802.3] processes: The [IEEE 802.3] processes represent the whole functionality of the PCS sublayer below the FlexE shim, as well as the PMA and physical medium dependent (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.

6.6.2 M-AI to FlexESGM adaptation sink function (M-AI/FlexESGM_A_Sk)

The information flow and processing of the M-AI/FlexESGM_A_Sk function is defined with reference to Figures 6-25 and 6-26.

Symbol

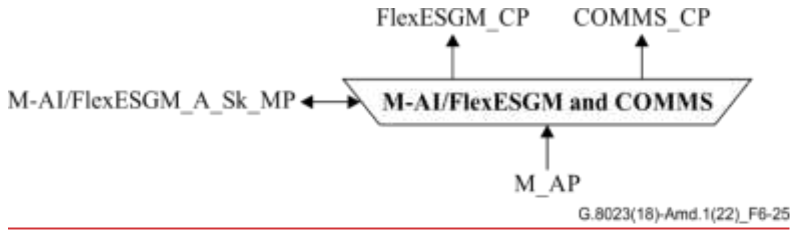


Figure 6-25 – M-AI/FlexESGM_A_Sk function

Interfaces

Table 6-13 – M-AI/FlexESGM_A_Sk inputs and outputs

Input(s)	Output(s)
M-AP: M-AI_D[1..N] **M-AI/FlexESGM_A_Sk_MP**: M-AI/FlexESGM_A_Sk_MI_[IEEE 802.3]	**FlexESGM_CP**: FlexESGM_CI_CK FlexESGM_CI_D FlexESGM_CI_FS FlexESGM_CI_MFS FlexESGM_CI_CRCerr FlexESGM_CI_SSF **COMMS_CP**: COMMS_CI_D COMMS_CI_CK COMMS_CI_SSF **M-AI/FlexESGM_A_Sk_MP**: M-AI/FlexESGM_A_Sk_MI_[IEEE 802.3] M-AI/FlexESGM_A_Sk_MI_cLOF M-AI/FlexESGM_A_Sk_MI_cLOM

Processes

The processes associated with the M-AI/FlexESGM_A_Sk function are as depicted in Figure 6-26.

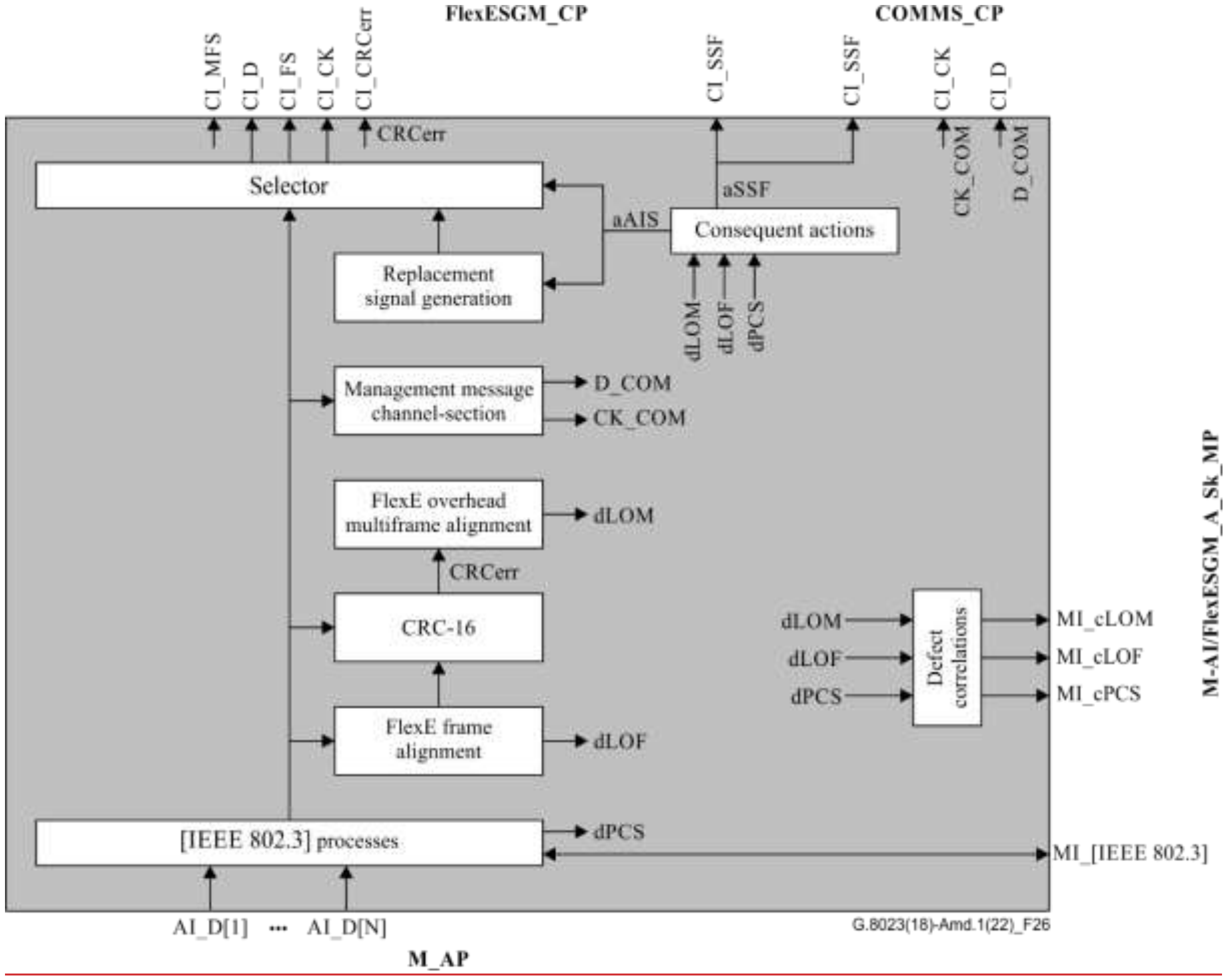


Figure 6-26 – M-AI/FlexESGM_A_Sk processes

[IEEE 802.3] processes: The [IEEE 802.3] processes represent the whole functionality of the PCS sublayer below the FlexE 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 FlexE frame start is recovered through non-intrusively monitoring the signal and recognizing block 1 of the FlexE overhead frame as described in clause 7.3.1 of [OIF FLEXE IA]. Block 1 of the FlexE overhead frame is encoded as a special ordered set (the sync header is 0x01, the control block type is 0x4B (ordered set), and the "O" code is 0x5) and is present once per (1023 × 20 + 1) × 8 blocks. The specific process is described in Annex B.2.1.1.

CRC-16: See clause 7.3.9 of [OIF FLEXE IA]. The CRC-16 field is non-intrusively monitored to enable multiframe alignment to be performed. If the CRC-16 value is divisible by the expected polynomial, CRCerr is set to 0; otherwise, CRCerr is set to 1.

FlexE overhead multiframe alignment: The FlexE overhead multi-frame alignment is found based on non-intrusively monitoring the overhead multi-frame (OMF) bit in the FlexE overhead. The specific process is described in clause B.2.1.2.

Management channel – section: The management channel is optional. When used, 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: A FlexESGM replacement signal (including clock, frame start, and multi-frame start) generation process is provided that generates a stream of local fault sequence ordered sets as specified in clause 5.2.2.3 of [OIF FLEXE IA]. The FlexESGM replacement signal clock, frame start, and multi-frame start are independent from the incoming clock, frame start, and multi-frame start. The FlexESGM replacement signal clock must be within the frequency range $103.125\ \text{Gb/s}\ \times \frac{16383}{16384}\ \pm 100$.

Selector: The normal FlexESGM signal or the replacement signal is selected. When aAIS is true, it shall select the replacement signals within two FlexE frames. Otherwise, it shall select the normal FlexE signals. On clearing aAIS, the FlexE replacement signal shall be removed within two FlexE frames, with normal data being output.

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 FlexE 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 clause B.1.1.1.1.

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 clause B.1.1.1.2.

Consequent actions

aSSF ← dLOM or dLOF or dPCS

aAIS ← dLOM or dLOF or dPCS

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

Defect correlations

cLOF ← dLOF and (not dPCS)

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

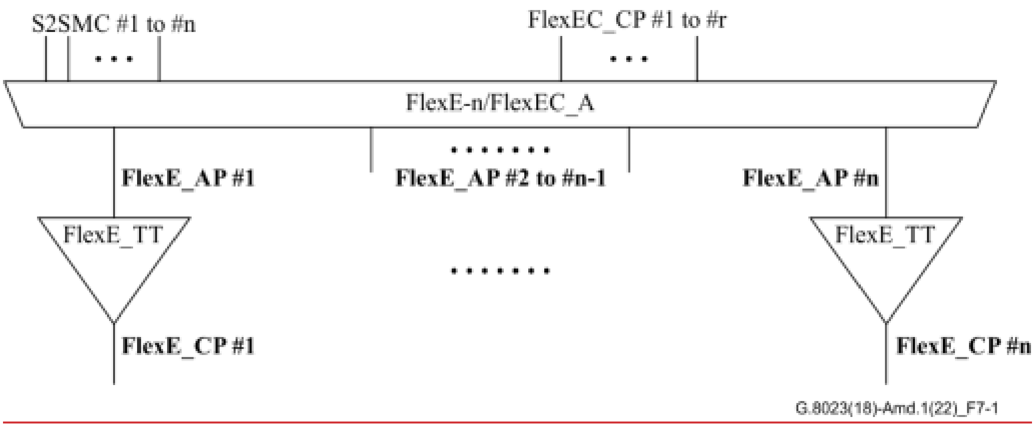
Performance monitoring

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

For an overview of the FlexE model, refer to Annex A. Figure 7-1 illustrates the atomic functions associated with the FlexE layer network. The n FlexE_TT functions compose a FlexE group with n 100G PHYs. A total of *r* FlexE client signals are multiplexed into the FlexE group. ~~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).~~



