Flex Ethernet 2.0 Implementation Agreement

IA # OIF-FLEXE-02.0

June 22, 2018

Implementation Agreement created and approved
by the Optical Internetworking Forum
www.oiforum.com

The OIF is an international non profit organization with over 90 member companies, including the world's leading carriers and vendors. Being an industry group uniting representatives of the data and optical worlds, OIF's purpose is to accelerate the deployment of interoperable, cost-effective and robust optical internetworks and their associated technologies. Optical internetworks are data networks composed of routers and data switches interconnected by optical networking elements.

With the goal of promoting worldwide compatibility of optical internetworking products, the OIF actively supports and extends the work of national and international standards bodies. Working relationships or formal liaisons have been established with IEEE 802.1, IEEE 802.3, IETF, IP-MPLS Forum, IPv6 Forum, ITU-T SG13, ITU-T SG15, MEF, ATIS-OPTXS,ATIS-TMOC, TMF and the XFP MSA Group.

For additional information contact:
The Optical Internetworking Forum, 48377 Fremont Blvd.,
Suite 117, Fremont, CA 94538
510-492-4040 Φ info@oiforum.com

www.oiforum.com

Working Group: **Physical and Link Layer**

TITLE: **Flex Ethernet Implementation Agreement 2.0**

SOURCE:

TECHNICAL EDITOR

Stephen J. Trowbridge, Ph. D.
Nokia
5280 Centennial Trail
Boulder, CO 80303 USA
Phone: +1 303 809 7423
Email: steve.trowbridge@nokia.com

WORKING GROUP CHAIR

David R. Stauffer, Ph.D.
Kandou Bus, S.A.
EPFL Innovation Park Bldg. I
1015 Lausanne Switzerland
Phone: +1 802 316-0808
Email: david@kandou.com

ABSTRACT: The Flex Ethernet (FlexE) Implementation Agreement provides a generic mechanism for supporting a variety of Ethernet MAC rates that may or may not correspond to any existing Ethernet PHY rate. This includes MAC rates that are both greater than (through bonding) and less than (through sub-rate and channelization) the Ethernet PHY rates used to carry FlexE. This can be viewed as a generalization of the Multi-Link Gearbox implementation agreements, removing the restrictions on the number of bonded PHYs (MLG2.0, for example, supports one or two 100GBASE-R PHYs) and the constraint that the FlexE Clients correspond to Ethernet rates (MLG2.0 supports only 10G and 40G clients).

FlexE 2.0 augments FlexE 1.0 by providing support for FlexE Groups composed of n x 200 Gb/s Ehernet PHYs and n x 400 Gb/s Ethernet PHYs, and several other features.

1 Table of Contents

2 List of Figures

3 List of Tables

4 Document Revision History

Working Group: Physical and Link Layer

SOURCE:

Editor's Name
Stephen J. Trowbridge, Ph. D.
Nokia
5280 Centennial Trail
Boulder, CO 80303 USA
Phone: +1 303 809 7423
Email: steve.trowbridge@nokia.com

Working Group Chair
David R. Stauffer, Ph.D.
Kandou Bus, S.A.
PFL Innovation Park Bldg. I
1015 Lausanne Switzerland
Phone: +1 802 316-0808
Email: david@kandou.com

DATE: **January 2018**

Issue No.	Issue Date	Details of Change
oif2017.256.00	July 2017	Initial Text Proposal
oif2017.256.02	August 2017	Results of initial review and consideration of contributions in 3Q2017 meeting (Halifax)
oif2017.256.03	January 2018	Incorporate resolutions to first straw ballot comments
oif2017.256.04	April 2018	Incorporate resolutions to second straw ballot comments and additional editorial changes resulting from contributions to the 2Q2018 TC meeting (Nuremberg)

5 Introduction

The Flex Ethernet (FlexE) implementation agreement provides a generic mechanism for supporting a variety of Ethernet MAC rates that may or may not correspond to any existing Ethernet PHY rate. This includes MAC rates that are both greater than (through bonding) and less than (through sub-rate and channelization) the Ethernet PHY rates used to carry FlexE. This can be viewed as a generalization of the Multi-Link Gearbox implementation agreements, removing the restrictions on the number of bonded PHYs (MLG2.0, for example, supports one or two 100GBASE-R PHYs) and the constraint that the FlexE Clients correspond to Ethernet rates (MLG2.0 supports only 10G and 40G clients).

FlexE 2.0 augments FlexE 1.0 by providing support for FlexE Groups composed of m x 200 Gb/s Ehernet PHYs and m x 400 Gb/s Ethernet PHYs, and several other features.

5.1 Requirements

The general capabilities supported by the FlexE implementation agreement are:

- Bonding of Ethernet PHYs, e.g., supporting a 200G MAC over two bonded 100GBASE-R PHYs.
- Sub-rates of Ethernet PHYs, e.g., supporting a 50G MAC over a 100GBASE-R PHY.
- Channelization within a PHY or a group of bonded PHYs, e.g, support a 150G and two 25G MACs over two bonded 100GBASE-R PHYs.

Note that hybrids are also possible, for example a sub-rate of a group of bonded PHYs, for example, a 250G MAC over three bonded 100GBASE-R PHYs.

The general approach is illustrated in Figure 1.

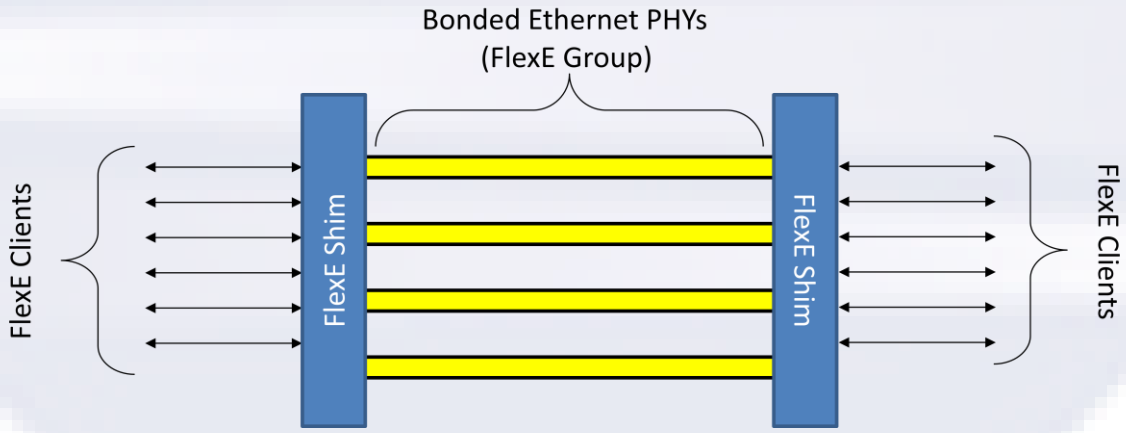


Figure 1: General Structure of FlexE

The *FlexE Group* refers to a group of from 1 to n 100G FlexE Instances that are carried by a group of from 1 to m bonded Ethernet PHYs. This version of the Implementation Agreement supports FlexE Groups composed of one or more bonded 100GBASE-R PHYs, one or more bonded 200GBASE-R PHYs, or one or more 400 Gb/s PHYs.

A *100G FlexE Instance* is a unit of information consisting of 100G of capacity able to carry FlexE Client data, together with its associated overhead. A *FlexE Client* is an Ethernet flow based on a MAC data rate that may or may not correspond to any Ethernet PHY rate. The FlexE Client MAC rates supported by FlexE Groups are 10, 40, and m ×

25 Gb/s. The FlexE Client MAC rates supported by FlexE Groups may support all, or only a subset of these FlexE Client rates, e.g., m × 25 Gb/s.

The *FlexE Shim* is the layer that maps or demaps the FlexE Clients carried over a FlexE Group. Similar to terminology of MLG, the *FlexE mux* refers to the transmit direction which maps the FlexE Clients over the FlexE Group. The *FlexE demux* refers to the receive direction which demaps the FlexE Clients from the FlexE Group.

5.2 Relationship to IEEE 802.3 Stack

The FlexE Shim can be envisioned as being in the middle of the PCS in the 100GBASE-R stack as illustrated in [802.3] Figure 80-1 or in the 200GBASE-R or 400GBASE-R stack as illustrated in [802.3bs] Figure 116-1. Each FlexE Client has its own separate MAC, Reconciliation Sublayer, and xMII above the FlexE Shim which operate at the FlexE Client rate. The layers below the PCS (100GBASE-R PMA, optional FEC, PMD) are used intact as specified for Ethernet.

5.2.1 FlexE mux functions

The functions of the FlexE mux (the FlexE Shim functions in the transmit direction) are illustrated in Figure 2 for FlexE Groups composed of 100GBASE-R PHYs, in Figure 3 for FlexE groups composed of 200GBASE-R PHYs, and in Figure 4 for FlexE Groups composed of 400GBASE-R PHYs.

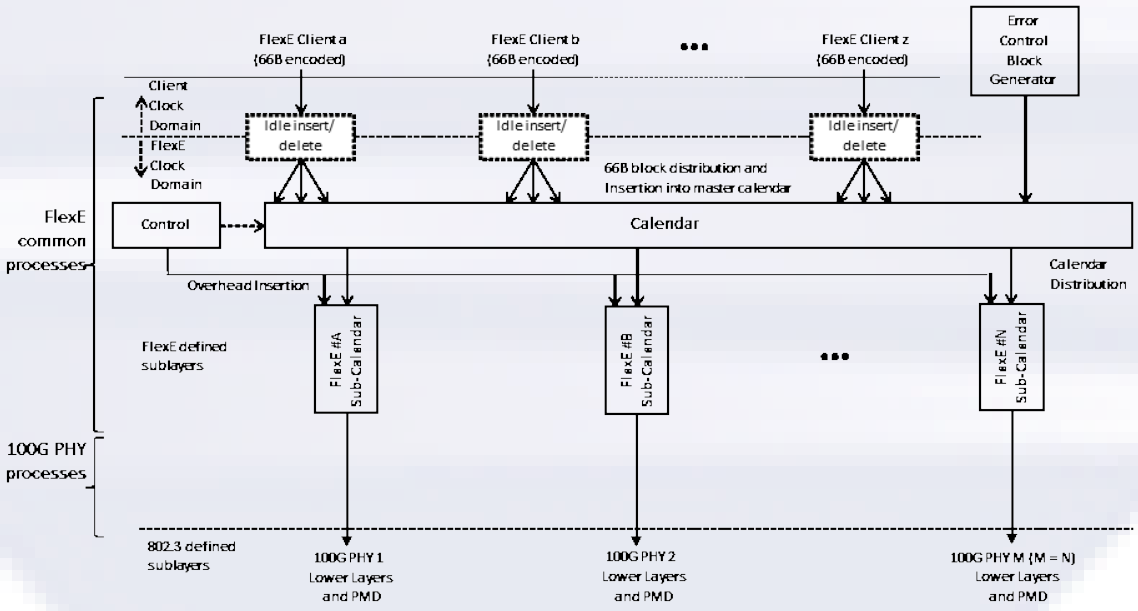


Figure 2: 100GBASE-R FlexE mux functions

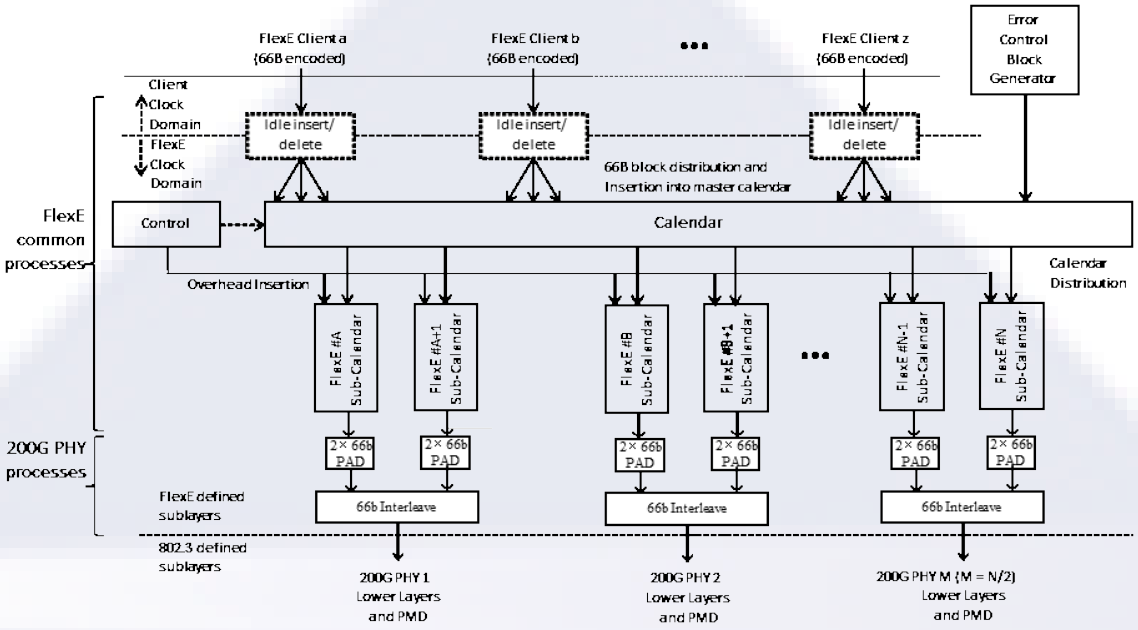


Figure 3: 200GBASE-R FlexE mux functions

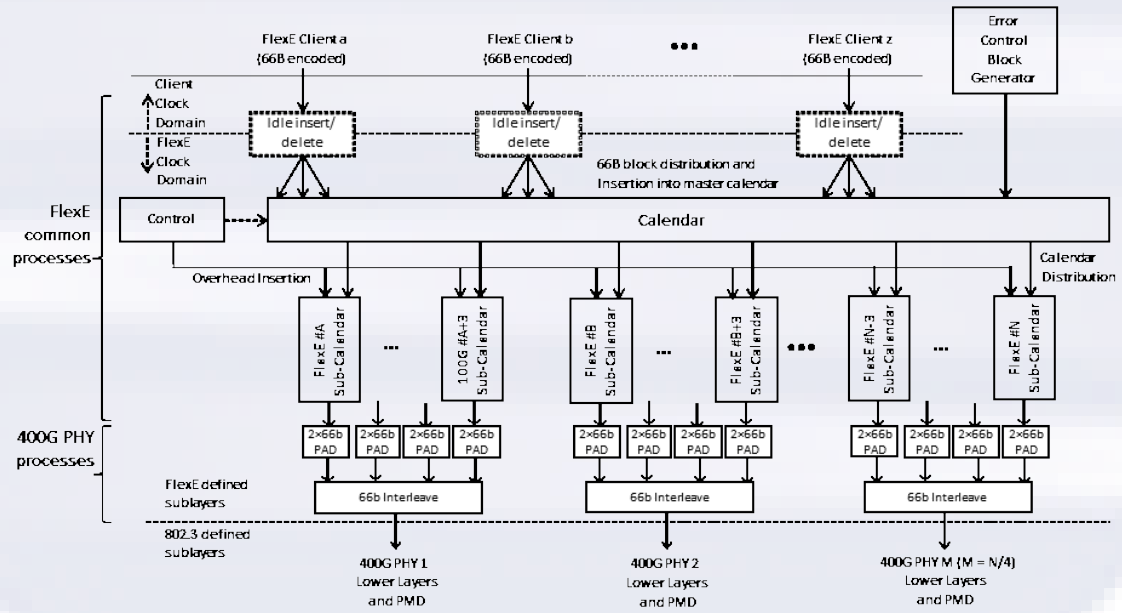


Figure 4: 400GBASE-R FlexE mux functions

5.2.1.1 FlexE Client

Each FlexE Client is presented to the FlexE Shim as a 64B/66B encoded bit-stream according to [802.3] Figure 82-5. How this bit-stream is created is application specific (see clause 7.2 for details), but should appear to the FlexE Shim as having been created from an Ethernet MAC operating at a rate of 10, 40, or m × 25 Gb/s (or a subset of these rates, e.g., m × 25 Gb/s), through a logical RS layer which performs the link fault

signaling functions described in [802.3] clause 81.3.4. The content of FlexE Client stream may be LF in the case of an upstream failure of the FlexE Client. The start control character is aligned to an 8-byte boundary, feeding through a logical xMII at the corresponding rate, and a 64B/66B encoder resulting in a FlexE Client rate of:

$$\frac{66}{64} \times \text{FlexE Client MAC rate } \pm 100\text{ppm}$$

5.2.1.2 Idle insert/delete

All FlexE Clients must be rate-adapted to match the clock of the FlexE Group. This is accomplished by idle insertion/deletion according to [802.3] clause 82.2.3.6 and/or ordered set deletion according to [802.3] clause 82.2.3.9. The nominal rate of the adapted signal is slightly less than the nominal rate of the FlexE Client to allow room for the alignment markers on the PHYs of the FlexE Group and insertion of the FlexE overhead.

