

**ITU-T**

TELECOMMUNICATION  
STANDARDIZATION SECTOR  
OF ITU

**G.709/Y.1331**

**Amendment 3**  
(04/2009)

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

Digital terminal equipments – General

SERIES Y: GLOBAL INFORMATION  
INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS  
AND NEXT-GENERATION NETWORKS

Internet protocol aspects – Transport

---

Interfaces for the Optical Transport Network (OTN)

**Amendment 3: 100 Gbit/s support, one-stage  
multiplexing and other improvements**

Recommendation ITU-T G.709/Y.1331 (2003) –  
Amendment 3

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
<b>General</b>	<b>G.700–G.709</b>
Coding of voice and audio signals	G.710–G.729
Principal characteristics of primary multiplex equipment	G.730–G.739
Principal characteristics of second order multiplex equipment	G.740–G.749
Principal characteristics of higher order multiplex equipment	G.750–G.759
Principal characteristics of transcoder and digital multiplication equipment	G.760–G.769
Operations, administration and maintenance features of transmission equipment	G.770–G.779
Principal characteristics of multiplexing equipment for the synchronous digital hierarchy	G.780–G.789
Other terminal equipment	G.790–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
ACCESS NETWORKS	G.9000–G.9999

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

# **Recommendation ITU-T G.709/Y.1331**

## **Interfaces for the Optical Transport Network (OTN)**

### **Amendment 3**

#### **100 Gbit/s support, one-stage multiplexing and other improvements**

##### **Summary**

Amendment 3 to Recommendation ITU-T G.709/Y.1331 contains extensions to the second version (03/2003) of this Recommendation. It specifies 100 Gbit/s ODU4/OTU4, support of gigabit Ethernet services via ODU0, ODU2e, ODU3 and ODU4, multi-lane OTU3 and OTU4 and the Lower Order and Higher Order ODU concept to align with the "one-stage multiplexing" specification described in clause 9.2 of Recommendation ITU-T G.872.

##### **Source**

Amendment 3 to Recommendation ITU-T G.709/Y.1331 (2003) was approved on 22 April 2009 by ITU-T Study Group 15 (2009-2012) under Recommendation ITU-T A.8 procedures.

## 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 received notice of intellectual property, protected by patents, 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 TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2010

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

## CONTENTS

	<b>Page</b>
1) Introduction .....	1
2) Additions .....	1
2.1) Clause 2 .....	1
2.2) Clause 4 .....	2
2.3) Clause 6.1 .....	2
2.4) Clause 7 Multiplexing/mapping principles and bit rates.....	2
2.5) Clause 7.3 Bit rates and capacity.....	6
2.6) Clause 8 and subclauses .....	8
2.7) Clause 9 and subclauses .....	15
2.8) Clause 10 .....	15
2.9) Clause 11.1 OTUk frame structure.....	15
2.10) Clause 12.1 ODUk frame structure .....	17
2.11) Clause 13 Optical channel payload unit (OPUk) .....	17
2.12) Clause 15.8.2.4 ODUk automatic protection switching and protection communication channel (APS/PCC).....	18
2.13) Clause 15.9.2.1.1 OPUk payload type (PT) .....	19
2.14) Clause 17.1 .....	20
2.15) Clause 17.1 .....	22
2.16) Clause 17.3 .....	23
2.17) Clause 17 .....	24
2.18) Clause 19.1 OPUk Tributary Slot definition .....	31
2.19) Clause 19.1 .....	31
2.20) Clause 19.2 .....	32
2.21) Clause 19.3 Multiplexing ODTUjk signals into the OPUk.....	33
2.22) Clause 19.3 .....	33
2.23) Clause 19.4 OPUk Multiplex Overhead.....	35
2.24) Clause 19.4.1 OPUk Multiplex Structure Identifier (MSI).....	37
2.25) Clause 19.4.1 .....	38
2.26) Clause 19.4.2 OPUk Payload Structure Identifier Reserved overhead (RES).....	38
2.27) Clause 19.4.3 OPUk multiplex justification overhead (JOH).....	39
2.28) Clause 19.4 .....	39
2.29) Clause 19.5 Mapping ODUj into ODTUjk.....	39
2.30) Clause 19.5 .....	40
2.31) New Annex B .....	42
2.32) New Annex C .....	48
2.33) Appendix I.....	52
2.34) Appendix I.....	52
2.35) Appendix III Example of ODUk multiplexing.....	54

	<b>Page</b>
2.36) New Appendix VII .....	55
2.37) New Appendix VIII.....	59
2.38) New Appendix IX.....	65

# Recommendation ITU-T G.709/Y.1331

## Interfaces for the Optical Transport Network (OTN)

### Amendment 3

#### 100 Gbit/s support, one-stage multiplexing and other improvements

##### 1) Introduction

This amendment contains extensions to the second version (03/2003) of Recommendation ITU-T G.709/Y.1331, related to the addition of:

- An ODU0 and OPU0 and transport of 1000BASE-X (1GE) signal over ODU0 (see clauses 2.5, 2.10, 2.11, 2.13, 2.17, and 2.38)
- An ODU2e and OPU2e and transport of CBR10G3 (10GE) and FC-1200 signals over ODU2e (see clauses 2.5, 2.10, 2.11, 2.13, 2.14, 2.15, 2.17, and 2.31)
- An OTU4, ODU4 and OPU4 (see clauses 2.5, 2.6, 2.7, 2.9, 2.10, and 2.11)
- An additional GFP frame mapping into an OPU2 with extended OPU2 payload area (see clauses 2.13, and 2.16)
- Multiplexing of two ODU0 signals into an OPU1 (see clauses 2.18, 2.19, 2.20, 2.21, 2.22, 2.23, 2.24, 2.25, 2.26, 2.27, 2.29, 2.30, 2.33, 2.34, and 2.35)
- Lower order and higher order ODU concept to align with the "one-stage multiplexing" specification described in clause 9.2 of Recommendation ITU-T G.872 (see clause 2.4)
- Multiplexing of ODU0 signals into an OPU2 or OPU3 (see clauses 2.20, 2.22, and 2.30)
- Multiplexing of ODU2e signals into an OPU3 (see clauses 2.20, 2.22, and 2.30)
- Multiplexing of ODUj signals into an OPU4 (see clauses 2.18, 2.19, 2.23, 2.24, 2.25, 2.26, 2.27, and 2.28)
- Multi-lane OTU3 and OTU4 interfaces (see clauses 2.3, 2.5, 2.6, 2.7, 2.8, and 2.32)
- OTM-32r.m interface specified in Recommendation ITU-T G.959.1 (see clauses 2.6, and 2.7)
- Extension of ODUk APS/PCC scope (see clause 2.12)
- Parallel to serial conversion of 40GBASE-R and 100GBASE-R interface signals (see clause 2.36)
- Improving robustness for mapping 40GBASE-R into OPU3 (see clause 2.37).

##### 2) Additions

###### 2.1) Clause 2

*Add the following reference:*

- IEEE Std. 802.3-2005, *Information Technology – Local and Metropolitan Area Networks – Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.*

**2.2) Clause 4**

Add the following abbreviations:

OTL      Optical channel Transport Lane

TSOH     Tributary Slot Overhead

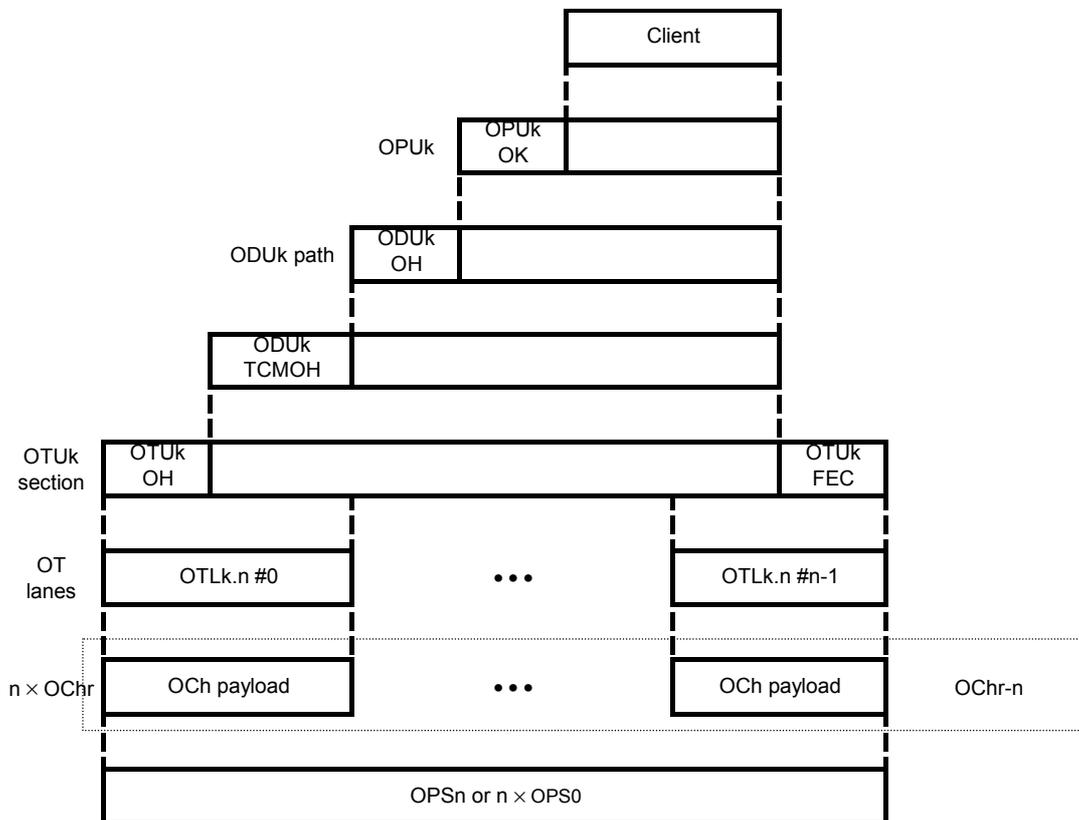
**2.3) Clause 6.1**

Add the following clause:

**6.1.4 Parallel OTM-0.mvn structure**

The OTM-0.mvn consists of the following layers (see Figure 6-6):

- optical physical section (OPS<sub>n</sub>);
- reduced functionality optical channel (OChr);
- optical channel transport lane (OTLk.n);
- optical channel transport unit (OTUk);
- optical channel data unit (ODUk).



**Figure 6-6 – OTM-0.mvn principal information containment relationships**

**2.4) Clause 7 Multiplexing/mapping principles and bit rates**

Modify the text and figure as follows:

Figures 7-1A and 7-1B shows the relationship between various information structure elements and illustrates the multiplexing structure and mappings (including wavelength and time division multiplexing) for the OTM-n.

Figure 7-1A shows that a (non-OTN) client signal is mapped into a lower order OPU, identified as "OPU (L)". The OPU (L) signal is mapped into the associated lower order ODU, identified as "ODU (L)". The ODU (L) signal is either mapped into the associated OTU[V] signal, or into an ODTU. The ODTU signal is multiplexed into an ODTU Group (ODTUG). The ODTUG signal is mapped into a higher order OPU, identified as "OPU (H)". The OPU (H) signal is mapped into the associated higher order ODU, identified as "ODU (H)". The ODU (H) signal is mapped into the associated OTU[V].

NOTE 1 – The OPU (L) and OPU (H) are the same information structures, but with different client signals. The ODU (L) and ODU (H) are the same information structures, but with different client signals.

Figure 7-1B shows that an OTU[V] signal is mapped either into an optical channel signal, identified as OCh and OChr, or into an OTLk.n. The OCh/OChr signal is mapped into an optical channel carrier, identified as OCC and OCCr. The OCC/OCCr signal is multiplexed into an OCC Group, identified as OCG-n.m and OCG-nr.m. The OCG-n.m signal is mapped into an OMSn. The OMSn signal is mapped into an OTSn. The OTSn signal is presented at the OTM-n.m interface. The OCG-nr.m signal is mapped into an OPSn. The OPSn signal is presented at the OTM-nr.m interface. A single OCCr signal is mapped into an OPS0. The OPS0 signal is presented at the OTM-0.m interface. A group of n OPS0 signals is presented at the OTM-0.mvn interface.

The ODU (L) into ODU (H) multiplexing and the OCh/OChr into OMSn/OPSn multiplexing provide a two-stage multiplexing capability typically present in administrative domains.

Figure 7-1A shows an "External ODU (H)". An External ODU (H) signal is an ODU (H) signal that terminates in another (administrative) domain and that is to be transported through this (administrative) domain. Such External ODU (H) may be mapped into an ODTU and that ODTU may be multiplexed into an ODTUG together with ODTU signals carrying an ODU (L).

This External ODU (H) into ODU (H) multiplexing supports the transport of, e.g., a carrier's ODU (H) through the (administrative) domain of a carrier-carrier, while maintaining the single stage ODU multiplexing in this carrier-carrier network.

NOTE 2 – It is possible to transport an ODU0 within an ODU1, an ODU1 within an ODU2, an ODU2 within an ODU3 and an ODU3 within an ODU4, and it would thus be possible to create a four-stage ODU multiplex hierarchy "ODU0 → ODU1 → ODU2 → ODU3 → ODU4". Such multi-stage multiplex hierarchy is an undesired hierarchy within one (administrative) domain, as specified in Recommendation ITU-T G.872.

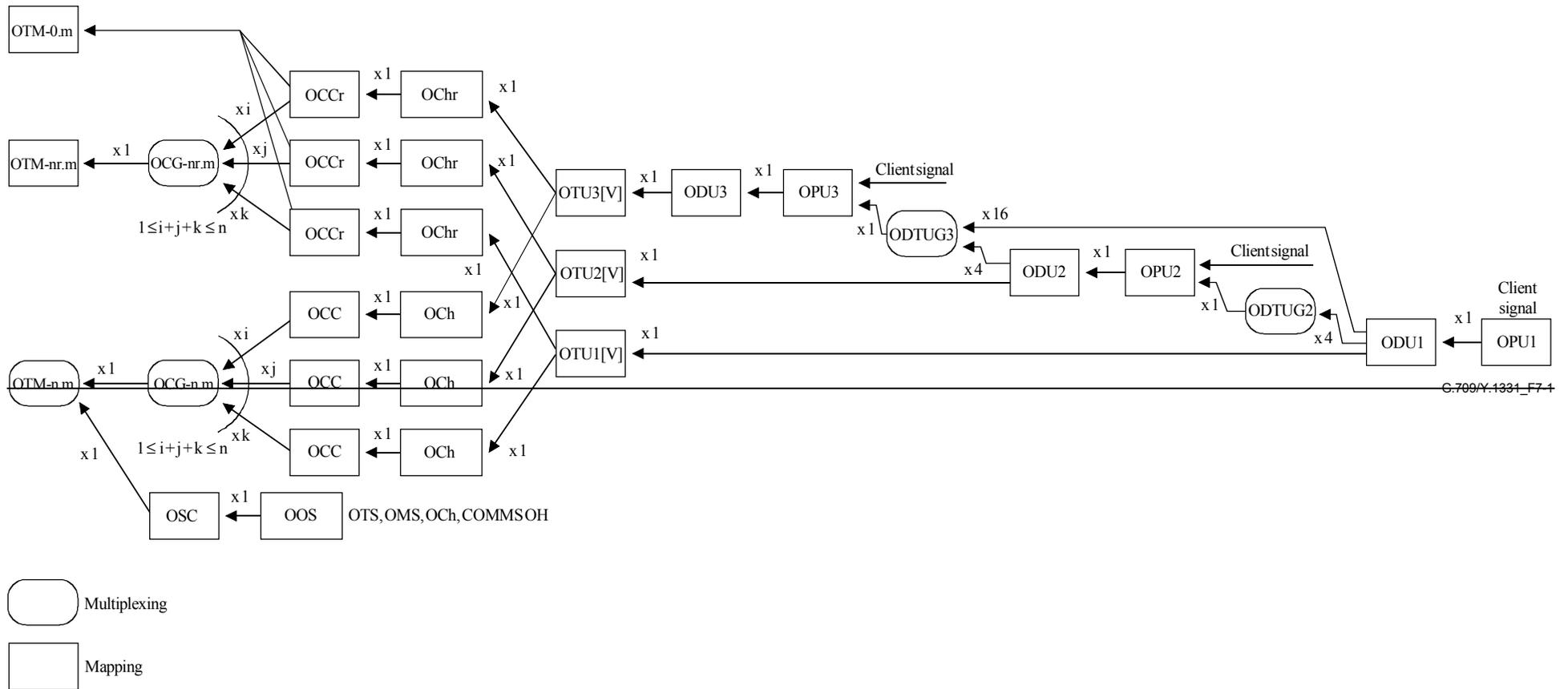
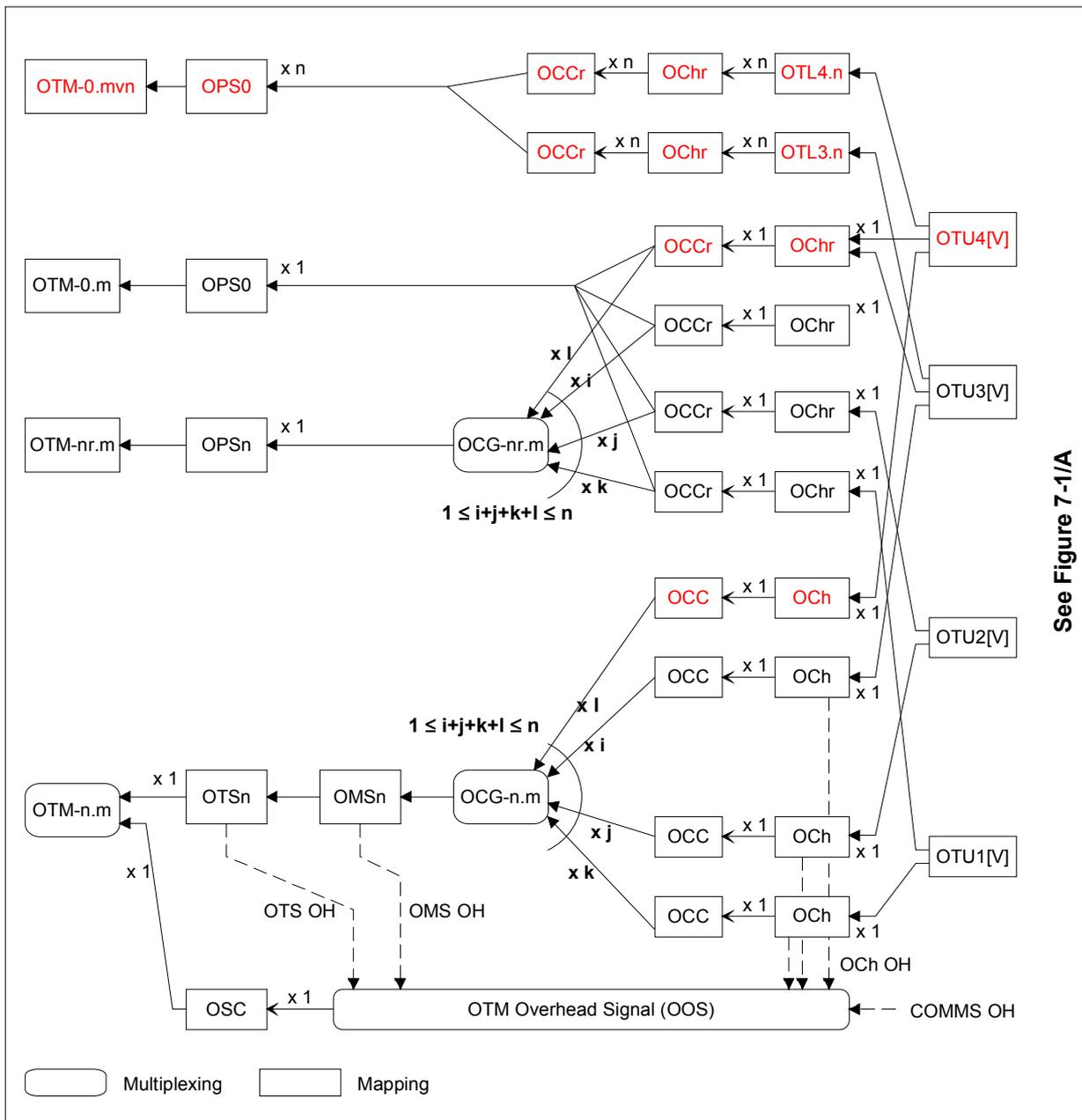


Figure 7-1/G.709/Y.1331 – OTM multiplexing and mapping structures





**Figure 7-1B – OTM multiplexing and mapping structures**

The OTS, OMS, OCh and COMMS overhead is inserted into the OOS using mapping and multiplexing techniques outside the scope of this Recommendation.

### 2.5) Clause 7.3 Bit rates and capacity

*Modify the text and tables as follows:*

The bit rates and capacity of the OTU<sub>k</sub> signals are defined in Table 7-1.

The bit rates and capacity of the ODU<sub>k</sub> signals are defined in Table 7-2.

The bit rates and capacity of the OPU<sub>k</sub> and OPU<sub>k</sub>-X<sub>v</sub> payload are defined in Table 7-3.

The OTU<sub>k</sub>/ODU<sub>k</sub>/OPU<sub>k</sub>/OPU<sub>k</sub>-X<sub>v</sub> frame periods are defined in Table 7-4.

**Table 7-1 – OTU types and capacity**

OTU type	OTU nominal bit rate	OTU bit-rate tolerance
OTU1	255/238 × 2 488 320 kbit/s	±20 ppm
OTU2	255/237 × 9 953 280 kbit/s	
OTU3	255/236 × 39 813 120 kbit/s	
OTU4	255/227 × 99 532 800 kbit/s	
NOTE 1 – The nominal OTUk rates are approximately: 2 666 057.143 kbit/s (OTU1), 10 709 225.316 kbit/s (OTU2), and 43 018 413.559 kbit/s (OTU3) and 111 809 973.568 kbit/s (OTU4).		
NOTE 2 – OTU0 and OTU2e are not specified in this Recommendation. ODU0 signals are to be transported over ODU1, ODU2, ODU3 or ODU4 signals and ODU2e signals are to be transported over ODU3 and ODU4 signals.		

**Table 7-2 – ODU types and capacity**

ODU type	ODU nominal bit rate	ODU bit-rate tolerance
ODU0	1 244 160 kbit/s	±20 ppm
ODU1	239/238 × 2 488 320 kbit/s	
ODU2	239/237 × 9 953 280 kbit/s	
ODU3	239/236 × 39 813 120 kbit/s	
ODU4	239/227 × 99 532 800 kbit/s	
ODU2e	239/237 × 10 312 500 kbit/s	±100 ppm
NOTE – The nominal ODUk rates are approximately: 2 498 775.126 kbit/s (ODU1), 10 037 273.924 kbit/s (ODU2), and 40 319 218.983 kbit/s (ODU3), 104 794 445.815 kbit/s (ODU4) and 10 399 525.316 kbit/s (ODU2e).		

**Table 7-3 – OPU types and capacity**

OPU type	OPU Payload nominal bit rate	OPU Payload bit rate tolerance
OPU0	238/239 × 1 244 160 kbit/s	±20 ppm
OPU1	2 488 320 kbit/s	
OPU2	238/237 × 9 953 280 kbit/s	
OPU3	238/236 × 39 813 120 kbit/s	
OPU4	238/227 × 99 532 800 kbit/s	
OPU2e	238/237 × 10 312 500 kbit/s	±100 ppm
OPU1-Xv	X × 2 488 320 kbit/s	±20 ppm
OPU2-Xv	X × 238/237 × 9 953 280 kbit/s	
OPU3-Xv	X × 238/236 × 39 813 120 kbit/s	
NOTE – The nominal OPUk Payload rates are approximately: 1 238 954.310 kbit/s (OPU0 Payload), 2 488 320.000 kbit/s (OPU1 Payload), 9 995 276.962 kbit/s (OPU2 Payload), and 40 150 519.322 kbit/s (OPU3 Payload), 104 355 975.330 (OPU4 Payload) and 10 356 012.658 kbit/s (OPU2e Payload). The nominal OPUk-Xv Payload rates are approximately: X × 2 488 320.000 kbit/s (OPU1-Xv Payload), X × 9 995 276.962 kbit/s (OPU2-Xv Payload), and X × 40 150 519.322 kbit/s (OPU3-Xv Payload).		

**Table 7-4 – OTUk/ODUk/OPUk frame periods**

<b>OTU/ODU/OPU type</b>	<b>Period (Note)</b>
<u>ODU0/OPU0</u>	<u>98.354 μs</u>
OTU1/ODU1/OPU1/OPU1-Xv	48.971 μs
OTU2/ODU2/OPU2/OPU2-Xv	12.191 μs
OTU3/ODU3/OPU3/OPU3-Xv	3.035 μs
<u>OTU4/ODU4/OPU4</u>	<u>1.168 μs</u>
<u>ODU2e/OPU2e</u>	<u>11.767 μs</u>
NOTE – The period is an approximated value, rounded to 3 digits.	

**Table 7-5 – OTL types and capacity**

<b><u>OTL type</u></b>	<b><u>OTL nominal bit rate</u></b>	<b><u>OTL bit rate tolerance</u></b>
<u>OTL3.4</u>	<u>255/236 × 9 953 280 kbit/s</u>	<u>±20 ppm</u>
<u>OTL4.4</u>	<u>255/227 × 24 883 200 kbit/s</u>	
<u>OTL4.10<sup>2</sup>(Note 2)</u>	<u>255/227 × 9 953 280 kbit/s</u>	
NOTE 1 – The nominal OTL rates are approximately: 10 754 603.390 kbit/s (OTL3.4), 27 952 493.392 kbit/s (OTL4.4), and 11 180 997.357 kbit/s (OTL4.10).		
NOTE 2 – IEEE 802.3 specifies 10-lane 100GBASE-R interfaces which have no corresponding physical layer specification in ITU-T, except that this may be the electrical interface to modules supporting an <u>OTM-0.4v4</u> .		

## 2.6) Clause 8 and subclauses

Modify the text and figures as follows:

### 8 Optical transport module (OTM-n.m, OTM-nr.m, OTM-0.m, OTM-0.mvn)

Two OTM structures are defined, one with full functionality and one with reduced functionality. For the IrDI only reduced functionality OTM interfaces are currently defined. Other full or reduced functionality OTM IrDIs are for further study.

#### 8.1 OTM with reduced functionality (OTM-0.m, OTM-nr.m, OTM-0.mvn)

The OTM-n supports n optical channels on a single optical span with 3R regeneration and termination of the OTUk[V] on each end. As 3R regeneration is performed on both sides of the OTM-0.m<sub>2</sub> ~~and~~ OTM-nr.m and OTM-0.mvn interfaces access to OTUk[V] overhead is available and maintenance/supervision of the interface is provided via this overhead. Therefore non-associated OTN overhead is not required across the OTM-0.m<sub>2</sub> ~~and~~ OTM-nr.m and OTM-0.mvn interfaces and an OSC/OOS is not supported.

