BCM88480

FlexE Overhead, OAM, and IEEE 1588 Handling

Application Note

Table of Contents

Chapter 1: Introduction

1.1 About this Document

This document describes the mechanism that supports Flexible Ethernet (FlexE) overhead (OH), FlexE client OAM, and FlexE IEEE 1588 processing between the FlexE core of the BCM88480 device and the external processing unit (FPGA or CPU). It also provides information about the formats of the packets sent from and received by the FlexE core and external processing unit.

An explanation of how to process FlexE OH, FlexE client OAM, and FlexE IEEE 1588 is beyond the scope of this document.

1.2 FlexE OAM, OH, and IEEE 1588 Overview

To support FlexE client OAM, FlexE OH processing, and FlexE IEEE 1588, the adjacent FPGA (or CPU) should perform processing. For external processing, the external device must be connected to the BCM88480 through one or two 10G interfaces. The extracted data received from FlexE ports is packetized and sent to the processing unit, while the inserted data is sent (in a packetized format) from the processing unit to the FlexE core.

The two dedicated FlexE virtual clients for that traffic are as follows:

- Client 80 for OAM and OH
- Client 81 for IEEE 1588 traffic

Each client is sent to one of the two 10G interfaces.

The maximum bandwidth assumed on both clients is approximately 1 Gb/s per direction; however, to limit bursts, each direction of aggregated traffic should be shaped to 3 Gb/s with burst_size = 0 (in other words, the total bandwidth of both clients should not exceed 3 Gb/s per direction).

The high-level concept of operation is for the FlexE core in the BCM88480 to extract the relevant data from the incoming FlexE streams, packetize the extracted messages, and send them to the processing unit (that is, the external CPU or external FPGA). In the other direction, the FlexE core receives packets from the processing unit, unpacketizes those packets, and inserts the data into the relevant locations in the outgoing FlexE streams.

Chapter 2: Packet Formats

2.1 IEEE 1588 Packet Format

IEEE 1588 messages are inserted in or extracted from the FlexE OH, packetized, and transmitted from or received by the processing unit. Virtual client 81 transmits and receives the messages.

Figure 1: Internal IEEE 1588 Packet Format

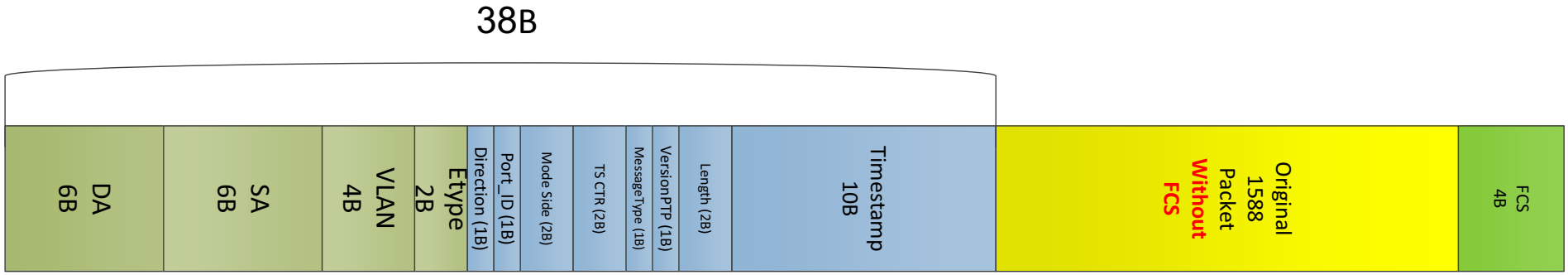


Table 1: IEEE 1588 Packet Fields

Field Name	Size	Description
DA	6B	Destination address (default value 0xFFFF_FFFF_FFFF).
SA	6B	Source address (default value 0x0).
VLAN	4B	The first 2 bytes are Vlan_Tag (default value 0x8100), and the next 2 bytes are Vlan_ID (values 0xB to 0x12 for the eight FlexE ports.
Etype	2B	EtherType (default value 0x88F7).
Direction	1B	0x2 for the FPGA to FlexE direction, and 0x1 for the opposite direction.
Port_ID	1B	Values 0x0 to 0x7 (OH management port).
Mode_side	2B	The bit values are as follows: ■ Bit[9:8] vlan_num: – 0x0: No VLAN. – 0x1: One VLAN. – 0x2: Two VLANs. ■ Bit[7:6] protocol_type: – 0x0: L2 Ethernet. – 0x1: IPv4/UDP. – 0x2: IPv6/UDP. ■ Bit[3]: Reserved. ■ Bit[2]: For testing. When set, the FlexE core will duplicate the PTP packet to transmit (standard path) and send back to the FPGA. ■ Bit[1]: work_step: (value = 1: Latch TX two-step timestamp). ■ Bit[0]: udpchsum_update: (Value = 1 update UDP checksum). ■ Other bits: Reserved 0.
TS_CTR	2B	The bit values are as follows: ■ Bit[8]: When set, update correction field (cf). ■ Bit[7:0]: cf offset value. ■ Other bits: Reserved 0.
Message_Type	1B	The bit values are as follows: ■ Bit[7:0]: 0x1f SSM. ■ Bit[7:4]: 0x0. ■ Bit[3:0]: Same with PTP message type.
VersionPTP	1B	Default value 0x2.
Length	2B	Total bytes of the blue and yellow areas in Figure 1, Internal IEEE 1588 Packet Format. In other words, the total bytes starting with the Direction field and ending with the original IEEE 1588 message (without FCS).
Timestamp	10B	{28'b0, 52b timestamp}. 52b are in a format of {48b nanoseconds, 4b fraction of nanoseconds}
Packet (without FCS)	Variable	Original PTP frame or SSM frame.
FCS	4B	FCS for the whole packet.

