



# **PM5440 DIGI 120G**

## **Hardware Specification**

### **DIGI 120G**

**PM5440**

PMC-2110137

Released

Issue 5: November 2013

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:06 PM

Copyright © 2013 PMC-Sierra, Inc. All rights reserved.

The information in this document is proprietary and confidential to PMC-Sierra, Inc. In any event, no part of this document may be reproduced or redistributed in any form without the express written consent of PMC-Sierra, Inc.

PMC-2110137, Issue 5

None of the information contained in this document constitutes an express or implied warranty by PMC-Sierra, Inc. as to the sufficiency, fitness or suitability for a particular purpose of any such information or the fitness, or suitability for a particular purpose, merchantability, performance, compatibility with other parts or systems, of any of the products of PMC-Sierra, Inc., or any portion thereof, referred to in this document. PMC-Sierra, Inc. expressly disclaims all representations and warranties of any kind regarding the contents or use of the information, including, but not limited to, express and implied warranties of accuracy, completeness, merchantability, fitness for a particular use, or non-infringement.

In no event will PMC-Sierra, Inc. be liable for any direct, indirect, special, incidental or consequential damages, including, but not limited to, lost profits, lost business or lost data resulting from any use of or reliance upon the information, whether or not PMC-Sierra, Inc. has been advised of the possibility of such damage.

For a complete list of PMC-Sierra's trademarks and registered trademarks, visit: <http://www.pmc-sierra.com/legal/>.

Other product and company names mentioned herein may be the trademarks of their respective owners.

United States Patent No. 7,733,149; 7,876,866; 7,986,190; 7,884,660; 8,261,167; 8,386,897; 8,533,546; 8,542,708.

Other relevant patent grants may also exist.

## Revision History

Issue	Issue Date	Details of Change
5	November 2013	<p>Fifth issue of the PM5440 DIGI_120G HWS.</p> <p><b>Legal</b></p> <ul style="list-style-type: none"> <li>• New patent grants added</li> <li>• Added US patent number 8,542,708.</li> </ul> <p><b>Section 4.5.2.1 System Interface Mapping</b></p> <ul style="list-style-type: none"> <li>• Added note that the eight Backplane SR(C-8) lanes cannot be split between the two Interlaken interfaces.</li> </ul> <p><b>Section 5.8 COREOTN:</b></p> <ul style="list-style-type: none"> <li>• Updated the section called "Pool-Based ODUk Defects" Pool-Based ODUk Defects, Client Defects, to specify the RCP channel mapping for 40G SONET streams: {3, 7, 11} vs. {0, 4, 8}.</li> </ul> <p><b>Section 5.8.4.7.2 ODUk-LCK/ODUk-AIS/ODUk-OCI Force Signal Replacement</b></p> <ul style="list-style-type: none"> <li>• Added note that when ODUk-LCK, ODUk-AIS or ODUk-OCI are forced, the transmit ODUk rate can revert to nominal timing to.</li> </ul> <p><b>Section 5.10 CPU Subsystem</b></p> <ul style="list-style-type: none"> <li>• Changed broadcast limitations in CPB Overview to record that there is support for a unidirectional "mirror" port, triple-broadcast in a limited fashion in addition to the typical 2-way broadcast.</li> </ul> <p><b>Section 5.16 Diagnostic and Test Features:</b></p> <ul style="list-style-type: none"> <li>• Updated Table 88 and Table 89 with new values for core clock, integer, numerator, and denominator.</li> <li>• Added Demapped Ethernet Clocks (DEMAP_OUT_CLK[11:0]) to Section 5.16.1.3 and Table 86</li> <li>• Added new section for Demapped ENET Recovered Clock Jitter Section 5.16.1.3.7.</li> </ul> <p><b>MAPOTN:</b></p> <ul style="list-style-type: none"> <li>• Revised ILKN Processor Limit (Gbps) for rows 49B and 50B of Table 48.</li> <li>• Updated Section 5.9.6, Section 5.9.6.6, Section 5.9.6.3, and Section 5.9.6 because the ordered set UPI value setting is global.</li> <li>• Updated Section 5.9.7.7 because eFCS cannot be enabled in the COPI &lt;&gt; DCPB &lt;&gt; MAPPER path.</li> </ul> <p><b>Section 5.18 Device Latency</b></p> <ul style="list-style-type: none"> <li>• In Section 5.18.6, replaced the Summary of OTN FPI Latencies table with Table 111 and updated the description.</li> </ul> <p><b>Section 5.6.5 FEC Block</b></p> <ul style="list-style-type: none"> <li>• Updated FEC descriptions.</li> </ul> <p><b>Section 5.8.12.2.3 Defects Information Ring Control Port</b></p> <ul style="list-style-type: none"> <li>• Revised note about dLOOMFI defect to RCP to subsection OTU and Channel-Based HO ODUk Defects.</li> </ul> <p><b>Section 5.8.12.2.3 OTU</b></p> <ul style="list-style-type: none"> <li>• Revised note about dLOOMFI defect to RCP.</li> </ul> <p><b>Section 5.8.12.2 Defects Information Ring Control Port.</b></p> <ul style="list-style-type: none"> <li>• Revised note about Channel-Based HO ODUk Defects.</li> </ul> <p><b>Section 5.8.4.6.1 ODU Receive Framer (ODU_RFRM) Atomics</b></p> <ul style="list-style-type: none"> <li>• Added note re: pN_DS availability</li> </ul> <p><b>Figure 118</b></p> <ul style="list-style-type: none"> <li>• Added note about not splitting the eight Backplane SR(C-8) lanes after section.</li> </ul> <p><b>Section 5.11.4.2.1 Combined OTN and Packet Interface (COPI) Block Feature Overview</b></p> <ul style="list-style-type: none"> <li>• Added note to expand the information about the behaviour of the device's receive interface when receiving non-EOP bursts. Also, added new paragraph and list as specified re: scenarios when receiving packets is not supported.</li> </ul> <p><b>Section 5.6.3.3 FEC Features</b></p> <ul style="list-style-type: none"> <li>• Added list of supported bypass FEC operations.</li> </ul> <p><b>Section 5.11.5.2.4</b></p> <ul style="list-style-type: none"> <li>• Rewrote the list items addressing the values transmitted on the multi-use field of the Interlaken control word.</li> </ul> <p><b>Section 6 Non Compliances</b></p>

Issue	Issue Date	Details of Change
		<ul style="list-style-type: none"> <li>• <a href="#">Section 6.1.7</a> added details about DDEG functionality.</li> <li>• Added section <a href="#">Section 6.3.1</a> to describe issues with HI_BER_V not being detected continuously when HiBER alarm sent.</li> </ul> <p><b>Section 6.2.1</b></p> <ul style="list-style-type: none"> <li>• Updated the workaround.</li> </ul> <p><b>Table 110</b></p> <ul style="list-style-type: none"> <li>• Updated SRAM Burst Write from 1620 to 532ns</li> </ul> <p><b>Section 9.4</b> Power Requirements</p> <p><b>Section 9.5</b> from "Nominal +2 sigmas of process variation" to "Nominal + expected process variation"</p> <ul style="list-style-type: none"> <li>• Changed Thermal Conditions.</li> <li>• Updated name and issue number of PMC-2100078, '40nm RAPTR SERDES Top Level Specifications for DIGI Family of Devices', issue 6.</li> </ul> <p><b>Section 9.5</b> Conditions for Power Requirements</p> <ul style="list-style-type: none"> <li>• Changed Thermal Conditions in from "Nominal +2 sigmas of process variation" to "Nominal + expected process variation".</li> <li>• Updated description.</li> </ul>
4	July 2013	<p>Fourth issue of the DIGI 120G Hardware Specification.</p> <p>New patent grants added</p> <p>Updated the reference to the G709 standard</p> <p>All Sections:</p> <ul style="list-style-type: none"> <li>• All references to the "DSIS" within the document changed to be either "LINE_DSIS" or "SYS_DSIS" to clarify.</li> <li>• All references to the SIFD "SERDES MUX" within the document changed to "SIFD_MUX".</li> <li>• All references to the "LIFD_MUX subsystem" within the document are now LIFD Subsystem. LIFD subsystem contains the LIFD_MUX and LINE_DSIS.</li> </ul> <p><b>Section 4</b> Pinout:</p> <ul style="list-style-type: none"> <li>• Updated TESTMODE pin description to include LVCMOS pad PUPD enable/disable <a href="#">Table 3</a></li> <li>• Updated SFI1_RXS, SFI2_RXS, SFI1_TXDSC_P/N, and SFI2_TXDSC_P/N description in <a href="#">Table 5</a></li> <li>• Updated the PLL_STATUS[1:0] description in <a href="#">Table 3</a> to sync with errata (<a href="#">Errata HWS-066</a>)</li> <li>• Corrected the IRIG_CLKI description in <a href="#">Table 16</a></li> <li>• Corrected the Interlaken Interface 2 lane support in <a href="#">Table 23</a> (<a href="#">Errata HWS-062</a>)</li> <li>• Updated the OTN Frame Pulse Interface description <a href="#">Section 4.2.10</a> to describe behavior during an OOF state (<a href="#">Errata HWS-064</a>)</li> <li>• Updated the maximum number of Interlaken lanes supported in <a href="#">Section 4.5.3</a></li> <li>• Updated <a href="#">Section 4.5.1.2</a>, <a href="#">Section 4.5.1.3</a>, <a href="#">Section 4.5.2.1</a> to sync with errata describing LINE_LOS and DLOS use restrictions (<a href="#">Errata HWS-063</a>)</li> <li>• Additional details added to <a href="#">Section 4.2.9</a></li> <li>• Updated the client protocol for the Line Interface SFI-S in <a href="#">Section 4.3</a></li> <li>• <b>Section 4.3:</b> CBR10G and CBR40G client protocol options were removed from the Line Interface, SFI5.1 Interface, and System Interface. These are a duplicate of OC-768, STM 256, OC-192, and STM-64 options. SFP+ Limiting and CXP electrical interface options were removed from the Line and System Interfaces.</li> <li>• The Interlaken Interface now supports arbitrary lane re-ordering instead of just high order to low order lane reversal (see <a href="#">Section 4.5.3</a>).</li> <li>• Changed "differential LVDS input" description to "differential LVDS input with integrated termination" for REFCLK_P/N, SFI*_TXREFCLK and SFI*_TXCKSRC in <a href="#">Section 4.2.1</a> and <a href="#">Section 4.2.3</a></li> </ul> <p><b>Section 5.1</b> Block Diagram:</p> <ul style="list-style-type: none"> <li>• For ENET(LINE) and ENET(SYS) subsystems in the diagram: <ul style="list-style-type: none"> <li>• "G.709 7.1/7.2 Enc/Dec" changed to "40GE G.709 17.7.4.1 Transcode"</li> <li>• For the DSI connection from LIFD/SIFD to the PMM, a block called "10GE GSup43 7.1/7.2" was added.</li> <li>• For the connection between the MLD and PMM, a block called "100GE G.709 17.7.5.1" was added.</li> </ul> </li> </ul>

Issue	Issue Date	Details of Change
		<ul style="list-style-type: none"> <li>The LIFD_MUX subsystem was renamed to LIFD and a block called "LIFD_MUX &amp; LINE_DSIS" replaced the original mux's. For details of this block please see <a href="#">Section 5.5</a>.</li> <li>The block "SERDES MUX &amp; DSIS" inside the SIFD subsystem was renamed to "SIFD_MUX &amp; SYS_DSIS"</li> <li>CBR10G and CBR40G options were removed from the Line Interface, SFI5.1 Interface, and System Interface. These are a duplicate of OC-768, STM 256, OC-192, and STM-64 options</li> <li>In the CBRC subsystem: <ul style="list-style-type: none"> <li>CBR10G and CBR40G were removed and FC1200 added</li> <li>FC1200 is now "FC1200 Transcode"</li> </ul> </li> <li>The OTN FPI Interface now connects to the MLD blocks and not the OTU Framer blocks within the LINEOTN and SYSOTN subsystems.</li> </ul> <p><b>Section 5.4 OPSA:</b></p> <ul style="list-style-type: none"> <li>Updated <a href="#">Section 5.4.17</a> to sync with erratas (<a href="#">Errata HWS-072</a> and <a href="#">Errata FCN-032</a>)</li> <li>Added block diagram clarification throughout <a href="#">Section 5.4</a></li> <li>Added clarification to <a href="#">Section 5.4.8</a></li> <li>Updated <a href="#">Section 5.4.15</a> and <a href="#">Section 5.4.16</a></li> <li>Updated datapaths in <a href="#">Section 5.4.10</a></li> </ul> <p><b>Section 5.5 LIFD:</b></p> <ul style="list-style-type: none"> <li>Updated <a href="#">Section 5.5.3</a> to sync with errata (<a href="#">Errata HWS-082</a>)</li> <li><a href="#">Figure 13</a> was updated to better show the premuxing in the LINE_DSIS before the 12x12 crosspoint mux. The diagram was also updated to show the LIFD_MUX and LINE_DSIS portions of the diagram</li> <li>References in this section to "DSIS" changed to "LINE_DSIS" throughout entire section (text and figures)</li> </ul> <p><b>Section 5.6 LINEOTN:</b></p> <ul style="list-style-type: none"> <li>LINEOTN block diagram was updated to show the OTN Frame Pulse Interface, additional labels were also added to better show the connections to the other subsystems <a href="#">Section 5.6.1</a>.</li> <li>Updated and moved the FEC Latency table in <a href="#">Section 5.6.5.5</a> to <a href="#">Section 5.18.2</a></li> <li>Updated section and added note about encoder / decoder bypass to <a href="#">Section 5.6.3.3</a> to sync with errata (<a href="#">Errata FCN-033</a>)</li> <li>Updated OTU_FRM features in <a href="#">Section 5.6.3.4</a></li> <li>Updated feature list in <a href="#">Section 5.6.4.4</a></li> <li>Added note on Swizzle FEC to <a href="#">Section 5.6.5.4</a></li> <li>Updated <a href="#">Section 5.6.8.1</a> and <a href="#">Section 5.6.8.2</a></li> <li>Added <a href="#">Section 5.6.8.5</a></li> <li>Updated <a href="#">Section 5.6.8.6</a></li> <li>Added a new section for the OTN Frame Pulse Interface <a href="#">Section 5.6.4.8</a>: <ul style="list-style-type: none"> <li>Some of the text content was taken from <a href="#">Section 4.2.10</a></li> <li>New diagram was created to show the how the muxing is wired up between LINEOTN and SYSOTN</li> <li>Similar section was added to the SYSOTN section with a link back to this section <a href="#">Section 5.7.7</a></li> </ul> </li> </ul> <p><b>Section 5.7 SYSOTN:</b></p> <ul style="list-style-type: none"> <li>Various updates to entire section</li> </ul> <p><b>Section 5.8 COREOTN:</b></p> <ul style="list-style-type: none"> <li>Main block diagram has been updated <a href="#">Section 5.8.1</a>.</li> <li>Updated G.7044 G.HAO red/blue protocol terminology to LCR/BWR</li> <li>Updated <a href="#">Section 5.8.11</a> (<a href="#">Errata HWS-061</a>)</li> <li>Updated diagram <a href="#">Figure 54</a>.</li> <li>Updated <a href="#">Section 5.8.8</a> to sync with errata (<a href="#">Errata HWS-057</a>)</li> <li>Added ODUk IAE description to <a href="#">Section 5.8.8</a> to sync with errata (<a href="#">Errata HWS-078</a>)</li> <li>Updated <a href="#">Section 5.8.4.13</a> to sync with errata HWS-058 (<a href="#">Errata HWS-058</a>).</li> <li>Added broadcast restriction to <a href="#">Section 5.8.5</a> to sync with errata (<a href="#">Errata HWS-081</a>)</li> </ul>

Issue	Issue Date	Details of Change
		<ul style="list-style-type: none"> <li>Added description about the state of dMSIM following ODUk channel in re-provisioning to <a href="#">Section 5.8.4.6.3</a> to sync with errata (<a href="#">Errata HWS-054</a>)</li> <li>Added multicast restrictions to the ODUk switch functional description in <a href="#">Section 5.8.5</a> to sync with errata (<a href="#">Errata HWS-073</a>)</li> <li>Removed ingress reference from <a href="#">Section 5.7.4.1.2</a></li> <li>Added the ability to add/remove and reconfigure LO clients into HO without affecting HO frame structure or other active LOs (<a href="#">Section 5.8.4.6.3</a>, <a href="#">Section 5.8.4.6.4</a>)</li> </ul> <p><b>Section 5.9 MAPOTN:</b></p> <ul style="list-style-type: none"> <li>Updated G.HAO red/blue protocol terminology to G.7044 LCR/BWR</li> <li>Added <a href="#">Section 5.9.3.4</a> to clarify throughput of the device GFP mapper and demapper (<a href="#">Errata HWS-060</a>)</li> <li>Updated Atomics Function diagram <a href="#">Figure 79</a> and removed OPUk (k=3e2) reference in <a href="#">Section 5.9.8.2</a> (<a href="#">Errata HWS-075</a>)</li> <li>Clarified that the first packet dropped after initial provisioning of a G.7041 7.9 datapath in <a href="#">Section 5.9.6.3</a> (<a href="#">Errata HWS-079</a>)</li> <li>Updated <a href="#">Section 5.9.6.5</a></li> </ul> <p><b>Section 5.10 CPB:</b></p> <ul style="list-style-type: none"> <li>Updated the SCPB Port Data Routing Options figures (<a href="#">Figure 97</a> and <a href="#">Figure 97</a>)</li> <li>Added <a href="#">Section 5.10.13</a> (<a href="#">Errata FCN 009</a>)</li> <li>Added broadcast restriction to <a href="#">Section 5.10.1</a> to sync with errata (<a href="#">Errata HWS-081</a>)</li> <li>Removed references to multicast since multicast by the CPB is not supported in <a href="#">Section 5.10.1</a>, <a href="#">Section 5.10.6</a>, <a href="#">Section 5.10.7</a>, and <a href="#">Section 5.8.5.1</a> to sync with errata (<a href="#">Errata HWS-073</a>)</li> <li>Updated <a href="#">Figure 100</a>.</li> <li>Updated table and text in describing DPI_TYPE usage in <a href="#">Section 5.10.3</a>.</li> <li>Removed rows "SIFD(PIF) -&gt; ENET120(GSup43 7.3)" and "ENET120(GSup43 7.3) -&gt; SIFD(PIF)" in <a href="#">Table 55</a></li> <li>Edits for grammar/style</li> <li>Added <a href="#">Section 5.10.8</a>.</li> <li>Added <a href="#">Section 5.10.13</a>.</li> </ul> <p><b>Section 5.11 SIFD:</b></p> <ul style="list-style-type: none"> <li>Updated main block diagram <a href="#">Section 5.11.1</a>.</li> <li>Updated the description of arbitrary channel number mapping in <a href="#">Section 5.11.4.5.7</a> (<a href="#">Errata 051</a>)</li> <li>Updated G.HAO/red/blue protocol terminology to G.7044/ LCR/BWR</li> <li>Updated the maximum number of Interlaken SERDES lanes supported in <a href="#">Section 5.11.4.5.3</a></li> <li>Added details about the support of the dynamic update of the CBRI channel lookup CAM for 2B header mode 0 described in <a href="#">Section 5.11.4.5.7</a> (<a href="#">Errata HWS-080</a>)</li> <li>Added note about the packet mode channel change error interrupt when CPB traffic is in segment mode and ODUk traffic is in packet mode to <a href="#">Section 5.11.2</a> and <a href="#">Section 5.11.4.2</a> (<a href="#">Errata HWS-077</a>)</li> <li>Updated <a href="#">Section 5.11.3.1</a> to describe Interlaken enhanced Scheduling support and link level flow control (<a href="#">Errata HWS-059</a>)</li> <li>Added <a href="#">Section 5.11.4.5.12</a> to sync with errata (<a href="#">Errata HWS-068</a>)</li> <li>Various edits for grammar/style</li> <li>References in this section to "SERDES_MUX" changed to "SIFD_MUX" throughout entire section (text and figures)</li> <li>References in this section to "DSIS" changed to "SYS_DSIS" throughout entire section (text and figures)</li> <li>A new section was added to describe latency management across an Interlaken based fabric (<a href="#">Section 5.11.4.5.5</a>)</li> <li>Various edits to correct ingress and egress fabric terminology including the <a href="#">Figure 106</a></li> <li>Updated <a href="#">Section 5.11.4.5.6</a> to include possible sources and destinations for packet traffic to/from the Interlaken (PIF) interface and the header formats</li> </ul>

Issue	Issue Date	Details of Change
		<ul style="list-style-type: none"> <li>Added a diagram to the SYS_DSIS section in the SIFD subsystem and also updated the associated paragraph to clarify its functionality (<a href="#">Section 5.11.6</a>).</li> <li>The following two sections were removed "DPI Internal Data Loopback" and "DPI External Data Loopback".</li> </ul> <p><b>Section 5.12 ENET(SYS):</b></p> <ul style="list-style-type: none"> <li>Added <a href="#">Section 5.12.9.2.2</a> to sync with errata (<a href="#">Errata HWS-069</a>)</li> <li>Correction to nominal size of an artificial packet in <a href="#">Section 5.12.3.2.2</a></li> <li>Correction made to <a href="#">Section 5.12.3.3.2</a></li> </ul> <p><b>Section 5.14 CBRC:</b></p> <ul style="list-style-type: none"> <li>Clarified OC-768 PRBS generation statement in <a href="#">Section 5.14.3.3.3</a></li> <li>Updated <a href="#">Section 5.14.3.3.6.1</a> to sync with errata (<a href="#">Errata FCN-036</a>)</li> <li>Updated <a href="#">Section 5.14.3.2</a></li> <li>Updated <a href="#">Section 5.14.3.3.4.2</a></li> <li>Updated <a href="#">Section 5.14.3.6</a></li> <li>Updated <a href="#">Section 5.14.3.3.2</a>, <a href="#">Section 5.14.3.3.5</a> and <a href="#">Section 5.14.3.3.7</a> to describe CBRC consequential actions</li> </ul> <p><b>Section 5.15 CPUP34K:</b></p> <ul style="list-style-type: none"> <li>Added <a href="#">Section 5.15.9.5</a> to sync with errata (<a href="#">Errata HWS-074</a>)</li> <li>Added <a href="#">Section 5.15.9.4.4</a> to sync with errata (<a href="#">Errata HWS-065</a>)</li> <li>Added note to <a href="#">Table 83</a></li> <li>Added <a href="#">Section 5.15.9.6</a> to sync with errata (<a href="#">Errata HWS-076 and Errata FCN-034</a>)</li> <li>Added <a href="#">Section 5.15.6.1</a> to sync with errata (<a href="#">Errata FCN-028</a>)</li> <li>Added <a href="#">Section 5.15.3.7</a> to sync with errata (<a href="#">Errata FCN-0037</a>)</li> <li>Added a footnote to <a href="#">Table 83</a></li> <li>Modified packet cache feature list in <a href="#">Section 5.15.8.2.2</a> to include hardware FCS calculation and insertion</li> <li>Added <a href="#">Section 5.15.5.4</a> to describe configuration bus lock-up possibilities</li> </ul> <p><b>Section 5.16 Diagnostic and Test Features:</b></p> <ul style="list-style-type: none"> <li>Updated <a href="#">Section 5.16.1.3.1</a> with additional restrictions to meet the S16 jitter specification to sync with errata (<a href="#">Errata HWS-070</a>)</li> <li>Several updates to the device loopback diagrams (<a href="#">Section 5.16.2</a>): <ul style="list-style-type: none"> <li>Removal of COPI loopbacks #20, #22, #24, and #26</li> <li>Removal of OTU Framer loopbacks in LINEOTN/SYSOTN #28 and 30</li> <li>Addition of loopback #33 in the SYSOTN</li> <li>A footnote was added to loopbacks #40, #41, #44, #47, and #50 stating that changes to the data rate may cause a disruption in the data stream</li> <li>Removal of loopback #11 in the LINE Interface SERDES as it was a duplicate of #51</li> <li>Addition of loopbacks #34 and #35 in the CBRC</li> <li>A footnote was added to loopbacks #2, #6, #8, #9, #10, #18, #19, #21, #23, #25, #27, #45, #46, #53, #54, and #55 stating that these loopbacks are for lab diagnostic use only</li> <li>A footnote was added to loopbacks #21, #23, #25, and #27 stating that the following pairs share their enables/disables (21,25) and (23,27)</li> <li>Loopback #13 was renamed</li> </ul> </li> <li>Added the PGMRCLK fractional divider limitations to <a href="#">Section 5.16.1.3</a></li> <li>Phase noise plot and jitter RMS Table added for the pgmrclk[3:0] outputs <a href="#">Section 5.16.1.3.6</a></li> <li>Removed DEMAPPED ENET sources from <a href="#">Section 5.16.1.3</a> and <a href="#">Table 86</a>.</li> <li>Updated <a href="#">Section 5.16.1.4</a></li> <li>Updated <a href="#">Section 5.16.1.6</a></li> <li>Updated # of registers in <a href="#">Section 5.16.1.5</a></li> </ul> <p><b>Section 5.17 Alarms and Consequential Actions:</b></p> <ul style="list-style-type: none"> <li>Added <a href="#">Section 5.17.1.2</a> for G.Sup43 7.3</li> <li>Updated CBRC diagrams in <a href="#">Figure 175</a> and <a href="#">Figure 176</a> to reflect "dMPMOUnderflow" instead of "dCPBUnderflow" to avoid confusion</li> </ul>

Issue	Issue Date	Details of Change
		<ul style="list-style-type: none"> <li>Updated CBRC Consequential Actions diagrams <a href="#">Figure 173</a>, <a href="#">Figure 174</a>, <a href="#">Figure 175</a>, <a href="#">Figure 176</a>, <a href="#">Figure 177</a> and <a href="#">Figure 178</a> to show that OPUk-CSF is asserted via firmware/interrupt and not hardware (<a href="#">Errata FCN-020</a>)</li> <li>Updated figure names and figures in <a href="#">Section 5.17.1</a>.</li> </ul> <p><b>Section 5.18 Device Latency:</b></p> <ul style="list-style-type: none"> <li>Added new latency details in <a href="#">Section 5.18.1</a>, <a href="#">Section 5.18.2</a>, <a href="#">Section 5.18.3</a>, <a href="#">Section 5.18.3.1</a> and <a href="#">Section 5.18.3.2</a></li> <li>Updated <a href="#">Table 107</a> and <a href="#">Table 108</a>: removed the 12x11.5 configuration, updated the 6x12.5 and 10x12.5 configurations, and added the 12x11.5 configuration (<a href="#">Errata FCN-031</a>)</li> <li>Removed all PCIe latency data in <a href="#">Section 5.18.5</a> and added a new summary in <a href="#">Table 110</a>. This new data is for 4x lanes only. Also added a comment indicating that this data is based off of simulation.</li> <li>Added new section that includes OTN FPI latencies <a href="#">Section 5.18.6</a></li> </ul> <p>Added new section <a href="#">Section 5.19</a> Hitless ODUFlex Resizing(<a href="#">Errata HWS-067</a>)</p> <p>Added new section <a href="#">Section 5.20</a> Enhanced PMON Modes</p> <p>Added new section <a href="#">Section 5.21</a> ODUK Switch and CPB Queue Sizing</p> <p>Added new section <a href="#">Section 6</a> Non-Compliances:</p> <ul style="list-style-type: none"> <li>Added <a href="#">Section 6.1.1</a> to sync with errata content (<a href="#">Errata FCN-012</a>)</li> <li>Added <a href="#">Section 6.1.2</a> to sync with errata content (<a href="#">Errata FCN-017</a>)</li> <li>Added <a href="#">Section 6.1.3</a> to sync with errata content (<a href="#">Errata FCN-039</a>)</li> <li>Added <a href="#">Section 6.1.4</a> to sync with errata content (<a href="#">Errata FCN-043</a>)</li> <li>Added <a href="#">Section 6.1.5</a> to sync with errata content (<a href="#">Errata FCN-047</a>)</li> <li>Added <a href="#">Section 6.1.6</a> to document restriction in monitoring GFP-F pFCS for demapped FC-1200</li> <li>Added <a href="#">Section 6.2.1</a></li> </ul> <p><b>Section 8 Functional Timing:</b></p> <ul style="list-style-type: none"> <li>Updated the device clock diagram <a href="#">Section 8.1</a>.</li> <li>Updated the OTN Frame Pulse Interface <a href="#">Section 8.3</a> to describe behavior during an OOF state. (<a href="#">Errata HWS-064</a>)</li> <li>Added <a href="#">Section 8.10</a></li> <li>Added note to <a href="#">Section 8.1</a></li> </ul> <p><b>Section 7 Operations:</b></p> <ul style="list-style-type: none"> <li>For channel mapping item "O", the description was changed to indicate that arbitrary lane re-ordering is supported and not just HO and LO lane reversal (<a href="#">Section 7.4</a>)</li> </ul> <p><b>Section 9 Power:</b></p> <ul style="list-style-type: none"> <li>Added <a href="#">Section 9.2.1</a> describing recommended power-up sequence for analog and digital VDD supplies (<a href="#">Errata HWS-055</a>)</li> <li>Added note that the 0.9V VDD analog supplies will collectively draw a large current if powered-up before the digital VDD 0.9V supply (<a href="#">Errata HWS-055</a>)</li> <li>Added <a href="#">Section 9.2.2</a> to describe the PCIe receiver Impedance dependency on power supply bring up (<a href="#">Errata HWS-084</a>)</li> <li>Updated <a href="#">Section 9.3</a></li> <li>Added new tables in <a href="#">Section 9.4</a></li> <li>Added note 5 to <a href="#">Section 9.1</a></li> </ul> <p><b>Section 10 Thermal:</b></p> <ul style="list-style-type: none"> <li>Updated the Junction-to-Case Thermal Resistance and Junction-to-Board Thermal Resistance parameters</li> </ul> <p><b>Section 13 DC Characteristics:</b></p> <ul style="list-style-type: none"> <li>Updated the TDO drive current (<a href="#">Errata HWS-071</a>)</li> <li>Updated the VOD minimum current value to 230 mV (<a href="#">Errata FCN-013</a>)</li> <li>Updated the Input Peak Differential Voltage in <a href="#">Table 129</a>(<a href="#">Errata FCN-030</a>)</li> </ul> <p><b>Section 14 High-Speed IO Characteristics:</b></p> <ul style="list-style-type: none"> <li>Updated <a href="#">Section 14.2.1</a>, <a href="#">Section 14.2.2</a>, to sync with errata describing LINE_LOS and DLOS use restrictions (<a href="#">Errata HWS-063</a>)</li> <li>Updated <a href="#">Table 131</a></li> </ul>

Issue	Issue Date	Details of Change
		<p><b>Section 15</b> AC Timing Characteristics:</p> <ul style="list-style-type: none"> <li>Changed duty cycle of SYS/LINE_REFCLK from 45-55% to 40-60% <a href="#">Table 140</a> and <a href="#">Table 141</a> (<a href="#">Errata HWS-028</a> and <a href="#">HWS-056</a>)</li> </ul> <p>Clarified the jitter specifications in <a href="#">Table 140</a> and <a href="#">Table 141</a> (<a href="#">Errata HWS-028</a> and <a href="#">HWS-056</a>)</p>
3	October, 2012	<p>Third issue of the DIGI 120G Hardware Specification.</p> <p>All sections:</p> <ul style="list-style-type: none"> <li>Priority Flow Control feature was removed.</li> </ul> <p><b>Section 2</b></p> <ul style="list-style-type: none"> <li>Updated DIGI 120G Documentation, Other PMC-Sierra References, and Standards References.</li> </ul> <p><b>Section 4</b></p> <ul style="list-style-type: none"> <li><b>Table 21</b> Power Management Pin Description: Changed power supply references to 0.875V and 0.925V.</li> </ul> <p><b>Section 5.6</b></p> <ul style="list-style-type: none"> <li><b>Section 5.6.8.6</b> OTU_FRM Maintenance Signal Insertion: This entire section was re-written.</li> <li><b>Section 5.6.8.7</b> OTU_FRM Overhead Insertion Capabilities: This entire section was added.</li> </ul> <p><b>Section 5.8</b></p> <ul style="list-style-type: none"> <li><b>Section 5.8.12.1</b> Remote Information Ring Control Port: This entire section was added.</li> </ul> <p><b>Section 5.16</b></p> <ul style="list-style-type: none"> <li><b>Section 5.16.1.3.1</b> S16 as Output Clock: This entire section was added.</li> </ul> <p><b>Section 7</b></p> <ul style="list-style-type: none"> <li>This entire section and its subsections were added for this revision.</li> </ul> <p><b>Section 8</b></p> <ul style="list-style-type: none"> <li><b>Section 8.6</b> SPI Master Interface: Removed sentence "The SPI_CLK is a 50 MHz clock derived from the CPU subsystem."</li> <li><b>Section 8.9</b> Ethernet Timing Modes: This entire section was added. Clocking diagrams updated.</li> </ul> <p><b>Section 9</b></p> <ul style="list-style-type: none"> <li><b>Table 118</b> Normal Operating Voltages: Changed digital supplies to 0.875V and 0.925V +/- 2.5%. Formerly 0.86V and 0.92V +/- 2%. Changed analog supplies to 0.875V and 0.925V +/- 2%. Formerly 0.86V and 0.92V +/- 2%.</li> </ul> <p><b>Section 13</b></p> <ul style="list-style-type: none"> <li><b>Table 126</b> DC Characteristics for 40nm 1.8V Drive LVC MOS: VIH(DC) min value changed from 0.65*VDDO_1v8 to 0.66*VDDO_1v8.</li> <li><b>Table 125</b> LVC MOS Interface Drive Strengths: OTU_TX_MFPO[11:0] and OTU_RX_MFPO[11:0] entries from 7mA to 15mA. SPI_SCLK, and SPI_MOSI entries from 15mA to 18mA.</li> </ul> <p><b>Section 14</b></p> <ul style="list-style-type: none"> <li><b>Section 14.2</b> SERDES LOS/DLOS Support: This entire section was added.</li> </ul> <p><b>Section 15</b></p> <ul style="list-style-type: none"> <li><b>Figure 215</b> PMON_SYNC_O AC Timing: removed tLO(pmon_sync_o) from diagram.</li> <li><b>Table 138</b> PMON_SYNC_O AC Timing Parameters: Removed tLO(pmon_sync_o) parameter. Removed PMON_SYNC_O Period Typical value and added min/max value.</li> <li><b>Table 133</b> REFCLK AC Timing Parameters: Added the following note "REFCLK Frequency Tolerance is +/- 20 ppm."</li> </ul>
2	August 2011	Document updated for second release.

Issue	Issue Date	Details of Change
1	April 2011	First issue of the DIGI 120G Hardware Specification.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:07 PM

## Table of Contents

1	Introduction .....	23
2	References .....	24
3	Definitions .....	27
4	Pinout .....	33
4.1	Device PCB Floor Planning Diagram .....	33
4.2	Pin Description .....	33
4.2.1	Device Interface .....	34
4.2.2	Line Interface .....	36
4.2.3	SFI-5.1 Interface .....	37
4.2.4	System Interface .....	39
4.2.5	Backplane Short Reach Interface .....	40
4.2.6	Backplane Long Reach Interface .....	41
4.2.7	Interlaken Flow Control Interface .....	42
4.2.8	OTN Ring Control Port Interface .....	43
4.2.9	OTN Overhead Interface .....	44
4.2.10	OTN Frame Pulse Interface .....	45
4.2.11	4-Wire OTN FEC Statistics Interface .....	46
4.2.12	Ethernet Management Interface .....	47
4.2.13	PCIe Interface .....	48
4.2.14	TOD and IRIG Interface .....	49
4.2.15	SPI Master Interface .....	51
4.2.16	JTAG Interface .....	52
4.2.17	Thermal Diode .....	53
4.2.18	Power and Ground Interface .....	54
4.3	Supported Electrical Interfaces and Clients .....	55
4.4	Supported Optical Modules .....	56
4.5	SERDES Lane Mapping .....	56
4.5.1	Line Side Interfaces Lane Mapping .....	56
4.5.2	System Side Interfaces Lane Mapping .....	58
4.5.3	Interlaken Over System Side Interfaces .....	59
5	Functional Description .....	60
5.1	Device Block Diagram .....	60
5.2	Device Overview .....	61
5.2.1	DIGI 120G Channelization .....	62
5.3	Internal Device Interfaces .....	63
5.3.1	Digital Serial Interface (DSI) .....	63
5.3.2	Digital CBR Interface (DCI) .....	63
5.3.3	Digital Packet Interface (DPI) .....	64
5.4	OTN Phase Signaling Algorithm (OPSA) .....	65
5.4.1	Overview .....	65
5.4.2	OTN Line-side Rate Capture from OTUk .....	67
5.4.3	OTN Line-side Rate Recovery to OTUk .....	69
5.4.4	Client Line / System-side Rate Capture to OTUk .....	71
5.4.5	GFP-F/TTT-mapped Client Line / System-side Rate Capture to OTUk .....	73
5.4.6	Client Line / System-side Rate Recovery from OTUk .....	74
5.4.7	GFP-F/TTT-mapped Client Line / System-side Rate Recovery from OTUk .....	77
5.4.8	Ethernet Tx MAC-sourced Client Rate Capture to OTUk .....	78
5.4.9	Ethernet Rx MAC Received Client Packet Recovery from OTUk .....	80
5.4.10	Support for Ethernet OEO Applications with Rx PMON .....	80

5.4.11	OTN Line-side Rate Capture to COPI .....	82
5.4.12	OTN Line-side Rate Recovery from COPI .....	84
5.4.13	Client Line / System-side Rate Capture to COPI .....	86
5.4.14	Client Line / System-side Rate Recovery from COPI .....	88
5.4.15	Multi-channel Packet Maker (MPMA) .....	91
5.4.16	Multi-channel Packet Monitor (MPMO) .....	92
5.4.17	Packet Maker Header .....	94
5.5	Line Interface (LIFD) .....	96
5.5.1	Block Diagram .....	96
5.5.2	SFI-5.1 Interface Multiplexing (LIFD_MUX) .....	96
5.5.3	Line Interface Multiplexing (LINE_DSIS) .....	97
5.6	Line OTN (LINEOTN) Subsystem .....	99
5.6.1	Block Diagram .....	99
5.6.2	Overview .....	99
5.6.3	Features .....	101
5.6.4	OTN_MLD Block .....	109
5.6.5	FEC Block .....	117
5.6.6	Shared RAM .....	119
5.6.7	FEC Selection .....	119
5.6.8	OTU_FRM Block .....	120
5.7	System OTN (SYSOTN) Subsystem .....	126
5.7.1	Block Diagram .....	126
5.7.2	Overview .....	126
5.7.3	Features .....	127
5.7.4	OPSM4/ SFIS Block .....	131
5.7.5	RSFEC .....	133
5.7.6	Framer .....	133
5.7.7	OTN Frame Pulse Interface .....	134
5.8	Core OTN (COREOTN) Subsystem .....	135
5.8.1	Block Diagram .....	135
5.8.2	Terminology .....	135
5.8.3	Overview .....	135
5.8.4	Features .....	140
5.8.5	ODUk Switch .....	160
5.8.6	ODUk Scheduler (ODUkSC) .....	166
5.8.7	ODUk Rx Framer (ODU_RFRM) .....	167
5.8.8	ODU Tx Framer (ODU_TFRM) .....	169
5.8.9	ODTUjk Demux (ODTU_DMX) .....	171
5.8.10	ODTUjk Mux (ODTU_MUX) .....	172
5.8.11	ODU Overhead Block (ODU_OHP) .....	173
5.8.12	ODU Ring Control Port Block (ODU_RCP) .....	181
5.9	OTN Mapper (MAPOTN) Subsystem .....	199
5.9.1	Block Diagram .....	199
5.9.2	Overview .....	199
5.9.3	Features .....	201
5.9.4	Top-level Interface Support Blocks .....	214
5.9.5	ODUk Scheduler (ODUkSC) .....	215
5.9.6	GFP-F Mapper .....	216
5.9.7	GFP-F De-mapper .....	222
5.9.8	GMP Mapper .....	232
5.9.9	GMP De-mapper .....	239
5.10	Central Packet Buffer (CPB) Subsystem .....	246
5.10.1	Overview .....	246
5.10.2	Slave DPI Block and Performance Counters .....	248
5.10.3	Sideband DPI "Type" .....	248
5.10.4	Ingress and Egress Mux .....	249

5.10.5	Channel Mapping .....	253
5.10.6	Scalable Central Packet Buffer (SCPB) .....	255
5.10.7	Broadcast of Packet Data .....	256
5.10.8	Memory Allocation .....	257
5.10.9	Zone Interface and Flow Control .....	257
5.10.10	Channel XOFF to FIFO Zone Cut-through .....	259
5.10.11	Dropping and Counting of Packets .....	259
5.10.12	Dynamic Resizing of Queues .....	259
5.10.13	Dynamic Remapping of Memory Allocation .....	259
5.10.14	Digital Common Scheduler and Pop Controller .....	260
5.11	System Interface (SIFD) .....	261
5.11.1	Block Diagram .....	261
5.11.2	Overview .....	261
5.11.3	Features .....	264
5.11.4	Combined OTN and Packet Interface (COPI) Block .....	270
5.11.5	SIFD_DFT Block .....	289
5.11.6	System Digital Serial Interface Switch (SYS_DSIS) .....	294
5.12	Ethernet Line-side (ENET(LINE)) Subsystem .....	295
5.12.1	Block Diagram .....	295
5.12.2	Overview .....	295
5.12.3	Features .....	299
5.12.4	Ethernet Ingress Management Packet Extract / Insert (EMPEI_I) .....	341
5.12.5	Ethernet Egress Management Packet Extract / Insert (EMPEI_E) .....	343
5.12.6	Ethernet Packet Classifier (ECLASS) .....	345
5.12.7	Ethernet Timing Shim (ETIME) .....	346
5.12.8	Ethernet Transcoder (ETRANS) .....	348
5.12.9	Ethernet PCS/MAC (EMAC) .....	352
5.12.10	Ethernet Packet Maker/Monitor (EPMM) .....	355
5.13	Ethernet System-side (ENET(SYS)) Subsystem .....	358
5.13.1	Block Diagram .....	358
5.13.2	Overview .....	359
5.13.3	Ethernet Ingress Management Packet Extract / Insert (EMPEI_I) .....	359
5.13.4	Ethernet Egress Management Packet Extract / Insert (EMPEI_E) .....	359
5.13.5	Ethernet Packet Classifier (ECLASS) .....	359
5.13.6	Ethernet Timing Shim (ETIME) .....	359
5.13.7	Ethernet PCS/MAC (EMAC) .....	359
5.13.8	MAC Assignments for 10GE/40GE/100GE Applications .....	359
5.13.9	Ethernet Packet Maker/Monitor (EPMM) .....	359
5.14	Constant Bit Rate Client (CBRC) Subsystem .....	360
5.14.1	Block Diagram .....	360
5.14.2	Overview .....	360
5.14.3	Features .....	363
5.15	CPU (CPUP34K) Subsystem .....	373
5.15.1	Block Diagram .....	373
5.15.2	Overview .....	373
5.15.3	Features .....	375
5.15.4	MIPS34K Processor .....	378
5.15.5	AMBA Network Interconnect .....	379
5.15.6	Interrupt Aggregation .....	381
5.15.7	Scratch Pad RAM (SPRAM) .....	385
5.15.8	CPU Packet Cache (PKT_CACHE) .....	386
5.15.9	PCIe Subsystem .....	388
5.15.10	PCIe PHY .....	393
5.15.11	Peripheral Configuration .....	394
5.16	Diagnostic and Test Features .....	395
5.16.1	PGMRCLK, LCLK, TIP, and FPI Windowing .....	395

5.16.2	Device Loopbacks .....	407
5.16.3	Test Pattern Generating and Monitoring .....	415
5.17	Alarms and Consequential Actions .....	417
5.17.1	ENET Consequential Actions Support .....	417
5.17.2	COREOTN Consequential Actions Support .....	425
5.17.3	LINEOTN Consequential Actions Support .....	425
5.17.4	SYSOTN Consequential Actions Support .....	425
5.17.5	MAPOTN Consequential Actions Support .....	425
5.17.6	CBRC Consequential Actions Support .....	425
5.17.7	CBRI Consequential Actions Support And Protection Switching .....	429
5.18	Device Latency .....	431
5.18.1	End to End Datapath Latency Examples .....	431
5.18.2	Subsystem Latency .....	431
5.18.3	ODUk Switch and CPB Latency .....	433
5.18.4	Interlaken Flow Control Latencies .....	436
5.18.5	PCIe Latencies .....	438
5.18.6	OTN Frame Pulse Interface Latencies .....	438
5.19	Hitless ODUflex Resizing (G.HAO – ITU-T G.7044) .....	440
5.19.1	Overview .....	440
5.19.2	Features .....	440
5.20	Enhanced PMON Modes .....	453
5.21	ODUk Switch and CPB Queue Sizing .....	454
5.21.1	ODUk Switch Queue Size Calculation .....	455
5.21.2	CPB Queue Size Calculation .....	456
6	Non-Compliances .....	459
6.1	OTN Non-Compliances .....	459
6.1.1	dCSF Is Not Declared After a Consistency Check .....	459
6.1.2	10G I.7 FEC Bit Errors When Unrelated 10G Channel Defects are Present .....	459
6.1.3	Adding a TCM or PM-NIM to A Signal in a Failure Condition will Result in SF Deasserting for One Frame .....	460
6.1.4	Early or Late Assertion of dDEG during dIAE for the OTUK_TT_Sk Function .....	460
6.1.5	Insertion and Extraction Points for APS Bytes are Not Compliant to the 12/2012 Version of G.798 .....	461
6.1.6	Fault Isolation and pFCS Handling for Transcoded and GFP Mapped FC-1200 .....	462
6.1.7	Update to Hardware DDEG Functionality .....	462
6.2	SONET/SDH Non-Compliances .....	463
6.2.1	Invalid RRMP LOS_V Bit .....	463
6.3	Ethernet Non-Compliances .....	463
6.3.1	High BER Status Bits of the 10GE MAC/PCS Blocks .....	463
7	Operations .....	465
7.1	SFI-5.1 SERDES Mapping .....	465
7.2	OH SERDES Mapping .....	466
7.3	LR SERDES Mapping .....	466
7.4	Channel Remapping .....	467
8	Functional Timing .....	470
8.1	Device Clock Diagram .....	470
8.2	Interlaken Flow Control Interface .....	470
8.3	OTN Frame Pulse Interface .....	470
8.4	4-Wire OTN FEC Statistics Interface .....	471
8.5	TOD and IRIG Interface .....	472
8.5.1	TOD Interface .....	472
8.5.2	IRIG Interface .....	473
8.6	SPI Master Interface .....	473
8.7	JTAG Interface .....	474
8.8	Power and Ground Interface .....	474

8.8.1	AVS Startup Sequence .....	474
8.9	Ethernet Timing Modes .....	474
8.9.1	Loop Timing .....	475
8.9.2	Through Timing .....	475
8.9.3	External Timing .....	475
8.9.4	Line Timing .....	475
8.10	PMON Latch Time .....	476
9	Power .....	477
9.1	Normal Operating Voltages .....	477
9.2	Power Supply Sequencing Requirements .....	478
9.2.1	AVD Current is Large when VDD Is Powered Off .....	478
9.2.2	PCIe Receiver Impedance Dependency on Power Supply Bring Sequencing .....	478
9.3	Power Supply Ramp Time .....	478
9.4	Power Requirements .....	478
9.5	Conditions for Power Requirements .....	489
10	Thermal Information .....	490
10.1	Leakage versus Temperature Curve .....	491
11	Package Specification .....	492
12	Absolute Maximum Ratings .....	493
13	DC Characteristics .....	494
13.1	LVCMOS DC Characteristics .....	494
13.2	LVDS DC Characteristics .....	495
14	High Speed I/O Characteristics .....	497
14.1	Multi-lane Interface Skew .....	497
14.2	SERDES LOS/DLOS Support .....	498
14.2.1	Line Side LOS/DLOS Support .....	498
14.2.2	System Side LOS/DLOS used for Line Side Support .....	499
14.2.3	System Side LOS/DLOS Support .....	499
14.3	SERDES DLOS Characteristics .....	500
15	AC Timing Characteristics .....	502
15.1	Device Interface .....	502
15.1.1	REFCLK .....	502
15.1.2	RSTB .....	502
15.1.3	PGMRCLK [3:0] .....	502
15.1.4	PLL_STATUS [1:0] .....	503
15.1.5	PMON_SYNC_I .....	503
15.1.6	PMON_SYNC_O .....	504
15.1.7	FPI .....	504
15.2	Line Interface .....	505
15.2.1	LINE_REFCLK_P/N[11:0] AC Timing Characteristics .....	505
15.3	System Interface .....	505
15.3.1	SYS_REFCLK_P/N[11:0] AC Timing Characteristics .....	505
15.4	Interlaken Flow Control Interface .....	507
15.5	OTN Frame Pulse Interface .....	508
15.6	4-Wire OTN FEC Statistics Interface .....	509
15.7	TOD and IRIG Interface .....	510
15.7.1	TOD Interface .....	510
15.7.2	IRIG-B B00x Interface .....	510
15.8	SPI Master Interface .....	511
15.9	JTAG Interface .....	512
16	End of Document .....	513

## List of Figures

Figure 1	Device PCB Floor Planning Diagram .....	33
Figure 2	PM5440 Device Block Diagram .....	60
Figure 3	Line-side Rate Capture from OTUk and Rate Recovery to OTUk .....	67
Figure 4	Line and System-side Client Rate Capture to OTUk .....	71
Figure 5	Line and System-side Client Rate Recovery from OTUk .....	74
Figure 6	Line-side Tx MAC-sourced Client Rate Capture to OTUk .....	78
Figure 7	Ethernet Rx MAC Received Client Packet Recovery from OTUk .....	80
Figure 8	Line-side Rate Capture from OTUk to COPI .....	82
Figure 9	Line-side Rate Capture from COPI to OTUk .....	84
Figure 10	Line and System-side Client Rate Capture to COPI .....	86
Figure 11	Line and System-side Client Rate Recovery from COPI .....	88
Figure 12	Packet Maker Header Format .....	94
Figure 13	LIFD Block Diagram .....	96
Figure 14	PM5440 LINEOTN Block Diagram .....	99
Figure 15	Atomic Functions Supported By OPSMnk (OTL) Datapaths .....	101
Figure 16	Atomic Functions Supported By Non-OPSMnk (Non-OTL) Datapaths .....	101
Figure 17	LINEOTN Consequential Actions Processing .....	108
Figure 18	OTU_TX_MFPO Timing Diagram .....	111
Figure 19	OTU_RX_MFPO Timing Diagram .....	112
Figure 20	DPSK Precoder Description .....	114
Figure 21	OTU RX and TX MFPO Muxing .....	115
Figure 22	OTU_FRM Datapath Diagram .....	120
Figure 23	SYSOTN Block Diagram .....	126
Figure 24	SYSOTN Consequential Actions Processing .....	131
Figure 25	Multi lane Distribution Block Diagram .....	132
Figure 26	COREOTN Block Diagram .....	135
Figure 27	COREOTN Line Interface Receive Datapath .....	137
Figure 28	COREOTN Line Interface Transmit Datapath .....	137
Figure 29	COREOTN Block Diagram - Without SYSOTN Interface .....	138
Figure 30	COREOTN Block Diagram with SYSOTN Configuration Mode .....	139
Figure 31	OTN ODUK Frame Structure .....	140
Figure 32	Multi-channel Bus .....	143
Figure 33	COREOTN ODUK/OPUK Layer Processing Functions .....	145
Figure 34	SYSOTN Atomic Functions .....	145
Figure 35	RFRM Atomic Function .....	147
Figure 36	TFRM Atomic Function .....	149
Figure 37	Demux Atomic Function .....	150
Figure 38	Mux Atomic Function .....	151
Figure 39	COREOTN Muxing/Demuxing Support .....	152
Figure 40	ODUKT_TT Sink Function Implementation .....	153
Figure 41	ODUKP_TT Sink and ODU[i][j] AP Function Implementation .....	153
Figure 42	OTN Overhead Insert and Extraction .....	155
Figure 43	Delay Measurement Diagram .....	157
Figure 44	Left Justified Data Processing .....	158
Figure 45	ODUK Switch Functional Diagram .....	160
Figure 46	ODUK Switch Capacity .....	161
Figure 47	Page Allocation to Port Pairs .....	162
Figure 48	Broadcast and Multicast .....	163
Figure 49	FIFO Zone Indication .....	165
Figure 50	ODU_RFRM Functional Diagram .....	167
Figure 51	ODU_TFRM Functional Diagram .....	169
Figure 52	ODTU_DMX Functional Diagram .....	171
Figure 53	ODTU_MUX Functional Diagram .....	172

Figure 54	ODU_OHP Functional Block Diagram .....	173
Figure 55	OTN Channel Overhead Data Frame .....	174
Figure 56	Ethernet Packet for Extracted Overhead Data .....	176
Figure 57	Request-Only Packet Structure .....	178
Figure 58	Overhead Data Insertion Frame .....	178
Figure 59	Example Showing Response Time Bandwidth .....	180
Figure 60	RCP Top Level Block Diagram .....	181
Figure 61	RI_RCP Top Level Block Diagram .....	183
Figure 62	RI_RCP Ethernet Frame Format .....	186
Figure 63	MAC Address Encoding .....	187
Figure 64	DI_RCP Top Level Block Diagram .....	188
Figure 65	DI_RCP Ethernet Frame Format .....	189
Figure 66	OTU and Channel Based HO ODUk PDU .....	190
Figure 67	Pool-Based HO ODUk PDU .....	190
Figure 68	Channel-Based ODUk PDU .....	191
Figure 69	Pool-Based ODUk PDU .....	191
Figure 70	GFP-F Mapper/Demapper PDU .....	192
Figure 71	Client Defects PDU .....	193
Figure 72	HO ODUk Stage 1 & N Performance Monitoring PDU .....	194
Figure 73	Pool-Based ODU Stage 1 & N Performance Monitoring PDU .....	194
Figure 74	Channel Based ODU Stage 2/3a/3b/4 Performance Monitoring PDU .....	195
Figure 75	Pool Based ODU Stage 2/3a/3b/4 Performance Monitoring PDU .....	195
Figure 76	SONET/SDH Performance Monitoring PDU .....	196
Figure 77	PRBS Performance Monitoring PDU .....	196
Figure 78	MAPOTN Block Diagram .....	199
Figure 79	MAPOTN Atomic Functions .....	203
Figure 80	GFP-F Mapper Block Diagram .....	216
Figure 81	UPI Byte Insertion in the Type Header .....	221
Figure 82	GFP-F De-mapper Block Diagram .....	222
Figure 83	PAUSE Frame Detection and Its Effect .....	227
Figure 84	GMP Mapper Block Diagram .....	232
Figure 85	Cn Selection for JC Byte Calculation .....	235
Figure 86	OPUk Frame Structure for AMP Mapping .....	235
Figure 87	GMP Data/Stuff Spread Pattern .....	236
Figure 88	OPUk Structure for GMP Mapping .....	237
Figure 89	Handling a Signal Failure .....	238
Figure 90	GMP De-mapper Block Diagram .....	239
Figure 91	OPUk Frame Structure for AMP/BMP Mapping .....	240
Figure 92	ODU2/ODU2e/ODU2f Fixed Stuff Column (AMP/BMP) Modes .....	242
Figure 93	ODU3 Fixed Stuff Columns (AMP/BMP) Modes .....	242
Figure 94	ODU4 Fixed Stuff Column for GMP .....	243
Figure 95	CPB Block Diagram .....	246
Figure 96	CPB Queue Linked List of Page Blocks .....	247
Figure 97	SCPB Port Data Routing Options .....	250
Figure 98	CPB Ingress Channel Mapping .....	254
Figure 99	CPB Egress Channel Mapping .....	255
Figure 100	Broadcast .....	257
Figure 101	FIFO Zone Indication .....	258
Figure 102	SIFD Block Diagram .....	261
Figure 103	Interlaken Internal Loopback/Flow Control Tx-Rx Loopback .....	267
Figure 104	Interlaken External Loopback/Flow Control Rx-Tx Loopback .....	268
Figure 105	COPI Block Diagram .....	270
Figure 106	Interlaken Block Diagram .....	273
Figure 107	Definition of the Virtual FIFO: Transmit Device -> Interlaken -> Receive Device .....	276
Figure 108	Sizing Relationship Between the Virtual FIFO and the Receive Device ODUk Switch Buffer .....	277
Figure 109	PIF Format Over Interlaken .....	278