Note – Idle insertion/deletion is appropriate when the nominal rate of the FlexE Client is the same as the nominal rate of the calendar slots into which it is mapped, differing by no more than a few hundred ppm based on the presence of alignment markers, FlexE overhead, and pad blocks in addition to the ±100ppm possible variation between the clocks of different Ethernet physical interfaces. Idle insertion/deletion cannot be used to bridge large differences between the rate of a FlexE Client and the calendar slot capacity, e.g., to map a 10G FlexE Client into a 25G calendar slot. Such implementations would require a MAC frame buffer.

5.2.1.3 66B Block Distribution and Insertion into Calendar

The 66B blocks from each FlexE Client are distributed sequentially into the calendar in the order described in clause 6.7.

5.2.1.4 Calendar Distribution

The 66B blocks from calendar are distributed to each PHY of the FlexE Group according to the ordering describe in clause 6.7. The FlexE overhead is inserted into the sub-calendar of each PHY as described in clause 7.4.

5.2.1.5 PHY functions (Scramble, lane distribution, AM insertion, PMA, FEC, PMD)

5.2.1.5.1 100GBASE-R PHY functions (Scramble, lane distribution, AM insertion, PMA, FEC, PMD)

The stream of 66B blocks of each PHY is distributed to the PCS lanes of that PHY with insertion of alignment markers, and this is presented at the PMA service interface in the 100GBASE-R stack. Lower layers and interfaces of the 100GBASE-R Ethernet PHY (e.g., CAUI, FEC, PMA, PMD) are used as specified in [802.3].

5.2.1.5.2 200GBASE-R and 400GBASE-R PHY functions (Scramble, lane distribution, AM insertion, FEC, PMA, PMD)

The stream of 66B blocks of each PHY is presented at the input of the 257B transcoder as shown in [802.3bs] Figure 119-2, so it will be 257B transcoded, scrambled, FEC encoded, distributed to PCS lanes, and alignment markers inserted for presentation at the PMA service interface in the 200GBASE-R or 400GBASE-R stack. Lower layers and

interfaces of 200GBASE-R and 400GBASE-R PHYs (e.g., 200G/400GAUI, PMA, PMD) are used as specified in [802.3bs].

5.2.1.6 Error Control Block Generator

Error Control blocks are generated for insertion into calendar slots that are unused or unavailable. See Figure 23.

5.2.1.7 Control

The control function manages which calendar slots each FlexE Client is inserted into and inserts the FlexE overhead on each FlexE PHY in the transmit direction.

5.2.2 FlexE Demux Functions

The functions of the FlexE demux (the FlexE Shim in the receive direction) are illustrated in Figure 5 for FlexE Groups composed of 100GBASE-R PHYs, in Figure 6 for FlexE Groups composed of 200GBASE-R PHYs, and in Figure 7 for FlexE Groups composed of 400GBASE-R PHYs.

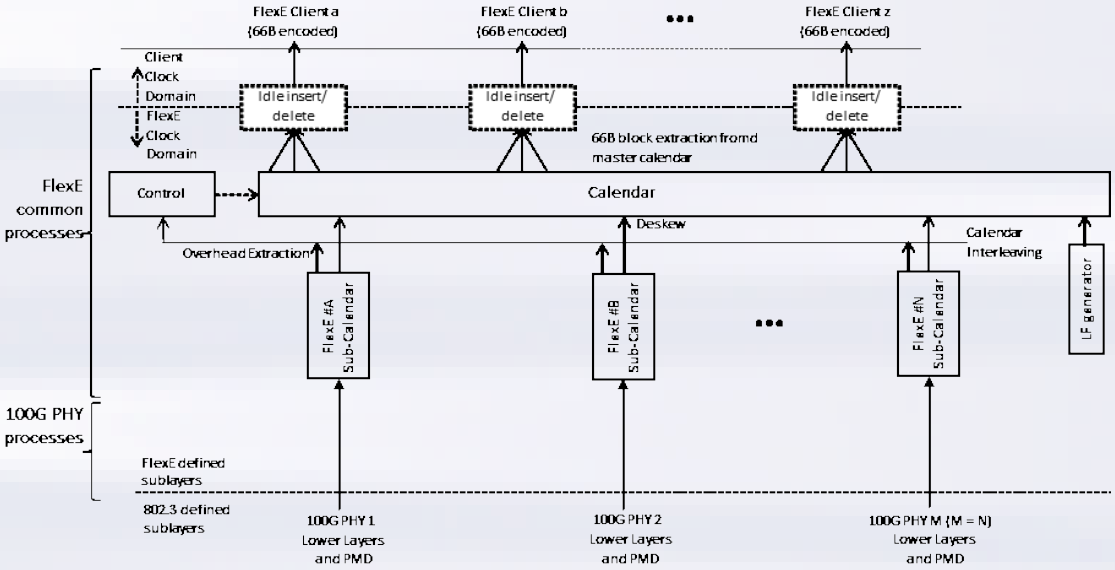


Figure 5: 100GBASE-R FlexE demux functions

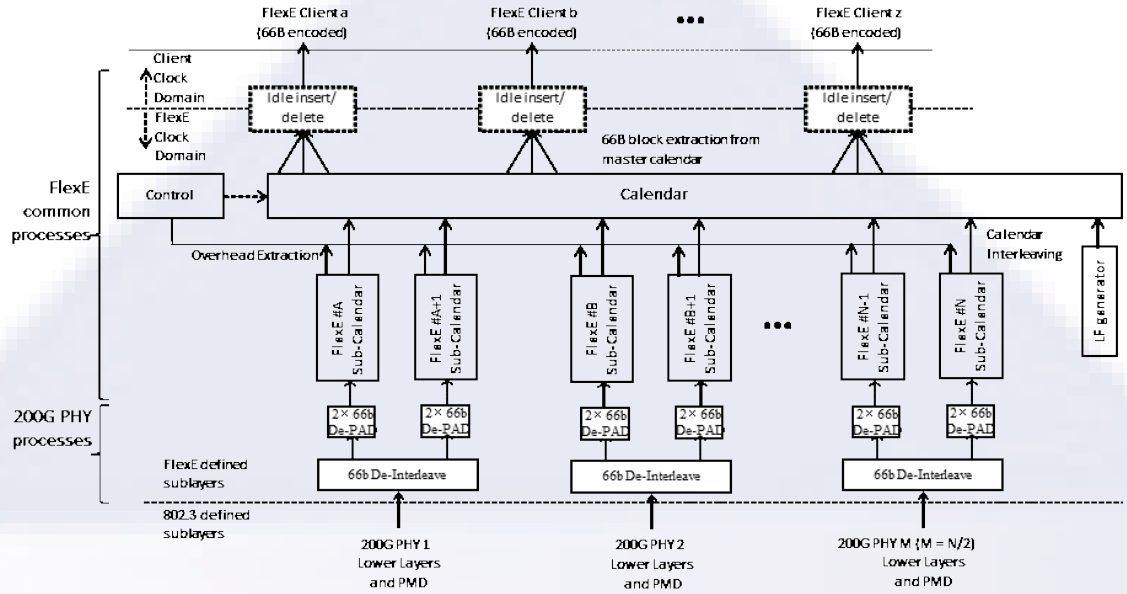


Figure 6: 200GBASE-R FlexE demux functions

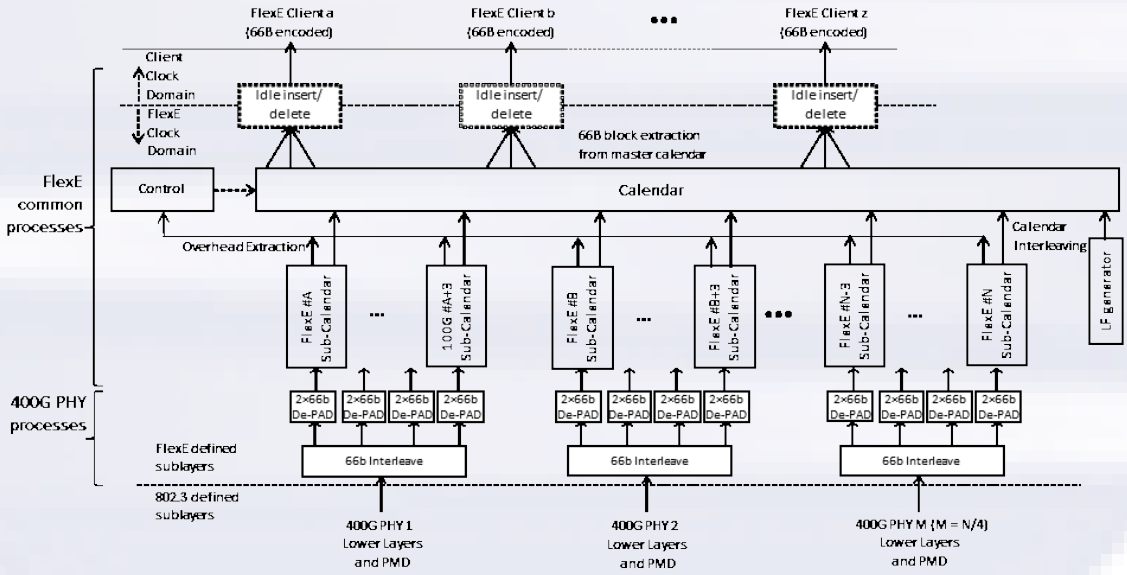


Figure 7: 400GBASE-R FlexE demux functions

5.2.2.1 PHY functions (PMD, PMA, FEC, lane deskew, interleave, AM removal, descramble)

5.2.2.1.1 100GBASE-R PHY functions (PMD, PMA, FEC, lane deskew, interleave, AM removal, descramble)

The layers of each 100GBASE-R PHY below the PCS are used exactly as specified in [802.3]. The PCS lanes are recovered, deskewed, reinterleaved, and the alignment markers are removed. The aggregate stream is descrambled. A 100GBASE-R PHY carries one 100G FlexE Instance.

5.2.2.1.2 200GBASE-R and 400G BASE-R PHY functions (PMD, PMA, lane deskew, interleave, FEC decode, AM removal, descramble, reverse transc0de)

The layers of each 200GBASE-R or 400GBASE-R PHY below the PCS are exactly as specified in [802.3bs]. The PCS lanes are recovered, deskewed, reordered, de-interleaved, FEC decoded, Post-FEC interleaved, the alignment markers are removed, the stream is descrambled, and reverse-transcoded to 66B format. A 200GBASE-R PHY can carry two 100G FlexE Instances, and a 400GBASE-R can carry four 100G FlexE Instances.

5.2.2.2 Calendar Interleaving and Overhead Extraction

The calendar slots of the sub-calendars on each 100G FlexE Instance are logically interleaved in the order specified in clause 6.8. The FlexE overhead is recovered from each 100G FlexE Instance.

5.2.2.3 LF Generator

In the case that any PHY of the FlexE Group has failed (PCS_Status=FALSE) or overhead frame lock or overhead multiframe lock has not been achieved on the overhead of any of the 100G FlexE Instances, LF is generated towards all FlexE Clients in the group.

5.2.2.4 66B Block Extraction from Calendar

The 66B blocks are extracted from the calendar positions assigned to each FlexE Client in the order described in clause 6.7.

5.2.2.5 Idle Insertion/Deletion

Idle insertion/deletion according to [802.3] clause 82.2.3.6 and/or ordered set deletion according to [802.3] clause 82.2.3.9 may be performed to rate-adapt the extracted 66B flow when necessary to adjust to the FlexE Client rate. Note that the nominal rate of the 66B flow carried in the calendar slots is slightly less than the nominal rate of the FlexE Client as the available space is reduced by the FlexE PHY PCS lane alignment markers the FlexE overhead, and 66B pad blocks where applicable.

5.2.2.6 Control

The control function manages which calendar slots each FlexE Client is extracted from each 100G FlexE Instance in the receive direction.

5.3 Sample Applications

FlexE can support a variety of applications. A non-exhaustive list includes:

- Router to Transport Connection (more examples below).
- Intra-Data Center “Fat Pipe” application: bonded PHYs for flows exceeding the PHY rate, or carrying traffic that doesn’t distribute efficiently with LAG.
- Generalized MLG applications, e.g., an n × 100G PHY as an umbilicus to a satellite shelf of lower rate ports.

One case of router to transport connection is where the transport network is unaware of FlexE. This case is illustrated in Figure 8. This may be used with legacy transport equipment that provides PCS-codeword transparent transport of 100GbE, 200GbE, or 400GbE, but provides no special support for FlexE.

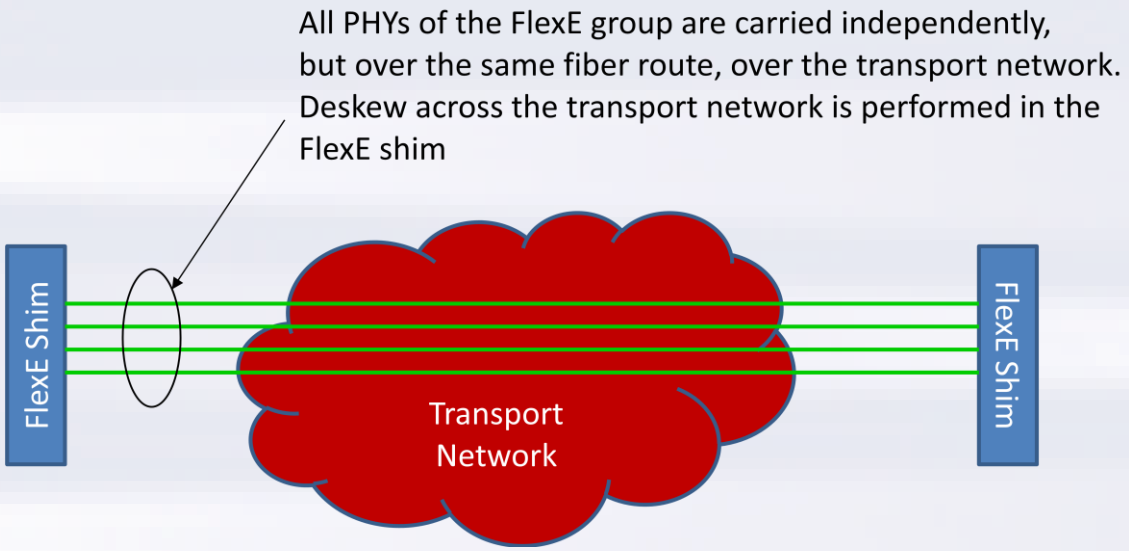


Figure 8: Router to FlexE unaware Transport Network connection

In the FlexE unaware case, the FlexE Shim, e.g., in a router, maps the FlexE Client(s) over a group of bonded Ethernet PHYs. Each of the Ethernet PHYs is carried independently over the transport network using a PCS codeword transparent mapping. The Ethernet PHYs are intended to be carried over the same fiber route: diverse routing is not envisioned. All of the PHYs of the FlexE Group need to be interconnected between the same two FlexE Shims. The FlexE Shim will need to tolerate and accommodate considerably more skew than if the FlexE Shims were only separated by an Ethernet link distance of 40km or less, as the transport network could carry the signal over thousands of kilometers. In Figure 8, it is the PHYs of the FlexE Group which are carried over the transport network.

Another case of router to transport connection is where the transport network equipment terminates the FlexE Group. This case is illustrated in Figure 9.