~~Two~~ ~~Three~~ OTM interfaces classes with reduced functionality are defined, OTM-0.m<sub>2</sub> ~~and~~ OTM-n<sub>16</sub>r.m and OTM-0.mvn. Other reduced functionality interfaces classes are for further study.

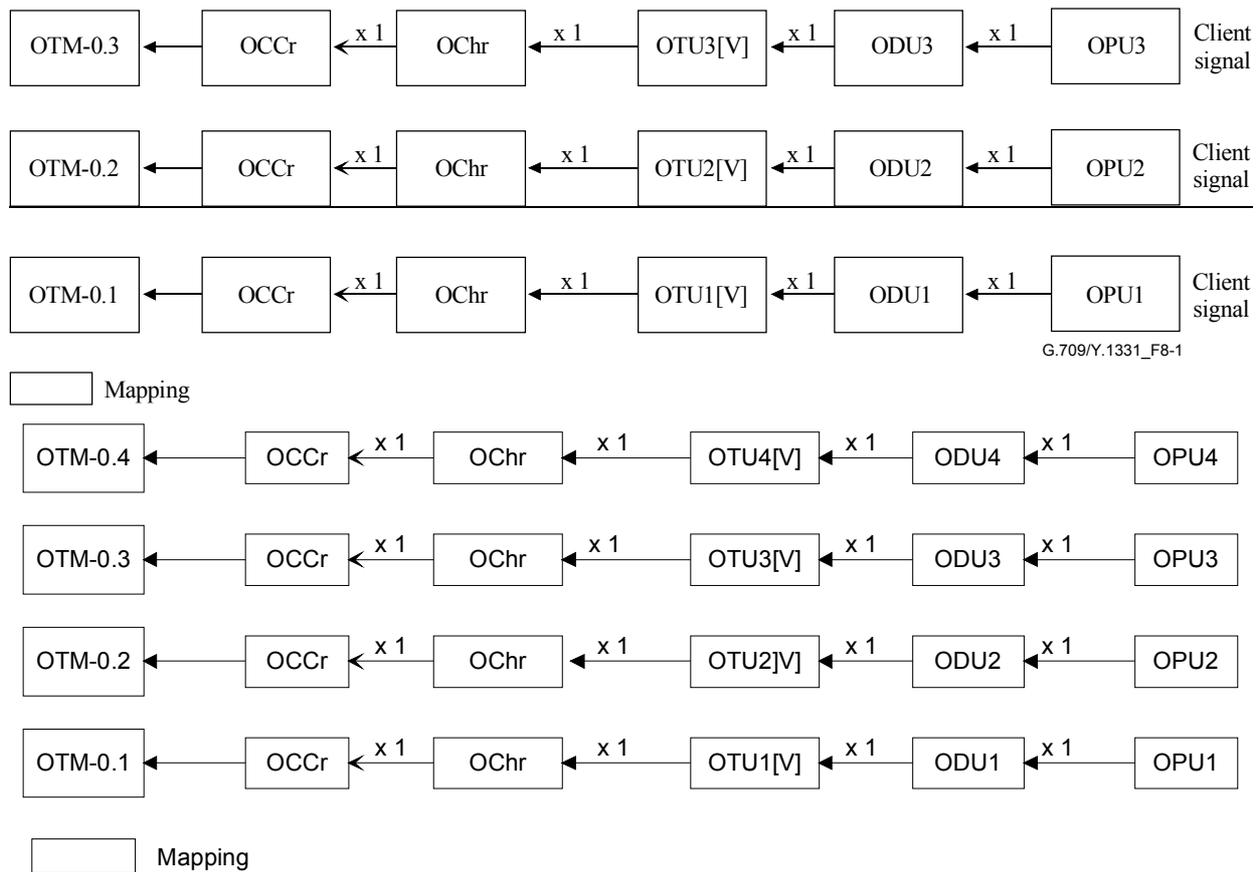
##### 8.1.1 OTM-0.m

The OTM-0.m supports a non-coloured optical channel on a single optical span with 3R regeneration at each end.

Three-Four OTM-0.m interface signals (see Figure 8-1) are defined, each carrying a single channel optical signal containing one OTUk[V] signal:

- OTM-0.1 (carrying an OTU1[V]);
- OTM-0.2 (carrying an OTU2[V]);
- OTM-0.3 (carrying an OTU3[V]).
- OTM-0.4 (carrying an OTU4[V]).

In generic terms: OTM-0.m.



**Figure 8-1 – OTM-0.m structure**

Figure 8-1 shows the relationship between various information structure elements that are defined below and illustrates possible mappings for the OTM-0.m.

An OSC is not present and there is no OOS either.

### 8.1.2 OTM-16r.m

#### 8.1.2.1 OTM-16r.m

This OTM-16r.m supports 16 optical channels on a single optical span with 3R regeneration at each end.

Several Six—OTM-16r interface signals are defined. The general form is OTM-16r.m. Some examples are:

- OTM-16r.1 (carrying  $i$  ( $i \leq 16$ ) OTU1[V] signals);
- OTM-16r.2 (carrying  $j$  ( $j \leq 16$ ) OTU2[V] signals);
- OTM-16r.3 (carrying  $k$  ( $k \leq 16$ ) OTU3[V] signals);
- OTM-16r.4 (carrying  $l$  ( $l \leq 16$ ) OTU4[V] signals);
- OTM-16r.1234 (carrying  $i$  ( $i \leq 16$ ) OTU1[V],  $j$  ( $j \leq 16$ ) OTU2[V],  $k$  ( $k \leq 16$ ) OTU3[V] and  $l$  ( $l \leq 16$ ) OTU4[V] signals with  $i + j + k + l \leq 16$ );
- OTM-16r.123 (carrying  $i$  ( $i \leq 16$ ) OTU1[V],  $j$  ( $j \leq 16$ ) OTU2[V] and  $k$  ( $k \leq 16$ ) OTU3[V] signals with  $i + j + k \leq 16$ );
- OTM-16r.12 (carrying  $i$  ( $i \leq 16$ ) OTU1[V] and  $j$  ( $j \leq 16$ ) OTU2[V] signals with  $i + j \leq 16$ );
- OTM-16r.23 (carrying  $j$  ( $j \leq 16$ ) OTU2[V] and  $k$  ( $k \leq 16$ ) OTU3[V] signals with  $j + k \leq 16$ );
- OTM-16r.34 (carrying  $k$  ( $k \leq 16$ ) OTU3[V] and  $l$  ( $l \leq 16$ ) OTU4[V] signals with  $k + l \leq 16$ );

~~in generic terms identified as OTM-16r.m.~~

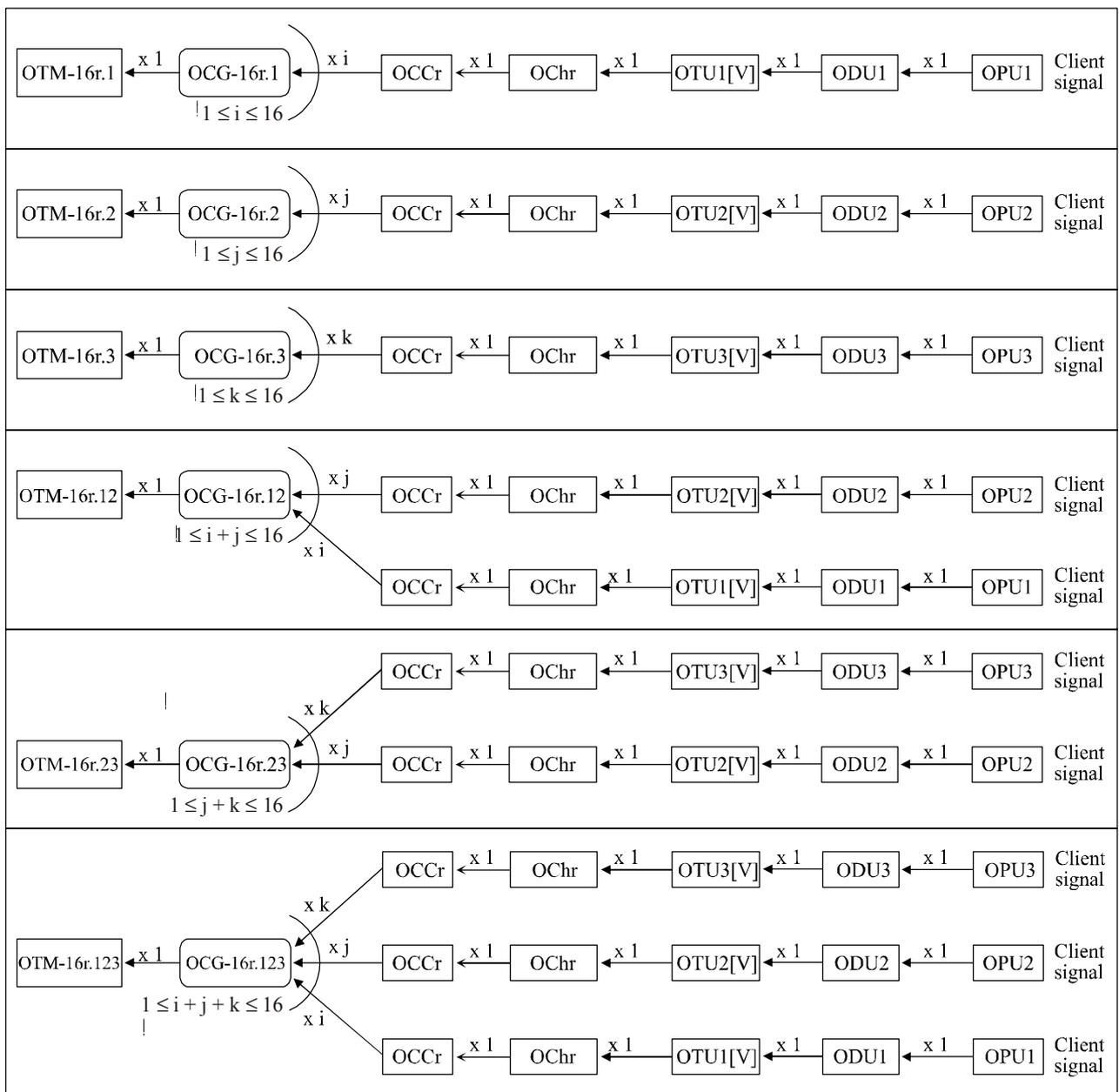
The OTM-16r.m signal (see Figure 6-5) is an OTM-nr.m signal with 16 optical channel carriers (OCCr) numbered OCCr #0 to OCCr #15. An optical supervisory channel (OSC) is not present and there is no OOS either.

At least one of the OCCrs is in service during normal operation and transporting an OTUk[V].

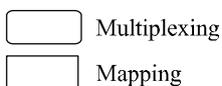
There is no predefined order in which the OCCrs are taken into service.

~~The six~~ Some examples of the defined OTM-16r.m interface signals and the OTM-16r.m multiplexing structure is are shown in Figure 8-2.

NOTE – OTM-16r.m OPS overhead is not defined. The interface will use the OTUk[V] SMOH in this multi-wavelength interface for supervision and management. OTM-16r.m connectivity (TIM) failure reports will be computed from the individual OTUk[V] reports by means of failure correlation in fault management. Refer to the equipment Recommendations for further details.



G.709/Y.1331-Amd.3(09)\_F8.2



**Figure 8-2 – OTM-16r.m multiplexing structure examples**

### 8.1.2.2 OTM-32r.m

This OTM-32r.m supports 32 optical channels on a single optical span with 3R regeneration at each end.

Several OTM-32r interface signals are defined. The general form is OTM-32r.m. Some examples are:

- OTM-32r.1 (carrying  $i$  ( $i \leq 32$ ) OTU1[V] signals);
- OTM-32r.2 (carrying  $j$  ( $j \leq 32$ ) OTU2[V] signals);
- OTM-32r.3 (carrying  $k$  ( $k \leq 32$ ) OTU3[V] signals);
- OTM-32r.4 (carrying  $l$  ( $l \leq 32$ ) OTU4[V] signals);

- OTM-32r.1234 (carrying  $i$  ( $i \leq 32$ ) OTU1[V],  $j$  ( $j \leq 32$ ) OTU2[V],  $k$  ( $k \leq 32$ ) OTU3[V] and  $l$  ( $l \leq 32$ ) OTU4[V] signals with  $i + j + k + l \leq 32$ );
- OTM-32r.123 (carrying  $i$  ( $i \leq 32$ ) OTU1[V],  $j$  ( $j \leq 32$ ) OTU2[V] and  $k$  ( $k \leq 32$ ) OTU3[V] signals with  $i + j + k \leq 32$ );
- OTM-32r.12 (carrying  $i$  ( $i \leq 32$ ) OTU1[V] and  $j$  ( $j \leq 32$ ) OTU2[V] signals with  $i + j \leq 32$ );
- OTM-32r.23 (carrying  $j$  ( $j \leq 32$ ) OTU2[V] and  $k$  ( $k \leq 32$ ) OTU3[V] signals with  $j + k \leq 32$ );
- OTM-32r.34 (carrying  $k$  ( $k \leq 32$ ) OTU3[V] and  $l$  ( $l \leq 32$ ) OTU4[V] signals with  $k + l \leq 32$ );

in generic terms identified as OTM-32r.m.

The OTM-32r.m signal (see Figure 6-5) is an OTM-nr.m signal with 32 optical channel carriers (OCCr) numbered OCCr #0 to OCCr #31. An optical supervisory channel (OSC) is not present and there is no OOS either.

At least one of the OCCrs is in service during normal operation and transporting an OTUk[V].

There is no predefined order in which the OCCrs are taken into service.

NOTE – OTM-32r.m OPS overhead is not defined. The interface will use the OTUk[V] SMOH in this multi-wavelength interface for supervision and management. OTM-32r.m connectivity (TIM) failure reports will be computed from the individual OTUk[V] reports by means of failure correlation in fault management. Refer to the relevant Recommendations for each equipment type for further details.

### **8.1.3 OTM-0.mvn**

The OTM-0.mvn supports a multi-lane optical signal on a single optical span with 3R regeneration at each end.

Two OTM-0.mvn interface signals are defined, each carrying a four-lane optical signal containing one OTUk signal striped across the four optical lanes:

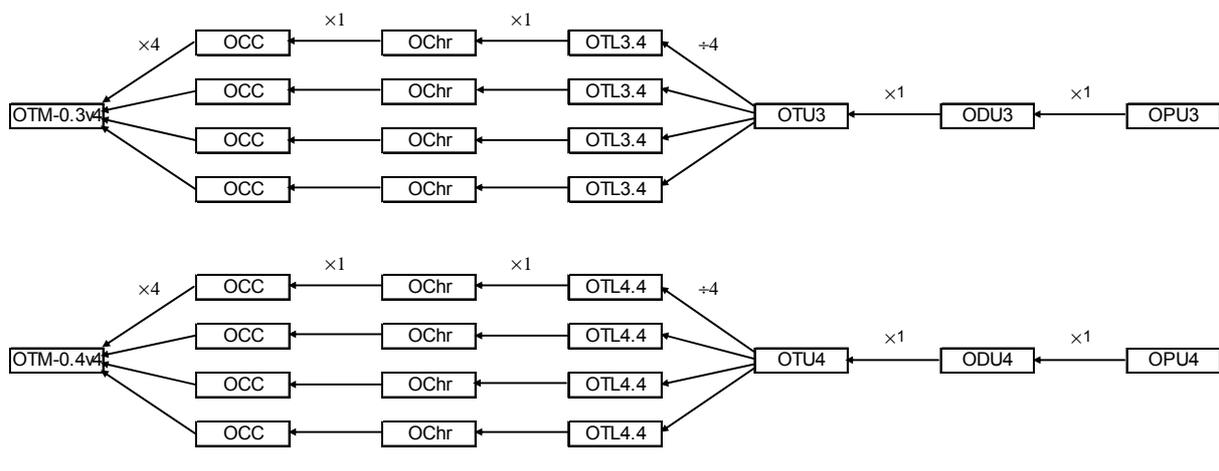
- OTM-0.3v4 (carrying an OTU3).
- OTM-0.4v4 (carrying an OTU4).

In generic terms: OTM-0.mvn.

The optical lanes are numbered of each OCCx,  $x = 0$  to  $n - 1$  where the  $x$  represents the optical lane number of the corresponding G.959.1 or G.695 application code for the multilane applications.

Figure 8-3 shows the relationship between various information structure elements for the OTM-0.3v4 and OTM-0.4v4.

An OSC is not present and there is no OOS either.



**Figure 8-3 – OTM-0.4v3 and OTM-0.4v4 structure**

## 8.2 OTM with full functionality (OTM-n.m)

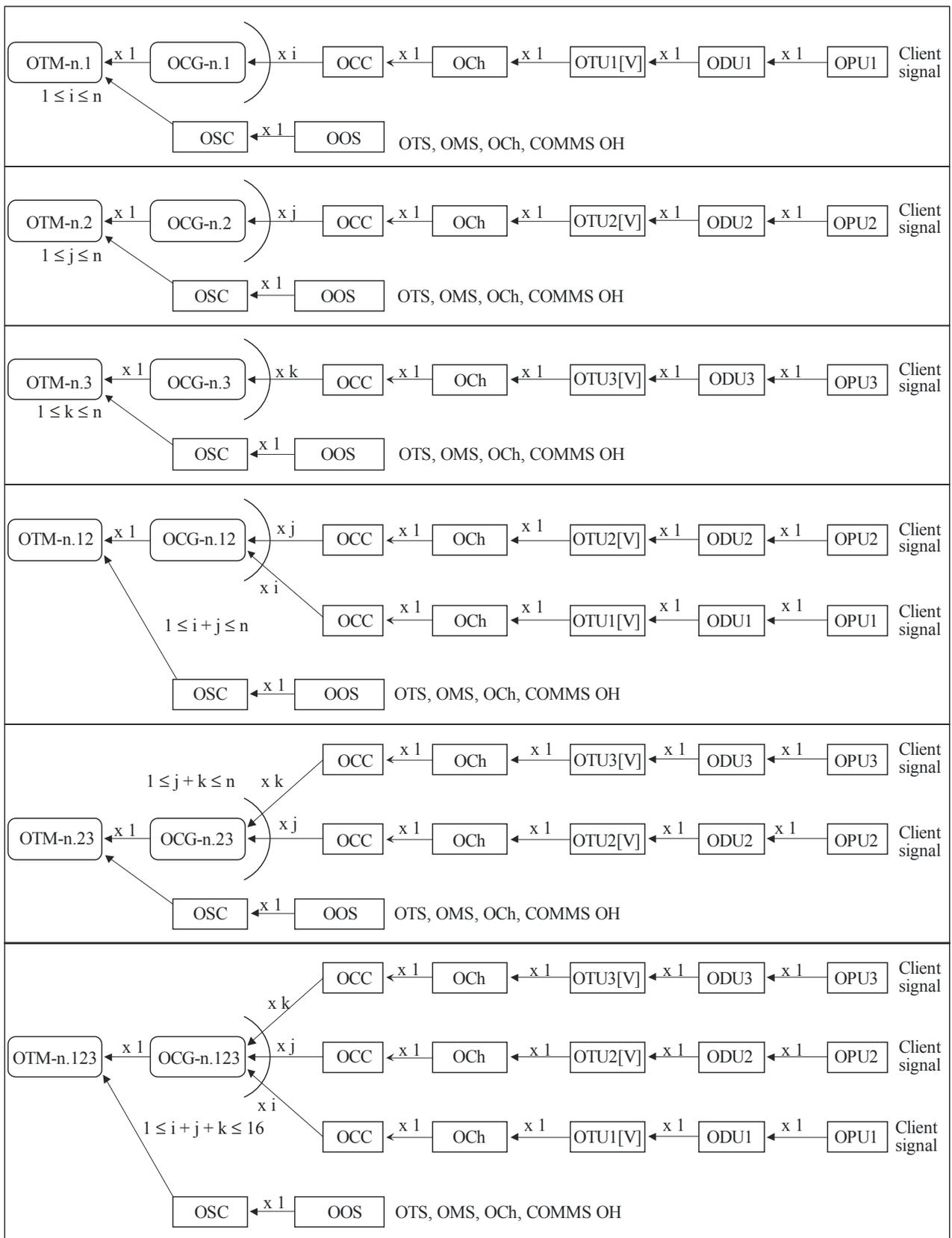
The OTM-n.m interface supports up to n optical channels for single or multiple optical spans. 3R regeneration is not required at the interface.

Several OTM-n interface signals are defined. The general form is OTM-n.m. Some examples are:

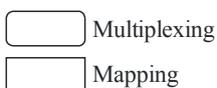
- OTM-n.1 (carrying i ( $i \leq n$ ) OTU1[V] signals);
- OTM-n.2 (carrying j ( $j \leq n$ ) OTU2[V] signals);
- OTM-n.3 (carrying k ( $k \leq n$ ) OTU3[V] signals);
- OTM-n.4 (carrying l ( $l \leq n$ ) OTU4[V] signals);
- OTM-n.1234 (carrying i ( $i \leq n$ ) OTU1[V], j ( $j \leq n$ ) OTU2[V], k ( $k \leq n$ ) OTU3[V] and l ( $l \leq n$ ) OTU4[V] signals with  $i + j + k + l \leq n$ );
- OTM-n.123 (carrying i ( $i \leq n$ ) OTU1[V], j ( $j \leq n$ ) OTU2[V] and k ( $k \leq n$ ) OTU3[V] signals with  $i + j + k \leq n$ );
- OTM-n.12 (carrying i ( $i \leq n$ ) OTU1[V] and j ( $j \leq n$ ) OTU2[V] signals with  $i + j \leq n$ );
- OTM-n.23 (carrying j ( $j \leq n$ ) OTU2[V] and k ( $k \leq n$ ) OTU3[V] signals with  $j + k \leq n$ );
- OTM-n.34 (carrying k ( $k \leq n$ ) OTU3[V] and l ( $l \leq n$ ) OTU4[V] signals with  $k + l \leq n$ );

in generic terms identified as OTM-n.m.

An OTM-n.m interface signal contains up to "n" OCCs associated with the lowest bit rate that is supported as indicated by m and an OSC (see Figure 8-4). It is possible that a reduced number of higher bit rate capable OCCs are supported. The value of "n", "m" and the OSC are not defined in this Recommendation.



G.709/Y.1331-Amd.3(09)\_F8-4



**Figure 8-4 – OTM-n.m multiplexing structure examples**

## 2.7) Clause 9 and subclauses

*Modify the text as follows:*

### 9 Physical specification of the ONNI

#### 9.1 OTM-0.m

Specifications for physical optical characteristics of the OTM-0.1, OTM-0.2 and OTM-0.3 signals are contained in ITU-T Recs G.959.1 and G.693.

Specifications for physical optical characteristics of the OTM-0.4 are for further study.

#### 9.2 OTM-16nr.m

##### 9.2.1 OTM-16r.m

Specifications for physical optical characteristics of the OTM-16r.1, ~~and OTM-16r.2 and OTM-16r.12~~ signals are contained in ITU-T Rec. G.959.1.

Specifications for physical optical characteristics of ~~the OTM-16r.3, OTM-16r.12, OTM-16r.23 and OTM-16r.123~~ other OTM-16r.m are for further study.

##### 9.2.2 OTM-32r.m

Specifications for physical optical characteristics of the OTM-32r.1, OTM-32r.2, and OTM-32r.12 signals are contained in ITU-T Rec. G.959.1.

Specifications for physical optical characteristics of other OTM-32r.m are for further study.

#### 9.3 OTM-n.m

Specifications for physical optical characteristics of the OTM-n.m are vendor specific and outside the scope of this Recommendation.

#### 9.4 OTM-0.mvn

Specifications for physical optical characteristics of the OTM-0.3v4 and OTM-0.4v4 signals will be specified in ITU-T Recs G.695 and G.959.1, respectively.

## 2.8) Clause 10

*Add the following subclause:*

### 10.3 OChr-n

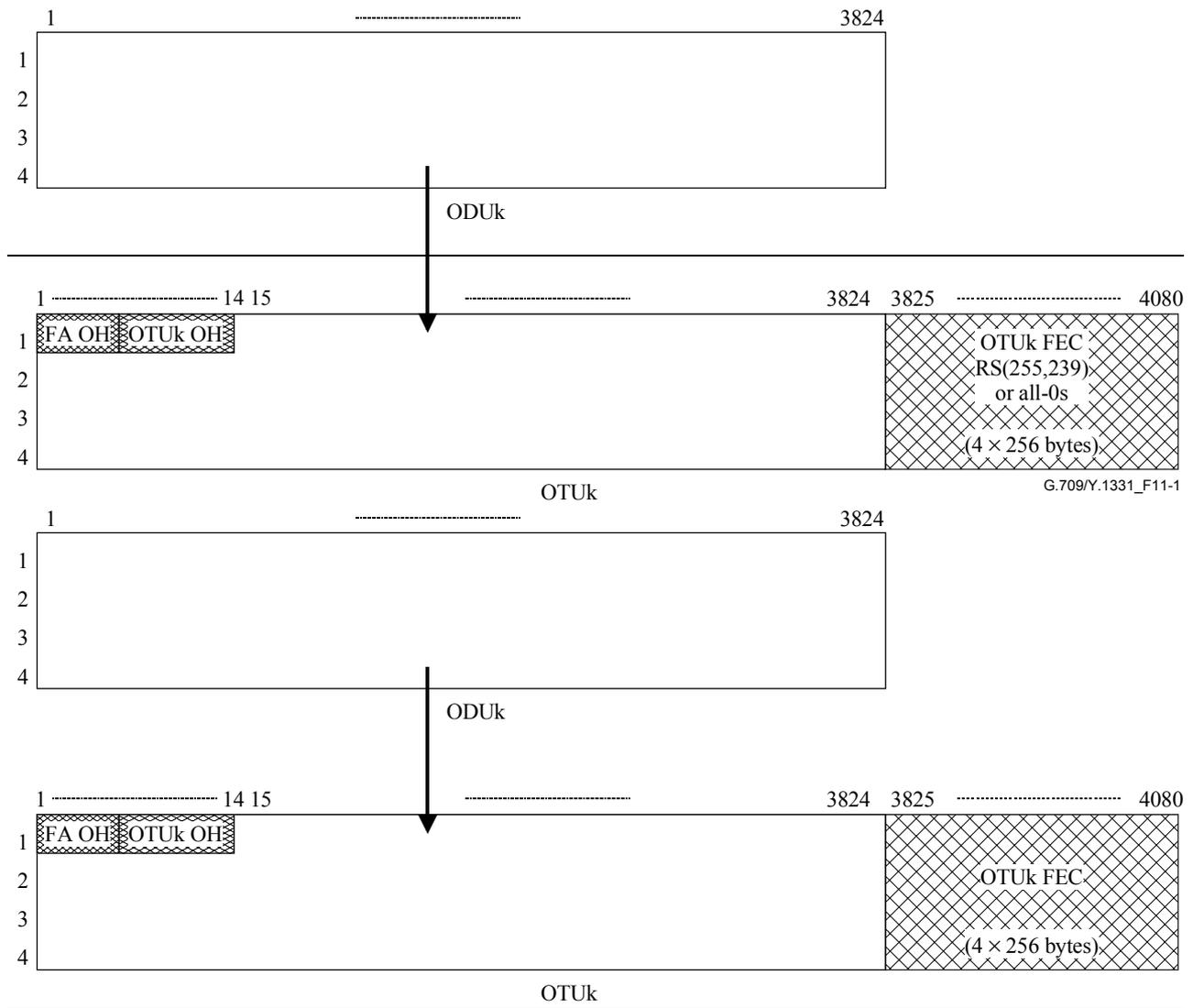
The OChr-n is a bundle of n optical channels used for the implementation of a parallel single-channel OTM-0.mvn signal, as shown conceptually in Figures 6-6 and 8-3. An OTUk is inversely multiplexed across n OTLs, each transported by an OChr that is part of this bundle. The inverse multiplexing process is described in Annex C.