2.2 FlexE OH and Client OAM Handling

The FlexE OH and client OAM are inserted and extracted in the FlexE core, packetized, and transmitted to or received from the processing unit. Virtual client 80 transmits and receives the FlexE OH and client OAM. The processor for both OAM and OH must be the same unit because both share the same client_id.

The BCM88480 supports the following:

- Up to 8 ports for FlexE overhead extraction and insertion
- Up to 80 channels for OAM extraction and insertion in FlexE client layers in the line side
- Up to 80 channels for OAM extraction and insertion in FlexE client layers in the system side

2.2.1 Extract Process

The extract features for FlexE OH and client OAM are as follows:

- One FlexE overhead frame consists of eight 66B overhead blocks. The FlexE core packetizes one FlexE OH frame into an Ethernet packet and transmits the packet to the OH processor.
- The FlexE core packetizes N ($1 \le N \le 16$) OAM blocks into one Ethernet packet and transmits it to the external OH processor. Each packet contains a single OAM message. Different OAM messages have different numbers of blocks (as described in various locations throughout this document).

The following list describes the package frame structure:

- OH – In the direction of the overhead extraction, each port gets the corresponding overhead from the data port. After a certain port buffers eight consecutive 66b blocks of OH (that is, one full FlexE OH frame), the Ethernet encapsulation module encapsulates the data into an Ethernet frame format and sends it to the OH processor.
- OAM – After the OAM message is extracted, the Ethernet encapsulation module encapsulates the complete OAM message as a payload in the Ethernet frame and sends it to the OAM processor.
- The Ethernet encapsulation module can send the following three types of frame formats:
  - Data frame – This frame encapsulates the OH or OAM packet data in its payload area, and its frame format is shown in Figure 2, Data Packet Structure.
  - Request frame – After a FlexE port has transmitted the eight OH frames for that port, a request frame is sent for the port, and its frame format is as shown in Figure 3, Request Packet Structure (the request frame is only for OH, and only for the direction from FlexE core to the processing unit).
  - Data + request frame – When a port buffers eight OHs and there is a request signal in the transmission direction of a certain port, the data + request frame is sent. The frame format is the same as the data frame (see Figure 2, Data Packet Structure). Only the 0th bit of the STAT byte is different. For details, see the description of the data frame in (the data + request frame is only for OH, and only for the direction from the FlexE core to the processing unit).

Figure 2: Data Packet Structure

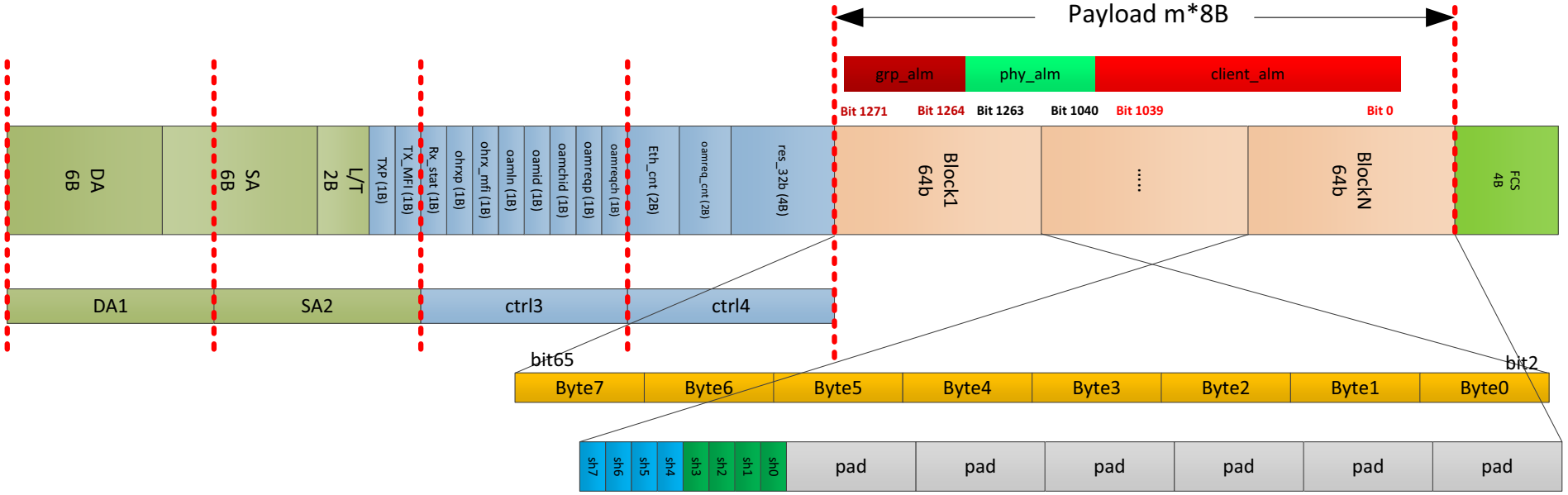
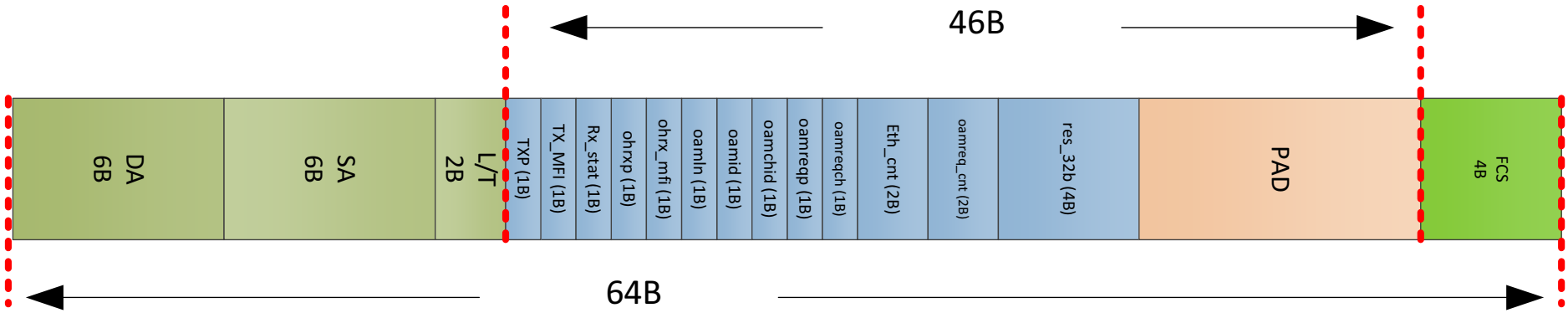