Figure 110	CBRI Packet Format Over Interlaken .....	281
Figure 111	Transmit Flow Control Arbitrary Channel Mapping .....	282
Figure 112	Receive Flow Control Arbitrary Channel Mapping .....	283
Figure 113	Arbitrary Channel Map and 6B Insertion Block .....	284
Figure 114	Tx Channel Map & 6B Insertion Block Flow Chart .....	285
Figure 115	Arbitrary Channel Map and Header Extraction Block .....	286
Figure 116	Rx Arbitrary Channel Map Flow Chart .....	287
Figure 117	SIFD_DFT Block Diagram .....	289
Figure 118	PM5440 SIFD_MUX System-side SERDES Muxing .....	292
Figure 119	SYS_DSIS Block Diagram .....	294
Figure 120	ENET(LINE) Block Diagram .....	295
Figure 121	The Termination Mode Ingress Data Path .....	299
Figure 122	The Transparent Mode Ingress Data Path .....	301
Figure 123	The Termination Mode Egress Data Path .....	303
Figure 124	The Egress Data Path Through the Packet Monitor .....	304
Figure 125	Ethernet Client Data Frame Mapping into GFP-F .....	306
Figure 126	Ethernet Client Data Frame and Preamble Mapping into GFP-F .....	307
Figure 127	Ethernet Ordered Set Mapping into GFP-F .....	308
Figure 128	Bit Transparent Mapping of a 10GBASE-R Signal into OPU2e .....	308
Figure 129	64/66B Block to 512B/513B Block Conversion .....	309
Figure 130	512B/513B to 1024B/1027B Block Construction .....	310
Figure 131	100GE Transmission Order .....	311
Figure 132	Pause Timer State Machine .....	313
Figure 133	Remote Link Pause Control Information Flow .....	314
Figure 134	ENET Management Paths to/from MGMT_GE .....	318
Figure 135	ENET Management Paths to/from CPU PKT_CACHE .....	318
Figure 136	Management Paths between CPU PKT_CACHE and MGMT_GE .....	319
Figure 137	Frame format through Header Insertion Block .....	321
Figure 138	Frame Format through Egress Header Extraction block .....	322
Figure 139	Frame format through Ingress Header Extraction Block .....	322
Figure 140	Generation of TOD from the FRC .....	328
Figure 141	Time Stamping on the Receive Path .....	329
Figure 142	Data Path for PTP in One-step Mode .....	330
Figure 143	Data Path for PTP in Two-step Mode .....	331
Figure 144	Signal Fail Condition in Ethernet Termination Mode with SERDES Interface .....	336
Figure 145	ENET Loopbacks .....	340
Figure 146	Functional Block Diagram of the EMPEI Ingress .....	341
Figure 147	Functional Block Diagram of EMPEI Egress .....	343
Figure 148	Functional Block Diagram of ECLASS .....	345
Figure 149	ETIME TOD Block Diagram .....	346
Figure 150	Top Level Block Diagram of Ingress ETRANS .....	348
Figure 151	Top Level Block Diagram of Egress ETRANS .....	348
Figure 152	EMAC120 High-Level Block Diagram .....	352
Figure 153	EPMM Ingress Block Diagram .....	355
Figure 154	EPMM Egress Block Diagram .....	355
Figure 155	ENET(SYS) Block Diagram .....	358
Figure 156	CBRC Block Diagram .....	360
Figure 157	CPUP34K Block Diagram .....	373
Figure 158	CPUP34K AMBA Network Interconnect Block Diagram .....	379
Figure 159	Interrupt Aggregation .....	383
Figure 160	PKT_CACHE Block Diagram .....	386
Figure 161	PCIe Subsystem Block Diagram .....	389
Figure 162	PGMRCLK Block Diagram .....	395
Figure 163	Typical Phasenoise Plot For 2.048MHz Recovered Clock on PGMRCLK[3:0] Outputs .....	402
Figure 164	Device Internal Loopbacks .....	408
Figure 165	Device Line Loopbacks .....	411

Figure 166	Device Diagnostic Loopbacks .....	413
Figure 167	Ethernet as GFP-F Atomic Functions (G.Sup43-6.2 / G.7041-7.1) .....	418
Figure 168	Ethernet as GFP-F Waterfall Diagram (G.Sup43-6.2 / G.709-17.4 / G.7041-7.1) .....	419
Figure 169	Ethernet as PP-OS Atomic Functions(G.Sup43-7.3) .....	421
Figure 170	Ethernet as PP-OS Waterfall Diagram (G.Sup43-7.3 / G.709-17.4.1 / G.7041-7.9 / G.8021-11.5.3) .....	422
Figure 171	Ethernet as CBR Atomic Functions (G.Sup43-7.1/7.2) .....	424
Figure 172	Ethernet as CBR Waterfall Diagram (G.Sup43-7.1 / G.709-17.2.4) .....	424
Figure 173	FC-800 and Generic 8b/10b Clients .....	426
Figure 174	Consequential Actions: FC-800 and 8b/10b Clients .....	426
Figure 175	SONET/SDH Clients .....	427
Figure 176	Consequential Actions: SONET/SDH Clients .....	428
Figure 177	FC-1200 Clients .....	429
Figure 178	Consequential Actions: FC-1200 Clients .....	429
Figure 179	Hitless ODUflex Resizing Block Diagram .....	440
Figure 180	HO ODUflex RCOH .....	441
Figure 181	LO ODUflex RCOH .....	442
Figure 182	G.HAO Bandwidth Increase Protocol .....	444
Figure 183	G.HAO Bandwidth Decrease Protocol .....	446
Figure 184	LINE_CORE_MAP Datapath .....	448
Figure 185	MAP_CORE_SIFD Datapath .....	449
Figure 186	LINE_CORE_LINE Datapath .....	450
Figure 187	LINE_CORE_SIFD Datapath .....	451
Figure 188	Queue XON, XOFF, and Hysteresis Regions .....	454
Figure 189	ODUk Switch Queue Size MAPOTN or COREOTN to/from ILKN .....	455
Figure 190	ODUk Switch Queue Size MAPTON to/from COREOTN .....	456
Figure 191	CPB Queue Size CPB to/from ILKN .....	456
Figure 192	CPB Queue Size ENET to/from CPB .....	457
Figure 193	CPB Queue Size CPB to/from MAPOTN .....	457
Figure 194	CPB Queue Size CBRC to/from CPB .....	457
Figure 195	CPB Queue Size ENET to/from CPB to/from ILKN .....	458
Figure 196	CPB Queue Size MAPOTN to/from CPB to/from ILKN .....	458
Figure 197	Channel Remapping .....	468
Figure 198	PM5440 Top Level Device Clock Diagram .....	470
Figure 199	OTN Frame Pulse Interface Functional Timing .....	471
Figure 200	FEC Statistics Logic Table (FSL-T) .....	471
Figure 201	FSL-T Timing .....	471
Figure 202	TOD Input Interface Functional Timing .....	472
Figure 203	TOD Output Interface Functional Timing .....	473
Figure 204	SPI Master Connected to Multiple Slaves .....	473
Figure 205	SPI Interface 24-bit Address Functional Timing .....	474
Figure 206	Ethernet Line Timing Application .....	475
Figure 207	DIGI 120G Typical Leakage Scaling Factor VS. Temperature .....	491
Figure 208	Device Package Specifications .....	492
Figure 209	LVDS Transmit DC Characteristics .....	495
Figure 210	REFCLK AC Timing (Input) .....	502
Figure 211	RSTB AC Timing (Input) .....	502
Figure 212	PGMRCLK[3:0] AC Timing (Output) .....	503
Figure 213	PLL_STATUS[1:0] AC Timing (Output) .....	503
Figure 214	PMON_SYNC_I AC Timing (Input) .....	504
Figure 215	PMON_SYNC_O AC Timing (Output) .....	504
Figure 216	FPI AC Timing (Input) .....	505
Figure 217	Interlaken Mode OOB Flow Control Interface AC Timing .....	507
Figure 218	4-Wire OTN FEC Statistics AC Timing .....	509
Figure 219	SPI Master Interface AC Timing .....	511
Figure 220	JTAG AC Timing .....	512

## List of Tables

Table 1	Acronyms And Terms Used Within This Document .....	27
Table 2	Device I/O Definition .....	33
Table 3	Device Interface Pin Description .....	34
Table 4	Line Interface Pin Description .....	36
Table 5	SFI-5.1 Interface Pin Description .....	37
Table 6	System Interface Pin Description .....	39
Table 7	Backplane Short Reach Interface Pin Description .....	40
Table 8	Backplane Long Reach Interface Pin Description .....	41
Table 9	Interlaken Flow Control Interface Pin Description .....	42
Table 10	OTN Ring Control Port Interface Pin Description .....	43
Table 11	OTN Overhead Interface Pin Description .....	44
Table 12	OTN Frame Pulse Interface Pin Description .....	45
Table 13	OTN FEC Statistics Interface Pin Description .....	46
Table 14	Ethernet Management Interface Pin Description .....	47
Table 15	PCIe Interface Pin Description .....	48
Table 16	TOD and IRIG Interfaces Pin Description .....	49
Table 17	SPI Master Interface Pin Description .....	51
Table 18	JTAG Interface Pin Description .....	52
Table 19	Thermal Diode Pin Description .....	53
Table 20	Power and Ground Pin Description .....	54
Table 21	Power Management Pin Description .....	54
Table 22	Supported Electrical Interfaces and Client Protocols .....	55
Table 23	Interlaken Lane Mapping Summary .....	59
Table 24	DIGI 120G Channelization .....	62
Table 25	Packet Maker Header Fields .....	94
Table 26	OFP CSI Field Definition .....	95
Table 27	LINE_DSIS Internal Lane Sources .....	97
Table 28	LINEOTN MLD Datapaths .....	102
Table 29	Statistics Per FEC .....	103
Table 30	Supported 4-Wire Interface Modes .....	104
Table 31	Event Processing in the OPSM23/4 Core .....	105
Table 32	Event Processing in the Data Processor Core .....	106
Table 33	Physical Lane to Logical Lane Channel Mapping .....	123
Table 34	SM Overhead Insertion .....	124
Table 35	FEC Statistics .....	128
Table 36	Event Processing in the OPSM4 .....	129
Table 37	Event Processing in the OTU_FRM Core .....	129
Table 38	Calendar Configuration on the Multi-channel Bus .....	143
Table 39	DPI Type Usage in ODUK Switch .....	165
Table 40	Overhead Byte Ordering .....	175
Table 41	Channel Offset Values for Insertion Block .....	177
Table 42	RI_RCP Ethernet Bandwidth Calculation .....	187
Table 43	SONET/SDH and OTN Performance Monitoring PDU Types .....	193
Table 44	DI_RCP Ethernet Bandwidth Calculation - Defects .....	196
Table 45	DI_RCP Ethernet Bandwidth Calculation - SONET/SDH B1 & B2 & M0/M1, OTN NEBC & FEBC Error Counts .....	197
Table 46	Supported GFP Mappings .....	201
Table 47	Supported CBR (AMP/BMP/GMP) Mappings .....	201
Table 48	Interlaken Protocol and Device Interlaken Processor Throughput Versus Number of Bytes .....	204
Table 49	OPUK Payload Capacity .....	212
Table 50	Baseline Small Packet Sustained Throughput for the GFP Mapper and Demapper .....	213
Table 51	GFP De-mapper PMON Status Counters .....	226
Table 52	Cm(t) Increment and Decrement Indicator Patterns .....	238

Table 53	JC, NJO and PJO Interpretation .....	241
Table 54	Cm(t) Increment and Decrement Indicator Patterns .....	244
Table 55	DPI_TYPE Use .....	249
Table 56	DCS Applications .....	252
Table 57	Interlaken Channel Mapping at the COPI Interface Level .....	274
Table 58	Channel Information Details .....	275
Table 59	Ingress Packet Traffic (Device to Network Processor) .....	278
Table 60	Egress Packet Traffic (Network Processor to Device) .....	279
Table 61	Header Insertion Operating Modes .....	283
Table 62	SERDES Usage Per Interface .....	291
Table 63	System-side SERDES CSU Information .....	293
Table 64	ENET(LINE) Subsystem Block Summary .....	296
Table 65	MAC Link Pause Control Frame .....	312
Table 66	Pause Reaction .....	315
Table 67	Insert / Extract Queue to Port Mapping .....	323
Table 68	RX MAC Stats: IEEE 802.3 oMacEntity and oPauseEntity Managed Object .....	324
Table 69	RX MAC Stats: IETF MIB (MIB-II) Objects .....	324
Table 70	RX MAC Stats: IETF RMON MIB Objects .....	325
Table 71	TX MAC Stats: IEEE 802.3 oMacEntity and oPauseEntity Managed Object .....	326
Table 72	TX MAC Stats: IETF MIB (MIB-II) Objects .....	326
Table 73	PTP Over Ethernet Frame Format .....	332
Table 74	SONET/SDH Ingress Client Stream Consequential Actions .....	365
Table 75	SONET/SDH Egress Client Stream Consequential Actions .....	365
Table 76	8B/10B Client Stream Consequential Actions .....	367
Table 77	FC-1200 Ingress Client Stream Consequential Actions .....	369
Table 78	FC-1200 Egress Client Stream Consequential Actions .....	370
Table 79	CBRC Internal DS1 to SERDES DS1 Channel Mappings .....	372
Table 80	Address View Via AMBA Interconnect .....	379
Table 81	Permanent Address Aliasing .....	380
Table 82	CPUP34K's External Interrupt Destination .....	381
Table 83	CPUP34K's External Interrupt Input Source Map In DIGI 120G .....	381
Table 84	Summary of 32:1 Interrupt Aggregation Towards PCIe Host .....	382
Table 85	32:1 Interrupt Aggregation Towards Device Interrupt Output Pin .....	382
Table 86	CLK_MUX_SEL[8:0] Values for Recovered Clock Selection .....	398
Table 87	SERDES Client Core Clock Frequencies .....	399
Table 88	Divider CFG for 2.048 MHz Recovered CLK .....	400
Table 89	Divider CFG for 8 KHz Recovered CLK .....	401
Table 90	Typical JitterRMS For 2.048MHz Recovered Clock on PGMRCLK[3:0] Outputs .....	403
Table 91	Typical TIE Jitter For 2.048MHz Demapped ENET Recovered Clock on PGMRCLK[3:0] Outputs .....	403
Table 92	Device Subsystem LCLK Connections .....	404
Table 93	Device Subsystem TIP Connections .....	405
Table 94	Device Internal Loopback Descriptions .....	409
Table 95	Device Line Loopback Descriptions .....	412
Table 96	Device Diagnostic Loopback Descriptions .....	414
Table 97	Unframed Pattern Generation Supported on SERDES Device Interfaces .....	415
Table 98	End to End Datapath Round Trip Latency Examples .....	431
Table 99	FEC Latency .....	432
Table 100	Datapath Segment Latency .....	432
Table 101	ODUk Switch Egress Latency From Interlaken .....	434
Table 102	Packet Interface (Interlaken) To / From MAPOTN GFP Mapper / Demapper CPB .....	434
Table 103	MAC Mapping / Demapping Paths (MAC to TGFP-F or RGFP-F to MAC) .....	435
Table 104	BMP/GMP Mapped Ethernet Latencies (Ethernet subsystems to or from CPB) .....	435
Table 105	GFP-F Mapped FC-1200 and BMP/GMP Mapped CBR Latencies (CBRC Subsystem To or From CPB) .....	436
Table 106	Interlaken Flow Control Timing Parameter Definitions .....	436
Table 107	Interlaken In-band Channel Level Flow Control Latency .....	436

Table 108	Interlaken Out-of-band Channel Level Flow Control Latency .....	437
Table 109	Interlaken In-band Link Level Flow Control Latency .....	437
Table 110	Summary of PCIe 4x Lane Latencies .....	438
Table 111	OTN Frame Pulse Interface Latencies and Uncertainties .....	439
Table 112	Datapath Support .....	440
Table 113	SERDES Mapping .....	465
Table 114	OH SERDES Mapping .....	466
Table 115	LR SERDES Mapping .....	466
Table 116	Core Clock Frequencies for OTN Clients .....	471
Table 117	FSL-T Description .....	472
Table 118	Normal Operating Voltages .....	477
Table 119	Power Requirements - 100G Hybrid Application .....	479
Table 120	Incremental Power Requirements .....	480
Table 121	Analog Filter Current Requirements .....	488
Table 122	Conditions for Power Requirements .....	489
Table 123	Device Thermal Information .....	490
Table 124	Device Absolute Maximum Ratings .....	493
Table 125	LVC MOS Interface Drive Strengths .....	494
Table 126	DC Characteristics for 1.8V Drive LVC MOS .....	494
Table 127	LVDS Pin List .....	495
Table 128	TRANSMIT DC Characteristics for 1.8V LVDS .....	495
Table 129	RECEIVE DC Characteristics for 1.8V LVDS .....	496
Table 130	Maximum Transmit Skew and Receive Skew Tolerance Between SERDES Lanes .....	498
Table 131	STL256.4 Transmit and Receive Skew Tolerance .....	498
Table 132	SERDES DLOS Characteristics .....	500
Table 133	REFCLK AC Timing Parameters (Input) .....	502
Table 134	RSTB AC Timing Parameters (Input) .....	502
Table 135	PGMRCLK[3:0] AC Timing Parameters (Output) .....	503
Table 136	PLL_STATUS[1:0] AC Timing Parameters (Output) .....	503
Table 137	PMON_SYNC_I AC Timing Parameters (Input) .....	504
Table 138	PMON_SYNC_O AC Timing Parameters (Output) .....	504
Table 139	FPI AC Timing Parameters (Input) .....	505
Table 140	LINE_REFCLK_P/N AC Timing Parameters .....	505
Table 141	SYS_REFCLK_P/N AC Timing Parameters .....	506
Table 142	Interlaken Mode OOB Flow Control Interface AC Timing Parameters .....	507
Table 143	OTN Frame Pulse Interface AC Timing Parameters .....	508
Table 144	4-Wire OTN FEC Statistics AC Timing Parameters .....	509
Table 145	TOD Interface AC Timing Parameters .....	510
Table 146	IRIG-B B00x Interface AC Timing Parameters .....	510
Table 147	SPI Master Interface AC Timing Parameters .....	511
Table 148	JTAG Port Interface AC Timing Parameters .....	512

## 1

## Introduction

This document defines the design parameters of the PM5440 DIGI 120G. It provides essential reading for system designers requiring a detailed understanding of this device's functions, physical characteristics, and interfaces. This document is intended to be read in conjunction with the following:

- The Product Overview. This document provides a high level overview of the device and provides system application examples [1].
- The Register Descriptions. This document provides specific details of all device configuration and status registers [2].
- The Hardware Design Guide. This document provides a high level overview of the device and provides system application examples [3].
- The Programmer's Manual. This document provides information required to develop control plane applications on a selected Host CPU [4].

Downloaded [controlled] by Benson Zhang of Danriver  
Thursday, 21 April, 2016 09:05:07 PM

## 2

# References

## Device Documentation

1. PMC-2091566, *PMC-Sierra, "DIGI 120G Product Overview"*, Issue 4.
2. PMC-2113758, *PMC-Sierra, "DIGI 120G Register Document"*, Issue 4.
3. PMC-2114167, *PMC-Sierra, "PM5440 DIGI 120G Hardware Design Guide"*, Issue 3.
4. PMC-2120045, *PMC-Sierra, "DIGI Device Family Programmer's Manual"*.
5. PMC-2131789, *PMC-Sierra, "CBRI Utilization Calculator"*, Issue 1.
6. PMC-2112680, *PMC-Sierra, "PM5440 DIGI 120G Ball Map"*, Issue 3.

## Other PMC-Sierra References

7. PMC-2100078, *PMC-Sierra, "40nm RAPTR SERDES Top Level Specifications for DIGI Family of Devices"*, Issue 6.
8. PMC-2040933, *PMC-Sierra, "Moisture Sensitivity Levels and Recommended Reflow Profiles for PMC-Sierra Devices"*, Issue 2.
9. PMC-2020246, *PMC-Sierra, "Attaching Passive Heat Sinks to Organic Flip Chip Packages"*, Issue 3.
10. PMC-2101547, *PMC-Sierra, "Precision Timing Protocol (PTP) Implementation on PMC Devices"*, Issue 1.
11. PMC-2124310, *PMC-Sierra, "PM5440 External Interrupt Hierarchy"*, Issue 1.
12. PMC-2131471, *PMC-Sierra, "PM5440 DIGI 120G ODUk Switch Buffer Size Calculator"*, Issue 1.
13. PMC-2131472, *PMC-Sierra, "PM5440 CPB Buffer Size Calculator"*, Issue 1.

## Standards References

14. ANSI INCITS 470, *"Fibre Channel Framing and Signalling - (FC-FS-3)"*, Revision 1.10, July 2010.
15. ANSI INCITS 364, *"Fibre Channel 10 Gigabit (10GFC)"*, Revision 4.0, April 2004.
16. ARM, *"AMBA 3 AHB-Lite Protocol Specification"*, Version 1.0, June 2006.
17. ARM, *"CoreLink™ Network Interconnect NIC-301 Revision: r2p2 Technical Reference Manual"*, Revision r2p2, August 2010.
18. Telcordia Technologies, *"Network Equipment-Building System (NEBS) Requirements: Physical Protection: Telcordia Technologies Generic Requirements GR-63-CORE"*, Issue 1, October 1995.
19. Electronic Industries Alliance, *"Integrated Circuit Thermal Test Method Environmental Conditions - Junction-to-Board: JESD51-8"*, October 1999.
20. IEEE 1149.1, *"IEEE Standard Test Access Port and Boundary Scan"*, March 2008.
21. IEEE 1149.6, *"IEEE Standard for Boundary-Scan Testing of Advanced Digital Networks"*, April 2003.
22. IEEE 1588, *"IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems"*, July 2008.
23. IEEE 802.3, *"IEEE Standard for Local and Metropolitan Area Networks: Carrier sense multiple access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications"*, December 2008.
24. IEEE P802.3ba, *"IEEE Standard for Local and Metropolitan Area Networks: Carrier sense multiple access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications – Media Access Control Parameters, Physical Layers and Management Parameters for 40Gb/s and 100 Gb/s Operation"*, June 2010.
25. Interlaken Alliance, *"Interlaken Protocol Definition"*, Revision 1.2, October 2008.
26. Interlaken Alliance, *"Interlaken Interoperability Recommendations"*, Revision 1.6, October 2011.

27. ITU-T Recommendation Series G Supplement 43, "*Transport of IEEE 10GBASE-R in optical transport networks (OTN)*", June 2010.
28. ITU-T Recommendation G.709/Y.1331, "*Interfaces for the Optical Transport Network (OTN)*", February 2012.
29. ITU-T Recommendation G.709/Y.1331, "*Interfaces for the Optical Transport Network (OTN): Erratum I*", May 2010.
30. ITU-T Amendment 1 to Recommendation G.709/Y.1331, "*Interfaces for the Optical Transport Network (OTN): Amendment 1*", July 2010.
31. ITU-T Recommendation G.7041/Y.1303, "*Generic Framing Procedure*", October 2008.
32. ITU-T Recommendation G.798, "*Characteristics of optical transport network hierarchy equipment functional blocks*", June 2010.
33. ITU-T Recommendation G.798 Amendment 1, "*Characteristics of optical transport network hierarchy equipment functional blocks*", February 2011.
34. ITU-T Recommendation G.798 Corrigendum 1, "*Characteristics of optical transport network hierarchy equipment functional blocks*", February 2011.
35. ITU-T Recommendation G.8021/Y.1341, "*Characteristics of Ethernet transport network equipment functional blocks*", October 2010.
36. ITU-T Recommendation G.806, "*Characteristics of transport equipment – Description methodology and generic functionality*", January 2009.
37. ITU-T Recommendation G.8261/Y.1361, "*Timing and synchronization aspects in packet networks*", April 2008.
38. ITU-T Recommendation G.8262/Y.1362, "*Timing characteristics of synchronous Ethernet Equipment Slave Clock (EEC)*", July 2010.
39. ITU-T Recommendation G.8264/Y.1364, "*Distribution of timing information through packet networks*", October 2008.
40. ITU-T Recommendation G.8251, "*Control of Jitter and wander within the OTN*", September 2010.
41. ITU-T Recommendation G.8251, Amendment 1 "*Control of Jitter and wander within the OTN, Amendment 1*", April 2011.
42. ITU-T Recommendation G.975.1, "*Forward error correction for high bit-rate DWDM submarine systems*", February 2004.
43. ITU-T Recommendation G.Sup43, "*Transport of IEEE 10GBASE-R in Optical Transport Networks (OTN)*", June 2010 draft.
44. ITU-T Recommendation G.7044/Y.1347, "*Hitless Adjustment of ODUflex(GFP) (HAO)*", February 2011.
45. MIPS®, "*BusBridge™ 3 Modules User's Manual*", Revision 2.0, October 2009.
46. OIF, "*Scalable Serdes Framer Interface (SFI-S): Implementation Agreement for Interfaces beyond 40G for Physical Layer Devices*", Version 1.00, November 2008.
47. OIF, "*Serdes Framer Interface Level 5 (SFI-5): Implementation Agreement for 40Gb/s Interface for Physical Layer Devices*", Version 1.02, January 2002.
48. PCI-SIG, "*PCI Express Base Specification*", Revision 1.1, March 2005.
49. RCC, "*IRIG Standard 200-04: IRIG Serial Time Code Formats*", September 2004.
50. ITU-T Amendment 2 to Recommendation G.707/Y.1322, "*Network node interface for the synchronous digital hierarchy (SDH) Amendment 2*", November 2009.
51. OIF, OIF-CEI-02.0, "*Common Electrical I/O (CEI) - Electrical and Jitter Interoperability agreements for 6G+ bps and 11G+ bps I/O*", February 2005.

52. OIF, OIF-SFI5-02.0, "Serdes Framer Interface Level 5 Phase 2 (SFI-5.2): Implementation Agreement for 40Gb/s Interface for Physical Layer Devices", October 2006.
53. SFF Committee, INF-8077i, "10 Gigabit Small Form Factor Pluggable Module", Revision 4.5, August 2005.
54. SFF Committee, SFF-8431, "Specifications for Enhanced 8.5 and 10 Gigabit Small Form Factor Pluggable Module SFP+", Revision 2.1, August 2007.
55. SFF Committee, INF-8438i, "Specification for QSFP (Quad Small Form factor Pluggable) Transceiver", Rev 1.0, November 2006.
56. SFF Committee, SFF-8436, "Specification for QSFP+ COPPER AND OPTICAL MODULES", Rev 3.6, February 2011.
57. CFP MSA, "CFP MSA Hardware Specification", Revision 1.4, June 2010.
58. OIF, OIF-MSA-100GLH-EM-01.0, "Multisource Agreement for 100G Long-Haul DWDM Transmission Module – Electromechanical", June 2010.
59. 300PINMSA, "Document for 300PIN 40Gb Transponder", Public Document Edition 2, May 2002
60. InfiniBand<sup>SM</sup> Trade Association, "Supplement to InfiniBand Architecture Specification Volume 2 Release 1.2.1, Annex A6:120 Gb/s 12x Small Form-factor Pluggable (CXP) Interface Specification for Cables, Active Cables, & Transceivers", September 2009.
61. OIF, OIF-OFP-01.0 "OTN Over Packet Fabric Protocol (OPF) Implementation Agreement", November 2011.

Downloaded [controlled] by Benson Zhang  
Thursday, 21 April, 2016 09:05:07

## 3 Definitions

Table 1 Acronyms And Terms Used Within This Document

Acronym/Term	Definition
ACB	Adaptive Clocking Block
ACK	Acknowledge
AHB	Advanced High performance Bus
AIS	Alarm Indication Signal
AMBA	Advanced Microcontroller Bus Architecture
AMP	Asynchronous Mapping Procedure
APB	Advanced Peripheral Bus
APS	Automatic Protection Switching
ARM	Advanced RISC Machine
AU	Alignment Unit
AXI	Advanced Extensible Interface
BDI	Backwards Defect Indicator
BEI	Bit Error Indicator
BER	Bit Error Rate
BIAE	Backwards Incoming Alignment Error
BIP	Bit Interleaved Parity
BIU	Bus Interface Unit
BMP	Bit synchronous Mapping Procedure
BWR	Bandwidth Resize
CAUI	100 Gbps (C) Attachment Unit Interface
CBR	Constant Bit Rate
CBRC	Constant Bit Rate Client
CBRI	Constant Bit Rate Interface
CDC	Clock Domain Crossing
CDM	Charged Device Model
CEI	Common Electrical I/O
CFC	Correction Field Counter
CH	Channel
CHID	Channel Identifier
CLK	Clock
CM	Connection Monitoring
CMF	Client Management Frames
CMOS	Complimentary Metal Oxide Semiconductor
CnD	Cn Delta (difference between Cn and {m/n x Cm})
COPI	Combined OTN and Packet Interface
COREOTN_FMF	COREOTN Framer-Mux-Framer (there are 2 sets)
COREOTN_FO	ODU framers adjacent to COREOTN_SW (there are 2 sets)
COREOTN_SW	COREOTN ODUK Switch
CPB	Central Packet Buffer

Acronym/Term	Definition
CPPI	100Gbps (C) Parallel Physical Interface
CPRI	Common Public Radio Interface
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSF	Client Signal Failure
CSMA	Carrier Sense Multiple Access
CSU	Clock Supply Unit
DC	Direct Current
DCI	Digital CBR Interface
DCM	Digital Clock Manager
DCS	Digital Common Scheduler
DCSU	Digital Clock Synthesis Unit
DI	Decrement Indicator
DMA	Direct Memory Access
DPI	Digital Packet Interface
DSI	Digital Serial Interface
DSM	Delta Sigma Modulator
DSP	Digital Signal Processing
ECC	Error Correction Code
EMAC	Ethernet Media Access Control
EMPEI	Ethernet Management Packet Extract Insert
ENB	Active Low Enable
ENET	Ethernet
EOB	End of Burst
EOP	End of Packet
EP	End Point
ESD	Electrostatic Discharge
FAS	Frame Alignment Signal
FC	Fiber Channel
FCS	Frame Check Sequence
FEC	Forward Error Correction
FIFO	First In First Out
FPGA	Field Programmable Gate Array
FS	Fixed Stuff
FSL	FEC Statistics Logic
FSM	Finite State Machine
F/W	Firmware
GE	Gigabit Ethernet
GDPS	Geographically Dispersed Parallel Sysplex
GE	Gigabit Ethernet
GFP	Generic Framing Procedure
GFP-F	Generic Framing Procedure Framed

Acronym/Term	Definition
GMP	Generic Mapping Procedure
GPIO	General Purpose Input Output
GSUP	General Supplement
HAO	Hitless Adjustment of ODUflex
HBM	Human Body Model
HO	High Order
H/W	Hardware
IAE	Incoming Alignment Error
IB	Input Boundary
IEEE	Institute of Electrical and Electronics Engineers
II	Increment Indicator
INT	Interrupt
IPG	Inter Packet Gap
IRIG	Inter Range Instrumentation Group
IRQ	Interrupt Request
ISO	International Standards Organization
ITU	International Telecommunications Union
JC	Justification Control
JEDEC	Joint Electron Devices Engineering Council
JOH	Justification Overhead
JTAG	Joint Test Action Group
LACP	Link Aggregation Control Protocol
LAN	Local Area Network
LCR	Link Connection Resize
LFD	Loss of Frame Delineation
LFSR	Linear Feedback Shift Register
LIFD	Line Interface Demultiplexer
LO	Low Order
LOF	Loss of Frame
LOL	Loss of Line
LOM	Loss of Multiframe
LOS	Loss of Signal
LR	Long Range
LSB	Least Significant Bit
LSS	Loss of Sequence Synchronization
LUT	Look Up Table
LVCMOS	Low Voltage CMOS
LVDS	Low Voltage Differential Signaling
LV TTL	Low Voltage Transistor Transistor Logic
MAC	Media Access Control
MDU	Multiply Divide Unit
MEM	Memory

Acronym/Term	Definition
MFA	Multi Frame Alignment
MFAS	Multi Frame Alignment Signal
MIPS	Microprocessor without Interlocked Pipeline Stages
MLD	Multi Lane Distribution
MSB	Most Significant Bit
MSI	Multiplex Structure Identifier
NEBS	Network Equipment Building System
NIC	Network Interface Controller
NJO	Negative Justification
NMI	Non Maskable Interrupt
NMS	Network Management System
OAM	Operations Administration Maintenance
OB	Output Boundary
OC	Optical Carrier
OCP	Open Core Protocol
ODU	Optical Data Unit
OH	Overhead
OIF	Optical Interworking Forum
OOF	Out Of Frame
OOB	Out Of Band
OPSA	OTN Phase Signaling Algorithm
OPU	Optical Payload Unit
OTN	Optical Transport Network
OTU	Optical Transport Unit
PCBI	Peripheral Component Bus Interface
PCI	Peripheral Component Interconnect
PCIE	Peripheral Component Interconnect Express
PCS	Physical Coding Sublayer
PDV	Packet Delay Variation
PHY	Physical
PID	Packet Identifier
PIF	Packet Interface
PJO	Positive Justification
PLL	Phase Locked Loop
PM	Path Monitoring
PMA	Physical Medium Attachment
PMC	Pacific Microelectronic Corporation
PMM	Packet Maker Monitor
PMON	Performance Monitoring
PMPM	Programmable Multi Port Memory
POH	Path Over Head
PRBS	Pseudo Random Bit Sequence

Acronym/Term	Definition
PSI	Payload Structure Identifier
PTP	Precision Timing Protocol
PWM	Pulse Width Modulation
RAM	Random Access Memory
RAPTR	Rate Agile Programmable Transceiver SERDES
RCOH	Resizing Overhead
RDI	Remote Defect Indication
REG	Register
REI	Remote Error Indication
REQ	Request
RMF	Resizing Multi-Frame
ROHS	Restriction Of Hazardous Substances
ROM	Read Only Memory
RSTB	Active Low Reset
RW	Read Write
RX	Receive
SCPB	Scalable Central Packet Buffer
SDH	Synchronous Digital Hierarchy
SDRAM	Synchronous Dynamic Access RAM
SER	Soft Error Rate
SERDES	Serializer Deserializer
SF	Signal Failure
SFD	Start Frame Delimiter
SFI	SERDES Framer Interface
SFI-5.1	SERDES Framer Interface Level 5
SFI-S	Scalable SERDES Framer Interface
SFP	Small Form factor Pluggable
SGMII	Serial Gigabit Media Independent Interface
SI	Slave Interface
SIF	System Interface
SM	Section Monitoring
SMF	Server Multi-Frame
SNCP	Sub Network Connection Protection
SOB	Start of Burst
SOF	Start of Frame
SONET	Synchronous Optical Network
SOP	Start of Packet
SPI	Serial Peripheral Interface
SPRAM	Scratchpad RAM
SR	Short Reach
SS	Subsystem
SSM	Synchronization Status Message

<b>Acronym/Term</b>	<b>Definition</b>
STM	Synchronous Transport Module
STS	Synchronous Transport Signal
S/W	Software
SWIZZLE	PMC-Sierra's FEC
TAP	Test Access Port
TC	Tandem Connection
TCM	Tandem Connection Monitoring
TDM	Time Division Multiplexing
TIM	Trace Identifier Mismatch
TIU	Trace Identifier Unstable
TOC	Top of Calender
TOD	Time of Day
TOH	Transport Over Head
TS	Tributary Slot
TSE	Test Sequence Error
TTI	Trail Trace Identifier
TTT	Timing-Transparent Transcoding
TX	Transmit
UDP	User Datagram Protocol
UPI	Universal Payload Identifier
VC	Virtual Concatenation
VLAN	Virtual Local Area Network
VPE	Virtual Processing Element
WRR	Weighted Round Robin
XFI	10 Gbps (X) form Factor Interface
XFP	10 Gbps (X) form Factor Pluggable
XLAUI	40 Gbps (XL) Attachment Unit Interface
XLPI	40 Gbps (XL) Parallel Physical Interface

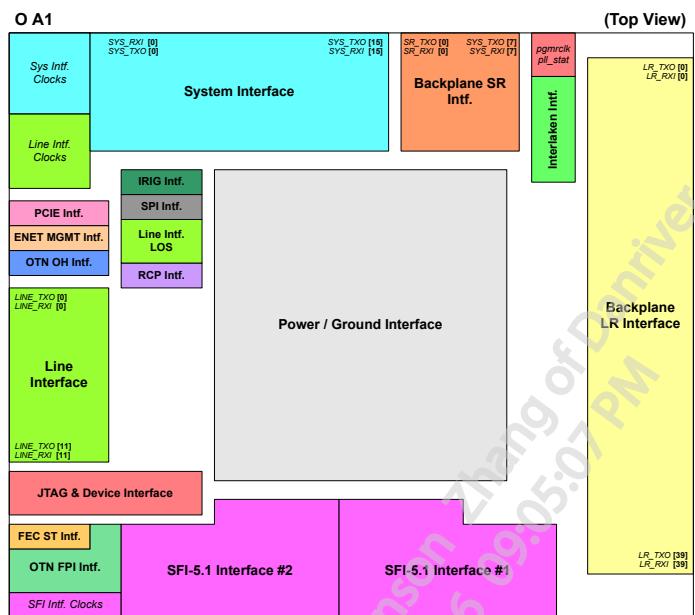
## 4

## Pinout

## 4.1

## Device PCB Floor Planning Diagram

Figure 1 Device PCB Floor Planning Diagram



The gray, center portion of Figure 1 is reserved for power and ground connections. The interfaces shown in this diagram are described in Section 4.2. For further information regarding SERDES lane mapping, please see Section 4.5. For further information regarding SERDES muxing, please see Section 4.2.4 and Section 5.11.5.2.5.

A detailed ballmap of the device is provided in [6].

## 4.2

## Pin Description

The following sections describe each device pin. Unless stated otherwise, all control signals are active-high logic. All unused SERDES from the RAPTR cell set [7] can be left unconnected.

Table 2 Device I/O Definition

Type	Definition
Input	Pin always functions as an input.
Output	Pin is always driven.
Three-State Output	Pin is either driven or held in high-impedance.
BiDir	Pin may be either an input or a three-state output (Bi-directional).
Open Drain	Pin is either driven low or held in high-impedance.

## 4.2.1 Device Interface

For the AC timing characteristics of this interface, see [Section 15.1](#).

Table 3 Device Interface Pin Description

Pin Name	Type	Function
REFCLK_P/N	Input	<p><b>155.52 MHz Reference Clock</b>            The nominal 155.52 MHz reference clock is used as a reference for all SERDES when operating at their nominal rates. It is also used as a reference for the internal data paths.            External holdover circuitry is required to ensure glitch free transition between sources.            Differential LVDS input with integrated termination.</p>
RSTB	Input	<p><b>Active Low Device Reset</b>            The device reset is an asynchronous active low reset. While reset is active, all serial receive interfaces are powered down and all serial transmit interfaces are squelched. RSTB resets all registers to their default value.            1.8V Schmitt Trigger Digital Input.</p>
INTB	Open Drain Output	<p><b>Active Low Device Interrupt</b>            INTB asserts low whenever an unmasked interrupt event is detected on any of the internal interrupt sources. Each major interrupt output from a major subsystem block can be independently unmasked to drive INTB and/or the PCIe interface interrupt (MSI/MSIX). Note that INTB remains low until all active unmasked interrupt sources are acknowledged at their source. At this time, INTB becomes high-impedance.            An external pull-up resistor of 1 to 10 kΩ is required on the pin to pull the line up to logic 1 when INTB is deasserted.            LVCMOS18 low-speed signal.</p>
PGMRCLK[3:0]	Output	<p><b>Programmable Recovered Clocks</b>            The PGMRCLK[3:0] outputs are configurable to provide access to all of the client recovered / transmit clocks and DCSU outputs within the device.            A fractional divider is provided on all of the PGMRCLK outputs. Each output supports 125 MHz maximum. The division ratio for each fractional divider is register configurable using an 8-bit integer, 32-bit numerator, and 32-bit denominator.            See <a href="#">Section 5.16.1.3</a> for more information.            LVCMOS18 low-speed signal.</p>

Pin Name	Type	Function
PLL_STATUS[1:0]	Output	<p><b>Phase Lock Loop Status</b></p> <p>The PLL_STATUS[0] output provides direct access to a divided-down version of the 155.52 MHz device REFCLK. This clock is divided down by a factor of 256 to 607.5 kHz.</p> <p>The PLL_STATUS[1] output provides direct access to a divided-down version of the 200 MHz device configuration bus clock (PCLK). This clock is divided down by a factor of 256 to 781.25 kHz.</p> <p>The PLL_STATUS[1:0] outputs will not begin toggling until loading and execution of the boot code from the external SPI flash device has been completed by device firmware. Toggling of the outputs can be used by an external device to detect when loading and execution of the boot loader code is complete. Before toggling they will assert a value of 1'b0.</p> <p>See <a href="#">Section 5.16.1.3</a> for more information.</p> <p>LVCMOS18 low-speed signal.</p>
PMON_SYNC_I	Input	<p><b>Performance Monitor Synchronous Update Input</b></p> <p>This device contains several periodic pulse generators that can be used to initiate periodic reads of performance monitoring counters in the device. These generators can be synchronized externally by providing a rising edge on PMON_SYNC_I. See <a href="#">Section 5.16.1.4</a> for more information.</p> <p>AC timing is provided in <a href="#">Section 15.1.5</a>.</p> <p>LVCMOS18 low-speed signal with pull down.</p>
PMON_SYNC_O	Output	<p><b>Performance Monitor Synchronous Update Output</b></p> <p>PMON_SYNC_O can be used to drive the PMON_SYNC_I of other devices. It can be configured to output a periodic pulse at a user-programmable interval. See <a href="#">Section 5.16.1.4</a> for more information.</p> <p>AC timing is provided in <a href="#">Section 15.1.6</a>.</p> <p>LVCMOS18 low-speed signal.</p>
FPI	Input	<p><b>8 kHz OPSA Delay Compensation Timing Reference</b></p> <p>The 8 kHz Delay Compensation Timing Reference is used to synchronize OPSA packet header timestamps carried by data transmitted across the backplane fabric or the ODUk switch. These timestamps are used to compensate for packet delay variation across a fabric and to maintain fabric buffers at a desired fill level. Additionally, these timestamps are used to track the ODUflex resizing ramp by tracking to maintain a constant fabric latency during a resizing ramp event.</p> <p>The FPI 8 kHz frame pulse is required for all applications in which ODUk packets are sent to an external fabric. The frame pulse is not required when the Interlaken interface is not used and ODUk packets do not exit the device. Internal logic will flywheel when the frame pulse is not present. The frame pulse period can be an integer multiple of 125 us (for example 2 kHz).</p> <p>See <a href="#">Section 5.16.1.6</a> for more information.</p> <p>LVCMOS18 low-speed signal; active high pulse.</p>

## 4.2.2

### Line Interface

The line interface supports multiple electrical interfaces and clients up to a total aggregate bandwidth of 120Gbps distributed over 12individual 10 Gbps lanes. Higher rate electrical interfaces (40G, 100G) are supported by combining multiple lanes together. Lanes can be reordered using a 12x12 switch located internally between the SERDES and protocol blocks. This provides optimal PCB routing of device pins to optical modules.

For the supported electrical interfaces and clients, see [Section 4.3](#).

Data signals use the Saphyre-16 family SERDES from the RAPTR cell set. For details, see [\[7\]](#).

Table 4 Line Interface Pin Description

Pin Name	Type	Function
LINE_RXI_P/N[11:0]	Input	<b>Line Receive Data</b> RAPTR Saphyre-16.
LINE_LOS[11:0]	Input	<b>Line Loss of Signal (LOS)</b> When LINE_LOS[x] is set to 1, the corresponding receive datapath performs protocol-specific consequential actions as defined in <a href="#">Section 5.17</a> . Each LINE_LOS[x] can be switched to any lane of the LINEOTN, CBRC, and ENET(LINE) subsystems. LOS switching is configured independently of Line Receive Data switching. Each LINE_LOS[x] can be independently inverted and disabled. LVCMS18 low-speed signal
LINE_TXO_P/N[11:0]	Output	<b>Line Transmit Data</b> RAPTR Saphyre-16.
LINE_REFCLK_P/N[11:0]	Output	<b>Line Transmit Reference Clock</b> Reference clock for optical modules. LINE_REFCLK_P/N is a nominally 50% duty cycle clock with frequency either 1/16 or 1/64 of the transmit data rate. Differential LVDS.

## 4.2.3 SFI-5.1 Interface

The SFI-5.1 interface supports two independent OIF-SFI5-01.0 interfaces for connecting to 40G optical modules or other external components.

For the supported electrical interfaces and clients, see [Section 4.3](#).

**Note:** The LIFD\_MUX block within the LIFD subsystem provides interface swapping of the two SFI-5.1 interfaces between the LINEOTN and CBRC subsystems. Also, the SFI-5.1 MLD layer in the LINEOTN and CBRC subsystems support lane re-ordering (HO to LO lane reversal) within each interface.

The data and deskew signals use the Delta-8 family SERDES from the RAPTR cell set. For details, see [\[7\]](#).

**Table 5 SFI-5.1 Interface Pin Description**

Pin Name	Type	Function
SFI1_RXDATA_P/N[15:0] SFI2_RXDATA_P/N[15:0]	Input	<b>Receive Data</b> Receive serial optical data. RXDATA[15] is the first bit received. RXDATA[0] is the last bit received. Each data signal is a 2.488 Gbps to 2.831 Gbps stream. RAPTR Delta-8.
SFI1_RXDSC_P/N SFI2_RXDSC_P/N	Input	<b>Receive Deskew Channel</b> Enables deskew of RXDATA[15:0]. RXDSC contains reference frames consisting of 4 framing bytes, 4 bytes of expansion header, and 16 sets of 8-byte samples of each RXDATA signal. Samples are taken round-robin starting with RXDATA[15] and ending with RXDATA[0]. RXDSC is a 2.488 Gbps to 2.831 Gbps stream. RAPTR Delta-8.
SFI1_RXS SFI2_RXS	Input	<b>Receive Status</b> When set to 0, the RXDATA is derived from the optical receive signal. When set to 1 a Receive Alarm has been detected. LVCMS18 low-speed signal.
SFI1_TXDATA_P/N[15:0] SFI2_TXDATA_P/N[15:0]	Output	<b>Transmit Data</b> Transmit data for an optical module. TXDATA[15] is the first bit transmitted. TXDATA[0] is the last bit transmitted. Each TXDATA signal is a 2.488 Gbps to 2.831 Gbps stream. TXDATA is frequency locked to TXDCK with an unspecified static phase offset. RAPTR Delta-8.
SFI1_TXDSC_P/N SFI2_TXDSC_P/N	Output	<b>Transmit Deskew Channel</b> Provides reference frames to enable skew measurements of TXDATA. TXDSC contains reference frames consisting of 4 framing bytes, 4 bytes of expansion header, and 16 sets of 8-byte samples of each TXDATA signal. TXDSC is a 2.488 Gbps to 2.831 Gbps stream. TXDSC is frequency locked to TXDCK with unspecified static phase offset. RAPTR Delta-8.

Pin Name	Type	Function
SFI1_TXDCK_P/N SFI2_TXDCK_P/N	Output	<b>Transmit Data Clock</b> Timing reference for TXDATA and TXDSC. TXDCK is a nominally 50% duty cycle clock with a frequency that is one quarter of the TXDATA and TXDSC rate. TXDCK is frequency locked to TXDATA and TXDSC. Static phase offset between TXDCK and TXDATA and TXDSC is unspecified. RAPTR Delta-8.
SFI1_TXREFCLK_P/N SFI2_TXREFCLK_P/N	Input	<b>Transmit Frequency Reference</b> Transmit frequency reference. TXREFCLK is a nominally 50% duty cycle clock with a frequency that is one quarter of the TXDATA and TXDSC rate. TXREFCLK is frequency locked to TXDATA and TXDSC. Static phase offset between TXREFCLK and TXDATA and TXDSC is unspecified. REFCLK_P/N and SFI1/2_TXCKSRC_P/N may also be used as transmit frequency references. Differential LVDS input with integrated termination.
SFI1_TXCKSRC_P/N SFI2_TXCKSRC_P/N	Input	<b>Transmit Clock Source</b> Transmit frequency reference. TXCKSRC is a nominally 50% duty cycle clock with a frequency that is one quarter of the TXDATA and TXDSC rate. TXCKSRC is frequency locked to TXDATA and TXDSC. Static phase offset between TXCKSRC and TXDATA and TXDSC is unspecified. REFCLK_P/N and SFI1/2_TXREFCLK_P/N may also be used as transmit frequency references. Differential LVDS input with integrated termination.

#### 4.2.4 System Interface

The system interface supports multiple electrical interfaces and clients. It consists of sixteen individual 10 Gbps lanes. Interlaken can use lanes [15:0] for a total aggregate bandwidth of 160 Gbps. All other clients can use lanes [11:0] for a total aggregate bandwidth of 120 Gbps. Lanes can be combined into higher rate electrical interfaces (e.g. 40G and 100G) by combining multiple lanes together. Lanes [11:0] can be reordered arbitrarily using a 12x12 switch located internally between the SERDES and the subsystems.

The system interface supports two Interlaken interfaces (#1 and #2) with lane rates of 3.125 Gbps, 6.25 Gbps, 10.3125 Gbps, 11.5 Gbps, or 12.5 Gbps. For the mapping of the Interlaken #1 and #2 lanes to the System Interface, see [Figure 118](#). A summary is also provided in [Table 62](#).

For the supported electrical interfaces and clients, see [Section 4.3](#).

Data signals use Saphyre-16 family SERDES from the RAPTR cell set. For more details, see [\[7\]](#).

**Table 6 System Interface Pin Description**

Pin Name	Type	Function
SYS_RXI_P/N[15:0]	Input	<b>System Receive Data</b> RAPTR Saphyre-16.
SYS_LOS[11:0]	Input	<b>System Loss of Signal (LOS)</b> When SYS_LOS is set to 1, the corresponding receive datapath performs protocol-specific consequential actions as defined in <a href="#">Section 5.17</a> . For Interlaken interfaces, SYS_LOS[x] is switched alongside SYS_RXI_P/N[x]. For all other interfaces, LOS switching is configured independently of System Receive Data switching. Each SYS_LOS[x] can be independently inverted and disabled. LVCMOS18 low-speed signal.
SYS_TXO_P/N[15:0]	Output	<b>System Transmit Data</b> RAPTR Saphyre-16.
SYS_REFCLK_P/N[11:0]	Output	<b>System Transmit Reference Clock</b> Reference clock for optical modules. SYS_REFCLK_P/N is a nominally 50% duty cycle clock with frequency either 1/16 or 1/64 of the transmit data rate. Note: reference clocks are only provided for the 12 lanes [11:0] that can carry client data. Differential LVDS.

#### 4.2.5 Backplane Short Reach Interface

The Backplane SR (Short Reach) interface can carry 8 x 3.125 Gbps or 8 x 6.25 Gbps signals from the Interlaken #1 and #2 interfaces. For the mapping of the Interlaken #1 and #2 lanes to the system interface, see [Figure 118](#). A summary is also provided in [Table 62](#).

All signals use Centaur-8 family SERDES from the RAPTR cell set. For details, see [\[7\]](#).

Table 7 Backplane Short Reach Interface Pin Description

Pin Name	Type	Function
SR_RXI_P/N[7:0]	Input	<b>Short Reach Receive Data</b> RAPTR Centaur-8.
SR_TXO_P/N[7:0]	Output	<b>Short Reach Transmit Data</b> RAPTR Centaur-8.

## 4.2.6 Backplane Long Reach Interface

The Backplane LR (Long Reach) interface can carry 40 x 3.125 Gbps or 40 x 6.25 Gbps signals from the Interlaken #1 and #2 interfaces. For the mapping of the Interlaken #1 and #2 lanes to the system interface, see [Figure 118](#). A summary is also provided in [Table 62](#).

All signals use Titan-8 family SERDES from the RAPTR cell set. For details, see [\[7\]](#).

Table 8 Backplane Long Reach Interface Pin Description

Pin Name	Type	Function
LR_RXI_P/N[39:0]	Input	<b>Long Reach Receive Data</b> RAPTR Titan-8.
LR_TXO_P/N[39:0]	Output	<b>Long Reach Transmit Data</b> RAPTR Titan-8.

## 4.2.7 Interlaken Flow Control Interface

Each Interlaken Flow Control interface corresponds to one of the device's Interlaken interfaces implemented in the SIFD subsystem. For this interface, pin [0] is connected to Interlaken interface #1 and pin [1] is connected to Interlaken interface #2.

For the functional timing and AC timing characteristics of this interface, see [Section 8.2](#) and [Section 15.4](#).

**Table 9 Interlaken Flow Control Interface Pin Description**

Pin Name	Type	Function
FC_RCLK[1:0]	Output	<b>Receive Packet Interface Flow Control Clock</b> FC_RCLK is the source-synchronous output clock for the flow control information advertised on FC_RDATA. LVCMOS18 low-speed signal.
FC_RSYNC[1:0]	Output	<b>Receive Packet Interface Flow Control Synchronization</b> FC_RSYNC is asserted when FC_RDATA carries the flow control information for channel 0 when it is in-order. FC_RSYNC is source-synchronous with both edges of FC_RCLK. LVCMOS18 low-speed signal.
FC_RDATA[1:0]	Output	<b>Receive Packet Interface Flow Control Data</b> FC_RDATA provides flow control information for each received channel, as programmed in the Interlaken flow control calendar. FC_RDATA is source-synchronous with both edges of FC_RCLK. LVCMOS18 low-speed signal.
FC_TCLK[1:0]	Input	<b>Transmit Packet Interface Flow Control Clock</b> FC_TCLK is the source-synchronous input clock for the flow control information advertised on FC_TDATA. LVCMOS18 low-speed signal.
FC_TSYNC[1:0]	Input	<b>Transmit Packet Interface Flow Control Synchronization</b> FC_TSYNC indicates when FC_TDATA carries flow control information for the first calendar entry. FC_TSYNC is sampled source-synchronous with both edges of FC_TCLK. LVCMOS18 low-speed signal.
FC_TDATA[1:0]	Input	<b>Transmit Packet Interface Flow Control Data</b> FC_TDATA carries flow control information for each transmit channel, as programmed in the Interlaken flow control calendar. FC_TDATA is sampled source-synchronous with both edges of FC_TCLK. LVCMOS18 low-speed signal.