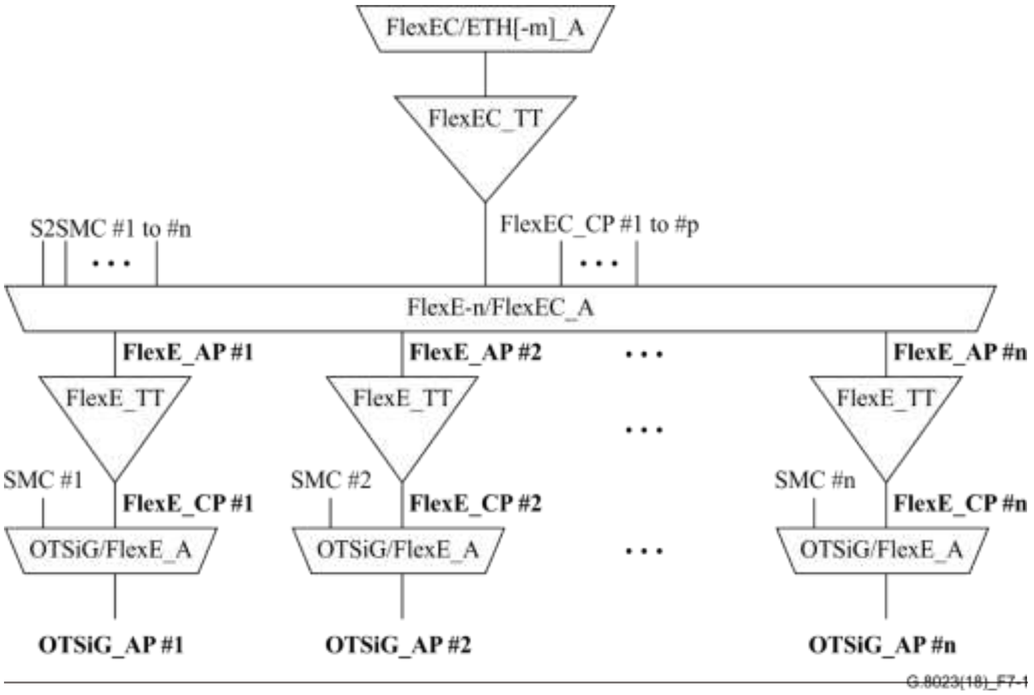


Figure 7-1 – FlexE ~~and FlexE client adaptation~~ functions

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~~ 24 or Figure 25 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~~M-AI/FlexE_A function; see clause 6.5.

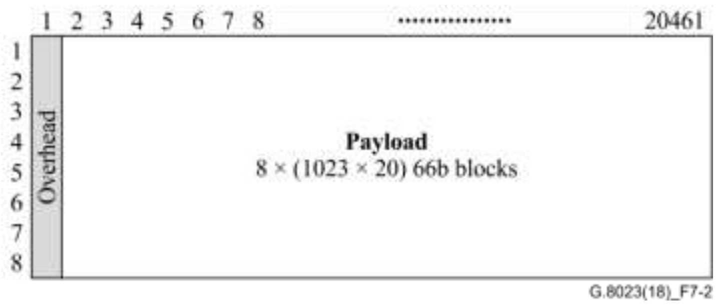


Figure 7-2 – FlexE frame

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 remote PHY fault (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 × 25 Gbit/s (n ≥ 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.

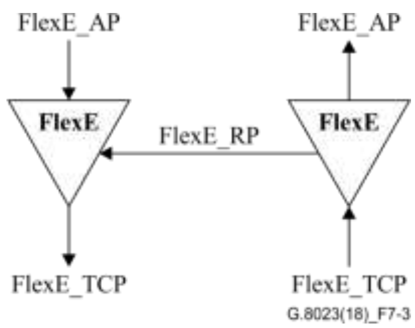


Figure 7-3 – FlexE_TT

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

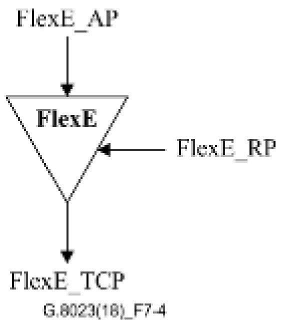


Figure 7-4 – FlexE_TT_So function

Interfaces

Table 7-12 – 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.

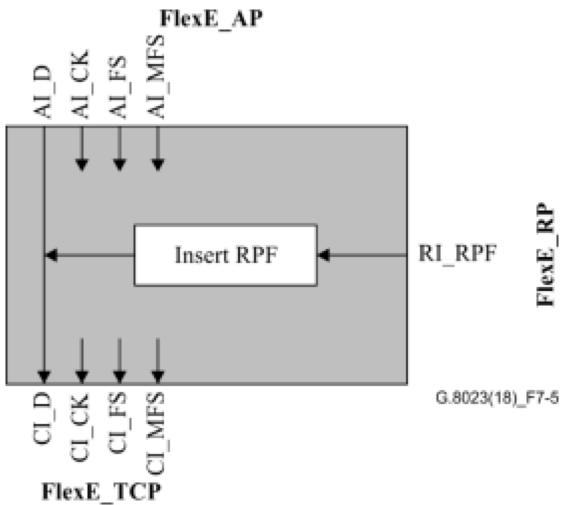


Figure 7-5 – FlexE_TT_So processes

Insert **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

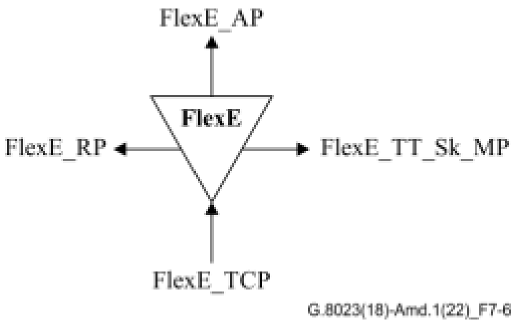


Figure 7-6 – FlexE_TT_Sk function

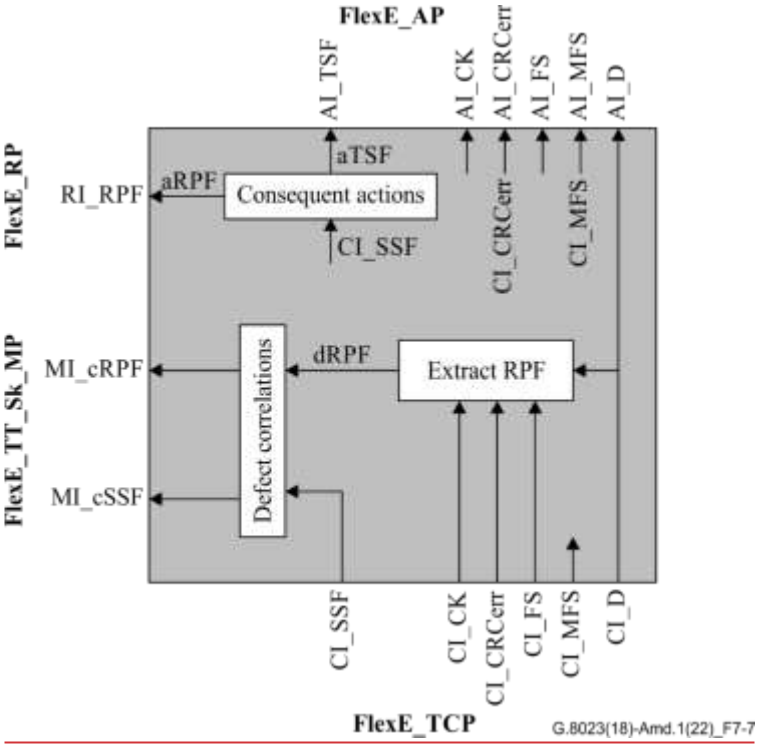
Interfaces

Table 7-2~~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_cRPF~~DI~~ FlexE_TT_Sk_MI_cSSF

Processes

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



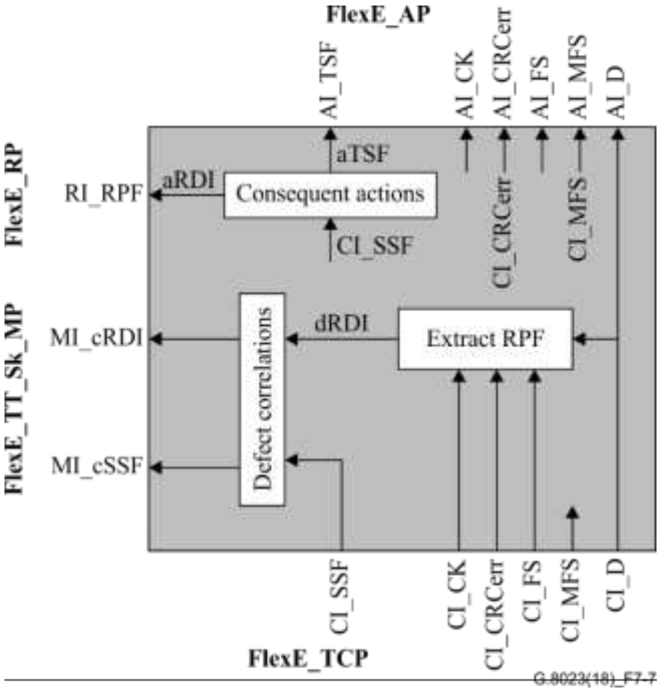


Figure 7-7 – FlexE_TT_Sk processes

Extract **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 dRPF~~DI~~ defect.

dRPF~~DI~~: If the extracted RPF is "1" with the good CRC, dRPF~~DI~~ shall be set to true; otherwise, dRPF~~DI~~ shall be set to false; dRPF~~DI~~ shall be set to false during CI_SSF.