Table 2: Data Packet Fields

Field Name	Size	Description
DA	6B	Destination address (default value 0xFFFF_FFFF_FFFF).
SA	6B	Source address (default value 0x0).
L/T	2B	Length/type field.
TXP	1B	Overhead request port ID. This is valid only if rx_stat[2] = 1'b1 (OH request).
TX_MFI	1B	Overhead request MFI (a sequence number of the frame in a multiframe. Values are 0 to 31 for 100G, 200G, and 400G PHYs and 0 to 15 for 50G PHYs). This is valid only if rx_stat[2] = 1'b1 (OH request).
rx_stat	1B	The bits are as follows: ■ Bit[0]: Overhead SSF (for example, loss of frame or loss of multiframe). This is valid only if rx_stat[1] = 1'b1 (OH data) ■ Bit[1]: OH data identity. ■ Bit[2]: OH request identity. ■ Bit[3]: OAM data identity. ■ Bit[4]: OAM alarm identity ■ Bit[7:5]: Reserved.
ohrxp	1B	OH extraction direction port ID. This octet is valid only if rx_stat[1] = 1'b1 (OH data).
ohrx_mfi	1B	OH extraction direction MFI (a sequence number of the frame in a multiframe. Values are 0 to 31 for 100G, 200G, and 400G PHYs and 0 to 15 for 50G PHYs). This octet is valid only if rx_stat[1] = 1'b1 (OH data).
oampln	1B	Only bits[3:0] are used. These 4 bits indicate how many OAM block are in the payload domain. This field is valid only if rx_stat[3] = 1'b1 (OAM data). A value of 0 means one block, 1 means two blocks, and so on.
oampid	1B	OAM port ID. A value of 0 means line side, and 1 means system side. This field is valid only if rx_stat[3] = 1'b1 or rx_stat[4] = 1'b1 (OH data or alarm).
oamchid	1B	OAM channel ID. This field is valid only if rx_stat[3] = 1'b1 (OH data).
oamreqp	1B	Reserved.
oamreqch	1B	Reserved.
eth_cnt	2B	Sequence number of the Ethernet frame (cyclic).
oam_req_cnt	2B	Reserved.
Alm	159B	1272b: (80 clients × 13b) + (8 PHYs × 28b) + (8 groups × 1b)
sh0–sh7	—	The 66b coding bits (that is, bits 65 and 64 of each 66b block). sh0 has the 2 bits for first word, and sh7 has the 2 bits for last word. This means the whole packet has 8 + 1 blocks, with 8 blocks of data and one (last) block with codings (only for OH data).
FCS	4B	Ethernet FCS.

Figure 3: Request Packet Structure



2.2.2 Insertion Process

The insertion direction involves three different applications, which are handled as follows:

- FlexE port OH insertion – After receiving the request frame from the extract direction, the OH processor must send the OH frame of the corresponding multiframe to the FlexE core according to a specific packet format (see the following figure).
- Client OAM insertion – Upon receiving an OAM insertion packet, the FlexE core inserts the OAM message into the relevant client when possible. The FlexE core might also update *real-time* fields, such as BIP and RDI, as well as the CRC4 of the OAM.
- 66b_switch protection instruction – The controller sends the protection switching instruction to the FlexE core according to the specific packet format (see the following figure).

The insertion direction has only one type of packet, which is shown in the following figure.