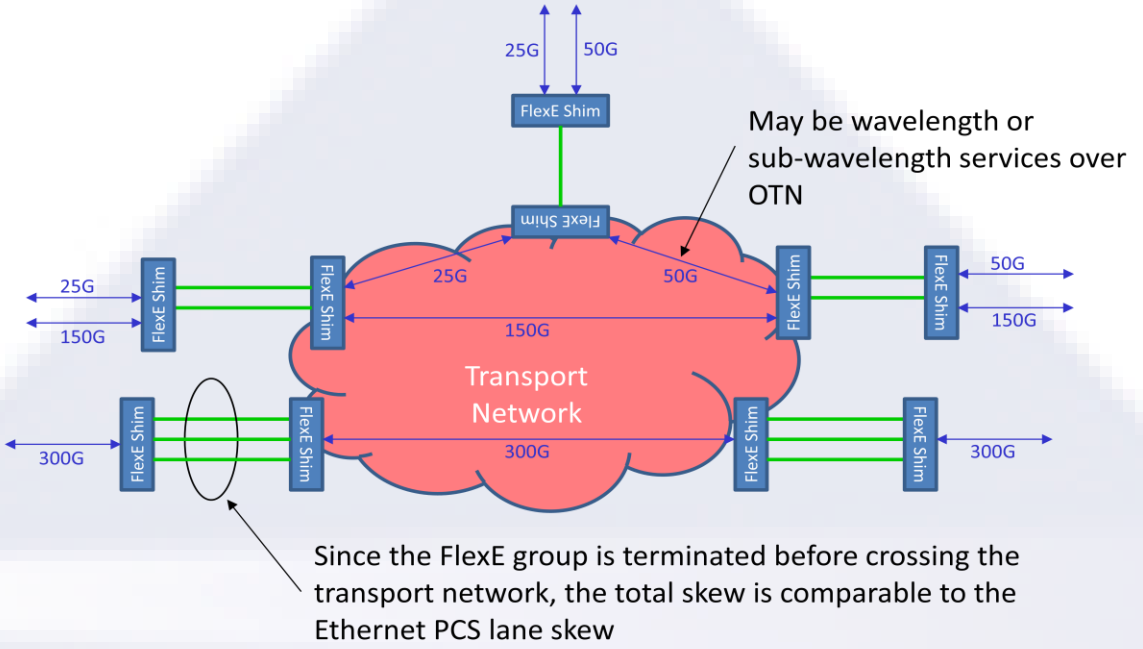


Figure 9: FlexE terminating transport network equipment

In the FlexE terminating case, the distance between any pair of FlexE Shims is limited to the Ethernet link distance (about 40km maximum), so the amount of skew that needs to be tolerated and compensated is considerably less. The other important distinction here is that it is the FlexE Clients, rather than the PHYs of the FlexE Group, which are carried over the transport network. The FlexE Client could be constructed to be the complete size of the payload that can be carried over a single wavelength (e.g., construct 200G to fill a DP-16QAM wavelength with the bonding of two 100GBASE-R PHYs), or could be a smaller client which is multiplexed and switched at a sub-wavelength level.

The final router to transport example described is one where the transport network is aware that it is carrying FlexE PHYs (as opposed to 100GbE), but the FlexE Group is not terminated on the transport equipment. This may be used to support cases where the Ethernet PHY rate is be greater than the wavelength rate, the wavelength rate is not an integral multiple of the PHY rate, or there is a reason (for example, wavelengths terminated on different transponder line cards) that it is not possible to terminate the FlexE Group in the transport equipment. This kind of example is illustrated in Figure 10.

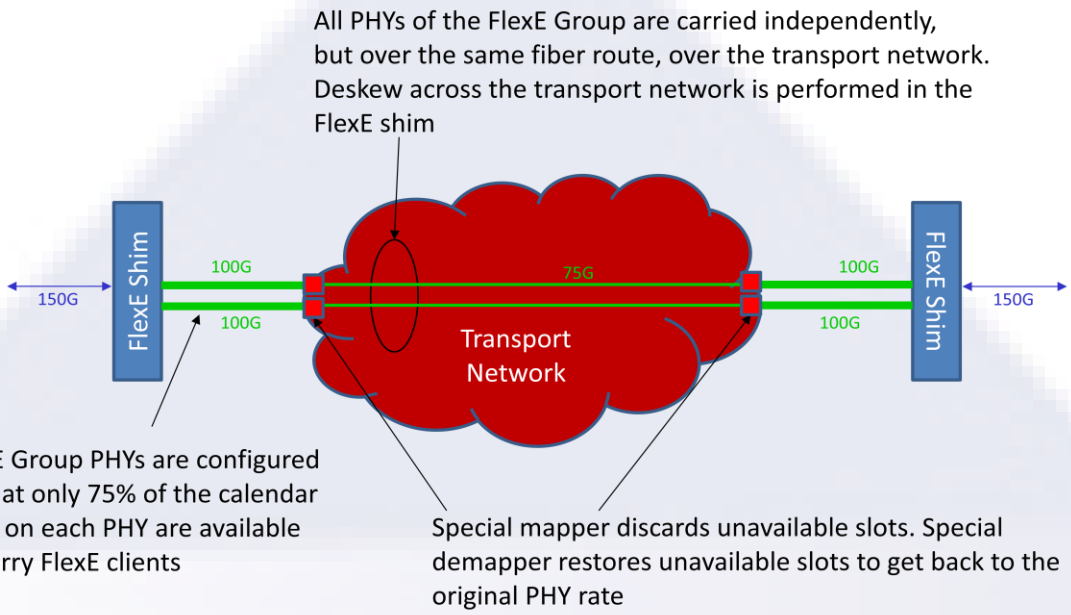


Figure 10: Example of FlexE aware transport of Ethernet PHYs of a FlexE Group

6 General Mechanism

6.1 FlexE Group

The FlexE Group is composed of from 1 to n 100G FlexE Instances which are carried over from 1 to m100GBASE-R, 200GBASE-R, or 400GBASE-R Ethernet PHYs. All PHYs in a group must operate at the same rate.

- For a group composed of 100GBASE-R PHYs, each PHY is identified by a number in the range [1-254]. The values of 0 and 255 are reserved. The FlexE PHY number and the 100G FlexE Instance number for 100GBASE-R PHYs are the same.
- For a group composed of 200GBASE-R PHYs, each PHY is identified by a number in the range [1-126]. The values 0 and 127 are reserved. The two 100G FlexE Instances that may be carried by a 200GBASE-R PHY are identified by an eight-bit 100G FlexE Instance number that consists of the PHY number in the upper seven bits, and 0 or 1 in the lower order bit.
- For a group composed of 400GBASE-R PHYs, each PHY is identified by a number in the range [1-62]. The values 0 and 63 are reserved. The four 100G FlexE Instances that may be carried by a 400GBASE-R PHY are identified by an eight-bit 100G FlexE Instance number that consists of the PHY number in the upper six bits, and 0, 1, 2, or 3 in the two lower-order bits.

A PHY number may correspond to the physical port ordering on equipment, but the FlexE Shim at each end of the group must identify each PHY in the group using the same PHY number, and each 100G FlexE Instance with the same 100G FlexE Instance number. PHY numbers within the group do not need to be contiguous, but 100G FlexE Instance numbers within the same PHY must be contiguous.

6.2 Groups composed of 100GBASE-R PHYs

Each 100GBASE-R PHY uses the bulk of the PCS functions described in [802.3] clause 82 including PCS lane distribution, lane marker insertion, alignment, and deskew. All

PHYs of the FlexE Group must use the same physical layer clock. Each PHY of the FlexE Group is able to deliver a logically serial stream of 64B/66B encoded blocks from the FlexE mux to the FlexE demux at a data rate of:

$$103.125\ \text{Gb/s} \times \frac{16383}{16384} \pm 100\text{ppm}$$

While the protocol supports a number of PHYs in the FlexE Group up to 254, practical implementations are likely limited to the range of 4-8 PHYs. The fraction applied to the base rate reflects the fact that 1/16K of the space of the interface is occupied by PCS lane alignment markers which are not space available to carry FlexE blocks. The FlexE blocks carried over each PHY of the FlexE Group has the format of a logically serial stream of (mostly) legal 64B/66B blocks with the format described in [802.3] Figure 82-5, although the blocks do not appear in a sequence that would make sense if interpreted as an Ethernet interface. The actual PHYs of the FlexE Group may transcode these blocks to 256B/257B format according to [802.3] clause 91.5.2.5 according to the PHY type, but they are trans-decoded back to 64B/66B blocks prior to delivery to the FlexE demux.

6.3 Groups composed of 200GBASE-R or 400GBASE-R PHYs

Each 200GBASE-R or 400GBASE-R PHY uses the bulk of the PCS functions described in [802.3bs] clause 119, including 257B transcoding, scrambling, alignment insertion, pre-FEC distribution, FEC encoding, and PCS lane distribution and interleaving.
Each 200GBASE-R PHY is able to deliver a logically serial stream of 66B encoded blocks from the FlexE mux to the FlexE demux at a data rate of:

$$206.25\ \text{Gb/s} \times \frac{20479}{20480} \pm 100\text{ppm}$$

Each 400GBASE-R PHY is able to deliver a logically serial stream of 66B encoded blocks from the FlexE mux to the FlexE demux at a data rate of:

$$412.5\ \text{Gb/s} \times \frac{20479}{20480} \pm 100\text{ppm}$$

The fraction applied to the base rate reflects that 1/20K of the space of the interface is occupied by PCS lane alignment markers which are not space available to carry FlexE blocks. The FlexE blocks carried over each PHY of the FlexE Group has the format of a logically serial stream of (mostly) legal 64B/66B blocks with the format described in [802.3] Figure 82-5 (through reference from [802.3bs] clause 119]), although the blocks do not appear in a sequence that would make sense if interpreted as an Ethernet interface. The actual PHYs of the FlexE Group will transcode these blocks to 256B/257B, scramble, and FEC encode for transmission over the physical interface, but they are trans-decoded back to 64B/66B blocks prior to delivery to the FlexE demux.

6.4 100G FlexE Instances, padding and interleaving

Each 100GBASE-R PHY carries a single 100G FlexE Instance in the 66B block positions between alignment markers.

Each 200GBASE-R PHY or 400GBASE-R PHY carries two or four 100G FlexE Instances and two or four sets of Pads, respectively. Pad blocks are used to compensate the difference between the 1/16K alignment marker spacing for 100GBASE-R PHYs and

the 1/20K alignment marker spacing for 200GBASE-R or 400GBASE-R PHYs so that the 100G FlexE Instances carried over all PHY types are the same nominal size. Two 66B pad blocks (P1, P2) are inserted per 163830 payload blocks at the same relative position on each 100G FlexE Instance.

The 100G FlexE Instances carried by a 200GBASE-R PHY or 400GBASE-R PHY are 66B block interleaved to fill the logical 66B block space between alignment markers on each respective PHY type (note that physically, the 66B blocks are 257B transcoded on these PHYs). The block interleaving is such that all 100G FlexE Instances on a PHY are fully aligned:

- On a 200GBASE-R PHY, the first overhead block of the overhead multiframe for 100G FlexE Instance xxxx_xxx0 is immediately followed by the first overhead block of the overhead multiframe for 100G FlexE Instance xxxx_xxx1. This results in a sequence of two consecutive FlexE overhead blocks followed by 2×20×1023 non-pad payload blocks followed by two consecutive FlexE overhead blocks, etc.
- On a 400GBASE-R PHY, the first overhead block of the overhead multiframe for 100G FlexE Instance xxxx_xx00 is immediately followed by the first overhead blocks of the overhead multiframe for 100G FlexE Instances xxxx_xx01, xxxx_xx10, and xxxx_xx11 in sequence. This results in a sequence of four consecutive FlexE overhead blocks followed by 4×20×1023 non-pad payload blocks followed by four consecutive FlexE overhead blocks, etc.

The alignment of the FlexE Overhead blocks of the interleaved 100G FlexE Instances on 200GBASE-R and 400GBASE-R PHYs, excluding the pad blocks is illustrated in Figure 11.

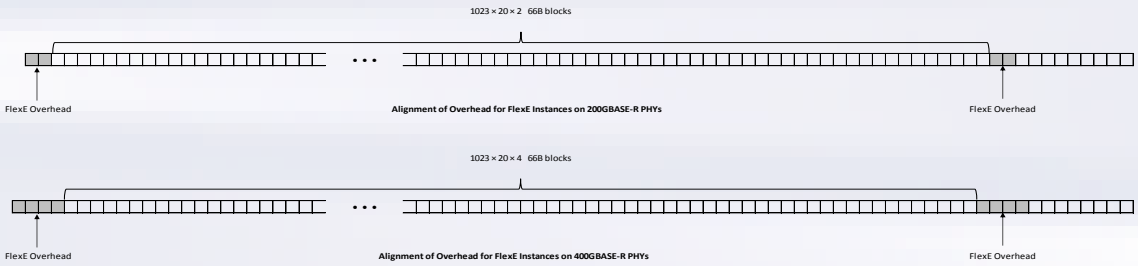


Figure 11 – Alignment of Overhead on Interleaved 100G FlexE Instances on 200GBASE-R and 400GBASE-R PHYs

Since the pad blocks are inserted at the same relative position on each 100G FlexE Instance, a consequence of this alignment is that the pad blocks will occur on 200GBASE-R PHYs in groups of four pad blocks followed by 2×163830 payload blocks followed by four pad blocks, etc., and on 400GBASE-R PHYs in groups of eight pad blocks followed by 4x163830 payload blocks followed by eight pad blocks, etc. While this implementation agreement is written in terms of pad block insertion and removal on a 100G FlexE Instance basis, implementations are free to insert or remove pad blocks on a PHY basis, or in a combined state machine with FlexE overhead insertion and removal, as long as the position of the pad blocks on the PHY is correct. Figure 12

illustrates the distribution of pad blocks on 200GBASE-R and 400GBASE-R in terms of 66B equivalent blocks between alignment markers.

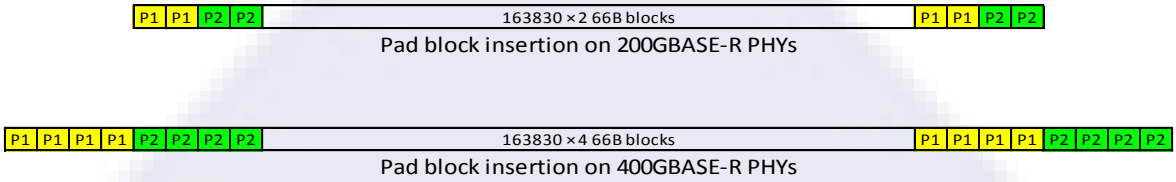


Figure 12 – Distribution of Pad blocks on 200GBASE-R and 400GBASE-R PHYs

The format of the P1 pad block is indicated in Figure 13. This uses the same ordered set "O" code as is used to identify the FlexE overhead, with the value 0xFFFFF in bits 12-31 (where the FlexE Group number would be for the first block of the FlexE overhead frame) distinguishing this as a pad block rather than a FlexE overhead block. Bits 8-11 of the P1 pad block are set to zero.

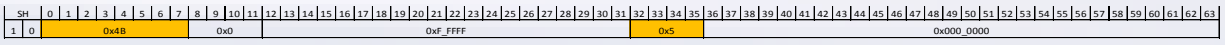


Figure 13 - Format of P1 Pad Block

The format of the P2 pad block is an Ethernet error control block as shown in Figure 23.

6.5 Unequipped 100G FlexE Instances

In certain cases, it may be desirable not to populate all 100G FlexE Instances on a 200GBASE-R or 400GBASE-R PHY. For example, in a FlexE aware configuration with 300G of transport network capacity with a single-member 400GBASE-R FlexE Group. This is functionally equivalent to having all calendar slots of the unequipped 100G FlexE Instance marked as unavailable, but avoids the necessity to process the overhead and manage the calendar for the unequipped 100G FlexE Instance(s).

The following rules govern unequipped instances:

- Unequipped instances must always be the highest numbered instance(s) on a PHY of the FlexE Group.
- There must always be at least one equipped 100G FlexE Instance on every PHY: note that some of the overhead such as management channels and the RPF bit are only carried in the first 100G FlexE Instance, which must always be present.

The format of an unequipped 100G FlexE Instance is as follows:

- The first block of the overhead frame is as shown in Figure 14. This is essentially the same as in any other overhead frame, except that the FlexE Group number is indicated as 0x00000, a reserved value to indicate that this instance doesn't belong to the group. Bits 8-11 of this block are set to zero.