Consequent actions

The function shall perform the following consequent actions:

aRPF~~DI~~ ← 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 equipment management function (EMF).

cSSF ← CI_SSF

cRPF~~DI~~ ← dRPF~~DI~~ and (not CI_SSF)

Performance monitoring: None.

7.2 FlexE-n to FlexE ~~c~~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. FlexE client signals can be a mix of 10G, 40G, and multiples of 25G.

7.2.1 FlexE-n to FlexE ~~c~~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

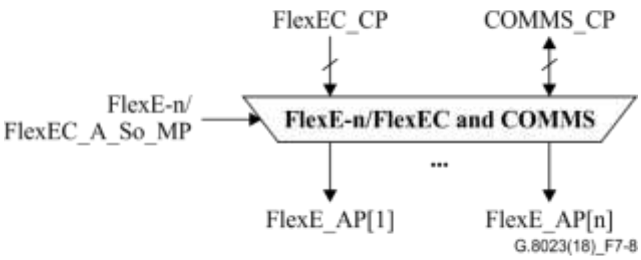


Figure 7-8 – FlexE-n/FlexEC_A_So function

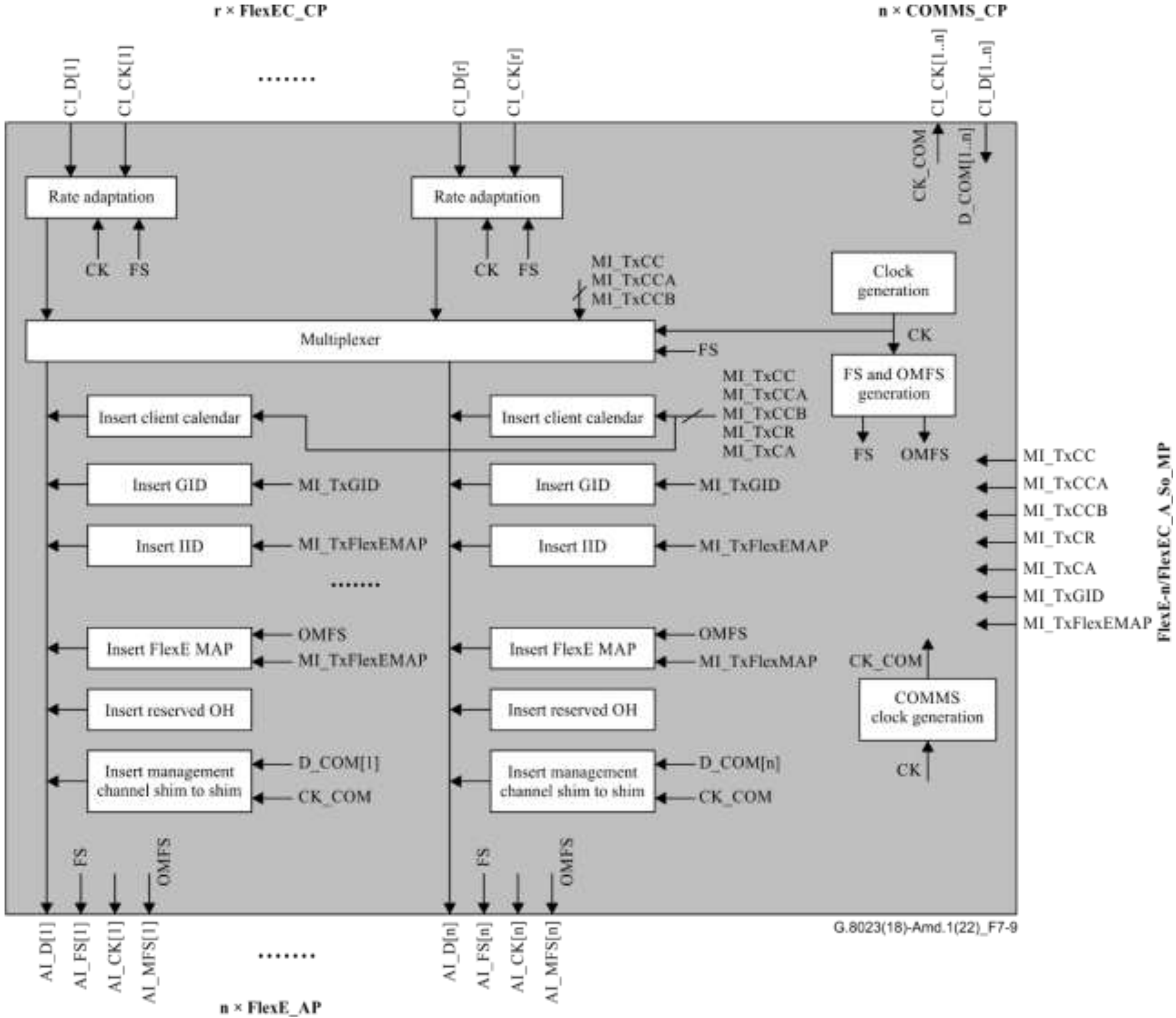
Interfaces

Table 7-34 – FlexE-n/FlexEC_A_So inputs and outputs

Input(s)	Output(s)
~~p~~ *r* × **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_Tx~~PHY~~FlexEMAP	**n × 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.


r × FlexEC_CP
n × COMMS_CP
CI_D[1]
CI_CK[1]
CI_D[r]
CI_CK[r]
CI_CK[1..n]
CI_D[1..n]
CK_COM
D_COM[1..n]
Rate adaptation
Rate adaptation
CK
FS
MI_TxCC
MI_TxCCA
MI_TxCCB
Clock generation
Multiplexer
CK
FS
FS and OMFS generation
MI_TxCR
MI_TxCA
OMFS
Insert client calendar
Insert GID
MI_TxGID
Insert IID
MI_TxFlexEMAP
Insert FlexE MAP
MI_TxFlexMAP
Insert reserved OH
Insert management channel shim to shim
D_COM[1]
D_COM[n]
CK_COM
COMMS clock generation
FlexE-n/FlexEC_A_So_MP
AI_D[1]
AI_FS[1]
AI_CK[1]
AI_MFS[1]
AI_D[n]
AI_FS[n]
AI_CK[n]
AI_MFS[n]
n × FlexE_AP
G.8023(18)-Amd.1(22)_F7-9