## 4.2.8 OTN Ring Control Port Interface

The OTN RCP (Ring Control Port) interface has two separate RCPs: Remote Information RCP (RI\_RCP), and Defect Information RCP (DI\_RCP). The receive and transmit RI\_RCP carries BDI/BEI/BIAE bits from the sink framers and the APS/PCC bytes. The DI\_RCP transmits defects and alarms. Both RI\_RCP and DI\_RCP are standard SGMIIs (Serial Gigabit Media Independent Interfaces) running at 1.25 Gbps. The frame format is described in [Section 5.8.12](#) for both RI\_RCP and DI\_RCP.

The signals use Delta-8 family SERDES from the RAPTR cell set. For details, see [\[7\]](#).

**Table 10 OTN Ring Control Port Interface Pin Description**

Pin Name	Type	Function
RI_RCP_RXI_P/N	Input	<b>Remote Information OTN Ring Control Port SGMII Receive</b> RAPTR Delta-8
RI_RCP_TXO_P/N	Output	<b>Remote Information OTN Ring Control Port SGMII Transmit</b> RAPTR Delta-8
DI_RCP_RXI_P/N	Input	<b>Defect Information OTN Ring Control Port SGMII Receive</b> RAPTR Delta-8
DI_RCP_TXO_P/N	Output	<b>Defect Information OTN Ring Control Port SGMII Transmit</b> RAPTR Delta-8

#### 4.2.9 OTN Overhead Interface

The OTN overhead ports provide insertion and extraction of a programmable selection of bytes within the OTU, ODU, and OPU overheads. The overhead is organized into frames whose format is described in [Section 5.8.11](#). Byte selection options are described in [Section 5.8.11](#). The overhead frames are carried in three standard SGMIIIs running at 1.25 Gbps.

Pin [0] of OH\_RXI/TXO\_P/N is connected to COREOTN Overhead Processor #1 (OHP1). Pin [1] is connected to COREOTN Overhead Processor #2 (OHP2). Pin [2] is connected to COREOTN Overhead Processor #3 (OHP3). The framer stages that are connected to the OHP2 and OHP3 blocks are configurable. OHP1 is dedicated to framer Stage 1 and Stage N while OHP2 and OHP3 can be configured to connect to Stage 2, Stage 3A, Stage 3B, or Stage 4.

The signals use Delta-8 family SERDES from the RAPTR cell set. For details, see [\[7\]](#).

**Table 11 OTN Overhead Interface Pin Description**

Pin Name	Type	Function
OH_RXI_P/N[2:0]	Input	<b>OTN Overhead SGMII Receive</b> RAPTR Delta-8.
OH_TXO_P/N[2:0]	Output	<b>OTN Overhead SGMII Transmit</b> RAPTR Delta-8.

#### 4.2.10 OTN Frame Pulse Interface

The device supports inserting PTP time stamps into OTN overhead based on the OTN frame provided by the OTU RX/TX multi-frame pulse outputs (for example, at the start of a multi-frame). See [Section 5.6.4.8](#) for details. This interface is shared between the LINEOTN and SYSOTN subsystems using a channelized mux.

For the functional timing and AC timing characteristics of this interface, see [Section 8.3](#) and [Section 15.5](#).

**Table 12 OTN Frame Pulse Interface Pin Description**

Pin Name	Type	Function
OTU_RX_MFPO[11:0]	Output	The OTU_RX_MFPO outputs mark the multi-frame and frame boundaries as observed at the RX framers. These outputs are asynchronous and are generated using an internal clock whose period varies from 2.82 ns to 2.97 ns according to client type. A pulse width of 17 clock cycles denotes a multi-frame boundary where MFAS = 0. A pulse width of 7 clock cycles denotes all other frame boundaries (MFAS = 1 - 255). LVCMS18 low-speed signal.
OTU_TX_MFPO[11:0]	Output	The OTU_TX_MFPO outputs mark the multi-frame and frame boundaries as observed at the TX Framers. These outputs are asynchronous and are generated using an internal clock whose period varies from 2.82 ns to 2.97 ns according to client type. A pulse width of 17 clock cycles denotes a multi-frame boundary where MFAS = 0. A pulse width of 7 clock cycles denotes all other frame boundaries (MFAS = 1 - 255). LVCMS18 low-speed signal.

#### 4.2.11 4-Wire OTN FEC Statistics Interface

The FSL (FEC Statistics Logic) block of the LINEOTN subsystem collects FEC statistics and assembles them in a FSL-T (FSL-Table). The FSL-T is then output using the four signals of the OTN FEC Statistics interface.

For the functional timing and AC timing characteristics of this interface, see [Section 8.4](#) and [Section 15.6](#).

**Table 13 OTN FEC Statistics Interface Pin Description**

Pin Name	Type	Function
GL_CLK	Output	<b>FEC Statistics Clock</b> Programmable rate clock (nominally 20 MHz) for timing GL_FP, GL_VAL, and GL_DAT. LVCMOS18 low-speed signal.
GL_FP	Output	<b>FEC Frame Pulse</b> Periodic frame pulse that marks a new set of FSL-T table data starting on GL_DAT. GL_FP interval is between 1ms and 1s. LVCMOS18 low-speed signal.
GL_VAL	Output	<b>FEC Valid</b> Set high for 128 consecutive GL_CLK cycles to mark each row in the FSL-T table. GL_VAL is set low for 8 consecutive cycles between rows in the FSL-T table. LVCMOS18 low-speed signal.
GL_DAT	Output	<b>FEC Statistics Data</b> FEC Statistics data per the FSL-T table format. GL_DAT is valid when GL_VAL is high. Otherwise, GL_DAT should be ignored. LVCMOS18 low-speed signal.

#### 4.2.12 Ethernet Management Interface

The Ethernet Management interface provides access to Ethernet Link OAM messages to/from any of the internal client Ethernet MACs. The Ethernet Management port is a standard SGMII running at 1.25 Gbps. This interface connects an internal 1000 Mbps Ethernet MAC to an external PHY. To carry frame data and link rate information between a 1000 Mbps full-duplex Ethernet PHY and an Ethernet MAC, SGMII uses a differential pair in each direction. This allows clocking information to be recovered from the data stream. Separate clock pins are not provided.

The Ethernet Management port serial signals use Delta-8 family SERDES from the RAPTR cell set. For details, see [7].

Table 14 Ethernet Management Interface Pin Description

Pin Name	Type	Function
MGMT_RXI_P/N	Input	<b>Ethernet Management Port SGMII Receive</b> RAPTR Delta-8
MGMT_TXO_P/N	Output	<b>Ethernet Management Port SGMII Transmit</b> RAPTR Delta-8

## 4.2.13 PCIe Interface

The PCIe interface is compatible with the PCIe specification 1.1. It supports x1, x2, or x4 lane widths with an individual lane rate of 2.5 Gbps.

The PCIe serial signals use Centaur-8 family SERDES from the RAPTR cell set. For details, see [7].

**Table 15** PCIe Interface Pin Description

Pin Name	Type	Function
PCIE_RXI_P/N[3:0]	Input	<b>PCIe Receive Data</b> RAPTR Centaur-8
PCIE_TXO_P/N[3:0]	Output	<b>PCIe Transmit Data</b> RAPTR Centaur-8

## 4.2.14 TOD and IRIG Interface

The TOD (Time-of-Day) and the IRIG (Inter Range Instrumentation Group) interfaces can be used to distribute or accept the TOD and/or a reference frequency. Only one of these interfaces can be enabled at a particular time. The selected interface is used for PTP mechanisms in the device. The outputs of these two interfaces are sourced from either the ENET(SYS) or the ENET(LINE) subsystem using a 2:1 mux. The inputs of these two interfaces are connected to both ENET(SYS) and ENET(LINE) using a T connection.

For the functional timing and AC timing characteristics of this interface, see [Section 8.5](#) and [Section 15.7](#).

**Table 16** TOD and IRIG Interfaces Pin Description

Pin Name	Type	Function
IRIG_CLKI	Input	<p><b>TOD Interface Clock</b></p> <p>When the TOD interface is enabled, this clock samples IRIG_DATI. It is a nominal 100Hz clock, but is not limited to this rate. It could be any clock rate between 100 Hz and 1 kHz.</p> <p>When the IRIG interface is enabled, IRIG_CLKI is not used.</p> <p>LVCMOS18 low-speed signal with pull down.</p>
IRIG_DATI	Input	<p><b>IRIG or TOD Interface Data</b></p> <p>When the TOD interface is enabled, IRIG_DATI contains the integer seconds value that corresponds to the interface's 1PPS pulse (ONE_PPS_I). IRIG_DATI is sampled by the rising edge of IRIG_CLKI.</p> <p>When the IRIG interface is enabled, IRIG_DATI contains the 100 bps IRIG-B B00x pulse-width modulated data stream.</p> <p>LVCMOS18 low-speed signal with pull up.</p>
ONE_PPS_I	Input	<p><b>One Pulse Per Second Input</b></p> <p>ONE_PPS_I is expected to pulse once per second when the IRIG master's time-of-day transitions from one integer second to another.</p> <p>ONE_PPS_I is only used with the TOD interface.</p> <p>When the IRIG interface is enabled, the 1PPS event is encoded in the IRIG data stream.</p> <p>ONE_PPS_I is an asynchronous signal.</p> <p>LVCMOS18 low-speed signal with pull down.</p>
IRIG_CLKO	Output	<p><b>TOD Interface Output Clock</b></p> <p>This clock generates IRIG_DATO. It is valid when the TOD interface is enabled.</p> <p>IRIG_CLKO is a 100 Hz clock that is derived from the 125 MHz PTP_CLK.</p> <p>IRIG_CLKO is high-impedance when the device is not enabled as a TOD master.</p> <p>LVCMOS18 low-speed signal.</p>
IRIG_DATO	Output	<p><b>IRIG Time-of-Day Data Stream</b></p> <p>When the TOD interface is enabled, this is the data stream that gives the integer seconds time-of-day that corresponds to the subsequent ONE_PPS_O pulse. IRIG_DATO is updated on the falling edge of IRIG_CLKO.</p> <p>When the IRIG interface is enabled, IRIG_DATO contains the 100 bps IRIG-B B00x pulse-width modulated data stream.</p> <p>IRIG_DATO may be configured to be high-impedance.</p> <p>LVCMOS18 low-speed signal.</p>

Pin Name	Type	Function
ONE_PPS_O	Output	<b>One pulse-per-second Output</b> This 1 Hz signal pulses when the local PTP time-of-day changes from one integer second to another. By default ONE_PPS_O remains at logic 1 for approximately 0.5 ms. ONE_PPS_O may be configured to be high-impedance. LVCMOS18 low-speed signal.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:08 PM

## 4.2.15 SPI Master Interface

The SPI (Serial Peripheral Interface) is used to read the device's boot code from an external flash boot ROM. It can also be used to write port activity information to a SPI address in order to allow an external device to control per-port status LEDs.

For the functional timing and AC timing characteristics of this interface, see [Section 8.6](#) and [Section 15.8](#).

**Table 17 SPI Master Interface Pin Description**

Pin Name	Type	Function
SPI_SCLK	Output	<p><b>SPI Serial Clock (output from master)</b></p> <p>This pin is used to provide a clock to the SPI slave devices and is used to control the flow of data to and from the device. Command, address, and output data present on the SPI_MOSI pin is always presented on the falling edge of SPI_SCLK. Input data from the SPI_MISO pin is always sampled on the rising edge of SPI_SCLK.</p> <p>The default clock frequency at power up is 1MHz.</p> <p>LVCMOS18 low-speed signal.</p>
SPI_MOSI	Output	<p><b>SPI Master Output Slave Input (output from master)</b></p> <p>The SPI_MOSI pin is used to shift data into the slave device. This pin is used for all data output including command and address sequences. Data on this pin is always clocked out on the falling edge of SPI_SCLK. Data present on this pin should be ignored by the slave whenever its corresponding SPI_SSB[3:0] is de-asserted.</p> <p>LVCMOS18 low-speed signal.</p>
SPI_MISO	Input	<p><b>SPI Master Input Slave Output (output from slave)</b></p> <p>The SPI_MISO pin is used to shift data out of the slave device. Data on this pin is always sampled on the rising edge of SPI_SCLK. This pin is expected to be in a high-impedance state whenever the device is deselected (i.e. when the corresponding SPI_SSB[3:0] is de-asserted).</p> <p>LVCMOS18 low-speed signal.</p>
SPI_SSB[3:0]	Output	<p><b>SPI Slave Select (active low; output from master)</b></p> <p>Four discreet slave pins. A slave select pin should be connected to the active low chip select pin of a discreet SPI slave. The slave select pin is asserted based on the address to which an SPI slave is mapped.</p> <p>Always connect SPI_SSB[0] to the SPI Boot FLASH slave device. If used, the SPI LED blinker slave device is always expected to be connected to SPI_SSB[3]. The device will write the port activity status to an address within the slave device. The slave device is expected to interpret the information and drive LEDs as required.</p> <p>LVCMOS18 low-speed signal.</p>

#### 4.2.16 JTAG Interface

The JTAG interface is an IEEE 1149.6 and IEEE 1149.1 compatible test interface. It is used for board-level production testing. Refer to the device BSDL file for a description of JTAG support and JTAG scan chain ordering.

For the functional timing and AC timing characteristics of this interface, see [Section 8.7](#) and [Section 15.9](#).

Table 18 JTAG Interface Pin Description

Pin Name	Type	Function
TRSTB	Input	<b>Active Low Test Access Reset</b> The active low test reset signal provides an asynchronous device TAP controller reset. For normal device operation, the TAP controller should be held in its Test-Logic-Reset state by holding the TRSTB pin low. It is also acceptable to connect the TRSTB input to the device RSTB input in order to reset the JTAG tap controller on power-up. 1.8V LVCMOS18 Schmitt Trigger Digital Input with Internal Pull-up.
TCK	Input	<b>Test Access Port (TAP) Clock</b> The test clock signal provides clock timing for test operations that are carried out using the test access port for the TAP controller. 1.8V LVCMOS18 Digital Schmitt Trigger Input.
TMS	Input	<b>Test Access Mode Select</b> The test mode select signal controls the test operations that are carried out in the TAP controller. TMS is sampled on the rising edge of TCK. 1.8V LVCMOS18 Digital Input with Internal Pull-up.
TDI	Input	<b>Test Access Data Input</b> TDI is sampled on the rising edge of TCK. 1.8V LVCMOS18 Digital Input with Internal Pull-up.
TDO	Output	<b>Test Access Data Output</b> TDO is updated on the falling edge of TCK. TDO is a 3-state output that is inactive except when shifting boundary scan data is in progress. 3-State LVCMOS18 Digital Output.

#### 4.2.17 Thermal Diode

The thermal diode can be used even when all power supplies are inactive.

Table 19 Thermal Diode Pin Description

Pin Name	Type	Function
THERM_PADE	Input	<b>Thermal Diode Anode</b> THERM_PADE is the emitter terminal of the PNP that is used as a thermal diode. This pin should be connected to ground when unused.
THERM_PADB	Output	<b>Thermal Diode Cathode</b> THERM_PADB is the base terminal of the PNP that is used as a thermal diode. This pin should be connected to ground when unused.

## 4.2.18 Power and Ground Interface

Table 20 Power and Ground Pin Description

Pin Name	Type	Function
VSS	Ground	<b>Device Ground</b>
VDDO_1V8, VDDO_1V8_REF	1.8V Digital Power Supply	<b>Digital I/O VDD</b> See <a href="#">Table 118</a> for details.
VDD	0.92V Digital Power Supply	<b>Digital Core VDD</b> See <a href="#">Table 118</a> for AVS details.
*_VDD, *_AVD	0.92V Analog Power Supply	See <a href="#">Table 118</a> for details.
*_VDD, *_AVD, VREF_1V2_LINE, VREF_1V2_SYS	1.2V Analog Power Supply	See <a href="#">Table 118</a> for details.
*_VDD, *_AVD	1.8V Analog Power Supply	See <a href="#">Table 118</a> for details.
AVD_LR_TX*		<b>Backplane Long Reach Interface Analog VDD</b> To be connected to the 1.0 V Analog T8 Overdrive only if the backplane channel has marginal performance since this will boost the transmit swing by ~10%. Otherwise, it can be connected to the 0.9V Analog supply. See <a href="#">Table 118</a> for details.

Table 21 Power Management Pin Description

Pin Name	Type	Function
AVS_ENB	Output	<b>Adaptive Voltage Scaling (AVS) Control</b> Use of the device requires Adaptive Voltage Scaling (AVS). The device will assert the AVS_ENB output to indicate the correct voltage to apply. When set to 0, the digital and analog core 0.925V supplies must be operated at a reduced voltage to meet datasheet power specifications 0.875V. When set to 1, the digital and analog core 0.925V supplies must be operated at the nominal voltage 0.925V. See <a href="#">Table 118</a> for power supply details. LVCMS18 low-speed signal.
AVS_RSTB	Input	<b>Adaptive Voltage Scaling (AVS) Reset</b> The device asserts the AVS_ENB output to indicate the correct voltage to apply (0.875V or 0.925V). The AVS_RSTB input controls the device logic that generates the AVS_ENB output. See <a href="#">Section 8.8.1</a> for functional timing.
VSENSE_OV9_BUMP	Output	<b>Digital Supply Voltage Sense</b> Connected to the device substrate digital core supply. Can be used to compensate for the voltage drop between the regulator and the die.
VSENSE_VSS_BUMP	Output	<b>Ground Voltage Sense</b> Connected to the device substrate ground. Can be used to compensate for the voltage drop between the on-die ground and the regulator ground.

## 4.3

## Supported Electrical Interfaces and Clients

Table 22 provides a summary of the supported electrical interfaces and client protocols for each SERDES interface. Refer to the RAPTR specification [7] for the electrical specifications for each SERDES type.

Table 22 Supported Electrical Interfaces and Client Protocols

Interface	SERDES	Electrical Interface	Client Protocol
Line Interface	12x 5/11.2G Saphyre-16	XFI / SFP+ (limiting)	12x 10GE LAN 12x OTU2/2e/2f/1e/1f 12x FC800/1200 12x OC-192/STM-64 12x 5/10G GDPS 12x CPRI 4.9G/6.1G/9.8G
		OTL3.4	3x OTU3/3e1/3e2
		STL256.4	3x OC-768/STM-256
		XLAUI, XLPPI	3x 40GE
		SFI-S	2x OTU3/3e1/3e2 2x OC-768/STM-256
		CAUI, CPPI, CXP	1x 100GE
		SFI-S, OTL4.10, CXP	1x OTU4
SFI-5.1 Interface	34x 2.5G Delta-8	SFI-5.1	2x OC-768/STM-256 2x OTU3/3e1/3e2
System Interface	16x 2.5/12.5G Saphyre-16	XFI / SFP+ (limiting)	12x 10GE LAN 12x FC800/1200 12x OC-192/STM-64 12x 5/10G GDPS 12x CPRI 4.9G/6.1G/9.8G
		STL256.4	3x OC-768/STM-256
		XLAUI, XLPPI	3x 40GE
		SFI-S	2x OC-768/STM-256
		CAUI, CPPI, CXP	1x 100GE
		CEI-11G-MR	Interlaken (PIF/CBRI)
		SFI-S, OTL4.10, CXP	1x OTU4
Backplane LR Intf.	40x 6G Titan-8	CEI-6G-LR	Interlaken (PIF/CBRI)
Backplane SR Intf.	8x 6G Centaur-8	CEI-6G-SR	Interlaken (PIF/CBRI)
ENET MGMT Intf.	1x 1.25G Delta-8	SGMII	GE
OTN OH Ins/Ext Intf.	3x 1.25G Delta-8	SGMII	GE
PCIe Interface	4x 2.5G Centaur-8	PCIe 1.1	1x, 2x, 4x lanes
OTN RCP Interface	2x 1.25G Delta-8	SGMII	GE

## 4.4 Supported Optical Modules

The device supports the following optical module MSA (multi-source agreement) specifications:

- Small Form Factor Pluggable (SFP+) Limiting Module [54]
- 10 Gigabit Small Form Factor Pluggable (XFP) [53]
- Quad Small Form Factor Pluggable Transceiver (QSFP) [55] (with XLPPI [24] compliant interface)
- Quad SFP+ (QSFP+) [56] (with XLPPI [24] compliant interface)
- 120 Gbps 12x Small Form Factor Pluggable (CXP) (Limiting modules only)
- 40G/100G (CFP)[57]
- 300-pin MSA
- OIF 168-pin MSA

Additionally, the following electrical interface standards are supported:

- “Ziffy” (XFI) [53]
- CEI-11G-SR [51]
- CEI-11G-MR [51]
- 40 Gbps Attachment Unit Interface (XLAUI), Annex 83B [24]
- 40 Gbps Parallel Physical Interface (XLPPI), Annex 86A [24]

**Note:**

- When the link rates are > 10.3125 Gbaud, the electrical parameters are scaled to the link rate.

- OTL3.4 [28]
- STL256.4 [50]
- 100 Gbps Attachment Unit Interface (CAUI), Annex 83B [24]
- OTL4.10 [28]
- 100 Gbps Parallel Physical Interface (CPPI), Annex 86A [24]

**Note:**

- When the link rates are > 10.3125 Gbaud, the electrical parameters are scaled to the link rate.

- SERDES Framer Interface Level 5 (SFI 5.1) [47]
- SERDES Framer Interface Level 5 Phase 2 (SFI 5.2) [52]
- Scalable SERDES Framer Interface (SFI-S) [46]

## 4.5 SERDES Lane Mapping

### 4.5.1 Line Side Interfaces Lane Mapping

#### 4.5.1.1 SFI-5.1 Interface Mapping

The device supports two independent OIF-SFI-5.1 interfaces . Each interface contains seventeen SERDES lanes, sixteen data and one deskew. The LIFD\_MUX block within the LIFD subsystem is used to connect the CBRC and LINEOTN subsystems to the SFI-5.1 Interface SERDES. The LIFD\_MUX provides multiplexing to connect either of the two SFI-5.1 interfaces to any of the two SFI-5.1 interfaces on the CBRC or LINEOTN subsystems. Each SFI-5.1 Interface SERDES can only connect to one of the CBRC or LINEOTN interfaces at a time. The LIFD\_MUX does not provide any support for individual lane reordering within each of the two SFI-5.1 interfaces. However, the SFI5.1 MLDs within the CBRC and LINEOTN provide

high order to low order lane reversal within each interface. For more details on the muxing supported in the LIFD\_MUX refer to [Section 5.5](#).

#### 4.5.1.2 Line Interface Mapping

The Line Interface SERDES are connected through a cross point mux (LINE\_DSIS), in the LIFD subsystem, to the CBRC, LINE ENET, and LINEOTN subsystems. The LINE\_DSIS provides multiplexing so that any of the twelve Line Interface SERDES can be connected to any of the twelve Line Interface lanes in the CBRC, LINE ENET, or LINEOTN subsystems. Any particular SERDES lane can only connect to one of the Line Interfaces in the CBRC, LINE ENET, or LINEOTN subsystems at a given time. In all cases, each logical lane is expected to use the same physical lane for both transmit and receive. For example, logical Line Interface lane Y must connect to LINE\_TXO\_P/N[X] and LINE\_RXI\_P/N[X]. For more details on the muxing supported in the LINE\_DSIS refer to [Section 5.5](#). The Line Interface also consists of LINE\_LOS[11:0] and LINE\_REFCLK\_P/N[11:0]. There is a one to one relationship between the physical lane and each reference clock, LINE\_REFCLK\_P/N[X] is associated with LINE\_TXO\_P/N[X]. The LINE\_LOS[11:0] inputs can be configurably inverted and/or enabled and then independently muxed to the CBRC, LINE ENET, or LINEOTN.

The Line Interface SERDES can optionally provide an internal LOS indication, referred to as DLOS ([Section 14.2](#)). The DLOS can be used in cases where an optical module does not provide a LOS indication. Each SERDES DLOS can be independently muxed to the CBRC, LINE ENET, or LINEOTN. When LOS is used exclusive of DLOS, the full flexibility is available. However, for each lane, it is possible to use LINE\_LOS and DLOS simultaneously for the same link with significantly reduced flexibility on the mapping of a LOS input lane to SERDES lane. Specifically, if SERDES lane [X] is used, then the successful, internal combination of DLOS and the LOS input requires that DLOS[X] and LOS[X] be associated with SERDES[X]. Any other combination could lead to the wrong lane being signaled as LOS. When combining LOS and DLOS for multi-lane interfaces with only a single LOS indication, the LOS input that corresponds to any one of the RX SERDES lanes may be used to signal for the entire interface. DLOS should be enabled for all of the lanes in the interface. Furthermore, if an optical module cage uses DLOS and is connected to device lanes [W, X, Y, Z] then the LOS inputs [W, X, Y, Z] cannot be used by other optical module cages, regardless of whether any lanes are using both LOS and DLOS simultaneously. These restrictions should be considered when selecting the LOS input pins to use for an optical module. PMC's recommendation for a module is to select a LOS pin from the same index range as the SERDES used to connect to the optical module.

#### 4.5.1.3 System Interface used as Line Interface Mapping

A subset of the System Interface SERDES, SYS\_RXI\_P/N[7:0] SYS\_TXO\_P/N[7:0], can be repurposed using the LIFD subsystem LINE\_DSIS to connect to the CBRC, ENET\_LINE, or LINEOTN subsystems to support a 3x40G Multi-Service Line Card where one port can optionally support either 40G or 100G. The Line Interface SERDES are used for the pluggable 40G/100G port and the eight System Interface SERDES are used to support up to two additional 40G ports. In this manner, the device supports either a single 100G port or three 40G ports. Note that, due to reuse of System Interface SERDES, some system side Interlaken interface lane restrictions apply [Section 4.5.3](#). As well, see [Table 27](#) for limitations on usage of Line Interface SERDES. In all cases, each logical lane is expected to use the same physical lane for both transmit and receive. For example, logical System Interface lane Y must connect to SYS\_TXO\_P/N[X] and SYS\_RXI\_P/N[X]. For more details on the muxing supported in the LINE\_DSIS refer to [Section 5.5](#). In this application SYS\_LOS[7:0] and SYS\_REFCLK\_P/N[7:0] are also repurposed. Once again, there is a one to one relationship between the physical lane and each reference clock, SYS\_REFCLK\_P/N[X] is associated with SYS\_TXO\_P/N[X]. The SYS\_LOS[7:0] inputs can be configurable inverted and/or enabled and then independently muxed to the CBRC, ENET\_LINE, or LINEOTN subsystems.

The System Interface SERDES can optionally provide an internal LOS indication, referred to as DLOS ([Section 14.2](#)). The DLOS can be used in cases where an optical module does not provide a LOS indication. Each SERDES DLOS can be independently muxed to the CBRC, ENET\_LINE, or LINEOTN. For each lane, the use of SYS\_LOS and DLOS are mutually exclusive. When LOS is used exclusive of DLOS, the full flexibility is available. However, for each lane, it is possible to use LINE\_LOS and DLOS simultaneously for the same link with significantly reduced flexibility on the mapping of a LOS input lane to SERDES lane. Specifically, if SERDES lane [X] is used, then the successful, internal combination of DLOS and the LOS input requires that DLOS[X] and LOS[X] be associated with SERDES[X]. Any other combination could lead to the wrong lane being signaled as LOS. When combining LOS and DOS for multi-lane interfaces with only a single LOS indication, the LOS input that corresponds to any one of the RX SERDES lanes may be used to signal for the entire interface. DLOS should be enabled for all of the lanes in the interface. Furthermore, if an optical module cage uses DLOS and is connected to device lanes [W, X, Y, Z] then the LOS inputs [W, X, Y, Z] cannot be used by other optical module cages, regardless of whether any lanes are using both LOS and DLOS simultaneously. These restrictions should be considered when selecting the LOS input pins to use for an optical module. PMC's recommendation for a module is to select a LOS pin from the same index range as the SERDES used to connect to the optical module.

## 4.5.2 System Side Interfaces Lane Mapping

### 4.5.2.1 System Interface Mapping

The device's System Interface SERDES supports client or system/backplane interfaces by providing sixteen lanes of SERDES. Only twelve lanes, SYS\_TXO\_P/N[11:0] and SYS\_RXI\_P/N[11:0], are used for connectivity to client interfaces. These twelve System Interface SERDES are connected through a cross point in the System Interface DSIS (SYS\_DSIS) to the CBRC, SYS ENET, and SYSOTN subsystems. The SYS\_DSIS provides multiplexing so that any of the twelve System Interface SERDES can be connected to any of the twelve System interface lanes in the CBRC, ENET(SYS), or SYSOTN subsystems. Any particular SERDES lane can only connect to one of the System interfaces in the CBRC, ENET(SYS), or SYSOTN subsystem at a given time. In all cases, each logical lane is expected to use the same physical lane for both transmit and receive. For example, logical Line Interface lane Y must connect to SYS\_TXO\_P/N[X] and SYS\_RXI\_P/N[X]. For more details on the system side muxing, please refer to [Section 5.11.5.2.5](#).

The System Interface also consists of SYS\_LOS[11:0] and SYS\_REFCLK\_P/N[11:0]. There is a one-to-one relationship between the physical lane and each reference clock, where SYS\_REFCLK\_P/N[X] is associated with SYS\_TXO\_P/N[X]. The SYS\_LOS[11:0] inputs can be optionally inverted and/or enabled and then independently muxed to the CBRC, ENET(SYS), and SYSOTN subsystems.

The System Interface SERDES optionally provides an internal LOS indication, referred to as DLOS. DLOS can be used in cases where an optical module does not provide a LOS indication. Each SERDES DLOS can be independently muxed to the CBRC, ENET(SYS), or SYSOTN. When LOS is used exclusive of DLOS, the full flexibility is available. However, for each lane, it is possible to use LINE\_LOS and DLOS simultaneously for the same link with significantly reduced flexibility on the mapping of a LOS input lane to SERDES lane. Specifically, if SERDES lane [X] is used, then the successful, internal combination of DLOS and the LOS input requires that DLOS[X] and LOS[X] be associated with SERDES[X]. Any other combination could lead to the wrong lane being signaled as LOS. When combining LOS and DOS for multi-lane interfaces with only a single LOS indication, the LOS input that corresponds to any one of the RX SERDES lanes may be used to signal for the entire interface. DLOS should be enabled for all of the lanes in the interface. Furthermore, if an optical module cage uses DLOS and is connected to device lanes [W, X, Y, Z] then the LOS inputs [W, X, Y, Z] cannot be used by other optical module cages, regardless of whether any lanes are using both LOS and DLOS simultaneously. These restrictions should be considered

when selecting the LOS input pins to use for an optical module. PMC's recommendation for a module is to select a LOS pin from the same index range as the SERDES used to connect to the optical module. For more information regarding DLOS, please see [Section 14.2](#).

#### 4.5.3 Interlaken Over System Side Interfaces

The device supports two Interlaken Interfaces, Interlaken 1 and Interlaken 2. These Interlaken interfaces make use of the Backplane Long Reach (T-8), Backplane Short Reach (C-8) and System (S-16) SERDES to support a wide range of interface rates and lane widths. In some cases an Interlaken Interface spans multiple SERDES types, in these cases rate and reach of the interface is defined by the highest common support across the SERDES. The C-8 and S-16 SERDES are intended to be used for interconnection to a device located on the same PCB, for longer reach support the T-8 SERDES must be used. The T-8 and C-8 SERDES support rates of 3.125 and 6.25 Gbps, while the S-16 SERDES support rates of 3.125, 6.25, 10.3125, 11.5 and 12.5 Gbps. The maximum number of lanes supported at 11.5 Gbps is 12 and at 12.5 the limit is 12. A summary of reach, rate, number of lanes and SERDES type for each Interlaken Interface is provided below, for additional details refer to [Section 5.11.5.2.5](#).

Table 23 Interlaken Lane Mapping Summary

Interface	Reach	Rate (Gbps)	Lanes (max)	SERDES
Interlaken 1	LR	3.125/6.25	20	T-8
	SR	3.125/6.25	24	T-8/C-8
	SR	3.125/6.25	24	S-16/C-8
	SR	10.3125	16	S16
	SR	11.5	12	S16
	SR	12.5	12	S16
Interlaken 2	LR	3.125/6.25	24 (see <sup>[1]</sup> )	T-8
	SR	3.125/6.25	16	S-16/C-8
	SR	10.3125/11.5/12.5	8	S-16

**Note:**

1. When Interlaken 2 is greater than 20 lanes, all T-8 SERDES must run at the same rate.
2. The eight Backplane SR(C-8) lanes cannot be split between the two Interlaken interfaces.

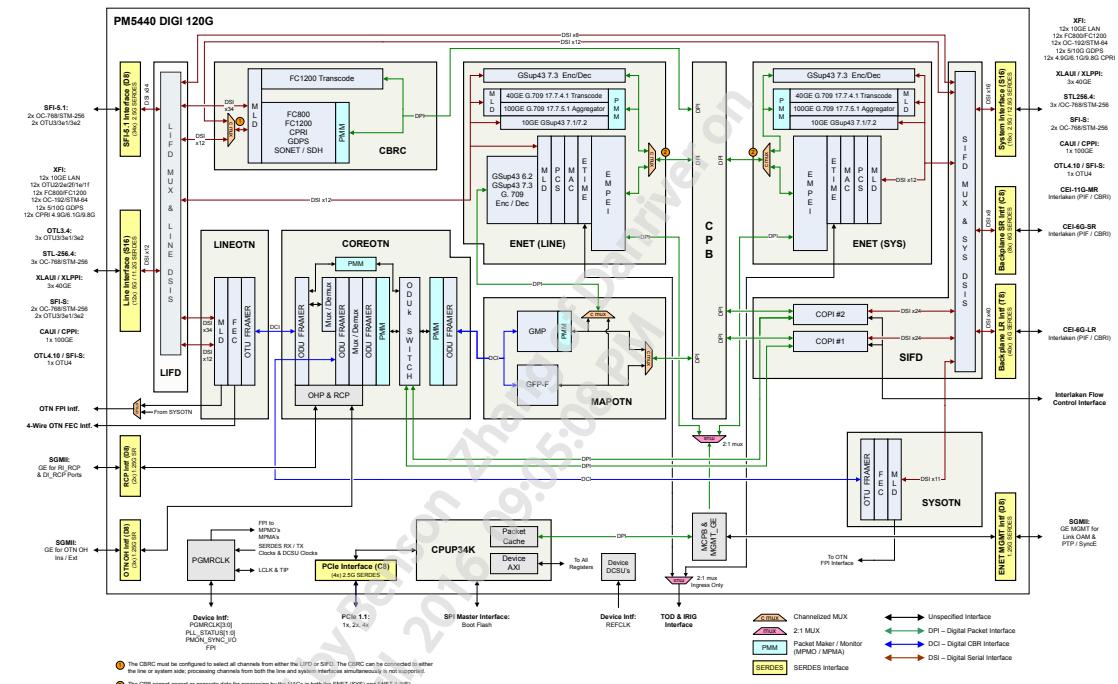
For each Interlaken Interface, arbitrary lane re-ordering is supported. As well, for each SERDES differential pair the P/N can be independently reversed.

## 5 Functional Description

### 5.1 Device Block Diagram

Please see [Figure 2](#) below.

[Figure 2](#) PM5440 Device Block Diagram



## 5.2

## Device Overview

The PM5440 DIGI 120G is a high-capacity, channelized, single-chip solution that provides unprecedented scalability and feature-integration. It enables a wide range of multi-service and packet transport over OTN applications, as described in the Product Overview [1]. The device line-side supports network and client interfaces with a set of 10 Gbps SERDES and two banks of 2.5 Gbps SERDES. Depending on the application, the device's system side supports client or system/backplane interfaces. The network and client interfaces can be logically aggregated via MLD (Multi-lane Distribution) functions to support network or client interfaces from sub-10 Gbps up to 100 Gbps. All network and client interfaces can be connected directly to optical modules.

Internally, the device's eight primary subsystems are interconnected with either a:

- DPI (Digital Packet Interface) that is used to transport packetized data;
- DCI (Digital CBR Interface) that is used to transport constant bit-rate data;
- DS1 (Digital Serial Interface) that is used to transport the serial data for each SERDES lane.

At the line side of the device, physical links are routed to the appropriate OTN or client adaptation layer subsystem through the LIFD (Line Interface Demultiplexer), providing a SERDES crossbar function. An additional DS1 connection between the LIFD and the SIFD (System Interface Demultiplexer) is provided to accommodate use of the system-side SERDES as line-side interfaces (as noted in [Section 4.5.1.3](#)). The OTN OTUk layer is implemented in the LINEOTN (Line OTN) subsystem. The ODUk layer is implemented in the COREOTN (Core OTN) subsystem. Ethernet termination is supported by the ENET (Ethernet) subsystem. All CBR clients are handled in the CBRC (Constant Bit Rate Client) subsystem.

Generally bit-transparent Ethernet client data and associated rate data are captured in a packetized data stream for use internally within the device. The bit-transparent Ethernet client data is packetized using PMC's OPSA (OTN Phase Signaling Algorithm) where the OPSA protocol is fully terminated and the stream is returned to its bit-transparent state. Ethernet clients can also be fully terminated into a packet data stream. These packet data streams, either packetized bit transparent clients or Ethernet packets from fully terminated Ethernet clients, are delivered into a CPB (Central Packet Buffer) before being routed to the appropriate destination. Note that packetized bit-transparent Ethernet is not supported over Interlaken.

An OTN stream entering the line side of the device first has its FEC processed and OTUk layer terminated by the LINEOTN subsystem. Various standard FECs are provided along with a proprietary PMC FEC (Swizzle) in the LINEOTN subsystem. The stream is then transferred to the COREOTN subsystem, which is responsible for ODUk and sublayer termination functions. The COREOTN provides multiple levels of OTN framing and multiplexing functions in addition to an external OTN overhead insert/extract interface. The ODUk switch within the COREOTN enables switching of ODUks and supports SNCP protection configurations. The resulting ODUk stream is then processed by the MAPOTN (OTN Mapper) subsystem or sent to the SIFD (System Interface) subsystem where an Interlaken interface is provided to the backplane. The MAPOTN subsystem supports GFP, GMP, BMP and AMP mapping functions across all possible OTN container and client rates. These de-mapped streams can be delivered to the Ethernet Line (ENET(LINE)) subsystem for termination and performance monitoring, or they can be packetized into the CPB.

In addition to the SIFD, the system side of the device integrates a second ENET(SYS) (Ethernet (System)) and the SYSOTN (System OTN) subsystem. This side of the device is responsible for driving an external backplane to a mate device, or interconnecting to an external system side device such as a network processor. An OTU4 stream entering the system side of the device can be processed by the SYSOTN

subsystem, which is responsible for G.709 RSFEC and OTUk layer processing before entering the COREOTN.

A CPB relays the independently configured packet streams back into Ethernet, CBR, or OTN adaptation functions. It is also capable of delivering these packets into the SIFD where the packet streams can be sent over an Interlaken interface to the backplane.

The external backplane interface is capable of glueless interoperation with network processors, switch fabrics, or mate connections. This flexibility addresses a variety of line card configurations and associated equipment protection schemes.

Supporting device initialization and run-time firmware processing, the CPU (CPUP34K) subsystem incorporates a number of external peripheral interfaces including PCIe and SPI. This subsystem has an internal fanout to all device registers via AMBA AXI and APB buses. A datapath is provided between the CPU subsystem and the Ethernet subsystems via a Management Packet Buffer. This enables inserted/extracted packets to be processed by firmware.

The following sections describe each subsystem in detail. These sections also define internal interfaces and describe device-level functions.

A summary of mappings supported by the device is provided in [Table 46](#) and [Table 47](#).

## 5.2.1

### DIGI 120G Channelization

The MPMA block is capable of supporting "N" channels depending on the subsystem in which the block is instantiated. See [Table 24](#) for details.

**Table 24 DIGI 120G Channelization**

Subsystem	# of Channels
LINEOTN	12
COREOTN Stage 1	12
COREOTN Stage 2 - 4	96
COREOTN Stage "N" (i.e. with SYSOTN enabled)	1
CBRC	12
ENET_LINE	12
MAPOTN	96

## 5.3 Internal Device Interfaces

This device uses three interfaces to facilitate the internal connections between the various functional blocks and subsystems of the device:

- Digital Serial Interface
- Digital CBR Interface
- Digital Packet Interface

The following subsections describe each of these interfaces in detail.

### 5.3.1 Digital Serial Interface (DSI)

The DSI is used to transfer data between any two internal functional blocks that operate in the transmit/receive clock domain. Mainly, this interface facilitates connections between the SERDES and functional blocks within the device. The DSI is a non-channelized unidirectional interface that operates at frequencies determined by the source or target SERDES. The DSI is a push-based interface with no back-pressure capabilities. For this reason, it requires no back-pressure signaling. Each SERDES within the device has two distinct DSI connections: one transmitting data toward the SERDES (transmit clock domain), and the other transmitting data away from the SERDES (recovered receive clock domain). In the case of the DSI transmitting data toward the SERDES, the DSI data rate must be synchronous with the target SERDES transmit clock. In the case of the DSI pushing data away from the SERDES, the DSI data rate must be synchronous with the source SERDES recovered receive clock. For an LOS condition, this interface also asserts a dedicated LOS signal.

The following subsystems are connected with a DSI:

- LIFD  $\iff$  SFI-5.1 interface
- LIFD  $\iff$  Line interface
- LIFD  $\iff$  ENET(LINE)
- LIFD  $\iff$  CBRC
- LIFD  $\iff$  LINEOTN
- LIFD  $\iff$  SIFD
- SIFD  $\iff$  System interface
- SIFD  $\iff$  Backplane SR interface
- SIFD  $\iff$  Backplane LR interface
- SIFD  $\iff$  ENET(SYS)
- SIFD  $\iff$  CBRC
- SIFD  $\iff$  SYSOTN

### 5.3.2 Digital CBR Interface (DCI)

The DCI is a channelized CBR interface that is able to realize framed internal CBR traffic. The maximum throughput of this interface is 134.4 Gbps. It can be configured to support 1 to 96 distinct channels. The DCI is able to pace its data rate with respect to the CBR rate by de-asserting a valid signal whenever the CBR rate is slower than the DCI nominal rate. Conversely, back-pressure is used by this interface in order to pace the data when it is delivered too quickly. Back-pressure is applied individually for each channel. There is no bus-level back-pressure provided within the interface. This interface is able to pace the data stream with respect to CBR traffic by using this combination of back-pressure and valid signals.

Along with the data, CBR characteristics such as SOF, FAS, and AIS are provided over additional signals. All of these signals are optional, and depend on the type of CBR traffic that is transmitted and received synchronously with the Channel ID.

The following subsystems are connected with a DCI:

- LINEOTN  $\iff$  COREOTN
- COREOTN  $\iff$  MAPOTN
- COREOTN  $\iff$  SYSOTN

### 5.3.3 Digital Packet Interface (DPI)

The DPI is a channelized packet interface that provides connectivity to the various packet-based subsystems internal to the device. Data on this interface is associated with a channel, and packets are delimited with a SOP/EOP indication. Each DPI port is defined to be either a master or a slave on the interface. The DPI port that initiates the transaction is defined as the master. The DPI port that receives the transaction is defined as the slave. Rate control between master and slave on the interface is controlled by a back-pressure signal asserted by the slave.

In order to send CBR or serial data across a DPI, data must be first packetized using a MPMA (Multi-channel Packet Maker), or de-packetized using a MPMO (Multi-channel Packet Monitor). See [Section 5.4.15](#) and [Section 5.4.16](#) for further information.

The following subsystems are connected with a DPI:

- CPB  $\iff$  CBRC
- CPB  $\iff$  MAPOTN
- CPB  $\iff$  ENET(LINE)
- CPB  $\iff$  SIFD
- CPB MGMT  $\iff$  ENET(LINE)
- CPB MGMT  $\iff$  ENET(SYS)
- CPB MGMT  $\iff$  CPUP34K
- CPB MGMT  $\iff$  ENET MGMT Interface
- COREOTN  $\iff$  SIFD
- MAPOTN  $\iff$  ENET(LINE)

## 5.4 OTN Phase Signaling Algorithm (OPSA)

### 5.4.1 Overview

Client and transport rate capture and its subsequent recovery is done in the device using the OTN Phase Signaling Algorithm. The OPSA strategy defines how rate information is captured for the various clients and transport containers. It also defines how rate information is transferred, translated, and subsequently used to recreate the client or transport rate for mapping, multiplexing, and retransmission. The fundamental building blocks of the OPSA are the:

- Multi-channel Packet Maker
- Multi-channel Packet Monitor
- Adaptive Clock Block

The following sections describe where these blocks are located within the device architecture, and how these blocks implement the OPSA strategy.

Rate information is captured and encoded by packetizing a stream using variable size packets. Packet makers capture rate information by looking at the arrival rate of data and by receiving justification information from demux and demap blocks. Rate information is recovered by receiving a packetized stream and converting it back into CBR data.

Packet monitors recover rate information by monitoring the sizes of incoming packets, producing a rate modulation stream that is sent to ACB blocks, and producing a rate stream suitable for use by GMP/AMP muxing and mapping blocks for justification control.

When de-mapping a client using OPSA, differences in the reference clock frequency between the source node where a client is mapped and the device where it is de-mapped can be determined. By determining this offset, the original client rate can be reproduced at the device for retransmission.

The primary OPSA operating modes are:

- OTN line-side rate capture from OTUK
- OTN line-side rate recovery to OTUK
- OTN system-side rate capture to OTUK
- OTN system-side rate recovery from OTUK
- Client line and system-side rate capture to OTUK
- GFP-F/TTT-mapped client line and system-side rate capture to OTUK
- Client line and system-side rate recovery from OTUK
- GFP-F/TTT-mapped client line and system-side rate recovery from OTUK
- Ethernet Tx MAC-sourced client rate capture to OTUK
- Ethernet Rx MAC-received client rate recovery from OTUK
- OTN line-side rate capture to COPI
- OTN line-side rate recovery from COPI
- Client line and system-side rate capture to COPI
- Client line and system-side rate recovery from COPI

In the following sections, when describing OPSA encoding of rates using variable sized packets, the letters "B" and "C" are used to describe the size of a nominal packet in bytes. "B" describes a nominal packet size for most operating modes. However, when both a client and a transport level of rate encoding or rate recovery are described for an operating mode, "B" is used to designate the nominal packet size for the transport rate operation and "C" is used to designate the nominal packet size for

the client rate operation. For example, "B" is used to describe the nominal packet size for an ODUK, while "C" is used to describe the nominal packet size for the client mapped within the ODUK. Similarly, " $T_B$ " is used for the packet decision period relating to a B-sized nominal packet, and " $T_C$ " is used for the packet decision period relating to a C-sized nominal packet.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:08 PM

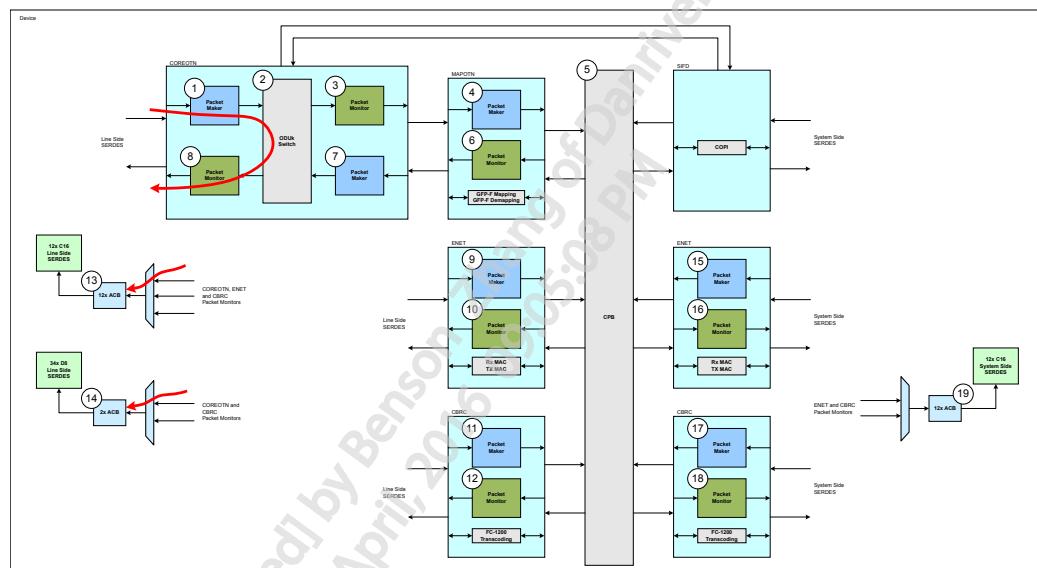
## 5.4.2

## OTN Line-side Rate Capture from OTUK

This operating mode is used to capture the rate of an ODUk/ODUflex received from an OTUk ingress on the device's line-side SERDES. This mode requires a single MPMA block. From the line-side SERDES, the OTUk is processed and demuxed down to the lowest order of ODUk/ODUflex that is requested for each constituent stream. These streams are then passed into the MPMA for packetization and rate capture. From the MPMA, the packetized streams are sent to the ODUk Switch before being directed to appropriate targets.

The signal flow for this operating mode is shown below in [Figure 3](#) as the red line from the line-side SERDES through the ingress MPMA in the COREOTN up to the ODUk Switch. The MPMA capturing the ODUk/ODUflex rate is labeled 1 and the ODUk Switch is labeled 2.

**Figure 3** Line-side Rate Capture from OTUk and Rate Recovery to OTUk



Within the MPMA, each ODUk stream is packetized to encode its data rate using modulations in packet size. The MPMA accumulates the amount of data arriving for each stream and uses a 2<sup>nd</sup> order DSM (Delta Sigma Modulator) to make B+/-1 sized packet decisions relative to a per-stream configurable nominal size of B bytes. Decisions are made at a per-stream configurable rational ratio of 311.04 MHz reference clock periods,  $T_B$ . The fractional component of  $T_B$  is defined as the ratio of two 24-bit integers. The rate is encoded compared to a nominal rate of  $B/T_B * 311.04$  MHz and the capture range is limited to B+/-1 byte packets for each stream. The MPMA packetizes each stream according to these packet size decisions, and sends the packetized data on to the ODUk Switch. The MPMA is capable of rate encoding up to 96 ODUk/ODUflex streams concurrently.

The channelized data path to the MPMA contains a side channel field driven by the ODUk de-multiplexers. This CnD field contains a byte count of data which would have been sent over the transport if the GMP m-byte granularity had allowed it. It is used as a secondary channelized input to the MPMA DSM to influence and smooth DSM decisions. This DSM input is used alongside the count of the number of bytes received on the data path, but is not accumulated by the DSM integrator. As such, the CnD field influences DSM decisions but, when removed, does not continue to influence future decisions. The extra word which was predicted by the CnD field will show up at the MPMA shortly after the CnD field is cleared, and decisions will continue to be made smoothly.

In the signal flow diagram for this section, the CBRC is shown as two blocks. It is drawn in this way to show that the CPB connects to both the line and system-side SERDES through the CBRC. However, there is only a single CBRC instance in the device.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:08 PM

### 5.4.3

### OTN Line-side Rate Recovery to OTUk

This operating mode is used to recover the rate of ODUk/ODUflex packetized streams after receiving them from the ODUk Switch. The OPSA blocks required for this mode are one MPMO block, and one ACB (Adaptive Clock Block) for each OTUk lane transmitted out the line-side. From the ODUk Switch, each packetized ODUk stream is passed to the MPMO. The MPMO restores the stream to CBR data and monitors the rate encoded in the stream's packet size modulations. The packet size modulations are used to recover the rate of each stream.

Depending on whether the ODUk/ODUflex is to be muxed to a higher order before being wrapped as an OTUk and transmitted, the rate encoding is used in one of two ways:

- If no muxing is requested for a stream, then rate encoding is used to control the ACB that is recreating the OTUk bitrate.
- If muxing is requested for a stream, then rate encoding is used to control the justification of the low order to higher order muxing of the stream by the GMP/AMP mux.

The signal flow for this operating mode is shown above in [Figure 3](#) as the red line from the ODUk Switch through the egress MPMO in the COREOTN and on to the line-side ACBs. The ODUk Switch is labeled 2, the MPMO recovering the ODUk/ODUflex rate is labeled 8, and the line-side ACBs are labeled 13 and 14.

In order to restore the packetized ODUk data to CBR form, the boundaries between each packet are removed for each stream. ODUk data arrives at the MPMO on a channelized bus, and is ordered according to a scheduler in the ODUk Switch. The CBR data leaving the MPMO is reordered according to a configurable calendar scheduler within the MPMO.

When an ODUk stream is not intended to be muxed to a higher order, the MPMO tracks the sizes of the packets as they pass through. Each packet's size is evaluated against a nominal size of B bytes and a  $\{+1,0,-1\}$  decision is extracted. The packet data is passed on towards the target SERDES that is used to transmit the OTUk. The sequence of extracted decisions forms a modulation stream that is directed to an ACB connected to this SERDES. This stream is used by the ACB in combination with a CSU to create a SERDES OTUk bit clock that is synchronized to the egress ODUk stream. A divided-down version of this clock is used to request OTUk data from the LINEOTN and COREOTN subsystems for transmission out of the line-side transmit SERDES.

When an ODUk stream is intended to be muxed to a higher order prior to transmission as an OTUk, the MPMO is still used to track the sizes of packets as they pass through. Each packet's  $\{+1,0,-1\}$  decision is extracted and used to form a Cn report that represents the number of bytes contained in the stream in a configurable reporting period. The Cn stream is sent to the GMP/AMP mux to control the justification of the low order ODUk/ODUflex into the higher order container. The Cn reporting period is a configurable rational ratio of 311.04 MHz reference clock periods,  $T_F$ , set to the first stage mux's multi-frame period. The fractional component of  $T_F/T_B$  is defined as the ratio of two 24-bit integers. The Cn rate reports are composed of an aggregation of a baseline sequence and a modulation sequence. The baseline sequence is  $B/T_B * T_F$ . The modulation sequence is an accumulation of  $T_F/T_B$  incoming packet sizes relative to B. These sequences are summed and losslessly low pass filtered before being used as Cn reports that are sent to the GMP/AMP mux. From the mux, the high order ODUk is sent to the OTUk framer and on to the target line-side SERDES. In this mode of operation, the higher order ODUk containers are created at nominal rates. For this reason, the ACB associated with the target SERDES is set to generate a nominal OTUk bitrate. In such a case, no modulation stream information is sent from the MPMO to the ACB.