Figure 14 – Format of First Block of FlexE overhead frame for unequipped 100G FlexE Instances

- All other non-pad blocks of the unequipped 100G FlexE Instance (including the remaining blocks of the overhead frame) are filled with Ethernet error control blocks as indicated in Figure 23. This ensures that no blocks from an unequipped 100G FlexE Instance can be accidentally interpreted as valid Ethernet data in the case of misconfiguration.
- Pad blocks are inserted on unequipped 100G FlexE Instances in the same relative position as for equipped instances as described in 6.4.

Unequipped 100G FlexE Instances are not indicated as being part of the FlexE map in describing the composition of the FlexE Group (see 7.3.3).

6.6 FlexE Client

The FlexE Client presented to the FlexE Shim is in the format described in clause 5.2.1.1.

All FlexE Clients to be transmitted over the same FlexE group are aligned to a common clock and rate-adapted to the available space in the FlexE calendar (whose nominal rate is slightly less than that of the FlexE Client due to space needed for the FlexE PHY PCS lane alignment markers and FlexE overhead) according to the process described in clause 5.2.1.2.. The rate-adapted FlexE Client operates at a rate of:

$$\text{FlexE Client MAC rate} \times \frac{66}{64} \times \frac{16383}{16384} \times \frac{1023 \times 20}{1023 \times 20 + 1} \pm 100\text{ppm}$$

This nominal rate is about 0.011% less than the nominal rate of the FlexE Client, which is well within what can be accomplished with idle insertion/deletion without packet loss. Note that this doesn't actually correspond to any clock that needs to be generated in an implementation, as the idle insertion deletion process will simply operate by filling the allocated block positions in the FlexE Group from a FlexE Client FIFO, inserting or deleting idles in the process of filling the block positions in the FlexE Group according to the calendar (see below).

6.7 FlexE Calendar

The FlexE mechanism operates using a calendar which assigns 66B block positions on sub-calendars on each 100G FlexE Instance of the FlexE Group to each of the FlexE Clients. The calendar is described with a granularity of 5G (although implementations may limit bandwidth assignment to coarser granularity, e.g., 25G), and has a length of twenty 5G slots per 100G of FlexE Group capacity. Two calendar configurations are supported: an "A" and a "B" calendar configuration. At any given time, one of the calendar configurations is used for mapping the FlexE Clients into the FlexE Group and demapping the FlexE Clients from the FlexE Group. The two calendar configurations are provided to facilitate reconfiguration. When a switch of calendar configurations adds or removes FlexE Clients from the FlexE Group, existing clients whose size and calendar

slot assignments are not changed by changing the calendar configuration are not affected.

For a FlexE Group composed of n 100G FlexE Instances, the logical length of the calendar is 20n. The blocks as allocated per the calendar are distributed to n sub-calendars of length 20 on each of the 100G FlexE Instances of the FlexE Group according to Figure 15.

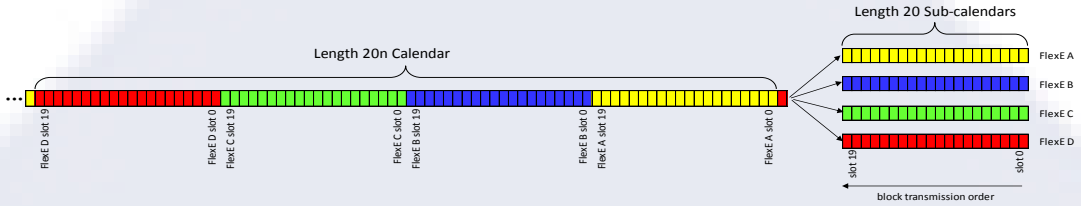


Figure 15: Illustration of FlexE Calendar Distribution based on 5G granularity

The order of distribution of twenty blocks at a time is selected over simple “round robin” distribution of 66B blocks. Calendar slots are identified by their 100G FlexE Instance number and the slot [0-19] (within that 100G FlexE Instance). The “logical” sequence number of a calendar slot is 20×the instance number plus the calendar slot number within the 100G FlexE Instance. The sequence is ascending order. Note that the sequence numbering is not necessarily consecutive when the assigned PHY numbers are not contiguous. This logical order only matters when calendar slots on different 100G FlexE Instances are assigned to the same FlexE Client.

Implementations may limit the bandwidth assignment granularity to 25G rather than 5G as illustrated in Figure 16. An effective 25G calendar slot occupies the same space as five consecutive 5G calendar slots 0-4, 5-9, 10-14 and 15-19 in a 100G FlexE Instance. The FlexE Client assignments in a 25G granularity calendar are indicated by constraining that groups of five consecutive 5G calendar slots must always carry the same FlexE Client.

Maintaining the description of the calendar at 5G granularity permits interoperability between implementations supporting 5G calendar slots and implementations supporting 25G calendar slots by requiring the 5G-capable implementation to always assign the same client to groups of five consecutive slots when connected to a 25G capable implementation.

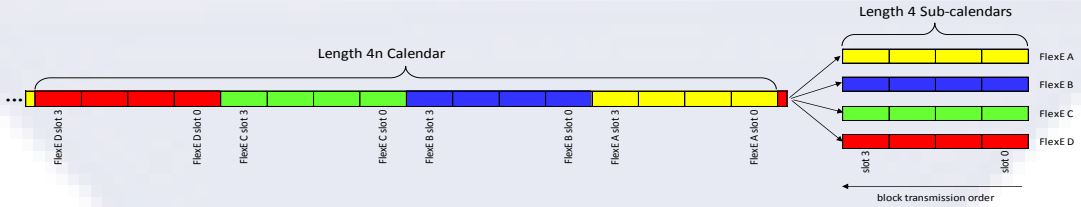


Figure 16 – Illustration of FlexE calendar distribution based on 25G granularity

6.8 FlexE Overhead and Alignment

The alignment of the data from the 100G FlexE Instances of the FlexE Group is accomplished by the insertion of FlexE overhead into the stream of 66B blocks carried over the group. The FlexE overhead is delineated by a 66B block which can be

recognized independently of the FlexE Client data. An illustration of the FlexE overhead on each 100G FlexE Instance of the FlexE Group is given in Figure 17.

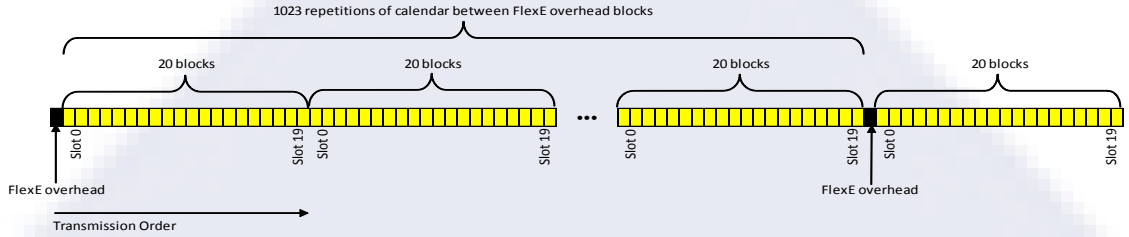


Figure 17: Illustration of insertion of FlexE overhead on each 100G FlexE Instance of a FlexE Group

On a 100G FlexE Instance, a FlexE overhead block will occur approximately once per 13.1µs. The actual format of the FlexE overhead blocks is such that they occur in a repeating sequence of eight blocks, so the sequence has a period of approximately 104.77µs. This sequence of overhead blocks is inserted in the same positions in the sub-calendar sequence on each 100G FlexE Instance and is used to align all of the 100G FlexE Instances of the FlexE Group at the FlexE demux to reconstruct the sequence in the order of the calendar so that the FlexE Clients can be recovered.

The FlexE aware transport network scenario illustrated in Figure 10 allows for marking a certain number of the calendar slots as unavailable. This is different from “unused”, in that it is known, due to transport network constraints, that not all of the calendar slots generated from the FlexE mux will reach the FlexE demux and therefore no FlexE Client should be assigned to those slots. The intention is that when a 100G FlexE Instance of the FlexE Group is carried across the transport network, the mapping is able to compress the signal to less than the 100G FlexE Instance rate by dropping the unavailable calendar slots. A case where 25% of the calendar slots are unavailable is illustrated in Figure 18. Unavailable slots are placed at the end of each relevant sub-calendar (the highest numbered slots).

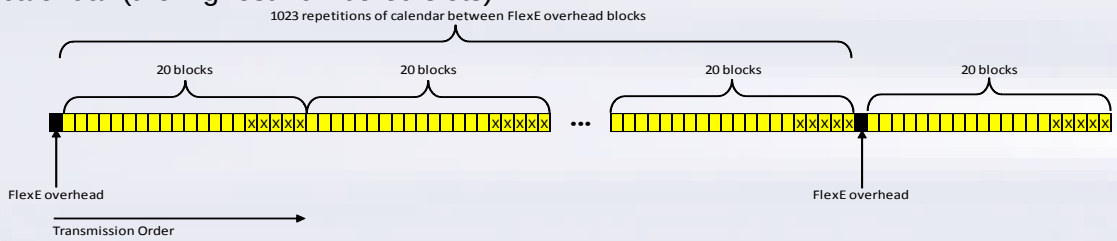


Figure 18: Illustration of Unavailable Calendar Slots to Facilitate Transport at lower rates

The anchor position of the FlexE overhead on each 100G FlexE Instance is encoded as an ordered set (control block type 0x4B). A distinct “O” code is used (0x5) which is different from that for the sequence ordered set used by Ethernet or the signal ordered set used by Fibre channel. The information to be transmitted in the FlexE overhead is encoded into the bytes D1, D2, and D3 of the ordered set block as indicated in Figure 19.

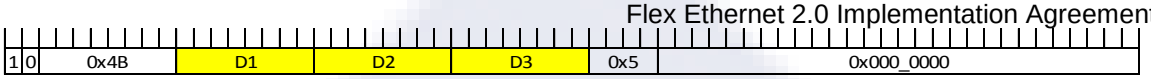
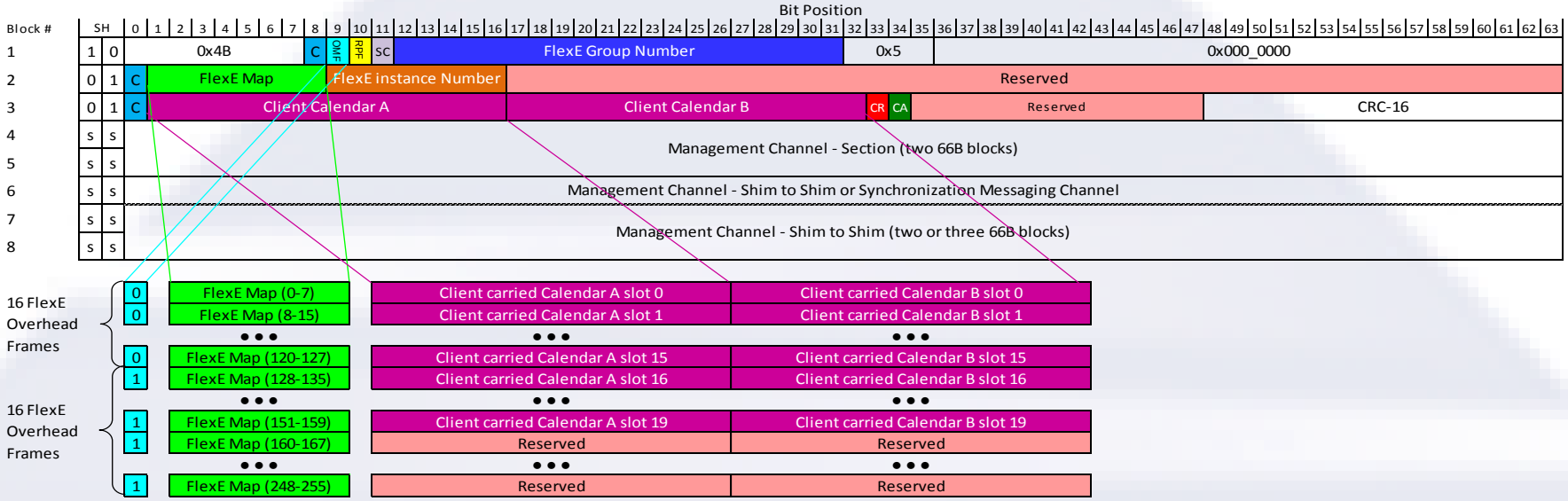


Figure 19: Encoding of Ordered Set block for FlexE overhead

The information which needs to be included in the overhead includes:

- Which 100G FlexE Instances are part of this FlexE Group
- The number identifying this 100G FlexE Instance within the FlexE Group. Note that multiple 100G FlexE Instances carried on a 200GBASE-R or 400GBASE-R each have a different 100G FlexE Instance number constructed using the PHY number, while for 100GBASE-R PHYs, the 100G FlexE Instance number and the PHY number are the same. See 6.1. (100G FlexE Instance numbers are unique within a FlexE Group. They are independent between FlexE Groups. Two 100G FlexE instances in different FlexE Groups may have the same 100G FlexE Instance number.)
- A way to transmit the programming of the sub-calendar configurations for each 100G FlexE Instance from the FlexE mux to the FlexE demux
- A way to indicate which calendar configurations (“A” or “B”) is in use at this time
- The FlexE Group number that this 100G FlexE instance is a member of (if necessary, for the case where the same 100G FlexE Instance number may be configured to be part of different FlexE Groups)
- Fields to support protocols necessary for changing the configuration of FlexE Clients into calendar slots.
- Two optional management channels. These may not be necessary in all applications (for example, if a network management system has direct access to the FlexE Shim at both ends of the connection), but may be useful for applications such as using FlexE for an n × 100G umbilicus to a remote shelf of lower-rate ports. One management channel (the 4^{th} and 5^{th} blocks of the FlexE overhead frame of the first 100G FlexE Instance on a given PHY) is available for communication across a section (for example, from a router with a FlexE Shim to FlexE aware transport equipment which does not terminate the FlexE Shim), and the other management channel (the 6^{th}-8^{th} or 7^{th}-8^{th} blocks of the FlexE overhead frame of the first 100G FlexE Instance on a given PHY) is used for communication between the FlexE Shims.
- An optional synchronization messaging channel. If configured, this is carried in the 6^{th} block of the FlexE overhead frame of the first 100G FlexE Instance on a given PHY.

The amount of information to be conveyed from the FlexE mux to the FlexE demux exceeds the 24 bits available in a single ordered set block per 100G FlexE Instance. This is addressed by spreading the relevant overhead across a sequence of eight FlexE overhead blocks on each 100G FlexE Instance, each separated by 20 × 1023 FlexE data blocks. This group of eight overhead blocks is referred to as the FlexE overhead frame. The FlexE overhead frame is illustrated in Figure 20. The meaning, interpretation and processing of this overhead is explained in clause 7.



C = Calendar configuration in use (see 7.3.2)
OMF = Overhead Multiframe Indicator (see 7.3.1)
C = Calendar configuration in use (see 7.3.2)
RPF = Remote PHY Fault (see 7.3.8)
SC = Synchronization Configuration - see 7.3.5
CR = Calendar Switch Request (see 7.3.4)
CA = Calendar Switch Acknowledge (see 7.3.4)
s s = Valid sync header bits (01 or 1 0)

FlexE Map = Control of which 100G FlexE Instances are members of this group (see 7.3.3)
FlexE instance Number = Identity of this 100G FlexE instance within the group (see 7.3.3)
FlexE Group Number - See 7.3.6
Client (being programmed) - See 7.3.4
Reserved - See 7.3.7
Management Channels - See 7.3.5
CRC-16 - See 7.3.8
Synchronization Messaging Channel - See 7.3.5

Figure 20: FlexE Overhead Frame and Multiframe of each 100G FlexE Instance

The first block of the FlexE overhead frame is encoded as an ordered set as shown in Figure 19. The next two FlexE overhead blocks are encoded as data 66B data blocks. The final five FlexE overhead blocks are reserved for the two optional management channels, and may carry any legal 66B block per [802.3] Figure 82-5 (excluding Ordered Sets with O code=0x5, as this is reserved for 100G FlexE Instance overhead).