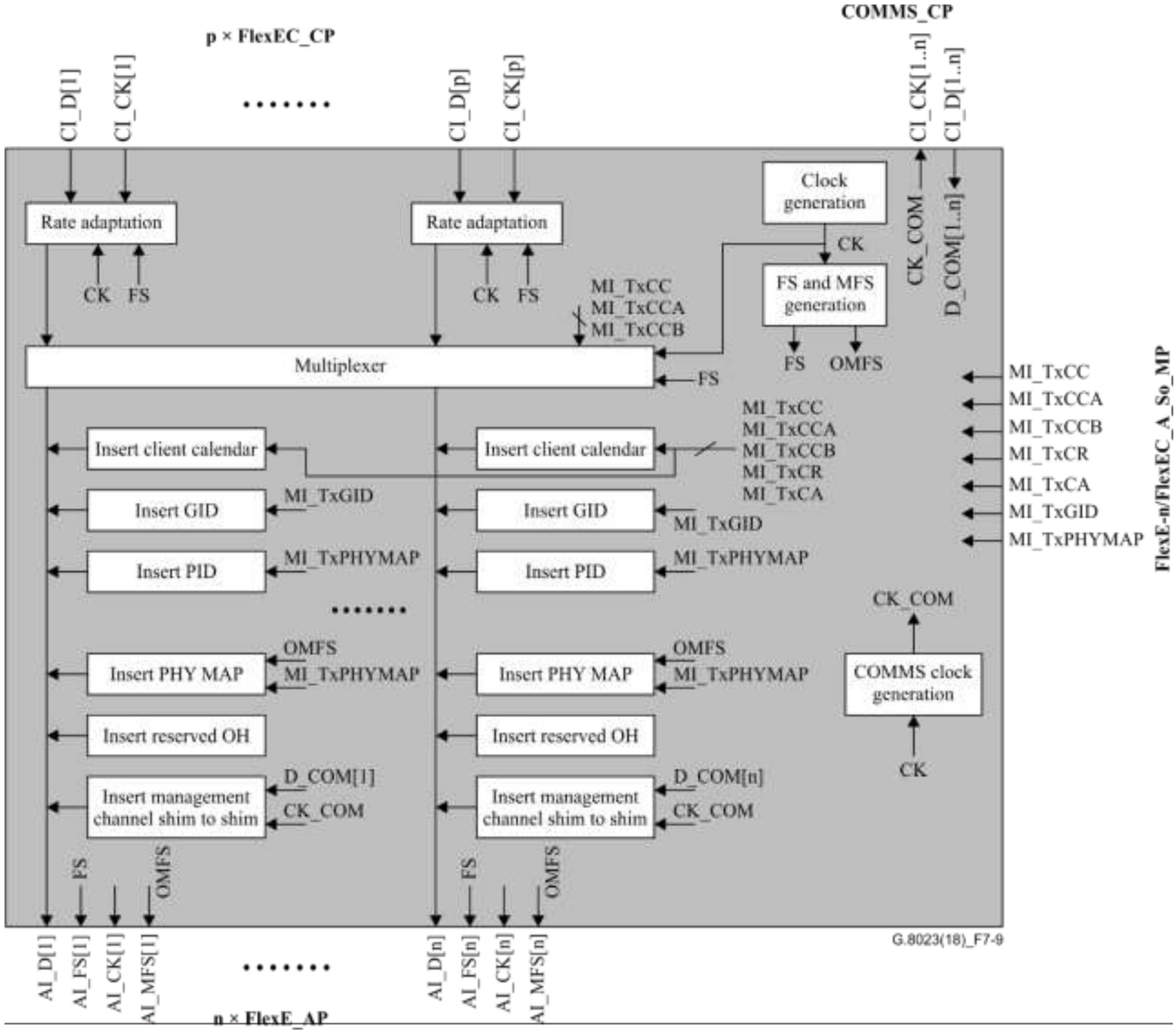


Figure 7-9 – FlexE-n/FlexEC_A_So processes

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 MFS generation: The function shall generate FlexE frame start (FS) and FlexE overhead multiframe start (MFS) 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 clause 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~~8, 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 ~~P~~IID: The FlexE ~~physical interface~~Instance identifier is inserted in the ~~PHY~~ FlexE i~~I~~nstance number field. Its value for each ~~PHY member~~FlexE i~~I~~nstance of the FlexE group is ~~a~~ unique ~~one~~ and can be configured via the MI_~~PHY~~TxFlexEMAP. The ~~P~~IID 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~~ FlexE MAP fields using the multiframe structure shown in Figure ~~11~~ 24 or Figure 25 of [OIF FLEXE IA]. Its value is configured via the MI_~~PHY~~TxFlexEMAP. The ~~PHY~~ FlexE 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 channel – shim to shim: The management channel is ~~for user specific and inserted~~optional. When used, the incoming COMMS (CI_D) data is inserted into the management 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 ~~c~~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

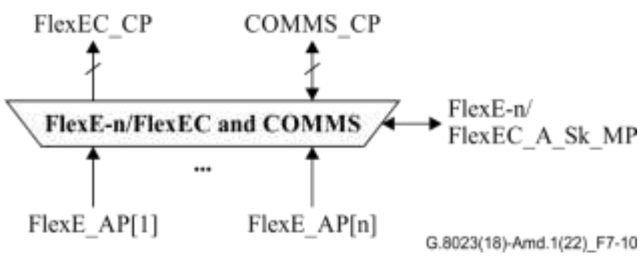


Figure 7-10 – FlexE-n/FlexEC_A_Sk function

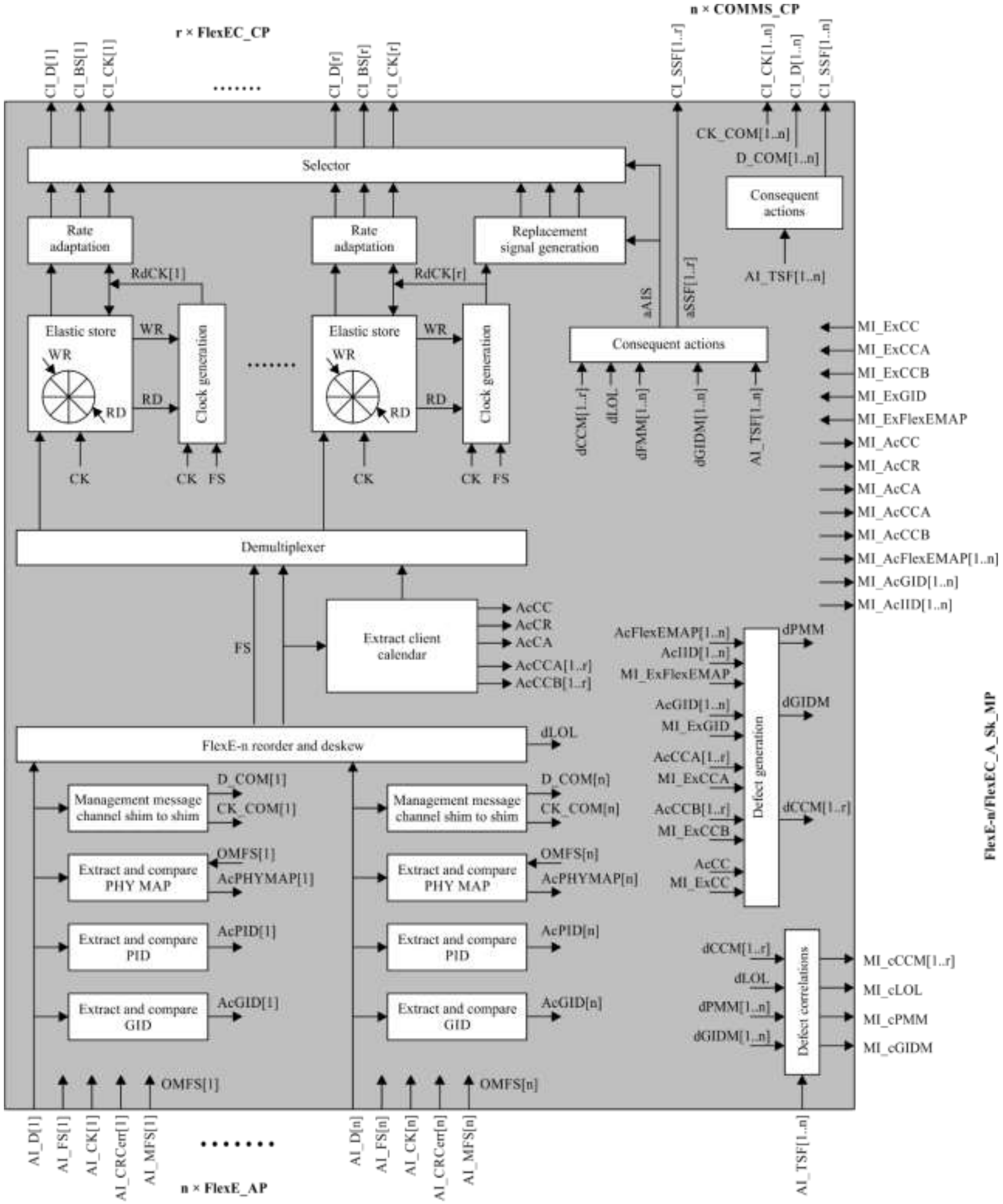
Interfaces

Table 7-4~~5~~ – FlexE-n/FlexEC_A_Sk inputs and outputs

Input(s)	Output(s)
n × 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-n/FlexEC_A_Sk_MI_ExCC FlexE-n/FlexEC_A_Sk_MI_ExCCA FlexE-n/FlexEC_A_Sk_MI_ExCCB FlexE-n/FlexEC_A_Sk_MI_ExGID FlexE-n/FlexEC_A_Sk_MI_Ex~~PHY~~FlexEMAP	~~p~~ *r* × **FlexEC_CP**: FlexEC_CI_CK FlexEC_CI_D FlexEC_CI_SSF **n × 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_AcCCA FlexE-n/FlexEC_A_Sk_MI_AcCCB FlexE-n/FlexEC_A_Sk_MI_AcFlexEMAP[1..n] FlexE-n/FlexEC_A_Sk_MI_AcGID[1..n] FlexE-n/FlexEC_A_Sk_MI_AcIID[1..n] FlexE-n/FlexEC_A_Sk_MI_cCCM[1..~~p~~*r*] FlexE-n/FlexEC_A_Sk_MI_cLOL FlexE-n/FlexEC_A_Sk_MI_c~~P~~FMM 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.