Figure 4: Data Packet Structure – Insertion

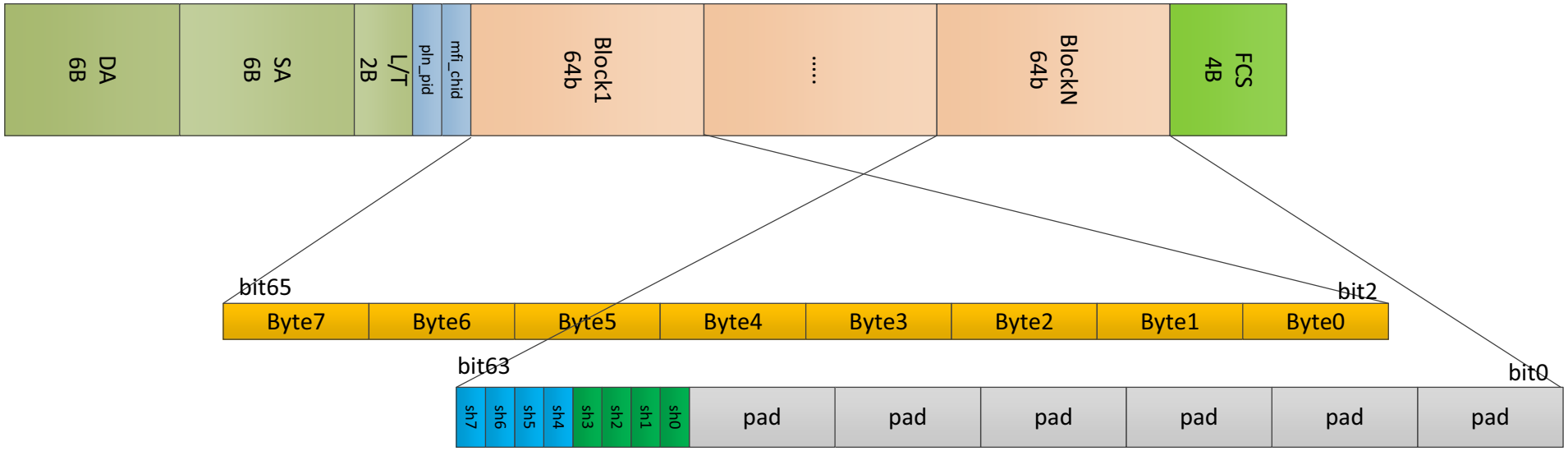


Table 3: Data Packet Fields – Insertion

Field Name	Size	Description
DA	6B	Destination address (default value 0xFFFF_FFFF_FFFF).
SA	6B	Source address (default value 0x0).
L/T	2B	Length/type field.
pln_pid	1B	The bits are as follows: ■ Bit[3:0] – Insertion port ID. – 0000 to 0111: 8 overhead ports. – 1000: Line-side OAM. – 1001: System-side OAM. – 1010: b66_switch switching instruction. ■ Bit[7:4]: The number of blocks in the payload (a value of 0 indicates one block, 1 indicates two blocks, and so on).
mfi_chid	1B	The bits are as follows: ■ Bit[4:0]: For an OH packet, it means MFI. ■ Bit[6:0]: For an OAM packet, it carries the channel ID.
FCS	4B	Ethernet FCS.

2.2.3 OAM Supported Features and Format

The BCM88480 FlexE core supports the following OAM features:

- Connectivity testing (CC)
- Connectivity verification (CV)
- Bit error detection (BIP)
- Remote error indication (REI)
- Remote defect indication (RDI)
- Delay measurement (1DM and 2DM)
- Protection switching (APS)
- Customer signal type (CS)
- Customer signal failure indication (CSF)

Chapter 3: External Processor (FPGA or CPU) Handling

NOTE: Throughout this document, references to the FPGA or CPU are collectively called *FPGA*.

3.1 IEEE 1588 Handling

Each PTP packet should be less than 256B.

It is assumed that the IEEE 1588 servo runs in the FPGA, so this chapter describes how to transform packets from the FlexE core into a format the servo can handle. This chapter also describes how to get packets from the servo, transform them, and transmit them into the FlexE core.

For the packet format to and from FlexE core, see Figure 1, Internal IEEE 1588 Packet Format.

3.2 FlexE OH Handling

As a general concept, the FlexE core collects the extracted OH data of each FlexE port's frame. The FlexE core transmits the OH data to the FPGA in a packetized form. In the other direction, the FPGA sends a packet with the FlexE port's frame data to be inserted into the FlexE stream. This type of packet is sent based on a request from the FlexE core and should arrive at the FlexE core within 300 µs of the request. The reason for the delay is because the FlexE core has a buffer that stores four frames per FlexE port. Each frame transmission is ~100 µs. The FlexE core stores the OH of three frames in the internal buffer. When sending a request for a new frame's OH, the FlexE core may transmit the OH stored in its buffer. Because of this, the new frame's OH must arrive at the FlexE core within 300 µs (3 × 100 µs).

The BCM88480 is capable of receiving a complete FlexE OH frame from the FPGA, but this is not currently supported. The default behavior is that blocks 1, 2, and 3 are handled by the BCM88480, and blocks 4 through 8 are handled by the FPGA and can optionally be provided (if the customer application requires them).

For insertion, blocks 1, 2, and 3 are completely handled by the BCM88480 FlexE core (by configuring SDK APIs) and should not be managed by the FPGA (that is, referring to Figure 8, OH Packet Format – Insertion, the BCM88480 will ignore blocks 1, 2, and 3 arriving from the FPGA).

The FPGA should supply blocks 4 through 8 to the BCM88480.

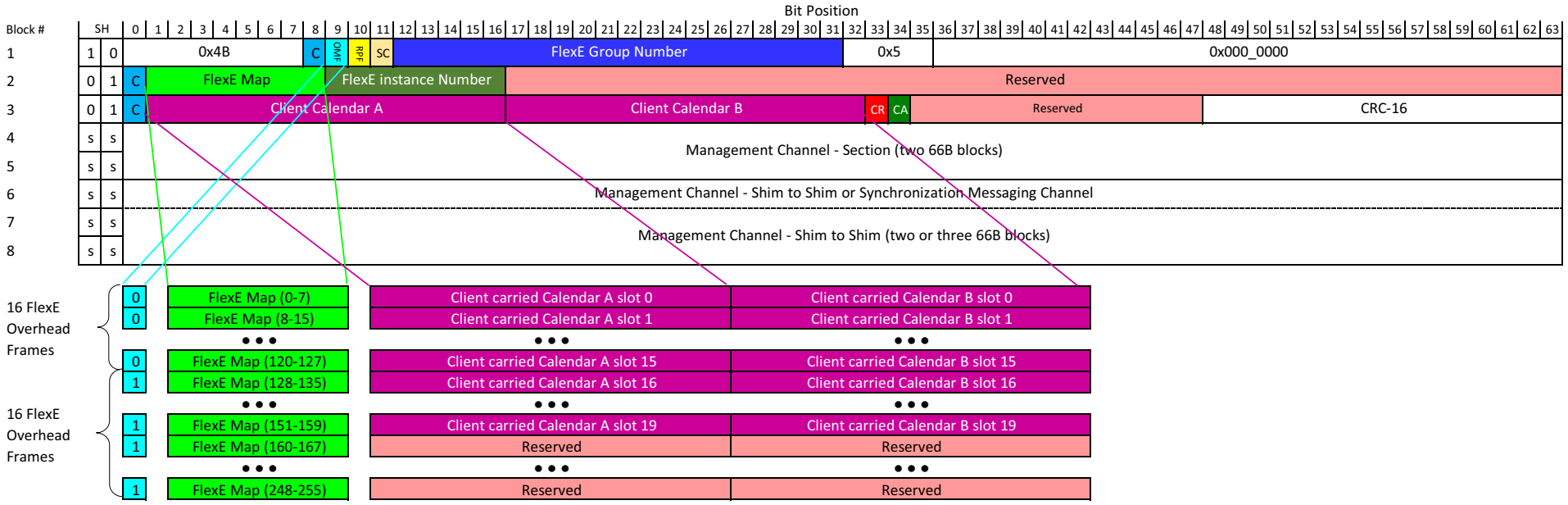
When the FPGA is not sending any OH packets towards the BCM88480, the BCM88480 FlexE core continues to send OH frames (similar to idle frames) towards the FlexE interface, which maintains the OH stream for the FPGA.