In the signal flow diagram for this section, the CBRC is shown as two blocks. It is drawn in this way to show that the CPB connects to both the line and system-side SERDES through the CBRC. However, there is only a single CBRC instance in the device.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:08 PM

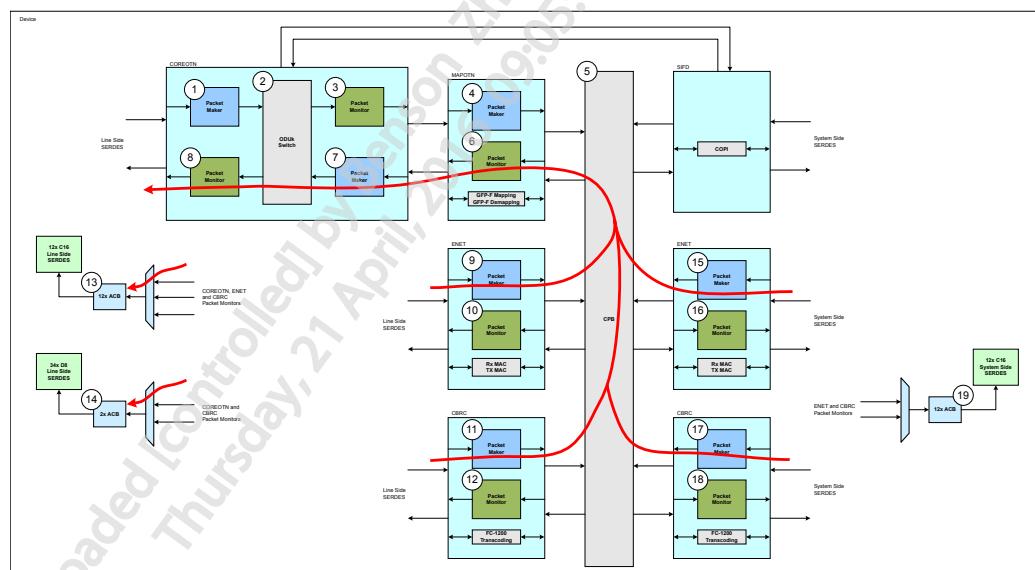
## 5.4.4

### Client Line / System-side Rate Capture to OTUK

This operating mode is used to capture the rate of an ingress client stream, map it into an ODUk, and then transmit it as an OTUk. Alternatively, the ODUk/ODUflex can be muxed up to a higher order before transmission as an OTUk. The OPSA blocks required for this mode are two MPMA blocks, two MPMO blocks, and one ACB. The data flow for this mode is from a client ingress to the CBRC or ENET on the line or system-side, to a MPMA, and on to the CPB. From the CPB, packetized client data is pushed to a MPMO, then to the client map, and on to a second MPMA. Following this, the packetized ODUk/ODUflex data is sent to the ODUk Switch, a second MPMO, and on to ODUk muxing before being transmitted out the line-side SERDES as an OTUk.

The signal flow for this operating mode is shown below in [Figure 4](#) as a red line from the line and system-side client SERDES, through the ingress MPMA in the line and system-side CBRC and ENET. From there, the signal flows to the CPB and on to the MPMO in the MAPOTN, to the egress MPMA in the COREOTN, on to the ODUk Switch, to the egress MPMO in the COREOTN and on to the line-side ACBs. The MPMA capturing the client rate is labeled 9, 11, 15 and 17, the CPB is labeled 5, the MPMO recovering the client rate is labeled 6, the egress MPMA capturing the ODUk/ODUflex rate is labeled 7, the ODUk Switch is labeled 2, the egress MPMO recovering the ODUk/ODUflex rate is labeled 8, and the line-side ACBs are labeled 13 and 14.

**Figure 4** Line and System-side Client Rate Capture to OTUk



The MPMA receiving the client CBR stream is used to packetize this data and encode its rate in a way that is similar to the procedure described in [Section 5.4.2](#). This MPMA accumulates the amount of data arriving for each stream and uses a 2nd order DSM to make B+-1 sized packet decisions relative to a per-stream configurable nominal size of B bytes. Decisions are made at a per-stream configurable rational ratio of 311.04 MHz reference clock periods,  $T_B$ . The packet size modulation encodes the rate compared to a nominal rate of  $B/T_B * 311.04$  MHz.

After passing through the CPB, the packetized stream is sent to a MPMO in the MAPOTN where it is monitored in a way that is similar to the procedure described in [Section 5.4.3](#). This MPMO is used to track the sizes of packets as they pass through. Each packet's {+1,0,-1} decision is extracted and used

to form a Cn report that represents the number of bytes contained in the stream in a configurable reporting period. The Cn stream is sent to the GMP/AMP mapper to control the justification of the client into the ODUK/ODUflex container. The Cn reporting period is a configurable rational ratio of 311.04 MHz reference clock periods,  $T_F$ , set to the ODUK/ODUflex frame period. The MPMO also restores the packetized client data to CBR form, removing the boundaries between each packet for each stream. Client data arrives at the MPMO on a channelized bus. Data is ordered according to a scheduler in the CPB. The CBR data leaving the MPMO to the GMP/AMP mapper is rescheduled according to a configurable calendar scheduler within the MPMO. This rescheduling allows for flexibility between MPMO ingress and egress, as well as for the differences in the ingress and egress bus rates.

Following the GMP/AMP mapper, ODUK/ODUflex data is pushed to a MPMA. This MPMA operates in a way that is similar to the procedure described in [Section 5.4.2](#). The MPMA accumulates the amount of data arriving for each stream and uses a 2nd order DSM to make C+/-1 sized packet decisions relative to a per-stream configurable nominal size of C bytes. This is done to encode the rate relative to a nominal rate of  $C/T_c * 311.04$  MHz within the packet size modulations. Next, the packetized ODUK is pushed to the ODUK Switch and on to a second MPMO. The streams pushed to the ODUK Switch are scheduled by a calendar scheduler within the MPMA. The streams subsequently pushed to the MPMO are scheduled by a scheduler within the ODUK Switch.

The second MPMO involved in this mode operates in a way that is similar to the procedure described in [Section 5.4.3](#). When an ODUK stream is to be muxed to a higher order prior to transmission as an OTUK, the MPMO generates a rate stream that is used by the GMP/AMP mux to justify the stream into the higher order container. The higher order ODUK containers and the OTUK are created at nominal rates and do not require MPMO control.

When an ODUK stream is not intended to be muxed to a higher order prior to transmission as an OTUK, the MPMO is used to create a modulation stream created from the {+1,0,-1} packet size decisions relative to C. This C port modulation stream is sent to the ACB associated with the SERDES targeted by the OTUK. This stream is used by the ACB to control the CSU creating the OTUK bitrate clock. The ACB is configured with a nominal phase accumulation of C bytes every  $T_c$  reference clock cycles. The modulation stream adjusts this rate to match the rate captured in the packet size encoding after scaling by the OTUK to ODUK ratio.

In the signal flow diagram for this section, the CBRC is shown as two blocks. It is drawn in this way to show that the CPB connects to both the line and system-side SERDES through the CBRC. However, there is only a single CBRC instance in the device.

## 5.4.5

### GFP-F/TTT-mapped Client Line / System-side Rate Capture to OTUk

This operating mode is used when TTT (Timing Transparent Transcoding) mapping a FC-1200 stream after it has been transcode encoded within the CBRC. TTT takes the FC-1200 PCS, encodes this into super-block groupings, and maps this as a client without using GFP-F idles to adapt the rate of the client to the rate of the ODUk/ODUflex container. Since no GFP-F idles are used, the client is bit synchronously mapped into the ODUk/ODUflex. Details of this mapping are found in G.709, 17.8.2. Since the rate of the client and the rate of the lowest level ODUk/ODUflex that contains the client are related by a fixed ratio, only the rate of the ODUk/ODUflex needs to be captured.

In this operating mode, the MPMA blocks labeled 11 and 17 in [Figure 4](#) are unused. A path through the FC-1200 transcode encoders bypasses these blocks. The ENET(SYS) and ENET(LINE) subsystems are also not used in this operating mode. Other aspects of the TTT operating mode are as described for standard GFP-F operations. Non-GFP-F aspects of this operating mode are described in [Section 5.4.4](#).

An alternative operating mode can be used to capture the rate of an ingress client stream, GFP-F map it into an ODUk/ODUflex, and then transmit it as an OTUk. The ODUk/ODUflex can also be muxed up to a higher order before transmission as an OTUk. The OPSA blocks required for this mode are two MPMA blocks, one MPMO block, and one ACB. The data flow for this mode is from a client ingress to the CBRC or ENET on the line or system-side to a MPMA and on to the CPB. From the CPB, packetized client data is pushed to the GFP-F mapper and on to the second MPMA. Following this, the packetized ODUk/ODUflex data is sent to the ODUk Switch, the MPMO, and on to ODUk muxing before being transmitted out the line-side SERDES as an OTUk.

The signal flow for this operating mode is similar to that shown in [Figure 4](#). The difference is that the red line showing the signal flow through the MPMO in the MAPOTN, labeled 6, passes instead through the GFP-F mapper. Packetized data from the CPB remains packetized as it enters the GFP-F mapper. A packet monitor is not required for depacketization prior to this. The remaining signal flow is as illustrated in the figure above. The packet sizes produced by the MPMA blocks labeled 9, 11, 15 and 17 are maintained into the GFP-F mapper. The client rate is encoded in these packet sizes relative to a nominal rate of  $C/T_c * 311.04$  MHz. These packet sizes are maintained as the payload of the GFP-F frames. GFP-F idles are inserted between these frames to enable asynchronous mapping of the client to the ODUk/ODUflex rate.

The functional operation of non-GFP-F aspects of this mode are described in [Section 5.4.4](#).

In the signal flow diagram for this section, the CBRC is shown as two blocks. It is drawn in this way to show that the CPB connects to both the line and system-side SERDES through the CBRC. However, there is only a single CBRC instance in the device.

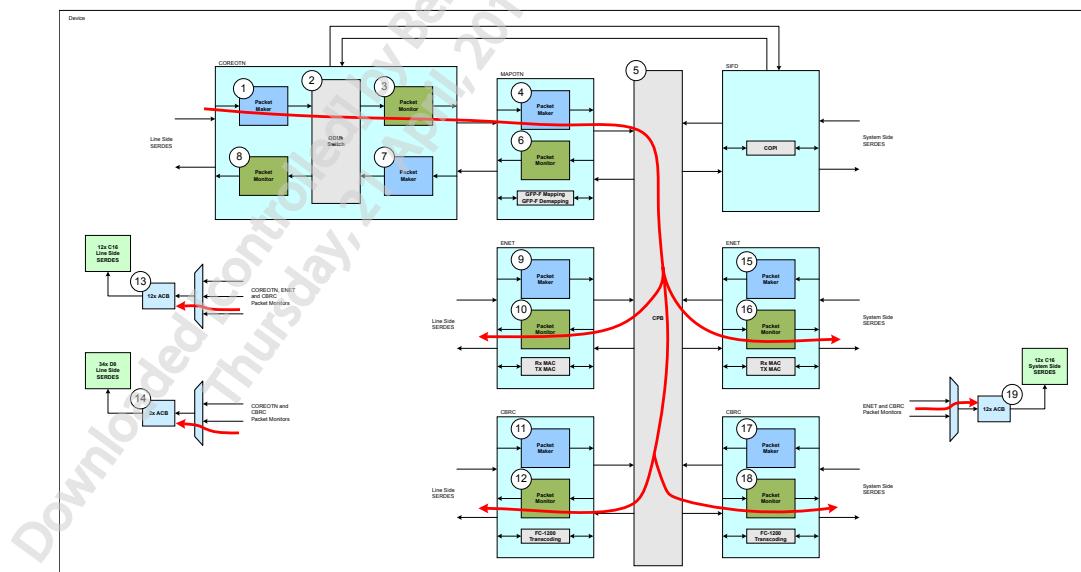
## 5.4.6

### Client Line / System-side Rate Recovery from OTUK

This operating mode is used to recover the rate of a client stream after receiving an OTUk ingress on the device's line-side SERDES. The OPSA blocks required for this mode are two MPMA blocks, two MPMO blocks, and one ACB for each client being transmitted out the line-side or system-side SERDES. The data flow for this mode is from the OTUk ingress to a MPMA and on to the ODUk Switch. From the ODUk Switch, the packetized ODUk is sent to a MPMO, then to the client de-map, and on to the second MPMA. Following this, the packetized client is sent to the CPB, the second MPMO, and on to the target client SERDES. In addition to pushing the datapath, the second MPMO sends a pair of modulation streams to an ACB associated with this SERDES. This mode can be divided into two main sections: one section concerned with capturing and recovering the rate of the lowest order of ODUk/ODUflex to be demuxed, and the second section concerned with capturing and recovering the rate of the client relative to its ODUk/ODUflex container.

The signal flow for this operating mode is shown below in [Figure 5](#) as a red line from the line-side SERDES, through the ingress MPMA in the COREOTN, on to the ODUk Switch, and to the ingress MPMO on the COREOTN. Next, the signal flows to the MPMA in the MAPOTN and on to the CPB, to the MPMO in the line and system-side CBRC and ENET and on to the line and system-side ACBs. The ingress MPMA capturing the ODUk/ODUflex rate is labeled 1, the ODUk Switch is labeled 2, the ingress MPMO recovering the ODUk/ODUflex rate is labeled 3, the MPMA capturing the client rate is labeled 4, the CPB is labeled 5, the MPMOs recovering the client rate are labeled 10, 12, 16 and 18, the line-side ACBs are labeled 13 and 14 and the system-side ACBs are labeled 19.

Figure 5 Line and System-side Client Rate Recovery from OTUK



The MPMA in the COREOTN is used to capture the rate of the lowest level of ODUk/ODUflex to be demuxed. This is done in an identical manner to the operation described in [Section 5.4.2](#). This MPMA is configured to make packet size decisions with a nominal size of B at a packet decision period of  $T_B$ . These  $B+/-1$  packet size decisions encode the rate of the ODUk/ODUflex relative to a nominal rate of  $B/T_B * 311.04$  MHz.

This rate encoding is used to adjust the recovered client rate to compensate for differences between the reference clock at the source where the client was originally mapped into the ODUk/ODUflex, and the local sink reference clock where the client is de-mapped from the ODUk/ODUflex. The ratio between these two reference clocks is reflected in the client bitrate clock generated at the target SERDES. From this MPMA, the packetized ODUk/ODUflex is passed to the ODUk Switch and on to a MPMO.

In this operating mode, the MPMO removes packet boundaries from the ODUk/ODUflex stream. ODUk/ODUflex stream data arrives at the MPMO on a channelized bus in an order determined by a scheduler in the ODUk Switch. The de-packetized CBR data leaving the MPMO is rescheduled according to a configurable calendar scheduler within the MPMO. The MPMO monitors the length of each ingress packet, evaluates it against a configurable nominal size of B, and extracts a  $\{+1,0,-1\}$  decision. Decisions are extracted per stream and placed as a side-channel on the egress data bus, accompanying the data of a corresponding channel. This side-channel is eventually used by the ACB associated with the target client SERDES generating the client bitrate.

From the MPMO, ODUk/ODUflex data is passed to the MAPOTN where the client streams are de-mapped. Extracted client data is then passed to a MPMA along with information about the justification that was detected within each ODUk/ODUflex frame. In this mode, the MPMA makes packet size decisions for the ingress client data based on this justification information. The DSM is set to make  $C \pm 1$  sized packet decisions relative to a per stream configurable nominal packet size of C bytes. Decisions are influenced only by the justification information from the MAPOTN and not by the actual arrival time of data at the MPMA. In this mode, the packet size decisions encode the rate of the client relative to the ODUk/ODUflex from which it was de-mapped. The packet decision period,  $T_C$ , is implied by knowledge of the client rate. However, since the decisions are not influenced by the actual arrival time of data at the MPMA, a  $T_C$  need not be configured in the MPMA. Conversely, a  $T_C$  is required at the ACB used to control the generated client bitrate at the target SERDES.

Packetized client data is pushed from the MPMA to the CPB, and is reordered according to a calendar scheduler within the MPMA. From the CPB, this packetized client data is pushed to an appropriate subsystem according to its stream type and intended target SERDES. The ENET subsystem is the destination for all CBR Ethernet streams. The CBRC subsystem is the destination for all other client streams. The line or system-side is the destination for each stream depending on the target SERDES for that client.

Within each subsystem, the packetized client stream enters a MPMO that operates in a manner that is very similar to the function described in [Section 5.4.3](#). Each packetized client stream is passed to a MPMO for restoration to CBR data and for monitoring of the rate encoded in its packet size modulations. Ingress packet sizes are compared against a configurable nominal packet size, C, and the resulting  $\{+1,0,-1\}$  C port modulation stream is sent to the ACB controlling the SERDES transmitting the client. Along with this client-based modulation stream, the ODUk-based modulation stream which was placed in the datapath side-channel is removed and sent as a second modulation stream, B port, to this same ACB. Following the MPMO, the CBR client stream is pushed towards the target SERDES.

At the target SERDES, the client bitrate is recovered using an ACB to control a CSU within the SERDES. The ACB is configured to create a nominal client bitrate based on the local reference clock if no information arrives on its C and B modulation stream ports. The C port modulation stream contains information that is used to adjust the client bitrate to account for mapping justification between the ODUk and the client by adjusting the amplitude of phase accumulation within the ACB. A client bitrate is produced given just this modulation stream and the nominal ACB configuration. This bitrate matches the original client bitrate if the source where the client was mapped had the identical reference clock to our local reference clock. The B port modulation stream contains information used to adjust the client bitrate to account for differences between the source reference clock and our local reference clock by adjusting the period of phase accumulation within the ACB. Since the nominal rate of the ODUk is known in

terms of our local reference clock, the modulation stream generated by the MPMA rate encoding contains information about the difference between our concept of a nominal rate and the nominal rate concept at the source. After processing to remove the difference between the expected ODUk rate and our  $B/T_B * 311.04$  MHz nominal rate, this modulation stream can be used by the ACB to adjust the client rate to account for the reference clock difference from the source. As such, the client bitrate from the source is recovered by setting the ACB to create a nominal client. The C port modulation stream is then used to adjust the ACB's phase accumulation to account for client justification relative to a nominal ODUk. The B port modulation stream is used to adjust the ACB's phase accumulation to account for difference between the source reference clock and our local reference clock.

In the signal flow diagram for this section, the CBRC is shown as two blocks. It is drawn in this way to show that the CPB connects to both the line and system-side SERDES through the CBRC. However, there is only a single CBRC instance in the device.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:08 PM

### 5.4.7 GFP-F/TTT-mapped Client Line / System-side Rate Recovery from OTUK

As described in [Section 5.4.5](#) and in G.709, 17.8.2, this operating mode is used to recover the rate of a TTT mapped client that was mapped without the use of GFP-F idles. This mode is used when de-mapping a FC-1200 stream prior to transcode decoding within the CBRC. In this case, since no GFP-F idles were used in the TTT mapping at the source, the client is now bit synchronously demapped from the ODUK/ODUflex. The rate of the client and the rate of the lowest level ODUk/ODUflex containing the client are related by a fixed ratio. In this operating mode, the ingress MPMO in the COREOTN (labeled 3) is used to directly control the ACB recreating the client rate. This MPMO sends a modulation stream to the ACB describing the rate of the ODUK/ODUflex. Since the client is a fixed ratio to this rate, no additional modulation stream is required by the ACB to recreate the client. Knowing the ratio between the client and the ODUK/ODUflex, the ACB is able to adapt the modulation information and adjust the ratio between this information and the client bitrate created.

The signal flow for this operating mode is very similar to the flow illustrated in [Figure 5](#). One difference is that, within the MAPOTN, the red line that represents the signal passes through the GFP de-mapper instead of through the MPMA (labeled 4). Additionally, the red line representing the signal does not pass through a MPMO block in the CBRC. Instead, it passes through the FC-1200 transcode decode block. The ENET subsystem is not used by this operating mode.

An alternative operating mode can be used to recover the rate of a GFP-F mapped CBR client stream after receiving an OTUk ingress on the device's line-side SERDES. The OPSA blocks required for this mode are one MPMA block, two MPMO blocks, and one ACB for each client being transmitted out the line-side or system-side SERDES. The data flow for this mode is from the OTUk ingress to a MPMA and on to the ODUK Switch. From the ODUK Switch, the packetized ODUk is sent to a MPMO, then to the de-map for the GFP-F mapped client. Following this, the GFP-F payload packets are sent to the CPB, to the second MPMO, and on to the target client SERDES. In addition to pushing the datapath, this MPMO sends a pair of modulation streams to the ACB associated with this SERDES. This mode can be divided into two main sections: one section concerned with capturing and recovering the rate of the lowest order of ODUk/ODUflex to be demuxed, and the second section concerned with capturing and recovering the rate of the GFP mapped client relative to its ODUk/ODUflex container.

The signal flow for this operating mode is very similar to the signal illustrated in [Figure 5](#). The difference is that, within the MAPOTN, the red line that represents the signal would pass through the GFP-F de-mapper instead of through the MPMA labeled 4. Otherwise, the line passing through the MPMA, MPMO and ACB blocks is as shown for the de-mapping of a GFP-F mapped client.

In the signal flow diagram for this section, the CBRC is shown as two blocks. It is drawn in this way to show that the CPB connects to both the line and system-side SERDES through the CBRC. However, there is only a single CBRC instance in the device.

## 5.4.8

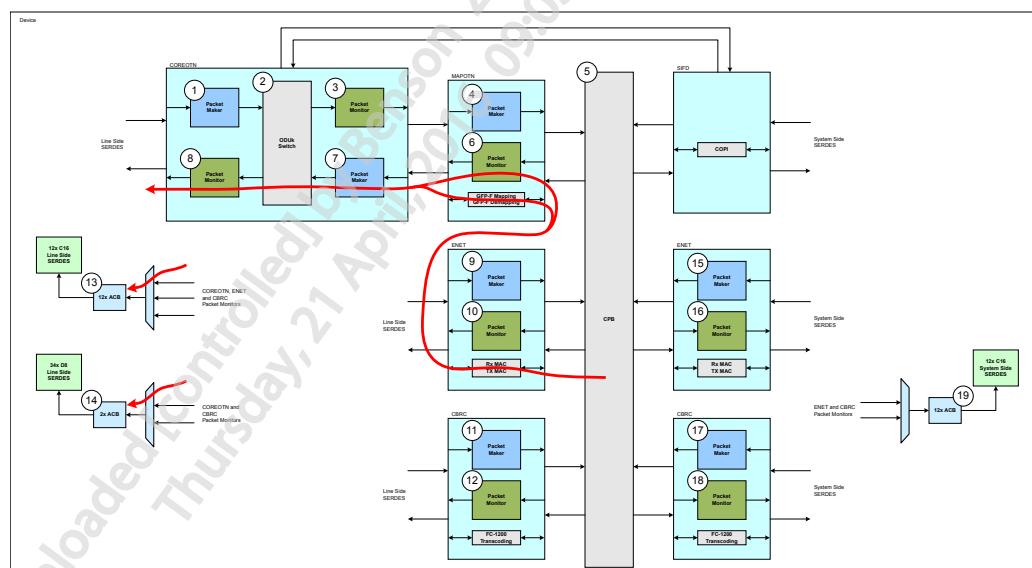
## Ethernet Tx MAC-sourced Client Rate Capture to OTUK

This operating mode is used to take packet data stored in the CPB, convert it to an Ethernet PCS stream using an ENET Tx MAC, map this stream into an ODUk/ODUflex and then transmit it as an OTUk. Alternatively, the ODUk/ODUflex can be muxed up to a higher order before transmission as an OTUk. The OPSA blocks required for this mode are one MPMA block, two MPMO blocks, and one ACB. The data flow for this mode is from the CPB to an ENET Tx MAC. From the Tx MAC, the Ethernet PCS is sent to the MAPOTN MPMO, to the GMP/AMP mapper and on to the COREOTN MPMA.

Alternatively, for a GFP-F mapped data stream, Ethernet PCS data sourced from the Tx MAC is sent to the GFP-F mapper and on to the COREOTN MPMA. In all mapping cases, GMP/AMP/BMP or GFP-F, ODUK/ODUflex data is sent to the COREOTN MPMA for rate capture. Following this MPMA, packetized ODUK/ODUflex data is sent to the ODUk Switch, to the second COREOTN MPMO, and on to the ODUK mux before being transmitted out the line-side SERDES as an OTUK.

The signal flow for this operating mode is shown below in [Figure 6](#) as a red line from the CPB, through the Tx MAC in the line-side ENET subsystem, through the MPMO and GFP-F mapper in the MAPOTN, to the egress MPMA in the COREOTN. From there, data flows on to the ODUK Switch, to the egress MPMO in the COREOTN and on to the line-side ACBs. The CPB is labeled 5, the MPMO recovering the client rate is labeled 6, the egress MPMA capturing the ODUK/ODUflex rate is labeled 7, the ODUK Switch is labeled 2, the egress MPMO recovering the ODUK/ODUflex rate is labeled 8 and the line-side ACBs are labeled 13 and 14.

**Figure 6 Line-side Tx MAC-sourced Client Rate Capture to OTUk**



The operation of the MPMA and MPMO within the CORETN and the ACBs controlling the transmit SERDES rates is described in [Section 5.4.4](#).

For GMP/AMP mapped clients, the MAPOTN MPMO is configured to produce a golden Cn along with the data which is transferred to the Tx mapper. Each value in the golden Cn stream represents the number of bytes contained in the datastream in a configurable reporting period. The Cn stream is sent to the GMP/AMP Tx mapper to control the justification of the client into the ODUK/ODUflex container. The Cn reporting period is a configurable rational ratio of 311.04 MHz reference clock periods,  $T_F$ , set to the ODUK/ODUflex frame period. For BMP mapped clients, the golden Cn rate is produced within

the Tx mapper through configuration of an ODUkSC; the MPMO generated Cn is unused. In both the GMP/AMP and BMP mapping cases, these golden Cn rates can be generated separate from the Tx MAC generated PCS because both the golden Cn and PCS rates are generated nominally based on the local DIGI reference clock.

In the signal flow diagram for this section, the CBRC is shown as two blocks. It is drawn in this way to show that the CPB connects to both the line and system-side SERDES through the CBRC. However, there is only a single CBRC instance in the device.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:08 PM

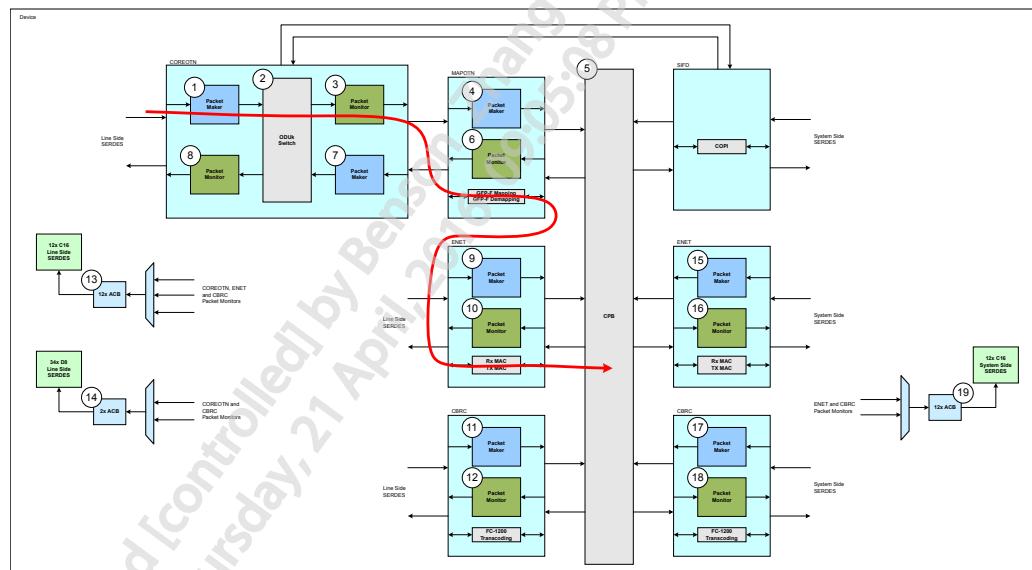
## 5.4.9

## Ethernet Rx MAC Received Client Packet Recovery from OTUK

This operating mode is used to receive Ethernet packets carried within an OTUK ingress on the device's line-side SERDES. The OPSA blocks required for this mode are one MPMA block and one MPMO block. The data flow for this mode is from the OTUK ingress to the MPMA and on to the ODUk Switch. From the ODUk Switch, the packetized ODUk is sent to the MPMO, then to either the GMP/AMP de-mapper or the GFP-F de-mapper. The GMP de-mapping path does not require a MPMA. It only requires simple packetization for merging with data from the GFP-F de-mapper. Following either de-mapper, the client data is sent to an Rx MAC. After this, the Ethernet packets are pushed to the CPB.

The signal flow for this operating mode is shown below in [Figure 7](#) as a red line from the line-side SERDES, through the ingress MPMA in the COREOTN and on to the ODUk Switch. From there, data flows to the ingress MPMO on the COREOTN, to the GFP-F de-mapper in the MAPOTN, to the Rx MAC in the ENET and on to the CPB. Since the GMP de-mapping path requires merging with data from the GFP-F de-mapper, only the path through the GFP-F de-mapper is shown. The ingress MPMA capturing the ODUk/ODUflex rate is labeled 1, the ODUk Switch is labeled 2, the ingress MPMO recovering the ODUk/ODUflex rate is labeled 3, the MPMA converting the CBR data to packetized data is labeled 4 and the CPB is labeled 5.

**Figure 7** Ethernet Rx MAC Received Client Packet Recovery from OTUK



The operation of the MPMA and MPMO within the COREOTN is described in [Section 5.4.6](#).

The MAPOTN is both for GMP/AMP de-mapping and for GFP-F de-mapping Ethernet data. The GMP/AMP de-mapper within the MAPOTN outputs channelized CBR data which is then converted to simple word length packets for merging with data from the GFP-F de-mapper. This merged data is then sent to the Rx MAC with the ENET for termination.

In the signal flow diagram for this section, the CBRC is shown as two blocks. It is drawn in this way to show that the CPB connects to both the line and system-side SERDES through the CBRC. However, there is only a single CBRC instance in the device.

### 5.4.10 Support for Ethernet OEO Applications with Rx PMON

The device supports Ethernet to Ethernet optical electrical optical applications using system and line side SERDES. The application allows the device to appear as a "bump on the wire".

The datapath through the device for 10 GE would be line side SERDES <= => ENET\_LINE (CBR) <= => CPB <= => ENET\_SYS <= => system side SERDES. The ENET CBR path is the same path that is used during OTN mapping for GSup43 clause 7.1 / G.709 clause 17.2.4 BMP mapping.

The datapath through the device for 40GE/100GE would be line side SERDES <= => ENET\_LINE (CBR) <= => CPB <= => ENET\_SYS (CBR) <= => system side SERDES. The ENET CBR path is the same path that would be used during OTN mapping for G.709 clause 17.7.4 / 17.7.5 GMP mapping.

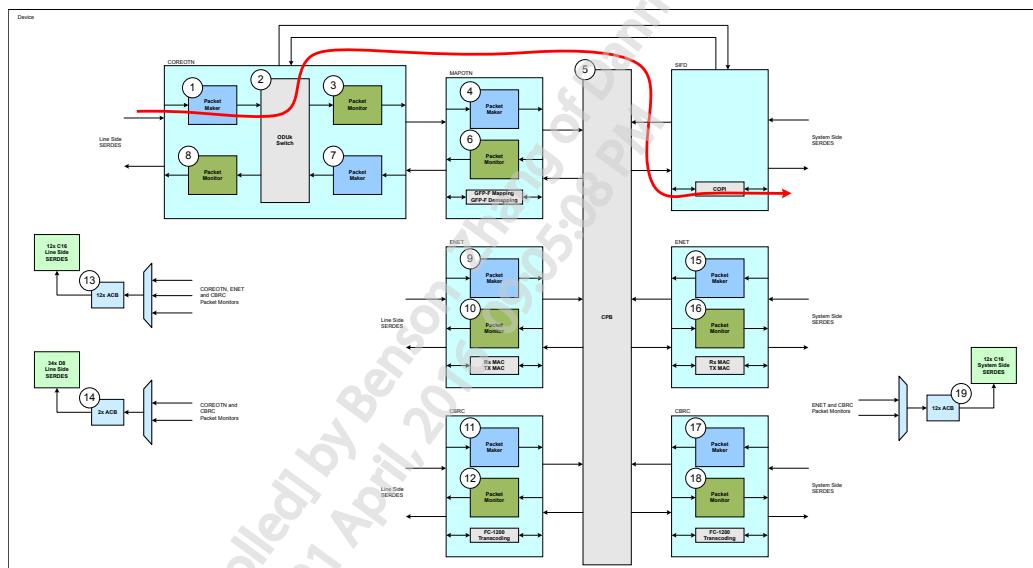
In all cases (10 / 40 / 100 GE) performance monitoring of the Ethernet client received on the line side SERDES is supported by the RX PCS/MAC in the ENET\_LINE block. Performance monitoring the Ethernet client received on the system side SERDES is supported by the RX PCS/MAC in the ENET\_SYS block. Performance monitoring of the Ethernet client transmitted on the line and system side SERDES is not supported.

### 5.4.11 OTN Line-side Rate Capture to COPI

This operating mode is used to capture the rate of an ODUk/ODUflex received from an OTUk ingress on the device's line-side SERDES, and to transport this ODUk/ODUflex over the COPI (Common OTN and Packet Interface). One MPMA block is required for this mode. From the line-side SERDES, the OTUk is processed and demuxed down to the lowest order of ODUk/ODUflex that is requested for each constituent stream. These streams are then passed into a MPMA for packetization and rate capture. From the MPMA, packetized streams are sent to the ODUk Switch before being forwarded to the COPI interface within the SIFD subsystem.

The signal flow for this operating mode is shown below in [Figure 8](#) as the red line from the line-side SERDES through the ingress MPMA in the COREOTN up to the ODUk Switch and on to the COPI interface. The MPMA capturing the ODUk/ODUflex rate is labeled 1 and the ODUk Switch is labeled 2.

[Figure 8](#) Line-side Rate Capture from OTUk to COPI



Within the MPMA, each ODUk stream is packetized in order to encode its data rate using modulations in packet size. The MPMA accumulates the amount of data arriving for each stream and uses a 2nd order DSM to make  $B+/-1$  sized packet decisions relative to a per-stream configurable nominal size of  $B$  bytes. Decisions are made at a per-stream configurable rational ratio of 311.04 MHz reference clock periods,  $T_B$ . The fractional component of  $T_B$  is defined as the ratio of two 24-bit integers. The rate is encoded compared to a nominal rate of  $B/T_B * 311.04$  MHz. The capture range is limited to  $B+/-1$  byte packets for each stream. The MPMA packetizes each stream according to these packet size decisions, and sends the packetized data on to the ODUk Switch. The MPMA block is capable of supporting "N" channels depending on the subsystem in which the block is instantiated. See [Table 24](#) for details.

The channelized data path to the MPMA contains a side channel field driven by the GMP de-multiplexers. This CnD field contains a byte count of data which would have been sent over the transport if the GMP m-byte granularity allowed it. It is used as a secondary channelized input to the MPMA DSM to influence and smooth DSM decisions. This DSM input is used alongside the count of the number of bytes received on the data path, but is not accumulated by the DSM integrator. As such, the CnD field influences DSM decisions but, when removed, does not continue to influence future decisions. The extra word which

was predicted by the CnD field will show up at the MPMA shortly after the CnD field is cleared, and decisions will continue to be made smoothly.

In the signal flow diagram for this section, the CBRC is shown as two blocks. It is drawn in this way to show that the CPB connects to both the line and system-side SERDES through the CBRC. However, there is only a single CBRC instance in the device.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:08 PM

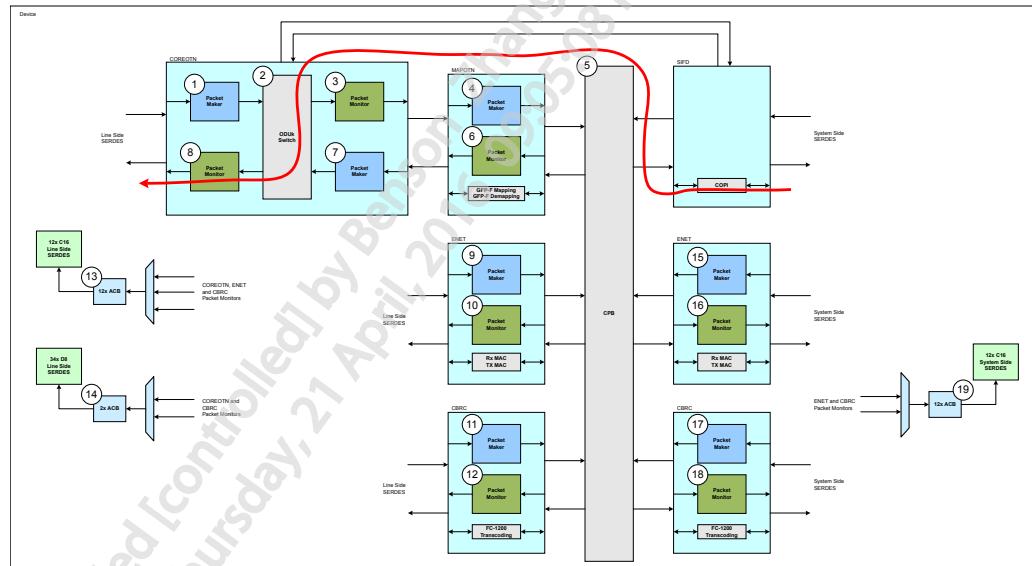
### 5.4.12 OTN Line-side Rate Recovery from COPI

This operating mode is used to recover the rate of ODUk/ODUflex packetized streams after they are received from the COPI. The OPSA blocks required for this mode are one MPMO and one ACB for each OTUk lane being transmitted out the line-side. From the COPI, each packetized ODUk stream is passed to the ODUk Switch and then to the MPMO for restoring to CBR data, and for monitoring of the rate encoded in its packet size modulations. The packet size modulations are used to recover the rate of each stream. Depending on whether the ODUk/ODUflex is to be muxed to a higher order before being wrapped as an OTUk and transmitted, the rate encoding is used in one of two ways:

- If no muxing is requested for a stream, then the rate encoding is used to control an ACB which is recreating the OTUk bitrate
- If muxing is requested for a stream, then the rate encoding is used to control the justification of the low order to higher order muxing of the stream by the GMP/AMP mux

The signal flow for this operating mode is shown below in [Figure 9](#) as the red line from the COPI to the ODUk Switch, through the egress MPMO in the COREOTN and on to the line-side ACBs. The ODUk Switch is labeled 2, the MPMO recovering the ODUk/ODUflex rate is labeled 8 and the line-side ACBs are labeled 13 and 14.

**Figure 9** Line-side Rate Capture from COPI to OTUk



In order to restore the packetized ODUk data to CBR form, the boundaries between each packet are removed for each stream. ODUk data arrives at the MPMO on a channelized bus in an order determined by a scheduler in the ODUk Switch. CBR data leaving the MPMO is reordered according to a configurable calendar scheduler within the MPMO.

When an ODUk stream is not intended to be muxed to a higher order, the MPMO tracks the sizes of the packets as they pass through. Each packet's size is evaluated against a nominal size of B bytes and a {+1,0,-1} decision is extracted. Packet data is passed on towards the target SERDES which is used to transmit the OTUk. The sequence of extracted decisions forms a modulation stream and is directed to an ACB connected to this SERDES. This stream is used by the ACB in combination with a CSU to create a SERDES OTUk bit clock that is synchronized to the egress ODUk stream. A divided-down version

of this clock is used to request OTUk data from the LINEOTN and COREOTN subsystems for transmission out of the line-side Tx SERDES.

When an ODUk stream is intended to be muxed to a higher order prior to transmission as an OTUk, the MPMO is still used to track the size of packets as they pass through. Each packet's  $\{+1,0,-1\}$  decision is extracted and used to form a Cn report that represents the number of bytes contained in the stream in a configurable reporting period. The Cn stream is sent to the GMP/AMP mux to control the justification of the low order ODUk/ODUflex into the higher order container. The Cn reporting period is a configurable rational ratio of 311.04 MHz reference clock periods,  $T_F$ , set to the first stage multiplexer's multi-frame period. The fractional component of  $T_F/T_B$  is defined as the ratio of two 24-bit integers. The Cn rate reports are composed of an aggregation of a baseline sequence and a modulation sequence. The baseline sequence is  $B/T_B \cdot T_F$ . The modulation sequence is an accumulation of  $T_F/T_B$  incoming packet sizes relative to B. These sequences are summed and losslessly low pass filtered before being used as Cn reports sent to the GMP/AMP mux. From the mux, the high order ODUk is sent to the OTUk framer and on to the target line-side SERDES. In this mode of operation, since the higher order ODUk containers are created at nominal rates, the ACB associated with the target SERDES is set to generate a nominal OTUk bitrate. No modulation stream information is sent from the MPMO to the ACB.

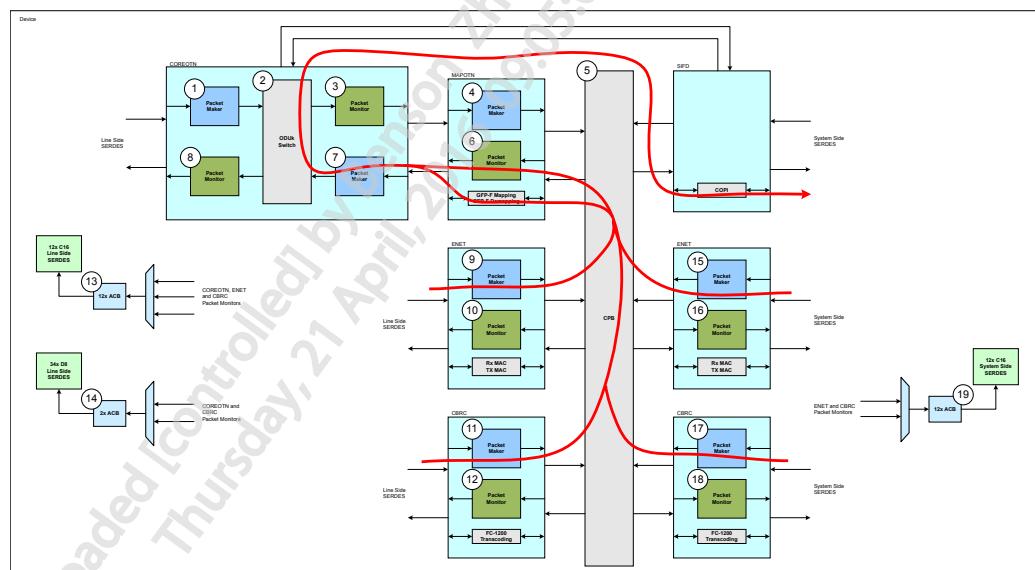
In the signal flow diagram for this section, the CBRC is shown as two blocks. It is drawn in this way to show that the CPB connects to both the line and system-side SERDES through the CBRC. However, there is only a single CBRC instance in the device.

#### 5.4.13 Client Line / System-side Rate Capture to COPI

This operating mode is used to capture the rate of an ingress client stream, map it into an ODUk and then transmit it over the COPI. The OPSA blocks required for this mode are two MPMA blocks, one MPMO block, and one ACB. The data flow for this mode is from a client ingress to the CBRC or ENET on the line or system-side to a MPMA and on to the CPB. From the CPB, packetized client data is pushed to a MPMO, then to the client map and on to the second MPMA. Following this, the packetized ODUk/ODUflex data is sent to the ODUk Switch and on to the COPI for transmission over the backplane fabric.

The signal flow for this operating mode is shown below in [Figure 10](#) as a red line from the line and system-side client SERDES, through the ingress MPMA in the line and system-side CBRC and ENET, to the CPB and on to the MPMO in the MAPOTN. From here, the signal flows to the egress MPMO in the COREOTN, to the ODUK Switch, and on to the COPI. The MPMA capturing the client rate are labeled 9, 11, 15 and 17, the CPB is labeled 5, the MPMO recovering the client rate is labeled 6, the egress MPMA capturing the ODUk/ODUflex rate is labeled 7 and the ODUK Switch is labeled 2.

**Figure 10** Line and System-side Client Rate Capture to COPI



The MPMA that receives the client CBR stream is used to packetize this data and encode its rate as described in [Section 5.4.2](#). The MPMA accumulates the amount of data arriving for each stream and uses a 2nd order DSM to make  $B+/-1$  sized packet decisions relative to a per-stream configurable nominal size of  $B$  bytes. Decisions are made at a per-stream configurable rational ratio of 311.04 MHz reference clock periods,  $T_B$ . The packet size modulation encodes the rate compared to a nominal rate of  $B/T_B * 311.04$  MHz.

After passing through the CPB, this packetized stream is sent to a MPMO where it is monitored as described in [Section 5.4.3](#). The MPMO is used to track the sizes of packets as they pass through. Each packet's  $\{+1, 0, -1\}$  decision is extracted and used to form a Cn report that represents the number of

bytes contained in the stream in a configurable reporting period. The Cn stream is sent to the GMP/AMP mapper to control the justification of the client into the ODUK/ODUflex container. The Cn reporting period is a configurable rational ratio of 311.04 MHz reference clock periods,  $T_F$ , set to the ODUk/ODUflex frame period. The MPMO also restores the packetized client data to CBR form, removing the boundaries between each packet for each stream. Client data arrives at the MPMO on a channelized bus in an order determined by a scheduler in the CPB. The CBR data leaving the MPMO to the GMP/AMP mapper is reordered according to a configurable calendar scheduler within the MPMO.

Following the GMP/AMP mapper, ODUk/ODUflex data is pushed to a MPMA. In this mode, the MPMA operates as described in [Section 5.4.2](#). The MPMA accumulates the amount of data arriving for each stream and uses a 2nd order DSM to make  $C+/-1$  sized packet decisions relative to a per-stream configurable nominal size of C bytes. This is done to encode the rate relative to a nominal rate of  $C/T_C * 311.04$  MHz within the packet size modulations. The packetized ODUk is next pushed to the ODUk Switch and on to the COPI. The streams pushed to the ODUk Switch are scheduled by a calendar scheduler within the MPMA, and the streams subsequently pushed to the COPI are scheduled by a scheduler within the ODUk Switch.

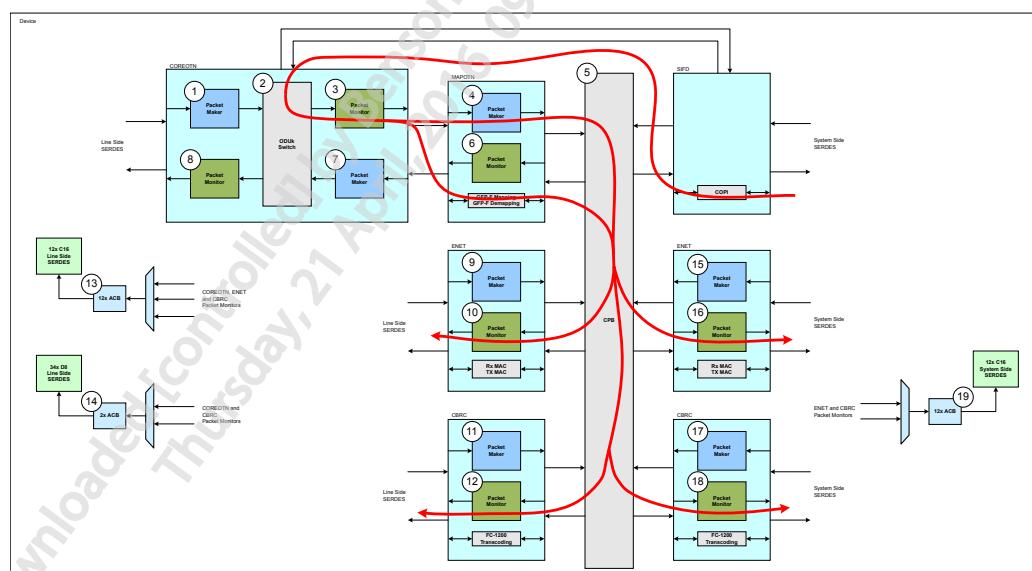
In the signal flow diagram for this section, the CBRC is shown as two blocks. It is drawn in this way to show that the CPB connects to both the line and system-side SERDES through the CBRC. However, there is only a single CBRC instance in the device.

### 5.4.14 Client Line / System-side Rate Recovery from COPI

This operating mode is used to recover the rate of a client stream after receiving an ODUk ingress on the COPI from the backplane fabric. The OPSA blocks required for this mode are one MPMA block, two MPMO blocks, and one ACB for each client being transmitted out the line or system-side SERDES. The data flow for this mode is from the COPI to the ODUk Switch. From the ODUk Switch, the packetized ODUk/ODUflex is sent to a MPMO, then to the client de-map and on to a MPMA for all de-mapped streams except GFP. Since GFP de-mapped client data is already packetized, this data bypasses the MPMA. Following this, the packetized client streams are sent to the CPB, to a second MPMO, and on to the target client SERDES. In addition to pushing the datapath, the second MPMO sends a pair of modulation streams to an ACB associated with this SERDES. The mode can be divided into two main sections: one section concerned with recovering the rate of the ODUk/ODUflex received from the COPI interface, and the second section concerned with capturing and recovering the rate of the client relative to its ODUk/ODUflex container.

The signal flow for this operating mode is shown below in [Figure 11](#) as a red line from the COPI to the ODUk Switch and on to the ingress MPMO in the COREOTN. Next, the signal flows to either the MPMA or GFP de-mapper in the MAPOTN and on to the CPB, to a second MPMO, and on to the target client SERDES. The ODUk Switch is labeled 2, the ingress MPMO recovering the ODUk/ODUflex rate is labeled 3, the MPMA capturing the client rate is labeled 4, the CPB is labeled 5, the MPMOs recovering the client rate are labeled 10, 12, 16 and 18, the line-side ACBs are labeled 13 and 14 and the system-side ACBs are labeled 19.

[Figure 11](#) Line and System-side Client Rate Recovery from COPI



The ingress MPMO in the COREOTN is used to recover the rate of the ODUk/ODUflex. At the source of the ODUk/ODUflex, the stream was packetized by making packet size decisions with a nominal size of  $B$  at a packet decision period of  $T_B$ . These  $B+/-1$  packet size decisions encode the rate of the ODUk/ODUflex relative to a nominal rate of  $B/T_B * 311.04$  MHz. This rate encoding is used to adjust the recovered client rate to compensate for differences between the reference clock at the source where the client was originally mapped into the ODUk/ODUflex, and the local sink reference clock where the

client is de-mapped from the ODUk/ODUflex. The ratio between these two reference clocks is reflected in the client bitrate clock generated at the target SERDES.

In this operating mode, the ingress MPMO removes the packet boundaries from the ODUK/ODUflex stream. ODUk/ODUflex stream data arrives at the MPMO on a channelized bus in an order determined by a scheduler in the ODUk Switch. De-packetized CBR data leaving the MPMO is reordered according to a configurable calendar scheduler within the MPMO. The MPMO monitors the length of each ingress packet, evaluates it against a configurable nominal size of B and extracts a  $\{+1,0,-1\}$  decision. These decisions are extracted per stream and placed as a side-channel on the egress data bus to accompany the data of a corresponding channel. This side-channel is eventually used by the ACB associated with the target SERDES that generates the client bitrate.

From the MPMO, ODUk/ODUflex data is passed to the MAPOTN where the client streams are de-mapped. For GMP/AMP/BMP mapped clients, the extracted client data is then passed to a MPMA along with information about the justification that was detected within each ODUK/ODUflex frame. In this mode, the MPMA makes packet size decisions for the ingress client data based on this justification information. The DSM is set to make  $C+/-1$  sized packet decision relative to a per-stream configurable nominal packet size of C bytes. Decisions are influenced only by the justification information from the MAPOTN and are not based on the actual arrival time of data at the MPMA. When operating in this mode, the packet size decisions encodes the rate of the client relative to the ODUK/ODUflex from which it was de-mapped. The packet decision period,  $T_C$ , is implied by knowledge of the client rate. However, since decisions are not influenced by the actual arrival time of data at the MPMA, a  $T_C$  need not be configured in the MPMA. Conversely, a  $T_C$  is required at the ACB used to control the generated client bitrate at the target SERDES. Alternatively, for GFP mapped clients, the extracted client data is extracted as packetized data and does not pass through the MPMA.

The packetized client data is pushed from the MPMA or GFP de-mapper to the CPB in an order determined by a calendar scheduler within the MPMA. From the CPB, this packetized client data is pushed to an appropriate subsystem according to its stream type and intended target SERDES. The ENET subsystem is the destination for all Ethernet streams, the CBRC subsystem is the destination for all other client streams, and the line-side or system-side is the destination for each stream depending on the target SERDES for that client. Within each subsystem, the packetized client stream enters a MPMO that operates in a manner that is similar to the procedure described in [Section 5.4.3](#). Each packetized client stream is passed to a MPMO for restoring to CBR data and for monitoring of the rate encoded in its packet size modulations. The ingress packet sizes are compared against a configurable nominal packet size, C, and the resulting  $\{+1,0,-1\}$  C port modulation stream is sent to the ACB that controls the SERDES that transmits the client. Along with this client-based modulation stream, the ODUk-based modulation stream which was placed in the datapath side-channel is removed and sent as a second modulation stream, B port, to this same ACB. Following the MPMO, the CBR client stream is pushed towards the target SERDES.

At the target SERDES, the client bitrate is recovered using an ACB to control a CSU within the SERDES. The ACB is configured to create a nominal client bitrate that is based on the local reference clock if no information arrives on its C and B ports. The C port modulation stream contains information that is used to adjust the client bitrate to account for mapping justification between the ODUk and the client by adjusting the amplitude of phase accumulation within the ACB. Given just this modulation stream and the nominal ACB configuration, a client bitrate is produced which matches the original client bitrate if the source where the client was mapped had the identical reference clock to our local reference clock. The B port modulation stream contains information used to adjust the client bitrate to account for differences between the source reference clock and our local reference clock by adjusting the period of phase accumulation within the ACB. Since the nominal rate of the ODUk is known in terms of our local reference clock, the modulation stream generated by the MPMA rate encoding contains information about the difference between our concept of a nominal rate and the nominal rate concept

at the source. After processing to remove the difference between the expected ODUk rate and our  $B/T_B * 311.04$  MHz nominal rate, this modulation stream can be used by the ACB to adjust the client rate to account for the reference clock difference from the source. For this reason, the client bitrate from the source is recovered by setting the ACB to create a nominal client. The C port modulation stream is then used to adjust the ACB's phase accumulation to account for client justification relative to a nominal ODUk. The B port modulation stream is used to adjust the ACB's phase accumulation to account for the difference between the source reference clock and our local reference clock.

In the signal flow diagram for this section, the CBRC is shown as two blocks. It is drawn in this way to show that the CPB connects to both the line and system-side SERDES through the CBRC. However, there is only a single CBRC instance in the device.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:08 PM

### 5.4.15 Multi-channel Packet Maker (MPMA)

The MPMA generates variable sized packets from a CBR stream or a channelized rate report stream. The size of the generated packets is used to convey timing information about the incoming CBR data or rate reports. The MPMA block is capable of supporting "N" channels depending on the subsystem in which the block is instantiated. See [Table 24](#) for further details. Both the MPMA ingress and egress datapaths are time-sliced, multi-channel interfaces. For an egress interface requiring a higher synchronous rate due to the packetization of the ingress CBR streams, the end of the previous packet terminates somewhere within a 48 byte word, and the subsequent packet starts word aligned at the following 48 byte word.