## 2.9) Clause 11.1 OTUk frame structure

*Modify the text as follows:*

The OTUk (k = 1,2,3,4) frame structure is based on the ODUk frame structure and extends it with a forward error correction (FEC) as shown in Figure 11-1. 256 columns are added to the ODUk frame for the FEC and the reserved overhead bytes in row 1, columns 8 to 14 of the ODUk overhead are used for OTUk specific overhead, resulting in an octet-based block frame structure with four rows and 4080 columns. The MSB in each octet is bit 1, the LSB is bit 8.

NOTE 1 – This Recommendation does not specify an OTUk frame structure for k = 0 or k = 2e.



**Figure 11-1 – OTUk frame structure**

The bit rates of the OTUk signals are defined in Table 7-1.

The OTUk ( $k = 1,2,3$ ) forward error correction (FEC) contains the Reed-Solomon RS(255,239) FEC codes. If no FEC is used, fixed stuff bytes (all-0s pattern) are to be used.

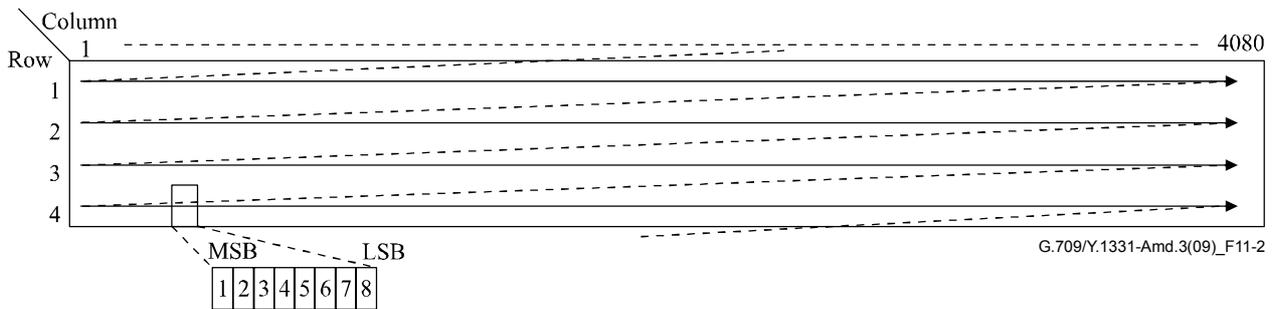
The RS(255,239) FEC code shall be computed as specified in Annex A.

For interworking of equipment supporting FEC, with equipment not supporting FEC (inserting fixed stuff all-0s pattern in the OTUk ( $k = 1,2,3$ ) FEC area), the FEC supporting equipment shall support the capability to disable the FEC decoding process (ignore the content of the OTUk ( $k = 1,2,3$ ) FEC).

The OTUk ( $k = 4$ ) forward error correction (FEC) is for further study.

NOTE 2 – The OTU4 FEC is mandatory.

The transmission order of the bits in the OTUk frame is left to right, top to bottom, and MSB to LSB (see Figure 11-2).

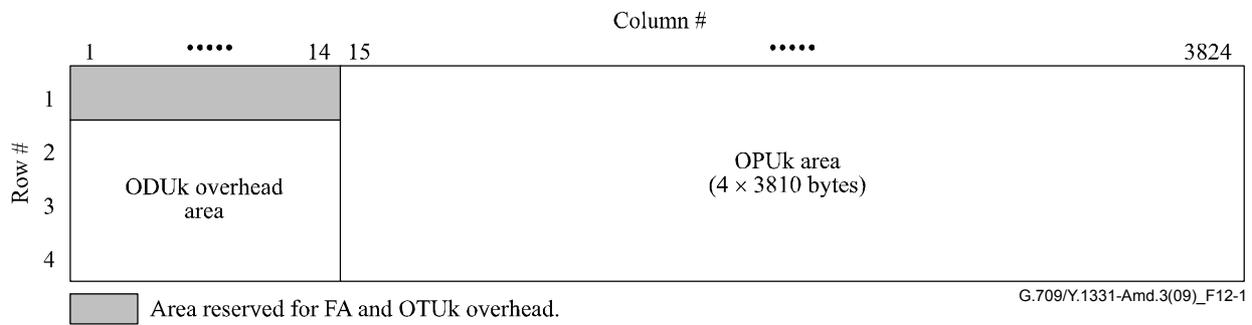


**Figure 11-2 – Transmission order of the OTUk frame bits**

**2.10) Clause 12.1 ODUk frame structure**

*Modify the text as follows:*

The ODUk ( $k = \underline{0}, \underline{1}, \underline{2}, \underline{2e}, \underline{3}, \underline{4}$ ) frame structure is shown in Figure 12-1. It is organized in an octet-based block frame structure with four rows and 3824 columns.



**Figure 12-1 – ODUk frame structure**

The two main areas of the ODUk frame are:

- ODUk overhead area;
- OPUk area.

Columns 1 to 14 of the ODUk are dedicated to ODUk overhead area.

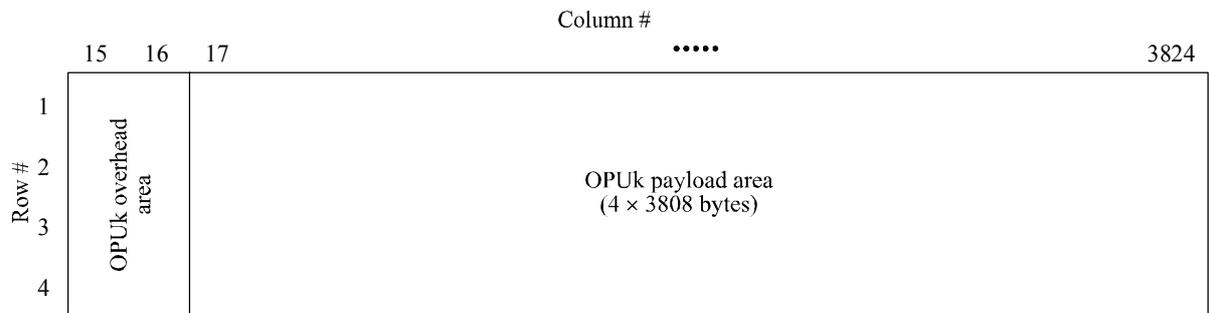
NOTE – Columns 1 to 14 of row 1 are reserved for frame alignment and OTUk specific overhead.

Columns 15 to 3824 of the ODUk are dedicated to OPUk area.

**2.11) Clause 13 Optical channel payload unit (OPUk)**

*Modify the text as follows:*

The OPUk ( $k = \underline{0}, \underline{1}, \underline{2}, \underline{2e}, \underline{3}, \underline{4}$ ) frame structure is shown in Figure 13-1. It is organized in an octet-based block frame structure with four rows and 3810 columns.



G.709/Y.1331-Amd.3(09)\_F13-1

**Figure 13-1 – OPUk frame structure**

The two main areas of the OPUk frame are:

- OPUk overhead area;
- OPUk payload area.

Columns 15 to 16 of the OPUk are dedicated to OPUk overhead area.

Columns 17 to 3824 of the OPUk are dedicated to OPUk payload area.

NOTE – OPUk column numbers are derived from the OPUk columns in the ODUk frame.

**2.12) Clause 15.8.2.4 ODUk automatic protection switching and protection communication channel (APS/PCC)**

*Modify Table 15-6 as follows:*

**Table 15-6 – Multiframe to allow separate APS/PCC for each monitoring level**

<b>MFAS bits</b> 6 7 8	<b>APS/PCC channel applies to connection monitoring level</b>	<b>Protection scheme using the APS/PCC channel (Note 1)</b>
0 0 0	ODUk Path	ODUk SNC/N
0 0 1	ODUk TCM1	ODUk SNC/S, ODUk SNC/N
0 1 0	ODUk TCM2	ODUk SNC/S, ODUk SNC/N
0 1 1	ODUk TCM3	ODUk SNC/S, ODUk SNC/N
1 0 0	ODUk TCM4	ODUk SNC/S, ODUk SNC/N
1 0 1	ODUk TCM5	ODUk SNC/S, ODUk SNC/N
1 1 0	ODUk TCM6	ODUk SNC/S, ODUk SNC/N
1 1 1	<del>OTUk Section</del> ODUk Server layer trail (Note 2)	ODUk SNC/I

NOTE 1 – An APS channel may be used by more than one protection scheme and/or protection scheme instance. In case of nested protection schemes, care should be taken when an ODUk protection is to be set up in order not to interfere with the APS channel usage of another ODUk protection on the same connection monitoring level, e.g., protection can only be activated if that APS channel of the level is not already being used.

NOTE 2 – Examples of ODUk server layer trails are an OTUk or a HO ODUk (e.g., an ODU3 transporting an ODU1).

### 2.13) Clause 15.9.2.1.1 OPUk payload type (PT)

Modify Table 15-8 as follows:

**Table 15-8 – Payload type code points**

<b>MSB 1 2 3 4</b>	<b>LSB 5 6 7 8</b>	<b>Hex code (Note 1)</b>	<b>Interpretation</b>
0 0 0 0	0 0 0 1	01	Experimental mapping (Note 3)
0 0 0 0	0 0 1 0	02	Asynchronous CBR mapping, see 17.1
0 0 0 0	0 0 1 1	03	Bit synchronous CBR mapping, see 17.1
0 0 0 0	0 1 0 0	04	ATM mapping, see 17.2
0 0 0 0	0 1 0 1	05	GFP mapping, see 17.3
0 0 0 0	0 1 1 0	06	Virtual Concatenated signal, see clause 18 (Note 5)
<u>0 0 0 0</u>	<u>0 1 1 1</u>	<u>07</u>	<u>1000BASE-X into OPU0 mapping, see 17.7.1</u>
<u>0 0 0 0</u>	<u>1 0 0 0</u>	<u>08</u>	<u>FC-1200 into OPU2e mapping, see 17.7.2</u>
<u>0 0 0 0</u>	<u>1 0 0 1</u>	<u>09</u>	<u>GFP mapping into Extended OPU2 payload, see 17.3.1 (Note 6)</u>
0 0 0 1	0 0 0 0	10	Bit stream with octet timing mapping, see 17.5.1
0 0 0 1	0 0 0 1	11	Bit stream without octet timing mapping, see 17.5.2
0 0 1 0	0 0 0 0	20	ODU multiplex structure, see clause 19
0 1 0 1	0 1 0 1	55	Not available (Note 2)
0 1 1 0	0 1 1 0	66	Not available (Note 2)
1 0 0 0	x x x x	80-8F	Reserved codes for proprietary use (Note 4)
1 1 1 1	1 1 0 1	FD	NULL test signal mapping, see 17.4.1
1 1 1 1	1 1 1 0	FE	PRBS test signal mapping, see 17.4.2
1 1 1 1	1 1 1 1	FF	Not available (Note 2)
<p>NOTE 1 – There are 2236 spare codes left for future international standardization. Refer to Annex A/G.806 for the procedure to obtain one of these codes for a new payload type.</p> <p>NOTE 2 – These values are excluded from the set of available code points. These bit patterns are present in ODUk maintenance signals.</p> <p>NOTE 3 – Value "01" is only to be used for experimental activities in cases where a mapping code is not defined in this table. Refer to Annex A/G.806 for more information on the use of this code.</p> <p>NOTE 4 – These 16 code values will not be subject to further standardization. Refer to Annex A/G.806 for more information on the use of these codes.</p> <p>NOTE 5 – For the payload type of the virtual concatenated signal a dedicated payload type overhead (vcPT) is used, see clause 18.</p> <p><u>NOTE 6 – Supplement 43 (02/2008) to the ITU-T G-series of Recommendations indicated that this mapping use Payload Type 87.</u></p>			

## 2.14) Clause 17.1

Modify the title and text as follows:

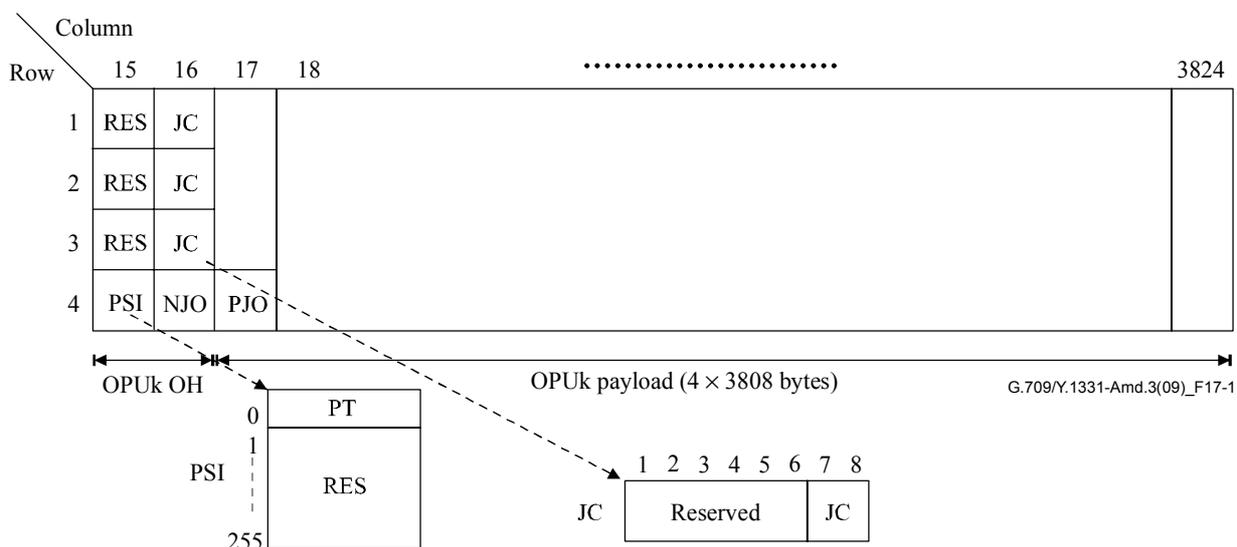
### 17.1 Mapping of CBR2G5, CBR10G, CBR10G3 and CBR40G signals ~~(e.g., STM-16/64/256)~~ into OPUk

Mapping of a CBR2G5, CBR10G or CBR40G signal (with up to  $\pm 20$  ppm bit-rate tolerance) into an OPUk ( $k = 1,2,3$ ) may be performed according to two different modes (asynchronous and bit synchronous) based on one generic OPUk frame structure (see Figure 17-1). Mapping of a CBR10G3 signal (with up to  $\pm 100$  ppm bit-rate tolerance) into an OPUk ( $k = 2e$ ) is performed using bit synchronous mapping.

NOTE 1 – Examples of ~~such~~ CBR2G5, CBR10G and CBR40G signals are STM-16, STM-64 and STM-256. An example of a CBR10G3 signal is 10GBASE-R.

NOTE 2 – The maximum bit-rate tolerance between OPUk ( $k = 1,2,3$ ) and the client signal clock, which can be accommodated by ~~this~~ the asynchronous mapping scheme, is  $\pm 65$  ppm. With a bit-rate tolerance of  $\pm 20$  ppm for the OPUk clock, the client signal's bit-rate tolerance can be  $\pm 45$  ppm.

NOTE 3 – For OPUk ( $k = 1,2,3$ ), the clock tolerance is  $\pm 20$  ppm. For OPU2e, the clock tolerance is  $\pm 100$  ppm and asynchronous mapping cannot be supported with this justification overhead.



**Figure 17-1 – OPUk frame structure for the mapping of a CBR2G5, CBR10G or CBR40G signal**

The OPUk overhead for these mappings consists of a payload structure identifier (PSI) including the payload type (PT) and 255 bytes reserved for future international standardization (RES), three justification control (JC) bytes, one negative justification opportunity (NJO) byte, and three bytes reserved for future international standardization (RES). The JC bytes consist of two bits for justification control and six bits reserved for future international standardization.

The OPUk payload for these mappings consists of  $4 \times 3808$  bytes, including one positive justification opportunity (PJO) byte.

The justification control (JC) signal, which is located in rows 1, 2 and 3 of column 16, bits 7 and 8, is used to control the two justification opportunity bytes NJO and PJO that follow in row 4.

The asynchronous and bit synchronous mapping processes generate the JC, NJO and PJO according to Tables 17-1 and 17-2, respectively. The demapping process interprets JC, NJO and PJO according to Table 17-3. Majority vote (two out of three) shall be used to make the justification decision in the demapping process to protect against an error in one of the three JC signals.

**Table 17-1 – JC, NJO and PJO generation by asynchronous mapping process**

<b>JC</b> <b>bits</b> 7 8	<b>NJO</b>	<b>PJO</b>
0 0	justification byte	data byte
0 1	data byte	data byte
1 0	not generated	
1 1	justification byte	justification byte

**Table 17-2 – JC, NJO and PJO generation by bit synchronous mapping process**

<b>JC</b> <b>bits</b> 7 8	<b>NJO</b>	<b>PJO</b>
0 0	justification byte	data byte
0 1	not generated	
1 0		
1 1		

**Table 17-3 – JC, NJO and PJO interpretation**

<b>JC</b> <b>bits</b> 7 8	<b>NJO</b>	<b>PJO</b>
0 0	justification byte	data byte
0 1	data byte	data byte
1 0 (Note)	justification byte	data byte
1 1	justification byte	justification byte
NOTE – A mapper circuit does not generate this code. Due to bit errors a demapper circuit might receive this code.		

The value contained in NJO and PJO when they are used as justification bytes is all-0s. The receiver is required to ignore the value contained in these bytes whenever they are used as justification bytes.

During a signal fail condition of the incoming CBR2G5, CBR10G or CBR40G client signal (e.g., in the case of a loss of input signal), this failed incoming signal is replaced by the generic-AIS signal as specified in 16.6.1, and is then mapped into the OPUk.

During a signal fail condition of the incoming 10GBASE-R type CBR10G3 client signal (e.g., in the case of a loss of input signal), this failed incoming 10GBASE-R signal is replaced by a stream of 66B blocks, with each block carrying two Local Fault sequence ordered sets (as specified in IEEE Std. 802.3). This replacement signal is then mapped into the OPU2e.

During signal fail condition of the incoming ODUk/OPUk signal (e.g., in the case of an ODUk-AIS, ODUk-LCK, ODUk-OCI condition) the generic-AIS pattern as specified in 16.6.1 is generated as a replacement signal for the lost CBR2G5, CBR10G or CBR40G signal.

During signal fail condition of the incoming ODU2e/OPU2e signal (e.g., in the case of an ODU2e-AIS, ODU2e-LCK, ODU2e-OCI condition) a stream of 66B blocks, with each block carrying two Local Fault sequence ordered sets (as specified in IEEE Std. 802.3) is generated as a replacement signal for the lost 10GBASE-R signal.

NOTE 4 – Local Fault sequence ordered set is /K28.4/D0.0/D0.0/D1.0/. The 66B block contains the following value SH=10 0x55 00 00 01 00 00 00 01.

NOTE 5 – Equipment developed prior to this amendment may generate a different 10GBASE-R replacement signal (e.g., Generic-AIS) than the Local Fault sequence ordered set.

### **Asynchronous mapping**

The OPUk signal for the asynchronous mapping is created from a locally generated clock (within the limits specified in Table 17-3), which is independent of the CBR2G5, CBR10G or CBR40G (i.e.,  $4^{(k-1)} \times 2\,488\,320$  kbit/s ( $k = 1,2,3$ )) client signal.

The CBR2G5, CBR10G, CBR40G (i.e.,  $4^{(k-1)} \times 2\,488\,320$  kbit/s ( $k = 1,2,3$ )) signal is mapped into the OPUk using a positive/negative/zero (pnz) justification scheme.

### **Bit synchronous mapping**

The OPUk clock for the bit synchronous mapping is derived from the CBR2G5, CBR10G, ~~or~~ CBR40G or CBR10G3 (i.e.,  $4^{(k-1)} \times 2\,488\,320$  kbit/s ( $k = 1,2,3$ )) client signal. During signal fail conditions of the incoming CBR2G5, CBR10G, ~~or~~ CBR40G or CBR10G3 signal (e.g., in the case of loss of input signal), the OPUk payload signal bit rate shall be within the limits specified in Table 17-3 and neither a frequency nor frame phase discontinuity shall be introduced. The resynchronization on the incoming CBR2G5, CBR10G, ~~or~~ CBR40G or CBR10G3 signal shall be done without introducing a frequency or frame phase discontinuity.

The CBR2G5, CBR10G, ~~or~~ CBR40G or CBR10G3 (i.e.,  $4^{(k-1)} \times 2\,488\,320$  kbit/s ( $k = 1,2,3$ )) signal is mapped into the OPUk without using the justification capability within the OPUk frame: NJO contains a justification byte, PJO contains a data byte, and the JC signal is fixed to 00.

## **2.15) Clause 17.1**

*Add the following subclause:*

### **17.1.4 Mapping a CBR10G3 signal (e.g., 10GBASE-R) into OPU2e**

Groups of 8 successive bits (not necessarily being a byte) of the CBR10G3 signal are bit-synchronously mapped into a data (D) byte of the OPU2e (see Figure 17-4 *bis*). Sixty-four fixed stuff (FS) bytes are added in columns 1905 to 1920.

NOTE – The NJO byte will always carry a stuff byte, the PJO byte will always carry a data (D) byte and the JC bytes will always carry the all-0's pattern.

Row #	Column #			Column #			Column #		
	15	16	17	1904	1905	1920	1921	3824	
1	RES	RES	JC	118 × 16D			16FS	119 × 16D	
2	RES	RES	JC	118 × 16D			16FS	119 × 16D	
3	RES	JC		118 × 16D			16FS	119 × 16D	
4	PSI	NJO	PJO	15D + 117 × 16D			16FS	119 × 16D	

G.709/Y.1331-Amd.3(09)\_F17-4bis

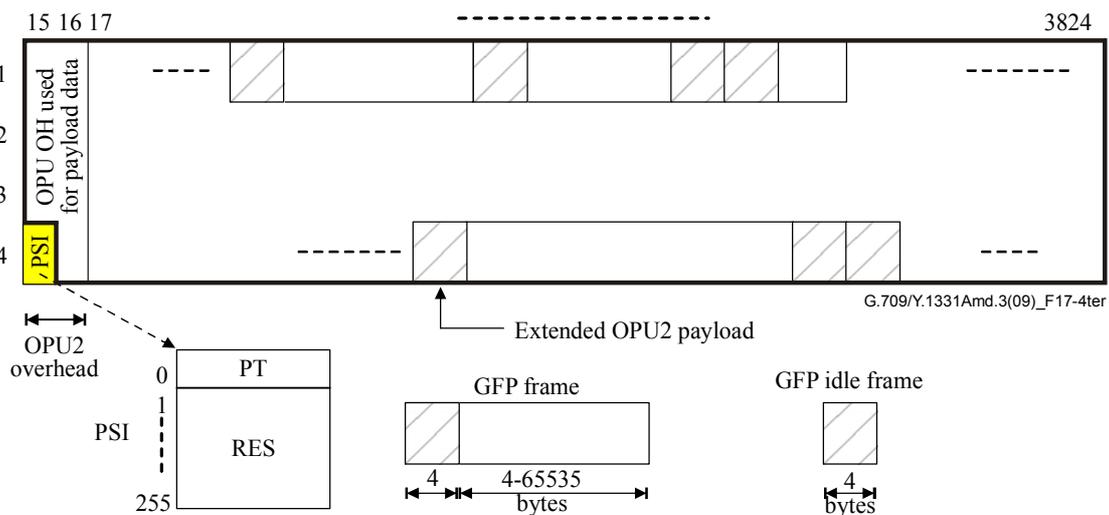
**Figure 17-4 bis – Mapping of a CBR10G3 signal into OPU2e**

## 2.16) Clause 17.3

Add the following subclause:

### 17.3.1 Mapping of GFP frames into Extended OPU2 payload area

The mapping of generic framing procedure (GFP) frames in an Extended OPU2 payload area is performed by aligning the byte structure of every GFP frame with the byte structure of the Extended OPU2 payload (see Figure 17-4 ter). Since the GFP frames are of variable length (the mapping does not impose any restrictions on the maximum frame length), a frame may cross the OPU2 frame boundary.



**Figure 17-4 ter – OPU2 frame structure and mapping of GFP frames into extended OPU2 payload area**

GFP frames arrive as a continuous bit stream with a capacity that is identical to the OPU2 payload area, due to the insertion of GFP-Idle frames at the GFP encapsulation stage. The GFP frame stream is scrambled during encapsulation.

NOTE – There is no rate adaptation or scrambling required at the mapping stage; this is performed by the GFP encapsulation process.

The OPU2 overhead for the GFP mapping consists of a payload structure identifier (PSI) including the payload type (PT) and 255 bytes reserved for future international standardization (RES).

The extended OPU2 payload for the GFP mapping consists of  $4 \times 3808$  bytes from the OPU2 payload plus 7 bytes from the OPU2 overhead.

## 2.17) Clause 17

*Add the following subclauses:*

### 17.7 Mapping a 1000BASE-X and FC-1200 signal via timing transparent transcoding into OPUk

#### 17.7.1 Mapping a 1000BASE-X signal into OPU0

Mapping of a 1 250 000 kbit/s 1000BASE-X signal (with up to  $\pm 100$  ppm bit-rate tolerance) into an OPU0 is performed by an asynchronous mapping using a sigma-delta based justification distribution after synchronous transcoding of the 1 250 000 kbit/s 8B/10B code word stream into a  $15/16 \times 1\,250\,000$  kbit/s  $\pm 100$  ppm (approximately 1 171 875 kbit/s  $\pm 100$  ppm) 75-octet GFP-T frame stream.