For extraction, blocks 1, 2, and 3 are handled by the BCM88480. Use SDK APIs to get the information extracted by the FlexE core from those blocks.

OH blocks 4 through 8 are extracted by the BCM88480 FlexE core and are sent to the FPGA inside blocks 4 through 8 in the OH packet format (see Figure 9, OH Packet Format – Extraction).

The following figure shows the frame and multiframe structure as defined by the Optical Internetworking Forum (OIF).

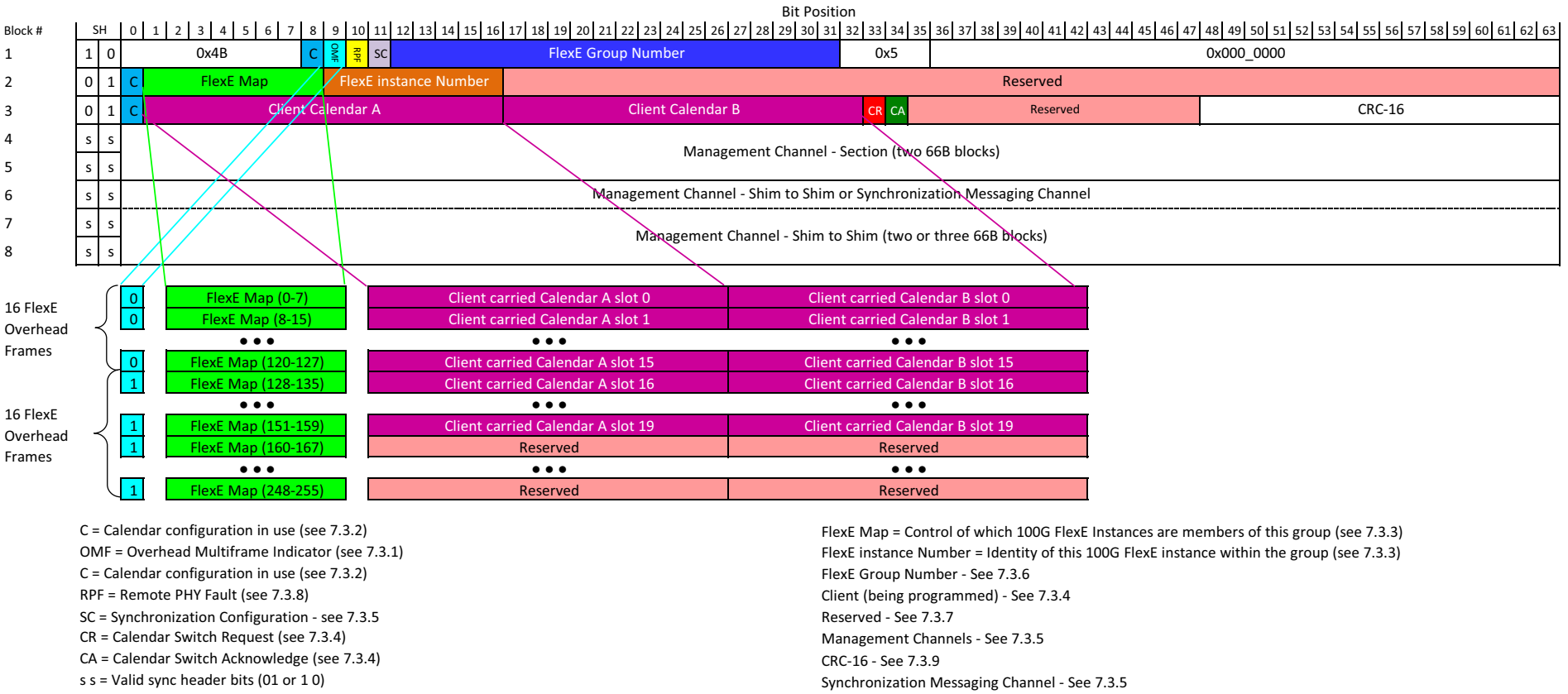
Figure 5: FlexE Overhead Frame and Multiframe of Each 50G FlexE Instance



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 FlexE instances are members of this group (see 7.3.3)
FlexE instance Number = Identity of this 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 6: FlexE Overhead Frame and Multiframe of Each 100G FlexE Instance



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)

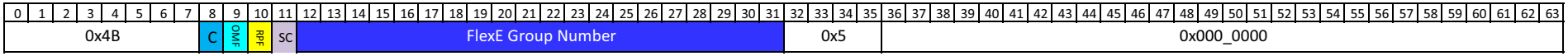
FlexE Map = Control of which 100G FlexE Instances are members of this group (see 7.3.3)
FlexE instance Number = Identity of this 100G FlexE instance within the group (see 7.3.3)
FlexE Group Number - See 7.3.6
Client (being programmed) - See 7.3.4
Reserved - See 7.3.7
Management Channels - See 7.3.5
CRC-16 - See 7.3.9
Synchronization Messaging Channel - See 7.3.5

Each OH packet should be 100B when transmitted from the FPGA to the FlexE core and 116B when it is received from the FlexE core.

As shown in Figure 5 and Figure 6, the full information transmitted on the FlexE OH actually consumes 32 consecutive frames (one multiframe) per instance.

For example, the first block of each frame should look like the block in the following figure.

Figure 7: First Block of FlexE Overhead Frame



The sync bits of this block are 2'b10 and are located in the 9th block of each frame in bits[49:48]. The format of this information is compatible with the packetization format of it, as shown in Figure 2, Data Packet Structure.