G.8023(18)-Amd.1(22)_F7-11

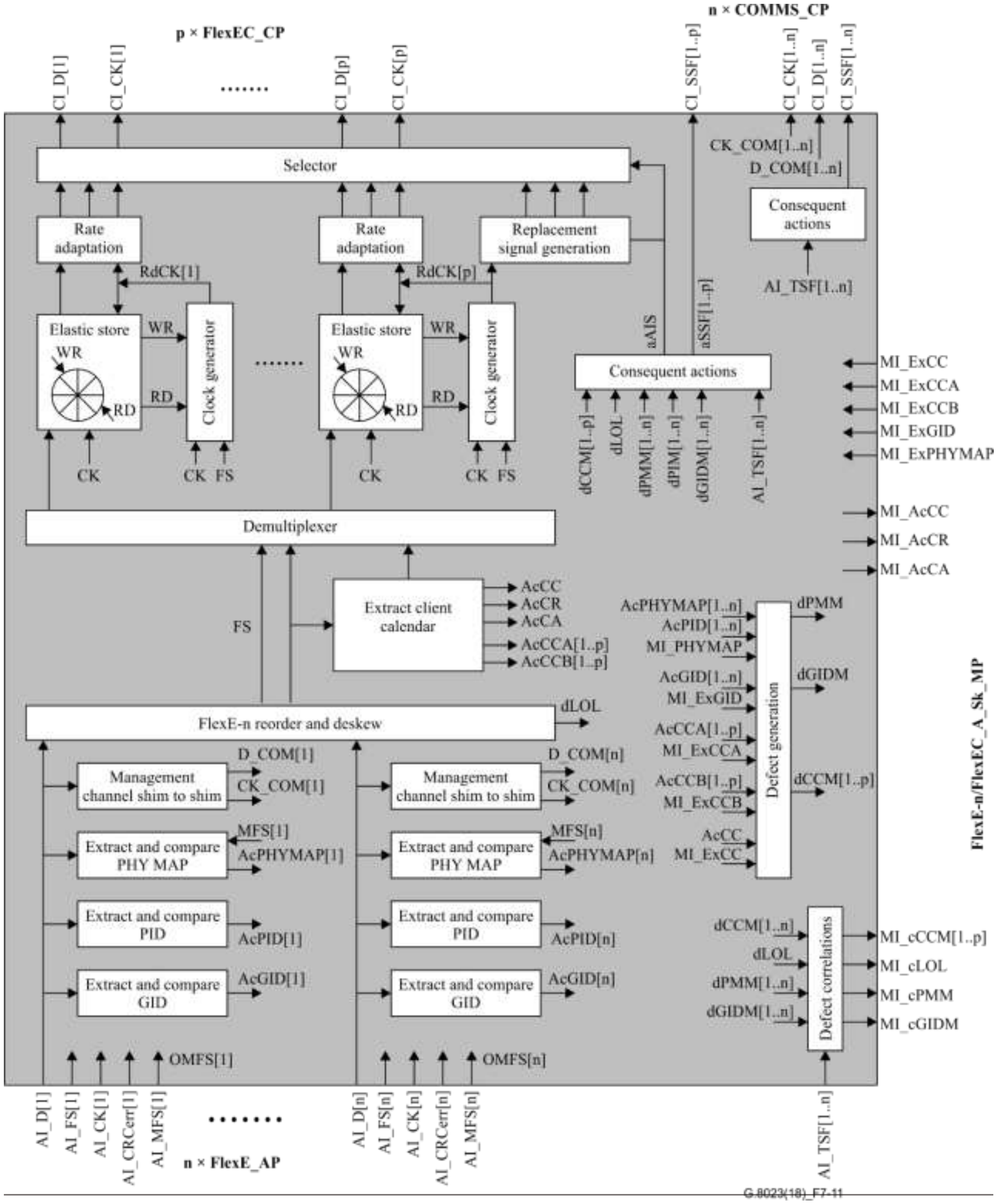


Figure 7-11 – FlexE-n/FlexEC_A_Sk processes

FlexE OH extraction: The function shall extract the overhead of FlexE group interface (GID, ~~P~~IID, ~~PHY~~ FlexE 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 ~~clause~~ 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 ~~P~~IID: The ~~P~~IID fields shall be read from the FlexE overhead and processed as specified in clause B.2.2.2.2 of [ITU-T G.798]. The accepted ~~PID~~ IID values are available at the MP (MI_~~AcPID~~AcIID[i]) and are used for ~~dPMM~~ dFMM defect detection.

FlexE ~~PHY~~ MAP: The FlexE MAP fields shall be read from the FlexE overhead and processed as specified in ~~clause~~ Annex B.2.2.3~~.2~~ ~~of [ITU-T G.798]~~. The accepted ~~PHY~~ FlexE MAP values are available at the MP (MI_~~AcPHYMAP~~AcFLEXEMAP[i]) and are used for d~~P~~FMM defect detection.

Management channel – shim to shim: The management message channel is ~~for user specific and~~optional. When used, the corresponding COMMS data (D_COM) and COMMS clock (CK_COM) is extracted from the management channel – shim to shim field described in clause 7.3.5 of [OIF FLEXE IA].

FlexE-n reorder and deskew: The function shall include reordering and deskewing processes.

Reordering: The process shall reorder the FlexE instances ~~n lanes of FlexE~~in the FlexE group ~~PHY~~ based on ~~P~~IID, per clause 7.3.3 of [OIF FLEXE IA].

Deskewing: The process shall compensate the skew between ~~n~~ the FlexE ~~PHY~~ instances 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~~FlexE instances as ~~given~~ described in clause 6.4 of [OIF FLEXE IA]. The compensation between ~~the PHY lanes~~FlexE instances is achieved by an elastic store per ~~PHY lane~~FlexE instance. Each ~~PHY lane~~FlexE instance 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~~n FlexE instances. 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~~n FlexE instance 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~~FlexE instances 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 6.8~~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~~*r*]) for *r*~~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~~r] or the MI_ExCCB[1..r~~…p~~] as indicated by the MI_ExCC or, when the MI_ExCC is set to any, by the accepted client calendar (AcCCA[1..r~~…p~~] or AcCCB[1..r~~…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 clause 7.3.4 of [OIF FLEXE IA], may be implemented by the EMF.

Clock generation: The function shall generate an FlexEC clock (FlexEC_CI_CK[1..~~p~~r]) from a free-running oscillator or the FlexE-n group clock with a bit rate as specified in Table ~~7~~8-1.

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 5.2.2.5 ~~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, d~~P~~FMM, dLOL and dCCM[*j*].

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

~~dPMM~~dFMM: See ~~clause~~ 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. dCCM shall be detected for each FlexEC. The index *j* ranges from 1 to ~~p~~ *r* and denotes the 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 '0', 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 '1', 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 == '0' and ExCC == '0'
            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 == '1' and ExCC == '1'
            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 d~~P~~FMM or dGIDM or ∑AI_TSF[1..n], where *j* = 1 to ~~*p*~~*r*

aAIS[*j*] ← dCCM[*j*] or dLOL or d~~P~~FMM or dGIDM or ∑AI_TSF[1..n], where *j* = 1 to ~~*p*~~*r*

Defect correlations

cGIDM ← dGIDM and (not ∑AI_TSF[1..n])

c~~P~~FMM ← d~~P~~FMM and (not dGIDM) and (not ∑AI_TSF[1..n])

cLOL ← dLOL and (not d~~P~~FMM) and (not dGIDM) and (not ∑AI_TSF[1..n])

cCCM[*j*] ← dCCM[*j*] and (not dLOL) and (not d~~P~~FMM) and (not dGIDM) and (not ∑AI_TSF[1..n]), where *j* = 1 to ~~*p*~~*r*

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 7-5~~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 tolerance(ppm)**	**Encoding**	**AM**
10GR	10 312 500	±100	Clause 49 64B~~b~~/66B~~b~~	No	10 312 500	±100	Clause 82 64B~~b~~/66B~~b~~	No
25GR	25 781 250	±100	Clause 49 64B~~b~~/66B~~b~~	No	25 781 250	±100	Clause 82 64B~~b~~/66B~~b~~	No
40GR	41 250 000	±100	Clause 82 64B~~b~~/66B~~b~~	Yes	41 250 000	±100	Clause 82 64B~~b~~/66B~~b~~	No
50GR	51 562 500	±100	Clause 82 64B/66B	Yes	51 562 500	±100	Clause 82 64B/66B	No
100GR	103 125 000	±100	Clause 82 64B~~b~~/66B~~b~~	Yes	103 125 000	±100	Clause 82 64B~~b~~/66B~~b~~	No

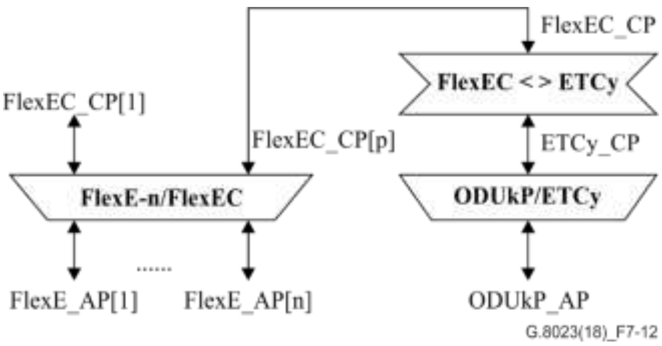


Figure 7-12 – FlexEC<>ETCy function position

7.3.1 FlexEC to ETCy interworking function (FlexEC> ETCy)

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

Symbol

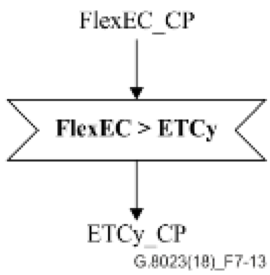


Figure 7-13 – FlexEC>ETCy function

Interfaces

Table 7-6~~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, 50GR, or 100GR).

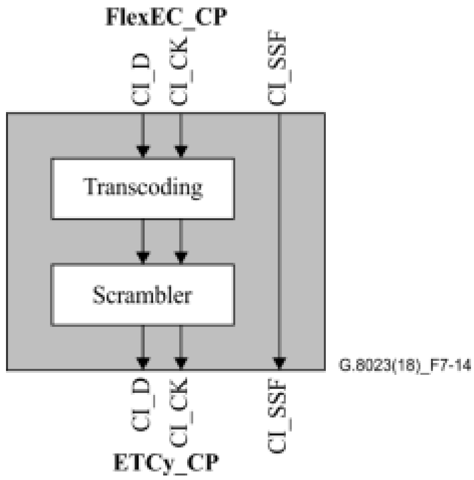


Figure 7-14 – FlexEC>ETCy processes (y = 10GR, 25GR)

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].

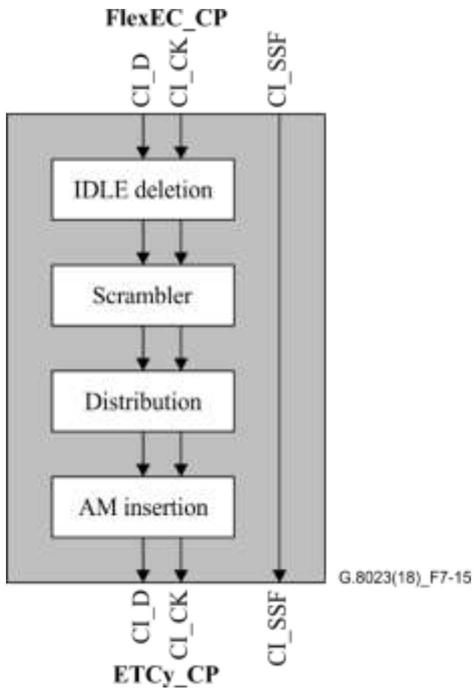


Figure 7-15 – FlexEC>ETCy processes (y = 40GR, 50GR, 100GR)

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 and ETC50GR) 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

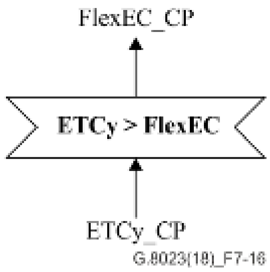


Figure 7-16 – ETCy>FlexEC function

Interfaces

Table 7-78 – 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, 50GR, or 100GR).