The first block in the FlexE overhead frame serves as a marker to be used for alignment and re-interleaving of the sub-calendars from each of the 100G FlexE Instances of the FlexE Group at the FlexE demux. One FlexE overhead frame consisting of eight 66B overhead blocks is transmitted in approximately 104.77µs. Subject to the amount of buffer provided in a given implementation, skew detection and compensation across the 100G FlexE Instances of the FlexE Group can be compensated up to nearly half of this amount.

Test vectors for the overhead format are found in Appendix A.

7 Detailed Functions

The detailed processing to implement the functionality described in clause 6 is provided in this clause.

7.1 FlexE Group Functions

The FlexE Group is composed of from 1 to n 100G FlexE Instances which are carried over from 1 to m 100GBASE-R, 200GBASE-R, or 400GBASE-R PHYs.

The FlexE Shim provides to each FlexE Group PHY a set of 64B/66B encoded blocks that are encoded according to Figure 82-5.

7.2 FlexE Client Generation

The format and bit rate of FlexE Clients is described in clause 6.2. FlexE Clients generally originate from one of the following sources.

7.2.1 FlexE Clients Generated internally within a system

A FlexE Client may be generated internally within a system, for example from a Network Processing Unit (NPU) within a router. The packet flow is generated at the determined FlexE Client MAC rate and 64B/66B encoded according to [802.3] Figure 82-5.

7.2.2 FlexE Clients received from an Ethernet PHY

FlexE Clients at the rates of 10G, 25G, 40G, 100G, 200G, 400G, and in the future 50G(per existing P802.3by, P802.3bs, and P802.3cd projects) can be created from an Ethernet PHY at the corresponding rate with some processing to convert to the FlexE Client format and rate.

A 10GBASE-R signal will be converted to a 10G FlexE Client format before presenting to a FlexE mux by converting the coding from the 66B codeword set of [802.3] Figure 49-7 to the 66B codeword set of [802.3] Figure 82-5. This performs idle insertion/deletion in groups of four idles and or ordered set deletion where necessary to align the start control character to an 8-byte boundary (eliminating the use of control block types 0x2d, 0x66, and 0x33), converting control blocks that contain two ordered sets to a control block that

contains one ordered set (changing control block type 0x55 to 0x4b by deleting one of the two ordered sets encoded in the block), and by replacing the unnecessary idle control characters from the end of the Figure 49-7 control block type 0x4b with zeros to produce the Figure 82-5 control block format with control block type 0x4b. A 10G FlexE Client may be converted to a 10GBASE-R signal by using the idle insertion/deletion process as described in [802.3] clause 49.2.4.7 to adapt to the 10GBASE-R nominal rate, ensuring that at least four idles appear between packets. Idles may be inserted or deleted in groups of four, and/or ordered sets may be deleted according to [802.3] clause 49.2.4.10, resulting in the start of packet on a four-byte boundary rather than the 8-byte boundary. Ordered sets encoded with control block type 0x4b have the zeros at the end of the block replaced with four idle control characters. The result is that the blocks are encoded according to Figure 49-7.

A 25GBASE-R signal will be converted to a 25G FlexE Client format before presenting to a FlexE mux by converting the coding from the 66B codeword set of [802.3] Figure 49-7 to the 66B codeword set of [802.3] Figure 82-5 in a similar manner as described for 10GBASE-R signal above. A 25G FlexE Client may be converted to a 25GBASE-R signal by using the idle insertion/deletion process as described in [802.3] clause 49.2.4.7 to adapt to the 25GBASE-R nominal rate, ensuring that at least four idles appear between packets in a similar manner as described for 10G FlexE Client signal above.

A 40GBASE-R signal can be converted to a FlexE Client by serializing and deskewing the PCS lanes, removing the PCS lane alignment markers, and using the idle insertion/deletion process as described in [802.3] clause 82.2.3.6 and/or ordered set deletion as described in [802.3] clause 82.2.3.9 to adapt the signal to the 40G FlexE Client rate. A 40G FlexE Client coming from a FlexE demux is converted to a 40GBASE-R interface by using the idle insertion/deletion process as described in [802.3] clause 82.2.3.6 and/or ordered set deletion according to [802.3] clause 82.2.3.9, distributing the blocks round-robin to the four PCS lanes, and inserting PCS lane alignment markers.

A 100GBASE-R signal without FEC can be converted to and from a FlexE Client in the same manner as 40GBASE-R described above (except that the number of PCS lanes is 20 rather than 4). A 100GBASE-R signal with FEC, in converting to a FlexE Client, also will correct any errors per the FEC code, remove the FEC, and trans-decode from 256B/257B prior to the idle insertion/deletion process. To convert a 100G FlexE Client coming from a FlexE demux to a 100GBASE-R signal with FEC involves the same processes as for 40GBASE-R, but in addition, transcoding the signal to 256B/257B, inserting the FEC lane alignment markers, and adding the FEC.

A 200GBASE-R or 400GBASE-R signal with FEC, in converting to a 200G or 400G FlexE Client, will correct any errors per the FEC code, remove the FEC, and trans-decode from 256B/257B prior to the idle insertion/deletion process. To convert a 200G or 400G FlexE Client coming from a FlexE demux to a 200GBASE-R or 400GBASE-R signal with FEC involves the same processes as for 40GBASE-R, but in addition, transcoding the signal to 256B/257B, inserting the FEC lane alignment markers, and adding the FEC.

7.2.3 FlexE Clients from another FlexE Shim

In the case of equipment which terminates the FlexE Group, FlexE Clients can be delivered from the one FlexE Shim to another: for example, from a FlexE Shim at the

transport network ingress to another FlexE Shim at the transport network egress. The FlexE Client, a sequence of 64B/66B encoded blocks, is expected to be carried over the transport network without packet loss. As no timing information is carried by this stream, idle insertion/deletion and ordered set deletion are possible in the mapping over the transport network. The FlexE Shim at the network egress will only need to perform idle insertion/deletion according to [802.3] clause 82.2.3.6 and/or ordered set deletion according to [802.3] clause 82.2.3.9, not due to any expected change in the nominal bit-rate, but simply to align the clock with the FlexE Group clock.

7.2.4 Interconnect flexibility

Note that since the format of the FlexE Client is simply a logically serial stream of 66B blocks at a given rate, FlexE Clients do not need to be produced or received in the same manner at both ends of the connection. For example, a 10G, 40G or 100G FlexE Client might be generated as a system internal signal in the main chassis of a system, connected using an n × 100G FlexE umbilicus to a satellite shelf, and connected to physical 10GBASE-R, 25GBASE-R, 40GBASE-R, 100GBASE-R, 200GBASE-R, or 400GBASE-R ports on the satellite shelf. In the case where the FlexE mux is receiving a FlexE Client from a physical Ethernet port and the FlexE demux is delivering that FlexE Client to a physical Ethernet port, the two ports obviously have to be the same nominal rate, but they may not have be the same PHY type.

7.3 FlexE Overhead Processing

The format of the FlexE overhead is indicated in Figure 20.

7.3.1 FlexE Overhead Frame and Multiframe Lock

The FlexE overhead is encoded as 66B blocks and are inserted on each 100G FlexE Instance of the FlexE Group. One overhead block is inserted after every 1023 iterations of the length 20 sub-calendar of 66B FlexE data blocks, so the sequence is one block of overhead followed by 1023 × 20 blocks of data followed by one block of overhead.

FlexE overhead frame lock is achieved at the receiver (FlexE demux) on each PHY by recognizing the FlexE block 1 of the FlexE overhead frame, encoded as a special ordered set (the sync header is 10, the control block type is 0x4B (ordered set), and the “O” code is 0x5), and then finding the FlexE ordered set block again (1023 × 20 + 1) × 8 block positions later. Once FlexE overhead frame lock is achieved, the next expected FlexE overhead block will be 1023 × 20 + 1 block positions later. While in FlexE overhead frame lock, bytes D1-D3 of the ordered set block, plus the 66Bblocks occurring at 20461, 40922, 61383, 81844, 102305, 122766, and 143227 blocks beyond the ordered set block will be interpreted as FlexE overhead frame. FlexE overhead is not interpreted if not in FlexE overhead lock. FlexE overhead lock will be lost if the sync header, control block type, or O code do not match at the expected position for 5 occurrences.

Certain information is transmitted in every FlexE overhead frame. Other information is distributed across a sequence of 32 FlexE overhead frames, referred to as the FlexE overhead multiframe. The OMF (overhead multiframe) bit has a value of “0” for the first sixteen overhead frames of the overhead multiframe, and a value of “1” for the last sixteen overhead frames of the overhead multiframe, as shown in Figure 20. The FlexE demux achieves overhead multiframe lock on each 100G FlexE Instance when the OMF bit transitions from a “0” to a “1” or a “1” to a “0” in consecutive overhead frames with

good CRC. There are two opportunities in the overhead multframe for the FlexE demux to achieve overhead multiframe lock.

7.3.2 Calendar Configuration in Use

There are two calendar configurations for each 100G FlexE Instance of the FlexE Group: the “A” calendar configuration (encoded as 0) and the “B” calendar configuration (encoded as one). The two calendars are used to facilitate reconfiguration. Clients can be added or removed from the FlexE Group without affecting the traffic on other clients, assuming the operating clients are carried in the same calendar slot locations. While clients can be resized or moved to different calendar slots through calendar updates, there is no assurance that the resizing or client moves can be “hitless” in all network scenarios. Normally, changes are only made to the calendar which is not currently in use.

Exceptions would include initial link configuration or replacement of a failed circuit pack where it is necessary to download the calendar information into the replacement pack. The data flow through the FlexE mux/demux pair is indeterminate during any changes to the active calendar configuration, until the assignment for all slots in the active calendar configuration have stabilized and been received by the FlexE demux in FlexE overhead frames with good CRC.

The calendar configuration in use is signaled from the FlexE mux to the FlexE demux on each 100G FlexE Instance in the three bit positions labeled “C” in Figure 20. While most of the FlexE overhead can be reliably protected by the CRC, the calendar configuration in use must be interpreted even if the CRC is bad, since the FlexE demux must be able to switch its calendar in use at precisely the same overhead frame boundary as the FlexE mux. So that this can be done reliably, three copies of the calendar configuration in use are transmitted, and interpreted by the receiver by majority vote. Since the three copies are separated into different FlexE overhead blocks across the overhead frame (the first and second copy are separated by 1,350,415 bits and the second and third copies are separated by 1,350,425 bits), the different copies will never be affected by the same burst error. Since each 100GBASE-R interface has a BER of 10^{-12} or better, and 200GBASE-R and 400GBASE-R interfaces have a BER of 10^{-13} or better, the probability of two C bits indicating the calendar in use being wrong is no more than 10^{-24}, which can safely be ignored.

When the calendar configuration in use changes from a 0 to a 1, or from a 1 to a 0, the calendar configuration used by both the FlexE mux and the FlexE demux will be changed beginning from the first data block following first block of the next FlexE overhead frame on each 100G FlexE Instance.

7.3.3 FlexE Map and 100G FlexE Instance Number

The set of 100G FlexE Instances in the FlexE Group (not necessarily consecutive 100G FlexE Instance numbers) are indicated in the “FlexE Map” field of the FlexE overhead. This is distributed as eight bits per overhead frame in each of the thirty-two overhead frames in the overhead multiframe (256 bits total), with each bit set to a one indicating a 100G FlexE Instance number that is a member of the group, and all other bits of the FlexE map set to zero. The FlexE Map values are only accepted from overhead frames with good CRC. The full FlexE map is sent on all 100G FlexE Instances of the FlexE

Group so that it is possible for the FlexE demux to verify that the same 100G FlexE Instance numbers are configured at the FlexE mux as at the FlexE demux, and can tell whether all expected 100G FlexE Instances are being received.

The 100G FlexE Instance number in the FlexE Group (from 1 to n) is encoded in the second block of the FlexE overhead frame. Note that this is persistent information which does not change while the group is in service. The receiver accepts a value for “100G FlexE Instance Number” when the same value is received in two consecutive overhead frames with good CRC. Updates to the respective group of eight bits of the FlexE map bit map are accepted from overhead frames with good CRC.

7.3.4 Calendar Configuration

The contents of both the A and B calendar configurations are transmitted continuously from the FlexE mux to the FlexE demux, with the contents of one calendar slot of the A and B sub-calendars for each 100G FlexE Instance being transmitted in the first twenty overhead frames of the FlexE overhead multiframe. The client fields are ignored by the FlexE demux when not in overhead multiframe lock since the FlexE demux would not know which slot in which calendar that client belongs to.

Note that implementations supporting 25G calendar slots signal the contents of the A/B calendars by indicating the same FlexE Client assignment in groups of the five 5G calendar slot positions that correspond to a given 25G calendar slot.

The sub-calendar configurations on each 100G FlexE Instance are transmitted by sending the clients assigned to each calendar slot in the same order as the corresponding 66B payload block positions occur in the transmission sequence on that 100G FlexE Instance.

The Client fields indicate which of the FlexE Clients is mapped into a given calendar slot in the A and B calendar configurations for the sub-calendar carried over that 100G FlexE Instance. The size of a given FlexE Client can be calculated based on the number of calendar slots that client is assigned to (i.e., how many calendar slots have the same numeric value in the Client field across the entire FlexE Group). The Clients are indicated by 16-bit fields transmitted in the 3rd block of the FlexE overhead frame. The value 0x0000 indicates a calendar slot which is unused (but available). The value 0xFFFF (all ones) indicates a calendar slot that is unavailable, for the case indicated in Figure 10 where the full FlexE Group PHY rate cannot be carried over the transport network in the FlexE aware transport use case. Any value other than 0x0000 or 0xFFFF may be used to designate a particular FlexE Client carried by the group.

The Client fields are ignored in overhead frames with a bad CRC, leaving previous assignments to the clients in the relevant slot unchanged.

The full contents of both calendar configurations are transmitted from the FlexE mux to the FlexE demux approximately once every 3.35ms. The fact that the calendar configurations are transmitted continuously avoids any inconsistency between the calendars at the FlexE mux and the FlexE demux due to a lost message.

The normal process of reconfiguration (e.g., adding or removing FlexE Clients to or from the FlexE Group) will involve programming the new or modified FlexE Client

assignments into the calendar configuration which is not in use, then switching to the updated calendar configuration, and finally updating the original calendar configuration.

The switch from one active calendar configuration to another can be coordinated between the FlexE mux and the FlexE demux using the Calendar Request (CR) bit sent from the FlexE mux to the FlexE demux, and the Calendar Acknowledge (CA) bit sent from the FlexE demux to the FlexE mux. Normally, the CR bit has the same value as the active calendar configuration being sent from the FlexE mux to the FlexE demux, and the CA bit has the same value as the calendar configuration currently being used by the FlexE demux.

When the FlexE mux has completed the programming of the offline calendar configuration and is ready to switch, it informs the FlexE demux by changing the CR bit to the value of the offline calendar configuration on all 100G FlexE Instances of the FlexE Group beginning with the same overhead frame in the overhead positions on each 100G FlexE Instance.

When the FlexE demux is prepared to accept the switch of calendar configuration, it informs the FlexE mux by changing its CA bit to match the incoming CR bit on all 100G FlexE Instances of the FlexE Group beginning with the same overhead frame in the overhead positions on each 100G FlexE Instance. When this occurs is application specific. At the earliest, it occurs once the assignment of every calendar slot in the offline configuration has been received by the FlexE demux in a FlexE overhead frame with a good CRC after the change of the incoming CR bit. But the CA bit indication may be delayed for a variety of reasons: for example, software may need to be prepared for the incoming bandwidth change, for example in SDN applications.

The FlexE mux normally will switch calendar configurations only after receiving the CA bit acknowledgment after telling the FlexE demux it is ready to switch. The FlexE mux should set a timer (suggested default value as 1 second) after changing the outgoing CR bit. Appropriate values for the timer and action to be taken if the timer expires prior to receiving the CA bit are application specific. The timer could be as short as about 15ms (allowing for three complete transmissions of the calendar), but may be longer, for example, if software on the far end must be prepared to accept the switch to the updated calendar. The action to be taken on timer expiry without receiving the CA bit response is either to proceed with the switch or to raise an alarm and wait for corrective action to be taken. The FlexE mux indicates to the FlexE demux the change of calendar by changing the value of all three “C” bits to indicate the new calendar in the same FlexE overhead frame on all 100G FlexE Instances.