The following two figures show the OH packet formats in the insertion and extraction directions.

NOTE: In the extraction direction, the data-req field in the packet may be set or not set.

Figure 8: OH Packet Format – Insertion

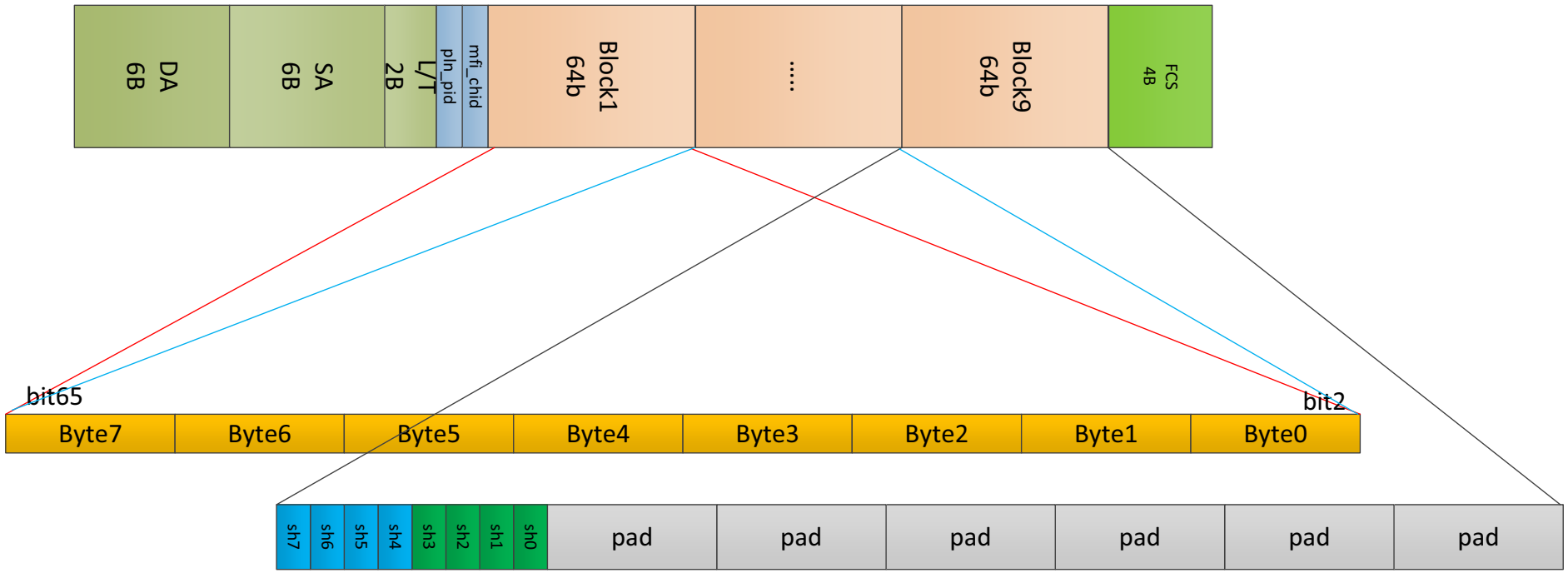
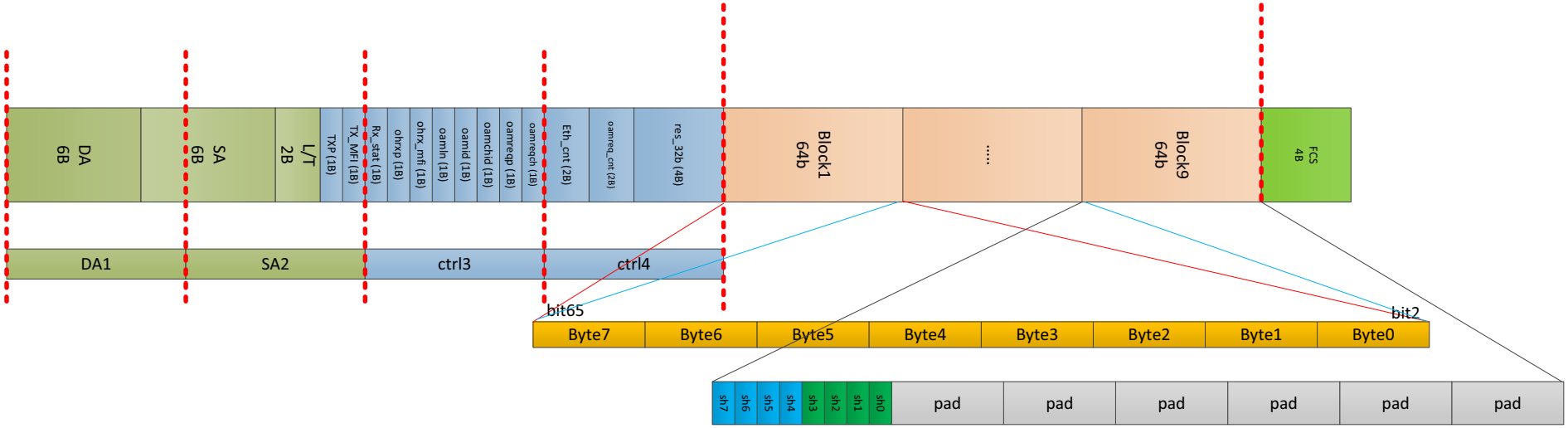