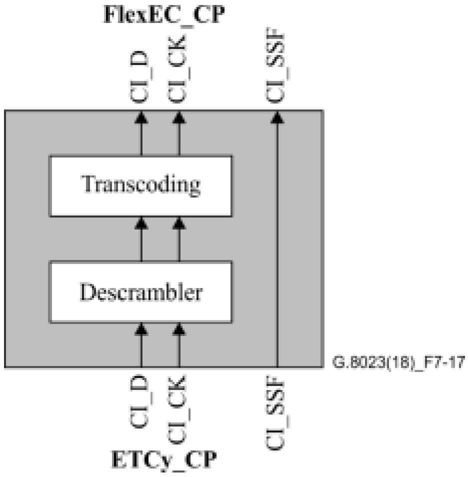


Figure 7-17 – ETCy>FlexEC processes (y = 10GR, 25GR)

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].

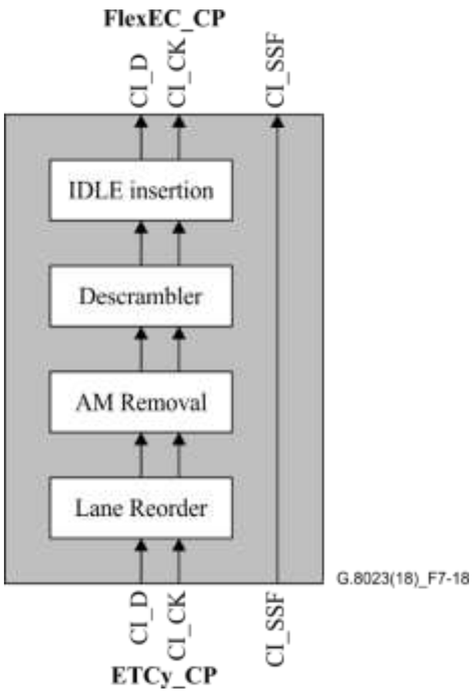


Figure 7-18 – ETCy>FlexEC processes (y = 40GR, 50GR, 100GR)

Lane reorder: The function shall order the 4 (ETC40GR and ETC50GR) 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 66Bb blocks. Four alignment markers are removed per 4*16383 66-bit blocks for ETC40GR and ETC50GR 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 ~~c~~Client functions

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 66~~b~~B block stream with the bit rates of 10, 40 and n × 25 Gbit/s (n ≥ 1) defined in Table 8-1.

Table 8-1 – FlexEC clients

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

8.1 FlexE ~~c~~Client trail termination function (FlexEC_TT)

The FlexEC_TT function terminates the FlexE ~~c~~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 ~~c~~Client, this is a null function.

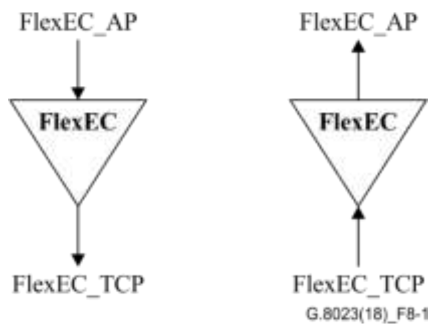


Figure 8-1 – FlexEC_TT

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

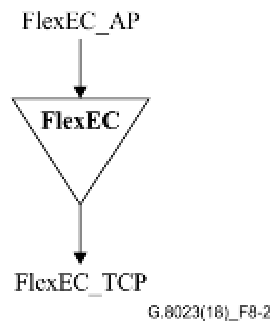


Figure 8-2 – FlexEC_TT_So function

Interfaces

Table 8-~~1~~ 2 – 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

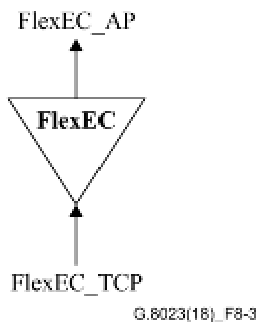


Figure 8-3 – FlexEC_TT_Sk function

Interfaces

Table 8-2 3 – 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.

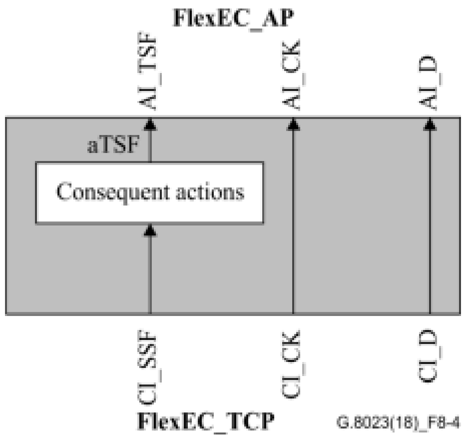


Figure 8-4 – FlexEC_TT_Sk processes

Defects: None.

Consequent actions

The function shall perform the following consequent actions:

aTSF ← CI_SSF

Defect correlations: None.

Performance monitoring: None.

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

The FlexEC to ETH adaptation functions perform the adaptation between a FlexE cClient 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

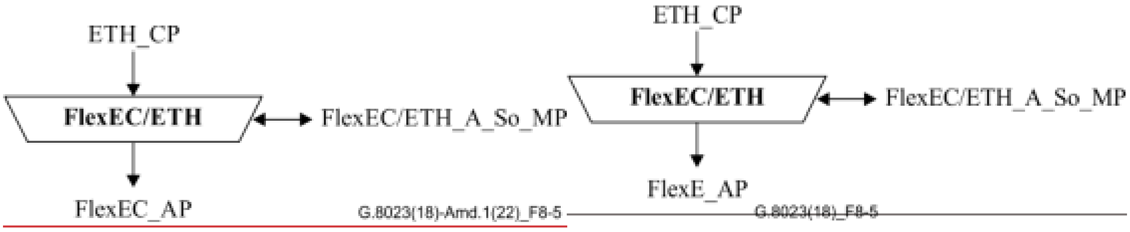


Figure 8-5 – FlexEC/ETH_A_So function

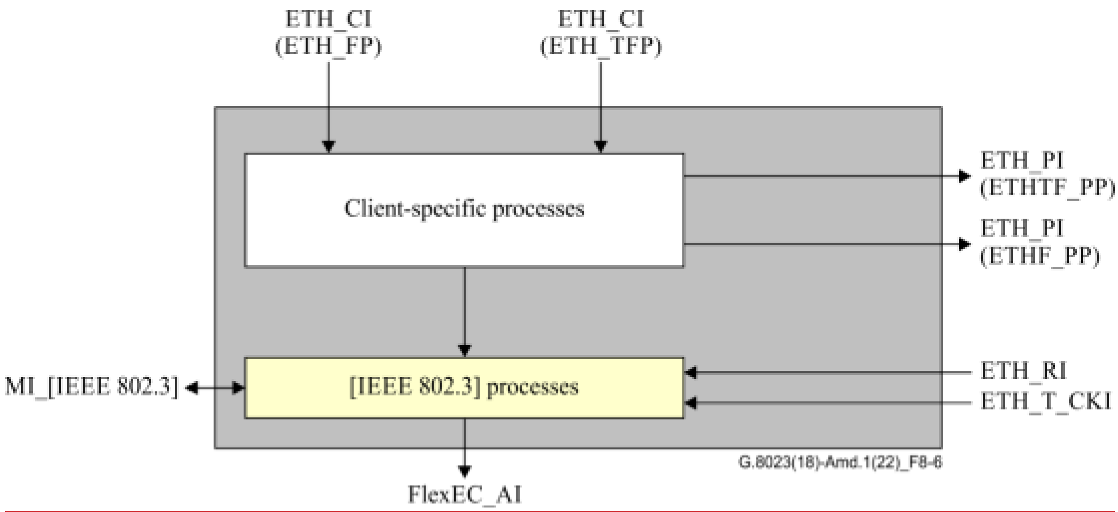
Interfaces

Table 8-3 4 – 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_CIPI_D ETH_CIPI_P ETH_CIPI_DE **ETHF_PP:** ETH_CIPI_D ETH_CIPI_P ETH_CIPI_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.



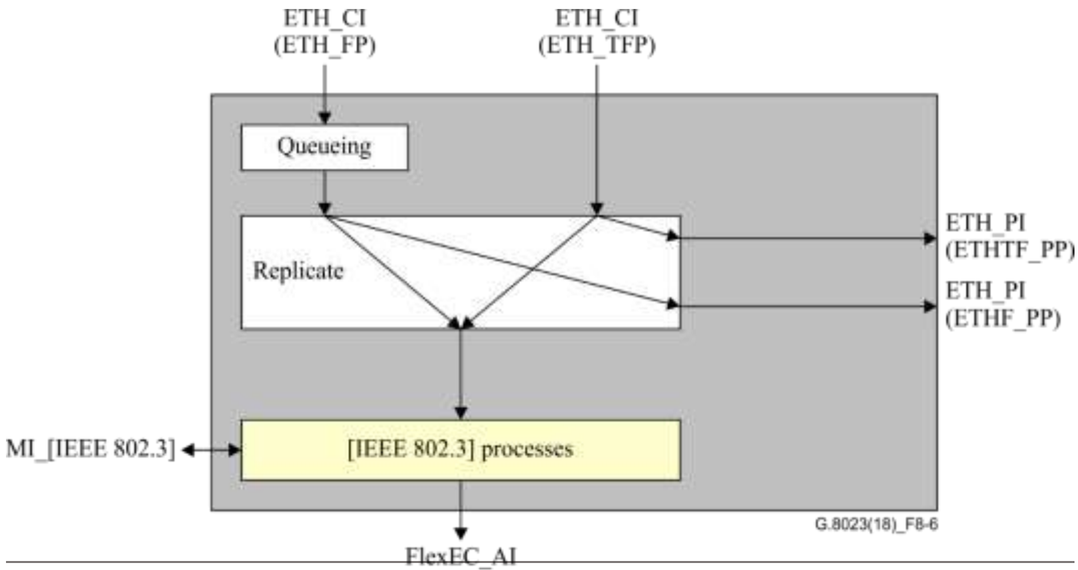


Figure 8-6 – FlexEC/ETH_A_So processes

The ~~queueing and replicate~~client-specific processes, and associated MI and PI signals, are defined in clause ~~8~~ 9.5 of [ITU-T G.8021].