The MPMA contains the time-sliced datapath functionality required to packetize the multi-channel CBR streams according to packet size decisions which are generated by the MPMA\_DSP (Multi-channel Packet Maker DSP). Packetized data is stored within the MPMA in a multi-channel queue before being transferred to the egress interface. Back pressure is able to be generated by the ingress interface based on per-channel queue depths. Data is transferred from the per-channel queues based on either a calendar scheduler instantiated within the MPMA, or by an externally driven calendar schedule. The egress interface is able to receive per-channel back pressure and stop the push of back pressured channels when received.

An optional packet header can be inserted by the MPMA. Generally, this is performed for packets heading to the ODUk Switch, and not for packets destined for the CPB. Per-channel error indications are transferred from the ingress DCI to the egress DPI. Additionally, error indications can be signaled within the optional packet header.

The MPMA\_DSP contained within the MPMA is used to generate variable sized packet decisions based on input phase information of a client signal. The size of the generated packet indications are used to encode timing information about the incoming client signal. Packet size decisions are generated at a configurable decision period that is defined to be a rational ratio of a local reference clock. When used by the MPMA data path component, the packet size indications can be used to convey the captured timing information across a packet switch, across a packet fabric backplane, or between two nodes in a network.

The MPMA\_DSP has a tiled DSP architecture capable of making packet size decisions for "N" channels depending on the subsystem in which the block is instantiated. See [Table 24](#) for further details. The MPMA\_DSP uses a 2nd order DSM to make packet size decisions based on a measurement of the incoming data rate compared to a local reference clock. A DSM control loop makes +1, 0, -1 packet size decisions relative to a configurable nominal packet size. The incoming data rate can be augmented by a preview indication of data that has not yet been delivered to the MPMA. This indication is used to reduce waiting time jitter effects due to the payload word size used in the transport stream and the word sizes delivered from the SERDES.

The MPMA\_DSP is also able to use the DSM to make packet size decisions based on an incoming channelized rate report stream. The MPMA\_DSP converts the rate report value into a number of packet size decisions. The exact number of decisions is dictated by the ratio of the rate reporting period and the requested packet decision period. This ratio is defined to be a rational fraction.

### 5.4.16 Multi-channel Packet Monitor (MPMO)

The MPMO evaluates the size of packets and removes packet boundaries from an ingress data stream. The MPMO sits on the data path and, as channelized words pass through, counts the number of bytes in each packet per channel. For each channel, any difference between the packet size and a programmable nominal packet size is reported via a modulation stream side-channel. Additionally, incoming packet size information can be converted to a channelized rate report stream. The MPMO is able to process "N" channels depending on the subsystem in which the block is instantiated. See [Table 24](#) for further details. Both the MPMO ingress and egress datapaths are time-sliced, multi-channel interfaces. For an ingress interface requiring a higher synchronous rate due to packetization that is removed prior to transmission of the egress CBR streams, the end of the previous packet will terminate somewhere within a 48 byte word at packet boundaries. The subsequent packet starts word aligned at the following 48 byte word.

The MPMO contains the time-sliced datapath functionality required to depacketize the multi-channel packetized streams. The depacketized data is stored within the MPMO in a multi-channel queue before being transferred to the egress interface. Back pressure is able to be generated by the ingress interface based on per-channel queue depths. Data is transferred from the per-channel queues based on either a calendar scheduler instantiated within the MPMO, or by an externally driven calendar schedule. The egress interface is able to receive per channel back pressure and stop the push of back pressured channels when received.

The channelized rate report stream is produced by an MPMO\_DSP component instantiated within the MPMO. During an ODUflex hitless resizing ramp, time-stamps contained in the ingress packet header are used to detect the ramp event and track it by minimizing the upstream packet delay variation. Additionally, various channelized hold-off and FIFO centering modes are available using header time-stamps. The MPMO can back pressure specific channels until packets have aged enough to ensure a sufficient level of upstream buffer fullness. Time-stamps are also used as an indication of packet delay variation, allowing for upstream buffer centering through the adjustment of either the corresponding channel's FIFO zone indication or the rate report stream.

Headers which are optionally present in the ingress packets are removed by the MPMO. Per-channel error indications are transferred from the ingress to the egress interface. Additionally, error indications can be signaled on the egress interface from status contained within the optional packet headers.

The MPMO\_DSP contained within the MPMO is used to generate a channelized rate report stream based on packet size indications delivered by the MPMO datapath. The MPMO\_DSP has a tiled DSP architecture capable of generating rate report values for "N" channels depending on the subsystem in which the block is instantiated. See [Table 24](#) for further details. For each channel, the rate report values are generated at a configurable period, and are an aggregation of the packet size indications received within this period. The rate report values are a combination of a locally generated nominal rate sequence and the aggregation of incoming packet size modulations. The rate report generation period is determined using either a locally generated count, or by accumulating a specified number of packet size indications when the period at which these packet decisions were originally generated is known.

The MPMO\_DSP is used during ODUflex hitless resizing to track a ramping data rate and generate a rate report stream to match the incoming ramp. The incoming ramp is tracked using two inputs to the MPMO\_DSP. One input indicates when an extra word has been de-mapped and delivered to an upstream MPMA. The second input indicates when the upstream packet delay variation has changed as indicated by the MPMO monitored packet timestamps. Using these two inputs, the incoming rate is tracked and used to generate a ramping rate report that is in turn used by a downstream GMP mux to justify the data into the target transport container. Since, on average, the packet delay variation should be constant

if the input and output data rates are equivalent, a change in latency can be used to detect a change in data rate. Therefore, the fabric latency can be used to adjust the rate reports to track the changing data rate. The MPMO\_DSP uses a combination of the extra word indications and fabric latency to compensate for, and adjust to, the incoming ODUflex resizing ramp.

The MPMO\_DSP has functionality for upstream FIFO centering based on a FIFO zone input, as well as on the channelized packet time-stamp latencies. The centering algorithm slowly moves the FIFO towards center by slightly altering the output rate report value, causing the downstream block receiving these rate reports to alter its back pressuring and adapt its rate of data acceptance.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:08 PM

### 5.4.17 Packet Maker Header

When enabled, the packet maker inserts a header into the packet data stream in the SOP position. The format of the header word is shown below in [Figure 12](#).

**Figure 12** Packet Maker Header Format



The features associated with these fields are described in [Table 25](#) below.

**Table 25** Packet Maker Header Fields

Field	Bits	Description
Timestamp	31:16	<b>Reference Clock Timestamp</b> The 16-bit timestamp value comes from an internal timer running off the input reference clock. This field is used so that the age of packets (analogous to system latency) can be measured by a packet monitor receiving the packets.
RSV	15:10	<b>Reserved</b> This field is filled with zeros.
SQ	9:8	<b>Sequence Number</b> SQ is a 2-bit sequence number used to detect dropped packets. SQ is a binary counting number that increments with every packet per ODUk/ODUflex.
PPSI	7:6	<b>Previous Packet Size Indication</b> The Packet Sequence Number is a 2-bit number inserted by the packet maker to sequentially enumerate every packet in the generated packet stream. This allows a downstream packet monitor receiving these packets to determine if a packet has been dropped in the packet fabric. The Previous Packet Size Indicator bits indicate the size of the previous packet relative to the nominal packet size. This allows the monitor to insert a packet of the appropriate size if one is dropped. The PPSI bits of the packet header are set as follows: <ul style="list-style-type: none"> <li>• 11 : Previous packet is -1 the nominal packet size</li> <li>• 01 : Previous packet is +1 the nominal packet size</li> <li>• 00 : Previous packet is the nominal packet size</li> <li>• 10 : NULL packet size decision.</li> </ul>
CSI	5:1	<b>Client Status Indication</b> The CSI bits of the timestamp are set to 5'b00100 during normal operation. Optionally, the CSI can also be set to indicate a failure state on the ingress DCI or forced to a configurable value by per-channel FORCE_TO and FORCE_AWAY registers.
P	0	<b>Parity</b> P is a 1-bit odd parity field computed over the entire header.

The device supports insertion of a user-defined value in the CSI field of the OPF ODUk packet header as described in [Table 25](#). The value to insert is selectable on a per-channel basis from one of two user configured global values, a force-to value and a force-away value. The device implementation meets the requirement defined in the OIF ODUk OTN Over Packet Fabric Header standard.

PMC devices implemented prior to ratification of the OIF standard such as the PM5450 HyPHY 20Gflex support insertion of an arbitrary 5-bit value into the CSI and PPSI2 fields where the arbitrary value is per channel configurable. System designs utilizing a mix of devices should consider the implications of the CSI field configuration options that are supported.

As well, note that the OTN Over Packet Fabric Protocol (OFP) Implementation Agreement defined by the OIF specifies the encoding for the CSI field as shown below.

**Table 26 OFP CSI Field Definition**

CSI Value	Definition
0b000	Forces the switching fabric to select this ODUk/ODUflex even if the alternate stream is in the No Defects state
0b001	No Defects detected
0b010	Signal Degrade detected
0b011	Signal Fail detected
0b100	Server Signal Fail Detected
0b101	Reserved
0b110	Reserved
0b111	Forces the switching fabric to not select this ODUk/ODUflex regardless of the state of the alternate stream

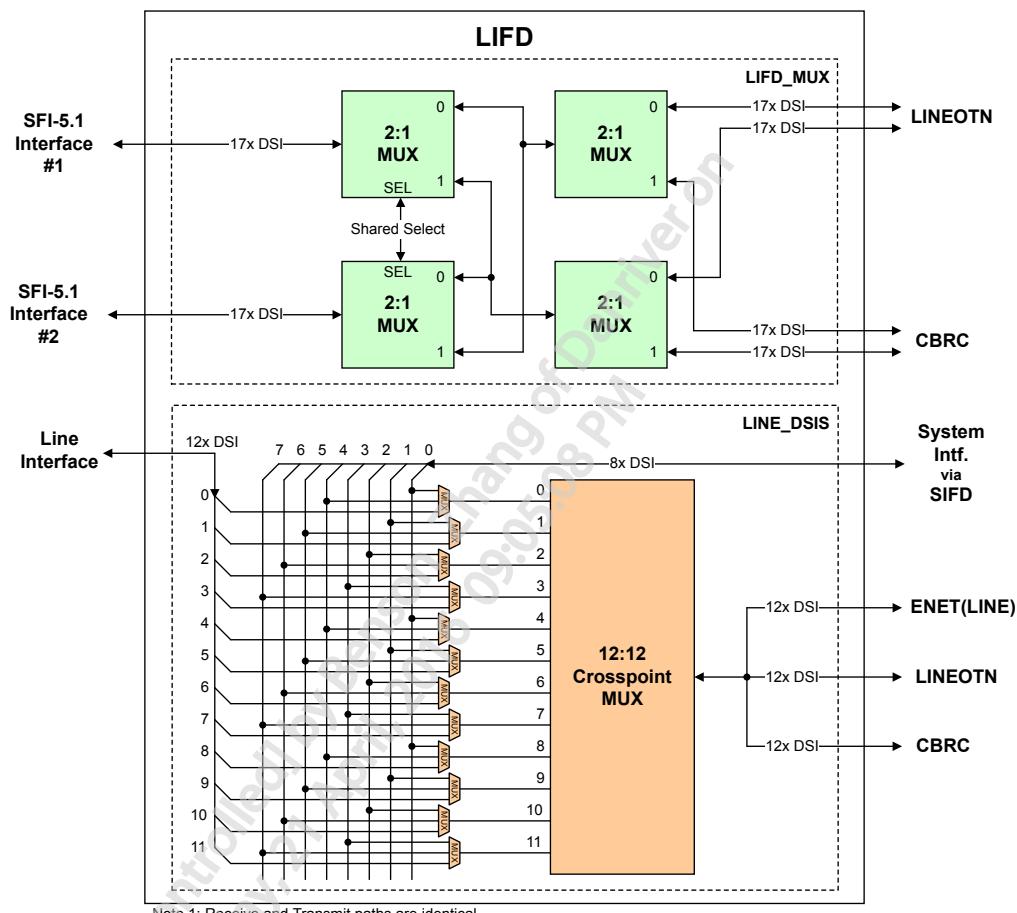
When one stage of demuxing is used (OTUk -> High Order ODUk -> Low Order ODUk), server signal fail will not propagate from the OTUk server and the alarm code for Server Signal Fail is not reported in the CSI field. Switch fabrics using the CSI field SSF indication to trigger a bulk switch of low order ODUk signals will not receive the required SSF indication.

Server Signal Fail will always result in signal fail for the low order ODUk signals provided at least one TCM for the receive framer for stage 2 is provisioned. Switch fabrics can use the signal fail indication to initiate protection switching of the ODUk signals. The switch controller will not receive the SSF indication for fault isolation purposes, and therefore, the SSF indication would need to be signalled to the controller out-of-band using the system control plane.

## 5.5 Line Interface (LIFD)

### 5.5.1 Block Diagram

Figure 13 LIFD Block Diagram



### 5.5.2 SFI-5.1 Interface Multiplexing (LIFD MUX)

The LIFD\_MUX is used to connect the CBRC and LINEOTN subsystems to the SFI-5.1 interface SERDES. The SFI-5.1 interface consists of two standard OIF SFI-5.1 interfaces that each contain 17 SERDES lanes operating at a rate of 2.5 Gbps. The SFI-5.1 interface SERDES are connected to each of these subsystems using 34 DS1 connections. Each DS1 connection consists of a clock and parallel data from a single SERDES lane.

The LIFD\_MUX provides the necessary multiplexing to connect either of the two SFI-5.1 interface SERDES to any of the two SFI-5.1 interfaces in the CBRC or LINEOTN subsystems. Software configures the destination of all SERDES lanes.

The LIFD\_MUX also selects the ACB signals driven by the CBRC and LINEOTN subsystems that are to be driven to the ACBs of the SFI 5.1 interface SERDES. The LOS signals are conditioned and sent to the CBRC and LINEOTN subsystems.

## 5.5.3

**Line Interface Multiplexing (LINE\_DSIS)**

The LINE\_DSIS is used to connect the CBRC, ENET(LINE), and LINEOTN subsystems to the Line interface SERDES or System interface lanes [7:0] SERDES. The Line interface consists of a bank of twelve SERDES lanes that operate within a range of 5 to 11.2 Gbps, and the System interface consists of a bank of sixteen SERDES lanes that operate within a range of 3.125 to 12.5 Gbps. Only the first eight lanes (i.e. lanes [7:0]) of the System interface SERDES can be connected to these subsystems via the SIFD\_MUX (see [Section 5.11.5.2.5](#)). The Line interface SERDES are connected to each of these subsystems using twelve DS1 connections. Each DS1 connection consists of a clock and parallel data from a single SERDES lane. The System interface lanes (i.e. lanes [7:0]) SERDES are connected to each of these subsystems using eight DS1 connections.

The LINE\_DSIS provides the necessary multiplexing so that any of the twelve Line interface SERDES or eight System interface (lanes [7:0]) SERDES can be connected to any of the twelve Line interface DS1 lanes in the CBRC, ENET(LINE), or LINEOTN subsystems. Any particular SERDES lane can only connect to one of the Line interfaces in the CBRC, ENET(LINE), or LINEOTN subsystems at a given time. Software configures the destination of all SERDES lanes.

The Line interface SERDES lanes and System interface SERDES lanes are first mapped to twelve internal lanes as seen by the pre-muxing before the 12:12 crosspoint mux in [Figure 13](#). Each internal lane can be sourced from three SERDES lanes, as shown in [Table 27](#). These twelve internal lanes can be connected to any of the twelve Line interface DS1 lanes in the CBRC, ENET(LINE), or LINEOTN subsystems.

**Table 27 LINE\_DSIS Internal Lane Sources**

Internal Lane	Line Interface SERDES Lane	System Interface SERDES Lane	System Interface SERDES Lane
0	0	0	4
1	1	1	5
2	2	2	6
3	3	3	7
4	4	0	4
5	5	1	5
6	6	2	6
7	7	3	7
8	8	0	4
9	9	1	5
10	10	2	6
11	11	3	7

**Note:**

1. The internal lanes are the inputs to the 12:12 crosspoint mux in [Figure 13](#).

[Table 27](#) describes restrictions for using eight system side S16 SERDES to connect to line side resources in the device. The purpose of optionally using eight of the system side SERDES on the line side is to allow a single PCB layout to support 1 x 100G CFP module or 3 x 40G CFP modules as a populate option. The 100G CFP (module 1) can operate as either a 100G module or a 40G module and connects to ten line side S16 SERDES. The two other 40G CFP modules (modules 2 and 3) connect to the eight system-side SERDES.

In this application, to guarantee 40G modules 2 and 3 can fit, module 1 when used as 40G, must be connected to four consecutive line-side S16 SERDES. For example, SERDES 0 to 3, 1 to 4, 2 to 5, or 8

to 11. Ordering within the 40G lanes does not matter. This is because each of the 8 system side SERDES lanes can be muxed into only 4 of the 12 slots with each slot separated by 4. If module 1 is assigned 4 consecutive slots, then it's not possible to block all 4 slots assigned to any given system side lane.

System S16 SERDES 7 to 0 must be used for 40G modules 2 and 3. System S16 SERDES 15 to 8 do not connect to the line side. Lanes for a 40G module need not be consecutive and ordering within the 40G does not matter.

Connecting System interface SERDES lanes to the LINE DSIS restricts the use of some Line interface SERDES lanes. For example, using lane 0 of the System interface SERDES prevents using lanes 0, 4, or 8 of the Line interface SERDES, depending on the internal lane that is connected to System interface SERDES lane 0.

The LINE\_DSIS also selects the ACB signals driven by the CBRC, ENET(LINE), and LINEOTN subsystems that are to be driven to the ACBs of the Line interface SERDES or System interface (lanes [7:0]) SERDES. The LINE\_DSIS allows selection of the C\_CKCTL to be driven to each of the SERDES ACBs from one of the following sources:

- ENET\_LINE Subsystem MPMO
- CBRC Subsystem MPMO
- COREOTN Line-side Egress MPMO
- COREOTN System-side Egress MPMO
- COREOTN Ingress MPMO feeding the de-mapper

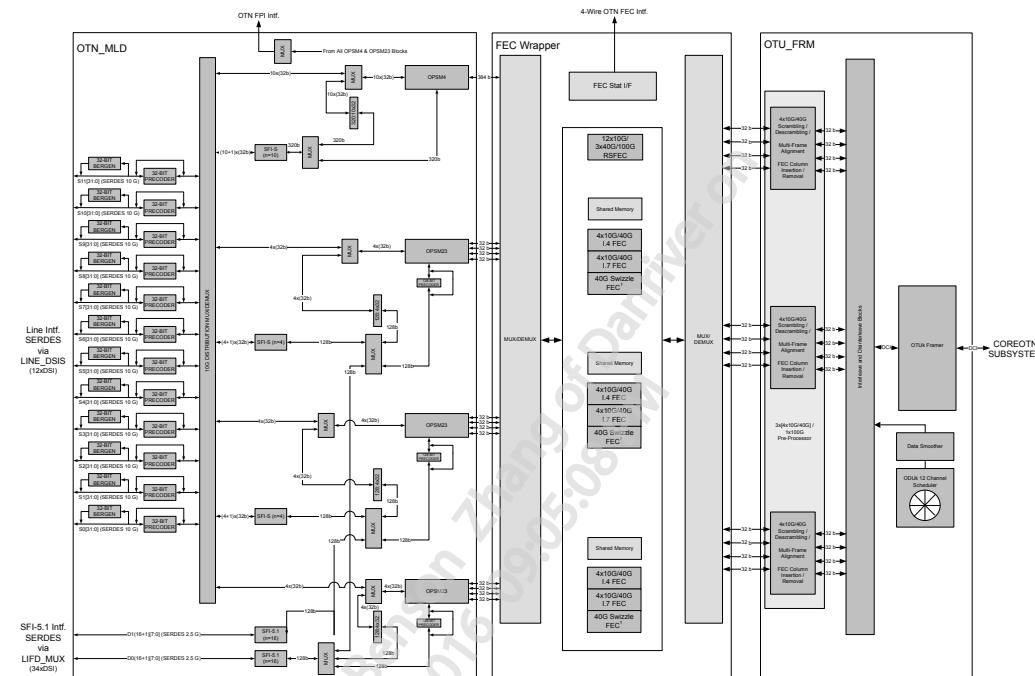
The B\_CKCTL inputs of the ACB are driven from the inputs of the COREOTN Ingress MPMO. The MPMO configures the channel mapping of the CKCTL/ACB ports. The LINE DSIS does not re-map the ACB ports from the subsystems when driving to the ACB ports of the SERDES.

LOS signals driven by the SERDES and from the device pins associated with the Line/System-side interfaces are conditioned and sent to the CBRC, ENET(LINE), and LINEOTN subsystems.

## 5.6 Line OTN (LINEOTN) Subsystem

### 5.6.1 Block Diagram

Figure 14 PM5440 LINEOTN Block Diagram



Note: 1. Three instances of 40G Swizzle can be used together to form a 100G Swizzle FEC.

### 5.6.2 Overview

The LINEOTN subsystem interfaces to the LIFD at the line side through a DSI (a combination of 10G and 2.5G links). The intention of this subsystem is to terminate the OTUK layer. It provides various functions including:

- OPSM<sub>n</sub> adaptation layers
- FEC encoding / decoding
- OTU (Optical Channel Transport Unit) termination
- Interfacing to the COREOTN subsystem through a multichannel DCI

The subsystem only supports Electrical OTN traffic from the SERDES line, which is compliant with G.709 OTUk frame sizes of 4080 columns x 4 rows (including FEC columns).

The LINEOTN subsystem crosses the data from the line (SERDES) clock domains into the 350 MHz system clock domain.

The major modes of operation within this block are governed by the muxing and demuxing of the SERDES links to the different MLD, FEC, and OTU functions. The LINEOTN subsystem complies with the applicable sub-clauses of G.709, G.798, G.806 G.975 and Gsup43 standards in processing data from the SERDES links through the MLD, FEC and the OTUk layers.

SFI-S (n) and SFI-5.1 are compliant with their respective OIF standards for interoperability. All ODUk/OPUk and client layer de-mapping protocols are handled outside this subsystem. In general, the subsystem provides three distinct layers of processing: MLD layer functions (including, for classification purposes, SFI layers), FEC layer functions, and OTUk termination layer functions.

There are two groups of SERDES links connecting to the MLD functions. The first group has 34 x 2.5G DS1 lanes, each 8-bits wide and running at approximately a 350 MHz SERDES line clock rate. The second group contains 12 x 10G DS1 lanes. Each 10G lane is 32-bits wide and runs at the SERDES line clock rate. The 10G DS1 lanes are multiplexed into several configurations in the MLD layer in order to support 10G, 40G, and 100G OTN rates. The 10G DS1 lanes are multiplexed into several configurations in the MLD layer in order to support 10G and 40G OTN rates. The 2.5G DS1 ports in the LINEOTN subsystem are only used to support 40G OTN rates.

The MLD layer functions perform multi-lane alignment (lane ordering and deskew) on 40G/100G multi-lane traffic. They also perform OTUk frame alignment as well as scrambling/descrambling. In addition, the MLD layer performs rate adaptation operations from the SERDES line rate to the FEC system clock rate.

The MLD functions can be split into three different rate categories: 10G, 40G, and 100G

- 10G rates do not require any MLD functions. 10G traffic is directly passed through from an XFI interface at the line side to the FEC and OTUk layers (and vice versa).
- 40G rates are transported using multiple lanes and can be obtained through 3 different MLD layer operations: SFI-S (n=4), SFI-5.1(n=16), and OTL3.4.
- 100G rates are transported using multiple lanes and can be obtained through different MLD layer operations: SFI-S(n=10), and OTL4.10.

The OTUk rates supported in each of the categories are summarized below:

- 10G rates: OTU1e, OTU1f, OTU2, OTU2e, OTU2f
- 40G rates: OTU3, OTU3e1, OTU3e2
- 100G rates: OTU4

The LINEOTN subsystem supports a combination of 12x10G FECs, 3x40G FECs, and 1x100 FEC:

- 10G FECs: RSFEC (G.709 FEC), I.4 FEC, I.7 FEC
- 40G FECs: RSFEC (G.709 FEC), I.4 FEC, I.7 FEC, Swizzle FEC
- 100G FECs: RSFEC (G.709 FEC), Swizzle FEC

The subsystem supports OTUk-based FEC statistics for OTU2, OTU3 and OTU4 traffic, and logical lane based FEC statistics for FECs that are able to report per-lane statistics for OTL3.4 or OTL4.10 formatted data. (RSFEC does not support lane based FEC statistics).

The LINEOTN subsystem interfaces to the COREOTN subsystem using a multi-channel DCI. In the receive (ingress) direction, the multi-channel DCI is formed from individual line side DS1 links using an Interleaver block and a programmable scheduler which determines which line side link (channel) drives the DCI during a given clock cycle. In the transmit direction, the scheduling of clients on the DCI bus is dictated to the LINEOTN subsystem by the COREOTN subsystem, and individual line side links are disassembled from the incoming DCI using a De-Interleaver block.

The following subsections highlight key features of the sub-blocks within the LINEOTN subsystem.

## 5.6.3 Features

### 5.6.3.1 Atomic Functions

The following diagrams illustrate the atomic functions supported by the LINEOTN. Line-side loopbacks are implemented in the SERDES.

Figure 15 Atomic Functions Supported By OPSMnk (OTL) Datapaths

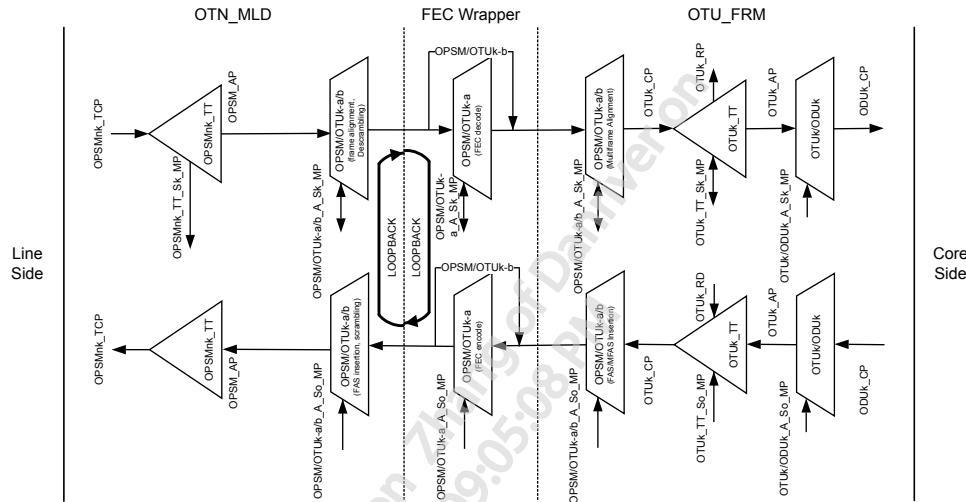
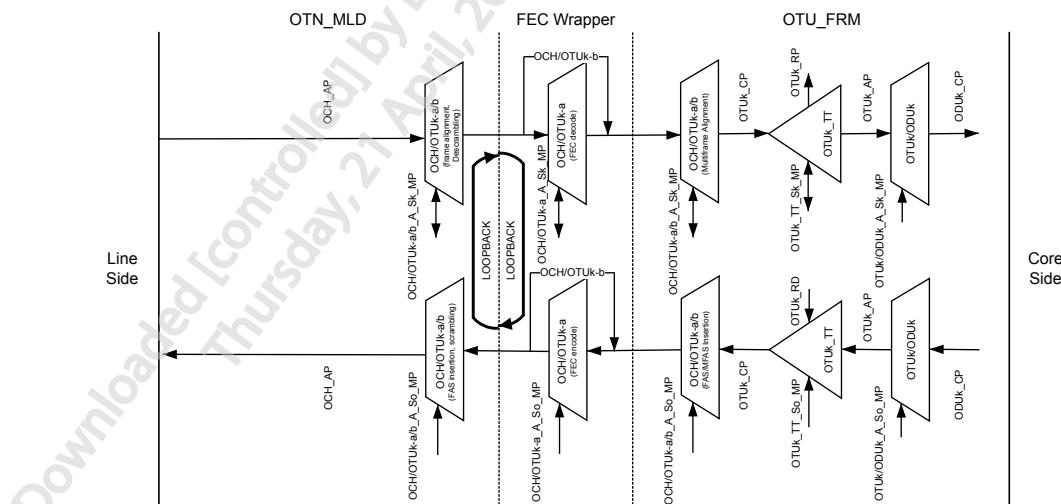


Figure 16 Atomic Functions Supported By Non-OPSMnk (Non-OTL) Datapaths



### 5.6.3.2 Multi-lane Distribution (MLD) Features

The OTN\_MLD block implements MLD transmit and receive functions.

Table 28 lists the supported and configured datapaths through the MLD subsystem.

Table 28 LINEOTN MLD Datapaths

Modes	Max Number of Paths	SERDES Interface #SERDES Lanes Per Path, Rate Per Lane, Width Per Lane	SFI Module in the path	OPSM23/OPSM4 Interface Path Interface, #Interface Lanes Per Path, Rate Per Lane, Width Per Lane	OPSM23 #	MLD Datapath
OTU2	3x4	1 , 10G , 32b	0	Line/OTU2 , 1 , 10G , 32b	1,2,3	SERDES $\iff$ (optional 10G DPSK Precoder) $\iff$ OPSM23
OTU3	3	4 , 10G , 32b	0	Line/OTL3.4 , 4 , 10G , 32b	1,2,3	SERDES $\iff$ (optional 10G DPSK Precoder) $\iff$ OPSM23
OTU3	2	4+1 , 10G , 32b	1	SFI/OTU3 , 1 , 40G , 128b	1,2	SERDES $\iff$ SFI-S(n=4) $\iff$ (optional 40G DPSK Precoder) $\iff$ OPSM23
OTU3	2	4+1 , 10G , 32b	1	Line/OTL3.4 , 4 , 10G , 32b	1,2	SERDES $\iff$ (optional 10G DPSK Precoder) $\iff$ SFI-S(n=4) $\iff$ OPSM23
OTU3	2	16+1 , 2.5G , 8b	1	SFI/OTU3 , 1 , 40G , 128b	1,2 or 2,3	SERDES $\iff$ SFI-5.1(n=16) $\iff$ (optional 40G DPSK Precoder) $\iff$ OPSM23
OTU3	2	16+1 , 2.5G , 8b	1	Line/OTL3.4 , 4 , 10G , 32b	1,2 or 2,3	SERDES $\iff$ (optional 10G DPSK Precoder) $\iff$ SFI-5.1(n=16) $\iff$ OPSM23
OTU4	1	10 , 10G , 32b	0	Line/OTL4.10 , 10 , 10G , 32b	N/A	SERDES $\iff$ (optional 10G DPSK Precoder) $\iff$ OPSM4
OTU4	1	10+1 , 10G , 32b	1	SFI/OTU4 , 1 , 100G , 320b	N/A	SERDES $\iff$ SFI-S(n=10) $\iff$ OPSM4
OTU4	1	10+1 , 10G , 32b	1	Line/OTL4.10 , 10 , 10G , 32b	N/A	SERDES $\iff$ (optional 10G DPSK Precoder) $\iff$ SFI-S(n=10) $\iff$ OPSM4

Two methods are used for MLD:

- Governed by OIF SFI-5.1 / SFI-S protocols, an additional lane is used to convey the skew of the lanes with respect to each other at the receiver end. The receiver uses this information to deskew the received lanes.
- The transmit data is distributed over several physical lanes, using one or more logical lanes for each physical lane. 40G traffic is formatted as OTL3.4: four logical lanes are assigned to four physical 10G lanes. 100G traffic is formatted as OTL4.10 traffic: 20 logical lanes are assigned to 10 physical 10G lanes, with two logical lanes for each physical lane.

The OTN\_MLD block:

- Provides OPSMnk Trail Termination (multi-lane alignment) according to the G.709 and G.798 standards:
  - In the receive path, the OTN\_MLD provides: (physical lane to logical lane) bit dis-interleaving, OTL frame alignment, OTL lane alignment recovery, multi-lane marker removal / OTUK FAS recreation, lane marker and delay processing (lane deskew) and 16-byte block multiplexing (16-byte interleaving).
  - In the transmit path, the OTN\_MLD provides: 16-byte block distribution and rotation (16-byte lane dis-interleaving), multi-lane marker insertion and (logical lane to physical lane) bit interleaving.
- Provides SFI-based multi-lane alignment.

- Provides OPSMnk-based (OTL-based) multi-lane alignment.
- Provides OPSMnk/OTUk adaptation functions of frame alignment in the receive path according to the G.709 and G.798 standards.
- Optionally, provides the OPSMnk/OTUk adaptation functions of scrambling / de-scrambling according to the G.709 and G.798 standards.

### 5.6.3.3 FEC Features

The LINEOTN subsystem supports several FEC Encode and Decode blocks. The following FECs are supported in the LINEOTN subsystem:

- FECs that operate on 10G datapaths: I.4 FEC, I.7 FEC, RSFEC.
- FECs that operate on 40G datapaths: I.4 FEC, I.7 FEC, RSFEC, Swizzle FEC.
- FECs that operate on 100G datapaths: RSFEC, Swizzle FEC.

All FEC types in the LINEOTN are capable of reporting corrected zeroes, corrected ones, and uncorrectable symbol error counts. These counts are accumulated using 1-second performance monitors (PMONs) and may be accessed via PCBI read access operations. PMONs may be updated using a (nominally) 1-second LCLK pulse provided by top-level logic to the LINEOTN subsystem or by writing to registers in the LINEOTN PCBI.

**Table 29 Statistics Per FEC**

Statistics	Statistic Description
UNCORRECTABLE_SYMBOL_COUNTER_VALUE	The 32-bit uncorrectable symbol counter value bits contain the count of the number of uncorrectable symbol errors detected by the FEC decoder since the last accumulation interval. This counter saturates when the maximum count value is reached.
FEC_CORRECTED_ZEROS_COUNTER_VALUE	The 32-bit FEC corrected zeros counter value bits contain the count of the number of corrected zeros detected by the FEC decoder since the last accumulation interval. This counter saturates when the maximum count value is reached.
FEC_CORRECTED_ONES_COUNTER_VALUE	The 32-bit FEC corrected ones counter value bits contain the count of the number of corrected ones detected by the FEC decoder since the last accumulation interval. This counter saturates when the maximum count value is reached.

All FECs except RSFEC in the LINEOTN can provide FEC statistics to an off-chip CPLD/FPGA via the device's 4-wire OTN FEC statistics interface. This allows external circuitry to control the analog levels of off-chip optical modules in order to reduce line errors in a closed loop fashion. Logical lane statistics communicated across this 4-wire interface may be summed by the external circuitry to obtain stream-based (OTUk based) statistics.

**Note:** Both logical lane and stream based statistics may be read via device registers. The use of the 4-wire FEC statistics interface is optional.

The modes supported by the 4-wire interface for each FEC type are summarized in [Table 30](#).

**Table 30 Supported 4-Wire Interface Modes**

FEC Type	10G OTU2	40G OTL3.4	40G OTU3	100G OTL4.10	100G OTU4
RSFEC	No	No	No	No	No
I4 FEC	No	Yes	Yes*	N/A	N/A
I7 FEC	No	Yes	Yes*	N/A	N/A
Swizzle	N/A	Yes	Yes*	Yes	Yes*

**Note:**

1. Yes\* denotes OTUk stream statistics are obtained by summing statistics reported over multiple rows of the 4-wire table.

The device supports these additional features:

- Decoder Disable:
  - Disables the FEC decoder while preserving the FEC latency
  - Traffic is passed through without correction
  - The FEC decoder will not report correctable error statistics in this mode.
    - It is not possible to preserve latency, disable correction, yet gather correction statistics.
  - Specific FEC limitations:
    - I.7: Decoder disable is not available
    - I.4: Far-end encoder must not be in bypass
- True Bypass
  - Provides maximum latency reduction by routing traffic around the FEC
  - Available on the encoder and decoder
  - Because I.4 and Swizzle FEC modify the OTU framer structure, the bypass configuration must match the incoming stream
    - If the incoming stream is encoded, the true bypass is not available
    - If the incoming stream has bypassed the encoder in the upstream device, the true bypass must be enabled
  - The remaining FEC's that do not modify the frame structure (i.e. RS-FEC and I.7 FEC) may enable the decoder true bypass regardless of the state of the upstream encoder

#### 5.6.3.4 OTU Features

The OTU\_FRM:

- Optionally, performs the OPSMnK/OTUk adaptation function of scrambling / de-scrambling according to the G.709 and G.798 standards.
- Optionally, performs the OPSMnK/OTUk adaptation function of multi-frame alignment in the receive path according to the G.709 and G.798 standards.
  - Performs OTUk Trail Termination according to the G.709 and G.798 standards:
    - In the receive path, OTU\_FRM provides: SMOH (TTI, RES, BDI, IAE, BEI/BIAE and BIP8), RES and GCC0 extraction.
    - In the transmit path, OTU\_FRM provides: SMOH (TTI, RES, BDI, IAE, BEI/BIAE and BIP8), RES and GCC0 insertion.
- Performs OTUk/ODUk adaptation according to the G.709 and G.798 standards:
  - In the receive path, OTU\_FRM provides: ODUk FAS and MFAS generation, OTUk to ODUk frame extraction, ODUk-LCK / ODUk-AIS generation and normal signal / ODUk-LCK / ODUk-AIS selection.

- In the transmit path, OTU\_FRM provides: ODUk-LCK generation, normal signal / ODUk-LCK selection and ODUk to OTUk frame mapping.

### 5.6.3.5 Defects, Defect Correlations, Consequential Actions and Performance Monitoring Events

Defects, defect correlations, consequential actions and performance monitoring events are treated on a layer-by-layer basis for the OTN protocols. In the LINEOTN subsystem there are four functional layers:

- OPSMnK Trail Termination Layer
- OPSMnK/OTUk Adaptation Layer
- OTUk Trail Termination Layer
- OTUk/ODUk Adaptation Layer

These functional layers are implemented in the OPSM23/4 cores in the MLD and by the logic in the OTN\_OTU MTSB. [Table 31](#), [Table 32](#), and [Figure 17](#) summarize the defects, defect correlations, consequential actions, performance monitoring events and their triggers supported by each layer of the LINEOTN subsystem in the receive and transmit directions. Note that defect correlations are supported through software.

**Table 31 Event Processing in the OPSM23/4 Core**

Name	Type	Activation / De-activation	Functional Layer	Direction Source (So): Transmit Sink (Sk): Receive	References
dLOS-P[i]	Defect	G.798: 6.2.1.1	OPSMnK_TT	Sk	G.798: 11.2.2.2 (section 11.2.2 has been mislabeled in G.798 as 11.2.1.2) & 6.2.1.1
dLOL	Defect	G.798: 6.2.5.5	OPSMnK_TT	Sk	G.798: 11.2.2.2 (section 11.2.2 has been mislabeled in G.798 as 11.2.1.2) & 6.2.5.5
dLOFLANE[i]	Defect	G.798: 6.2.5.6	OPSMnK_TT	Sk	G.798: 11.2.2.2 (section 11.2.2 has been mislabeled in G.798 as 11.2.1.2) & 6.2.5.6
aTSF	Consequential Action	$\Sigma\{dLOS-P[i]\}$ or $dLOL$ or $\Sigma\{dLOFLANE[i]\}$	OPSMnK_TT	Sk	G.798: 11.2.2.2 (section 11.2.2 has been mislabeled in G.798 as 11.2.1.2)
cLOS*	Defect Correlation	$\Sigma\{dLOS-P[i]\}$	OPSMnK_TT	Sk	G.798: 11.2.2.2 (section 11.2.2 has been mislabeled in G.798 as 11.2.1.2)
cLOL*	Defect Correlation	$(dLOL \text{ or } \Sigma\{dLOFLANE[i]\}) \text{ and } (\text{not } \Sigma\{dLOS-P[i]\})$	OPSMnK_TT	Sk	G.798: 11.2.2.2 (section 11.2.2 has been mislabeled in G.798 as 11.2.1.2)
dAIS	Defect	G.798: 6.2.6.3.1	OPSM/OTUk_A	Sk	G.798: 11.3.4.3 & 6.2.6.3.1
dLOF	Defect	G.798: 6.2.5.1	OPSM/OTUk_A	Sk	G.798: 11.3.4.3 & 6.2.5.1
cLOF*	Defect Correlation	$dLOF \text{ and } (\text{not } dAIS) \text{ and } (\text{not } AI\_TSF-P)$	OPSM/OTUk_A	Sk	G.798: 11.3.4.3

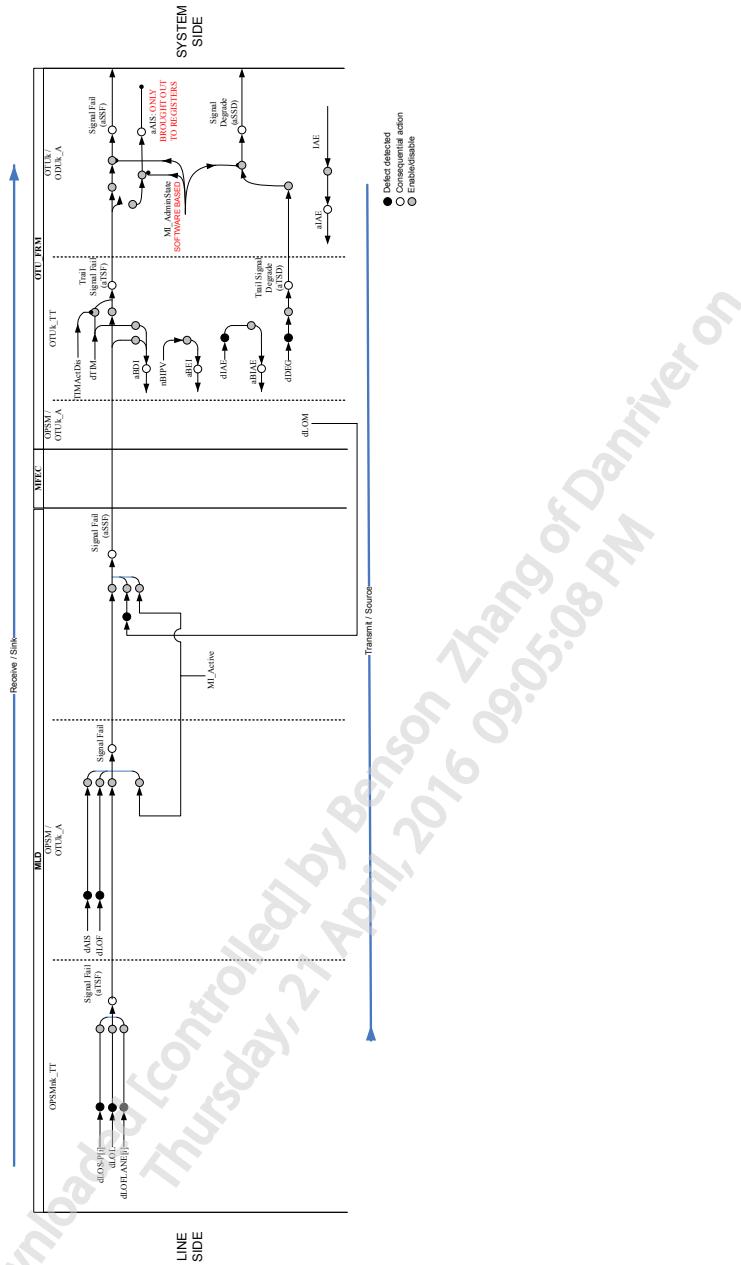
**Table 32 Event Processing in the Data Processor Core**

Name	Type	Activation / De-activation	Functional Layer	Direction Source (So): Transmit Sink (Sk): Receive	References
aIAE	Consequential Action	IAE (G.798: 13.3.1.1)	OTUk/ODUk_A	So	G.798: 13.3.1.1
dLOM	Defect	G.798: 6.2.5.2	OPSM/OTUk_A	Sk	G.798: 11.3.4.3 & 6.2.5.2
aSSF	Consequential Action	dAIS or dLOF or dLOM or AI_TSF-P or (not MI_Active)	OPSM/OTUk_A	Sk	G.798: 11.3.4.3
cLOM*	Defect Correlation	dLOM and (not dLOF) and (not dAIS) and (not AI_TSF-P)	OPSM/OTUk_A	Sk	G.798: 11.3.4.3
dTIM	Defect	G.798: 6.2.2.1	OTUk_TT	Sk	G.798: 13.2.1.2 & 6.2.2.1
dDEG	Defect	G.798: 6.2.3.4	OTUk_TT	Sk	G.798: 13.2.1.2 & 6.2.3.4
dBDI	Defect	G.798: 6.2.6.6.1	OTUk_TT	Sk	G.798: 13.2.1.2 & 6.2.6.6.1
dIAE	Defect	G.798: 6.2.6.10.1	OTUk_TT	Sk	G.798: 13.2.1.2 & 6.2.6.10.1
dBIAE	Defect	G.798: 6.2.6.11.1	OTUk_TT	Sk	G.798: 13.2.1.2 & 6.2.6.11.1
aBDI	Consequential Action	CI_SSF or dTIM	OTUk_TT	Sk	G.798: 13.2.1.2
aBEI	Consequential Action	nBIPV	OTUk_TT	Sk	G.798: 13.2.1.2
aBIAE	Consequential Action	dIAE	OTUk_TT	Sk	G.798: 13.2.1.2
aTSF	Consequential Action	CI_SSF or (dTIM and (not TIMActDis))	OTUk_TT	Sk	G.798: 13.2.1.2
aTSD	Consequential Action	dDEG	OTUk_TT	Sk	G.798: 13.2.1.2
cTIM*	Defect Correlation	dTIM and (not CI_SSF)	OTUk_TT	Sk	G.798: 13.2.1.2
cDEG*	Defect Correlation	dDEG and (not CI_SSF) and (not (dTIM and (not TIMActDis)))	OTUk_TT	Sk	G.798: 13.2.1.2
cBDI*	Defect Correlation	dBDI and (not CI_SSF) and (not (dTIM and (not TIMActDis)))	OTUk_TT	Sk	G.798: 13.2.1.2
cSSF*	Defect Correlation	CI_SSF	OTUk_TT	Sk	G.798: 13.2.1.2
pN_DS	Performance Monitoring	CI_SSF or dTIM	OTUk_TT	Sk	G.798: 13.2.1.2
pF_DS	Performance Monitoring	dBDI	OTUk_TT	Sk	G.798: 13.2.1.2
pN_EBC	Performance Monitoring	Sigma{nN_B}	OTUk_TT	Sk	G.798: 13.2.1.2
pF_EBC	Performance Monitoring	Sigma{nF_B}	OTUk_TT	Sk	G.798: 13.2.1.2
pBIAE	Performance Monitoring	dBIAE	OTUk_TT	Sk	G.798: 13.2.1.2
pIAE	Performance Monitoring	dIAE	OTUk_TT	Sk	G.798: 13.2.1.2
aSSF	Consequential Action	AI_TSF and (not MI_Admin-State = LOCKED)	OTUk/ODUk_A	Sk	G.798: 13.3.1.2

Name	Type	Activation / De-activation	Functional Layer	Direction Source (So): Transmit Sink (Sk): Receive	References
aAIS	Consequential Action	AI_TSF and (not MI_Admin-State = LOCKED)	OTUk/ODUk_A	Sk	G.798: 13.3.1.2
aSSD	Consequential Action	AI_TSD and (not MI_Admin-State = LOCKED)	OTUk/ODUk_A	Sk	G.798: 13.3.1.2

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:08 PM

Figure 17 LINEOTN Consequential Actions Processing



The consequential actions above are implemented in hardware. However, they are enabled and disabled through software.

## 5.6.4 OTN\_MLD Block

### 5.6.4.1 Datapaths

#### 5.6.4.1.1 General Datapath

On the line interface side, data is transmitted through 10G or 2.5G DS1 lanes. There are  $12 \times 10G$  DS1 lanes and  $34 \times 2.5G$  DS1 lanes. The 10G and 2.5G DS1 lanes are grouped according to the following MLD layer functions in order to carry the specified OTUk payloads:

- OTU1e/1f/2/2e/2f (10G rate category) occupying a single 10G DS1 lane processed through FEC and OTUk functions
- OTU3/3e1/3e2 (40G rate category) occupying  $4 \times 10G$  DS1 lanes processed through OTL3.4 , FEC and OTUk functions
- OTU3/3e1/3e2 (40G rate category) occupying  $(16+1) \times 2.5G$  DS1 lanes processed through SFI-5.1, FEC and OTUk functions
- OTU3/3e1/3e2 (40G rate category) occupying  $(4+1) \times 10G$  DS1 lanes through SFI-S( $n=4$ ), FEC and OTUk functions
- OTU4 (100G rate category) occupying  $10 \times 10G$  DS1 lanes processed through OTL 4.10, FEC and OTUk functions
- OTU4 (100G rate category) occupying  $(10+1) \times 10G$  DS1 lanes processed through SFI-S( $n=10$ ), FEC and OTUk functions
- The subsystem supports a single OTU4 datapath through either  $1 \times$  OTL4.10 or  $1 \times$  SFI-S( $n=10$ ) MLD interfaces
- A RSFEC capable of processing 1x100G. Alternatively, it can process up to 12x10G or 3x40G streams of data for a total of 120G of processing bandwidth
- Three I.4 FECs each capable of processing 4x10G or 1x40G traffic
- Three I.7 FECs each capable of processing 4x10G or 1x40G traffic
- 10G streams can select between I.4 and I.7 FECs in groups of 4 (per port selection is not supported)
- Three Swizzle FECs, each capable of processing 1x40G traffic. Also, the three Swizzle FECs when combined can process 1x100G traffic.

In the transmit direction, the line-side SERDES provides the clock to pull data out of the OPSM23/4. Moreover, the data from the COREOTN is pushed at an over-speed rate into the OPSM23/4. For this reason, back pressure signaling from the OPSM23/4 to the COREOTN subsystem adjusts the incoming data from the COREOTN subsystem to the SERDES transmit rate.

In the receive direction, the data from the line-side SERDES flows through the LINEOTN subsystem and into the COREOTN. Valid signals are used to realize the SERDES line rate in the over-speed system clock domain of the LINEOTN and COREOTN subsystems.

### 5.6.4.2 SFIS Block

The SFIS block implements the OIF-SFI-S-01.0 technical specification.

In the receive direction, the SFIS block provides SFI-S ( $N+1$ ) receive functionality for ingress streams. It provides translation from  $N$  lanes of data plus 1 lane of deskew on ingress from the SERDES to a single output stream. This is achieved using channel deskew framing, lane alignment, lane deskew and bit deinterleaving. The ingress interface is  $(N+1)$  lanes, 32-bits per lane, DS1 interface with each lane synchronous to its own independent SERDES recovered receive clock. The egress interface is a 1 lane,  $32 \times N$ -bits DS1 interface synchronous to a system receive clock.

In the transmit direction, the SFIS block provides SFI-S (N+1) transmit functionality for egress streams. It provides translation from a single stream interface at the ingress to N lanes of data plus 1 lane of deskew on egress to the SERDES. This is achieved using bit interleaving and deskew channel generation functions. The ingress interface is a 1 lane, 32\*N-bits DS1 interface synchronous to a transmit system clock. The egress interface is a (N+1) lanes, 32-bits per lane, DS1 interface with all lanes synchronous to the same SERDES transmit clock.

#### 5.6.4.3 SFI-5.1 Block

##### 5.6.4.3.1 SFI-5.1 Rx

The SFI-5.1 receive MLD performs the functions defined in OIF-SFI5-01.02 [47], providing the translation from the ingress SERDES multi-lane PMA interface to a single stream interface. The main function of this block is to receive 16 lanes of SERDES stream data and one lane of SERDES deskew data, use the deskew lane to align the other lanes, and merge the data lanes into a single stream. The SFI-5.1 receiver aggregates the 16 lanes of 8-bit data into a single 128-bit data lane that is synchronized with the deskew channel.

The SFI-5.1 receive MLD performs the following functions:

- Desksews and bit interleaves the 16 8-bit data lanes into a single data lane
- Performs bit-level deskew of the 16 lanes of data using the deskew channel. (The deskew algorithm is based on alternating between even and odd priority of lanes to allow optimal lane alignment for the 16 data lanes. The deskew algorithm continuously monitors the skew of the incoming lanes.)
- Asserts an error indication signal when lane alignment is lost

##### 5.6.4.3.2 SFI-5.1 TX

The SFI-5.1 transmit MLD performs transmit MLD functions as defined in OIF-SFI5-01.02 [47], providing the translation from the ingress single parallel interface to the egress SERDES multi-lane PMA interface. The main function of this block is to receive a single parallel data stream and to distribute this data stream to 16 lanes and a deskew lane. Data for each of the output lanes is transitioned from a common reference clock domain into its target SERDES transmit clock domain.

The SFI-5.1 transmit MLD performs the following functions:

- Performs bit de-interleaving of a single data lane into 16 data lanes
- Creates and transmits a deskew channel alongside 16 lanes of transmitted data stripes

#### 5.6.4.4 OPSM4 Block

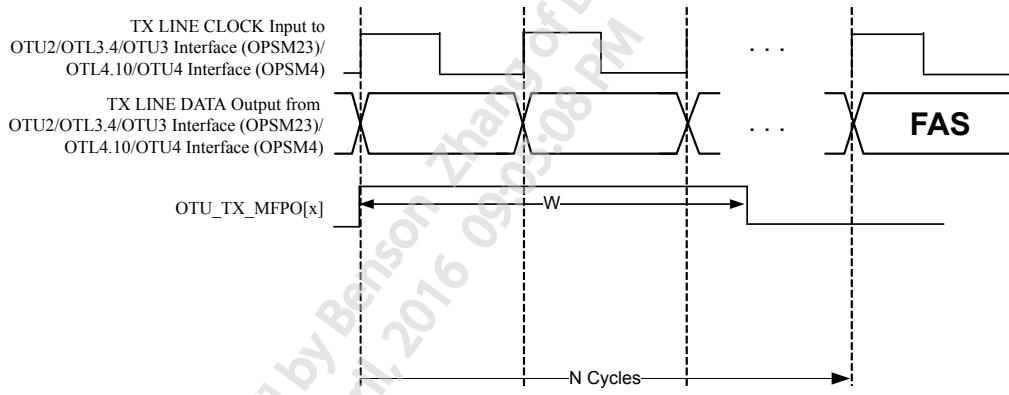
The OPSM4 block performs OPSMn<sub>k</sub> Trail Termination (k=4) and OPSM<sub>k</sub>/OTU<sub>k</sub> adaptation (k=4) functions (bit interleave/disinterleave, OTL4.10 frame alignment, OTL 4.10 lane alignment recovery, lane deskew, multi-lane identifier processing, lane rotation, frame alignment and optional scrambling/descrambling) as outlined in the G.709 and G.798 specifications. The block is configured to perform 100G OTU4 / OTL4.10 operations. The transmit processor and receive processor are independent entities within this block.