Figure 9: OH Packet Format – Extraction

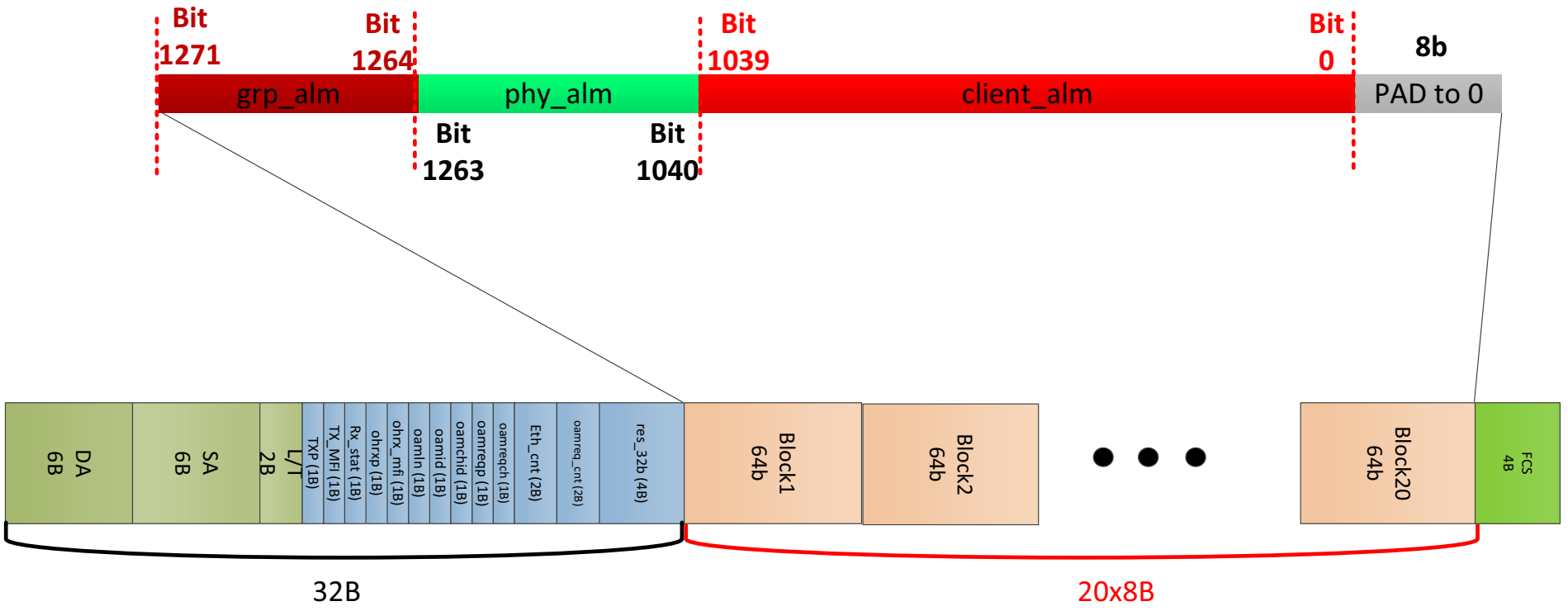


Chapter 4: FlexE Alarms

4.1 Alarm Packet Data Structure

The following figure shows the packet format for the FlexE alarms.

Figure 10: Alarm Packet Format



The alarm packet is generated once every 70 µs (configurable). Client 80 extracts the 1272-bit alarm in the direction from the FlexE core to the FPGA.

The following list describes the alarms:

- Alm[1271:1264]: FlexE group deskew alarm.
- Alm[1263:1040]: PHY layer alarm. Each PHY instance has 28 bits of alarms (see the following table).

Table 4: PHY Layer Alarm

Bit number	Alarm Name	Description
Bit[27:8]	ccm	Calendar mismatch
Bit[7]	link_down	PCS layer loss of signal (LoS)
Bit[6]	lof_alm	LoS of frame alarm
Bit[5]	lom_alm	LoS of multi-frame alarm
Bit[4]	gid_alm	GID mismatch
Bit[3]	pmm_alm	PHY map mismatch
Bit[2]	lop_alm	LoS of pad (only for 50G FlexE)
Bit[1]	pid_alm	PHYID mismatch
Bit[0]	rpf_alm	RPF alarm

- Alm[1039:0] client layer alarm. Each client has 13 bits (see the following table).
  - bits[12:0] are alarms of ch0.
  - bits[25:13] are alarms of ch1.

Table 5: Client Layer Alarm

Bit Number	Alarm Name	Description
Bit[12]	client_bas_period_alm	BAS period mismatch
Bit[11]	sfbip_alm	sf_bip alarm
Bit[10]	sfbei_alm	sf_bei alarm
Bit[9]	bas_los_alm	BAS loss alarm
Bit[8]	bas_cs_lf_alm	Local fault alarm in BAS
Bit[7]	bas_cs_rf_alm	Remote fault alarm in BAS
Bit[6]	bas_csf_lpi_alm	LPI alarm in BAS
Bit[5]	bas_rdi_alm	RDI alarm in BAS
Bit[4]	sdbip_alm	Bip-SD alarm in BAS
Bit[3]	sdbei_alm	BEI-SD alarm in BAS
Bit[2]	client_lf_alm	Local fault alarm in client
Bit[1]	client_rf_alm	Remote fault alarm in client
Bit[0]	client_lpi_alm	LPI alarm in client

NOTE: For the alarm packet of clients in SAR, bits [1271:1040] are 0.

Chapter 5: Protection Command Packet

Two protection schemes are supported 1:1 and 1 + 1.