The ingress 1000BASE-X signal is mapped into GFP-T as specified in ITU-T G.7041 with the following parameters:

- Each GFP-T frame contains one superblock,
- the 65B\_PAD character is not used,
- GFP Idle frames are not used.

During a signal fail condition of the incoming 1000BASE-X client signal (e.g., in the case of a loss of input signal), either:

- this failed incoming 1000BASE-X signal is replaced by a stream of 10B blocks, with a bit rate of  $1\,250\,000$  kbit/s  $\pm 100$  ppm, each carrying a Link Fault indication as specified in IEEE Std. 802.3, which stream is then applied at the GFP-T mapper; or
- the GFP-T signal is replaced by a stream of GFP Client Signal Fail (CSF) and GFP-Idle frames as specified in ITU-T G.7041 with a bit rate of  $15/16 \times 1\,250\,000$  kbit/s  $\pm 100$  ppm.

During either:

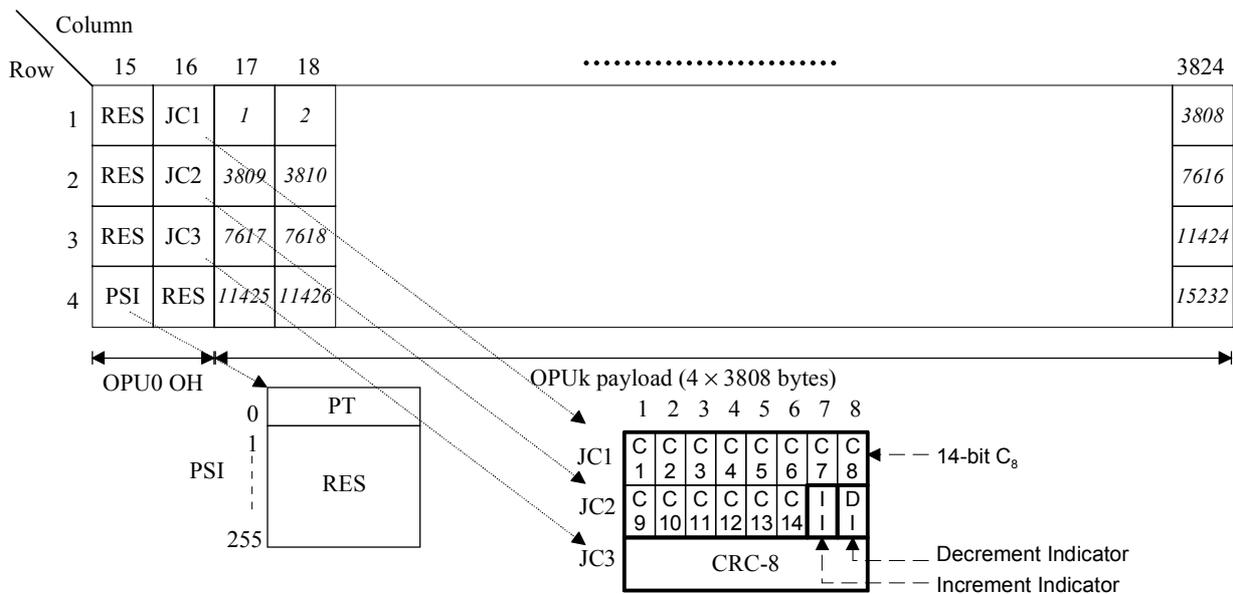
- a signal fail condition of the incoming ODU0/OPU0 signal (e.g., in the case of an ODU0-AIS, ODU0-LCK, ODU0-OCI condition), or
- incoming CSF frames as specified in ITU-T G.7041

the GFP-T demapper process generates a stream of 10B blocks, with each block carrying a Link Fault indication as specified in IEEE Std. 802.3 as a replacement signal for the lost 1000BASE-X signal.

NOTE 1 – The Ethernet link fault indication is a stream of repeating /C1/C2/C1/C2/ ... ordered sets, where C1 = /K28.5/D21.5/D0.0/ and C2 = /K28.5/D2.2/D0.0/. This character stream is then processed by the GFP-T mapper process in the same manner as if it were the received 8B/10B data stream, mapping it into GFP-T superblocks for transmission.

The OPU0 overhead for this mapping consists of a payload structure identifier (PSI) including the payload type (PT) and 255 bytes reserved for future international standardization (RES), three justification control (JC1, JC2, JC3) bytes and four bytes reserved for future international standardization (RES). The JC1, JC2 and JC3 bytes consist of a 14-bit  $C_8$  field (bits C1, C2, ..., C14), a 1-bit Increment Indicator (II) field, a 1-bit Decrement Indicator (DI) field and an 8-bit CRC-8 field which contains an error check code over the JC1, JC2 and JC3 fields.

The OPU0 payload for this mapping consists of  $4 \times 3808$  bytes.



**Figure 17-10 – OPU0 frame structure for the mapping of a 1000BASE-X signal**

$C_8$  is a binary count of the number of OPU0 payload bytes (8 bits) that carry a client (i.e., GFP-T) byte; it has values between 14405 and 14410. The  $C_i$  ( $i = 1..14$ ) bits that comprise  $C_8$  are used to indicate whether the  $C_8$  value is incremented or decremented from the value in the previous frame. Table 17-4 shows the inversion patterns for the  $C_i$  bits that are inverted to indicate an increment or decrement of the  $C_8$  value. A "1" entry in the table indicates an inversion of that bit.

- When the value of the  $C_8$  is incremented with +1 or +2, a subset of the  $C_i$  bits, as specified in Table 17-4, is inverted and the increment indicator (II) bit is set to 1.
- When the value of the  $C_8$  is decremented with –1 or –2, a subset of  $C_i$  bits, as specified in Table 17-4, is inverted and the decrement indicator (DI) bit is set to 1.
- When the value of  $C_8$  is changed with a value larger than +2 or –2, both the II and DI bits are set to 1 and the  $C_i$  bits contain the new  $C_8$  value. The CRC-8 verifies whether the  $C_8$  value has been received correctly, and provides optional single error correction.
- When the value of  $C_8$  is unchanged, both the II and DI bits are set to 0.

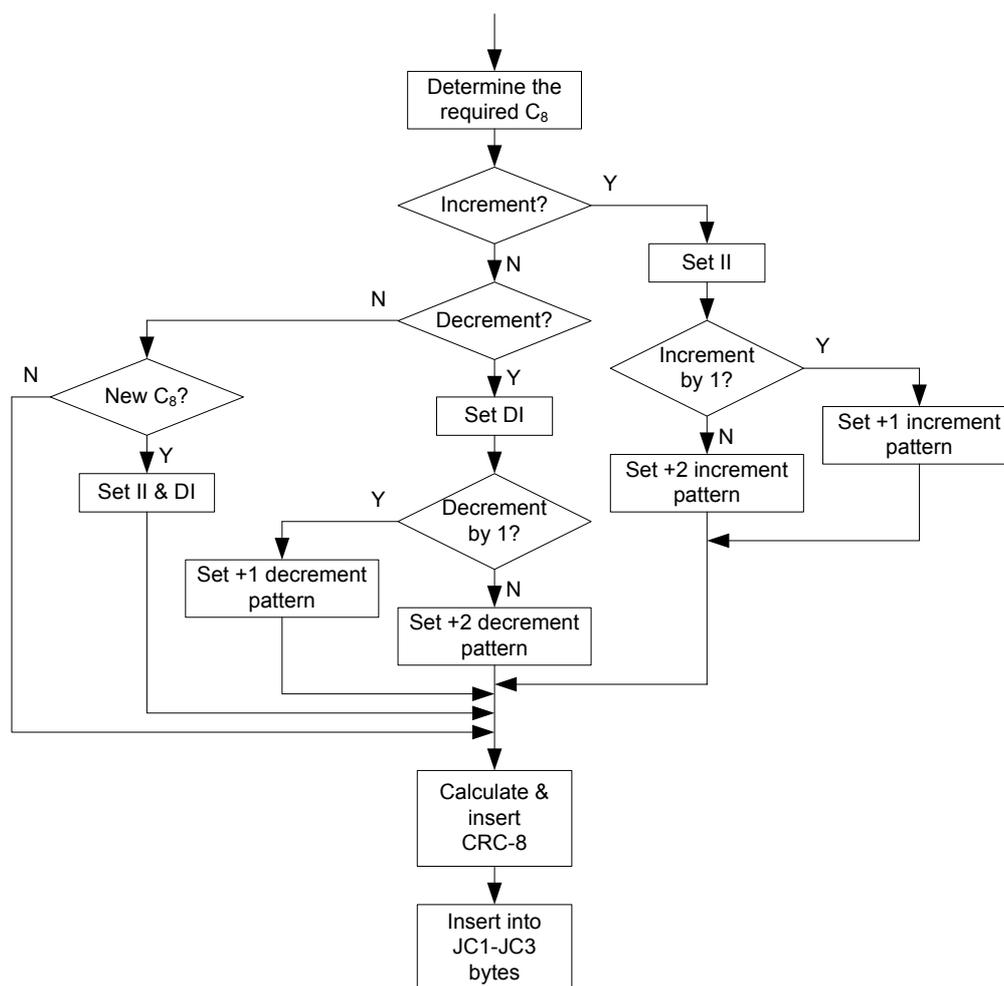
**Table 17-4 –  $C_8$  increment and decrement indicator patterns**

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	II	DI	Change
U	U	U	U	U	U	U	U	U	U	U	U	U	U	0	0	0
I	U	I	U	I	U	I	U	I	U	I	U	I	U	1	0	+1
U	I	U	I	U	I	U	I	U	I	U	I	U	I	0	1	–1
U	I	I	U	U	I	I	U	U	I	I	U	U	I	1	0	+2
I	U	U	I	I	U	U	I	I	U	U	I	I	U	0	1	–2
binary value														1	1	More than +2/–2
NOTE –																
– I indicates inverted $C_8$ bit																
– U indicates unchanged $C_8$ bit																

The CRC-8 is calculated over the JC1 and JC2 bits. The CRC-8 uses the  $g(x) = x^8 + x^3 + x^2 + 1$  generator polynomial, and is calculated as follows:

- 1) The JC1 and JC2 octets are taken in network octet order, most significant bit first, to form a 16-bit pattern representing the coefficients of a polynomial  $M(x)$  of degree 15.
- 2)  $M(x)$  is multiplied by  $x^8$  and divided (modulo 2) by  $G(x)$ , producing a remainder  $R(x)$  of degree 7 or less.
- 3) The coefficients of  $R(x)$  are considered to be an 8-bit sequence, where  $x^7$  is the most significant bit.
- 4) This 8-bit sequence is the CRC-8 where the first bit of the CRC-8 to be transmitted is the coefficient of  $x^7$  and the last bit transmitted is the coefficient of  $x^0$ .

The demapper process performs steps 1-3 in the same manner as the mapper process. In the absence of bit errors, the remainder shall be 0000 0000.



**Figure 17-11 – JC1, JC2 and JC3 generation**

A parallel logic implementation of the source CRC-8 is illustrated in Appendix IX.

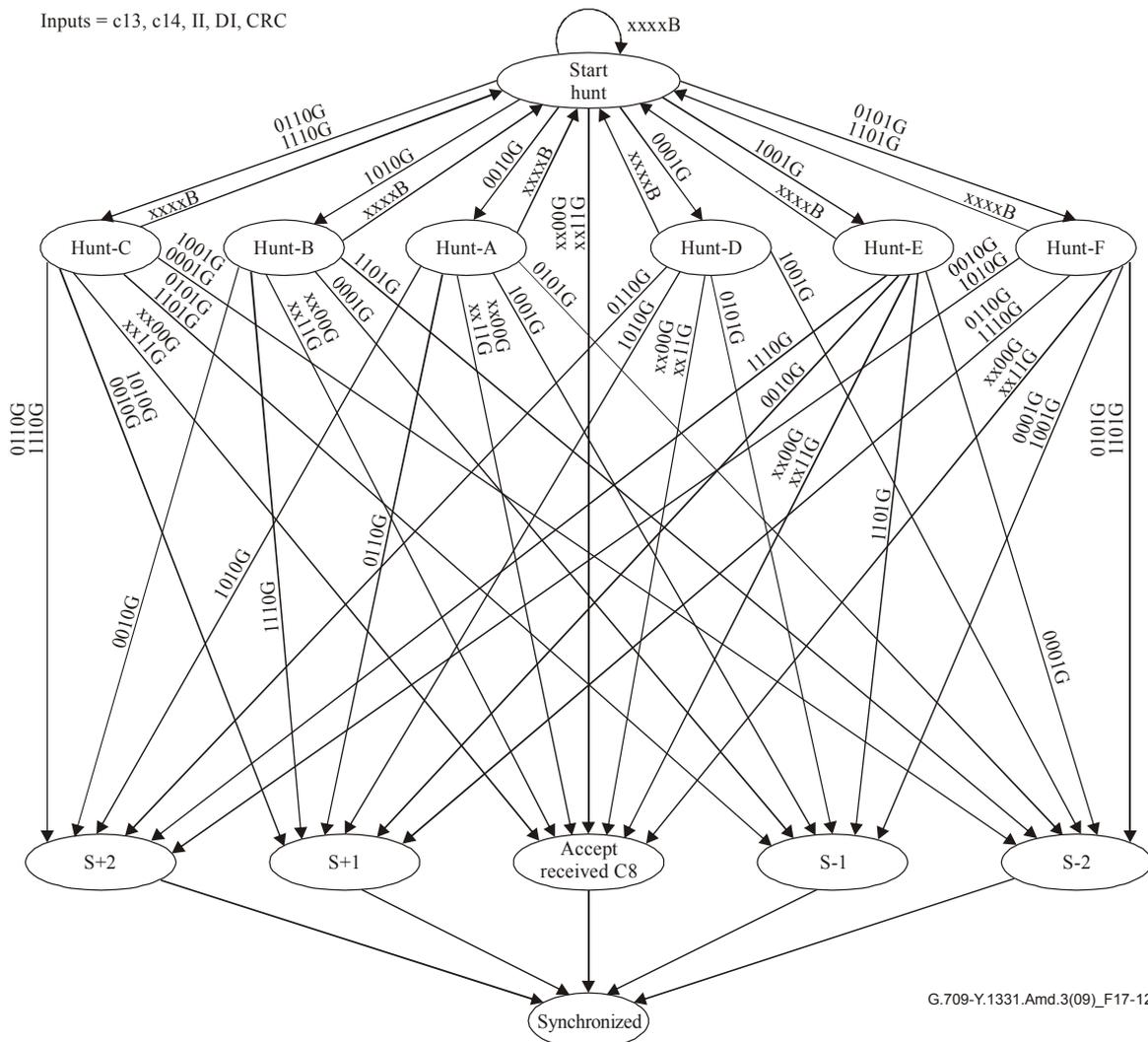
The justification control (JC1 to JC3) signal is used to control the distribution of the GFP-T octets within the OPU0 payload. The value of  $C_8$  in frame  $n$  indicates the number of OPU0 data bytes in frame  $n+1$ . If the bit-rate of the client is  $f_c$  and the OPU0 payload bit-rate is  $f_p$ , then the average value of  $C_8$  is  $f_c / f_p \times 15232$ , which is approximately 14407.311 for the 1000BASE-X into OPU0 mapping. Since this value is not an integer, an integral number of data bytes are inserted into each

OPU0 frame, and as the 1000BASE-X and OPU0 bit-rates may each vary within their timing tolerance,  $C_8$  may vary from one OPU0 frame to the next in the range 14405 to 14410. The remaining 822 to 827 bytes in the OPU0 payload will be used as justification octets, carrying an all-0's pattern.

The payload area of the OPU0 consists of columns 17-3824 of rows 1-4, for a total of 15232 bytes, which are numbered from 1 to 15232, as indicated in Figure 17-10. Each byte of the OPU0 payload contains either a data byte or a justification byte. A frame which contains  $C_8$  data bytes contains  $15232 - C_8$  justification bytes. Byte  $n$  of the OPU0 payload area is a data byte if  $n \times C_8 \bmod 15232 < C_8$ . Byte  $n$  of the OPUk payload area is a justification byte if  $n \times C_8 \bmod 15232 \geq C_8$ .

The ODU0 sink synchronizes its  $C_8$  value to the ODU0 source through the following process, which is illustrated in Figure 17-12. When the received JC octets contain  $II = DI$  and a valid CRC-8, the ODU0 sink accepts the received C1-C14 as its  $C_8$  value for the next frame. At this point, the ODU0 sink is synchronized to the ODU0 source. When  $II \neq DI$  with a valid CRC-8 in the current received frame (frame  $i$ ), the ODU0 sink must examine the received JC octets in the next frame (frame  $i + 1$ ) in order to obtain  $C_8$  synchronization.  $II \neq DI$  in frame  $i$  indicates that the source is performing a count increment or decrement operation that will modify the  $C_8$  value it sends in frame  $i + 1$ . Since this modification to the  $C_8$  will affect C13, C14, or both, the ODU0 sink uses C13, C14, II, and DI in frame  $i$  to determine its synchronization hunt state (Hunt – A-F in Figure 17-12) when it receives frame  $i + 1$ . If  $II = DI$  with a valid CRC-8 in frame  $i + 1$ ,  $C_8$  synchronization is achieved by directly accepting the received C1-C14 as the new  $C_8$ . If  $II \neq DI$  with a valid CRC-8 in frame  $i + 1$ , the sink uses the new C13, C14, II, and DI values to determine whether the source is communicating an increment or decrement operation and the magnitude of the increment/decrement step. This corresponds to the transition to the lower row of states in Figure 17-12. At this point, the ODU0 sink has identified the type of increment or decrement operation that is being signalled in frame  $i + 1$ . The sink then applies the appropriate bit inversion pattern from Table 17-4 to the received C1-C14 field to determine transmitted  $C_8$  value. Synchronization has now been achieved since the ODU0 sink has determined the current  $C_8$  and knows the expected  $C_8$  change in frame  $i + 2$ .

Inputs = c13, c14, II, DI, CRC



S+2 Count = C1–C14 after inverting C2, C3, C6, C7, C10, C11 and C14;  
Increment +2 for the next frame  
S+1 Count = C1–C14 after inverting C1, C3, C5, C7, C9, C11 and C13;  
Increment +1 for the next frame

S–1 Count = C1–C14 after inverting C2, C4, C6, C8, C10, C12 and C14;  
Decrement –1 for the next frame  
S–2 Count = C1–C14 after inverting C1, C4, C5, C8, C9, C12 and C13;  
Decrement –2 for the next frame

**Figure 17-12 – ODU0 sink count synchronization process diagram**

Note that the state machine of Figure 17-12 can also be used for off-line synchronization checking. When the ODU0 sink has synchronized its  $C_8$  value to the ODU0 source, it interprets the received JC octets according to the following principles.

- When the CRC-8 is good and  $II = DI$ , the ODU0 sink accepts the received  $C_8$  value.
- When the CRC-8 is good and  $II \neq DI$ , the ODU0 sink compares the received  $C_8$  value to its expected  $C_8$  value to determine the difference between these values. This difference is compared to the bit inversion patterns of Table 17-4 to determine the increment or decrement operation sent by the source and updates its  $C_8$  accordingly. Since the CRC-8 is good, the sink can use either JC1 or JC2 for this comparison.
- When the CRC-8 is bad, the ODU0 sink compares the received  $C_8$  value to its expected  $C_8$  value. The sink then compares the difference between these values, per Table 17-4, to the valid bit inversion patterns in JC1, and the bit inversion, II and DI pattern in JC2.
  - If JC1 contains a valid pattern and JC2 does not, the sink accepts the corresponding increment or decrement indication from JC1 and updates its  $C_8$  accordingly.

- If JC2 contains a valid pattern and JC1 does not, the sink accepts the corresponding increment or decrement indication from JC2 and updates its  $C_8$  accordingly.
- If both JC1 and JC2 contain valid patterns indicating the same increment or decrement operation, this indication is accepted and the sink updates its  $C_8$  accordingly.
- If neither JC1 nor JC2 contain valid patterns, the sink shall keep its current count value and begin the search for synchronization.

NOTE 2 – If JC1 and JC2 each contain valid patterns that are different from each other, the receiver can either keep the current  $C_8$  value and begin a synchronization search, or it can use the CRC-8 to determine whether JC1 or JC2 contains the correct pattern.

The ODU0 sink uses the updated  $C_8$  value to extract the client data from the next OPU0 frame.

### 17.7.2 Mapping a FC-1200 signal into OPU2e

The nominal line rate for FC-1200 is 10 518 750 kbit/s  $\pm$  100 ppm, and must therefore be compressed to a suitable rate to fit into an OPU2e.

The adaptation of the 64B/66B encoded FC-1200 client is done by transcoding a group of eight 66B blocks into one 513B block (as described in Annex B), assembling eight 513B blocks into one 516-octet superblock and encapsulating seventeen 516-octet superblocks into a 8800-octet GFP frame as illustrated in Figure 17-14.

NOTE 1 – The GFP encapsulation stage does not generate GFP-Idle frames and therefore the generated GFP stream is synchronous to the FC-1200 client stream. The adaptation process performs a 50/51 rate compression, so the resulting GFP stream has a signal bit rate of  $50/51 \times 10.51875$  Gbit/s  $\pm$  100 ppm (i.e. 10 312 500 kbit/s  $\pm$  100 ppm).

The stream of 8800-octet GFP frames is byte-synchronous mapped into the OPU2e payload by aligning the byte structure of every GFP frame with the byte structure of the OPU2e payload (see Figure 17-13). Sixty-four fixed stuff (FS) bytes are added in columns 1905 to 1920 of the OPU2e payload. All the GFP frames have the same length (8800 octets). The GFP frames are not aligned with the OPU2e payload structure and may cross the boundary between two OPU2e frames.

During a signal fail condition of the incoming FC-1200 signal (e.g., in the case of a loss of input signal), this failed incoming FC-1200 signal is replaced by a stream of 66B blocks, with each block carrying two Local Fault sequence ordered sets as specified in INCITS 364. This replacement signal is then applied at the transcoding process.

NOTE 2 – Local Fault sequence ordered set is /K28.4/D0.0/D0.0/D1.0/. The 66B block contains the following value SH = 10 0x55 00 00 01 00 00 00 01.

During signal fail condition of the incoming ODU2e/OPU2e signal (e.g., in the case of an ODU2e-AIS, ODU2e-LCK, ODU2e-OCI condition) a stream of 66B blocks, with each block carrying two Local Fault sequence ordered sets as specified in INCITS 364, is generated as a replacement signal for the lost FC-1200 signal.

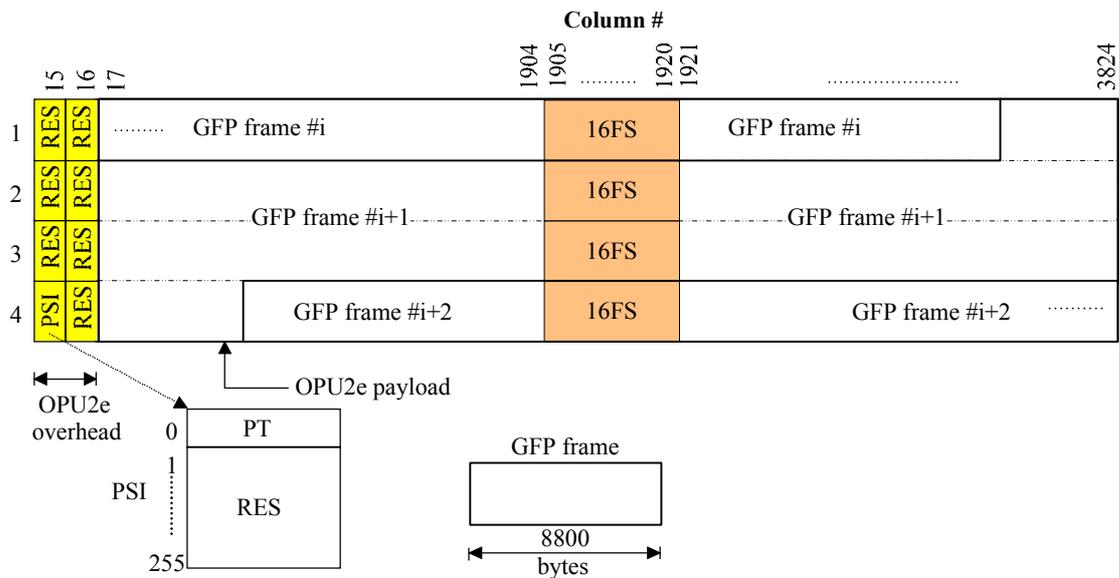


Figure 17-13 – Mapping of transcoded FC-1200 into OPU2e

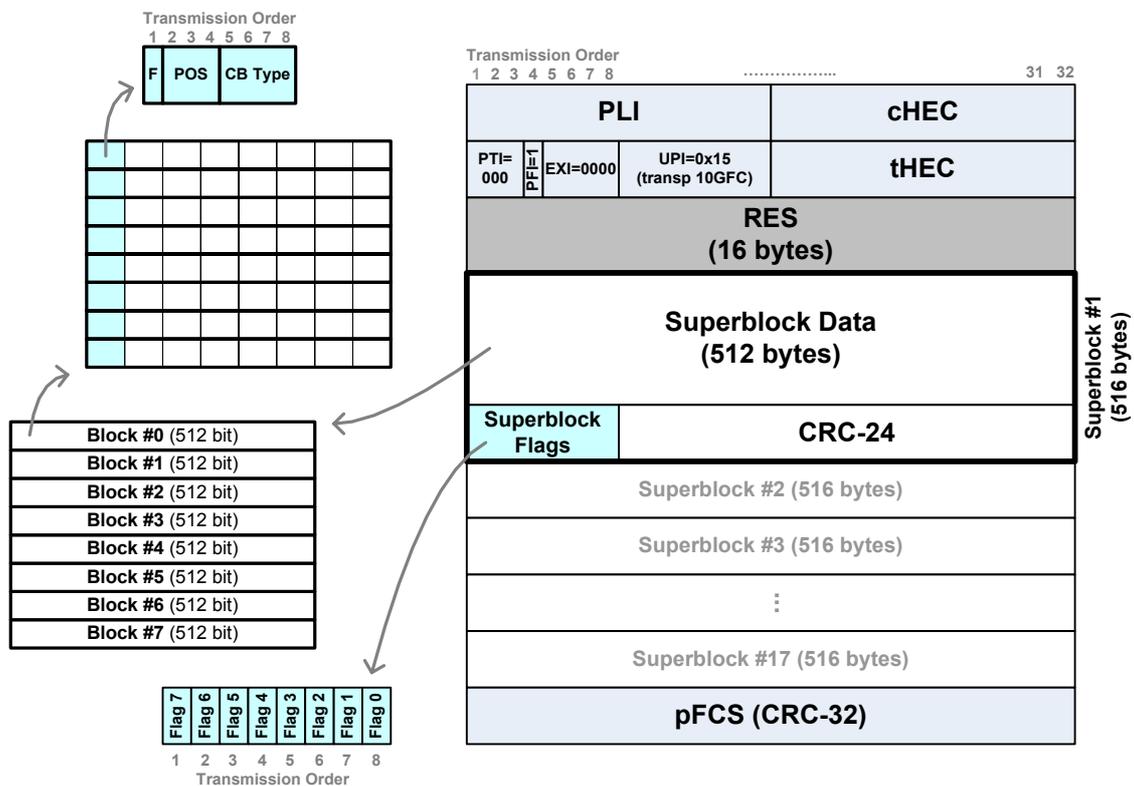


Figure 17-14 – GFP frame format for FC-1200

GFP framing is used to facilitate delineation of the superblock structure by the receiver. The leading flag bits from each of the eight 513B blocks are relocated into a single octet at the end of the 513-octet Superblock Data field (labelled "Superblock Flags").