In the transmit direction, the OPSM4 performs the following functions:

- Receives 100G of data and FAS indication from the FEC blocks
- Transmits 100G data to the SFI-S n=10 transmit interface
- Transmits OTL4.10 formatted data to the line interface

- Performs the following OPSMn\_TT functions on the OTL 4.10 transmit datapath:
  - 16-byte block distribution and rotation (lane rotation) over 20 logical lanes, as per G.798 Section 11.2.2.1 and G.709 Annex C
  - Multi-lane identifier insertion, as per G.798 Section 11.2.2.1 and G.709 Annex C
  - Half-bit interleaving, as per G.798 Section 11.2.2.1 and G.709 Annex C
- Performs the following OPSM/OTUk\_A functions:
  - Optional scrambling post-FEC, as per G.798 Section 11.3.4.1 and G.709 Section 11.2
- Generates unframed (pn-11) OTUk-AIS triggered by software registers when the transmit SFI port is used
- Provides configurable logical lane to physical lane mapping
- Generates a frame pulse OTU\_TX\_MFPO in the transmit path. The time (in clock cycles) from TX\_FP assertion to FAS exiting the OTL4.10 Line Interface or the OTU4 Line Interface is deterministic. **Figure 18** shows the timing of OTU\_TX\_MFPO. In the figure, "W" is nominally 7 clock cycles when FAS is present, and 17 clock cycles when MFAS is present on OTU\_TX\_MFPO[x].

Figure 18 OTU\_TX\_MFPO Timing Diagram

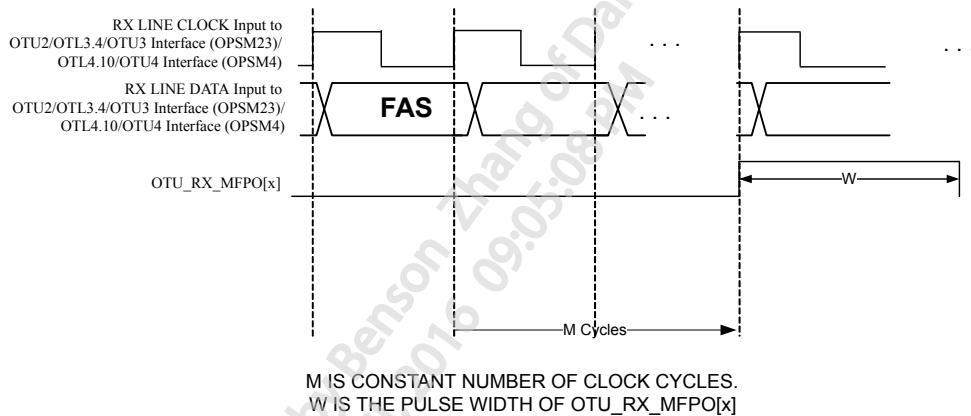


In the receive direction, the OPSM4 performs the following functions:

- Receives OTL4.10 formatted data from the line interface
- Receives 100G data from an SFI-S n=10 interface
- Transmits 100G data and FAS indication to the FEC blocks
- Performs the following OPSMn\_TT functions for OTL 4.10 interface arriving stream:
  - Half-bit de-interleaving, as per G.798 Section 11.2 and G.709 Annex C
  - OTL 4.10 frame alignment, as per G.798 Sections 11.2 and 8.2.5
  - OTL4.10 lane alignment recovery, as per G.798 Sections 11.2 and 8.2.6
  - Lane deskew (multi-lane identifier and delay processing / elastic store), as per G.798 Section 11.2 and G.709 Annex C
  - Multi-lane identifier removal and OTU4 FAS re-creation, as per G.798 Section 11.2
  - 16-byte block multiplexing per G.798 Section 11.2 and G.709 Annex C
- Performs the following OPSM/OTUk\_A functions:

- Frame alignment to OTUk frames with the ability to detect AIS and LOF, as per G.798 Sections 11.3.4.3 and 8.2.1 (with optional interrupt generation), and with programmable OOF and LOF algorithm state transitions
- Optional de-scrambling, as per G.798 Section 11.3.4.3 and G.709 Section 11.2
- Detects LOS, LOL and LOFLANE, AIS and LOF conditions with optional interrupt generation
- During OTL4.10 operation, transmits Logical lane ID towards the FEC interface
- The maximum OTL4.10 lane skew tolerance is 500ns. The lane skew tolerance specifies the maximum time difference between the earliest arriving and latest arriving lanes that can be accommodated by the OTL4.10 lane deskew algorithm.
- Generates a frame pulse OTU\_RX\_MFPO in the receive path. The time (in clock cycles) from FAS entering the OTL4.10 Line Interface or the OTU4 Line Interface to OTU\_RX\_MFPO assertion is deterministic. [Figure 19](#) shows the timing of OTU\_RX\_MFPO. In the figure, "W" is nominally 7 clock cycles when FAS is present, and 17 clock cycles when MFAS is present on OTU\_RX\_MFPO[x].

**Figure 19 OTU\_RX\_MFPO Timing Diagram**



In addition, the OPSM4 supports the following general features:

- Complies with relevant sub-clauses of the G.709 and G.798 specifications
- Supports terminal loop back capabilities
- Provides line clocks at OTU4 rates with fixed data widths
- All defects may be individually suppressed via software accessible registers
- All consequential actions may be forced via software accessible registers and have software configurable enables and disables
- All consequential actions have software configurable enables
- Provides a FEC\_FAIL signal, to be used by downstream 100G FEC decoders to suppress error correction during alarm conditions. The defects selected to drive the FEC\_FAIL output are register programmable.
- Supports the defects, defect correlations and consequential actions described in [Section 5.6.3.5](#)

#### 5.6.4.5 OPSM23 Block

The OPSM23 block performs OPSM<sub>k</sub> Trail Termination ( $k=3$ ) and OPSM<sub>k</sub>/OTU<sub>k</sub> Adaptation ( $k=2,3$ ) functions (OTL3.4 frame alignment, OTL 3.4 lane alignment recovery, lane deskew, lane rotation, frame alignment and optional scrambling/descrambling), as outlined in the G.709 and G.798 specifications.

The block is configured to perform 4x10G or 40G operations at the respective OTU1e, OTU1f, OTU2, OTU2e, OTU2f, OTU3, OTU3e1, OTU3e2 data rates. The transmit processor and receive processor are independent entities within this block.

In the transmit direction, the OPSM23 performs the following functions:

- Receives a 4x10G/40G data and frame alignment signal (FAS) from the upstream FEC blocks
- Processes up to four 10G streams or one 40G stream
- Transmits 4 x 10G data streams to line side XFI
- Transmits a 40G data stream to a SFI-S (n=4) interface
- Transmits a 40G data stream to the Line side interface as OTL3.4 formatted data
- Performs the following OPSMn<sub>k</sub>\_TT functions for the OTL 3.4 interface destined data stream:
  - 16-byte block distribution and rotation (lane rotation) over four logical lanes per G.798 Section 11.2 and G.709 Annex C
- Performs the following OPSM/OTUk\_A functions:
  - Optional scrambling post-FEC per G.798 Section 11.3 and G.709 Section 11.2
- Generates unframed (pn-11) OTUk-AIS triggered by software registers
- Provides configurable logical lane to physical lane mapping
- Generates a frame pulse (OTU\_TX\_MFPO) in the transmit path. The time (in clock cycles) from TX\_FP assertion to FAS exiting the OTU2/OTL3.4 Line Interface or the OTU3 Line Interface is deterministic. [Figure 18](#) shows the timing of OTU\_TX\_MFPO. In the figure, "W" is nominally 7 clock cycles when FAS is present, and 17 clock cycles when MFAS is present on OTU\_TX\_MFPO[x]. .

In the receive direction, the OPSM23 performs the following functions:

- Receives 4 x 10G data stream from the line-side XFI
- Receives an OTL3.4 formatted 40G data stream from the line side interface
- Receives 40G data stream from an SFI-S (n=4) interface
- Transmits 4x10G/40G data and FS to the FEC blocks using system clock upon valid signal assertion
- Performs the following OPSMn<sub>k</sub>\_TT functions on the incoming OTL 3.4 stream:
  - OTL 3.4 frame alignment, as per G.798 Sections 11.2 and 8.2.5
  - OTL3.4 lane alignment recovery, as per G.798 Sections 11.2 and 8.2.6
  - Lane deskew (multi-lane identifier and delay processing / elastic store), as per G.798 Section 11.2 and G.709 Annex C.
  - 16-byte block multiplexing, as per G.798 Section 11.2 and G.709 Annex C
- Performs the following OPSM/OTUk\_A functions:
  - Frame alignment to the OTUk frames with ability to detect AIS and LOF, as per G.798 Sections 11.3.4.3 and 8.2.1 (with optional interrupt generation) and with programmable OOF and LOF algorithm state transitions
  - Optional de-scrambling, as per G.798 Section 11.3.4.3 and G.709 Section 11.2
  - Provides per-link FEC\_FAIL signals, to be used by downstream FEC decoders to suppress error correction during alarm conditions. The defects selected to drive the FEC\_FAIL outputs are register programmable.
  - Detects LOS, LOL and LOFLANE, AIS and LOF conditions with optional interrupt generation
  - During OTL3.4 operation, transmits a Logical lane ID towards the FEC interface
  - The lane deskew elastic store is able to correct up to 500 ns of differential skew in the received lanes
  - Generates a frame pulse (OTU\_RX\_MFPO) in the receive path. The time (in clock cycles) from FAS entering the OTU2/OTL3.4 Line Interface or the OTU3 Line Interface to OTU\_RX\_MFPO assertion

is deterministic. [Figure 19](#) shows the timing of OTU\_RX\_MFPO. In the figure, "W" is nominally 7 clock cycles when FAS is present, and 17 clock cycles when MFAS is present on OTU\_RX\_MFPO[x].

In addition, the OPSM23 supports the following general features:

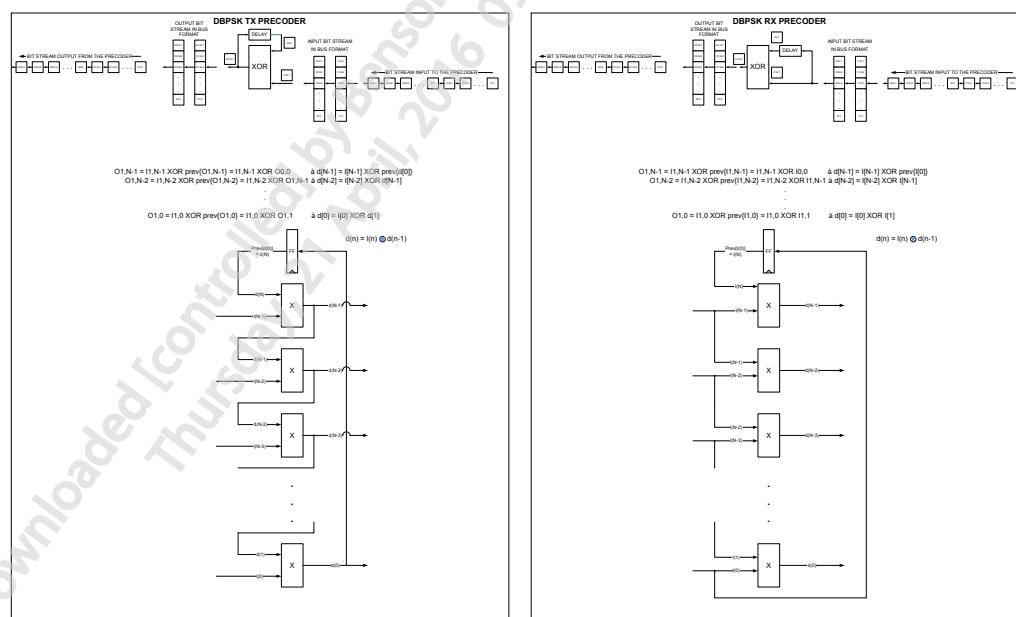
- Complies with relevant sub-clauses of the G.709 and G.798 specifications
- Supports both 40G and 4x10G modes of operation
- Supports terminal loop back capabilities
- Supports OTU1f, OTU1e, OTU2, OTU2e, OTU2f, OTU3, OTU3e1, OTU3e2 data rates
- All defects can be suppressed using software accessible registers
- All consequential actions can be forced via software accessible registers
- All consequential actions can be enabled or disabled via software configurable registers
- Supports the defects, defect correlations, and consequential actions described in [Section 5.6.3.5](#)

#### 5.6.4.6 Precoders

The MLD subsystem contains a pair of 32-bit DPSK precoders (receive and transmit) on each of its twelve 10G lanes. The 32-bit DPSK precoders are used for precoding OTU1e/1f/2/2e/2f streams.

The MLD subsystem also contains a pair of 128-bit DPSK precoders (receive and transmit) on the OTU3 interface of each of the three OPSM23 blocks. The 128-bit DPSK precoders are used for precoding OTU3/OTU3e1/OTU3e2 streams before they are multi-lane distributed through the SFI-S(n=4)/SFI-5.1 modules.

**Figure 20 DPSK Precoder Description**



#### 5.6.4.7 BERGEN Blocks

Bit Error Rate Generator (BERGEN) blocks are available on each 10G, 32-bit line-side interface. These blocks are typically used to evaluate FEC performance.

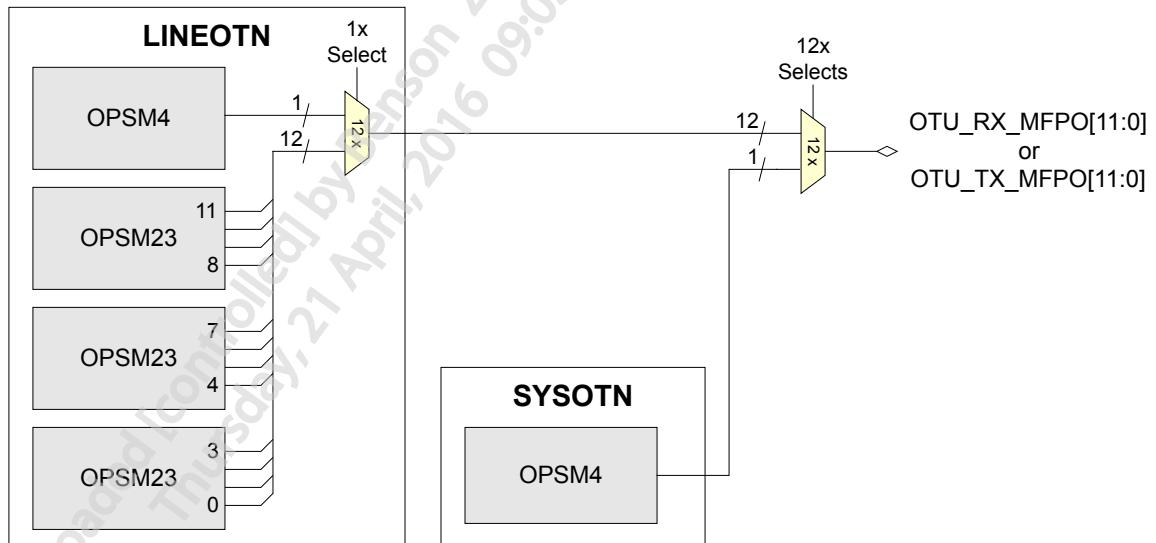
The BERGEN blocks have the following features:

- Optionally, insert errors into the transmit stream (towards the SERDES)
- Provide three error injection modes:
  - Normal mode: When activated, the bit-stream is inverted. If a bit is a 1 and an error is to be injected, it will be inverted to a 0 (and vice versa).
  - Zero Inject mode: When activated, an error will only be injected if the signal is already a 1 (by setting it to 0)
  - One Inject mode: When activated, an error will only be injected if the signal is already a 0 (by setting it to 1)
- An error rate from 0% to 100 % can be specified using software-programmable registers

#### 5.6.4.8 PTP Over OTN

Implementation of PTP over OTN can be achieved by inserting and extracting PTP packets into user-defined locations of the OTN overhead via the device overhead insertion and extraction ports. The OTN Frame Pulse interface (see [Section 4.2.10](#)) provides frame pulse signals that are synchronous to the frame alignment of OTN frames received by, or transmitted from, the device. An external PTP processor can use these frame pulses to determine the time PTP packets are received or sent via OTN overhead. Refer to [Section 5.18.6](#) for the latency from an OTN frame being sent or received on the device pins to the assertion of the associated frame pulse. When the OTN framer for a given link is in an OOF state, the behavior of OTU\_RX\_MFPO and OTU\_TX\_MFPO is undefined.

Figure 21 OTU RX and TX MFPO Muxing



*Note: all above logic is duplicated for both OTU\_RX\_MFPO[11:0] and OTU\_TX\_MFPO[11:0]*

As seen in [Figure 21](#), OTU\_RX\_MFPO[x] can be configured per bit to be driven by either LINEOTN or SYSOTN. SYSOTN broadcasts the OTU4 frame pulse from the OPSM4 block to all bits. Similarly, LINEOTN in OTU4 mode broadcasts the OTU4 frame pulse from the OPSM4 block to all bits. When a group of four LINEOTN lanes (0-3, 4-7, or 8-11) is in OTU3 mode using the OPSM23 block, the OTU\_RX\_MFPO[x] bit that corresponds to the highest numbered lane (3, 7, or 11) is valid. The other three bits are invalid. When a group of four LINEOTN lanes is in OTU2 mode using the OPSM23 block, all corresponding OTU\_RX\_MFPO[x] bits are valid.

**Note:** When the source is LINEOTN, OTU\_RX\_MFPO and OTU\_TX\_MFPO are indexed according to the output of the LIFD subsystem. For example, the frame pulse of an OTU2 that arrives on LINE\_RXI\_P/N[x] and is switched to internal lane y by the LIFD will be output on OTU\_RX\_MFPO[y]. Similarly, the frame pulse of an OTU2 driven on LINE\_TXO\_P/N[x] that is switched from internal lane y by the LIFD will be output on OTU\_TX\_MFPO[y].

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:08 PM

## 5.6.5 FEC Block

### 5.6.5.1 10G / 40G / 100G RSFEC

The 10G/40G/100G RSFEC block:

- Has 7% overhead, and complies with RSFEC/G.709
- Provides performance comparable to the reference values in RSFEC/G.709
- Interfaces to 10G channels, 40G channels, or one 100G channel.
- Provides accurate counts of scrambled corrected ones, scrambled corrected zeros, uncorrectable symbols, and a matching count of OTN frames over which that count is collected
- Provides OTUk (stream-based) FEC statistics for 10G, 40G, and 100G streams
- Maintains datapath latency during a Signal Fail condition

### 5.6.5.2 4x10G / 40G I.4 FEC

The 4x10G/40G I.4 FEC:

- Has 7% overhead, and complies with I.4/G.975.1
- Provides performance comparable to the reference values in I.4/G.975.1
- Interfaces to four distinct 10G channels or one 40G channel
- Provides accurate counts of scrambled corrected ones, scrambled corrected zeros, uncorrectable symbols for each stage, and a matching count of OTN frames over which that count is collected
- Provides stream-based (OTUk) FEC statistics for 10G and 40G data streams
- Provides logical-lane FEC statistics for OTL3.4 formatted 40G data streams
- Maintains datapath latency during a Signal Fail condition

### 5.6.5.3 4x10G / 40G I.7 FEC

The 4x10G/40G I.7 FEC:

- Has 7% overhead and complies with I.7/G.975.1
- Provides performance comparable to the reference values in I.7/G.975.1
- Interfaces to four distinct 10G channels or one 40G channel
- Provides accurate counts of scrambled corrected ones, scrambled corrected zeros, uncorrectable symbols for each stage, and a matching count of OTN frames over which that count is collected
- During 40G operation, provides two modes of statistic reporting to the system:
  - Mode 1 - report for the entire 40G channel
  - Mode 2 - report according to the Logical lane Identifier provided from the MLD layers
- Maintains datapath latency during Signal Fail condition

### 5.6.5.4 40G / 100G Swizzle FEC

The 40G / 100G Swizzle FEC is PMC's FEC. There are three Swizzle FEC instances inside the chip, each of which is capable of processing one 40G data stream. The three 40G Swizzle instances may be used concurrently to realize a 100G (OTU4) Swizzle FEC.

The Swizzle FEC can correct up to a random bit error rate of 4.8E-3, corresponding to 9.45 dB of net effective coding gain. It can correct up to 2048 consecutive errors at 40G and up to 6000 consecutive errors at 100G.

### 5.6.5.5 FEC Latency

Table 99 describes the LINEOTN FEC latency.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:08 PM

## 5.6.6 Shared RAM

The LINEOTN subsystem contains three slices of SHARED\_FEC\_RAM.

The SHARED\_FEC\_RAM subsystem is a wrapper that contains one instance of 40G SHARED RAM.

The LINEOTN subsystem has several FEC blocks, each with its own unique RAM requirements. Since the RAM requirement of the FECs in the LINEOTN is high, and since only one of the I.4 FEC, I.7 FEC and Swizzle FEC is active in a given 40G region, a method is needed to share the RAMs needed by these FECs in order to reduce the device die size. The 40G shared RAM provides each of these FECs (I.4 FEC, I.7 FEC and Swizzle FEC) with their unique interfaces using a common set of RAMs.

## 5.6.7 FEC Selection

The 3x40G slice architecture of the shared RAM limits the selection of 10G I.4 and 10G I.7 FEC to groupings of four 10G ports. The device supports I.4, I.7, and G.709 RSFEC for 10G OTN. The I.7 and I.4 FEC are configurable in multiples of 40G as a result of the shared RAM implementation. Groups of 4x10G must use either I.4 or I.7 FEC. Any 10G G.709 RSFEC port can be mixed with any I.4 or I.7 10G FEC port since RSFEC does not use the shared memory. Therefore, the RSFEC can be configured on any 10G port. Examples of FEC options for a 6xOTU2 to 6xOTU2 transponder are:

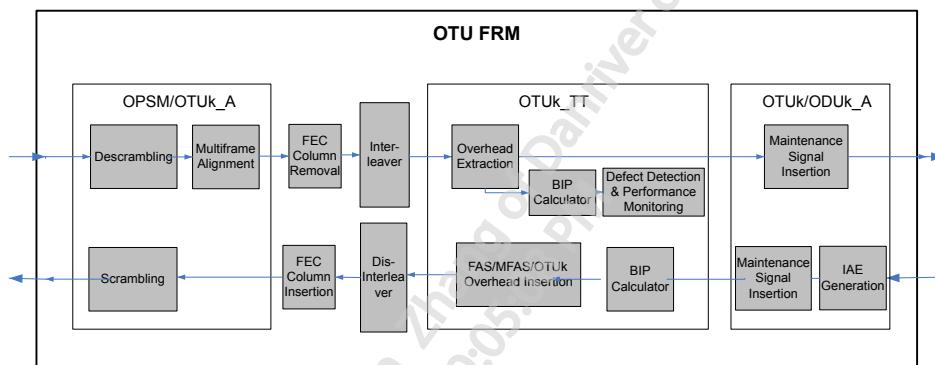
- 6xOTU2 RSFEC <> 6xOTU2 I.4
- 6xOTU2 RSFEC <> 6xOTU2 I.7
- 6xOTU2 RSFEC <> 6xOTU2 RSFEC
- 6xOTU2 I.4 <> 6xOTU2 I.4
- 6xOTU2 I.7 <> 6xOTU2 I.7
- 6xOTU2 I.X <> 2xOTU2 I.X + 4xOTU2 I.Y
- Note that 6xOTU2 I.4 <> 6xOTU2 I.7 is not supported since 2x40G slices of I.4 and 2x40G slices of I.7 are required, but the shared RAM limits availability to 2x40G I.4 + 1x40G I.7 or 2x40G I.7 + 1x40G I.4.

## 5.6.8 OTU\_FRM Block

The 12x10G / 3x40G / 100G OTU\_FRM block performs OTUk Trail Termination (k=2,3,4) and OTUk/ODUK Adaptation (k=2,3,4) functions, as outlined in the G.709 and G.798 specifications. The block processes 12x10G, 3x40G or 1x100G streams of OTU1e, OTU1f, OTU2, OTU2e, OTU2f, OTU3, OTU3e1, OTU3e2, or OTU4. The block terminates all OTUk functions, including FEC columns, to form the ODUk frame during receive processing. The OTU\_FRM block inserts user-programmed OTUk OH bytes in the OTUk stream, or forwards OTUk overhead bytes inserted by upstream logic. The OTU\_FRM block also inserts empty OTUk FEC columns to form an OTUk frame.

Figure 22 illustrates the OTU\_FRM datapath processing.

Figure 22 OTU\_FRM Datapath Diagram



The transmit processor and receive processor within the OTU\_FRM are independent entities that include the features described in the following subsections.

### 5.6.8.1 Transmit Processor Features

The transmit processor includes the following general features:

- Provides twelve 10G egress ports. The 10G egress ports can be merged to form three 40G streams or a single 100G stream.
- Dis-interleaves incoming data from a channelized DCI bus on up to twelve non-channelized buses. The channel to bus mapping is user-configurable.
- Provides ODUK-LCK maintenance signal insertion capabilities via software override

Additionally, the transmit processor:

- Processes 1x100G, 3x40G, 12x10G, or a combination of 40G and 10G framed, 16-byte aligned, channelized, ODUK data up to a combined maximum of 120G from upstream logic
- Supports the following OTU3/ODU3 and OTU2/ODU2 receive / transmit hybrid configurations:
  - Transmits 1x40G and up to 8x10G of framed OTUk / ODUk data
  - Transmits 2x40G and up to 4x10G of framed OTUk / ODUk data
- Does not support a mix of 10G OTU2/ODU2 and 100G OTU4/ODU4 data. This is not a valid combination of data to achieve 120G: 100G and 10G parallel processing is not supported due to hardware datapath restrictions.
- Performs the following OTUk/ODUk\_A\_So functions:
  - ODUK-LCK generation, as per G.798 Section 13.3.1.1 and G.709 section 16.5.
  - Normal signal / ODUk-LCK selection, as per G.798 Section 13.3.1.1

- IAE generation upon ODUk frame start position incoming alignment error, as per G.798 Section 13.3.1.1
- ODUk to OTUk frame mapping, as per G.798 section 13.3.1.1 and G.709 Section 11.1
- Performs the following OTUk\_TT\_So functions:
  - SM-TTI insertion, as per G.798 section 13.2.1.1 and G.709 Sections 15.2 and 15.7
  - SM-BIP8 computation and insertion, as per G.798 sections 13.2.1.1 and 8.3.4.1 and G.709 Section 15.7
  - SM-BEI/BIAE insertion, as per G.798 section 13.2.1.1 and G.709 Section 15.7
  - SM-BDI insertion, as per G.798 Section 13.2.1.1 and G.709 Section 15.7
  - SM-IAE insertion, as per G.798 Section 13.2.1.1
  - SM-RES insertion, as per G.798 Section 13.2.1.1
- Performs the following OPSM/OTUk-a\_A\_So functions:
  - Optional FAS / MFAS insertion, as per G.798 Section 11.3.4.1 and G.709 Section 15.6
  - Pre-FEC scrambling, as per G.798 Section 11.3.4.1 and G.709 Section 11.2. This function can be disabled
- Performs register based, constant value, GCC0 and RES insertion
- Optionally, inserts OTUk overhead bytes through statically programmed registers or passes through overhead bytes that are inserted upstream prior to the transmit ingress to the block
- Inserts empty FEC columns in the OTUk frame structure

### 5.6.8.2 Receive Processor Features

The receive processor includes the following general features:

- Provides twelve 10G ingress ports. The 10G ingress ports can be merged to form three 40G streams or a single 100G stream.
- Schedules and interleaves incoming data onto a channelized DCI bus using a ninety-six entry calendar. The bus to channel mapping is user-configurable.
- Allows user-configurable channel number assignment to incoming streams
- Provides for ODUk-AIS insertion at the nominal rate during a signal fail condition
- Provides for forced insertion of ODUk-AIS, and ODUk-LCK maintenance signals at the nominal rate via software override

Additionally, the receive processor:

- Processes 1x100G, 3x40G, 12x10G, or a combination of 40G and 10G framed OTUk data up to a combined maximum of 120G from upstream logic
- Supports the following OTU3/ODU3 and OTU2/ODU2 receive / transmit hybrid configurations:
  - Transmits / receives 1x40G and up to 8x10G of framed OTUk / ODUk data
  - Transmits / receives 2x40G and up to 4x10G of framed OTUk / ODUk data
- Does not support a mix of 10G OTU2/ODU2 and 100G OTU4/ODU4 data. This is not valid combination of data to achieve 120G: 100G and 10G parallel processing is not supported due to hardware datapath restrictions.
- Performs the following OPSM/OTUk-a\_A\_Sk functions:
  - Post-FEC de-scrambling, as per G.798 Section 11.3.4.3 and G.709 Section 11.2. This function can be disabled.
  - Multi-frame alignment and LOM detection, as per G.798 Sections 11.3.4.3 and 8.2.2
- Performs the following OTUk\_TT\_Sk functions:

- SM-TTI extraction, as per G.798 Sections 13.2.1.2 and 8.6 and G.709 Sections 15.2 and 15.7
- SM-BIP8 computation, extraction and processing, as per G.798 Sections 13.2.1.2 and 8.3.4.2 and G.709 Section 15.7
- SM-BEI/BIAE extraction and nF\_B detection, as per G.798 Section 13.2.1.2 and G.709 Section 15.7
- SM-BDI extraction and BDI defect detection, as per G.798 Section 13.2.1.2 and G.709 Section 15.7
- SM-IAE extraction and IAE detect detection, as per G.798 Section 13.2.1.2 and G.709 Section 15.7
- SM-RES extraction, as per G.798 Section 13.2.1.2 and G.709 Section 15.7
- Performs the following OTUk/ODUk\_A\_Sk functions:
  - ODUk MFAS generation, as per G.798 Section 13.3.1.2. There is no need to generate an ODUk clock. The system clock and valid signals should be used instead. ODUk FAS is passed into the block along with data on the ingress data bus
  - OTUk to ODUk frame extraction, as per G.798 Section 13.3.1.2 and G.709 Section 11.1
  - OTUk to ODUk consequential action processing, as per G.798 Section 13.3.1.2
  - ODUk-LCK / ODUk-AIS generation and insertion at the ODUk nominal rate, as per G.798 Section 13.3.1.2.
- Note: Maintenance signal insertion at nominal rate in the OTU\_FRM results in all overhead being overwritten with zeros. Headers for the maintenance signals can be inserted in COREOTN.
- Normal signal / ODUk-LCK / ODUk-AIS selection, as per G.798 Section 13.3.1.2
- Performs register based, constant value OTUk GCC0 and RES extraction
- Terminates the OTUk layer excluding FEC columns, but passes through OTUk OH bytes to the receive egress port so that they can be extracted after the block
- Provides OTUk OH processing for SM layer termination, including a communication channel to return BDI/BEI/BIAE values to the transmit processor
- Removes empty FEC columns in the OTUk frame structure
- Generates a signal fail condition based on OTLk/OTUk/ODUk Trail Termination / adaptation layer processing and propagates signal fail downstream to the receive egress port
- Receives a signal fail condition on the block's receive ingress port and performs the corresponding consequential actions

### 5.6.8.3 OTU\_FRM Data Scheduling and Interleaving

The OTU\_FRM contains an interleaver that can merge up to twelve incoming single-channel DCI streams onto a common multi-channel DCI bus. As it is transferred onto the multi-channel bus, each ingress DCI is assigned a configurable channel ID ranging from 0 to 11. See [Section 5.6.8.5](#) for channel mapping details.

Data is sent out from the interleaver when:

- A channel is scheduled
- A full 48-byte word is available for transmission in the interleaver

Data is scheduled onto the DCI bus using a twelve-channel non-work conserving SCBS3 scheduler. Each DCI channel ID is scheduled with a 10G OTU2 client receiving one slot, a 40G OTU3 client receiving four slots and a 100G OTU4 client receiving 10 slots.

The interleaver contains per-channel FIFOs that serve two functions:

- Provides clock domain crossing capabilities from the LINEOTN clock domain to the system clock domain
- Holds incoming data while a channel is waiting to be scheduled

#### 5.6.8.4 OTU\_FRM Dis-interleaving

The OTU\_FRM contains a dis-interleaver to handle the de-muxing of a multi-channel DCI bus onto twelve single channel DCI buses. Each egress DCI is assigned a user-configurable channel ID ranging from 0 to 11. Ingress data on the multi-channel DCI bus is mapped to the appropriate single channel egress DCI using the channel number. See [Section 5.6.8.5](#) for mapping details.

The dis-interleaver contains a FIFO that serves three functions:

- Provides clock domain crossing capabilities from the system clock domain to the LINEOTN clock domain
- Holds outgoing data while FEC columns are being inserted
- Provides the capability to hold data during back-pressuring from the line and contains sufficient depth to store upstream data while upstream logic is being back-pressed

#### 5.6.8.5 Physical Lane to Logical Lane Channel Mapping

The following table specifies lane mapping in the OTU\_FRM.

Table 33 Physical Lane to Logical Lane Channel Mapping

Data Rate	Physical 10G Lane(s) (OPSM Line Side)	Physical 10G Lane(s) (OPSM Core Side)	OTN_OTU Line Side Master Lane	Logical Lane DCI Channel ID
10G	0	0	0	0
	1	1	1	3
	2	2	2	6
	3	3	3	9
	4	4	4	1
	5	5	5	4
	6	6	6	7
	7	7	7	10
	8	8	8	2
	9	9	9	5
	10	10	10	8
40G <sup>1</sup>	0-3 (OTL3.4) 0-4 (SFI-S)	0-3	3	0
	4-7 (OTL3.4) 5-9 (SFI-S)	4-7	7	1
	8-11 (OTL3.4) N/A (SFI-S)	8-11	11	2
	0-9 (OTL4.10) 0-10 (SFI-S)	0-11	11	0

**Note:**

1. From the perspective of the SW driver, for a given 40G or 100G link, the lowest OPSM Line Side Physical Lane number is the ID to be used for configuring the DSI stream.

### 5.6.8.6 OTU\_FRM Maintenance Signal Insertion

In the RX sink direction, the OTU\_FRM can insert both ODUk-AIS and ODUk-LCK at the nominal rate. OTU\_FRM replaces the ODUK2 / 3 / 4 datastream with newly generated frames which are void of any header information and only contain the AIS or LCK pattern which is mapped into payload of the frame. Sideband framing signals such as start of frame (SOF) are also regenerated. The header fields can be inserted in the COREOTN.

The priority of RX maintenance signal insertion is as follows (in order of priority from highest to lowest):

1. Forced ODUk-AIS / ODUk-LCK insertion via register.
2. Enabling of MI\_AdminState. This state is enabled via register, and forces ODUk-LCK.
3. Signal Fail induced ODUk-AIS insertion. Signal fail may be triggered by a contributing defect, or may be forced via register.

In the TX source direction, the OTU\_FRM can insert ODUk-LCK. The OTU\_FRM maps the ODUk-LCK pattern into the payload of an existing ODUk frame. SM Overhead insertion functions normally and the TX rate is controlled by the line side SERDES.

The priority of TX maintenance signal insertion is as follows (in order of priority from highest to lowest):

1. Forced ODUk-LCK insertion via register.
2. Enabling of MI\_AdminState. This state is enabled via register, and forces ODUk-LCK.

### 5.6.8.7 OTU\_FRM Overhead Insertion Capabilities

The following table contains details on the SM Overhead insertion capabilities of the OTU\_FRM. All fields can either be placed in passthrough mode, where the data is passed through from upstream, or each field can be overwritten by the OTU\_FRM. Each of the following configurations are per channel, and each channel has a dedicated constant value insertion register for each OTUk overhead field, with the exception of FAS, which is a constant value for all channels.

Table 34 SM Overhead Insertion

Overhead Field	Mode	OH Insertion Method
FAS	0	Passthrough
	1	Register based FAS insertion is enabled. The FAS value
MFAS	0	Passthrough
	1	MFAS is inserted based on a free-wheeling counter slaved to the upstream de-bounced MFAS F_ALIGN sideband signal
	2	Register based constant value MFAS insertion
SM-TTI (SAPI, DAPI, OP_SPEC)	0	Passthrough
	1	Register based SM-TTI insertion
SM-BIP	0	Passthrough
	1	Calculated SM-BIP insertion enabled
	2	Passthrough with corruption to SM-BIP via configurable bit-mask
	3	Calculated BIP insertion with corruption to SM-BIP via a configurable bit-mask
	4	Register based constant value SM-BIP insertion
SM-BEI/BIAE	0	Passthrough
	1	Backwards consequential action from OTUk_Sk (RX path)
	2	Register based constant value SM-BEI/BIAE insertion

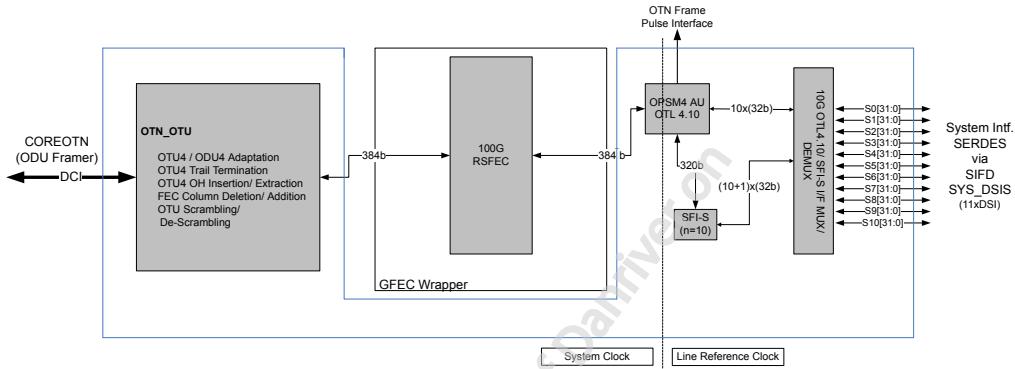
Overhead Field	Mode	OH Insertion Method
SM-BDI	0	Passthrough
	1	Backwards consequential action from OTUk_Sk (RX path)
	2	Register based constant value SM-BDI insertion
SM-IAE	0	Passthrough
	1	Backwards consequential action from OTUk_Sk (RX path)
	2	Register based constant value SM-IAE insertion
SM-RES	0	Passthrough
	1	Register based constant value SM-RES insertion
GCC0	0	Passthrough
	1	Register based constant value GCC0 insertion
RES	0	Passthrough
	1	Register based constant value RES insertion

Downloaded [controlled] by Benson Zhang of Dan Rivertron  
Thursday, 21 April, 2016 09:05:08 PM

## 5.7 System OTN (SYSOTN) Subsystem

### 5.7.1 Block Diagram

Figure 23 SYSOTN Block Diagram



### 5.7.2 Overview

The SYSOTN subsystem enables a single-chip OTN to OTU4 muxponder application with SYSOTN as the OTU4 line-side interface and low order clients mapped into ODU4 at the core side interface.

The SYSOTN subsystem also enables OTU4 to OTU4 transponder applications with the LINEOTN as the line-side interface with EFEC (Enhanced Forward Error Correction) and the SYSOTN as the client-side interface with RSFEC (Reed Solomon Forward Error Correction).

RSFEC is the supported FEC in the SYSOTN. The RSFEC can be hardware bypassed in the SYSOTN, in both the encoder (transmit) and decoder (receive) directions.

The SYSOTN subsystem interfaces to the device SIFD mux on the SERDES side through a DS1 (a combination of 10G links), and provides various OTU4 layer functions:

- OPSMnk adaptation layers
- FEC encoding / decoding
- OTU (Optical Channel Transport Unit) termination
- Interfacing to the COREOTN subsystem through a multi-channel DCI

The subsystem only supports Electrical OTN traffic from the SERDES line which is compliant to G.709 OTUk frame size of 4080 columns x 4 rows (including FEC columns). The SYSOTN subsystem only supports one line rate, OTU4. The intent of the SYSOTN subsystem is to terminate the OTUk layer.

There are 11 SERDES links connected to the MLD logic. Each lane is 32-bits wide and runs at the SERDES line clock rate.

The MLD layer functions perform multi-lane alignment (lane ordering and deskeew) on 100G multi-lane traffic. They also perform OTUk frame alignment as well as scrambling/descrambling. In addition, the MLD layer performs rate adaptation between the SERDES line rate to the system (core side) clock rate.

The MLD layer transports 100G rates using multiple SERDES lanes. SYSOTN supports two MLD modes of operation: SFI-S(n=10) and OTL4.10.

The SYSOTN subsystem supports 100G RSFEC. The subsystem supports OTUk-based FEC statistics for OTU4 traffic.

The subsystem interfaces to the COREOTN subsystem through a multi-channel DCI, which is formed through the interleave and dis-interleave block. The channel ID is fixed to a constant value (0x00), since there is only one channel. The interleave/dis-interleave block takes 100G channels from the OTUk layer and places them on the multi-channel DCI bus interfacing to the COREOTN subsystem.

The following subsections highlight some key features of the sub-blocks within the SYSOTN subsystem.

## 5.7.3 Features

### 5.7.3.1 Multi-lane Distribution (MLD) Features

The SYSOTN supports SFI-S and OTL4.10 Multi-lane De-skew (MLD) transmit and receive functions.

- SFI-S: In this mode, 11 SERDES links are used. Ten lanes are used to transfer data, and the other lane is used to convey the skew information of the data lanes with respect to each other.
- OTL4.10: In this mode, 10 SERDES links are used. The transmit data is distributed over ten physical lanes. Each physical lane contains two logical lanes, for a total of 20 logical lanes. Multi-lane de-skew at the far end is performed as per G.709, Annex C.

MLD Features:

- Provides SFI-based multi-lane alignment
- Provides OPSMnk Trail Termination (multi-lane alignment) according to G.709 and G.798 standards:
  - In the receive path: (physical lane to logical lane) bit dis-interleaving, OTL frame alignment, OTL lane alignment recovery, multi-lane marker removal / OTUk FAS recreation, lane marker and delay processing (lane deskew) and 16-byte block multiplexing (16-byte interleaving)
  - In the transmit path: 16-byte block distribution and rotation (16-byte lane dis-interleaving), multi-lane marker insertion and (logical lane to physical lane) bit interleaving
- Provides OPSMnk/OTUk adaptation functions of frame alignment in the receive path according to the G.709 and G.798 standards
- Optionally, provides OPSMnk/OTUk adaptation functions of scrambling / de-scrambling according to the G.709 and G.798 standards
- Optionally, provides OPSMnk/OTUk adaptation functions of multi-frame alignment in the receive path according to the G.709 and G.798 standards

### 5.7.3.2 FEC Features

The SYSOTN subsystem supports RSFEC Encode and Decode blocks, which are compliant with the G.975 and G.709 standards.

The FEC block provides OTU4 stream based FEC corrected 0's, corrected 1's and uncorrectable error count statistics. FEC statistics are typically accumulated over 1 second intervals, and are stored in performance monitor counters. These counters are updated by 1 of 2 methods; 1) via a rising edge on the SYSOTN lclk primary input, or 2) via writes to the FEC statistics UPDATE register bit. The table below describes the three types of FEC statistics supported by SYSOTN.

**Table 35 FEC Statistics**

Statistics	Statistic Description
UNCORRECTABLE_SYMBOL_COUNTER_VALUE	The 32-bit uncorrectable symbol counter value bits contain the count of the number of symbols which the FEC decoder was unable to correct. This counter saturates when the maximum count is reached.
FEC_CORRECTED_ZEROS_COUNTER_VALUE	The 32-bit FEC corrected zeros counter value bits contain the count of the number of corrected bits which were '1' on the optical line and corrected to '0' by the FEC decoder. This counter saturates when the maximum count is reached.
FEC_CORRECTED_ONES_COUNTER_VALUE	The 32-bit FEC corrected ones counter value bits contain the count of the number of corrected bits which were '0' on the optical line and corrected to '1' by the FEC decoder. This counter saturates when the maximum count is reached.

Two types of FEC bypass are supported:

- Decoder bypass: This disables the FEC decoder, so that the FEC columns are passed through the RS-FEC untouched. There is no reduction in latency through the RS-FEC processing logic in this case. Note that FEC encoder bypass is not supported.
- True bypass: The RS-FEC processing logic is completely bypassed for both encoder and decoder, which offers maximum latency reduction.

### 5.7.3.3 OTU Features

The OTU:

- Optionally performs scrambling prior to FEC encoding, and de-scrambling after FEC decoding.
- Performs multi-frame alignment
- Performs OTUk Trail Termination according to G.709 and G.798 standards:
  - In the receive path: SMOH (TTI, RES, BDI, IAE, BEI/BIAE and BIP8) extraction
  - In the transmit path: SMOH (TTI, RES, BDI, IAE, BEI/BIAE and BIP8) insertion
- Performs OTUk/ODUk adaptation according to G.709 and G.798 standards:
  - In the receive path: ODUk FAS and MFAS generation, OTUk to ODUk frame extraction, ODUk-LCK / ODUk-AIS generation and normal signal / ODUk-LCK / ODUk-AIS selection
  - In the transmit path: ODUk-LCK generation, normal signal / ODUk-LCK selection and ODUk to OTUk frame mapping

### 5.7.3.4 Defects, Defect Correlations, Consequential Actions and Performance Monitoring Events

Defects, defect correlations, consequential actions, and performance monitoring events are treated on a layer-by-layer basis for the OTN protocols. The SYSOTN subsystem has four functional layers:

- OPSMn Trail Termination layer
- OPSMn/OTUk Adaptation layer
- OTUk Trail Termination layer
- OTUk/ODUk Adaptation layer

These functional layers are implemented in the OPSM4 core and the OTN\_OTU. [Table 36](#), [Table 37](#), and [Figure 24](#) summarizes the defects, defect correlations, consequential actions, and performance monitoring events and their triggers supported by each layer of the SYSOTN subsystem.

**Table 36** Event Processing in the OPSM4

Name	Type	Activation / De-activation	Functional Layer	Direction Source (So): Transmit Sink (Sk): Receive	References
dLOS-P[i]	Defect	G.798: 6.2.1.1	OPSMnk_TT	Sk	G.798: 11.2.2.2 (section 11.2.2.2 has been mis-labeled in G.798 as 11.2.1.2) & 6.2.1.1
dLOL	Defect	G.798: 6.2.5.5	OPSMnk_TT	Sk	G.798: 11.2.2.2 (section 11.2.2.2 has been mis-labeled in G.798 as 11.2.1.2) & 6.2.5.5
dLOFLANE[i]	Defect	G.798: 6.2.5.6	OPSMnk_TT	Sk	G.798: 11.2.2.2 (section 11.2.2.2 has been mis-labeled in G.798 as 11.2.1.2) & 6.2.5.6
aTSF	Consequential Action	$\Sigma\{dLOS-P[i]\}$ or dLOL or $\Sigma\{dLOFLANE[i]\}$	OPSMnk_TT	Sk	G.798: 11.2.2.2 (section 11.2.2.2 has been mis-labeled in G.798 as 11.2.1.2)
cLOS	Defect Correlation	$\Sigma\{dLOS-P[i]\}$	OPSMnk_TT	Sk	G.798: 11.2.2.2 (section 11.2.2.2 has been mis-labeled in G.798 as 11.2.1.2)
cLOL	Defect Correlation	(dLOL or $\Sigma\{dLOFLANE[i]\}$ ) and (not $\Sigma\{dLOS-P[i]\}$ )	OPSMnk_TT	Sk	G.798: 11.2.2.2 (section 11.2.2.2 has been mis-labeled in G.798 as 11.2.1.2)
dAIS	Defect	G.798: 6.2.6.3.1	OPSM/OTUk_A	Sk	G.798: 11.3.4.3 & 6.2.6.3.1
dLOF	Defect	G.798: 6.2.5.1	OPSM/OTUk_A	Sk	G.798: 11.3.4.3 & 6.2.5.1
cLOF	Defect Correlation	dLOF and (not dAIS) and (not AI_TSF-P)	OPSM/OTUk_A	Sk	G.798: 11.3.4.3

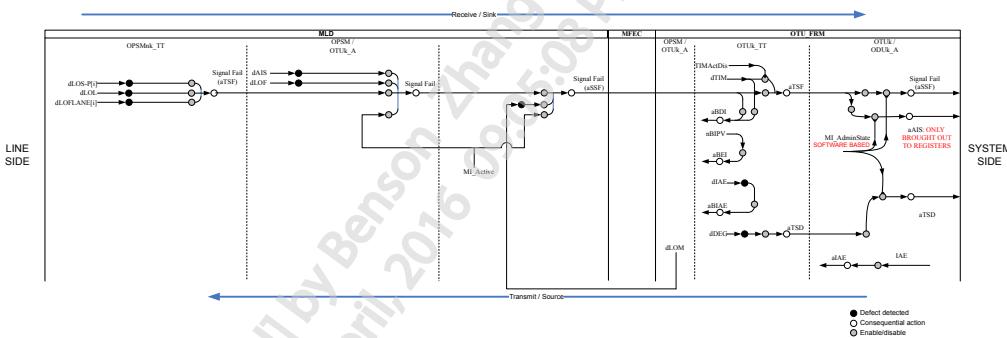
**Table 37** Event Processing in the OTU\_FRM Core

Name	Type	Activation / De-activation	Functional Layer	Direction Source (So): Transmit Sink (Sk): Receive	References
aIAE	Consequential Action	IAE (G.798: 13.3.1.1)	OTUk/ODUk_A	So	G.798: 13.3.1.1
dLOM	Defect	G.798: 6.2.5.2	OPSM/OTUk_A	Sk	G.798: 11.3.4.3 & 6.2.5.2
aSSF	Consequential Action	dAIS or dLOF or dLOM or AI_TSF-P or (not MI_Active)	OPSM/OTUk_A	Sk	G.798: 11.3.4.3

Name	Type	Activation / De-activation	Functional Layer	Direction Source (So): Transmit Sink (Sk): Receive	References
cLOM	Defect Correlation	dLOM and (not dLOF) and (not dAIS) and (not AI_TSF-P)	OPSM/OTUk_A	Sk	G.798: 11.3.4.3
dTIM	Defect	G.798: 6.2.2.1	OTUk_TT	Sk	G.798: 13.2.1.2 & 6.2.2.1
dDEG	Defect	G.798: 6.2.3.4	OTUk_TT	Sk	G.798: 13.2.1.2 & 6.2.3.4
dBDI	Defect	G.798: 6.2.6.6.1	OTUk_TT	Sk	G.798: 13.2.1.2 & 6.2.6.6.1
dIAE	Defect	G.798: 6.2.6.10.1	OTUk_TT	Sk	G.798: 13.2.1.2 & 6.2.6.10.1
dBIAE	Defect	G.798: 6.2.6.11.1	OTUk_TT	Sk	G.798: 13.2.1.2 & 6.2.6.11.1
aBDI	Consequential Action	CI_SSF or dTIM	OTUk_TT	Sk(So)	G.798: 13.2.1.2
aBEI	Consequential Action	nBIPV	OTUk_TT	Sk	G.798: 13.2.1.2
aBIAE	Consequential Action	dIAE	OTUk_TT	Sk	G.798: 13.2.1.2
aTSF	Consequential Action	CI_SSF or (dTIM and (not TIMActDis))	OTUk_TT	Sk	G.798: 13.2.1.2
aTSD	Consequential Action	dDEG	OTUk_TT	Sk	G.798: 13.2.1.2
cTIM	Defect Correlation	dTIM and (not CI_SSF)	OTUk_TT	Sk	G.798: 13.2.1.2
cDEG	Defect Correlation	dDEG and (not CI_SSF) and (not (dTIM and (not TIMActDis)))	OTUk_TT	Sk	G.798: 13.2.1.2
cBDI	Defect Correlation	dBDI and (not CI_SSF) and (not (dTIM and (not TIMActDis)))	OTUk_TT	Sk	G.798: 13.2.1.2
cSSF	Defect Correlation	CI_SSF	OTUk_TT	Sk	G.798: 13.2.1.2
pN_DS	Performance Monitoring	CI_SSF or dTIM	OTUk_TT	Sk	G.798: 13.2.1.2
pF_DS	Performance Monitoring	dBDI	OTUk_TT	Sk	G.798: 13.2.1.2
pN_EBC	Performance Monitoring	$\Sigma\{nN_B\}$	OTUk_TT	Sk	G.798: 13.2.1.2
pF_EBC	Performance Monitoring	$\Sigma\{nF_B\}$	OTUk_TT	Sk	G.798: 13.2.1.2

Name	Type	Activation / De-activation	Functional Layer	Direction Source (So): Transmit Sink (Sk): Receive	References
pBIAE	Perform- ance Moni- toring	dBIAE	OTUk_TT	Sk	G.798: 13.2.1.2
pIAE	Perform- ance Moni- toring	dIAE	OTUk_TT	Sk	G.798: 13.2.1.2
aSSF	Consequen- tial Action	AI_TSF and (not MI_Ad- minState = LOCKED)	OTUk/ ODUK_A	Sk	G.798: 13.3.1.2
aAIS	Consequen- tial Action	AI_TSF and (not MI_Ad- minState = LOCKED)	OTUk/ ODUK_A	Sk	G.798: 13.3.1.2
aSSD	Consequen- tial Action	AI_TSD and (not MI_Ad- minState = LOCKED)	OTUk/ ODUK_A	Sk	G.798: 13.3.1.2

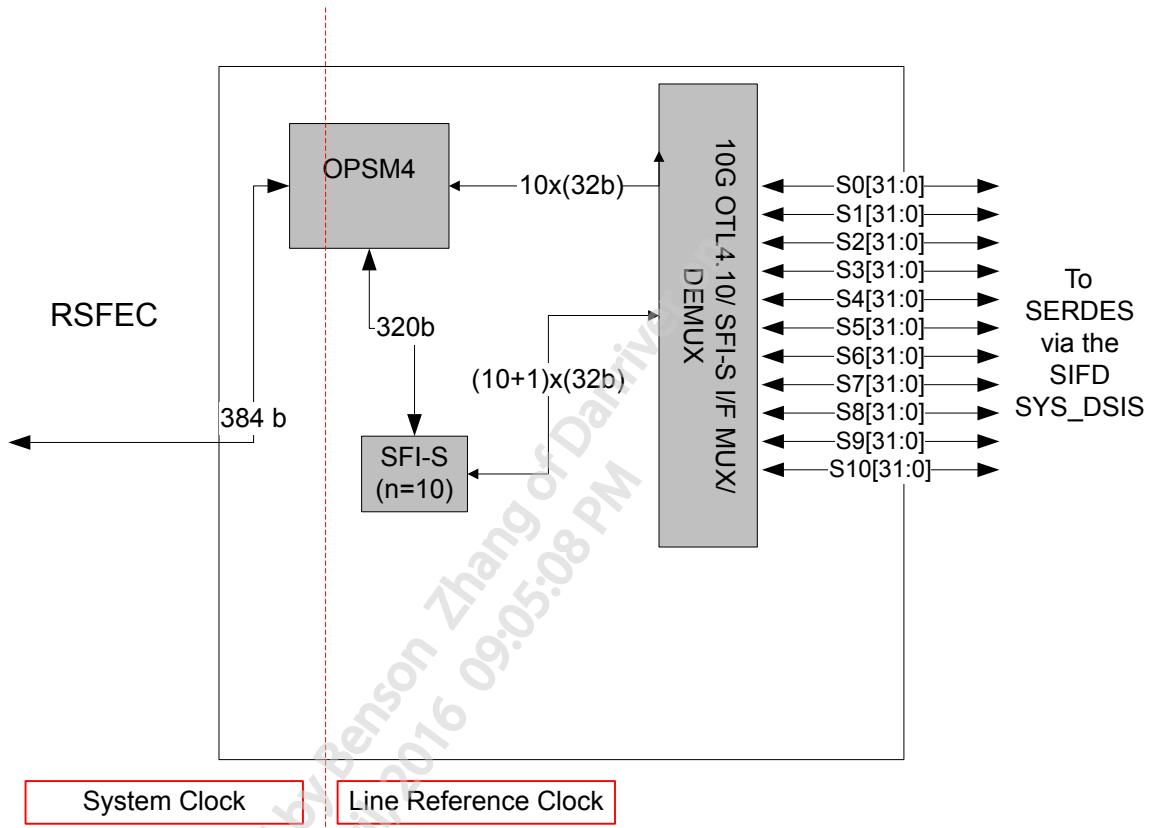
Figure 24 SYSOTN Consequential Actions Processing



The consequential actions above are implemented in hardware, but they are controlled (enabled and disabled) through software.