Figure 11: Protection Packet Format

data[63]	en[63 : 56]	en[127 : 120]	en[191 : 184]	en[255 : 248]	en[319 : 312]	en[383 : 376]	en[447 : 440]	Reserved	Reserved	...	Reserved
	en[55 : 48]	en[119 : 112]	en[183 : 176]	en[247 : 240]	en[311 : 304]	en[375 : 368]	en[439 : 432]	Reserved	Reserved	...	Reserved
	en[47 : 40]	en[111 : 104]	en[175 : 168]	en[239 : 232]	en[303 : 296]	en[367 : 360]	en[431 : 424]	en[495 : 488]	Reserved	...	Reserved
	en[39 : 32]	en[103 : 96]	en[167 : 160]	en[231 : 224]	en[295 : 288]	en[359 : 352]	en[423 : 416]	en[487 : 480]	Reserved	...	Reserved
	en[31 : 24]	en[95 : 88]	en[159 : 152]	en[223 : 216]	en[287 : 280]	en[351 : 344]	en[415 : 408]	en[479 : 472]	Reserved	...	Reserved
	en[23 : 16]	en[87 : 80]	en[151 : 144]	en[215 : 208]	en[279 : 272]	en[343 : 336]	en[407 : 400]	en[471 : 464]	Reserved	...	Reserved
	en[15 : 8]	en[79 : 72]	en[143 : 136]	en[207 : 200]	en[271 : 264]	en[335 : 328]	en[399 : 392]	en[463 : 456]	Reserved	...	Reserved
data03]	en[7 : 0]	en[71 : 64]	en[135 : 128]	en[199 : 192]	en[263 : 256]	en[327 : 320]	en[391 : 384]	en[455 : 448]	Reserved	...	Reserved

64b Block#	0	1	2	3	4	5	6	7	8	...	15

Table 6: Protection Packet Fields

Field	Bits	Definition
pln_pid	1B	The bits are as follows: ■ Bit[3:0]: Insertion port ID: 4'b1010. b66_switch switching instruction. ■ Bit[7:4]: 0xF.
mfi_chid	1B	0x0.

A total of 242 clients exist for protection: 80 clients on the line side, 80 clients on SAR, and 80 + 2 clients on the MAC side (that is, between the FlexE core and MAC). The two additional clients are the virtual clients for IEEE 1588, OH, and OAM). All control (enable) bits are active high.

- Bits[1:0] – Line_side client 0
- Bits[3:2] – Line_side client 1
- ….
- Bits[159:158] – Line_side client 79
- Bits[163:160] – Reserved
- Bits[165: 164] – SAR client 0
- Bits[167: 166] – SAR client 1
- …..
- Bits[323: 322] – SAR client 79
- Bits[327:324] – Reserved
- Bits[329: 328] – MAC_side client 0
- Bits[331:330] – MAC _side client 1
- …..
- Bits[489: 488] – MAC_side client 80
- Bits[491:490] – MAC _side client 81
- Bits[495:492] – Reserved

5.1 Protection Examples

This section describes how the protection schemes work.

For any client without protection, en[1:0] = 2'b11.

5.1.1 Example – 1 + 1 Protection

This example has three clients called clientA, clientB, and clientC. The main path is clientA ↔ clientB, and the redundant path is clientA ↔ clientC.

The en[1:0] bit values for each client is as follows:

- For clientA, en[1:0] = 2'b11.
- For clientB (when the main path is selected), en[1:0] = 2'b01.
- For clientB (when the redundant path is selected), en[1:0] = 2'b00.
- For clientC (when the main path is selected), en[1:0] = 2'b00.
- For clientC (when the redundant path is selected), en[1:0] = 2'b10.

5.1.2 Example – 1:1 Protection

This example has three clients called clientA, clientB, and clientC. The main path is clientA ↔ clientB, and the redundant path is clientA ↔ clientC. In this scenario, clientB and clientC cannot coexist.

The en[1:0] bit values for each client is as follows:

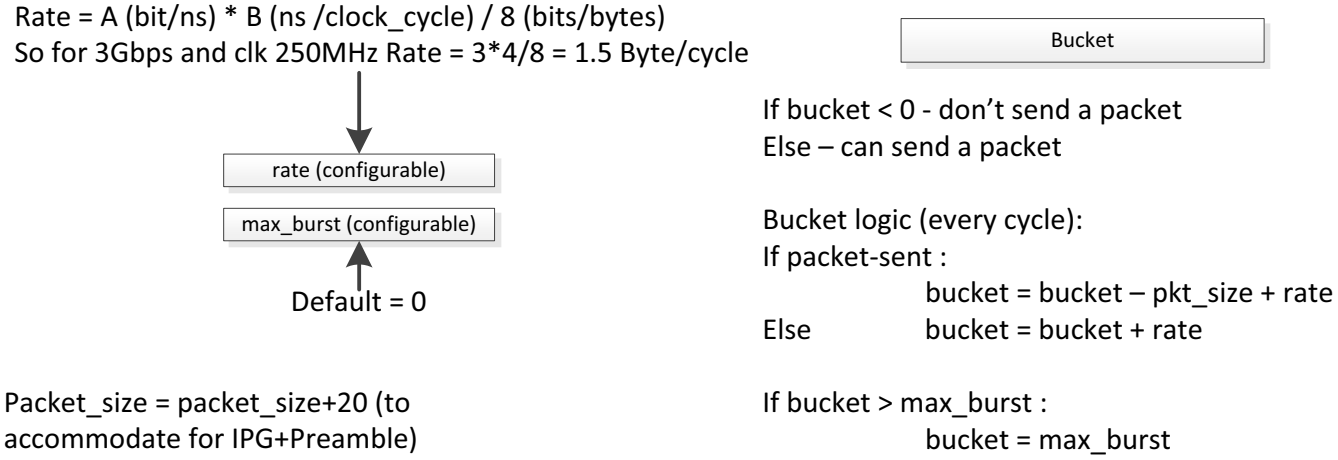
- For clientA (when the main path is selected), en[1:0] = 2'b01.
- For clientA (when the redundant path is selected), en[1:0] = 2'b10.
- For clientB (when the main path is selected), en[1:0] = 2'b01.
- For clientB (when the redundant path is selected), en[1:0] = 2'b00.
- For clientC (when the main path is selected), en[1:0] = 2'b00.
- For clientC (when the redundant path is selected), en[1:0] = 2'b10.

Chapter 6: Shaping

The overall rate of traffic from the FPGA to the BCM88480 device should be less than 3 Gb/s.

To achieve this rate, all packets to the BCM88480 should be stored in a FIFO before they are transmitted to the MACs of the FPGA. The read rate from the FIFO should be 3 Gb/s with burst_size=0.

Figure 12: Shaping




BROADCOM®