To minimize the risk of incorrect decoding due to errors in the 1 to 65 octets of "control" information (Flags, POS, CB\_Type), a CRC-24 is calculated over the 65 octets within each superblock that may contain such "control" information and appended to form a 516-octet superblock. The 65 octets in the 516-octet superblock over which the CRC-24 is calculated are the

octets  $(1 + 8n)$  with  $n = 0..64$  (i.e., octets 1, 9, 17, .., 513). The generator polynomial for the CRC-24 is  $G(x) = x^{24} + x^{21} + x^{20} + x^{17} + x^{15} + x^{11} + x^9 + x^8 + x^6 + x^5 + x + 1$  with an all-ones initialization value, where  $x^{24}$  corresponds to the MSB and  $x^0$  to the LSB. This superblock CRC is generated by the source adaptation process using the following steps:

- 1) The 65 octets of "control" information (flags, POS, CB\_Type) are taken in network octet order (see Figure 17-14), most significant bit first, to form a 520-bit pattern representing the coefficients of a polynomial  $M(x)$  of degree 519.
- 2)  $M(x)$  is multiplied by  $x^{24}$  and divided (modulo 2) by  $G(x)$ , producing a remainder  $R(x)$  of degree 23 or less.
- 3) The coefficients of  $R(x)$  are considered to be a 24-bit sequence, where  $x^{23}$  is the most significant bit.
- 4) After inversion, this 24-bit sequence is the CRC-24.

Exactly 17 of these 516-octet superblocks are prefixed with the standard GFP core and type headers and 16 octets of "reserved" (padding). Because the number of 516-octet superblocks per GFP frame is known *a priori*, it is possible for this mapping scheme to operate in a cut-through (as opposed to store and forward) fashion, thus minimizing the mapping latency.

The payload FCS (a CRC-32) is appended to the end of each GFP frame and is calculated across the Payload Information field of the GFP frame per ITU-T G.7041. The purpose of the payload FCS is to provide visibility of bit errors occurring anywhere in the GFP Payload Information field, and thus augments the coverage provided by the per-Superblock CRC-24 (which only provides coverage for the "control" overhead in each superblock). The payload FCS is for purposes of statistics gathering only.

All octets in the GFP payload area are scrambled using the  $X^{43} + 1$  self synchronous scrambler, again per ITU-T G.7041.

## 2.18) Clause 19.1 OPUk Tributary Slot definition

*Modify the text as follows:*

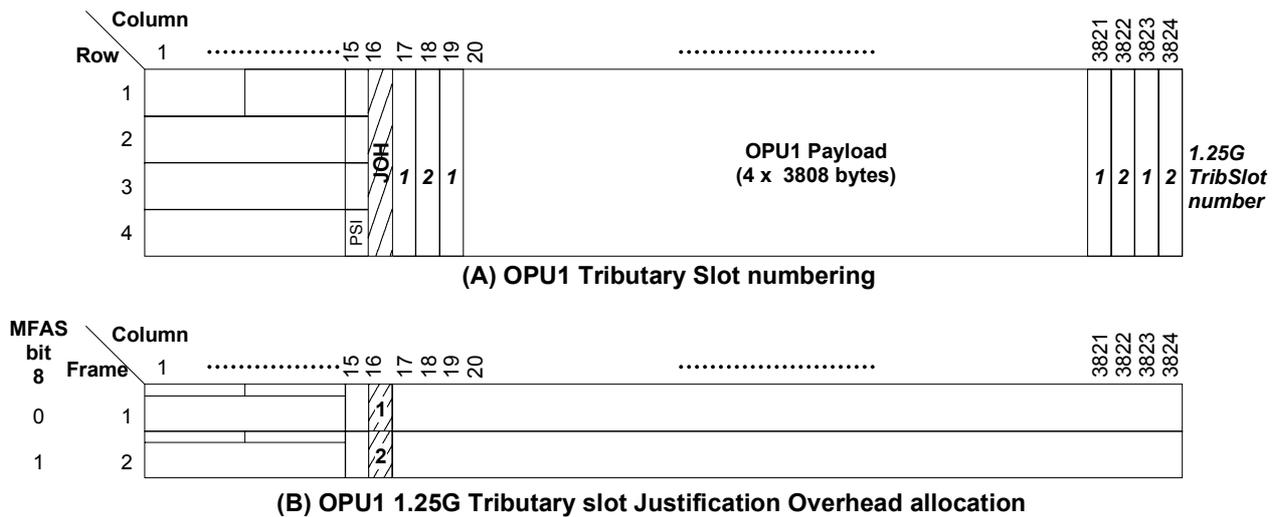
The OPUk is divided in a number of Tributary Slots (TS) and these Tributary Slots are interleaved within the OPUk. A Tributary Slot includes a part of the OPUk OH area and a part of the OPUk payload area. The bytes of the ODUj frame are mapped into the ODTj payload area and the ODTU bytes are mapped into the OPUk Tributary Slot or slots. The bytes of the ODTUjk Justification Overhead are mapped into the OPUk OH area.

## 2.19) Clause 19.1

*Add the following two subclauses:*

### 19.1.3 OPU1 Tributary Slot allocation

Figure 19-2 *bis* presents the OPU1 1.25G tributary slot allocation. An OPU1 1.25G Tributary Slot occupies 50% of the OPU1 payload area. It is a structure with 1904 columns by 4 rows (see Figure 19-2 *bis*). The two OPU1 1.25G TSs are byte interleaved in the OPU1 payload area.



**Figure 19-2 bis – OPU1 Tributary Slot allocation**

In addition, the justification overhead (JOH) consisting of justification control (JC) and negative justification opportunity (NJO) signals of the OPU1 TSs are located in the overhead area, column 16 of rows 1 to 4. The JOH is assigned to the related tributary slots on a per frame base.

JOH for a 1.25G Tributary Slot is available once every 2 frames. A 2-frame multiframe structure is used for this assignment. This multiframe structure is locked to bit 8 of the MFAS byte as shown in Table 19-2 bis.

**Table 19-2 bis – OPU1 Justification OH tributary slots**

MFAS bit 8	JOH 1.25G TS
0	1
1	2

**19.1.4 OPU4 Tributary Slot allocation**

OPU4 1.25 Gbit/s Tributary Slot definition and allocation are for further study.

**2.20) Clause 19.2**

*Add the following four subclauses:*

**19.2.4 ODTU01**

The optical channel data tributary unit 01 (ODTU01) is a structure with 1904 columns by 8 (2 × 4) rows plus 1 column of justification overhead (JOH). It carries a justified ODU0 signal. The ODTU01 structure is illustrated in Figure 19-14. The location of the JOH column depends on the OPU1 tributary slot used when multiplexing the ODTU01 in the OPU1 (see clause 19.1.3).

### **19.2.5 ODTU02**

For further study.

### **19.2.6 ODTU03**

For further study.

### **19.2.7 ODTU2e3**

The optical channel data tributary unit 2e3 (ODTU2e3) is a structure with 1190 columns by 64 ( $16 \times 4$ ) rows plus 5 times 7 bytes of tributary slot overhead (TSOH). It carries a justified ODU2e signal. The location of the TSOH depends on the OPU3 tributary slot used when multiplexing the ODTU2e3 in the OPU3 (see clause 19.1.2). They might not be equally distributed.

## **2.21) Clause 19.3 Multiplexing ODTUjk signals into the OPUk**

*Add the following text at the beginning of the clause:*

Multiplexing an ODTU01 signal into an OPU1 is realized by means of the mapping of the ODTU01 signal in one of the two OPU1 Tributary Slots.

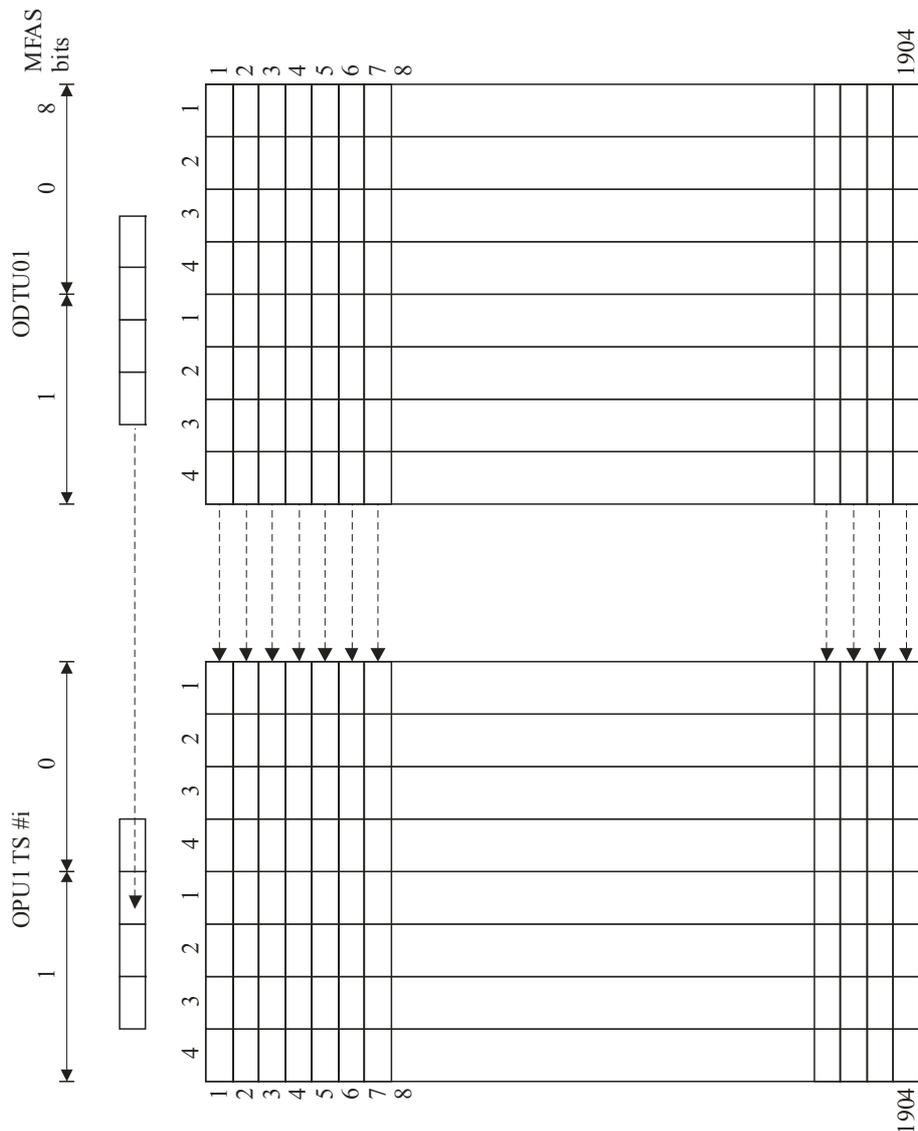
## **2.22) Clause 19.3**

*Add the following four subclauses:*

### **19.3.4 ODTU01 mapping into one OPU1 1.25G Tributary Slot**

A byte of the ODTU01 signal is mapped into a byte of an OPU1 1.25G TS #i ( $i = 1,2$ ), as indicated in Figure 19-5 *bis* for a group of 4 rows out of the ODTU01.

A byte of the ODTU01 JOH is mapped into a JOH byte within the OPU1 OH allocated to OPU1 1.25G TS #i.



G.709-Y.1331-Amd.3(09)\_F19-5bis

**Figure 19-5 bis – Mapping of ODTU01 (excluding JOH) into OPU1 1.25G TribSlot**

**19.3.5 ODTU02 mapping into one OPU2 1.25G Tributary Slot**

For further study.

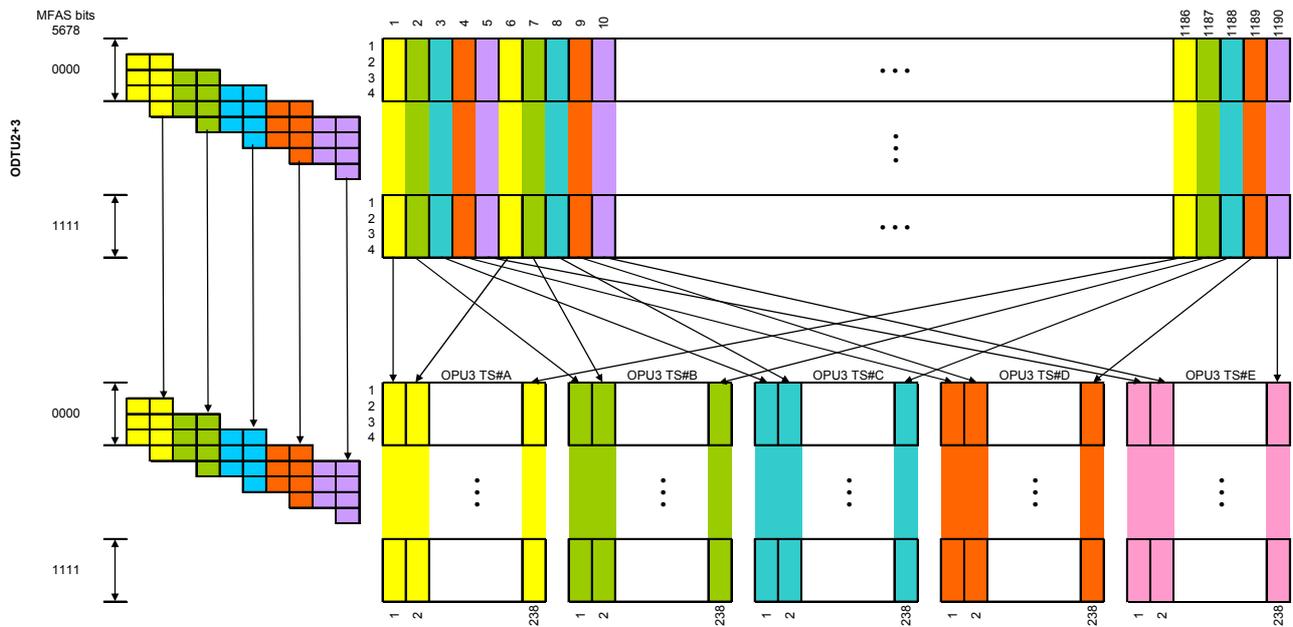
**19.3.6 ODTU03 mapping into one OPU3 1.25G Tributary Slot**

For further study.

**19.3.7 ODTU2e3 mapping into five OPU3 2.5G Tributary Slots**

A byte of the ODTU2e3 signal is mapped into a byte of one of five OPU3 TS #A,B,C,D,E (A,B,C,D,E = 1,2,...,16), as indicated in Figure 19-5 *ter* for a group of 4 rows out of the ODTU23.

A byte of the ODTU2e3 TSOH is mapped into a TSOH byte within the OPU3 OH allocated to OPU3 TS #A,B,C,D,E.



**Figure 19-5 ter – Mapping of ODTU2e3 into five tributary slots of OPU3**

**2.23) Clause 19.4 OPUk Multiplex Overhead**

*Modify the text as follows:*

The OPU<sub>k</sub> (k=1,2,3) multiplex overhead consists of Multiplex Structure Identifier (MSI), Justification Control (JC) and Negative Justification Opportunity (NJO) overhead. The OPU<sub>k</sub> MSI, JC and NJO overhead locations are shown in Figure 19-6A. In addition, two Positive Justification Overhead bytes (PJO1, PJO2) are located in the OPU<sub>k</sub> payload. Note that the PJO1 and PJO2 locations are multiframe, ODU<sub>j</sub> and OPU<sub>k</sub> tributary slot dependent.

The PJO1 for an ODU1 in OPU2 or OPU3 2.5G tributary slot #i (i: 1..4 or 1..16 respectively) is located in the first column of OPU<sub>k</sub> 2.5G tributary slot #i (OPU<sub>k</sub> column 16+i) and the PJO2 is located in the second column of OPU<sub>k</sub> 2.5G tributary slot #i (OPU2 column 20+i, OPU3 column 32+i) in frame #i of the four or sixteen frame multiframe. The four PJO1s for an ODU2 in OPU3 2.5G tributary slots #a, #b, #c and #d are located in the first column of OPU3 2.5G tributary slot #a (OPU3 column 16+a) in frames #a, #b, #c and #d of the sixteen frame multiframe. The four PJO2s for an ODU2 in OPU3 2.5G tributary slots #a, #b, #c and #d are located in the first column of OPU3 2.5G tributary slot #b (OPU3 column 16+b) in frames #a, #b, #c and #d of the sixteen frame multiframe. Figure 19-6A presents an example with four ODU2s in the OPU3 mapped into 2.5G tributary slots (1,5,9,10), (2,3,11,12), (4,14,15,16) and (6,7,8,13).

The PJO1 for an ODU0 in OPU1 1.25G tributary slot #i (i: 1,2) is located in the first column of OPU1 1.25G tributary slot #i (OPU1 column 16+i) and the PJO2 is located in the second column of OPU1 1.25G tributary slot #i (OPU1 column 18+i) in frame #i of the two frame multiframe.

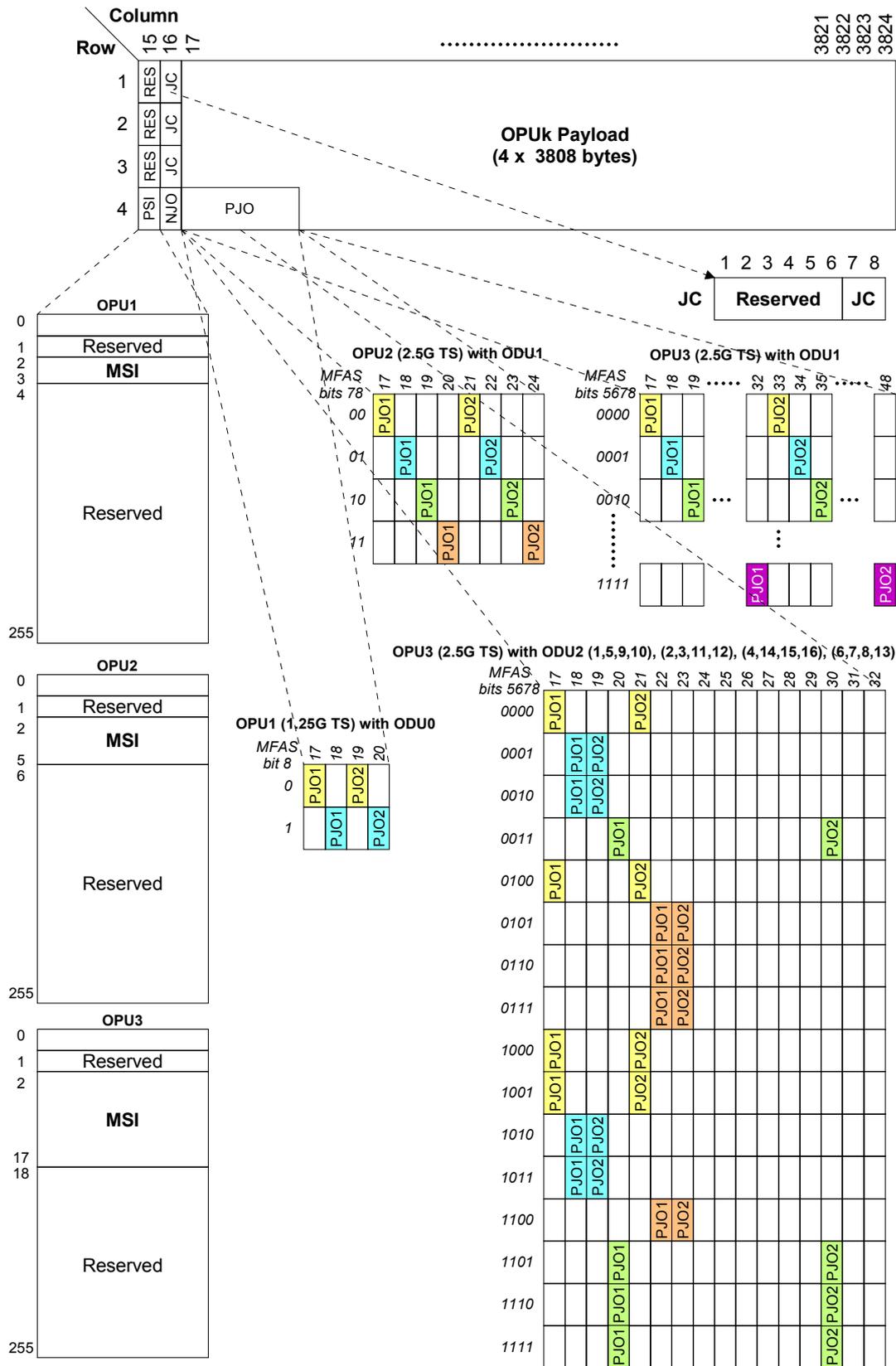
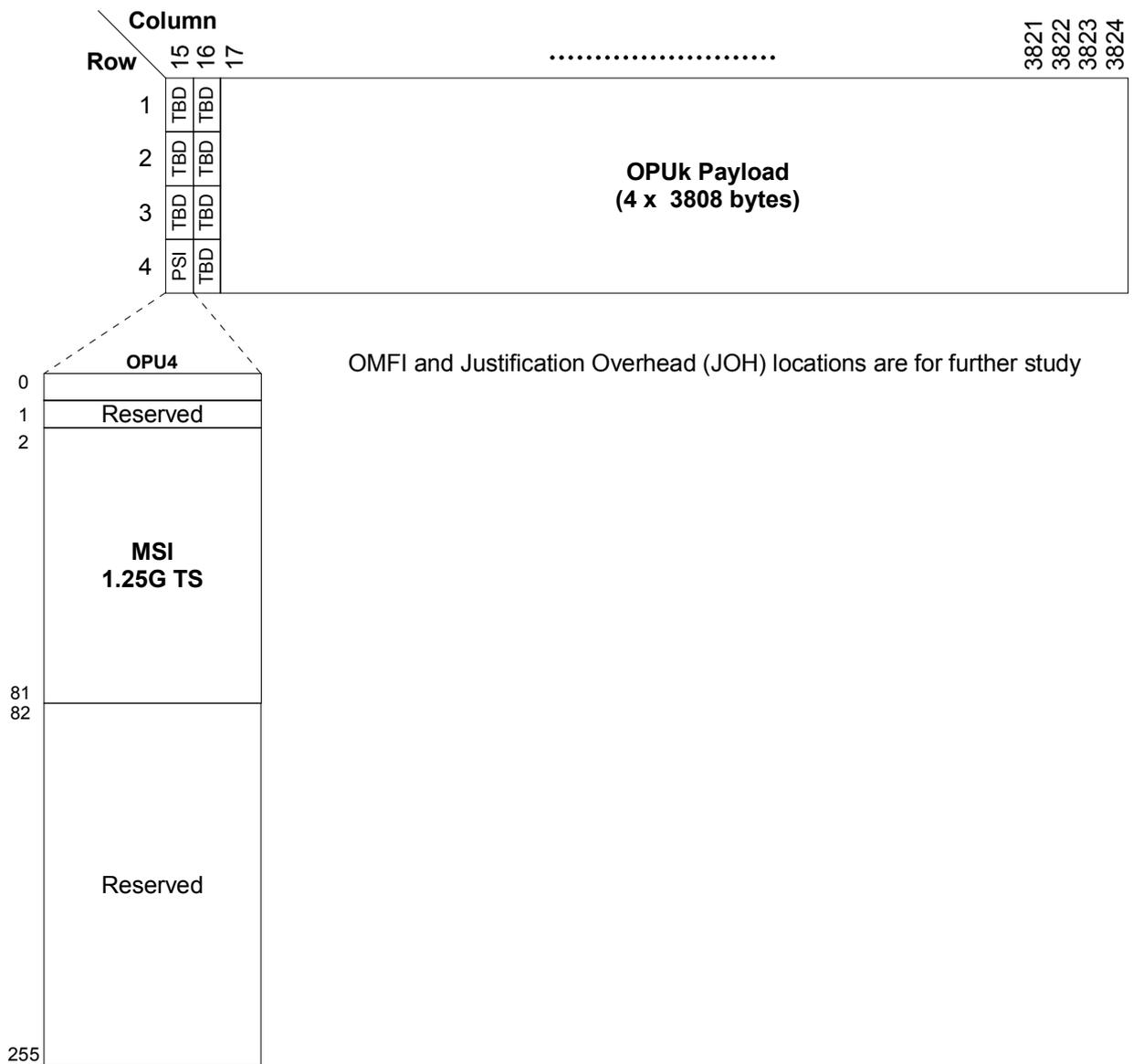


Figure 19-6A – OPUk (k = 1,2,3) multiplex overhead

The OPUk (k=4) multiplex overhead consists of multiplex structure identifier (MSI), OPU multiframe identifier (OMFI) and justification overhead. The OPUk MSI overhead locations are shown in Figure 19-6B. The OMFI and justification overhead locations are for further study.