## 5.7.4 OPSM4/ SFIS Block

Figure 25 Multi lane Distribution Block Diagram



The subsystem supports a single OTU4 data path through either 1 x OTL4.10 or 1 x SFI-S(n=10) MLD interfaces:

On the system interface SERDES side, data is transmitted and received using up to 11 DSIs. In SFI-S mode, all 11 DSIs are used; 10 for data, and the 11th lane to convey skew information between the 10 data lanes. In OTL4.10 mode, 10 DSIs are used. In summary:

- OTU4 over OTL4.10, 10 x DSIs lanes processed through the OPSM4 OTL4.10 datapath logic.
- OTU4 over SFI-S (10+1), 11 x DSIs lanes processed through the SFI-S TSB and the OPSM4 OTU4 datapath.

In the transmit direction, the system-side SERDES provides the clock to pull data out of the MLD alignment units. Data from the core is pushed to the SYSOTN at an over-speed rate. An XOFF back-pressure mechanism is used to regulate the flow of the transmit data so that it is equal to the OTU4 line transmission rate.

In the receive direction, the data from the system-side SERDES is input to the SYSOTN at the OTU4 line data rate. There is no way to mitigate the flow of traffic from the line side SERDES in the receive direction.

### 5.7.4.1 SFI-S

#### 5.7.4.1.1 SFI-S RX

The SFI-S receive MLD component performs the receive MLD functionality as defined in OIF-SFI-S-01.0 [46], providing the translation from the ingress SERDES multi-lane interface to a single stream interface. The main function of this block is to receive 10 lanes of SERDES stream data and one lane of SERDES deskew data, using the deskew lane to align the other lanes and merge the data lanes into a single stream. The receive MLD transitions each lane's data from the individual recovered receive clock domains of the source SERDES to the common recovered receive clock domain of the deskew lane.

The SFI-S receive MLD performs the following functions:

- Deskewing and bit interleaving of 10 data lanes into a single data lane
- Reception of deskew channel based on 10-lane bit level data stripes
- Bit-level deskew of 10 lanes using the deskew channel

**Note:**

- The deskew algorithm is based on alternating between even and odd priority of lanes to allow optimal lane alignment for the number of lanes in a given SFI-S configuration.
- The deskew algorithm operates continuously to monitor skew tracking.

- Assertion of an error indication signal when loss of lane alignment occurs

#### 5.7.4.1.2 SFI-S TX

The SFI-S transmit MLD component performs transmit MLD functionality as defined in OIF-SFI-S-01.0 [46], providing the translation from the single stream interface to the egress SERDES multi-lane interface. The main function of this block is to receive a stream of parallel data and to distribute this stream to the output lanes and a deskew lane.

The SFI-S transmit MLD performs the following functions:

- Bit de-interleaves a single data lane into 10 data lanes
- Creates a deskew channel based on 10 lanes of data
- Transmits the deskew lane alongside the data lanes

**Note:**

- The deskew creation algorithm is based on alternating between even and odd priority of lanes to allow optimal lane alignment for the number of lanes

## 5.7.5 RSFEC

### 5.7.5.1 G.709 compliant Reed-Solomon FEC (RSFEC)

The SYSOTN RSFEC block re-uses the LINEOTN subsystem RSFEC design. The SYSOTN uses the RSFEC block in 100G mode only.

## 5.7.6 Framer

The SYSOTN OTU framer is re-used from the LINEOTN subsystem. See [Section 5.6.8](#) for a description of this block.

## 5.7.7 OTN Frame Pulse Interface

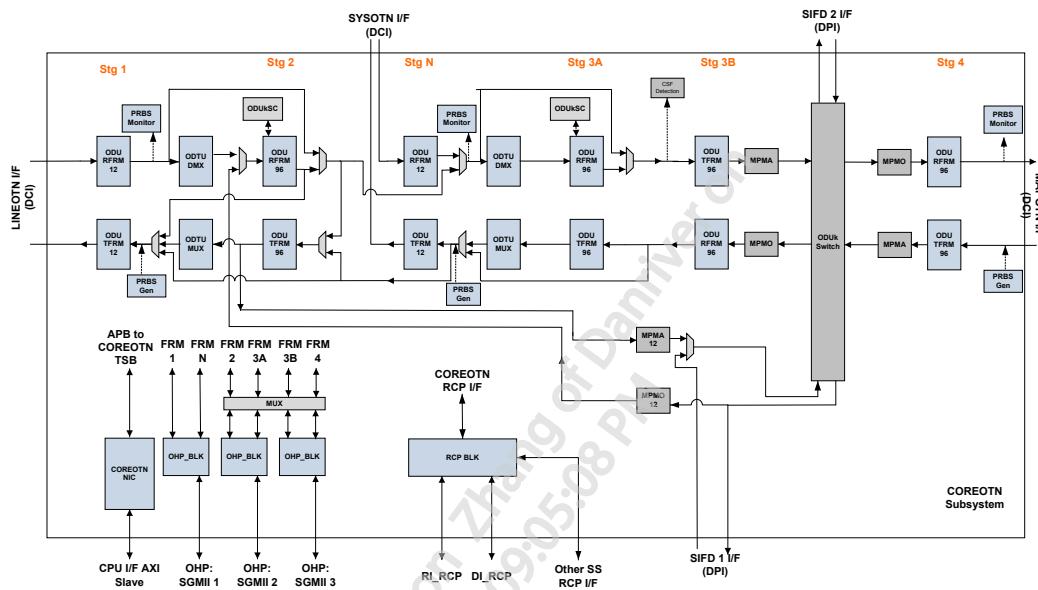
See [Section 5.6.4.8](#).

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:08 PM

## 5.8 Core OTN (COREOTN) Subsystem

### 5.8.1 Block Diagram

Figure 26 COREOTN Block Diagram



### 5.8.2 Terminology

The following notation is used in this section to describe the COREOTN subsystem:

- "LO" designates Low Order, as per the G.709 standard.
- "HO" designates High Order, as per the G.709 standard.
- "MO" is defined in this document as the Middle Order ODUk layer. This denotes the intermediate ODUk container during a 2-stage demux/mux mode of operation.

### 5.8.3 Overview

The COREOTN subsystem processes ODU and OPU overhead, data, and consequential actions across multiple channels concurrently. The COREOTN subsystem has three bidirectional ODUk interfaces:

- Line interface (towards the LINEOTN subsystem)
- Client interface (towards the MAPOTN subsystem)
- Backplane interface (towards the SIFD subsystem)

The subsystem supports any combination of 10G, 40G, and 100G OTN protocols up to an aggregate of 120G multi-channel support.

The COREOTN supports up to two levels of ODTUjk multiplexing using AMP or GMP protocols. From an initial ODUk, the COREOTN is able to demultiplex into an intermediate set of ODUj streams and subsequently demultiplex those ODUj streams into a set of ODUh streams. Note that the two levels of ODTUjk multiplexing is optional. The COREOTN can also support single level ODTUjk multiplexing. There is no restriction on the type of ODU in ODUh, ODUj or ODUk, although it is subject to  $k > j > h$  (as long as it is a legal combination, as per G.709 and Gsup43).

The COREOTN subsystem complies with the relevant sub-clauses of G.709, G.798, G.806 and Gsup43 data processing from the ODUk layers to the OPUk layers. All client layer de-mapping/mapping protocols are handled outside this subsystem except muxing to/from LO ODU clients. The subsystem can support up to two stages of demuxing/muxing to incorporate up to two stages of LO ODU clients.

The subsystem's architecture is based on a multi-channel bus that can support up to a 120G datapath. Each stream roughly contains 1.25G of data that can belong to one channel. Alternatively, multiple streams can be allocated to one channel. A single stream uses one calendar cycle of the multi-channel bus. Calendar cycles to channel configuration can be modified by each functional unit along the datapath and are further discussed in the COREOTN block descriptions.

As listed below with the supported general rate containers, there are three distinct ODUk data interfaces to the subsystem:

- Line interface: Consists of one multi-channel DCI port interfacing to/from the LINEOTN subsystem. The ODUk rates supported in this interface are a combination of 10G, 40G, and 100G OTN container rates.
- Client interface: Consists of one multi-channel DCI port interfacing to/from the MAPOTN subsystem. The ODUk rates supported in the interface are any valid rates between 1.25G to 100G, including variable ODUFlex container rates.
- Backplane interface: Consists of two 120G multi-channel DPI ports interfacing to/from the SIFD subsystem. The ODUk rates supported in the interface are any valid rates between 1.25G to 100G, including variable ODUflex container rates.

**Note:** The ODU rates supported by the subsystem are: ODU0, ODU1, ODU1e, ODU1f, ODU2, ODU2e, ODU2f, ODU3, ODU3e1, ODU3e2, ODU4 and ODUflex rates from 1.25G to 100G.

As mentioned above, the backplane interface is a packet-based interface to the COREOTN. However, through the OPSA, the packets received are monitored on the egress of the ODUk switch towards the line or client interface to convert the DPI bus to a DCI bus. In the ingress from the line or client interface to the ODUk switch, the DCI bus is converted to a DPI in order to interface to the backplane. The client interface carries processed and framed ODUk data, but expects valid OPUk data through the DCI port for the MAPOTN to demap/map client signals. The line interface carries processed and framed ODUk data through the DCI port for the LINEOTN to source or sink the OTUK layer.

In general, any of the three interfaces to the subsystem can process data through the COREOTN and communicate the processed data out to any of the three interfaces. [Figure 27](#) and [Figure 28](#) show the combinations of the transmit and receive configurable paths (with reference to the line interface) from a functional perspective. Each color (blue, green and orange) indicates the different datapath modes that can be active per channel. This document also explains several submodes in connection with these datapaths.

Figure 27 COREOTN Line Interface Receive Datapath

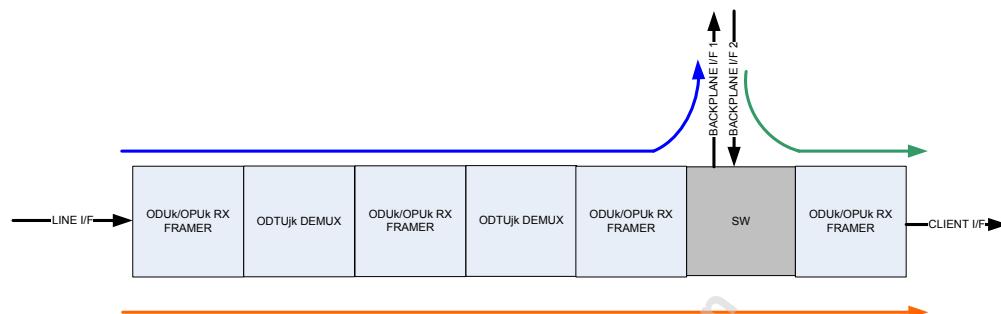
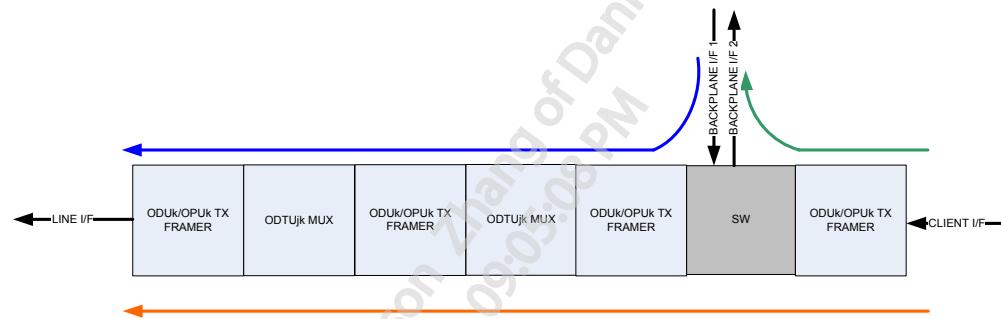


Figure 28 COREOTN Line Interface Transmit Datapath



With regards to ODUk layer processing, the COREOTN subsystem terminates or generates all ODUk layer functionality, including the ODU Connection function (part of the G.798 standard). This function is provided through the ODUk switch interface to enable different forms of SNC protection switching to and from other ODUk layers from different ODUk layer processing systems. Apart from the ODUk layer processing, when data interfaces to and from the line interface, the ODUk layer can optionally mux/demux to/from several LO, MO and HO ODUk layers. The subsystem supports the following mux/demux protocols: GMP 1.25G, AMP 1.25G and AMP2.5G.

As part of its dynamic reconfigurability features, the COREOTN can handle hitless ODUflex resizing, OTN PTP, and ODUk delay test measures from both PM and TCM layers. It also provides several forms of protection switching and fall back, consequential actions on the ODUk layer, user-configured performance monitoring, and frame-by-frame ODUk/OPUk overhead insertion.

With respect to external ports, the COREOTN has three OHP (overhead add/drop port) interfaces and two Ring Control Port (RCP) interfaces. The first OHP interface is unique in that it only interfaces to a 12-channel framer that uses Stage 1 and Stage N (for SYSOTN users). Each of the other two OHP ports can be configured to listen to any two of the four framer stages: Stage 2, 3A, 3B, and 4.

The RCP port has two interfaces:

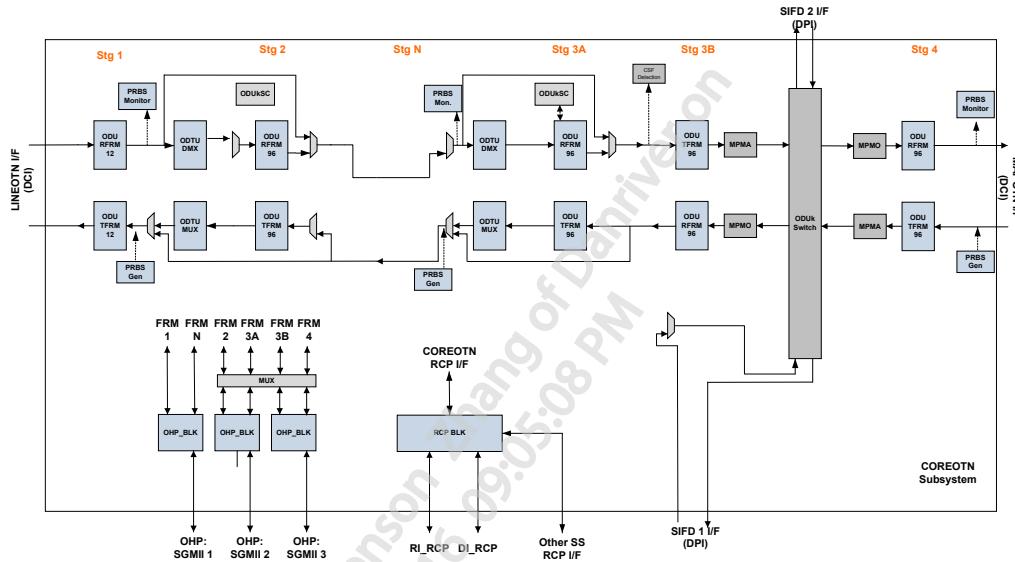
- A bidirectional RI\_RCP (remote information) interface. This interface is used to communicate backward information between Rx and Tx framers.
- A transmit-only DI\_RCP (defect Information) interface. This interface is used to communicate defect information from the OTUk, ODUk, OPUk and Client layers from across the chip at 125  $\mu$ s intervals.

It also communicates performance monitoring information for ODUk, SDH, and ODU-PRBS layers at a 5 ms interval.

## Configuration Options

The COREOTN subsystem can be configured with or without SYSOTN functionality. When configured without SYSOTN support, the COREOTN provides five stages of ODUk frame processing, as shown in [Figure 29](#). In this figure, these stages are designated 1, 2, 3a, 3b, and 4. Stage N is not included.

[Figure 29](#) COREOTN Block Diagram - Without SYSOTN Interface

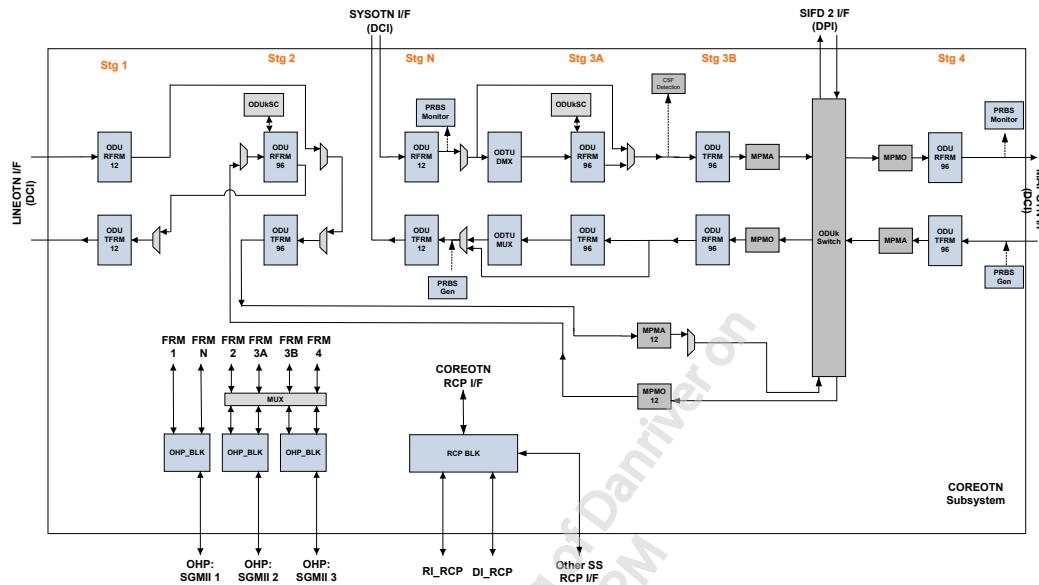


- Stage 1 is at the left most of the diagram (close to the line interface)
- Stage 2 is the middle framer between two mux/demux blocks
- Stage 3a is the last full framer before the ODUk switch
- Stage 3b is configured as a TCM processor
- Stage 4 is right of the ODUk switch (close to the client interface)

Stage 4 is sometimes referred to as the SNC framing stage since it is used during SNC protection switching. It returns backward indications which are then bridged to both the working and protect ODUk signals using either the device ODUk switch or an external fabric (via the SIFD). Likewise, the working or protect ODUk can be selected by the ODUk switch; or the ODUk can be selected by an external fabric and sent to the stage 4 framer via the SIFD.

The COREOTN can also be configured to interface to the SYSOTN subsystem to enable 100G OTN-OTN muxponder and transponder solutions. In this mode of operation, 100G of framed ODU traffic is interfaced from the LINEOTN or MAPOTN towards the SYSOTN. This mode also provides one stage muxing towards the SYSOTN subsystem. The following figure highlights how the datapath muxes are configured to enable SYSOTN configuration mode.

Figure 30 COREOTN Block Diagram with SYSOTN Configuration Mode



As seen in this diagram, this mode of operation provides a new stage of framers (Stage N). This stage is primarily used to interface to the SYSOTN subsystem. The datapath can be understood as follows:

- SYSOTN interface  $\iff$  Stage N framer (ODU4)  $\iff$  Mux/Demux (optional)  $\iff$  Stage 3A framer (ODU2(e))  $\iff$  Stage 3B framer  $\iff$  ODUk switch.

Traffic from the ODUk switch goes out the MAPOTN interface as follows:

- ODUk switch  $\iff$  Stage 4 framer  $\iff$  MAPOTN interface

Alternatively, the traffic to the LINEOTN interface is as follows:

- ODUk switch  $\iff$  Stage 2 framer  $\iff$  Stage 1 framer  $\iff$  LINEOTN interface

Alternatively, the traffic to the SIFD interface is as follows:

- SYSOTN/MAPOTN/LINEOTN interface  $\Leftrightarrow$  ODUk switch  $\Leftrightarrow$  SIED Interface

There is only a single 120G interface to the SIED subsystem supported.

Apart from the datapath, all other aspects of this subsystem (such as RCPs and OHPs) are supported. The Stage N framer's overhead data is interfaced to the first OHP interface.

## 5.8.4 Features

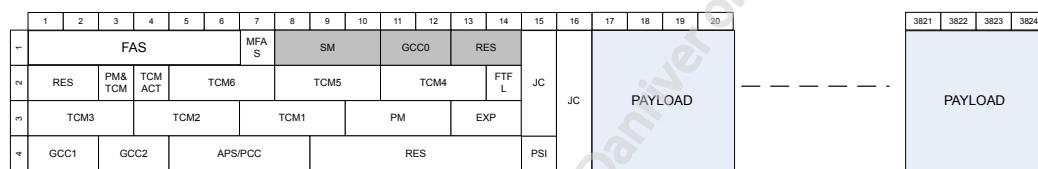
### 5.8.4.1 Feature Overview

This section highlights the main features of the COREOTN subsystem.

### 5.8.4.2 OTN Frame Structure and Overhead Processing

Figure 31 illustrates the OTN frame structure supported within the COREOTN. As shown, the frame size is 4 rows by 3824 columns and each unit is byte granularity.

Figure 31 OTN ODUk Frame Structure



ODUk and OPUk layers are processed in the COREOTN subsystem. In the diagram above, the main overhead bytes of interest have a white background, and the OTUk layer overhead bytes are shown with a greyed-out background. The COREOTN insert/extract(s) OTUk layer overhead in the HO frames only through the ODU\_OHP block(s). This enables external overhead insertion/extraction. The COREOTN does not process any OTUk layer functions. Instead, these functions are processed by the LINEOTN subsystem.

For overhead extraction, the COREOTN supports the following features for the respective ODUk/OPUk layer:

- Register extraction of the RES, PM&TCM, TCM ACT, EXP, GCC1, GCC2 bytes frame-by-frame (only one frame is stored). Users can lock the bytes to view what is in the registers for these fields (*at the RFRM block*)
- Register extraction of 10 forward FTFL fields (bytes [9:0]) and 10 backward FTFL fields (bytes [9:0]) (*at the RFRM block*)
- Register extraction of the accepted TTI, accepted STAT, accepted APS bytes for all levels (*at the RFRM and DEMUX blocks*)
- Register extraction of the RCOH fields for firmware or software processing and execution of the 'Hitless ODUflex Resizing protocol' (*at the DEMUX block*)
- All overhead fields are 'extractable' through the OHP\_BLOCK (external interface) (*from the RFRM to the OHP blocks*)

For overhead insertion, the COREOTN supports the following features for the respective ODUk/OPUk layer:

- If enabled, one register per overhead field for RES, PM&TCM, TCM ACT, EXP, GCC1, GCC2 is inserted frame-by-frame. If disabled, these fields are passed through (*at the TFRM block*)
- If enabled, register insertion capability of 10 fields of forward FTFL (bytes [9:0]) and 10 backward FTFL fields (bytes [9:0]). Forward and backward fields can be selectively passed-through or inserted (*at the TFRM block*)
- Full field insertion of the TTI, STAT, PT, MSI, APS bytes through registers (*at the TFRM and MUX blocks*)

- Register insertion capability of the RCOH fields for firmware or software processing and execution and execute of the 'Hitless ODUflex Resizing protocol' (*at the MUX block*)
- Even if register insertion or pass-through is enabled, the external OHP\_Block port can override any of the ODUk/OPUk fields (*in the OHP to TFRM blocks*). Hence the insertion priority is OHP/RCP block takes the highest priority, followed by internal TFRM registers and then the passthru overhead fields.

The transmission of the OTN frame structure is primarily on the DCI bus. However, OTN data can also be packetized through OPSA-specific blocks (packet makers and monitors) into the DPI bus. When OTN frames are in the DPI bus, the notion of frame structure is lost and the data is treated as packets with sizes configured by the packet makers. When in DCI bus format, the frame structure has to be maintained along with the associated CBR characteristics (i.e. Frame Pulse, Multi-frame number, Signal Fail). The transmission order of OTN frames, regardless of DPI or DCI, is always from column 1 to column 3824 of row 1 and down to row 2, column 1 to column 3824 and so on until end of row 4. The data is packed in 'Big-Endian' format.

#### 5.8.4.3 Transmit Features

The LINEOTN DCI transmit port carries framed ODUK data that is ready to be transmitted and processed to the OTUk layers. Optionally, the OTUk overhead can be inserted through the overhead insertion ports prior to transmission to the LINEOTN subsystem. The following container rates are supported on this egress DCI port:

- 10G: ODU1e, ODU1f, ODU2, ODU2e, ODU2f,
- 40G: ODU3, ODU3e1, ODU3e2
- 100G: ODU4

The SYSOTN DCI transmit port carries framed ODUK data ready to be transmitted and processed to the OTUk layers. The OTUk overhead can be optionally inserted through the overhead insertion ports prior to transmission to the SYSOTN subsystem.

The following container rate is supported on this egress DCI port:

- 100G: ODU4

Since the MAPOTN DCI transmit port carries terminated ODUK with valid OPUk bytes, the DCI port frame size is the ODUK container size. The following container rates are supported on this egress DCI port:

- ODUflex: GFP and CBR type
- 1.25G: ODU0
- 2.5G: ODU1
- 10G: ODU1e, ODU2, ODU2e, ODU1f, ODU2f
- 40G: ODU3, ODU3e1, ODU3e2
- 100G: ODU4

The backplane interface is provided through the SIFD subsystem. The COREOTN interfaces to the SIFD through a DPI port.

#### 5.8.4.4 Receive Features

The LINEOTN DCI receive port accepts terminated OTUk frames that are ready to be processed in the ODUk layer. Optionally, OTUk overhead can be extracted through the overhead extraction port. The receive LINEOTN supports the following container rates:

- 10G: ODU1e, ODU1f, ODU2, ODU2e, ODU2f
- 40G: ODU3, ODU3e1, ODU3e2
- 100G: ODU4

The SYSOTN DCI receive port accepts terminated OTUk frames that are ready to be processed in the ODUk layer. Optionally, OTUk overhead can be extracted through the overhead extraction port. The receive SYSOTN supports the following container rate:

- 100G: ODU4

The MAPOTN DCI receive port carries framed ODUk data with valid OPUk OH bytes from the mapped client. The COREOTN inserts the rest of the ODUk. The receive MAPOTN supports the following container rates:

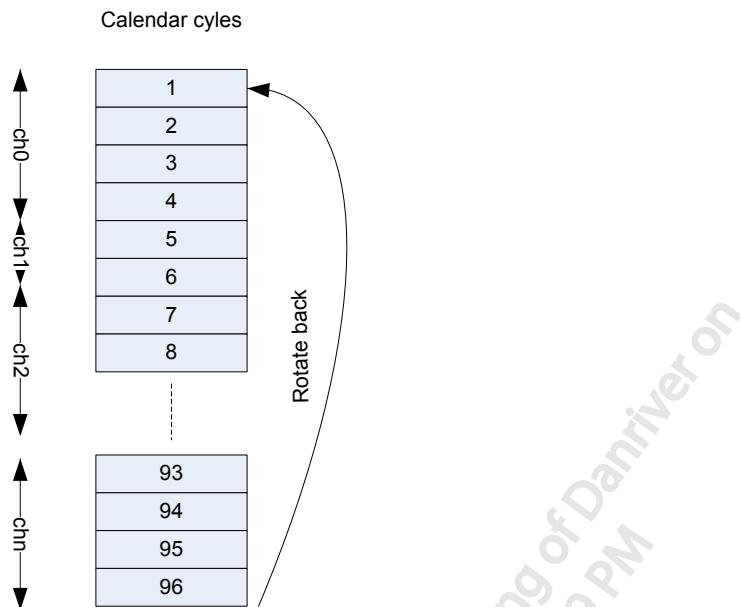
- ODUflex: GFP and CBR type
- 1.25G: ODU0
- 2.5G: ODU1
- 10G: ODU1e, ODU1f, ODU2f, ODU2, ODU2e
- 40G: ODU3, ODU3e1, ODU3e2
- 100G: ODU4

The backplane interface is provided through the SIFD subsystem. The COREOTN interfaces to the SIFD subsystem through a DPI port.

#### 5.8.4.5 Multi-channel Processing

A DCI structure provides the main multi-channel bus through the COREOTN subsystem. The DPI is only used when interfacing to the ODUk switch buffer and the backplane interface. Regardless of using DPI or DCI, the multi-channel bus is scheduled with a similar architecture. There are up to 96 calendar cycles on the multi-channel bus. N calendar slots can be allocated to one channel, with a minimum of 1 slot per channel or a maximum of 96 slots per channel. A NULL channel (ID 0xFF) can be inserted to evenly distribute the bandwidth allocation on the multi-channel bus. A single channel is a single ODUk stream. [Figure 32](#) provides an example of allocating calendar slots to channels.

Figure 32 Multi-channel Bus



Each calendar cycle is roughly 1.3 G. For this reason, the calendar allocation for a channel is based on multiples of 1.25G calendar entries. [Table 38](#) shows the number of calendar entries required for each rate supported by the COREOTN subsystem.

Table 38 Calendar Configuration on the Multi-channel Bus

Channel Rate	ODUk Container Types	Number of Calendar Allocations Over 96 Calendar Slots
1.25G	ODU0	1
2.5G	ODU1	2
10G	ODU1e, ODU1f, ODU2, ODU2e, ODU2f	8 9 for ODU2e/ODU2f/ODU1f if LO->HO muxing requires 9 timeslots for certain containers.
40G	ODU3, ODU3e1, ODU3e2	32 31 for ODU3 when muxing ODU3->ODU4 33 for ODU3e1/ODU3e2 when muxing ODU3e1/3e2 -> ODU4
100G	ODU4	80
Flex rate	ODUflex (any rate between 1.25G and 100G)	1-80

Note that all of these schedulers are configured at various calendar scheduler blocks that are either located throughout the subsystem, or interface the subsystem. Within the COREOTN, the calendar schedulers are located at the mux/demux blocks, MPMA/MPMO and at the ODUk switch egress ports. Each calendar has an active and standby page to do hitless re-configuration of channels without affecting active channels (for G.7044 or APS events). Essentially, the standby page contains the full 96 calendar entry that software would need to hit the appropriate changes to the calendar and sequence an update

command (to do switch-over from active to standby). The specific calendar will do the update in a position where it is 'safe' to do so (i.e. when active channels are not affected).

Calendar allocation is directly related to timeslot allocation. A ODU2e being muxed into an ODU3 container requires 9 calendar cycles since it uses 9 timeslots of the ODU3. However, when the ODU2e is muxed into an ODU3e2, the ODU2e requires 8 calendar entries since it uses 8 timeslots of the ODU3e2. In general, an ODU2e as a HO requires 8 timeslots and 8 calendar entries like all 10G containers. Similar conditions apply as per table above for muxing ODU3/ODU3e1/ODU3e2 into ODU4.

Although the COREOTN TDM structure is agnostic to the calendar scheduling of multiple channels onto one bus, PMC recommends evenly spreading out the specific channels based on the TDM bus. The following provides an example of channel scheduling from the LINEOTN, MAPOTN, SYSOTN or SIFD ports:

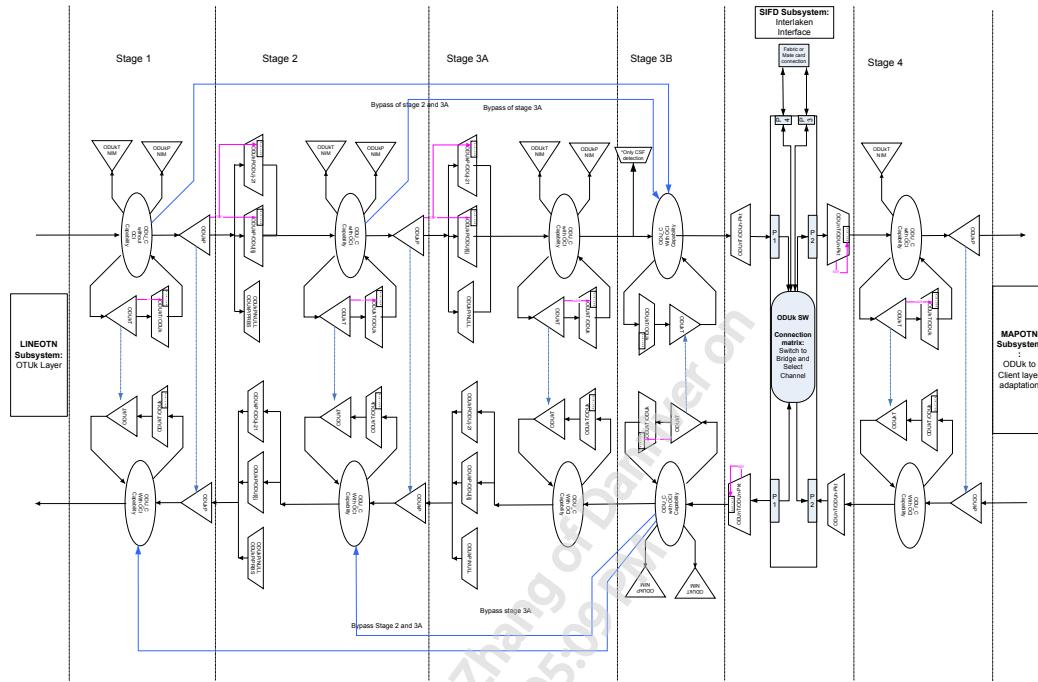
- If the ODU2 traffic requires 8 calendar slots, the scheduling of these calendars can be every 12th calendar slot
- If the ODU3 traffic requires 32 calendar slots, the scheduling of these calendars can be every 3rd calendar slot
- If the ODU4 traffic requires 80 calendar slots, the scheduling of these calendars can be 10/12 calendar slots up to 96 calendars
- For MO and LO channels created due to demuxing, the HO calendar slots can be inherited by the respective children (LO, MO) channels

#### 5.8.4.6 ODUk/OPUk Layer Processing Functions

The ITU-G.798 standard defines processing functions for how to handle each of the OTN layers with respect to functionality, defect detection, consequential actions, and performance monitoring. The standard highlights all possible processing data flows and functions when they are cascaded together. For the COREOTN subsystem, these functions can be categorized through the datapath showing how G.798 is realized through the subsystem and its respective blocks.

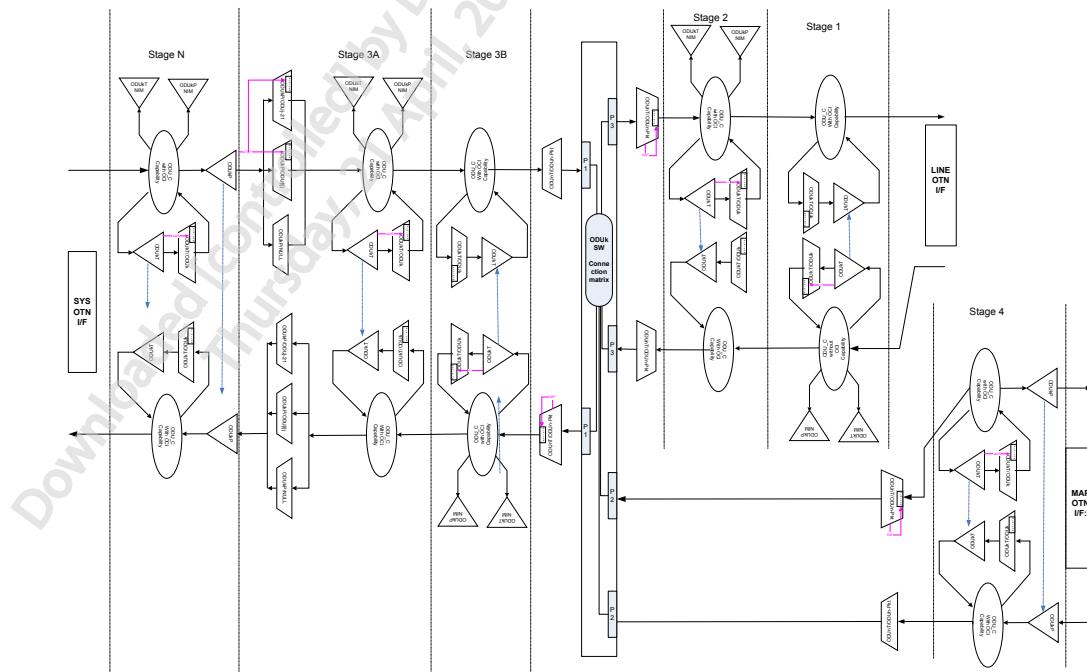
[Figure 33](#) illustrates functions with respect to the COREOTN (for detailed definitions of each functions please refer to G.798 standard). These functions operate on one channel at a time for multi-channel operation. The context of each of these functions is switched based on pre-determined calendar scheduling.

Figure 33 COREOTN ODUk/OPUk Layer Processing Functions



The atomic functions with the SYSOTN interface are as follows.

Figure 34 SYSOTN Atomic Functions



In general, each of the functions highlighted in [Figure 33](#) above are implemented as per the G.798 standard except the items noted within the blocks. The PRBS atomics are not listed in the atomics above, as this is used primarily as a test feature. The PRBS capability in the subsystem is highlighted in the PRBS section below.

Another atomic is shown in this diagram for the ODUkP/Packet adaptation function (not in G.798). This is in the sink direction. It performs the following features:

- Converts DPI packet traffic (SOP/EOP) into DCI ODUk traffic and measures the ODUk rate used to control the SERDES rate and justification rates (AMP/GMP) into server containers.
- When traffic supplied by the ODUk switch is 'dry' (that is, it cannot feed SERDES or mappers at constant rate) due to a fabric/backplane mapper failure for a particular channel, the channel outputs an aAIS signal. The aAIS signal is used automatically to generate ODUk-AIS.
- After CBR traffic is formed, ODUk frame alignment is performed to find ODUk frames. In-frame (IF) and out-of-frame (OOF) criteria are held as per G.798, but dLOFLOM contributing to aAIS to generate ODUk-AIS streams can be optionally disabled.
- Optionally inserts ODUk-OCI and ODUk-LCK at a nominal rate. Note that OCI insertion position is exactly shown in the RFRM/TFRM atomics.
- The packet traffic measure and justification generation is part of the MPMO block in COREOTN. The frame alignment and maintenance signal generation is part of the RFRM block (stage 3B and stage 4).

The ODUkP/Packet adaptation source converts DCI ODUk traffic into DPI traffic to measure the rate of the traffic and store in ODUk switch buffer. This feature is completely implemented in the subsystem MPMA block.

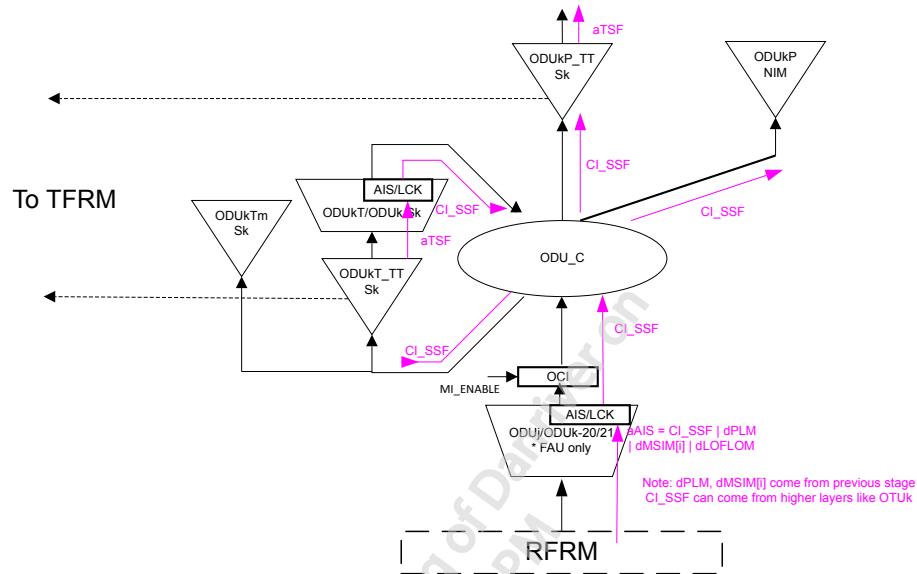
Non-intrusive CSF detection is also provided. This is done after stage 3A processing is complete. This function has been added for SNC/Nc protection switching where the APS engine in the management layer can be notified of a client failure through the OPU-CSF bit prior to going into the demapper interface at the MAPOTN.

The next few sections describe how the above atomics are implemented through the main datapath blocks: ODU\_RFRM, ODU\_TFRM, ODTU\_MUX, ODTU\_DMX.

#### 5.8.4.6.1 ODU Receive Framer (ODU\_RFRM) Atomics

The atomics represented in the ODU\_RFRM are shown in [Figure 35](#).

Figure 35 RFRM Atomic Function



In the above diagram, the black arrows show the datapath and the pink arrows show the consequential actions and AIS/LCK/OCI replacement path. Most of the pink notation is defined by standards although areas where there is only a slight difference are also noted in pink.

During Stage 1 instantiation and Stage N, the ODUk Frame Alignment Unit (FAU) is disabled since the incoming traffic is already framed from the OTU layer. In the other stage framers, the FAU is enabled and contributes to dLOFLOM, which optionally contributes to aAIS (as shown above).

The consequential action aAIS, as shown above, triggers nominal rate ODUk-AIS towards the ODUKT and ODUkP atomic functions along with CI\_SSF towards these functions.

Following an ODUk-AIS maintenance signal insertion, the ODU\_C connects to multiple ODUKT\_TT sinks and ODUKT/ODUK adaptation sink functions.

In the ODU\_RFRM, the ODUKT\_TT sinks and ODUKT/ODUK adaptation function supports:

- Defect detection per G.798 for: dTIM, dOCI, dAIS, dLTC, dLCK, dIAE, dBIAE, dDEG, dBDI and interrupt registers that can be polled on a regular basis with current status and event occurrence 'sticky' bit. Method used to retrieve the defects out of RFRM is via a defect LCLK.
- Performance and output of the following consequential actions per G.798: aAIS, aTSF, aTSD, aBDI, aBEI, aBIAE. With channelized disable of dTIM and dLTC contributing to the appropriate consequential actions. A global disable for each defect contributing to each respective consequential action (that is, dOCI can be disabled globally for all channels to contribute to aAIS). A channelized disable per consequential action is also supported.
- During aAIS, replacement of the stream to 'at-rate' ODUk-AIS. The aTSF adapts to the CI\_SSF to the next layer (TCM or PM) to be processed.
- Transmission of aBDI, aBIAE and aBEI as per standard to the source (ODU\_TFRM)
- Along with the defects and consequential actions, the following accepted values can be read from the ODU\_RFRM: accepted TTI, accepted STAT, accepted APS/PCC levels
- Both NEBC and FEBC as per standard are captured and the counters are sized to saturate at a little over 1 second interval

- dDEG is declared based on 'bursty mode' operation with programmable DEGM thresholds
- Defect seconds for pN\_DS, pF\_DS, pBAE, pIAE are also captured in registers
- Pass delay measurement bit for the ODU\_TFRM to perform/start delay measurement
- Defects are correlated with CI\_SSF and dAIS in hardware by setting the defect to false during CI\_SSF and dAIS as required in G.798. All other defect correlations are performed in software.

**Note:**

pN\_DS is not available for the Stage 3B or Stage 4 RFM blocks when there is no ODUk data being sourced from the ODUk\_SW.

The ODU\_RFRM also contains an additional ODUkTm (TCM NIM - floating) resource that is of the same level of an already processed resource but can be placed in any location in the hierarchy. The NIM functionality supports all the features of the termination except there are no consequential actions performed (only defects and performance counts are reported). In total there are seven resources, six of which can be configured as NIM/Operational/Monitored/Transparent and one additional resource of same level that is in the NIM.

After the TCM has been processed, the ODU\_RFRM can optionally terminate the ODUkP\_TT (PM layer). The features of this atomic resource are as follows:

- Defect detection per G.798 for: dTIM, dOCI, dAIS, dLCK, dDEG, dBdi and interrupt registers that can be polled on a regular basis with current status and event occurrence 'sticky' bit. Method used to retrieve the defects out of RFRM is via a defect LCLK.
- Performance and output of the following consequential actions per G.798: aTSF, aTSD, aBDI, aBEI. With channelized disable of dTIM contributing to the appropriate consequential actions. A global disable for each defect contributing to each respective consequential action (i.e. dOCI can be disabled globally for all channels to contribute to aTSF). A channelized disable per consequential action is also supported.
- Transmission of aBDI, aBAE and aBEI as per standard to the source (ODU\_TFRM)
- Along with the defects and consequential actions, the following accepted values can be read from the ODU\_RFRM: accepted TTI, accepted STAT, accepted APS/PCC levels
- Both NEBC and FEBC as per standard is captured and the counters are sized to saturate at a little over 1 second interval.
- dDEG is declared based on 'bursty mode' operation with programmable DEGM thresholds.
- Defect seconds for pN\_DS, pF\_DS are also captured in registers
- Pass Delay measurement bit for the ODU\_TFRM to perform/start delay measurement.
- Defects are correlated with CI\_SSF and dAIS in hardware by setting the defect to false during CI\_SSF and dAIS, as required in G.798. All other defect correlations are performed in software.

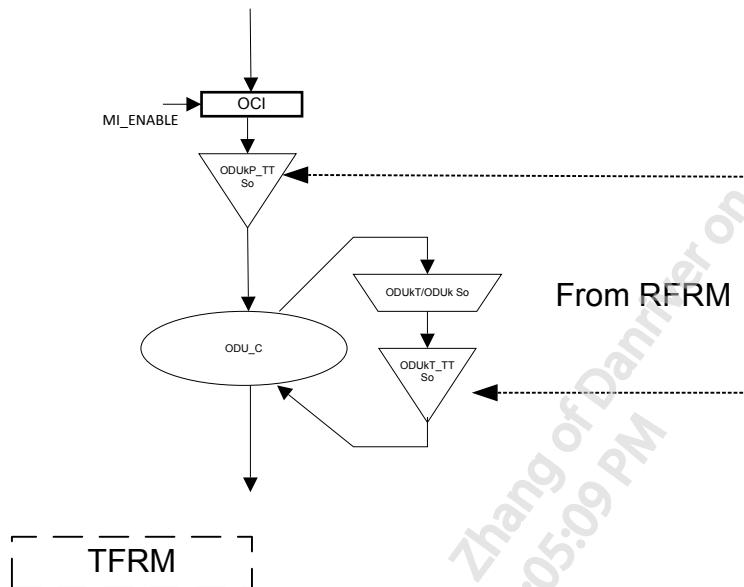
Along with a PM ODUkP\_TT resource, there is an additional ODUkPm (NIM - floating) resource that can be placed anywhere between the various TCM and PM sinks. The NIM is identical to the termination resource except it does not perform any consequential actions, but only reports defects and performance monitoring counts.

The ODUkT\_TT, ODUkTm and ODUkPm resources (up to eight per channel) are picked up from shared pool resource among all channels. Stage 1 and Stage N framers have 72 'pool' resources that are shared among all active channels. Stage 2, 3A, 3B, 4 framers have 256 pool resources that are shared among all active channels. The ODUkP\_TT function is a dedicated resource so hence there is one resource per channel, which can either be in termination or non-intrusive (NIM) mode of operation.

#### 5.8.4.6.2 ODU Transmitter Framer (ODU\_TFRM) Atomics

The atomics represented in the ODU\_TFRM are as follows.

Figure 36 TFRM Atomic Function



As seen from the above diagram, the maintenance signals are forced through the TFRM. The OCI/AIS/LCK can all be forced.

These are the following features of the ODUkP\_TT So layer:

- Inserts a PM-STAT,TTI, APS/PCC and BIP8 calculation
- Inserts BDI/BEI from ODUkP\_TT sink from the RFRM atomics
- Performs delay measurement testing as described in the section below

These are the following features of the ODUkT\_TT So layer and ODUkT/ODUk adaptation source functions:

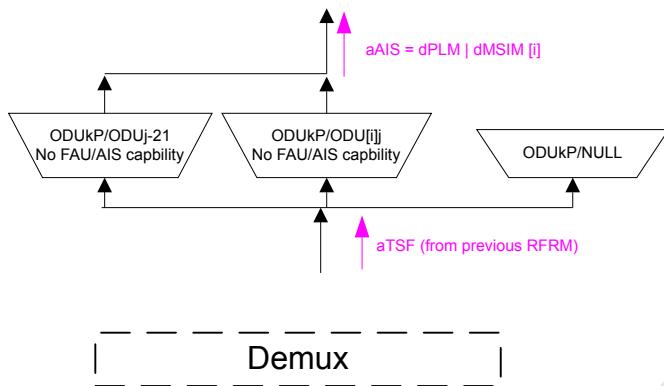
- Ability to force LCK/OCI/AIS at the adaptation source
- Inserts TCMi- STAT, TTI, APS/PCC and BIP8 calculations
- Detects and inserts STAT-IAE if the incoming frame alignment is mismatched. (This is after a maintenance signal insertion.) The IAE Stat is inserted in TFRM when the SOF pulse is off from its expected location, which is determined by a fly-wheeling frame pulse signal captured in the TFRM based on previous frame pulses.
- Inserts BDI/BEI from the ODUkT\_TT Sink from the RFRM atomics
- Performs delay measurement test as per section below

The TCMs are assigned from a pool that is shared among all channels for a particular framer stage. Stage 1 and Stage N have 72 pool elements. Stages 2, 3A, 3B, and 4 have 256 pool elements.

#### 5.8.4.6.3 HO ODU to LO ODU De-multiplexer (ODTU\_DMX) Atomics

The following atomics represent the functionality of the ODTU\_DMX.

Figure 37 Demux Atomic Function



The ODTU\_DMX supports the following features with respect to the above atomics:

- De-multiplexing protocols defined as per mux/demux options section below
- ODUkP/ODUj-P20 adaptation function that de-multiplexes AMP 1.25 G and 2.5 G tributary slots
- ODUkP/ODUj-P21 adaptation function that de-multiplexes a mixture of GMP 1.25 G and AMP 1.25 G tributary slots
- G.7044 receiver protocol: support for intermediate/sink node negotiation detection (RP, TSCC, CTRL words and RCOH changes)
- ODUkP/Null adaptation sink function that terminates NULL channels at the DEMUX block
- Performs and interrupts on dMSIM, dPLM and dLOOFMI defects
- Optionally contributes dMSIM[j], dLOOFMI, dPLM to the appropriate LO SSF to conduct aAIS in neighboring RFRM block
- Stores accepted MSI and PT values on the device to read
- Ability to add/remove and reconfigure LO clients into HO without affecting HO frame structure or other active LO's.

The ODTU\_DMX does not support defect correlations in hardware.

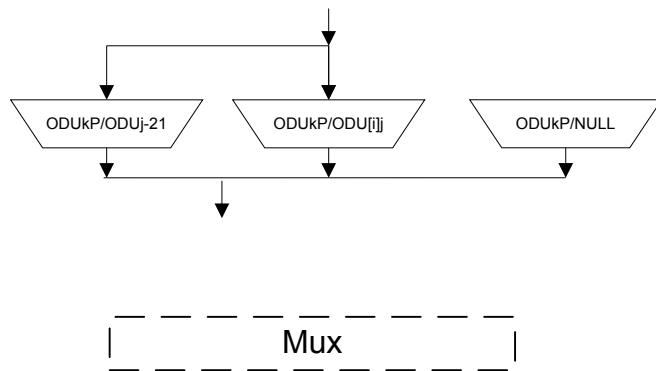
**Note:** The defect status indication for a Multiplex Structure Identifier Mismatch (dMSIM) is invalid following an ODUk channel re-provisioning until the traffic recovers and a new MSI has been accepted. dMSIM can be falsely asserted when the new ODUk channel is provisioned for up to 256 frames while the new MSI indication is being accepted by the MSIM defect detect logic.

Software must ignore the dMSIM indication for 256 frames following an ODUk channel re-provision. After the new MSI is accepted for the new ODUk the dMSIM indication will be valid.

#### 5.8.4.6.4 HO ODU to LO ODU Multiplexer (ODTU\_MUX) Atomics

The following atomics represent the functionality of the ODTU\_MUX.

Figure 38 Mux Atomic Function



The ODTU\_MUX supports the following features with respect to the atomics:

- Multiplexing protocols defined as per [Section 5.8.4.6.5](#)
- ODUkP/ODUj-P20 adaptation function that multiplexes AMP 1.25G and 2.5G tributary slots
- ODUkP/ODUj-P21 adaptation function that multiplexes a mixture of GMP 1.25G and AMP 1.25 G tributary slots
- G.7044 transmitter protocol: support for intermediate/source node negotiation (RP, TSCC, CTRL words and RCOH additions)
- ODUkP/NULL adaptation source function that generates NULL payload for specific channels
- Source PT, MSI and OMFI bytes into the ODUk container along with a new FAS and MFAS bytes for active channels.
- Receives Cn (bytes by multi-frame from the rate engine, MPMO: Packet Monitor) to do AMP/GMP justification and payload/stuff byte calculations
- Ability to add/remove and reconfigure LO clients into HO without affecting HO frame structure or other active LO's. The ability is restricted to reconfiguration of LO clients for a HO container with either 1.25G or 2.5G tributary slots. Changing a HO container from 1.25G to 2.5G tributary slots or vice-versa will result in HO frame slips during the transition. For example, a HO ODUK initially configured for 1.25G timeslots and a TCM terminated on the HO will require the TCM to be de-provisioned and then reprovisioned when the HO is reconfigured for 2.5G timeslots.

#### 5.8.4.6.5 Mux/Demux Options

The ODUkP/ODU[i][j] adaptation function for GMP or AMP of G.798 defines the requirements for muxing or demuxing LO to HO and HO to LO, respectively.

The different combinations of HO and LO supported in the COREOTN are shown in [Figure 39](#).