Note that the availability of the above information in the protocol is not intended to limit the ways in which FlexE can be used. The FlexE demux may not act as a “slave” of the FlexE mux in terms of calendar configurations in every application. For example:

- A static configuration (e.g., one composed of a fixed number of 100G FlexE Instances and PHYs, perhaps performing simple bonding for a single client, or supporting only a fixed calendar configuration for something like a port expander) would not need to fully implement this protocol. Such a configuration would simply transmit the A and B calendar configurations as fixed, always indicate the A calendar configuration as the calendar configuration in use, and would alarm the configuration inconsistency if the received calendar configuration from the far

end is not the values expected or if the far end attempts to switch calendars (e.g., sends the calendar in use bits or the CR bit indicating calendar B rather than calendar configuration A).

- An application where a management system or SDN controller has access to the FlexE mux/demux at each end of the FlexE Group, that controller may configure the FlexE mux and demux configurations directly and instruct the two ends when to switch calendars. The information sent inband over the 100G FlexE Instances within the PHYs might just be used as a check for the consistency of the configuration rather than as control for how the FlexE demux is configured.

7.3.5 Management Channel(s) and Synchronization Messaging Channel

Certain applications may require use of management channel(s). Two optional management channels and an optional Synchronization Messaging Channel are provided on each PHY of the FlexE Group:

- A “section” management channel is carried from a FlexE Shim to the adjacent FlexE aware node (which may be the far end FlexE Shim in a simple router to router connection, or a FlexE aware transport network interface in the case of transport network does not terminate the FlexE Group) in the first 100G FlexE instance on a PHY.
- A “shim to shim” management channel which is carried end-to-end between the shims that terminate the FlexE Group in the first 100G FlexE instance on a PHY.
- A optional synchronization messaging channel in the FlexE overhead of the first 100G FlexE Instance of the FlexE Group, which is applicable only for simple point-to-point bonding scenarios, e.g., creating a 300GbE equivalent by bonding three 100GBASE-R PHYs. they will need to be carried in the FlexE overhead.

For 200GBASE-R and 400GBASE-R PHYs, the management channels and (if applicable) synchronization messaging channel are carried in the overhead of the first 100G FlexE Instance within the PHY. Block positions 4-8 in the FlexE overhead of the remaining 100G FlexE Instances are reserved, encoded with a data sync header (01), and zeros in the rest of the block as for any other reserved bits.

When a management channel is not used, it is transmitted as an Ethernet idle control blocks as illustrated in Figure 21.

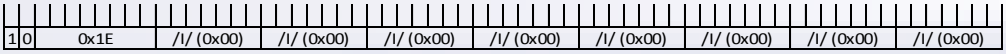


Figure 21: Ethernet Idle Control Block

The format of the management channel is application specific. The section management channel occupies two 66B blocks (blocks 4 and 5) of each FlexE overhead frame. The total capacity of the section management channel is approximately 1.222 Mb/s (not counting the sync headers), or 1.260 Mb/s (counting the sync headers).

The shim to shim management channel may occupy two or three 66B blocks of each FlexE overhead frame, depending on configuration.

For groups not configured to provide a synchronization messaging channel, the shim-to-shim management channel occupies blocks 6-8 of the FlexE overhead frame. The total

capacity of the shim to shim management channel for this configuration is approximately 1.833 Mb/s (not counting the sync headers), or 1.890 Mb/s (counting the sync headers). For groups configured to provide a synchronization messaging channel, the shim-to-shim management channel occupies blocks 7-8 of the FlexE overhead frame. The total capacity of the shim-to-shim management channel in this configuration is approximately 1.222 Mb/s (not counting the sync headers), or 1.260 Mb/s (counting the sync headers).

When a group is provisioned to support a synchronization messaging channel, block position 6 of the FlexE overhead frame (on a 100GBASE-R PHY, or the first 100G FlexE Instance on a 200GBASE-R or 400GBASE-R PHY) is used to carry PTP and/or SSM messages as specified in [G.8264] or [1588]. These messages are 66B block encoded as per [802.3] clause 82. The interface timestamp point for a PTP event message transported over the synchronization messaging channel shall be the start of the FlexE overhead multiframe preceding the first 66B block (the /S/ block) of the PTP event message.

Whether a group is configured to support a synchronization messaging channel is indicated in the Synchronization Configuration (SC) bit in the 1st block of the FlexE overhead frame in the first 100G FlexE Instance. If the SC bit has the value 0, the shim-to-shim management channel occupies blocks 6-8 of the FlexE overhead frame on a in the first 100G FlexE Instance. If the SC bit has the value 1, the shim-to-shim management channel occupies only blocks 7-8 of the FlexE overhead frame in the first 100G FlexE Instance, and block 6 is allocated to the synchronization messaging channel. This is expected to be a static configuration aspect, and a synchronization messaging channel cannot be added or removed while the group is in service. A misconfiguration alarm can be raised if the SC bit persistently (e.g., 3 consecutive occurrences in FlexE overhead frames with good CRC) does not match the expected value for the configuration. The SC bit position in all but the first 100G FlexE Instance on a 200GBASE-R or 400GBASE-R PHY is reserved.

The only constraint on the management channel is that every 66B block is a legal format according to [802.3] clause 82. The protocol used over a management channel may be Ethernet based, using a combination of data and control blocks, or may be any other application-specific format using only data blocks.

A possible application is to carry the LLDP protocol [802.1AB] in the shim-to-shim management channel for verification of the connectivity of FlexE PHYs between FlexE Shims, or in the section management channel to verify connectivity of PHYs between a router and transport equipment in a FlexE aware network configuration. The LLDPDUs are 66B block encoded per [802.3] clause 82.2.3.
LLDP is a simple, one-way protocol that allows an LLDP agent on the near end system to transmit interface identification (and certain capability) information to the far-end system. Each end asynchronously informs the far end of the interface identification (chassis and port ID) of the near-end interface. When received, each end stores the information received in the LLDPDU in the Remote Systems MIB, where software can verify that the physical connectivity is as intended. Information in the remote systems MIB si removed (forgotten) if not refreshed by another LLDPDU within the TTL interval. When used over the shim-to-shim or section management channels, the following shall apply:

- Transmission of LLDPDUs over the shim-to-shim or section management channels of a particular FlexE PHY does not require that all PHYs of the FlexE Group are in service or connected.
- As there is no bridge in the middle of either the shim-to-shim or section management channels, all LLDPDUs shall use the “nearest bridge” destination MAC address 01-80-C2-00-00-0E.
- Each LLDPDU contains all of the mandatory TLVs of the LLDP protocol, specifically the Chassis ID, Port ID, and TTL. Use of any optional TLVs is application specific.
- When the link is brought up for each FlexE PHY, that PHY will send LLDPDUs over the respective management channel(s) for the default txFast iterations of four LLDPDUs at one-second intervals.
- Subsequently, each FlexE PHY will send LLDPDUs at the default TxInterval rate of one per 30 seconds, with the default TTL being 4×TxInterval+1=121 seconds.
- Note that there is no need to include any new TLVs to describe the FlexE Group structure (e.g., PHY numbers and PHY map) as this information is fully contained in the FlexE overhead information.

Each PHY of the FlexE Group can carry its own management channels in the first 100G FlexE Instance on that PHY. The management channels are not aggregated across the FlexE Group.

7.3.6 FlexE Group Number

A 20-bit FlexE Group number is available to allow checking that the correct 100G FlexE Instance is part of the correct FlexE Group.

The FlexE Group number is normally provisioned to the same value in both directions. All equipped 100G FlexE Instances on all PHYs of the FlexE Group must have the same FlexE Group number. It is not possible to have different 100G FlexE Instances on the same PHY be members of different FlexE Groups.

The FlexE Group number is selected from the range 1-0xFFFFE. The values of 0x00000 and 0xFFFFF may not be used to designate a FlexE Group. The received group number is checked against the provisioned group number and any mismatch will be alarmed to indicate the misconnection.

7.3.7 Reserved Bits

The reserved bits in the FlexE overhead frame are reserved for possible future extensions to this implementation agreement. The reserved bits shall be transmitted as zero before scrambling. An implementation of this version of the IA should ignore these bits on receipt and leave the responsibility to an implementation of a newer version of the implementation agreement to recognize receipt of zeros as an indication of interconnection with an older version, and presumably the newer version knows whether it is interoperable with the older version.

Note that one of the FlexE overhead bits was a reserved bit in FlexE 1.0 and has a specific use in FlexE 2.0. The value “0” for this bit is compatible with the FlexE 1.0 mode of operation.

7.3.8 Remote PHY Fault (RPF)

This is used to inform the far-end shim of a locally detected failure of the PHY. Since there is no 100G, 200G, or 400G RS layer per PHY, the FlexE overhead is used to covey this information. For 200GBASE-R and 400GBASE-R PHYs, the RPF bit is carried only in the FlexE overhead for the first 100G FlexE Instance on each PHY, and the RPF bit position is a reserved bit in the other 100G FlexE Instances. See clause 7.5.2.

7.3.9 CRC-16

Primarily to avoid corrupting the content of the calendar configurations in the presence of bit errors, the FlexE overhead is protected by a CRC. The CRC is calculated over the following bits across the first three blocks of the FlexE overhead frame (in the order transmitted and received, not the order described):

- The D1, D2, and D3 bytes of the ordered set in overhead block 1.
- All eight octets after the sync header of overhead block 2.
- The first six octets after the sync header of overhead block 3.

The CRC is calculated using the polynomial $x^{16} + x^{12} + x^5 + 1$ with an initialization value of zero, where x^{16} corresponds to the MSB and x^0 corresponds to the LSB. This value is inserted by the FlexE mux into the transmitted overhead with the bit corresponding to x^{15} in the position transmitted first and the bit corresponding to x^0 transmitted last. Note that while this is the opposite of normal Ethernet bit-transmission order, it is consistent with the order of transmission of the Ethernet FCS. It is calculated by the FlexE demux over the same set of bits and compared to the received value. Various overhead described in the previous clauses is either accepted or ignored based on whether the CRC matches the expected value.

7.4 FlexE Mux Data Flow

The FlexE Mux creates a logically serial stream of 66B blocks by interleaving FlexE Clients according to a calendar of length 20n slots for a FlexE Group composed of n 100G FlexE Instances. Each slot corresponds to 5G of bandwidth. A FlexE Client is assigned a number of slots according to its bandwidth divided by 5G. The calendar configuration is distributed as described earlier in Figure 15.

Figure 22 presents an example of insertion of different bandwidth FlexE Clients into a logical calendar. The slots assigned to a particular FlexE Client do not all need to be on the same 100G FlexE Instance of the FlexE Group, and new clients can be added as long as there are sufficient slots available.

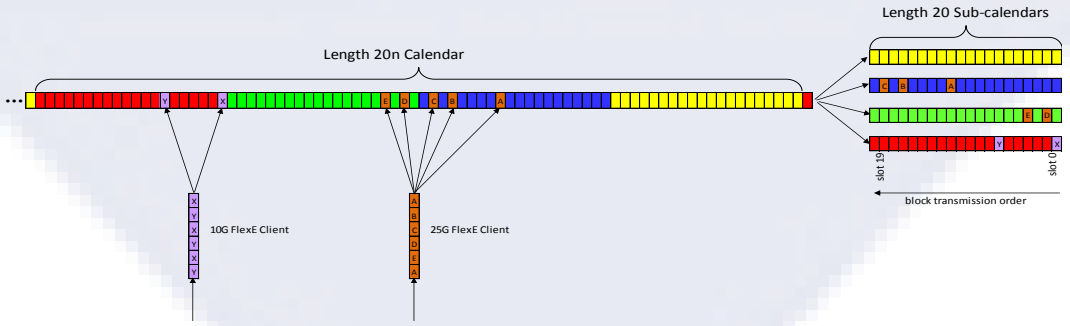


Figure 22: Illustration of Data Flow for FlexE Mux (5G calendar slots)

Any slot in the calendar configuration which is either “unused” or “unavailable” will be filled with Ethernet Error control blocks with the format given in Figure 23. This ensures that any error in calendar slot assignment cannot appear to the FlexE demux as valid FlexE Client data.

1	0	0x1E	/E/ (0x1E)	/E/ (0x1E)	/E/ (0x1E)	/E/ (0x1E)	/E/ (0x1E)	/E/ (0x1E)	/E/ (0x1E)	/E/ (0x1E)

Figure 23: Ethernet Error Control Block Format

These rules allow for creation of the complete data sequence on each 100G FlexE Instance of the FlexE Group, which are then padded and interleaved if necessary (see 6.4) to form the data sequence that is carried over each PHY. The FlexE overhead as described in clause 7.3 is inserted onto each 100G FlexE Instance after every 1023 repetitions of the sub-calendar sequence in the same relative position to the calendar sequence on every 100G FlexE Instance. This provides a marker which allows the data from the different 100G FlexE Instances of the FlexE Group to be re-interleaved in the original sequence so that the FlexE Clients can be extracted. The 66B block stream is then converted into the format for the individual FlexE Group PHY, which includes block distribution and alignment marker insertion, along with (if applicable) 256B/257B transcoding and FEC calculation and insertion.

In an implementation that restricts bandwidth allocation to 25G calendar granularity, the FlexE mux may interleave clients according to a calendar of 4n slots for a group composed of n 100G FlexE Instances. Each slot corresponds to 25G of bandwidth. A FlexE Client is assigned a number of slots according to its bandwidth divided by 25G. A 25G calendar slot occupies five consecutive 66B blocks within a 100G FlexE Instance, which do not remain adjacent when multiple 100G FlexE Instances are 66B block interleaved on the PHY. An illustration of 25G calendar slot allocation is provided in Figure 24. Appendix B illustrates how the 66B blocks from 25G calendar slots distribute when 100G FlexE Instances are interleaved for 200G or 400G PHYs.

As with 5G calendar slots, 25G calendar slots that are unused or unavailable are filled with (in groups of five) error control blocks as indicated in Figure 23. This ensures that any errors in calendar slot assignment cannot be interpreted by the FlexE demux as valid client data.

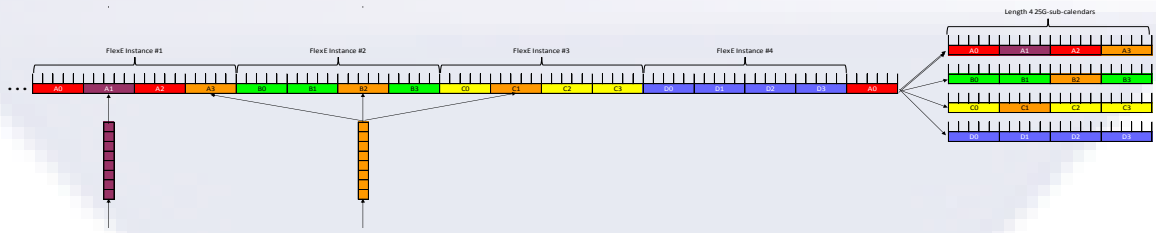


Figure 24: Illustration of Data Flow for FlexE Mux (25G calendar slots)

7.5 FlexE Demux Data Flow

The FlexE Demux operates on a sequence of 66B blocks received from each 100G FlexE Instance of the FlexE Group. Recovering this sequence of blocks includes (if applicable), FEC error correction and FEC remove and trans-decoding to 64B/66B, PCS or FEC lane alignment, re-interleaving, and alignment marker removal. If necessary, pad

blocks are removed and multiple 100G FlexE Instances present on the PHY are dis-interleaved (see 6.4). Once this has occurred, the 100G FlexE Instances of the FlexE Group are re-interleaved so that FlexE Clients can be recovered as illustrated in Figure 25.

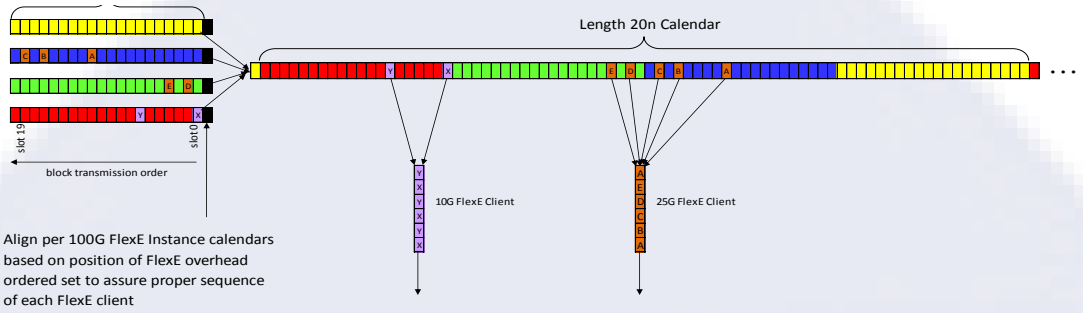


Figure 25: Illustration of FlexE Demux Data Flow (5G calendar slots)

Note that the FlexE overhead frame repeats on a cycle of approximately 104.77μs, which allows measuring skew differences between 100G FlexE Instances recovered from the PHYs of the FlexE Group of approximately ±52μs.