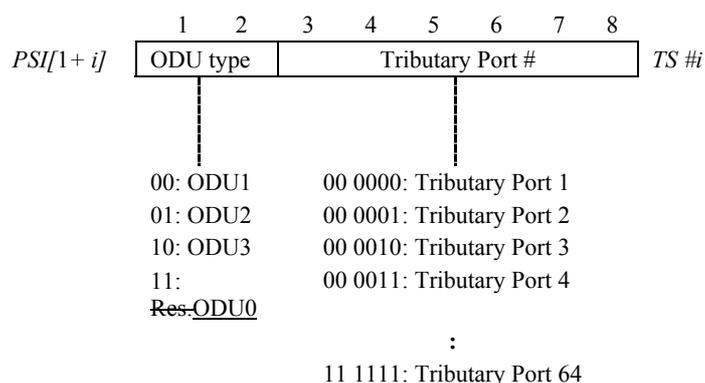
**Figure 19-6B – OPUk (k = 4) multiplex overhead**

**2.24) Clause 19.4.1 OPUk Multiplex Structure Identifier (MSI)**

*Modify the text as follows:*

The OPUk (k = 1,2,3) multiplex structure identifier (MSI) overhead, which encodes the ODU multiplex structure in the OPU, is located in the mapping specific area of the PSI signal (OPU1: PSI[2] .. PSI[3], OPU2: PSI[2] .. PSI[5], OPU3: PSI[2] .. PSI[17]). The MSI has an OPU specific length (OPU1: 2 bytes, OPU2: 4 bytes, OPU3: 16 bytes) and indicates the content of each tributary slot (TS) of an OPU. The generic coding for each TS is shown in Figure 19-7. One byte is used for each TS.

- Bits 1 and 2 indicate the ODU type transported in the TS.
- Bits 3 to 8 indicate the tributary port of the ODU transported. This is of interest in case of flexible assignment of ODUs to tributary slots (e.g., ODU2 into OPU3). In case of fixed assignment the tributary port number corresponds to the tributary slot number.



**Figure 19-7 – Generic OPU<sub>k</sub> (k = 1,2,3) MSI coding**

The OPU<sub>k</sub> (k = 4) multiplex structure identifier (MSI) overhead, which encodes the ODU multiplex structure in the OPU, is located in the mapping specific area of the PSI signal (OPU4: PSI[2] .. PSI[81]). The MSI has an OPU specific length (OPU4: 80 bytes) and indicates the content of each 1.25G tributary slot (TS) of an OPU. The coding for each TS is for further study. One byte is used for each TS.

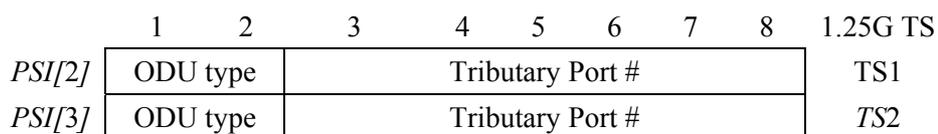
## 2.25) Clause 19.4.1

*Add the following two subclauses:*

### 19.4.1.3 OPU1 multiplex structure identifier (MSI)

For the 2 OPU1 tributary slots, 2 bytes of the PSI are used (PSI[2], PSI[3]), as shown in Figures 19-6A and 19-9 *bis*.

- The ODU type is fixed ODU0.
- The tributary port # indicates the port number of the ODU0 that is being transported in this TS; the assignment of ports to tributary slots is fixed, the port number equals the tributary slot number.



**Figure 19-9 *bis* – OPU1 MSI coding**

### 19.4.1.4 OPU4 multiplex structure identifier (MSI)

For further study.

## 2.26) Clause 19.4.2 OPU<sub>k</sub> Payload Structure Identifier Reserved overhead (RES)

*Modify the text as follows:*

253 (OPU1), 251 (OPU2), and 239 (OPU3) and 175 (OPU4) bytes are reserved in the OPU<sub>k</sub> PSI for future international standardization. These bytes are located in PSI[1] and PSI[3] (OPU1), PSI[6] (OPU2), PSI[18] (OPU3), PSI[82] (OPU4) to PSI[255] of the OPU<sub>k</sub> overhead. These bytes are set to all ZEROS.

**2.27) Clause 19.4.3 OPUk multiplex justification overhead (JOH)**

*Modify the text as follows:*

The justification overhead (JOH) located in column 16 of the OPUk ( $k = 1,2,3$ ) as indicated in Figure 19-6A consists of three justification control (JC) bytes and one negative justification opportunity (NJO) byte. The three JC bytes are located in rows 1, 2 and 3. The NJO byte is located in row 4.

Bits 7 and 8 of each JC byte are used for justification control. The other six bits are reserved for future international standardization.

The justification overhead (JOH) of the OPUk ( $k = 4$ ) is for further study.

**2.28) Clause 19.4**

*Add the following subclause:*

**19.4.4 OPU multiframe identifier overhead (OMFI)**

For further study.

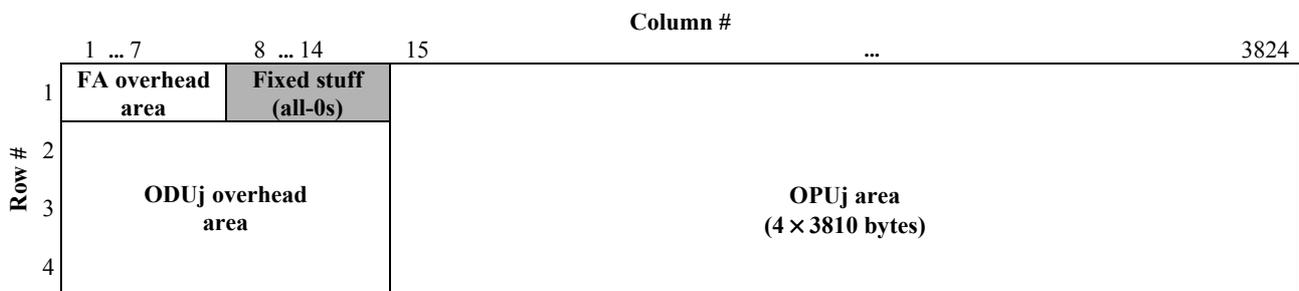
**2.29) Clause 19.5 Mapping ODUj into ODTUjk**

*Modify the text as follows:*

The mapping of ODUj signals (with up to  $\pm 20$  ppm bit-rate tolerance) into the ODTUjk signal ( $j = 0,1,2; k = 1,2,3$ ) is performed as an asynchronous mapping.

NOTE 1 – The maximum bit-rate tolerance between OPUk and the ODUj signal clock, which can be accommodated by this mapping scheme, is  $-130$  to  $+65$  ppm (ODU0 into OPU1),  $-113$  to  $+83$  ppm (ODU1 into OPU2),  $-96$  to  $+101$  ppm (ODU1 into OPU3) and  $-95$  to  $+101$  ppm (ODU2 into OPU3).

The ODUj signal is extended with Frame Alignment Overhead as specified in 15.6.2.1 and 15.6.2.2 and an all-0s pattern in the OTUj Overhead field (see Figure 19-10).



**Figure 19-10 – Extended ODUj frame structure (FA OH included, OTUj OH area contains Fixed Stuff)**

The OPUk signal for the multiplexed ODUj structure is created from a locally generated clock (within the limits specified in Table 7-3), which is independent of the ODUj ( $j = 0,1,2$ ) client signals.

The extended ODUj signal is adapted to the locally generated ODUk clock by means of an asynchronous mapping with  $-1/0/+1/+2$  positive/negative/zero (pnz) justification scheme.

An ODUj byte is mapped into an ODTUjk byte.

The asynchronous mapping process generates the JC, NJO, PJO1 and PJO2 according to Table 19-3. The demapping process interprets JC, NJO, PJO1 and PJO2 according to Table 19-3.

Majority vote (two out of three) shall be used to make the justification decision in the demapping process to protect against an error in one of the three JC signals.

**Table 19-3 – JC, NJO, PJO1 and PJO2 generation and interpretation**

JC 7 8	NJO	PJO1	PJO2	Interpretation
0 0	justification byte	data byte	data byte	no justification (0)
0 1	data byte	data byte	data byte	negative justification (-1)
1 0*	justification byte	justification byte	justification byte	double positive justification (+2)
1 1	justification byte	justification byte	data byte	positive justification (+1)
*) Note that this code is not used for the case of ODU0 into OPU1.				

The value contained in NJO, PJO1 and PJO2 when they are used as justification bytes is all-0s. The receiver is required to ignore the value contained in these bytes whenever they are used as justification bytes.

During a signal fail condition of the incoming ODU<sub>j</sub> client signal (e.g., OTU<sub>j</sub>-LOF), this failed incoming signal will contain the ODU<sub>j</sub>-AIS signal as specified in 16.5.1. This ODU<sub>j</sub>-AIS is then mapped into the ODTU<sub>jk</sub>.

For the case the ODU<sub>j</sub> is received from the output of a fabric (ODU<sub>j</sub> connection function), the incoming signal may contain (case of open matrix connection) the ODU<sub>j</sub>-OCI signal as specified in 16.5.2. This ODU<sub>j</sub>-OCI signal is then mapped into the ODTU<sub>jk</sub>.

NOTE 2 – Not all equipment will have a real connection function (i.e., switch fabric) implemented; instead, the presence/absence of tributary interface port units represents the presence/absence of a matrix connection. If such unit is intentionally absent (i.e., not installed), the associated ODTU<sub>jk</sub> signals should carry an ODU<sub>j</sub>-OCI signal. If such unit is installed but temporarily removed as part of a repair action, the associated ODTU<sub>jk</sub> signal should carry an ODU<sub>j</sub>-AIS signal.

~~The OPU<sub>k</sub> and therefore the ODTU<sub>jk</sub> (k = 2,3) signals are created from a locally generated clock (within the limits specified in Table 17-3), which is independent of the ODU<sub>j</sub> (j = 1,2) client signal.~~

~~The ODU<sub>j</sub> (j = 1,2) signal is mapped into the ODTU<sub>jk</sub> (k = 2,3) using a 1/0/+1/+2 positive/negative/zero (pnz) justification scheme.~~

The demapping of ODU<sub>j</sub> signals from the ODTU<sub>jk</sub> signal (j = 0,1,2; k = 1,2,3) is performed by extracting the extended ODU<sub>j</sub> signal from the OPU<sub>k</sub> under control of its justification overhead (JC, NJO, PJO1, PJO2).

NOTE 3 – For the case the ODU<sub>j</sub> signal is output as an OTU<sub>j</sub> signal, frame alignment of the extracted extended ODU<sub>j</sub> signal is to be recovered to allow frame synchronous mapping of the ODU<sub>j</sub> into the OTU<sub>j</sub> signal.

During signal fail condition of the incoming ODU<sub>k</sub>/OPU<sub>k</sub> signal (e.g., in the case of an ODU<sub>k</sub>-AIS, ODU<sub>k</sub>-LCK, ODU<sub>k</sub>-OCI condition) the ODU<sub>j</sub>-AIS pattern as specified in 16.5.1 is generated as a replacement signal for the lost ODU<sub>j</sub> signal.

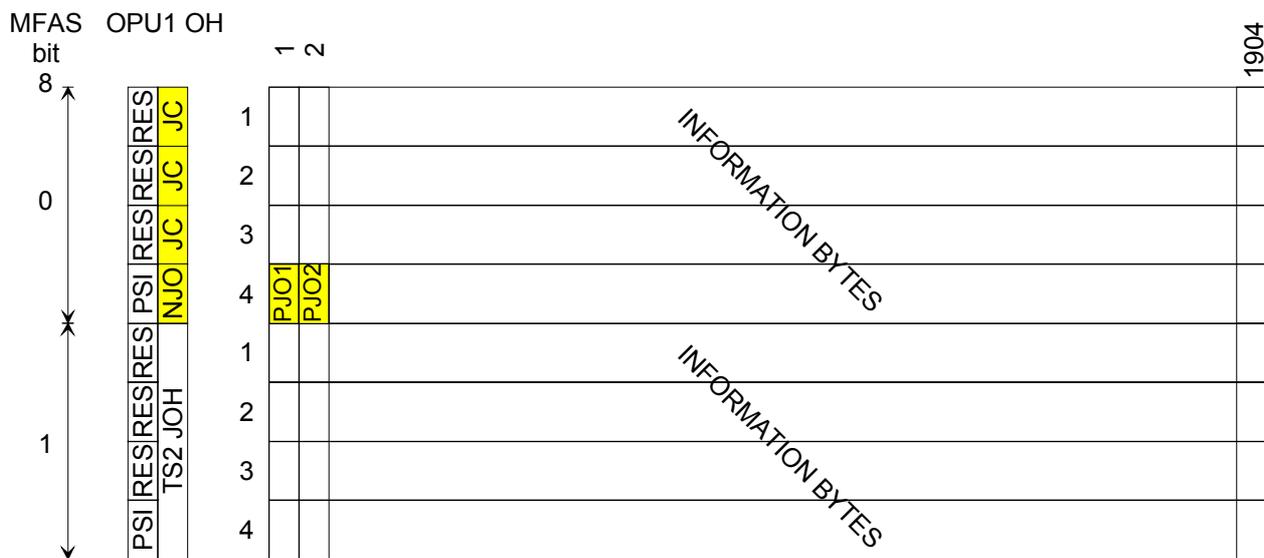
### 2.30) Clause 19.5

*Add the following subclause:*

#### 19.5.4 Mapping ODU0 into ODTU01

A byte of the ODU0 signal is mapped into an information byte of the ODTU01 (see Figure 19-2 *bis*). Once per 2 OPU1 frames, it is possible to perform either a positive or a negative justification action.

The frame in which justification can be performed is related to the JOH of the OPU1 TS in which the ODTU01 is mapped (Figure 19-14). Figure 19-5 *ter* shows the case with mapping in OPU1 TS1.  
 NOTE – The PJO2 field will always carry an information byte.



**Figure 19-14 – Mapping of ODU0 in OPU1 TS1**

## 2.31) New Annex B

Add new Annex B as follows:

# Annex B

## Adapting 64B/66B encoded clients via transcoding into 513B code blocks

(This annex forms an integral part of this Recommendation)

Clients using 64B/66B coding can be adapted in a codeword and timing transparent mapping via transcoding into 513B code blocks to reduce the bit-rate that is required to transport the signal. The resulting 513B blocks can be mapped in one of several ways depending on the requirements of the client and the available bandwidth of the container into which the client is mapped. This mapping can be applied to serial or parallel client interfaces.

### B.1 Transmission order

The order of transmission of information in all the diagrams in this annex is first from left to right and then from top to bottom.

### B.2 Client frame recovery

Client framing recovery consists of the recovering 64B/66B block lock per the state diagram in Figure 49-12 of IEEE 802.3 and the descrambling per the process shown in Figure 49-10 of IEEE 802.3.

Each 66B codeword (after block lock) is one of the following:

- A set of eight data bytes with a sync header of "01";
- a control block (possibly including seven or fewer data octets) beginning with a sync header of "10";

The 64 bits following the sync header are scrambled as a continuous bit-stream (skipping sync headers and PCS lane markers) according to the polynomial  $G(x) = 1 + x^{39} + x^{58}$ . The 64B/66B PCS receive process will descramble the bits other than (1) the sync header of 66B data and control blocks, and (2) the PCS lane markers.

Figure B.1 illustrates the ordering of 64B/66B code blocks after the completion of the recovering process for an interface.

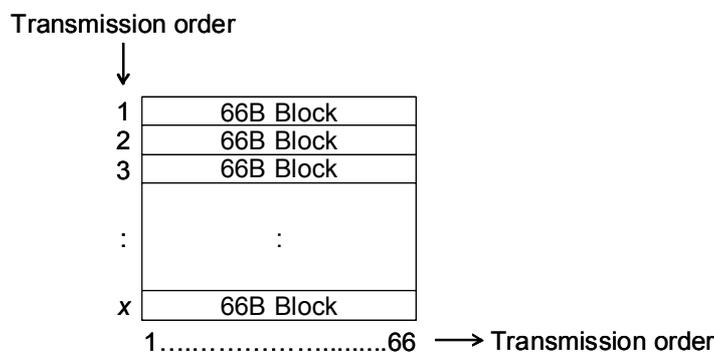


Figure B.1 – Stream of 64B/66B code blocks for transcoding

### B.3 Transcoding from 66B blocks to 513B blocks

The transcoding process at the encoder operates on an input sequence of 66B code blocks.

66B control blocks (after descrambling) follow the format shown in Figure B.2:

Input Data	S Y N C	Block Payload																																																																	
Data Block Format	Bit Position	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65
	D <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> D <sub>4</sub> D <sub>5</sub> D <sub>6</sub> D <sub>7</sub>	0	1	D <sub>0</sub>				D <sub>1</sub>				D <sub>2</sub>				D <sub>3</sub>				D <sub>4</sub>				D <sub>5</sub>				D <sub>6</sub>				D <sub>7</sub>																																			
Control block formats		Block type field																																																																4-bit code	
C <sub>0</sub> C <sub>1</sub> C <sub>2</sub> C <sub>3</sub> C <sub>4</sub> C <sub>5</sub> C <sub>6</sub> C <sub>7</sub>	1	0	0x1e				C <sub>0</sub>				C <sub>1</sub>				C <sub>2</sub>				C <sub>3</sub>				C <sub>4</sub>				C <sub>5</sub>				C <sub>6</sub>				C <sub>7</sub>				0001																												
C <sub>0</sub> C <sub>1</sub> C <sub>2</sub> C <sub>3</sub> O <sub>4</sub> D <sub>5</sub> D <sub>6</sub> D <sub>7</sub>	1	0	0x2d				C <sub>0</sub>				C <sub>1</sub>				C <sub>2</sub>				C <sub>3</sub>				O <sub>4</sub>				D <sub>5</sub>				D <sub>6</sub>				D <sub>7</sub>				0010																												
C <sub>0</sub> C <sub>1</sub> C <sub>2</sub> C <sub>3</sub> S <sub>4</sub> D <sub>5</sub> D <sub>6</sub> D <sub>7</sub>	1	0	0x33				C <sub>0</sub>				C <sub>1</sub>				C <sub>2</sub>				C <sub>3</sub>								D <sub>5</sub>				D <sub>6</sub>				D <sub>7</sub>				0111																												
O <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> S <sub>4</sub> D <sub>5</sub> D <sub>6</sub> D <sub>7</sub>	1	0	0x66				D <sub>1</sub>				D <sub>2</sub>				D <sub>3</sub>				O <sub>0</sub>								D <sub>5</sub>				D <sub>6</sub>				D <sub>7</sub>				1011																												
O <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> O <sub>4</sub> D <sub>5</sub> D <sub>6</sub> D <sub>7</sub>	1	0	0x55				D <sub>1</sub>				D <sub>2</sub>				D <sub>3</sub>				O <sub>0</sub>				O <sub>4</sub>				D <sub>5</sub>				D <sub>6</sub>				D <sub>7</sub>				1101																												
S <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> D <sub>4</sub> D <sub>5</sub> D <sub>6</sub> D <sub>7</sub>	1	0	0x78				D <sub>1</sub>				D <sub>2</sub>				D <sub>3</sub>				D <sub>4</sub>				D <sub>5</sub>				D <sub>6</sub>				D <sub>7</sub>				1110																																
O <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> C <sub>4</sub> C <sub>5</sub> C <sub>6</sub> C <sub>7</sub>	1	0	0x4b				D <sub>1</sub>				D <sub>2</sub>				D <sub>3</sub>				O <sub>0</sub>				C <sub>4</sub>				C <sub>5</sub>				C <sub>6</sub>				C <sub>7</sub>				1000																												
T <sub>0</sub> C <sub>1</sub> C <sub>2</sub> C <sub>3</sub> C <sub>4</sub> C <sub>5</sub> C <sub>6</sub> C <sub>7</sub>	1	0	0x87								C <sub>1</sub>				C <sub>2</sub>				C <sub>3</sub>				C <sub>4</sub>				C <sub>5</sub>				C <sub>6</sub>				C <sub>7</sub>				0011																												
D <sub>0</sub> T <sub>1</sub> C <sub>2</sub> C <sub>3</sub> C <sub>4</sub> C <sub>5</sub> C <sub>6</sub> C <sub>7</sub>	1	0	0x99				D <sub>0</sub>								C <sub>2</sub>				C <sub>3</sub>				C <sub>4</sub>				C <sub>5</sub>				C <sub>6</sub>				C <sub>7</sub>				0101																												
D <sub>0</sub> D <sub>1</sub> T <sub>2</sub> C <sub>3</sub> C <sub>4</sub> C <sub>5</sub> C <sub>6</sub> C <sub>7</sub>	1	0	0xaa				D <sub>0</sub>				D <sub>1</sub>								C <sub>3</sub>				C <sub>4</sub>				C <sub>5</sub>				C <sub>6</sub>				C <sub>7</sub>				1001																												
D <sub>0</sub> D <sub>1</sub> D <sub>2</sub> T <sub>3</sub> C <sub>4</sub> C <sub>5</sub> C <sub>6</sub> C <sub>7</sub>	1	0	0xb4				D <sub>0</sub>				D <sub>1</sub>				D <sub>2</sub>								C <sub>4</sub>				C <sub>5</sub>				C <sub>6</sub>				C <sub>7</sub>				1010																												
D <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> T <sub>4</sub> C <sub>5</sub> C <sub>6</sub> C <sub>7</sub>	1	0	0xcc				D <sub>0</sub>				D <sub>1</sub>				D <sub>2</sub>				D <sub>3</sub>								C <sub>5</sub>				C <sub>6</sub>				C <sub>7</sub>				1100																												
D <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> D <sub>4</sub> T <sub>5</sub> C <sub>6</sub> C <sub>7</sub>	1	0	0xd2				D <sub>0</sub>				D <sub>1</sub>				D <sub>2</sub>				D <sub>3</sub>				D <sub>4</sub>								C <sub>6</sub>				C <sub>7</sub>				0110																												
D <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> D <sub>4</sub> D <sub>5</sub> T <sub>6</sub> C <sub>7</sub>	1	0	0xe1				D <sub>0</sub>				D <sub>1</sub>				D <sub>2</sub>				D <sub>3</sub>				D <sub>4</sub>				D <sub>5</sub>								C <sub>7</sub>				0000																												
D <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> D <sub>4</sub> D <sub>5</sub> D <sub>6</sub> T <sub>7</sub>	1	0	0xff				D <sub>0</sub>				D <sub>1</sub>				D <sub>2</sub>				D <sub>3</sub>				D <sub>4</sub>				D <sub>5</sub>				D <sub>6</sub>								1111																												

Figure B.2 – 66B block coding

A group of eight 66B blocks is encoded into a single 513B block. The format is illustrated in Figure B.3:

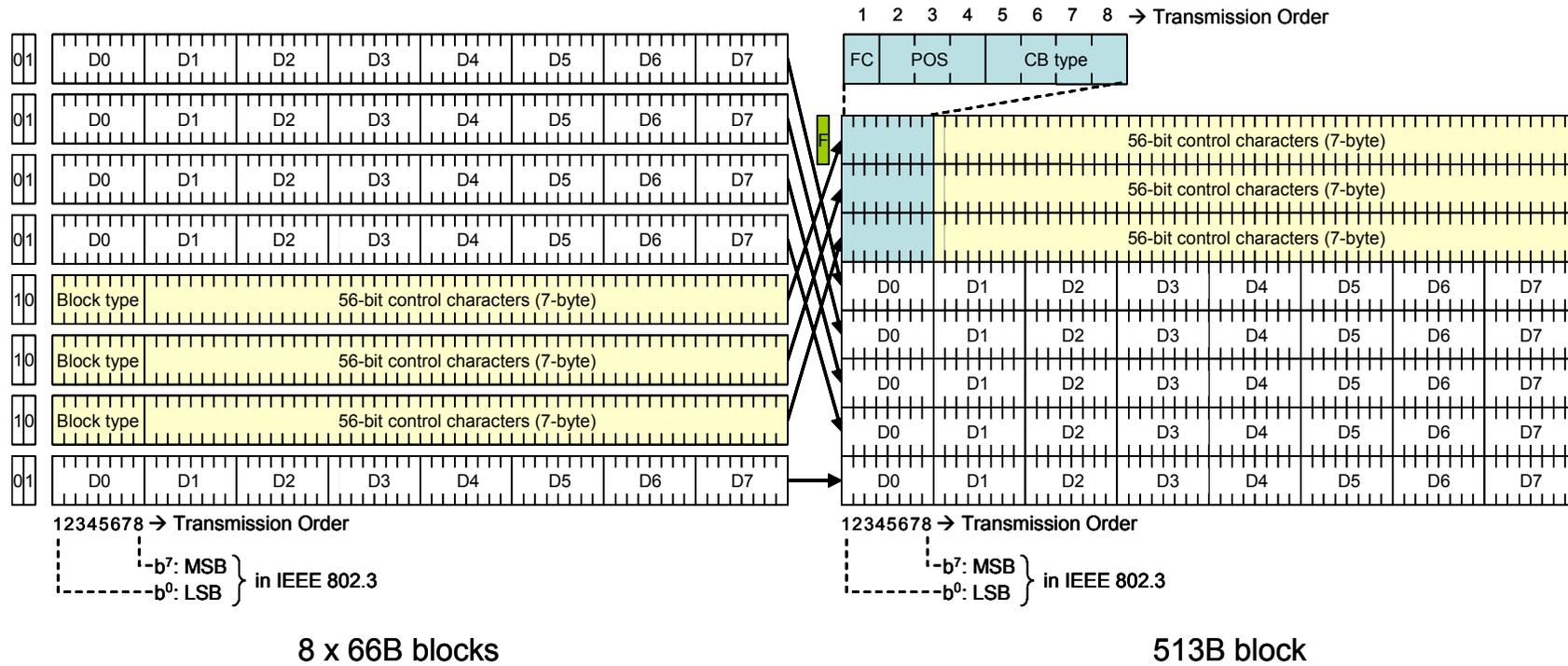
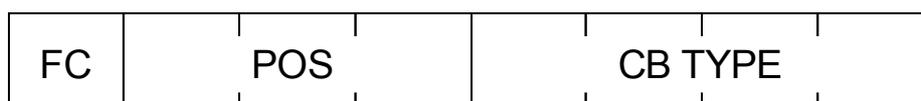


Figure B.3 – 513B block code format

Each of the 66B blocks is encoded into a row of the 8-byte by 8-row structure. Any 66B control blocks are placed into the uppermost rows of the structure in the order received, while any all-data 66B blocks are placed into the lowermost rows of the structure in the order received.

The flag bit "F" is 1 if the 513B structure contains at least one 66B control block, and 0 if the 513B structure contains eight all-data 66B blocks. Because the 66B control blocks are placed into the uppermost rows of the block, if the flag bit "F" is 1, then the first row will contain a mapping of a 66B control block.