[IEEE 802.3] processes: The [IEEE 802.3] processes represent the whole functionality of the PHY above the FlexE shim in the IEEE 802.3 model. This includes the reconciliation sublayer and the process of ~~64b~~64B/~~66b~~ 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

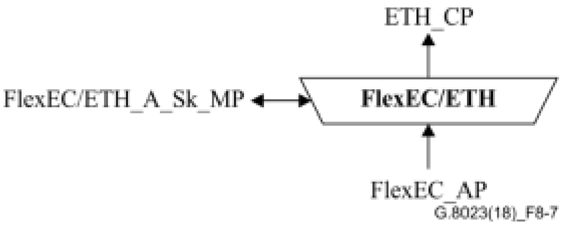


Figure 8-7 – FlexEC/ETH_A_Sk function

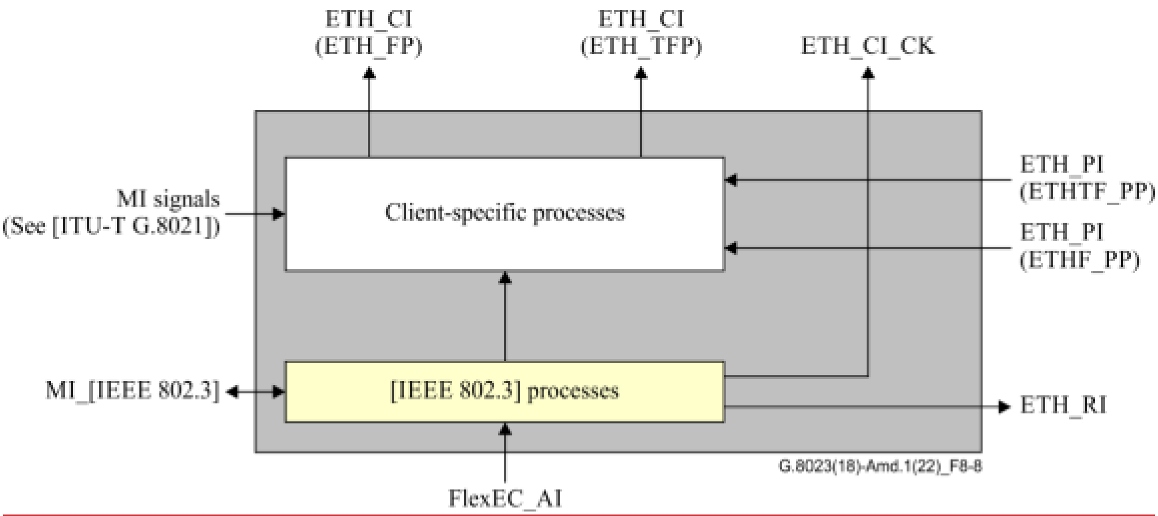
Interfaces

Table 8-~~4~~ 5 – 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~~PI_D ETH_~~CI~~PI_P ETH_~~CI~~PI_DE **ETHF_PP**: ETH_~~CI~~PI_D ETH_~~CI~~PI_P ETH_~~CI~~PI_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.



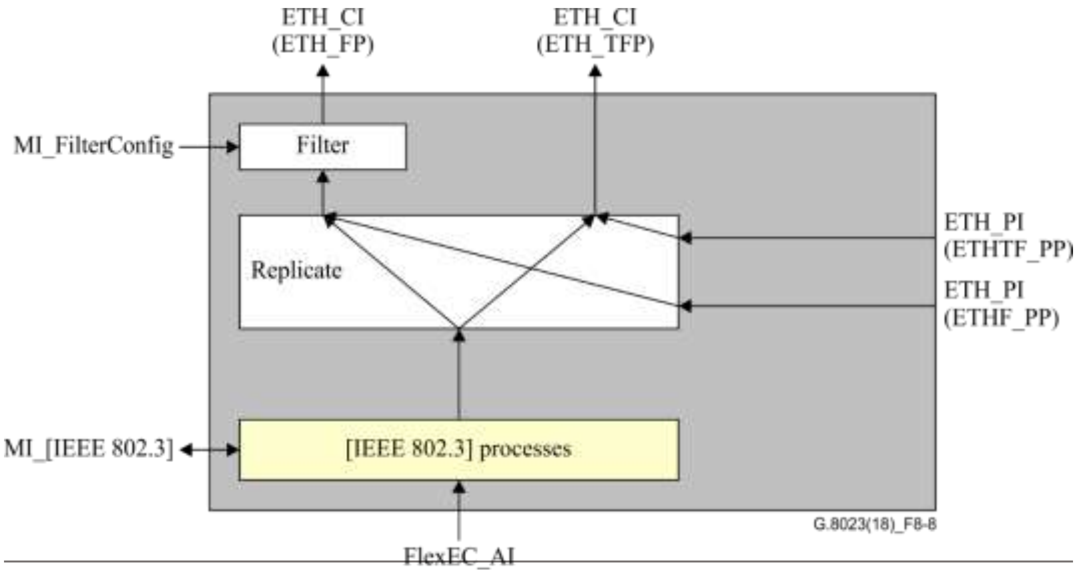


Figure 8-8 – FlexEC/ETH_A_Sk processes

The filter and replicateclient-specific processes, and associated MI and PI signals, are defined in clause 8 9.5 of [ITU-T G.8021].

[IEEE 802.3] processes: The [IEEE 802.3] processes represent the whole functionality of the PHY layer above the FlexE shim; this includes 64b64B/66b 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 of related of FlexE-related atomic functions

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

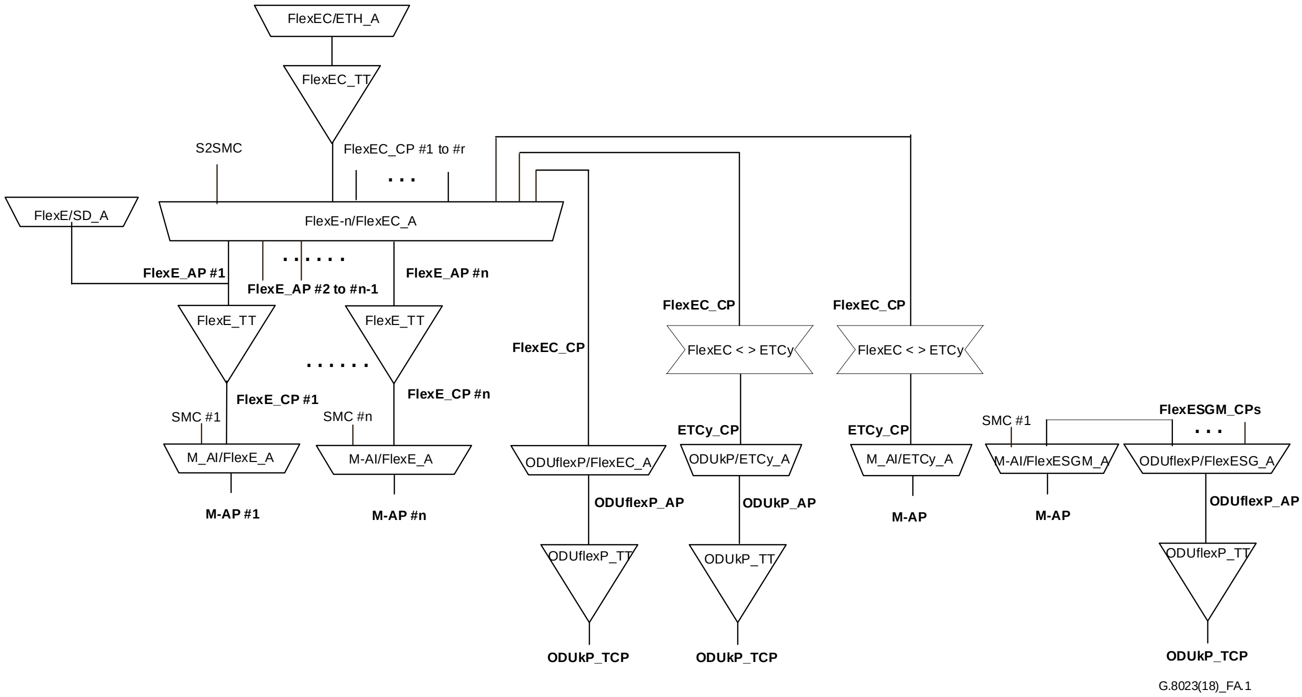
This annex describes an overview of related of FlexE-related atomic functions. While FlexE technology is modelled using multiple layer networks, the intent of FlexE technology is to provide a mechanism for bonding PCS lanes from one or more PHYs into a larger PHY. Inverse multiplexing mechanisms such as this are traditionally modelled as an adaptation function (which may be part of a compound trail termination function). The FlexE model is broken out into more detail to facilitate the different mechanisms for interworking FlexE and OTN. The FlexE model is not intended to imply that connection functions exist for the FlexE or FlexEC characteristic information.