In an implementation that restricts bandwidth allocation to 25G calendar granularity, the FlexE demux may extract clients from the recovered 66B block stream according to a calendar of 4n slots for a group composed of n 100G FlexE Instances. Each slot corresponds to 25G of bandwidth. A 25G calendar slot occupies five consecutive 66B blocks within a 100G FlexE Instance, which are not adjacent on a PHY when multiple 100G FlexE Instances are 66B block interleaved. An illustration of FlexE Clients extracted based on 25G calendar slot allocation is provided in Figure 26.

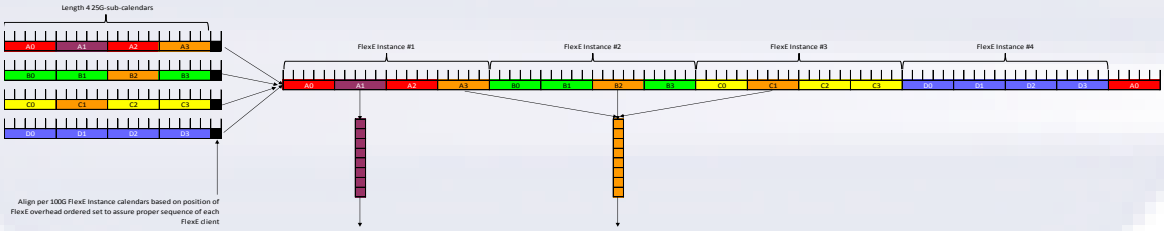


Figure 26: Illustration of Data Flow for FlexE Demux (25G calendar slots)

7.5.1 Skew Tolerance Requirements

The amount of skew to be expected between the PHYs of the FlexE Group are application specific. Note that skew does not accumulate between 100G FlexE Instances interleaved onto the same PHY, so deskew will be performed across 100G FlexE Instances from different PHYs that have skew between them. This implementation agreement specifies skew requirements for two classes of applications.

Low Skew Applications include intra-data-center applications, plus those transport network applications where the FlexE Shim is implemented in the transport equipment and the FlexE Clients rather than the PHYs of the FlexE Group are carried across the transport network. The skew tolerance requirement for low skew applications is 300ns. Note that the intra-PCS-lane skew tolerance requirement for 100GBASE-R, 200GBASE-R, and 400GBASE-R is 180ns. A larger skew budget is established for FlexE applications of similar reach to account for the fact that the PCS lane deskew is not synchronized across the PHYs of the FlexE Group, and there may be other variation, such as cable length, or even heterogeneous Ethernet PHY types which are not present within a single Ethernet interface.

High Skew Applications include the broadest range of transport network applications where the PHYs of the FlexE Group rather than the FlexE Clients are carried over the transport network (FlexE aware/unaware transport). The skew tolerance for high skew applications may need to be as high as 10µs. This is established to account for about 6µs of dispersion-related skew if the PHYs are mapped over lambdas at opposite ends of the “C” band over large distances (e.g., trans-Pacific), with extra margin for things like split-band amplifiers and patch cords or the processing time to crunch and uncrunch the signal in the case where not all of the calendar slots can be carried over the transport network connection.

7.5.2 FlexE Demux Fault Handling

If the inter-PHY skew (detected when attempting to deskew 100G FlexE Instances on different PHYs) exceeds the skew tolerance of the implementation, the FlexE Clients will not be demapped from the incoming PHYs, but will be sent continuous Ethernet Local Fault Ordered sets as illustrated in Figure 27 at the FlexE Client rate.

If one or more of the PHYs of the FlexE Group has failed (e.g., loss of signal, failure to achieve block lock or alignment lock, hi BER, or any other condition that results in PCS_status=FALSE), the Remote PHY fault bit is set to one in the reverse direction on that PHY in the first 100G FlexE Instance carried on that PHY. Note that this must be done in the FlexE overhead since there is no normal 100GBASE-R RS layer available to indicate remote fault to the far end. In addition, when one or more of the PHYs of the FlexE Group have failed, if one or more of the 100G FlexE Instances fails to achieve overhead frame lock or overhead multiframe lock, if there is inconsistency among the FlexE maps, 100G FlexE Instance numbers, or FlexE Group numbers received on the different 100G FlexE Instances of the group, or if the skew between 100G FlexE instances from different PHYs exceeds the deskew buffer provided by the implementation, an appropriate alarm should be raised and all of the FlexE Clients will be sent continuous Ethernet Local Fault Ordered sets as illustrated in Figure 27 at the FlexE Client rate.

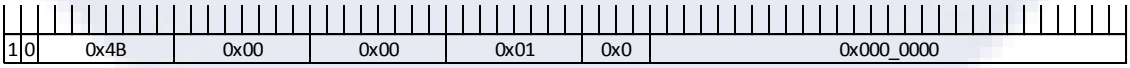


Figure 27: Ethernet Local Fault Ordered Set

7.6 FlexE Group Configuration

When a new FlexE Group is brought into service, the initial configuration must be provisioned from both ends, and the initial configuration must be the same. The group is configured to consist of from 1 to n 100G FlexE Instances carried over from 1 to m PHYs

of the same rate (100GBASE-R, 200GBASE-R, or 400GBASE-R). All of the PHYs which are configured as part of the group are brought into service, all transmitting the stream of 66B blocks including the FlexE overhead of the 100G FlexE Instances carried over that PHY. See 6.4 concerning the manner in which 100G FlexE Instances are carried over PHYs. The respective bits of the "FlexE Map" field are set to indicate which 100G FlexE Instances (and by extension, which PHYs) the near end has configured as part of the group. The group is brought into service when all incoming PHYs are up, receiving FlexE overhead on each of the expected 100G FlexE Instances within the allowable skew tolerance of each other, and the received 100G FlexE Instances all indicate the expected 100G FlexE Instances as being part of the group. When the FlexE Group is first configured and brought into service, all of the calendar slots are normally set to "unused" or "unavailable" (as needed for partial-rate transport configurations), although there may be certain implementations such as a static configuration supporting simple bonding which are brought up with a single FlexE Client filling all available calendar slots.

7.7 Energy Efficient Ethernet (EEE)

EEE is not supported on the PHYs of a FlexE Group, as there is no good way to verify that every FlexE Client is idle and put the entire group into a low power idle mode.

The "Fast Wake" mode of EEE can be supported for a FlexE Client. The LPI control characters used during Fast Wake can simply pass through the 66B payload block positions allocated to that client by the calendar configuration, allowing an implementation to be aware when data is not arriving on a given FlexE Client.

For FlexE Clients generated internally to the system or created from an optical Ethernet PHY, "Fast Wake" is the only mode of operation possible, and negotiation between the FlexE Client endpoints (above the FlexE Client MAC) to confirm the use of "Fast Wake" will occur via LLDP. For a FlexE Client generated from a copper Ethernet PHY, the AN capabilities should always indicate "No" for both "Deep Sleep" and "Fast Wake", and the "Fast Wake" mode can then be enabled via LLDP negotiation.

8 Transport Network Mappings for Flex Ethernet Signals

Three different methods of mapping of FlexE signals over transport networks are possible.

8.1 FlexE Unaware Transport

The case of FlexE unaware transport involves the transport network mapping each of the Ethernet PHYs independently over the transport network using the existing PCS codeword transparent mapping. Since the FlexE mux and FlexE demux are separated by transport network distances, this requires a "high skew" implementation of the FlexE Shim as described in clause 7.5.1.

Note that certain existing OTN mappers/demappers are not fully PCS codeword transparent with respect to LF and RF ordered sets, and may mistake an LF or RF sent for an individual FlexE Client as an LF or RF for the entire link and bring the link down.

This is not an issue in the case of FlexE unaware transport of simple bonding to carry a larger rate flow, as failure of the single high-rate FlexE Client is equivalent to a failure of the group. But it may be an issue if FlexE unaware transport is used to carry a group of multiple lower-rate FlexE Client services using less than fully PCS codeword transparent mappings.

8.2 FlexE termination in the Transport

The next case is where the FlexE Shim is terminated by the transport network equipment, and rather than carrying the PHYs of the FlexE Group over the transport network, the FlexE Clients are carried over the transport network. The rate-adapted FlexE Client bit-rate is given in clause 6.6.

Note that since the stream of blocks presented to the transport network from the FlexE Shim does not have any timing information, the transport network is not required to transport the signal at the exact adapted FlexE Client bit-rate: idle insertion/deletion or padding may be used in the mapping specified by ITU-T if it provides a more convenient rate for the transport.

When a FlexE Client is mapped in this manner, it may be connected, at the OTN egress, to another FlexE Shim where it will be clock aligned with the FlexE Group at the network egress. It may also be connected to an Ethernet PHY with the same nominal MAC rate as the FlexE Client using the appropriate conversion as described in clause 7.2.2.

Note that in the case where it is necessary to connect a FlexE Client to an Ethernet PHY across an OTN where the Ethernet PHY uses a legacy mapper/demapper, it may be necessary to perform the conversion of the FlexE Client to the Ethernet PHY format according to clause 7.2.2 immediately after the FlexE Shim and to map the FlexE Client over OTN as if it were an Ethernet PHY of the corresponding rate.

This second case can use a “low skew” implementation of the FlexE Shim as described in clause 7.5.1.

8.3 FlexE Aware Transport

The third case is where the transport network equipment is aware it is carrying 100G FlexE Instances of a FlexE Group, but does not terminate the FlexE Group in the transport network equipment.

The transport network equipment terminates the section management channel on the first 100G FlexE Instance on each of the FlexE PHYs, extracting (but not forwarding) this channel at the network ingress and inserting this channel at the network egress. At the network ingress, any content of the section management channel is replaced with Ethernet idle control blocks (see Figure 21) into the transport network.

This may be used to support cases where the Ethernet PHY rate is be greater than the wavelength rate, the wavelength rate is not an integral multiple of the PHY rate, or there is a reason (for example, wavelengths terminated on different transponder line cards) that it is not possible to terminate the FlexE Shim in the transport equipment. In cases where the Ethernet rate is greater than the wavelength rate or is not an integral multiple of the wavelength rate, the transport network equipment may “crunch” a 100G FlexE

Instance of the FlexE Group by allowing bits or bytes to be discarded from the unavailable calendar slots at the transport network ingress and these bits or bytes re-inserted with fixed values at the transport network egress. The mapping of this requires serializing and deskewing the PCS lanes of the PHY, recovering the 100G FlexE Instance(s) from the PHY, then discarding from the "UNAVAILABLE" calendar slots to reduce the bit-rate. For example, if only 15 of 20 calendar slots are available in a sub-calendar for a given 100G FlexE Instance, there are effectively 1023 repetitions of a length 15 calendar after discarding the unavailable slots included in the information that must be transported. At the transport network egress, the bits or bytes removed from the unavailable slots are restored to the 100G FlexE Instance stream of 66B blocks so that error control blocks occur in every unavailable slot as illustrated in Figure 23. The net bit-rate of this reduced-rate flow (the information that must be transported) when there are "n" available 5G calendar slots on a given 100G FlexE Instance is:

$$103.125\ \text{Gb/s} \times \frac{16383}{16384} \times \frac{1+1023n}{20461} \pm 100\text{ppm}$$

For a 100G FlexE Instance where bandwidth is allocated in increments of 25G, the bit-rate is:

$$103.125\ \text{Gb/s} \times \frac{16383}{16384} \times \frac{1+5115n}{20461} \pm 100\text{ppm}$$

It is expected that the granularity of partial-rate transport is 25Gb/s. Note that what is specified in this IA is only the information to be carried over the OTN, and how it is mapped is over transport networks is specified by ITU-T Q11/15.

The rates of the information that must be transported of the 64B/66B flows for a given number of available calendar slots to be carried over the transport network are given in Table 1.

Table 1: 64B/66B Rates given number of Available Calendar Slots on a 100G FlexE Instance

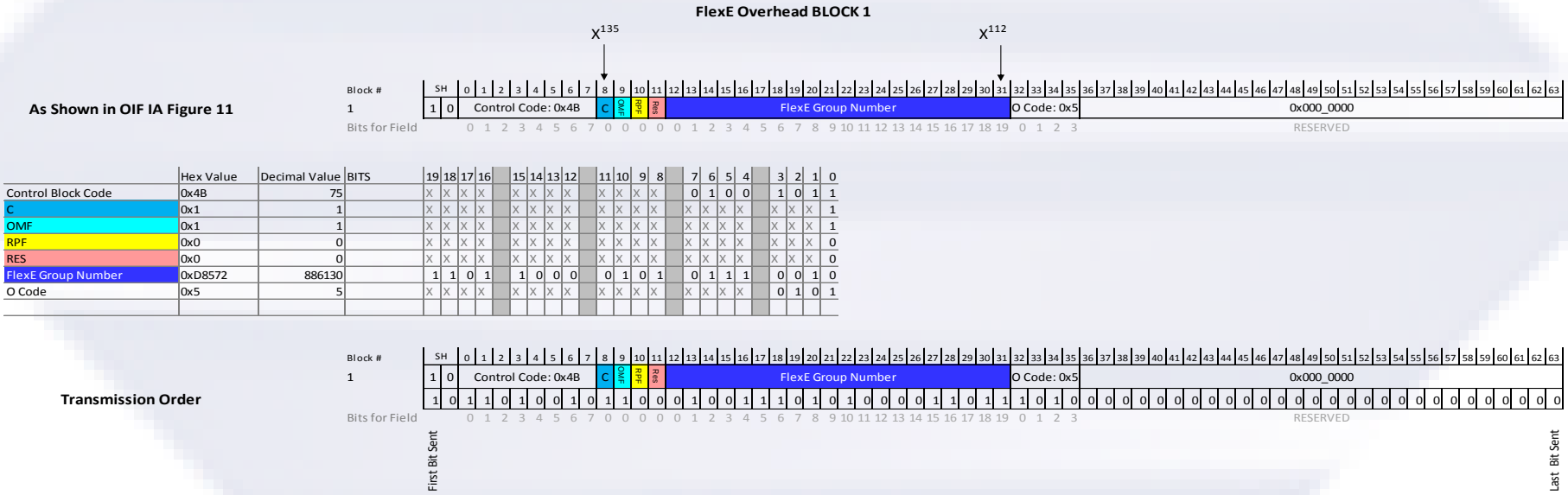
FlexE Instance Available 5G Slots	64B/66B Flow rate ±100ppm
5	25.78345626
10	51.56187276
15	77.34028925
20	103.1187057

As described in clause 6.8, unavailable slots are always at the end of the sub-calendar configuration for the respective 100G FlexE Instance. The partially-filled 100G FlexE Instance should normally be the last equipped 100G FlexE Instance on the last (highest numbered) PHY of the FlexE Group. It is expected that since the rate of a wavelength

isn't expected to change in-service, when a partial-rate signal is carried over the OTN, the mapper is statically configured to drop bits or bytes from a certain number of calendar slots at the ingress and to restore those same bits or bytes to the calendar slots at the egress to contain Ethernet error control blocks as described in clause 7.4 in the payload block positions for each unavailable calendar slot. The OTN mapper is not expected to dynamically react to which slots are marked as unavailable in the calendar configurations, but may non-intrusively monitor the FlexE overhead and detect as an error condition if a calendar slot the mapper has been configured to drop indicates that it is carrying FlexE Client data rather than being marked as unavailable.

9 Appendix A: Test Vectors

Test vectors for the first three overhead blocks of the FlexE overhead frame are given in Figure 28, Figure 29 and Figure 30. Note that the last five overhead blocks of the FlexE overhead frame form the two management channels and/or synchronization messaging channel. The positions of the 136 bits as coefficients of the polynomial across which the CRC is calculated is indicated. For example, the C bits in overhead blocks 1, 2, and 3 correspond to the coefficients of x^{135}, x^{111}, and x^{47} as shown in Figure 28, Figure 29, and Figure 30 respectively.