A 66B control block is encoded into a row of the structure shown in Figure B.3 as follows: The sync header of "10" is removed. The byte representing the Block Type field (see Figure B.2) is replaced by the structure shown in Figure B.4:



**Figure B.4 – Mapping of control block type or PCS lane alignment marker indication**

The byte indicating the control block type (one of 15 legal values) is translated into a 4-bit code according to the rightmost column of Figure B.2. The 3-bit POS field is used to encode the position in which this control block was received in the sequence of eight 66B blocks. The flag continuation bit "FC" will be set to a 0 if this is the final 66B control block or PCS lane alignment marker encoded in this 513B block, or to a 1 if one or more 66B control blocks or PCS lane alignment markers follow this one. At the decoder, the Flag bit for the 513B block as a whole, plus the flag continuation bits in each row containing the mapping of a 66B control block or PCS lane alignment marker will allow identification of those rows, which can then be restored to their original position amongst any all-data 66B blocks at the egress according to the POS field. The remaining 7 bytes of the row are filled with the last 7 bytes of the 66B control block.

An all-data 66B block is encoded into a row of the 513B block by dropping the sync header and copying the remaining eight bytes into the row. If all eight rows of the 513B block are placements of 66B all-data blocks, the flag bit "F" will be 0. If fewer than eight rows of the 513B block are placements of 66B all-data blocks, they will appear at the end, and the row containing the placement of the final 66B control block will have a flag continuation bit "FC" value of 0.

The decoder operates in the reverse of the encoder to reconstruct the original sequence of 66B blocks. If the Flag bit "F" is 1, then 66B control blocks starting from the first row of the block are reconstructed and placed in the position indicated by the POS field. This process continues through all of the control blocks working downward from the top row. The final 66B control block placed within the 513B block will be identified when the flag continuation bit "FC" is zero.

The structure of the 512B/513B code block is shown in Figure B.5. For example, if there is a single 64B/66B control block CB1 in a 512B/513B code block and it was originally located between 64B/66B data blocks DB2 and DB3, the first octet of the 64B character will contain 0.010.1101.CB1; the leading bit in the control octet of 0 indicates the Flag Continuation "FC" that this 64B control block is the last one in the 512B/513B code block, the value of 010 indicates CB1's Position "POS" between DB2 and DB3, and the value of 1101 is a four-bit representation of the control code's block type "CB TYPE" (of which the eight-bit original block type is 0x55).

Input client characters	Flag bit	512-bit (64-Octet) field							
All data block	0	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
7 data block, 1 control block	1	0 AAA aaaa CB1	DB1	DB2	DB3	DB4	DB5	DB6	DB7
6 data block, 2 control block	1	1 AAA aaaa CB1	0 BBB bbbb CB2	DB1	DB2	DB3	DB4	DB5	DB6
5 data block, 3 control block	1	1 AAA aaaa CB1	1 BBB bbbb CB2	0 CCC cccc CB3	DB1	DB2	DB3	DB4	DB5
4 data block, 4 control block	1	1 AAA aaaa CB1	1 BBB bbbb CB2	1 CCC cccc CB3	0 DDD dddd CB4	DB1	DB2	DB3	DB4
3 data block, 5 control block	1	1 AAA aaaa CB1	1 BBB bbbb CB2	1 CCC cccc CB3	1 DDD dddd CB4	0 EEE eeee CB5	DB1	DB2	DB3
2 data block, 6 control block	1	1 AAA aaaa CB1	1 BBB bbbb CB2	1 CCC cccc CB3	1 DDD dddd CB4	1 EEE eeee CB5	0 FFF ffff CB6	DB1	DB2
1 data block, 7 control block	1	1 AAA aaaa CB1	1 BBB bbbb CB2	1 CCC cccc CB3	1 DDD dddd CB4	1 EEE eeee CB5	1 FFF ffff CB6	0 GGG gggg CB7	DB1
8 control block	1	1 AAA aaaa CB1	1 BBB bbbb CB2	1 CCC cccc CB3	1 DDD dddd CB4	1 EEE eeee CB5	1 FFF ffff CB6	1 GGG gggg CB7	0 HHH hhhh CB8

-Leading bit in a 66B control block FC = 1 if there are more 66B control block and = 0 if this payload contains the last control block in that 513B block  
-AAA = 3-bit representation of the 1 st control code's original position (1 st control code locator: POS)  
-BBB = 3-bit representation of the 2 nd control code's original position (2 nd control code locator: POS)  
....  
-HHH = 3-bit representation of the 8 th control code's original position (8 th control code locator: POS)  
-aaaa = 4-bit representation of the 1 st control code's type (1 st control block type: CB TYPE)  
-bbbb = 4-bit representation of the 2 nd control code's type (2 nd control block type: CB TYPE)  
....  
-hhhh = 4-bit representation of the 8 th control code's type (8 th control block type: CB TYPE)  
-CBi = 56-bit representation of the i th control code characters  
-DBi = 64-bit representation of the i th data value in order of transmission

**Figure B.5 – 513B code block components**

### B.3.1 Errors detected before 512B/513B encoder

A set of errors might be detected at the 64B/66B PCS receive process which, in addition to appropriate alarming, need to send the appropriate signal downstream.

Errors encountered before the encoder, such as loss of client signal, will result in the insertion of an Ethernet LF sequence ordered set prior to this process, which will be transcoded as any other control block. The same action should be taken as a result of failure to achieve 66B block lock on an input signal.

An invalid 66B block will be converted to an error control block before transcoding. An invalid 66B block is one which does not have a sync header of "01" or "10", or one which has a sync header of "10" and a control block type field which does not appear in Figure B.2. An error control block has sync bits of "10", a block type code of 0x1e, and 8 seven-bit /E/ error control characters. This will prevent the Ethernet receiver from interpreting a sequence of bits containing this error as a valid packet.

### B.3.2 Errors detected by 512B/513B decoder

Several mechanisms will be employed to reduce the probability that the decoder constructs erroneous 64B/66B encoded data at the egress if bit errors have corrupted. Since detectable corruption normally means that the proper order of 66B blocks to construct at the decoder cannot be reliably determined, if any of these checks fail, the decoder will transmit eight 66B error control blocks (sync="10", control block type=0x1e, and eight 7-bit /E/ control characters).

Mechanisms for improving the robustness and for 513B block lock are discussed in Appendix VII.

#### **B.4 Link Fault Signalling**

In-band link fault signalling in the client 64B/66B code (e.g., if a Local Fault or Remote Fault sequence ordered set is being transmitted between Ethernet equipments) is carried transparently according to this transcoding.

## 2.32) New Annex C

Add new Annex C as follows:

### Annex C

#### Adaptation of OTU3 and OTU4 over multichannel parallel interfaces

(This annex forms an integral part of this Recommendation)

NOTE 1 – This mechanism is designed to allow the use of the optical modules being developed for IEEE 40GBASE-R and 100GBASE-R signals for short-reach client-side OTU3 and OTU4 interfaces, respectively. The corresponding physical layer specifications are being added to ITU-T G.695 and ITU-T G.959.1.

OTU3 signals may be carried over parallel interfaces consisting of four lanes.

OTU4 signals may be carried over parallel interfaces consisting of four or ten lanes, which are formed by bit multiplexing of 20 logical lanes.

NOTE 2 – Ten lane IEEE 100GBASE-R interfaces have no corresponding ITU-T physical layer interface specification.

The OTU3 and OTU4 frames are inversely multiplexed over physical/logical lanes on a 16-byte boundary aligned with the OTUk frame as illustrated in Figure C.1.

1	1:16(FAS)	17:32	33:48	49:64	...	4065:4080
2	4081:4096	4097:5012	5013:5028	5029:5044	...	9145:9160
3	9161:9176	9177:9192	9193:9208	9209:9224	...	12225:12240
4	12241:12256	12257:12272	12273:12288	12289:12304	...	16305:16320

**Figure C.1 – OTU3 and OTU4 frames divided on 16-byte boundary**

Each 16-byte increment of an OTU3 frame is distributed, round robin, to each of the four physical lanes. Each 16-byte increment of an OTU4 frame is distributed, round robin, to each of the 20 logical lanes. On each OTUk frame boundary, the lane assignments are rotated.

For OTU3, the lane rotation and assignment is determined by the two LSBs of the MFAS as described in Table C.1, which indicates the starting group of bytes of the OTU3 frame that are sent on each lane.

NOTE 3 – MFAS is scrambled as defined in clause 11.2.

The pattern repeats over each 64 bytes until the end of the OTU3 frame. The following OTU3 frame will use different lane assignments, according to the MFAS.

**Table C.1 – Lane rotation assignments for OTU3**

MFAS 7-8	Lane 0	Lane 1	Lane 2	Lane 3
*00	1:16	17:32	33:48	49:64
*01	49:64	1:16	17:32	33:48
*10	33:48	49:64	1:16	17:32
*11	17:32	33:48	49:64	1:16

The distribution of 16-byte blocks from the sequence of OTU3 frames is illustrated in Figure C.2.

The parallel lanes can be reassembled at the sink by first recovering framing on each of the parallel lanes. The lane rotation mechanism will place the first 16 bytes of the OTU3 frame on each lane once per  $4080 \times 4$  bytes (the same as an OTUk itself). The two LSBs of the MFAS will be the same in each FAS on a particular lane, which allows the lane to be identified. Since the MFAS cycles through 256 distinct values, the lanes can be deskewed and reassembled by the receiver as long as the total skew does not exceed 127 OTU3 frame periods (approximately  $385\mu\text{s}$ ). The receiver must use the MFAS to identify each received lane, as lane positions may not be preserved by the optical modules to be used for this application.

For distribution of OTU4 to twenty logical lanes, since the MFAS is not a multiple of 20, a different marking mechanism must be used. Since the frame alignment signal is 6 bytes (48 bits) and per ITU-T G.798 only 32 bits must be checked for frame alignment, the 3rd OA2 byte position will be borrowed as a logical lane marker. For maximum skew detection range, the lane marker value will increment on successive frames from 0-239 (240 values being the largest multiple of 20 that can be represented in 8-bits). The logical lane number can be recovered from this value by a modulo 20 operation. Figure C.3 illustrates how bytes of the OTU4 are distributed in 16-byte increments across the 20 logical lanes.

As with OTU3, the FAS (with the substitution for lane marking, a pattern of  $3 \times \text{OA1} + 2 \times \text{OA2}$  followed by the lane identifier byte once per  $4080 \times 4$  bytes on each lane) is recovered at the sink on each logical lane. The lanes are identified, deskewed, and reassembled into the original OTU4 frame according to the lane marker. The MFAS can be combined with the lane marker to provide additional skew detection range, the maximum being up to one half LCM(240, 255) or 1912 OTU4 frame periods (approximately 2.223 ms). In mapping from lanes back to the OTU4 frame, the 6th byte of each OTU4 frame which was borrowed for lane marking is restored to the value OA2.

Each physical lane of an OTM-0.4v4 interface is formed by simple bit multiplexing of five logical lanes. At the sink, the bits are disinterleaved into five logical lanes from each physical lane. The sink will identify each logical lane according to the lane marker in the 3rd OA2 byte position. The sink must be able to accept the logical lanes in any position as the ordering of bit multiplexing on each physical lane is arbitrary, and the optical module hardware to be used for this application may not preserve the ordering of lanes and bit order (except within a logical lane).

NOTE 4 – Ten lane IEEE 100GBASE-R interfaces are specified, although not with ITU-T physical layer specifications. These interfaces may be compatible with a 10-lane interface for OTU4, each lane consisting of two bit-multiplexed logical lanes.

Lane 0	1:16(FAS)	65:80	...	16247:16262	49:64	...	16305:16320	33:48	...	16289:16304	17:32	...	16263:16288	1:16(FAS)
Lane 1	17:32	81:96	...	16263:16288	1:16(FAS)	...	16247:16262	49:64	...	16305:16320	33:48	...	16289:16304	17:32
Lane 2	33:48	97:112	...	16289:16304	17:32	...	16263:16288	1:16(FAS)	...	16247:16262	49:64	...	16305:16320	33:48
Lane 3	49:64	113:128	...	16305:16320	33:48	...	16289:16304	17:32	...	16263:16288	1:16(FAS)	...	16247:16262	49:64

Figure C.2 – Distribution of bytes from OTU3 to parallel lanes

Lane 0	1:16(FAS)	...	16001:16016	305:320	...	33:48	...	16033:16048	17:32	...	16017:16032	1:16(FAS)
Lane 1	17:32	...	16017:16032	1:16(FAS)	...	49:64	...	16049:16064	33:48	...	16033:16048	17:32
Lane 18	289:304	...	16289:16304	273:288	...	1:16(FAS)	...	16001:16016	305:320	...	16305:16320	289:304
Lane 19	305:320	...	16305:16320	289:304	...	17:32	...	16017:16032	1:16(FAS)	...	16001:16016	305:320

Figure C.3 – Distribution of bytes from OTU4 to parallel lanes

This mechanism handles any normally framed OTU3 or OTU4 sequence. The additional sequence to be handled is OTU3-AIS or OTU4-AIS, which is an unframed PN-11 sequence at the OTU3 or OTU4 rate, respectively. The source function for this adaptation will detect OTUk-AIS by recognizing the PN-11 sequence.

While receiving OTU3-AIS, the source function for distributing the OTU3 to parallel lanes will generate an OTUk framing pattern ( $3 \times OA1 + 3 \times OA2$  bytes) followed by an MFAS byte which is fixed at 0xFF once per  $4080 \times 4$  bytes, at one quarter the OTU3 bit-rate (the OTL3.4 bit rate as specified in Table 7-5). The remainder of the frame is the PN-11 pattern distributed in 16-byte increments across the lanes. When the sink function sees the MFAS fixed at 0xFF on any lane, it will generate a PN-11 sequence at the OTU3 rate in the egress direction.

While receiving OTU4-AIS, the source function for distributing the OTU4 to parallel lanes will generate the 6-byte pattern  $3 \times OA1 + 2 \times OA2 + 0xFF$  once per  $4080 \times 4$  bytes at the OTL4.4 bit rate as specified in Table 7-5 on each of the logical lanes. The remainder of the frame is the PN-11 pattern distributed in 16-byte increments across the lanes. When the sink function finds 0xFF as a lane marker on any lane, it will generate a PN-11 sequence at the OTU4 rate in the egress direction.

### 2.33) Appendix I

*Modify the text of the second and third paragraphs as follows:*

The +1/0/-1 mapping in 17.1 provides for one positive justification opportunity (PJO) and one negative justification opportunity (NJO) in each ODUk frame. The +2/+1/0/-1 mapping in clause 19 provides for 2 PJOs and one NJO in each ODUk frame. For the case of ODU multiplexing (i.e., the latter case), the ODUj being mapped will get only a fraction of the full payload capacity of the ODUk. There can be, in general, a number of fixed stuff bytes per ODUj or CBR client. Note that in both mapping cases, there is one stuff opportunity in every ODUk frame. For mapping of a CBR client into ODUk, the CBR client is allowed to use all the stuff opportunities (because only one CBR client signal is mapped into an ODUk). However, for mapping ODUj into ODUk (k > j), the ODUj can only use 1/2 (ODU0 into ODU1), 1/4 (ODU1 into ODU2 or ODU2 into ODU3) or 1/16 (ODU1 into ODU3) of the stuff opportunities ~~(the former for mapping ODU1 into ODU2 or ODU2 into ODU3; the latter for mapping ODU1 into ODU3)~~. The other stuff opportunities are needed for the other clients being multiplexed into the ODUk.

Traditionally, the justification ratio (stuff ratio) for purely positive justification schemes is defined as the long-run average fraction of justification opportunities for which a justification is done (i.e., for a very large number of frames, the ratio of the number of justifications to the total number of justification opportunities). In the +1/0/-1 scheme, positive and negative justifications must be distinguished. This is done by using different algebraic signs for positive and negative justifications. With this convention, the justification ratio can vary at most (for sufficiently large frequency offsets) from -1 to +1 (in contrast to a purely positive justification scheme, where justification ratio can vary at most from 0 to 1). In the case of ODUk multiplexing, the justification ratio is defined relative to the stuff opportunities available for the client in question. Then, the justification ratio can vary (for sufficiently large frequency offsets) from -1 to +2. (If the justification ratio were defined relative to all the stuff opportunities for all the clients, the range would be -1/2 to +1 for multiplexing ODU0 into ODU1, -1/4 to +1/2 for multiplexing ODU1 into ODU2 and ODU2 into ODU3, and -1/16 to +1/8 for multiplexing ODU1 into ODU3.)

### 2.34) Appendix I

*Add the following text at the end of Appendix I:*

#### **ODU0 into ODU1 multiplexing**

The ODU0 nominal client rate is (see clause 7.3):

$$S = \frac{1}{2}(R_{16}) \quad (\text{I-31})$$

The ODU1 nominal frame time is:

$$T = \frac{(3824)(4)}{\frac{239}{238}(R_{16})} \quad (\text{I-32})$$

The fraction  $p$  is 0.5. Inserting into Equation I-3 produces:

$$\frac{1}{2}R_{16} \frac{(3824)(4)}{\frac{239}{238}(R_{16})} \beta = \frac{\alpha}{2} + 7616 - N \quad (\text{I-33})$$

Simplifying and solving for  $\alpha$  produces:

$$\alpha = \frac{238}{239}(15296)\beta + 2N - 15232 \quad (\text{I-34})$$

As before, let  $\beta = 1 + y$ , where  $y$  is the net frequency offset (and is very nearly equal to  $y_c - y_s$  for client and server frequency offset small compared to 1). Then:

$$\begin{aligned} \alpha &= \frac{238}{239}(15296) - 15232 + 2N + \frac{238}{239}(15296)y \\ &= 2N + 15232y \end{aligned} \quad (\text{I-35})$$

The total number of fixed stuff bytes  $N$  is zero, as given in clause 19.5.4. The client and mapper frequency offsets are in the range  $\pm 20$  ppm, as given in clause 7.3. Then, the net frequency offset  $y$  is in the range  $\pm 40$  ppm. Inserting these values into Equation I-35 gives for the range for  $\alpha$ :

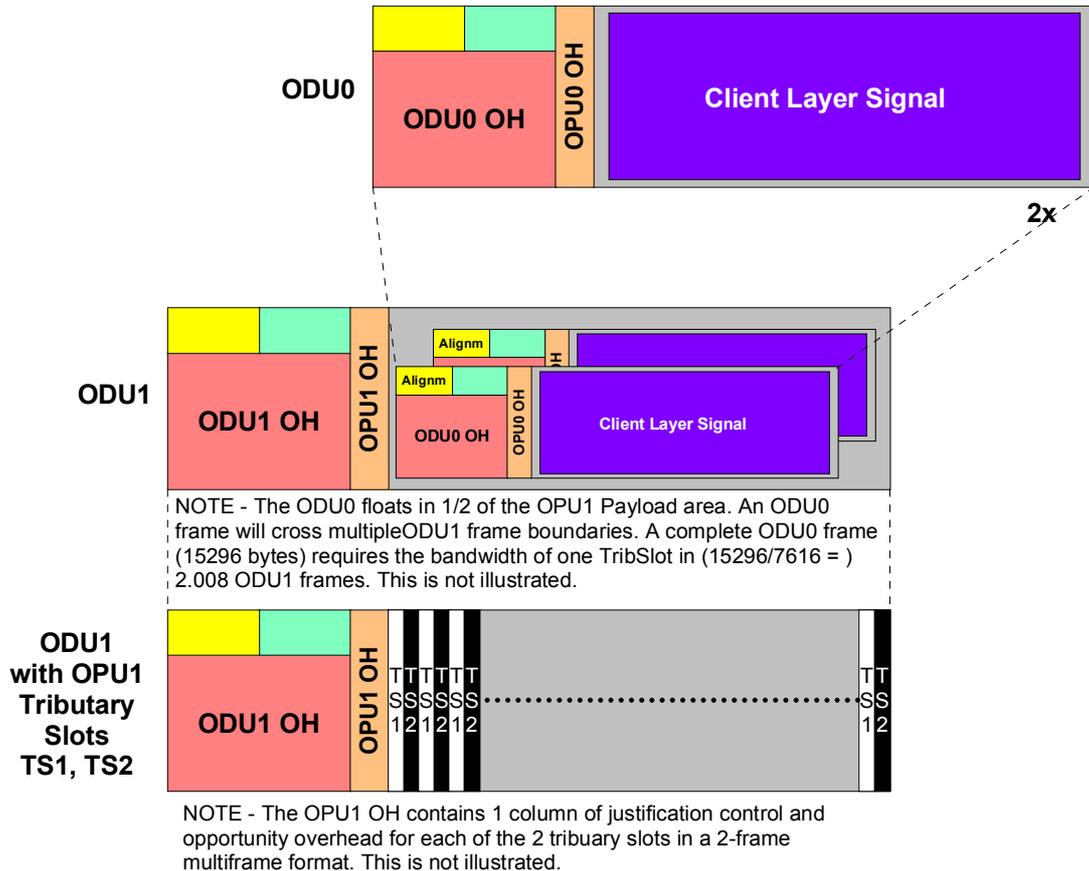
$$\begin{aligned} \alpha &= 0.6092800 && \text{for } y = +40 \text{ ppm} \\ \alpha &= 0.0000000 && \text{for } y = 0 \text{ ppm} \\ \alpha &= -0.6092800 && \text{for } y = -40 \text{ ppm} \end{aligned} \quad (\text{I-36})$$

In addition, stuff ratios of  $-2$  and  $+1$  are obtained for frequency offsets of  $-130$  ppm and  $65$  ppm, respectively. As above, the range of frequency offset that can be accommodated is approximately  $195$  ppm.

**2.35) Appendix III Example of ODUk multiplexing**

*Add the following text and figure at the end of Appendix III:*

Figure III.2 illustrates the multiplexing of two ODU0 signals into an ODU1. The ODU0 signals including the Frame Alignment Overhead and an all-0s pattern in the OTUk overhead locations are adapted to the ODU1 clock via justification (asynchronous mapping). These adapted ODU0 signals are byte interleaved into the OPU1 payload area, and their justification control and opportunity signals (JC, NJO) are frame interleaved into the OPU1 overhead area and ODU1 overhead is added.



**Figure III.2 – Example of multiplexing 2 ODU0 signals into an ODU1 (artist impression)**

## 2.36) New Appendix VII

*Add new Appendix VII as follows:*

# Appendix VII

## Adaptation of parallel 64B/66B encoded clients

(This appendix does not form an integral part of this Recommendation)

### VII.1 Introduction

IEEE 40GBASE-R and 100GBASE-R interfaces currently being specified by the IEEE P802.3ba task force will be parallel interfaces intended for short-reach (up to 40 km) interconnection of Ethernet equipment. This appendix describes the process of converting the parallel format of these interfaces into a serial bit stream to be carried over OTN.

The order of transmission of information in all the diagrams in this appendix is first from left to right and then from top to bottom.

### VII.2 Clients signal format

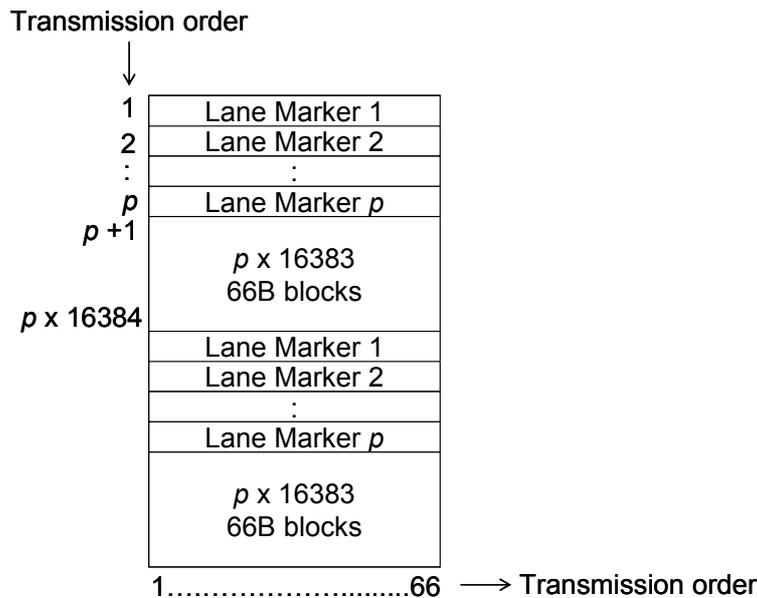
40GBASE-R and 100GBASE-R clients are initially parallel interfaces, but in the future may be serial interfaces. Independent of whether these interfaces are parallel or serial, or what the parallel interface lane count is, 40GBASE-R signals are comprised of four PCS lanes and 100GBASE-R signals are comprised of twenty PCS lanes. If the number of physical lanes on the interface is fewer than the number of PCS lanes, the appropriate number of PCS lanes are bit-multiplexed onto each physical lane of the interface. Each PCS lane consists of 64B/66B encoded data with a PCS lane alignment marker inserted on each lane once per 16384 66-bit blocks. The PCS lane alignment marker itself is a special format 66B codeword.

The use of this adaptation for 40GBASE-R into OPU3 also applies the transcoding method which appears in Annex B and the framing method which appears in Appendix VIII. The adaptation described in this appendix alone can be used for adaptation of 100GBASE-R into OPU4.

### VII.3 Client frame recovery

Client framing recovery consists of the following:

- Disinterleave the PCS lanes, if necessary. This is necessary whenever the number of PCS lanes and the number of physical lanes is not equal, and is not necessary when they are equal (e.g., a 4-lane 40GBASE-R interface).
- Recover 64B/66B block lock per the state diagram in Figure 49-12 of IEEE 802.3 (or Figure 82-12 of IEEE 802.3ba draft 1.0).
- Recover Lane Alignment marker framing on each PCS lane per the state diagram in Figure 82-13 of IEEE 802.3ba.
- Reorder and deskew the PCS lanes into a serialized stream of 66B blocks (including lane alignment markers). Figure VII.1 illustrates the ordering of 66B blocks after the completion of this process for an interface with p PCS lanes.



**Figure VII.1 – Deskewed/serialized stream of 66B blocks**

Each 66B codeword is one of the following:

- A set of eight data bytes with a sync header of "01";
- A control block (possibly including seven or fewer data octets) beginning with a sync header of "10";
- A PCS lane alignment marker, also encoded with a sync header of "10". Of the 8 octets following the sync header, 6 octets have fixed values allowing the lane alignment markers to be recognized (see Tables VII.1 and VII.2). The fourth octet following the sync header is a BIP-8 calculated over the data from one alignment marker to the next. The eighth octet is the complement of this BIP-8 value to maintain DC balance. Note that these BIP-8 values are not manipulated by the mapping or demapping procedure, but simply skipped in the process of recognizing lane alignment markers and copied intact as they are used for monitoring the error ratio of the Ethernet link between Ethernet PCS sublayers.

For all-data blocks and control blocks, the 64 bits following the sync header are scrambled as a continuous bit-stream (skipping sync headers and PCS lane alignment markers) according to the polynomial  $G(x) = 1 + x^{39} + x^{58}$ .