The left part of Figure A.1 shows the atomic functions that represent a FlexE group, illustrating the creation of FlexE clients from the ETH layer, the FlexE mux/demux, and the FlexE PHYs, as well as the adaptation for the optional synchronization messaging channel in the first PHY and the connection points for the section management (SMC) and shim-to-shim management (S2SMC) channels.

The middle part of Figure A.1 illustrates some alternative ways a FlexE Client might be connected into a FlexE group. ODUflexP/FlexEC_A (see clause 14.3.17 of [ITU-T G.798]) is used for the mapping of a FlexE client into ODUflex in support of the FlexE termination in the transport network application described in clause 8.2 of [OIF FLEXE IA]. The two branches with interworking functions illustrate cases where the FlexE client is created by interworking an ETCy sublayer to the FlexEC coding. The ETCy signal may originate from a native Ethernet port or from an OTN port if the ETCy was mapped into ODUk by another node. These interworking options can be used for port-based services that do not require processing in the ETH layer.

The right part of Figure A.1 illustrates the functions used for mapping of a FlexE subgroup into ODUflex in support of FlexE aware transport application described in clause 8.3 of [OIF FLEXE IA].

~~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 three 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 [ITU-T 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 [ITU-T 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 [ITU-T G.798]).~~



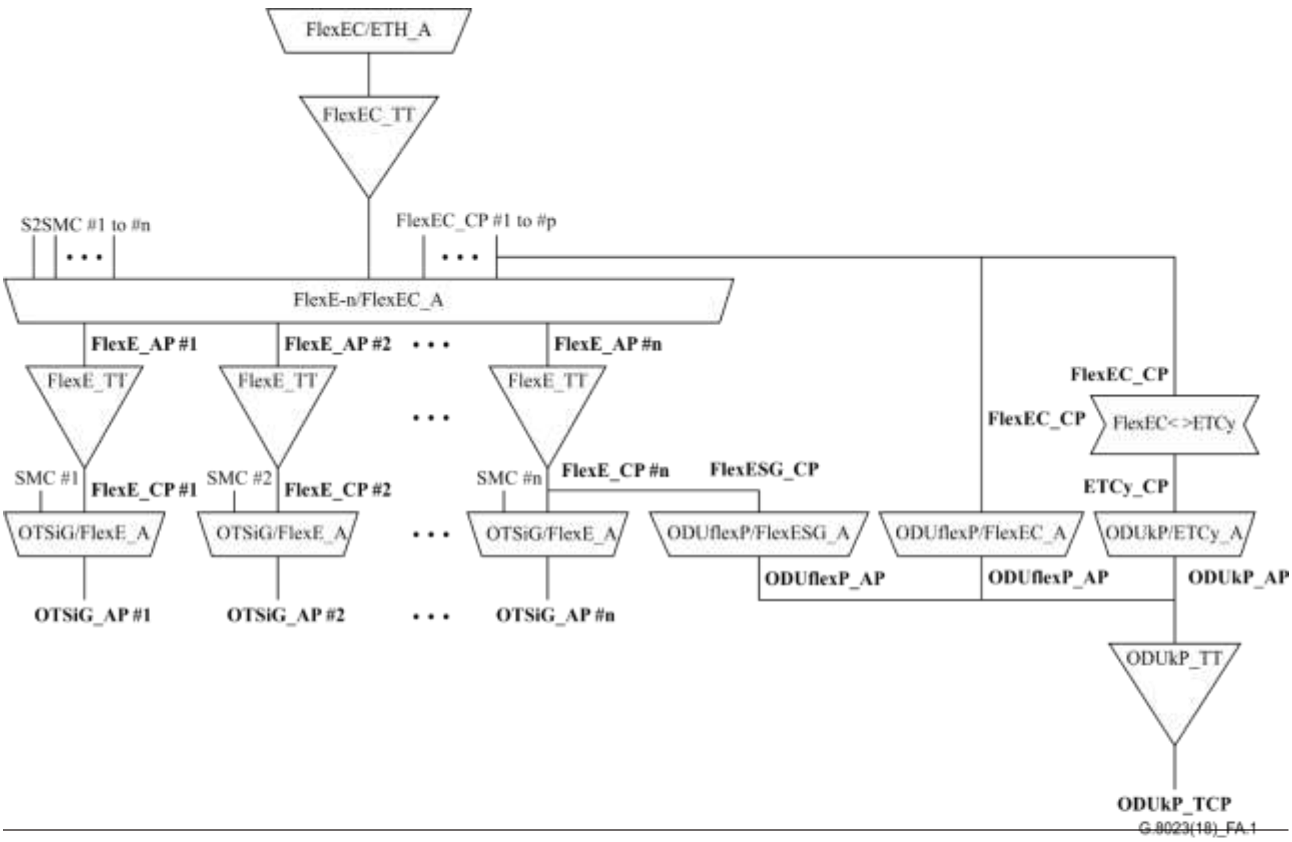


Figure A.1 – Overview of FlexE-related atomic functions

NOTE – The OTN-related functions in Figure A.1 are defined in [ITU-T G.798].

Annex B

Generic FlexE supervision and processes

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

B.1 Supervision

B.1.1 Defects

B.1.1.1 Alignment supervision

B.1.1.1.1 FlexE overhead loss of frame (dLOF)

FlexE overhead dLOF is generated based on the state of the frame alignment process defined in clause B.2.1.1.

dLOF shall be declared if the FlexE overhead frame alignment process is in the out-of-frame (OOF) state.

dLOF shall be cleared when the frame alignment process is in the IF state.

B.1.1.1.2 FlexE overhead loss of multi-frame (dLOM)

FlexE overhead dLOM is generated based on the state of the multi-frame alignment process defined in clause B.2.1.2.

dLOM shall be declared if the multi-frame alignment process is in the out-of-multiframe (OOM) state.

dLOM shall be cleared immediately when the multi-frame alignment process is in the in-multiframe (IM) state.

B.1.1.2 Group supervision

B.1.1.2.1 Group ID mismatch (dGIDM)

dGIDM shall be declared if the accepted GID (AcGID) of any of the group member interfaces is not equal to the expected GID (ExGID).

dGIDM shall be cleared if the accepted GID of all group member interfaces is equal to the expected GID.

dGIDM shall be detected within 100 ms of changes to the AcGID.

B.1.1.2.2 FlexE map mismatch (dFMM)

The FlexE Map mismatch defect of a FlexO group or FlexE (sub)group of p members is based on the comparison of the p AcFlexEMAP values, the p AcIID values and the expected FlexE map (ExFlexEMAP).

dFMM shall be cleared if:

– the AcFlexEMAP value of each member matches the value of ExFlexEMAP;

– each value of AcIID has the corresponding bit set in ExFlexEMAP;

– every AcIID value is unique.

Otherwise, dFMM shall be declared.

dFMM shall be detected within 100 ms of changes to the AcFlexEMAP[1..p] or AcIID[1..p] values.

B.2 Generic processes

B.2.1 Alignment processes

B.2.1.1 FlexE overhead frame alignment

The FlexE overhead frame alignment shall be found by searching for FlexE block 1 of the FlexE overhead frame every (1023 × 20 + 1) × 8 blocks as described in clause 7.3.1 of [OIF FlexE IA].

NOTE – 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.

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 IF state shall be entered, when a valid FlexE block 1 is found in two consecutive FlexE overhead frames.

In the IF state, OOF shall be entered when the FlexE block 1 of the FlexE overhead frame has mismatches on the sync header, control block type or O code fields for 5 consecutive occurrences.

B.2.1.2 FlexE overhead multi-frame alignment

The Flex overhead multi-frame alignment shall be found based on the overhead multi-frame (OMF) bit in the FlexE overhead. The FlexE overhead multi-frame is a sequence of 32 FlexE overhead frames. The OMF bit has a value of "0" for the first sixteen overhead frames of the FlexE overhead multiframe, and a value of "1" for the last sixteen overhead frames of the FlexE overhead multiframe, as described in clause 7.3.1 of [OIF FLEXE IA].

The process has two states, out-of-multi-frame (OOM) and in-multi-frame (IM). The multi-frame start shall be maintained during the OOM state.

In the OOM state, the state will change from OOM to IM upon detecting a "0" to "1" or a "1" to "0" transition of the OMF bit in two consecutive FlexE overhead frames with good CRC. In the IM state, the state shall change from IM to OOM, when the two consecutive FlexE overhead frames where the "0" to "1" or "1" to "0" transition of the OMF bit is expected, have a good CRC, but do not have the expected transition.

B.2.2 Overhead acceptance processes

B.2.2.1 Group identification (GID) acceptance process

A new GID value (AcGID) is accepted if a new value of the Group Identification field of the Group Number field of the FlexE overhead is received in an overhead frame with good CRC.

B.2.2.2 Instance identification (IID) acceptance process

A new IID value (AcIID) is accepted when the same value of the FlexE Instance number field of the FlexE overhead is received in two consecutive overhead frames with good CRC.

B.2.2.3 FlexE map acceptance process

The 256-bit FlexE Map[0:255] is recovered from the FlexE Map field of the FlexE overhead.

When FlexE overhead frame #i in the multi-frame is received with good CRC, the bits AcFlexEMAP[(i-1)×8:(i-1)×8+7] are accepted from the PHY map field.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	**Transmission systems and media, digital systems and networks**
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2022