	Hex Value	Decimal Value	BITS	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Control Block Code	0x4B	75		X	X	X	X	X	X	X	X	X	X	X	X	0	1	0	0	1	0	1	1
C	0x1	1		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1
OMF	0x1	1		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1
RPF	0x0	0		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0
RES	0x0	0		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0
FlexE Group Number	0xD8572	886130		1	1	0	1	1	0	0	0	0	1	0	1	0	1	1	1	0	0	1	0
O Code	0x5	5		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	1	0	1

Figure 28: Test Vector for first block of FlexE overhead frame

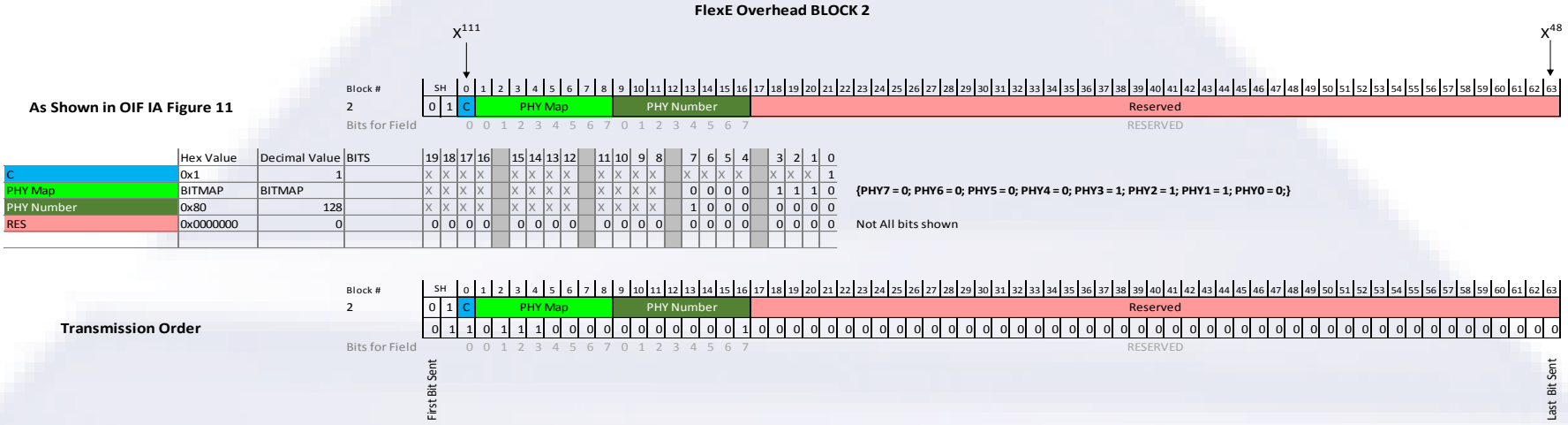


Figure 29: Test Vector for second block of FlexE overhead frame

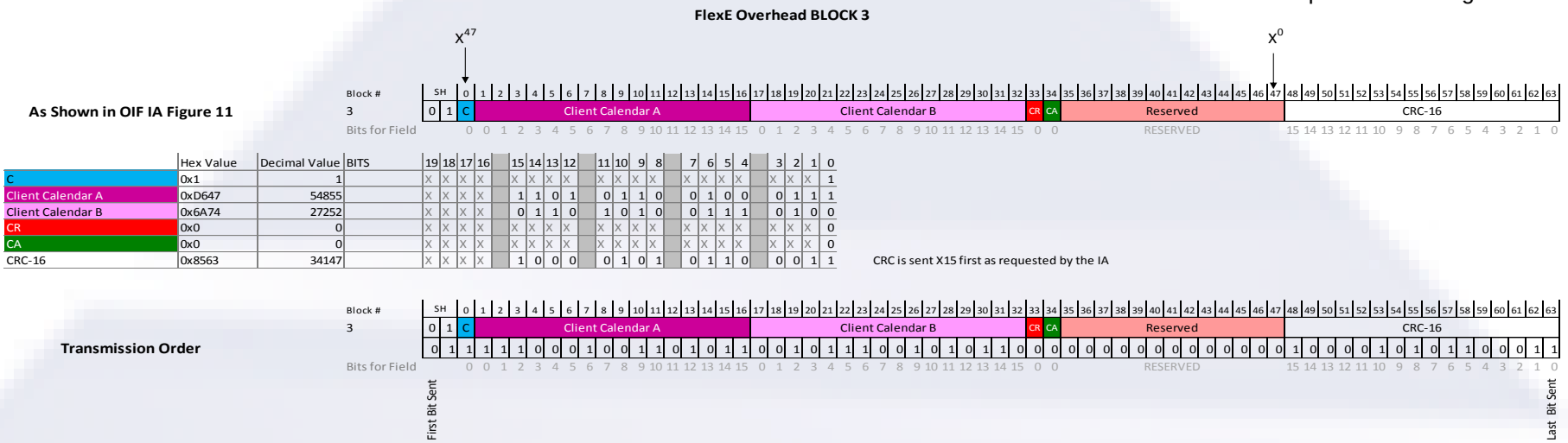


Figure 30: Test Vector for third block of FlexE overhead frame

10 Appendix B: (Informative) Illustration of 25G Calendar Slot Distribution across 200G or 400G PHYs

Figure 31 and Figure 32 illustrate the distribution of 25G Calendar slots across 200G and 400G PHYs respectively at the FlexE mux. Figure 33 and Figure 34 illustrate the corresponding behavior at the FlexE demux. It is important to note that while a 25G calendar slot is composed of 5 adjacent 66B block positions within a logical 100G FlexE Instance, these block positions do not remain adjacent when multiple 100G FlexE Instances are interleaved on a FlexE PHY.

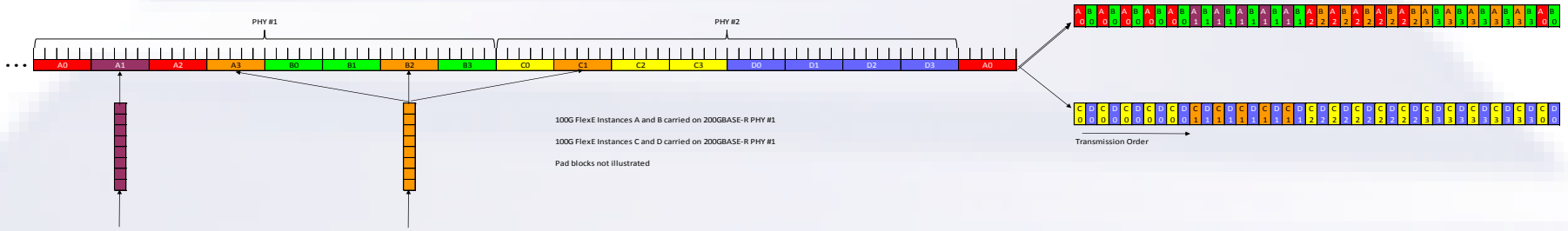


Figure 31: Example of 25G Calendar Slots Multiplexed onto 200G FlexE PHYs

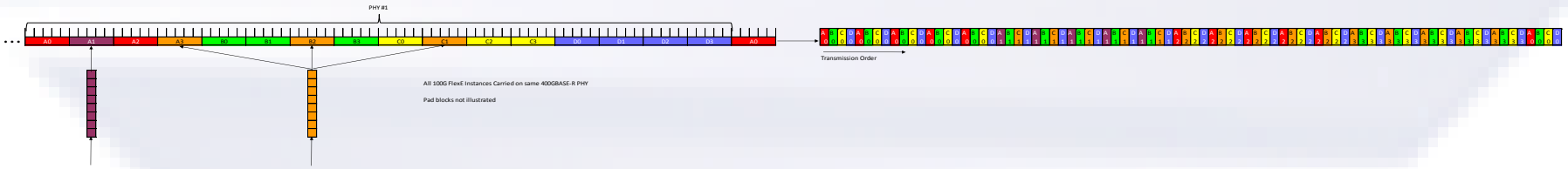


Figure 32: Example of 25G Calendar Slots Multiplexed onto a 400G FlexE PHY

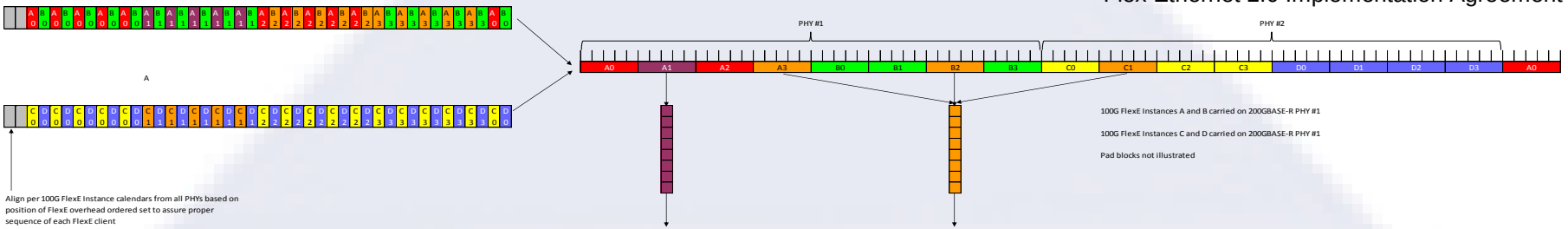


Figure 33: Example of 25G Calendar Slots Demultiplexed from 200G FlexE PHYs

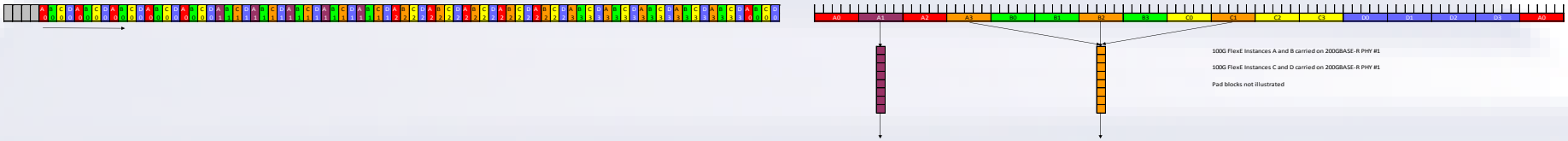


Figure 34: Example of 25G Calendar Slots Demultiplexed from a 400G FlexE PHY

11 Appendix C: (Informative) FlexE Client Synchronization

Clause 7.3.5 describes the mechanism for transporting synchronization information via the FlexE Group. The entities being synchronized are the FlexE Shims at each end of the group.

An additional application that may be useful in some implementation is to provide synchronization between MAC clients, where the client streams are transported over a FlexE Group. The application scenario is illustrated in Figure 35.

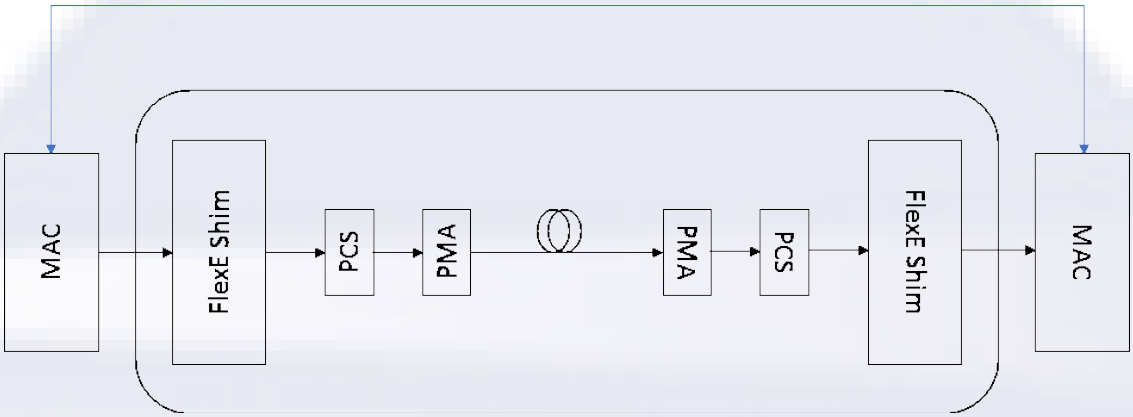


Figure 35: Illustration of exchange of PTP event messages between MAC clients

In [802.3] and [1588], the interface time stamp point for a PTP event message of a normal Ethernet client would be the time when the beginning of the first symbol after the SFD of the PTP event message passes the reference plane. When Ethernet streams are carried over a FlexE Group as FlexE Clients, additional considerations are required to emulate normal Ethernet PTP behavior. For example, a complication of an implementation that does this is that since the PTP message is sent in the FlexE Client stream, the actual position of the SFD at the MDI could be on any lane of any PHY of the FlexE Group. In the transmit direction, this implies that after the PTP messages are inserted into the client stream, the SFD position must be marked, and the marking of that position must be carried through the FlexE PHY distribution and the per PHY PCS lane distribution of each PHY so that the timestamps can be taken when the markers are detected at the PMA. In the receive direction, candidate timestamp positions (any SFD occurrence) must be recorded on each group of received data on each PHY, and this information must be propagated up through the FlexE demux so that once the FlexE Client stream is reassembled and the PTP message can be identified, the corresponding timestamp position can be identified. How to carry out this processing in an implementation is outside of the scope of this Implementation Agreement.

12 References

12.1 Normative references

[802.3] IEEE Std 802.3™-2015 *Standard for Ethernet*.

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

[G.709] ITU-T Recommendation G.709 (06/2016), *Interfaces for the Optical Transport Network*.

[G.8264] ITU-T Recommendation G.8264 (05/2014), *Distribution of timing information through packet networks*.

[802.1AB] IEEE Std 802.1™AB-2016, Station and Media Access Control Connectivity Discovery.

[1588] IEEE Std 1588-2008, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems

13 Appendix D: List of companies belonging to OIF when document was approved

Acacia Communications
ADVA Optical Networking
Alibaba
Amphenol Corp.
Anritsu
Arista Networks
Barefoot Networks
Broadcom Limited
Cadence Design Systems
Cavium
CenturyLink
China Telecom Global Limited
Ciena Corporation
Cisco Systems
Coriant
Corning
Credo Semiconductor (HK) LTD
Dell, Inc.
Elenion Technologies, LLC
Epson
eSilicon Corporation
Fiberhome Technologies Group
Finisar Corporation
Foxconn Interconnect Technology, Ltd.
Fujikura
Fujitsu
Furukawa Electric Japan
Gigamon Inc.
Global Foundries
Google
Hewlett Packard Enterprise (HPE)
Hitachi
Huawei Technologies Co., Ltd.
IBM Corporation
Infinera
Inphi
Integrated Device Technology
Intel
Invecas, Inc.
IPG Photonics Corporation
JCRFO
Juniper Networks
Kaiam
Kandou Bus
KDDI Research, Inc.
Keysight Technologies, Inc.
Lumentum
MACOM Technology Solutions
Marvell Semiconductor, Inc.
Maxim Integrated Inc.
MaxLinear Inc.
MediaTek
Mellanox Technologies
Microsemi Inc.
Microsoft Corporation
Mitsubishi Electric Corporation
Molex
Multilane SAL Offshore
NEC Corporation
NeoPhotonics
Nokia
NTT Corporation
O-Net Communications (HK) Limited
Oclaro
Orange
PETRA
Precise-ITC, Inc.
Qorvo
Ranovus
Renesas
Rianta Solutions, Inc.
Rockley Photonics
Rosenberger Hochfrequenztechnik GmbH & Co. KG
Roshmere
Samsung
Samtec Inc.
Semtech Canada Corporation
SiFotonics Technologies Co., Ltd.
Silab Tech Private Ltd.
Sino-Telecom Technology Co., Inc.
SK Telecom
SM Optics S.r.l.
Socionext Inc.
Spirent Communications
Sumitomo Electric Industries
Sumitomo Osaka Cement
Synopsys, Inc.
TE Connectivity
Tektronix
Teledyne LeCroy
Telefonica I + D
TELUS Communications, Inc.
UNH InterOperability Laboratory (UNH-IOL)
Verizon
Viavi
Xelic
Xilinx
Yamaichi Electronics Ltd.