These 66B blocks are re-distributed to PCS lanes at the egress interface. The 66B blocks (including PCS lane alignment markers) resulting from the decoding process are distributed round-robin to PCS lanes. If the number of PCS lanes is greater than the number of physical lanes of the egress interface, the appropriate numbers of PCS lanes are bit-multiplexed onto the physical lanes of the egress interface.

### VII.3.1 40GBASE-R client frame recovery

PCS Lane Alignment Markers have the values shown in Table VII.1 for 40GBASE-R signals which use PCS lane numbers 0-3. Note that these values will need to be aligned with the published IEEE 802.3ba amendment once it is approved.

**Table VII.1 – PCS lane alignment marker format for 40GBASE-R**

Lane number	SH	Encoding
0	10	0x90, 0x76, 0x47, BIP <sub>3</sub> , 0x6f, 0x89, 0xb8, BIP <sub>7</sub>
1	10	0xf0, 0xc4, 0xe6, BIP <sub>3</sub> , 0x0f, 0x3b, 0x19, BIP <sub>7</sub>
2	10	0xc5, 0x65, 0x9b, BIP <sub>3</sub> , 0x3a, 0x9a, 0x64, BIP <sub>7</sub>
3	10	0xa2, 0x79, 0x3d, BIP <sub>3</sub> , 0x5d, 0x86, 0xc2, BIP <sub>7</sub>

Since 40GBASE-R client signal must be transcoded into 1024B/1027B for rate reduction, the 64B/66B PCS receive process at the ingress interface further descrambles the bit-stream skipping sync headers and PCS lane alignment markers, and the 64B/66B PCS transmit process at the egress interface scrambles the bit-stream again skipping sync headers and PCS lane alignment markers, as shown in Figure VII.1.

### VII.3.2 100GBASE-R client frame recovery

PCS Lane Alignment Markers have the values shown in Table VII.2 for 100GBASE-R signals which use PCS lane numbers 0-19. Note that these values will need to be aligned with the published IEEE 802.3ba amendment once it is approved.

**Table VII.2 – PCS lane alignment marker format for 100GBASE-R**

Lane number	SH	Encoding	Lane number	SH	Encoding
0	10	0xc1, 0x68, 0x21, BIP <sub>3</sub> , 0x3e, 0x97, 0xde, BIP <sub>7</sub>	10	10	0xfd, 0x6c, 0x99, BIP <sub>3</sub> , 0x02, 0x93, 0x66, BIP <sub>7</sub>
1	10	0x9d, 0x71, 0x8e, BIP <sub>3</sub> , 0x62, 0x8e, 0x71, BIP <sub>7</sub>	11	10	0xb9, 0x91, 0x55, BIP <sub>3</sub> , 0x46, 0x6e, 0xaa, BIP <sub>7</sub>
2	10	0x59, 0x4b, 0xe8, BIP <sub>3</sub> , 0xa6, 0xb4, 0x17, BIP <sub>7</sub>	12	10	0x5c, 0x b9, 0xb2, BIP <sub>3</sub> , 0xa3, 0x46, 0x4d, BIP <sub>7</sub>
3	10	0x4d, 0x95, 0x7b, BIP <sub>3</sub> , 0xb2, 0x6a, 0x84, BIP <sub>7</sub>	13	10	0x1a, 0xf8, 0xbd, BIP <sub>3</sub> , 0xe5, 0x07, 0x42, BIP <sub>7</sub>
4	10	0xf5, 0x 07, 0x09, BIP <sub>3</sub> , 0x0a, 0xf8, 0xf6, BIP <sub>7</sub>	14	10	0x83, 0xc7, 0xca, BIP <sub>3</sub> , 0x7c, 0x38, 0x35, BIP <sub>7</sub>
5	10	0xdd, 0x14, 0xc2, BIP <sub>3</sub> , 0x22, 0xeb, 0x3d, BIP <sub>7</sub>	15	10	0x35, 0x36, 0xcd, BIP <sub>3</sub> , 0xca, 0xc9, 0x32, BIP <sub>7</sub>
6	10	0x9a, 0x4a, 0x26, BIP <sub>3</sub> , 0x65, 0xb5, 0xd9, BIP <sub>7</sub>	16	10	0xc4, 0x31, 0x4c, BIP <sub>3</sub> , 0x3b, 0xce, 0xb3, BIP <sub>7</sub>
7	10	0x7b, 0x45, 0x66, BIP <sub>3</sub> , 0x84, 0xba, 0x99, BIP <sub>7</sub>	17	10	0xad, 0xd6, 0xb7, BIP <sub>3</sub> , 0x52, 0x29, 0x48, BIP <sub>7</sub>
8	10	0xa0, 0x24, 0x76, BIP <sub>3</sub> , 0x5f, 0xdb, 0x89, BIP <sub>7</sub>	18	10	0x5f, 0x66, 0x2a, BIP <sub>3</sub> , 0xa0, 0x99, 0xd5, BIP <sub>7</sub>
9	10	0x68, 0xc9, 0xfb, BIP <sub>3</sub> , 0x97, 0x36, 0x04, BIP <sub>7</sub>	19	10	0xc0, 0xf0, 0xe5, BIP <sub>3</sub> , 0x3f, 0x0f, 0x1a, BIP <sub>7</sub>

### VII.4 Additions to Annex B transcoding for parallel 64B/66B clients

When OPUk is large enough for the serialized 66B block stream (e.g., for 100GBASE-R client signals into OPU4), the recovered client frames are adapted directly per this appendix.

When used in combination with the transcoding into 513B code blocks described in Annex B (e.g., for 40GBASE-R client signals into OPU3), this clause describes the additions to the Annex B transcoding process for transport of PCS lane alignment markers.

PCS lane alignment markers are encoded together with 66B control blocks into the uppermost rows of the 513B code block shown in Figure B.4 together with 66B control blocks. The flag bit "F" of the 513B structure is 1 if the 513B structure contains at least one 66B control block or PCS lane alignment marker, and 0 if the 513B structure contains eight all-data 66B blocks.

The transcoding into 512B/513B must encode PCS lane alignment marker into a row of the structure shown in Figure B.3 as follows: The sync header of "10" is removed. The PCS lane number is determined by examining the rest of the bytes in the block for the values shown in Table VII.1 or VII.2. The first byte of the row will contain the structure shown in Figure B.4, with a CB-TYPE field of "0100". The POS field will indicate the position where the PCS lane alignment marker was received among the group of eight 66B codewords being encoded into this 513B block. The flag continuation bit "FC" will indicate whether any other 66B control blocks or PCS lane

alignment markers are encoded into rows below this one in the 513B block. Beyond this first byte, the other seven bytes of the row are populated with the PCS lane number, which can be validated at the decoder by taking a majority vote across three or more of the bytes. At the decoder, a PCS lane alignment marker will be generated in the position indicated by the POS field among any 66B all-data blocks contained in this 513B block, the sync header of "10" is generated followed by the bit pattern indicated for the particular PCS lane in Table VII.1 or VII.2.

#### **VII.4.1 Errors detected by mapper**

Errors encountered before the mapper, such as loss of client signal on any physical lane of the interface, will result in the insertion of an Ethernet LF sequence ordered set prior to this process. The same action should be taken as a result of failure to achieve 66B block lock on any PCS lane, failure to achieve lane alignment marker framing on each PCS lane, or failure to deskew because the skew exceeds the buffer available for deskew.

An invalid 66B block will be converted to an error control block before transcoding or direct adaptation. An invalid 66B block is one which does not have a sync header of "01" or "10", or one which has a sync header of "10" and a control block type field which does not appear in Figure B.2 (and for 40GBASE-R and 100GBASE-R, is not a valid PCS lane alignment marker). An error control block has sync bits of "10", a block type code of 0x1e, and 8 seven-bit /E/ error control characters. This will prevent the Ethernet receiver from interpreting a sequence of bits containing this error as a valid packet.

## 2.37) New Appendix VIII

Add new Appendix VIII as follows:

### Appendix VIII

#### Improved robustness for mapping of 40GBASE-R into OPU3 using 1027B code blocks

(This appendix does not form an integral part of this Recommendation)

##### VIII.1 Introduction

When a parallel 40GBASE-R signal is transcoded per Annex B and directly mapped into OPU3 without GFP framing, another method is needed to locate the start of 513B blocks and to provide protection to prevent that bit errors create an unacceptable increase in mean time to false packet acceptance (MTTFPA).

##### VIII.2 513B code block framing and Flag bit protection

The mapping of 513B code blocks into OPU3 requires a mechanism for locating the start of the code blocks. A mechanism is also needed to protect the flag bit, whose corruption could cause data to be erroneously interpreted as control and vice versa.

Both of these requirements can be addressed by providing parity across the flag bits of two 513B blocks produced from the transcoding of Annex B.

Figure VIII.1 illustrates the flag bit parity across two 513B blocks. This creates a minimum two-bit Hamming distance between valid combinations of flag bits.

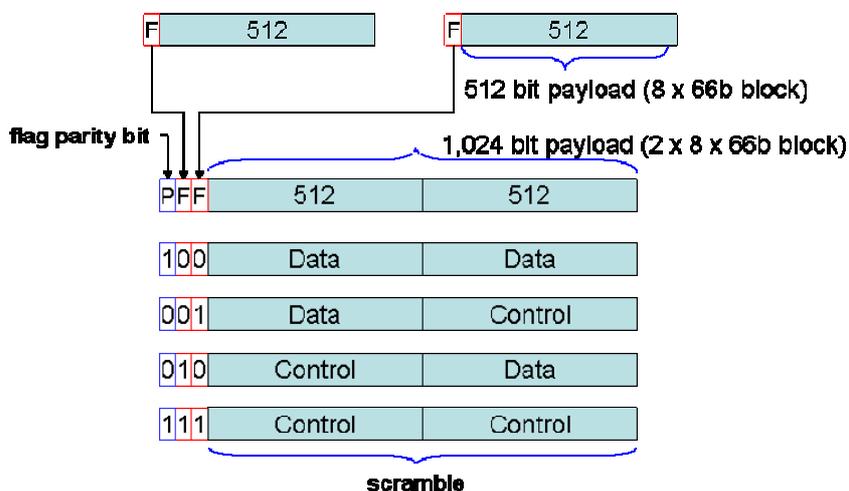


Figure VIII.1 – Flag parity bit on two 513B blocks (1027B code)

The flag bit parity creates a sequence that can be used for framing to locate the 513B blocks in a stream of bits. The state diagram of Figure 49-12 of IEEE 802.3 is applied to locate a 3-bit pattern appearing once per 1027 bits (rather than a 2-bit pattern appearing once per 66 bits) where four out of eight 3-bit sequences (rather than two out of four two-bit values as used in IEEE 802.3) match the pattern. The additional step required is to scramble the non-flag or flag parity bits so that the legal sequences of these bits are not systematically mimicked in the data itself. The scrambler to be

used for this purpose is the Ethernet self-synchronous scrambler using the polynomial  $G(x) = 1 + x^{39} + x^{58}$ .

At the demapper, invalid flag bit parity will cause both of the 513B blocks across which the flag bit parity applies to be decoded as  $8 \times 2$  66B error control blocks ("10" sync header, control block type 0x1e, followed by eight 7-bit /E/ control characters).

### VIII.3 66B block sequence check

Bit error corruption of the position or flag continuation bits could cause 66B blocks to be demapped from 513B code blocks in the incorrect order. Additional checks are performed to prevent that this results in incorrect packet delineation. Since detectable corruption normally means that the proper order of 66B blocks to construct at the decoder cannot be reliably determined, if any of these checks fail, the decoder will transmit eight 66B error control blocks (sync = "10", control block type = 0x1e, and eight 7-bit /E/ control characters).

Other checks are performed to reduce the probability that invalid data is delivered at the egress in the event that bit errors have corrupted any of the POS fields or flag continuation bits "FC".

If the Flag bit "F" is 1 (i.e., the 513B block includes at least one 64B/66B control block), for the rows of the table up until the first one with a flag continuation bit of zero (the last one in the block), it is verified that no two 66B control blocks or lane alignment markers within that 513B block have the same value in the POS field, and further, that the POS field values for multiple control or lane alignment rows are in ascending order, which will always be the case for a properly constructed 513B block. If this check fails, the 513B block is decoded into eight 66B error control blocks.

The next check is to ensure that the block sequence corresponds to well-formed packets, which can be done according to the state diagram in Figures VIII.2 and VIII.3. This check will determine if 66B blocks are in an order that does not correspond to well-formed packets, e.g., if during an IPG you find an all-data 66B block without first seeing a control block representing packet start, or if during a packet you see control/idle block without first seeing a control block representing packet termination, control blocks have likely been misordered by corruption of either the POS bits or a flag continuation bit. Failure of this check will cause the 513B block to be decoded as eight 66B error control blocks. Note that PCS lane alignment markers are accepted in either state and do not change state as shown in Figure VIII.3.

The sequence of PCS lane alignment markers is also checked at the decoder. For an interface with  $p$  PCS lanes, the PCS lane alignment markers for lanes 0 through  $p-1$  will appear in a sequence, followed by  $16383 \times p$  non-lane-marker 66B blocks, followed by another group of PCS lane alignment markers. A counter is maintained at the decoder to keep track of when the next group of lane alignment markers is expected. If, in the process of decoding lane alignment markers from a 513B block, a lane alignment marker is found in a position where it is not expected, or a lane alignment marker is missing in a position where it would have been expected, the entire 513B block is decoded as eight 66B error control blocks as shown in Figures VIII.2, VIII.3, and VIII.4.

#### VIII.3.1 State diagram conventions

The body of this subclause is comprised of state diagrams, including the associated definitions of variables, constants, and functions. Should there be a discrepancy between a state diagram and descriptive text, the state diagram prevails.

The notation used in the state diagrams follows the conventions of 21.5 of IEEE 802.3. State diagram timers follow the conventions of 14.2.3.2 of IEEE 802.3. The notation ++ after a counter or integer variable indicates that its value is to be incremented.

## VIII.3.2 State variables

### VIII.3.2.1 Constants

EBLOCK\_T<65:0>

66-bit vector to be sent to the PCS containing /E/ in all the eight character locations

Mi<65:0>

66-bit vector containing the transcoded alignment marker of i-th PCS lane ( $0 < i \leq p$ ). ( $p = 4$  for 40GBASE-R, and  $p = 20$  for 100GBASE-R).

### VIII.3.2.2 Variables

align\_status

A variable set by the PCS deskew process to reflect the status of the lane-to-lane alignment. Set true when all lanes are synchronized and aligned, set false when the deskew process is not complete.

high\_ber

Boolean variable which is asserted true when the ber\_cnt exceeds 16 indicating a bit error ratio  $>10^{-4}$ .

Mseq\_violation

Boolean variable that is set in each PCS lane alignment marker cycle based on the PCS lane marker position and order. It is true if the unexpected marker sequence is detected and false if not.

POS\_violation

Boolean variable that is set in each rx513\_raw<527:0> based on the POS field values for rx\_tcd<65:0>. It is true if the two or more have the same POS values or if they are not in ascending order, and false if their POS values are in ascending order.

reset

Boolean variable that controls the resetting of the PCS. It is true whenever a reset is necessary including when reset is initiated from the MDIO, during power on, and when the MDIO has put the PCS into low-power mode.

Rx513\_coded<512:0>

Vector containing the input to the 512B/513B decoder.

rx513\_raw<527:0>

Vector containing eight successive 66 bit vectors (tx\_coded).

rx\_tcd<65:0>

66-bit vector transcoded-decoded from a 513-bit block following the rules shown in Figure B.5.

seq\_violation

Boolean variable that is set in each rx513\_raw<527:0> based on the sequence check on a rx\_tcd<65:0> stream. It is true if the unexpected sequence is detected and false if not.

tx\_coded<65:0>

Vector containing the output from the transcoder that converts 512B/513B to 64B/66B

### VIII.3.2.3 Functions

DECODE(rx513\_coded<512:0>)

Decodes the 513-bit vector returning rx513\_raw<527:0> which is sent to client interface. The DECODE function shall decode the block as specified in Figure VIII.2.

MDECODE(rx\_tcd<65:0>)

Decodes the 66-bit vector returning reconstructed PCS lane alignment marker.

R\_BLOCK\_TYPE = {C, S, T, D, E, M}

This function classifies each 66-bit rx\_tcd vector as belonging to one of the six types depending on its contents.

Values: C, S, T, and D are defined in 49.2.13.2.3 in IEEE 802.3.

M: The vector contains a sync header of 10 and is recognized as a valid PCS lane alignment marker by using the state machine shown in Figure VIII.3.

E: The vector does not meet the criteria for any other value.

R\_TYPE(rx\_tcd<65:0>)

Returns the R\_BLOCK\_TYPE of the rx\_tcd<65:0> bit vector.

R\_TYPE\_NEXT

Prescient end of packet check function. It returns the R\_BLOCK\_TYPE of the rx\_tcd vector immediately following the current rx\_tcd vector.

### VIII.3.2.4 Counters

cnt

Count up to a maximum of  $p$  of the number of PCS lanes.

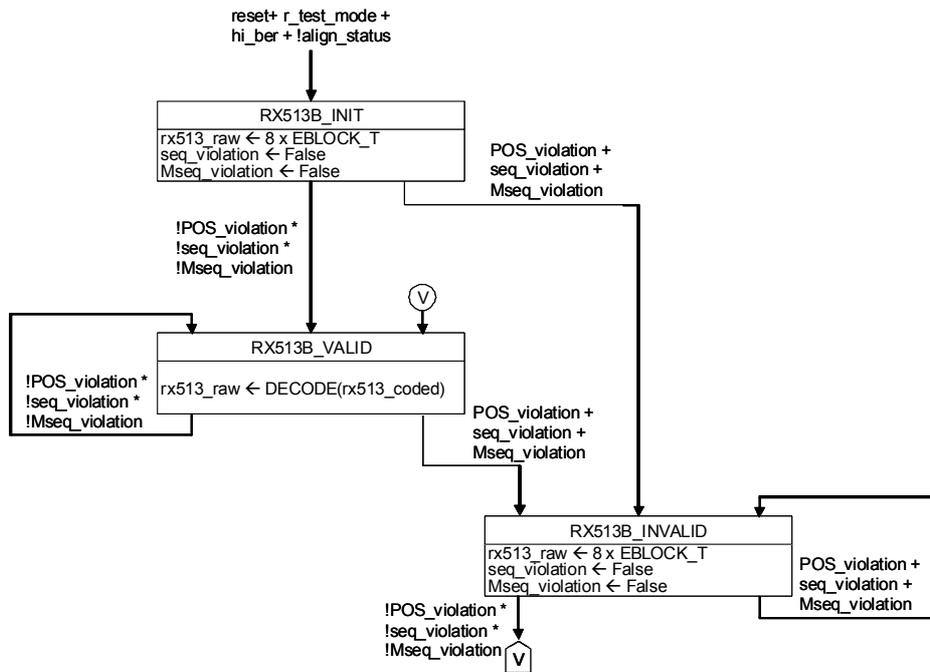
### VIII.3.2.5 Test mode control

r\_test\_mode

Boolean variable controlling receive channel operating mode. When false, the receive channel operates in normal mode. When true, the receive channel operates in test-pattern mode.

### VIII.3.3 State diagrams

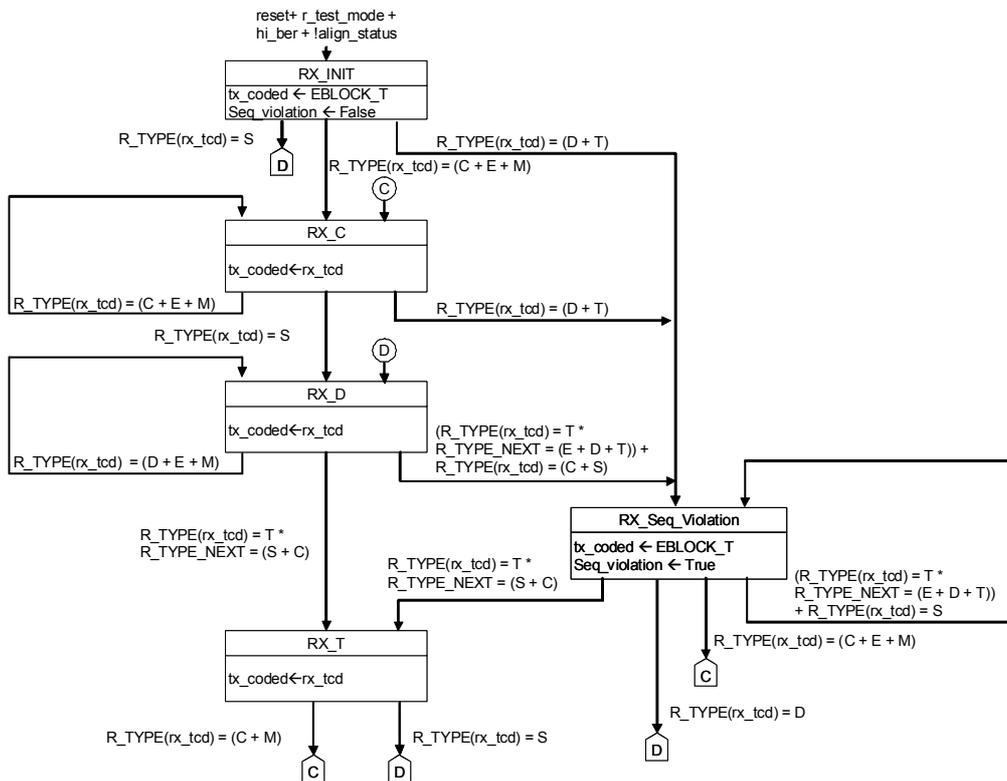
The Receive state machine for a series of 513-bit blocks shown in Figure VIII.2 determines whether the 513-bit block contains valid eight 66-bit blocks or not.



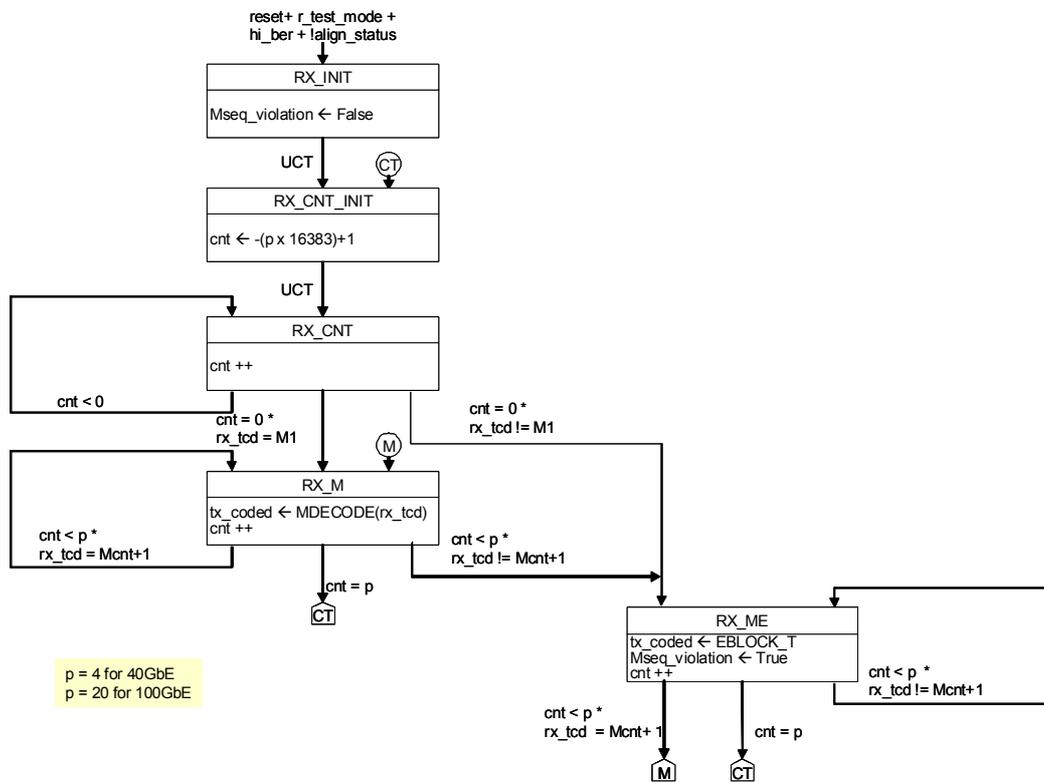
**Figure VIII.2 – Receive state machine for the 512B/513B code blocks including lane alignment markers**

The Trans-decode state machine for a series of 66-bit blocks shown in Figure VIII.3 checks the block type sequence of recovered 66-bit blocks.

The PCS lane alignment marker state machine for a series of 66-bit blocks shown in Figure VIII.4 detects the alignment markers every  $p \times 16384$  blocks and checks whether the marker is in ascendant order or not.



**Figure VIII.3 – Trans-decode state machine for the 64B/66B code blocks including the lane alignment markers**



**Figure VIII.4 – Receive state machine for the lane alignment markers**

## 2.38) New Appendix IX

Add new Appendix IX as follows:

### Appendix IX

#### Parallel logic implementation of the CRC-8

(This appendix does not form an integral part of this Recommendation)

Table IX.1 illustrates example logic equations for a parallel implementation of the CRC-8. An "X" in a row of the table indicates that the message bit of that column is an input to the Exclusive-OR equation for calculating the CRC bit of that row. JC1.C1 corresponds to the first bit (MSB) of the first mapping overhead octet (JC1); JC1.C2 corresponds to bit 2 of the first mapping overhead octet, etc. After computation, CRC bits crc1 to crc8 are inserted into the JC3 octet, with crc1 occupying MSB and crc8 the LSB of the octet.

**Table IX.1 – Parallel logic equations for the CRC-8 implementation**

mapping overhead bits	CRC checksum bits							
	crc1	crc2	crc3	crc4	crc5	crc6	crc7	crc8
JC1.C1		X				X		X
JC1.C2	X		X			X		
JC1.C3		X		X			X	
JC1.C4			X		X			X
JC1.C5	X			X			X	
JC1.C6		X			X			X
JC1.C7	X		X				X	
JC1.C8		X		X				X
JC2.C9	X		X		X	X	X	
JC2.C10		X		X		X	X	X
JC2.C11	X		X		X	X		X
JC2.C12	X	X		X				
JC2.C13		X	X		X			
JC2.C14			X	X		X		
JC2.II				X	X		X	
JC2.DI					X	X		X



ITU-T Y-SERIES RECOMMENDATIONS

**GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS**

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
<b>Transport</b>	<b>Y.1300–Y.1399</b>
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Numbering, naming and addressing	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Future networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999

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

## SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
<b>Series G</b>	<b>Transmission systems and media, digital systems and networks</b>
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	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	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
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
<b>Series Y</b>	<b>Global information infrastructure, Internet protocol aspects and next-generation networks</b>
Series Z	Languages and general software aspects for telecommunication systems