Figure 39 COREOTN Muxing/Demuxing Support

ODTUjk Mixing Options											
HO	ODU0	ODU1	ODU2	OTU1e	OTU2e	OTU1f	OTU2f	OTU3	OTU3e1	OTU3e2	OTU4
LO											
<i>ODU0</i>	N/A	AMP, 1.25G TS	GMP, 1.25G TS	N/A	N/A	N/A	N/A	GMP, 1.25G TS	NA	GMP, 1.25G TS	GMP, 1.25G TS
<i>ODU1</i>	N/A	N/A	AMP 1.25G TS, AMP 2.5G TS	N/A	N/A	N/A	N/A	AMP 1.25G TS, AMP 2.5G TS	NA	GMP, 1.25G TS	GMP, 1.25G TS
<i>ODU2</i>	N/A	N/A	Output	N/A	N/A	N/A	N/A	AMP 1.25G TS, AMP 2.5G TS	NA	GMP, 1.25G TS	GMP, 1.25G TS
<i>ODU1e</i>	N/A	N/A	N/A	Output	N/A	N/A	N/A	GMP, 1.25G TS	NA	NA	NA
<i>ODU2e</i>	N/A	N/A	N/A	N/A	Output	N/A	N/A	GMP, 1.25G TS	AMP, 2.5G TS	GMP, 1.25G TS	GMP, 1.25G TS
<i>ODU1f</i>	N/A	N/A	N/A	N/A	N/A	Output	N/A	GMP, 1.25G TS	NA	GMP, 1.25G TS	GMP, 1.25G TS
<i>ODU2f</i>	N/A	N/A	N/A	N/A	N/A	N/A	Output	GMP, 1.25G TS	NA	GMP, 1.25G TS	GMP, 1.25G TS
<i>ODU3</i>	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Output	N/A	GMP, 1.25G TS	GMP, 1.25G TS
<i>ODU3e1</i>	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Output	N/A	GMP, 1.25G TS
<i>ODU3e2</i>	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Output	GMP, 1.25G TS
<i>ODUflex(CBR)</i>	N/A	N/A	GMP, 1.25G TS	NA	NA	NA	NA	GMP, 1.25G TS	NA	GMP, 1.25G TS	GMP, 1.25G TS
<i>ODUflex(GFP)</i>	GMP, 1.25G TS	GMP, 1.25G TS	GMP, 1.25G TS	NA	NA	NA	NA	GMP, 1.25G TS	NA	GMP, 1.25G TS	GMP, 1.25G TS

#### 5.8.4.6.6 CSF Detection

Non-intrusive CSF detection is added between the Stage 3A and 3B framers in the ingress direction from the LINEOTN. This is used as criteria to do SNC/Nc protection. This dCSF indication can be associated with framer Stage 1 when the channel does not perform de-multiplexing, framer Stage 2 when there is a single stage de-multiplexing, or framer Stage 3A when there is dual stage de-multiplexing.

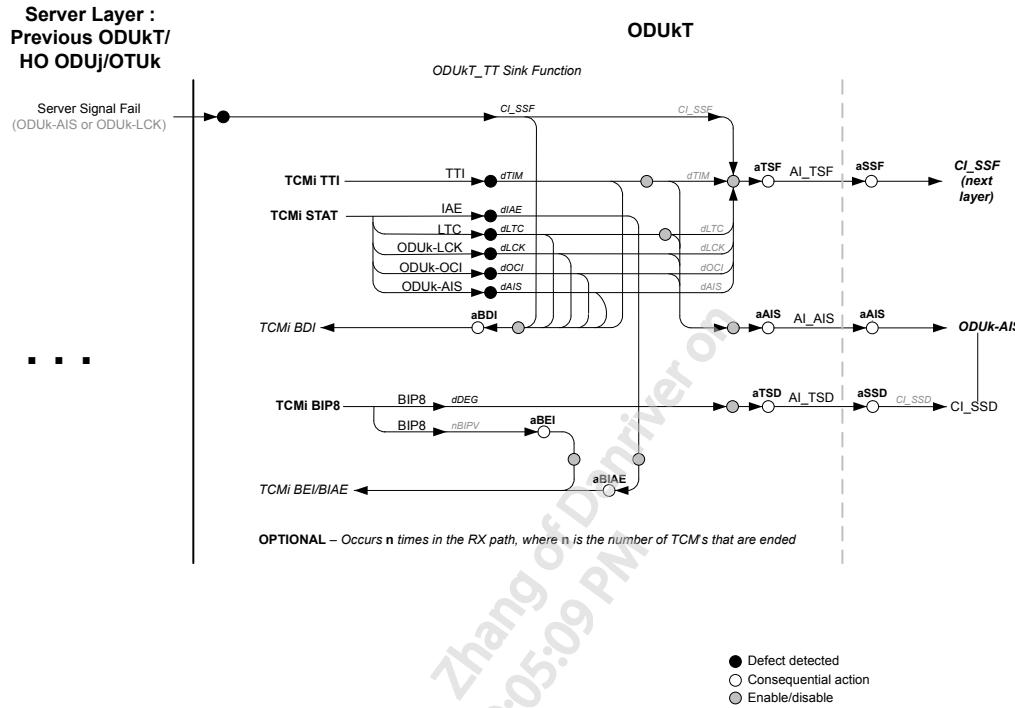
### 5.8.4.7 Defect Detection and Consequential Actions Support Across Multiple Channels and Stages

Defect detection and consequential actions are enabled in all stages of framers. The following diagrams summarize the supported defect detection and consequential actions in each framer stage with the generation of ODUk-AIS streams in stages 1, 2, 3a, and 4 disabled.

**Figure 40** highlights the ODUKT function. This function is implemented in each RFRM block. The RFRM has a pool of 256 TCMs that can be either terminated with the ODUKT\_TT function (as below) or non-intrusively monitored without stream replacement indications. Each defect below is indicated to software to read and can be disabled leading to the respective consequential actions as shown in the diagram. The output of this function is either ODUk-AIS or the clean ODUk stream. All the TCMs are prioritized through register control, such that nested TCM configuration in the networks can be realized.

The software role is to provision the defects that contribute to the various consequential actions. By default it is setup as per G.798 boolean functions.

Figure 40 ODUkT\_TT Sink Function Implementation



The function below highlights the ODUkP\_TT function as implemented in the ODU\_RFRM block and the ODU[i][j] adaptation function as implemented in the ODTU\_DM block.

Figure 41 ODUkP\_TT Sink and ODU[i][j] AP Function Implementation

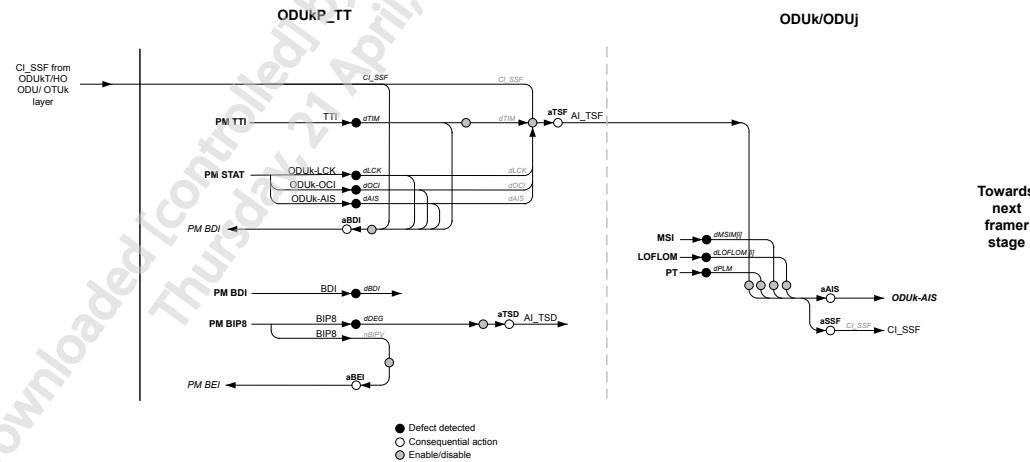


Figure 40 and Figure 41 illustrate the various defects and consequential actions gathered from G.798 atomic functions for each 'order' of the OTN container. The ODU\_RFRM is responsible for the ODUkT and ODUkP sink functions and also the adaptation of an ODUkP to an ODUk container. The ODUkP to ODUj adaptation is handled by the ODTU\_MUX/ODTU\_DM block, this includes dPLM, dMSIM, dLOFLOM detection for the various OTN containers. dLOFLOM consequently is handled by the ODU\_RFRM after the demux stage since the frame alignment unit exists within the RFRM unit.

#### 5.8.4.7.1 ODUk-AIS Maintenance Signal Auto-replacement

The auto maintenance signal (ODUk-AIS) is replaced only at the adaptation functions implemented within the RFRM. This is the ODUkP/ODUj adaptation sink (HO to LO, only FAU in the ODU\_RFRM) and various ODUkT/ODUk sink (TCM adaptations).

The ODUkP/ODUj maintenance signal insertion is at nominal rate (that is, 0 ppm). The contributors to this maintenance signal insertion are dPLM, dMSIM, dLOFLOM, aTSF from the ODUkP\_TT layer. In the case of the ODUk/Pck-AP sink function, this is due to fabric failure.

The ODUkT/ODUk maintenance signal insertion is at line rate. The contributors to this maintenance signal insertion is aTSF from processed TCM layer.

ODUk-AIS structure is similar to G.709 definition with FTFL field passed through and the OTUk overhead preserved (where applicable). The RFRM can then extract the FTFL field (10 bytes backward and 10 bytes forward from registers) and the TFRM can insert the FTFL field (10 bytes backward and 10 bytes forward from registers), with backward and forward insertion/extraction control.

After the maintenance signal is auto inserted, CI\_SSF is still passed to the next layer, as per G.798.

#### 5.8.4.7.2 ODUk-LCK/ODUk-AIS/ODUk-OCI Force Signal Replacement

Each RFRM/TFRM has the capability of forcing ODUk-LCK/AIS/OCI at the entrance of the block at nominal rate. These follow the same structure as defined by the G.709 standard.

Note that when ODUk-LCK, ODUk-AIS or ODUk-OCI are forced, the transmit ODUk rate can revert to nominal timing. This will result in ODUk\_SW buffer overflows or underflows for transparent ODUks due to the difference between the incoming ODUk rate and the outgoing ODUk-OCI/AIS/LCK rate. The resulting ODUk\_SW buffer overflows or underflows will cause the Stage 3B FRM block to report LOFLOM continuously. Therefore, these indications should be ignored when the forced ODUk-OCI/AIS/LCK are being transmitted at the nominal ODUk rate.

In addition to forcing at the entrance, the ODU\_RFRM/ODU\_TFRM can also force ODUk-LCK/AIS at each TCM adaptation function.

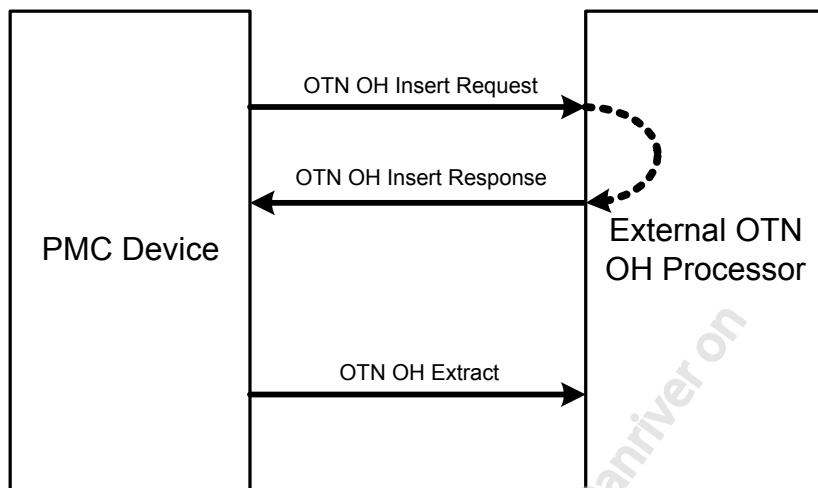
#### 5.8.4.8 Overhead Insertion/Extraction Feature

The overhead insertion/extraction for the OTU/ODU/OPU layers are done using three different methods:

- Passed down by the predecessor
- Inserted through the default registers (general for all ODU frames within the channel)
- Inserted using the overhead insert/extract block

There are distinct framer stages. Two stages are grouped together to share an SGMII to communicate the overhead to an external processor. [Figure 42](#) shows OTN overhead insertion/extraction through a single overhead insertion/extraction port.

Figure 42 OTN Overhead Insert and Extraction



Per overhead insertion/extraction pair, there are two physical ports:

- An insert overhead port from an external OTN overhead processor to the device (receive to the device)
- An extract overhead port from the device to the OTN overhead processor (transmit from the device)

In the transmit direction, there is also an OTN overhead insert request sent to indicate that the OTN overhead processor can do an OTN overhead insert (receive to the device). The frame format of the receive and transmit frames on this interface is described in the OHP block in the following sections. Essentially the external OTN overhead must be the slave to the transmit frames while doing an insertion in the receive direction. The extraction supports extraction of OTUK, ODUK, and OPUK bytes. These overhead bytes extracted are raw bytes as seen from the framed OTN data.

Since this interface is framed Ethernet in nature, the SA and DA are used to carry a tag from the frame number, channel ID and user specific ID field the overhead payload belongs to.

Stage 1 and Stage N framers are hooked up to OHP1. Stages 2, 3A, 3B and 4 can be muxed into either OHP2 or OHP3 ports.

#### 5.8.4.9 Sub Network Connection Protection (SNCP) Support

There are several schemes for SNCP (Sub Network Connection Protection). The implementations of these schemes are covered in section 14.1 of the G.798 specification in the ODU\_C function section. In the COREOTN, several connection points described in the ODU\_C connection function (of G.798) are limited through the ODUk switch, which can interconnect any channel from any of the three interfaces (line interface, client interface and backplane interface) in any combination. The COREOTN also supports multi-casting and dynamic channel selection of the various interfaces, which can be configured through software (serves as a bridge and select within the chip). Configuring the protection scheme is done by management software external to the device.

The forms of SNCP schemes in a network include:

- SNC/N
- SNC/I
- SNC/S or SNC/Nc

- In some networks, a combined version of the above

In the COREOTN, these protection schemes are enabled through the ODUk switch, which allows working and protect lines to be created by reconfiguring the output channels or allowing ODUk traffic to be connected to the customer ODUk fabric through a backplane interface.

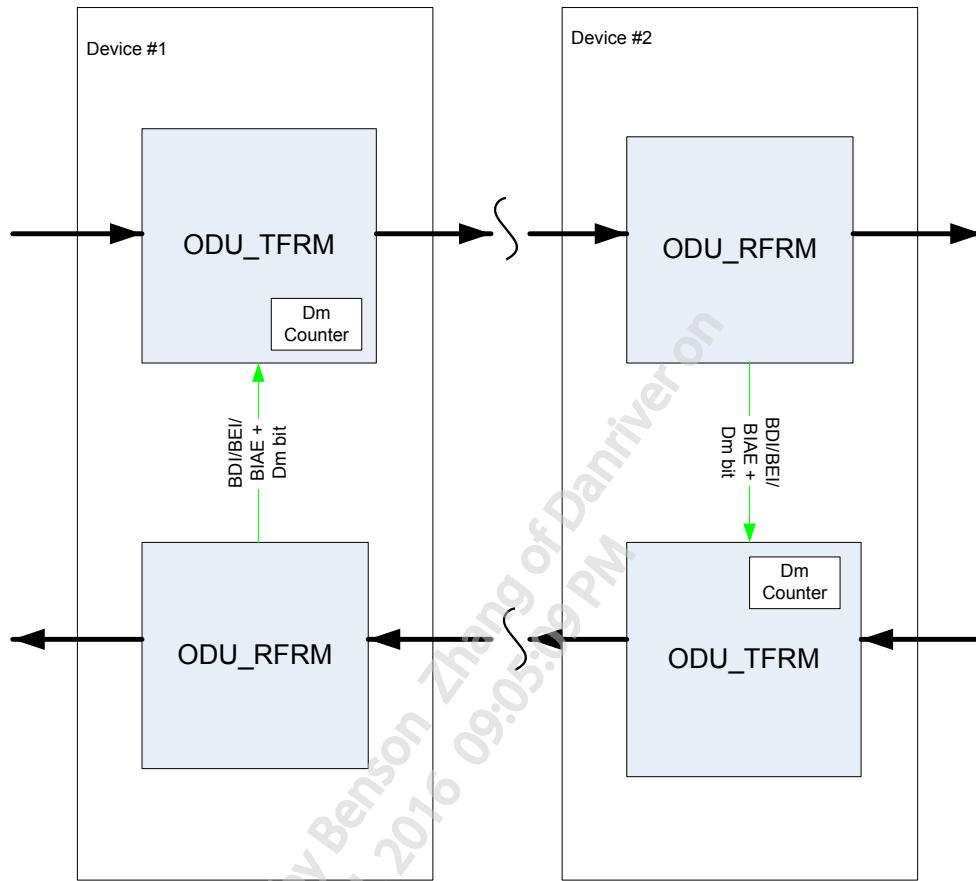
The management layer can optionally insert/extract APS bytes or monitor the BDI/BEI/BIP-8 functions through any of the four stages of overhead insertion and extraction on a frame-by-frame basis. The hardware does not perform direct APS reconfiguration of channels based on incoming streams without a management layer indication. If reconfiguration of streams occurs (as in a 1:1 or 1:n cases), this operation can only occur at the ODUk switch. Hence, HO or MO ODUk containers cannot perform protection switching operations within the COREOTN subsystem.

The COREOTN also provides all the defects and SF/SD conditions in each layer processed in the ODU\_RFRM and ODTU\_DMX to assist software with making a switch over via the external switch fabric or the ODUk switch located in the device.

#### 5.8.4.10 Delay Test Support in the ODUk Layers

As per the G.709 standard, there is a delay test feature in the ODUk layers that loops back ODUk layers with a delay encoding mechanism for operators to determine the ODUk layer latency across the network system. The COREOTN subsystem supports the delay test feature across all ODU stages and all active TCM layers and PM layers through signaling between the Rx and associated Tx framers. The signaling mechanism defined in the G.709 standard is by default, implemented by hardware unless disabled by software. The delay measurement count exists within the ODU\_TFRM and the ODU\_RFRM relays the DMx bit to the ODU\_TFRM as illustrated in [Figure 43](#).

Figure 43 Delay Measurement Diagram



The TFRM delay measurement count is 18 bits in precision for all ODUks (covers up to 200 ms in distance). From a TFRM perspective, a delay measurement may be performed between two TCM or PM end points by controlling and observing the DMtx and DMp bits in the PM&TCM overhead byte. When a TCMx is transparent or ODUkP overhead insertion is disabled, the corresponding bit (DMtx or DMp) in the PM&TCM byte is passed transparently. When a TCMx is operational or ODUkP overhead insertion is enabled, the corresponding bit (DMtx or DMp) in the PM&TCM byte is inserted based on whether the delay measurement state machine is provisioned as a source:

- If the PM DM state machine is a source, then the DMP\_VAL bit in the CHANNEL\_CONTROL register is inserted in the DMp bit position.
- If the TCM DM state machine is a source, then the DMT\_VAL bit in the TCM\_CONTROL register is inserted in the corresponding DMtx bit.
- If the DM state machine is not a source, then the backwards information contained in the RFRM communication link is inserted in the corresponding DMtx/DMp bit position.

#### 5.8.4.11 OPSA Features

The COREOTN implements an innovative OPSA that enables the COREOTN to act as an O-E-O regenerator. The OPSA consists of two parts:

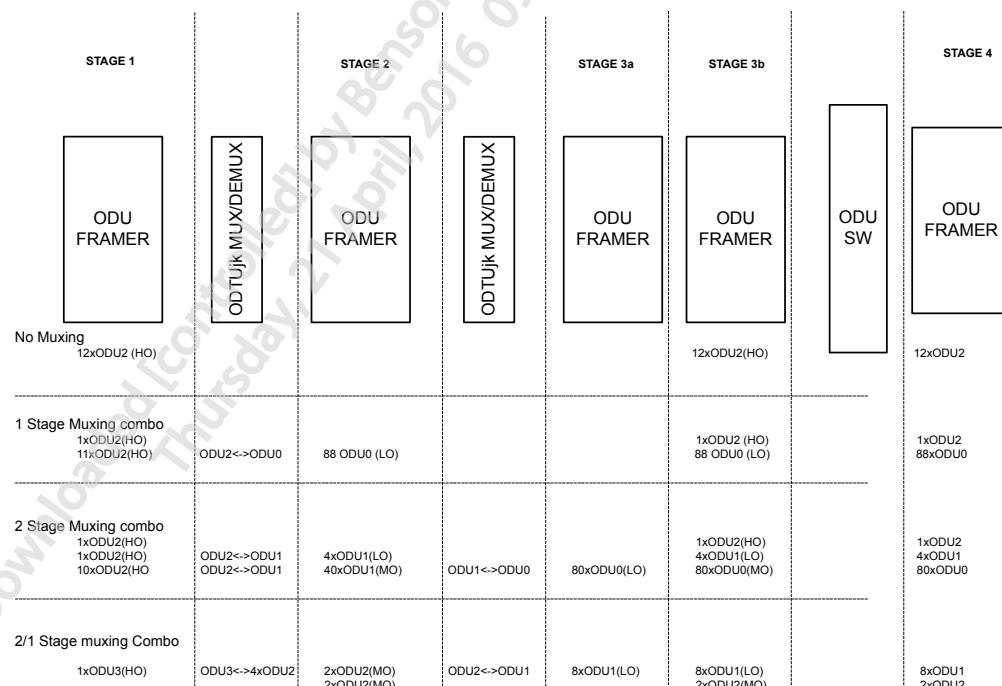
- On the receive side, the OPSA continuously monitors the rate of the incoming ODUk stream. In the case where an ODUj stream is the result of an ODTUjk de-multiplex operation, the rate of the ODUj stream is derived from the rate of the parent ODUk stream and the justification controls in the OPUk overhead. Depending on the rate of the incoming stream relative to the system reference clock, the COREOTN generates artificial packets of B-1, B, or B+1 byte sizes into the ODUk switch block.
- On the transmit side, the COREOTN uses packet sizes to control a PLL to generate the transmit clock or to control Justification Control in an ODTUjk multiplexer. Because the PLL control is only a function of packet sizes and not packet arrival times, the wander of the outgoing OTN stream is independent of the packet delay variations in the OTN packet fabric.

The OPSA is also used when de-mapping CBR clients from an OTN stream. The rate of the CBR client is the sum of the rate of the bearer OTN stream plus the relative rate encoded in the Justification Control signals. OPSA is used to extract the rate of the bearer OTN stream.

#### 5.8.4.12 Left-justified Data Processing

The framers in the COREOTN are configured and built to operate in 'Left Justified Data Processing' style. This style of data processing moves all the data processing to the left most framers. Hence all HO ODUK frames always operate in Framer Stage 1. If a specific channel has 1-stage muxing, the LO lands in Stage 2 and if there is 2-stage muxing, the MO lands in framer Stage 2 and the LO lands in framer Stage 3A. [Figure 44](#) illustrates the different configurations.

[Figure 44](#) Left Justified Data Processing



Stage 4 framers optionally receive the same channels as Stage 3b, but are not required to do any processing.

#### 5.8.4.13 PRBS Capability in the COREOTN

The COREOTN can insert PRBS-31 and PRBS-11 (plus inverted) sequences at the following locations in the ODU payload (monitors are in adjacent locations):

- After the MAPOTN interface before Stage 4 TFRM towards the ODUk switch (LO PRBS and towards Fabric PRBS Capability)
- After Mux 2 and before Stage 2 TFRM towards the LINE\_OTN (MO PRBS capability)
- After Mux 1 and before Stage 1 TFRM towards the LINE\_OTN (HO PRBS capability)

Each PRBS instantiation (monitor/generation) has up to 12 channels of capability that can be inserted/monitored simultaneously. The monitor can detect dLSS (Loss of Sequence Synchronization) and contains nTSE (Test Loss of Sequence) counts that saturates at 16 bits per channel.

**Note:** The PRBS monitor's Loss of Synchronization Status defect interrupt and status indication bits, PRBS\_DLSS\_I and PRBS\_DLSS\_V, respectively, are only updated by the device latch clock (LCLK). LCLK is the device wide trigger used to update PMON counters within the device and is only updated once per second. There will be a delay of up to one second between updates to the status of the dLSS, which means higher layer software will only be able to report dLSS status on per second boundaries.

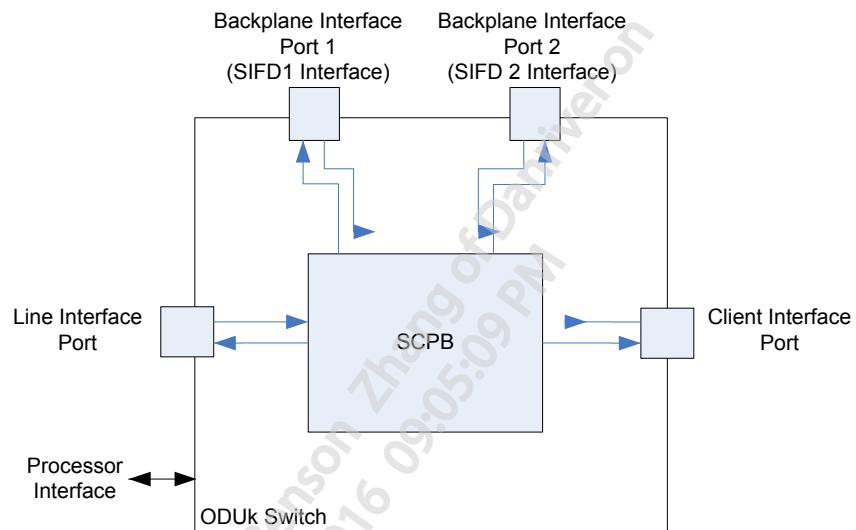
Downloaded [controlled] by Benson Zhang of L3 Communications  
Thursday, 21 April, 2016 09:05:09 PM

## 5.8.5 ODUk Switch

The ODUk switch is a central repository for buffering ODUk packets (OTN frames in the form of OPSA packets) between source clients and destination clients. It is a store and forward, non-blocking, FIFO based, packet switch.

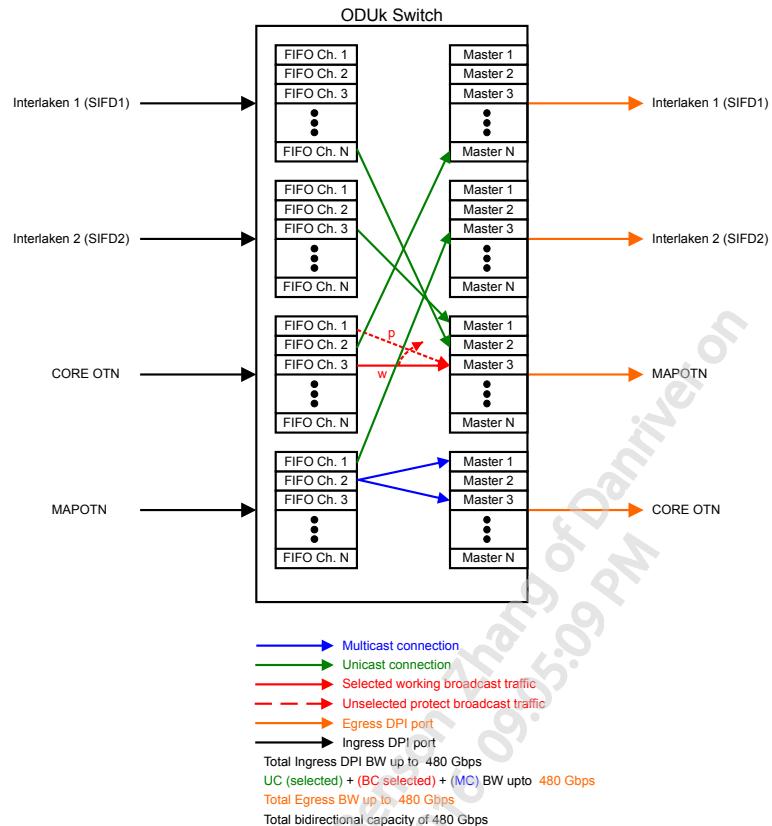
Clients communicate to the ODUk switch through a channelized (up to 96 channels) DPI, each client having a dedicated DPI port. [Figure 45](#) illustrates the four bidirectional ports of the ODUk Switch and shows how they are connected.

[Figure 45 ODUk Switch Functional Diagram](#)



[Figure 46](#) illustrates the switch capacity.

Figure 46 ODUk Switch Capacity

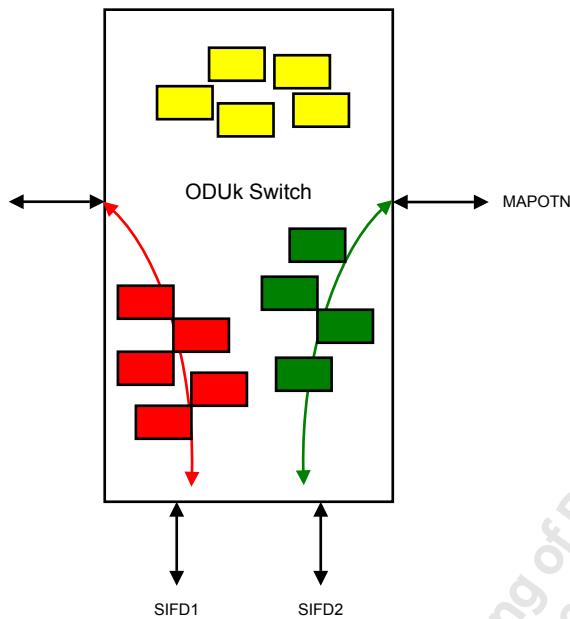


The ODUk switch supports selection of up to 480 Gbps of traffic by the egress ports for a total bidirectional capacity of 480 Gbps.

Ingress ports are slave DPI ports and egress ports in the switch are master DPI ports. The ODUk switch maps ingress channels per port to internal queues and independently maps internal queues to egress channels and ports. The total capacity of all the internal queues is 1 MB of data.

The ODUk switch uses the same architecture as the CPB but has fewer ports and less memory for data queues. The ODUk switch contains 22 pages with 64 blocks per page and a block size of 0.75 KB. Each page is then 64 blocks of 768 bytes each for a page size of 48 KB. The 22 pages collectively provide 1056 KB of queue storage. [Figure 47](#) illustrates the allocation of pages to port pairs.

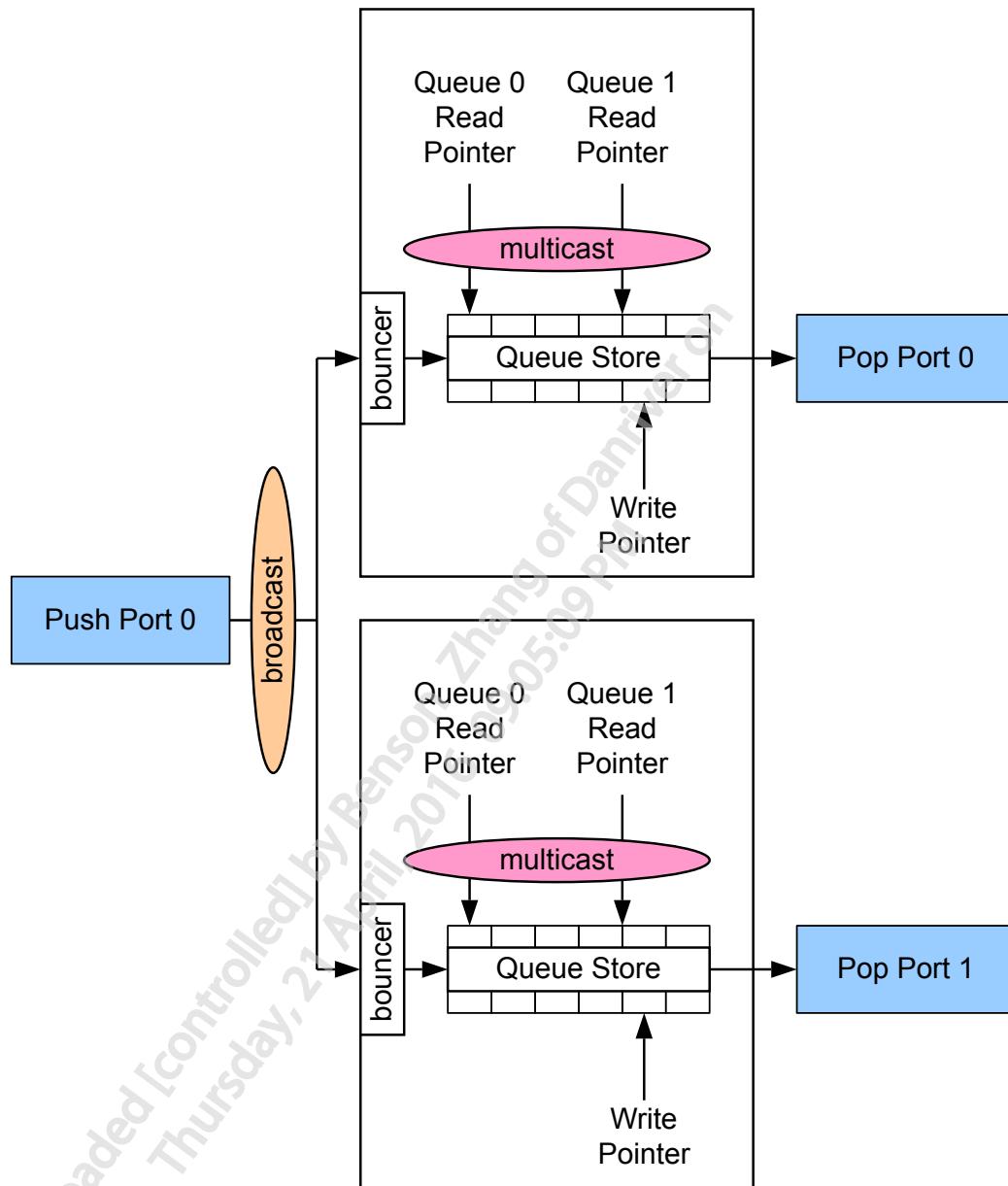
Figure 47 Page Allocation to Port Pairs



A queue is composed of a link list of blocks from pages associated with a port pair. Each page needs to be assigned to an ingress to egress port pair. A page cannot be shared between port-pairs. A queue is created by specifying a link list of page blocks from the pool of pages assigned to an ingress-egress port pair. A queue can consist of blocks from different pages.

The ODUk switch supports both multicast and broadcast. Multicast is used to duplicate traffic to egress channels within the same egress port. For example, multicast is used when bridging an ODUk to two ODUk channels in the COREOTN subsystem. Multicast results in bandwidth expansion since multiple readers are associated with a single writer. Broadcast is used to duplicate traffic to different ports. For example, broadcast can be used to send an ODUk to both Interlaken interfaces. Broadcast is restricted to 2-way broadcast. Note broadcast causes storage expansion since data is duplicated to multiple queues, which needs to be considered when budgeting the available ODUk switch memory.

Figure 48 Broadcast and Multicast



Section 5.8.5.1 describes the queue zone reporting mechanism for flow control purposes when using broadcast or multicast.

For the ODUk switch, multicast of traffic destined for the same Interlaken interface is not supported. Protection is expected to occur in a fabric external to the device. Multicast to the same Interlaken interface is not expected to be a required system application of the device. Multicast traffic destined to an interface other than Interlaken is supported for the ODUk switch. A maximum of 1 to 2 multicast cast is supported. The architecture supports N:1 selection. Note 1:2 multicast requires two times the bandwidth of the traffic source through the ODUk Switch due to the duplication of queue readers.

The ODUK switch supports broadcast to channels in different switch ports, including the two different Interlaken interfaces.

The ODUK switch performs hole insertion for populating the 4-byte packet maker monitor header and optional 2/4/6 byte fields inserted by the SIFD block. See [Figure 110](#) for the more information on the header format.

### 5.8.5.1 Zone Interface and Flow Control

The ODUK switch allows destination clients to back pressure the scheduler driving the master DPI port. Destination clients can issue link XOFF or channel XOFF.

The ODUK switch can be configured to enter the channel XOFF state, as signaled by the destination client, on a segment, burst or packet boundary. Once a channel has been requested to enter the XOFF state and after waiting for the appropriate transmission boundary, the ODUK switch will associate the channel to an internal queue and the scheduler will cease to request data from that queue.

The ODUK switch can be configured to enter the link XOFF state, as signaled by the destination client, on a segment, burst or packet boundary. Once a link has been requested to enter the XOFF state and after waiting for the appropriate transmission boundary, the ODUK switch will instruct the scheduler not to schedule any queues associated to the link. A link can contain all the channels of a master DPI port, or a grouping of channels.

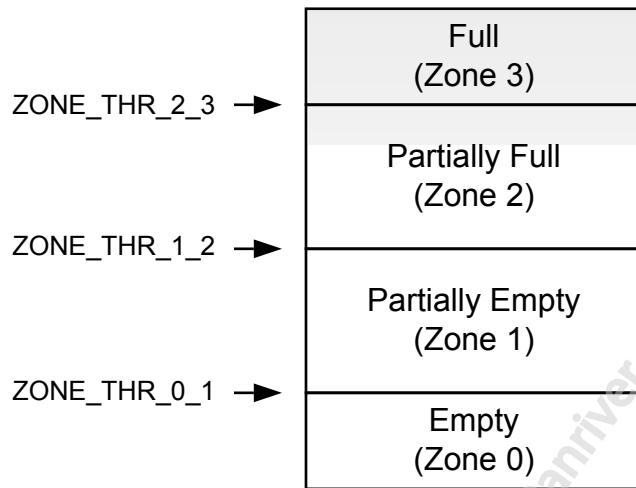
The zone interface is used to transmit the relative queue fill level when compared against a programmable threshold. Each queue has three programmable thresholds (48-byte granularity) which creates four queue zones: Full, Partially Full, Partially Empty, Empty. The ODUK switch provides a zone interface for each pop interface and a zone interface for each push interface. Zone signaling includes a zone 'level' and a Zone ID. The Zone ID represents the queue ID for which the 'level' applies to. For a pop zone interface, the ODUK switch will send out a Zone update associated to a queue whenever the queue being popped is crossing into a new zone, else the ODUK switch will round robin through each of the enabled queues and transmit the zone for each. Similarly for a push zone interface, the ODUK switch will send out a zone update associated to a queue whenever the queue being pushed is crossing into a new zone, else the ODUK switch will round robin through each of the enabled queues and transmit the zone for each.

When performing a broadcast push operation, multiple queues can simultaneously be crossing into a new zone. In this scenario a primary queue is identified through register interface configuration. Only the zone of the primary queue is updated on the push zone interface, all associated broadcast queue zones are ignored.

The zones supported are: Zone 0 "Empty", Zone 1 "Partially Empty", Zone 2 "Partially Full", Zone 3 "Full".

- A queue is in Zone 0 when the queue fill level contains `ZONE_THR_0_1` or fewer entries
- A queue is in Zone 1 when the queue fill level contains (`ZONE_THR_0_1 + 1`) to `ZONE_THR_1_2` entries
- A queue is in Zone 2 when the queue fill level contains (`ZONE_THR_1_2 + 1`) to `ZONE_THR_2_3` entries
- A queue is in Zone 3 when the queue fill level contains (`ZONE_THR_2_3 + 1`) or greater entries

Figure 49 FIFO Zone Indication



The zone thresholds are checked from highest to lowest. For example if the fill level exceeds ZONE\_THR\_2\_3, the queue is considered to be in "Zone 3" regardless of the ZONE\_THR\_1\_2 value being less, equal or greater than ZONE\_THR\_2\_3.

#### 5.8.5.2 DPI Type Usage Through the Switch

The ODUK Switch supports the transmission of a sideband "TYPE" indication that accompanies each packet transiting the switch. The ODUKSW does not interpret the content of the DPI TYPE field and is therefore agnostic to its usage. At a device level, the DPI TYPE bus is in the ODUk switch to convey signal failure (SF) conditions as well as G.7044 resizing ramp indications (BWR\_IND).

Table 39 DPI Type Usage in ODUK Switch

Source	Destination	DPI_TYPE[Field]	Usage
COREOTN	MAPOTN	[1]	BWR_IND
COREOTN	MAPOTN	[0]	SF (Signal Fail)

Note: All other bits of DPI\_TYPE[15:0] not defined in the above table are reserved for future use.

## 5.8.6 ODUk Scheduler (ODUkSC)

The ODUkSC is reused block is reused from the MAPOTN subsystem. Refer to [Section 5.9](#) for a more detailed feature specification.

The COREOTN has two instances of the ODUkSC. Both maintain a calendar as provisioned on the DCI bus to schedule nominal rate ODUk on the provisioned calendar as required.

The COREOTN block may generate AIS as a consequential action of SSF. Maintenance signals (LCK/OCI) may also be generated. The maintenance and AIS signals need to be generated at the nominal rate of the ODUk. The ODUk scheduler is used to define the nominal rate.

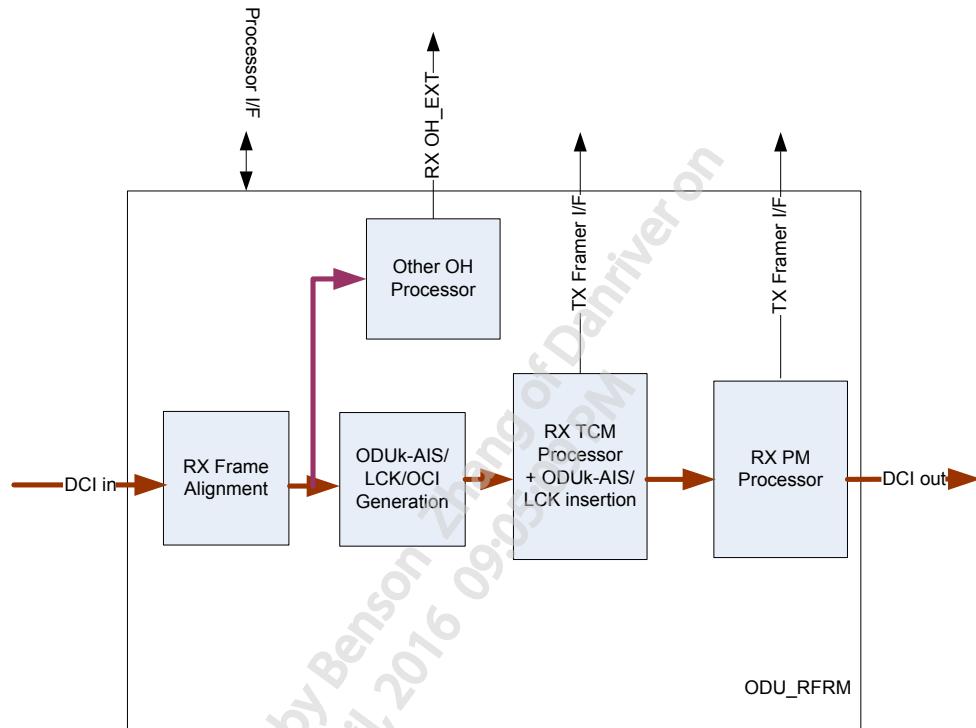
To support the different traffic rates for different type of ODUk containers, a credit generation mechanism is implemented to decide the scheduling. Data is transferred in a particular time-slot only if the time slot has a credit associated with it. The credit generator uses modulo M/N counter that can effectively generate any  $M/N * 1.39968$  Gbps rate, where M and N are integer values programmed through software. Different ODUk containers have a different number of entries in the calendar and each entry has software-programmed M and N values for the container. Each time the credit generator is triggered, with 349.92 MHz clock, it generates M/N credits. Inside the credit generator there is an accumulator, which is incremented by M at every clock. If the resulting sum is  $\geq N$ , the accumulator wraps around to the value of  $Sum - N$  and generates a credit. This way a total M credits are generated in N number of clocks, generating an M/N rate.

For example, the ODU0 container has a single entry in the calendar, but data for this entry is not transferred each time. Software programs M=8 and N=9 to provide a total of eight or nine credits per entry. Hence, out of nine times, the channel is scheduled only eight times. This provide the exact ODU0 rate ( $8/9 * 1.39968$  Gbps = 1.24416 Gbps).

Similarly in case of ODU1, M is  $239 * 8 = 1912$  and N is  $238 * 9 = 2142$ . There are a total of two entries or time-slots for the ODU1 channel in the calendar. The rate per entry is  $(239/238) * 8/9 * 1.39968 = 239/238 * ODU0$  rate. Hence, the rate per channel is  $2 * 239/238 * ODU0$  Rate =  $239/238 \times 2.488\ 320$  Gbps, which is exactly equal to ODU1 nominal bit rate.

### 5.8.7 ODUk Rx Framer (ODU\_RFRM)

Figure 50 ODU\_RFRM Functional Diagram



ODU\_RFRM blocks are multi-channel. Each block is capable of supporting "N" channels depending on the subsystem in which the block is instantiated. See [Table 24](#) for further details.

The Receive Frame Alignment unit:

- Performs FAS and MFAS incoming alignment checks including generation of OOF, OOM, LOFLOM interrupts
- Optionally, passes a dLOFLOM onto a signal fail
- Automatically turns a signal fail into a ODUK-AIS signal

The Receive TCM and PM overhead processors:

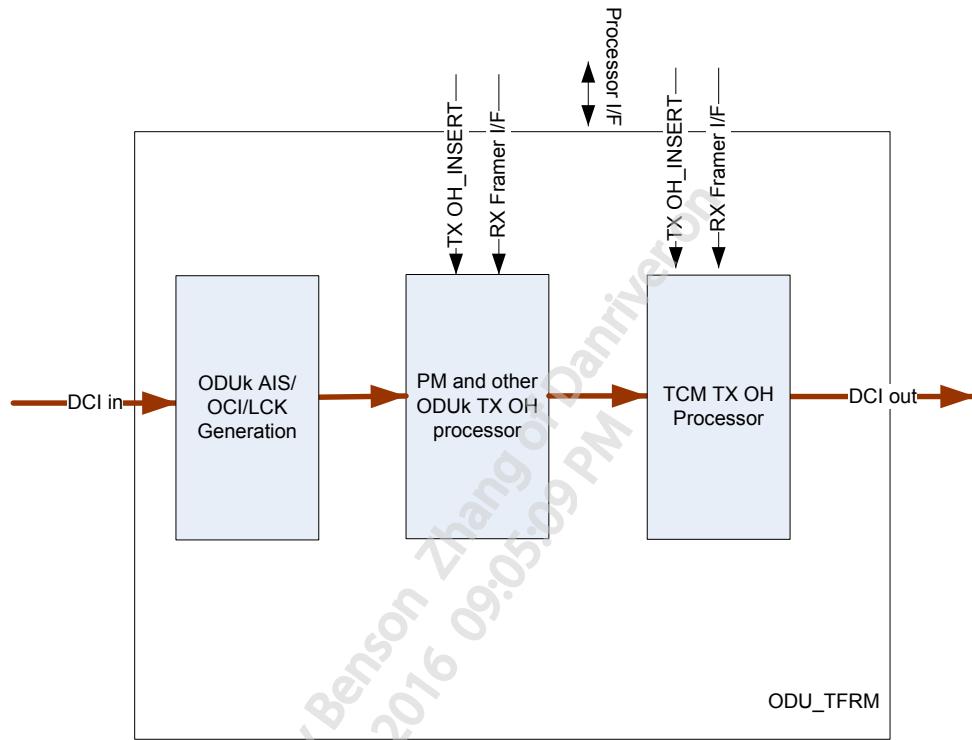
- Conditionally process all receive ODUKT and ODUKP sub-layer functions mentioned in the above features and the appropriate consequential actions. Supports a total of 256 TCMs in the ODUKT sub-layer for all channels and 72 TCMs for 12 channelframers.
- For cases leading to a signal fail, optionally generate ODUK-AIS (using above the COREOTN consequential action features)
- Check the receive overhead against expected register values (see the COREOTN overhead extraction features above)
- Extract receive overhead to overhead extract interface
- Transmit a received delay measurement signal to the corresponding transmit framer

- Broadcast BDI/BEI/BIAE/BIP-8 values to the TFRM port
- Using the dAIS/dOCI/dTIM/dDEG/dBDI/dLTC part of the ODUkP or ODUkT sink function, optionally trigger an interrupt.
- Defects/Consequential action status/interrupt can be retrieved out of RFRM through toggling the defect LCLK from PGMRCLK applied from top-level. This LCLK can be applied in any granularity(i.e periodically or after IRQ fires) . A combined defect structure supplies to the IRQ pin directly without defect LCLK toggling. The defect LCLK is applied globally for all RFRM's.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:09 PM

## 5.8.8 ODU Tx Framer (ODU\_TFRM)

Figure 51 ODU\_TFRM Functional Diagram



ODU\_TFRM blocks are multi-channel. Each block is capable of supporting "N" channels depending on the subsystem in which the block is instantiated. See [Table 24](#) for further details.

The transmit TCM and PM overhead processor unit:

- Conditionally process all transmit ODUkT and ODUkP functions mentioned in the above features and the appropriate consequential actions. Supports a total of 256 TCMs in the ODUkT sub-layer for all channels. For 12 channel framers, the pool is 72.
- Provide three means of inserting overhead according to priority (first to last priority):
  - Overhead Insertion port
  - Default registers
  - An incoming in-line DCI channel that already carries the appropriate overhead
- Choose BDI/BEI/BIAE from the receive framer interface
- If a delay measurement signal is received from the receive framer interface, perform the delay measurement protocol on the appropriate TCM or PM

The ODUk AIS/OCI/LCK unit provides the ability to insert ODUk-AIS, ODUk-OCI and ODUk-LCK patterns when configured.

The transmit ODUk framer will return backward defect indications from the receive framer to the far end. When the rate of the received ODUk is less than the rate of the transmitted ODUk, the BDI will periodically pulse low for one frame. The frequency of the low pulse will be at most once every 5000

frames. G.798 requires a persistency check of three to five frames before declaring dBFI, and therefore, the far end will not declare dBFI and the periodic pulse is benign from a system design perspective.

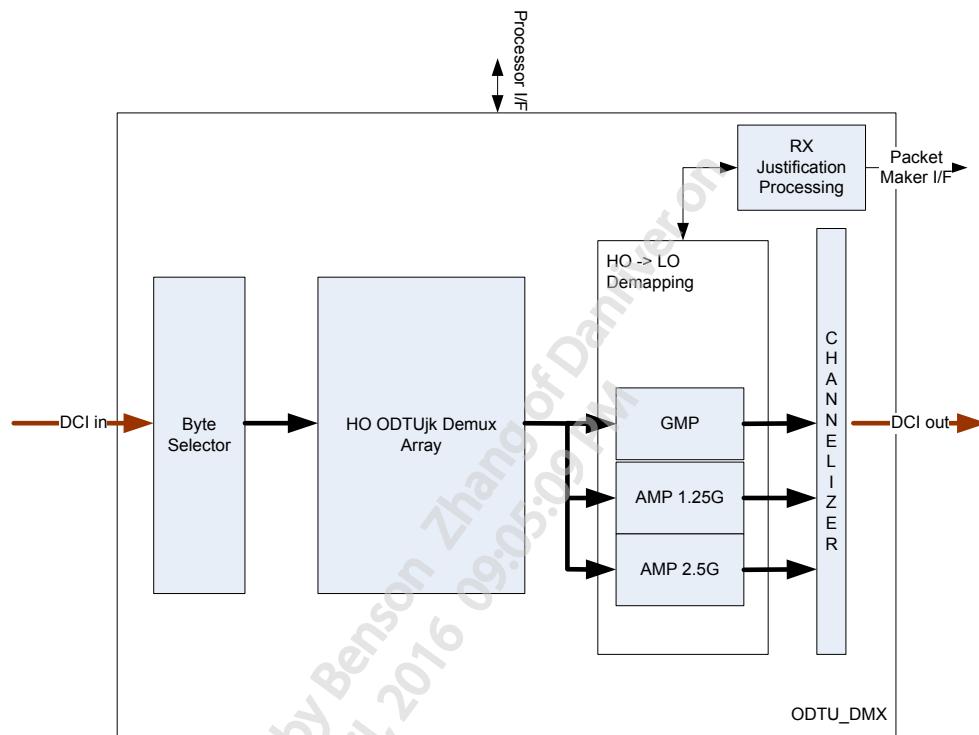
The ability to force ODUK IAE in the transmit framers is supported using a per-channel configurable IAE\_INS register bit. The IAE\_INS bit is used to force an incoming alignment error in the ODUk overhead of the operational TCMs sourced by the associated channel. A change of state (0 to 1 or 1 to 0 transition) on IAE\_INS causes a 010 pattern to be inserted in the STAT field of each sourced TCM for 16 multiframe.

Note that successive writes to the IAE\_INS bit must be spaced by at least 1  $\mu$ s to account for the synchronization delay associated with an ODUO channel.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:09 PM

## 5.8.9 ODTUjk Demux (ODTU\_DMX)

Figure 52 ODTU\_DMX Functional Diagram



The ODTU\_DMX blocks are multi-channel. The input channel configuration is a minimum one channel to a maximum of 96 channels. The appropriate output channel configuration is one since one input channel can lead to many output channels (a maximum of 96 channels).

The ODTU\_DMX has two distinct functions:

- Demuxes the HO ODUj container into several ODTUjk containers according to Tributary Slot allocation in the HO
- Takes the ODTUjk container and forms a LO ODUk container by performing appropriate GMP or AMP de-mapping protocols. Once in LO format, the ODUk must be re-channelized appropriately on the DCI out port.

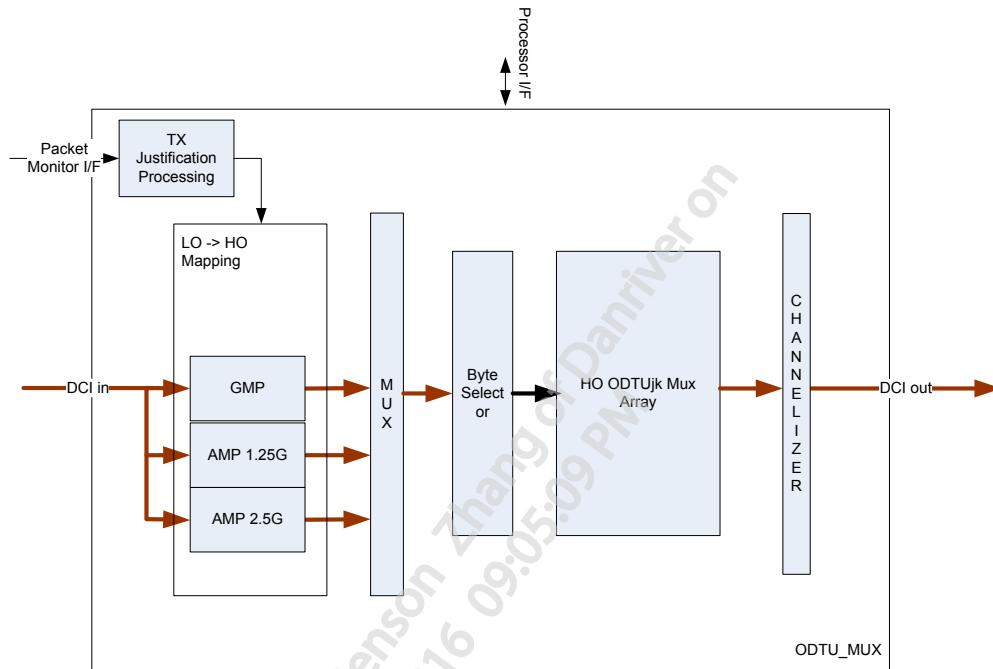
The ODTU\_DMX block also evaluates the justification controls (JC) for the appropriate de-mapping protocols to determine if the received LO is within a range of acceptable data rates for the ODUk container. If the JC is within range, the actual JC is indicated to the packet maker through its dedicated interface. If the JC is out of range, an interrupt is issued for that particular channel, but the out-of-range justification is still passed and processed.

All combinations of the ODTU\_DMX from LO to HO are highlighted in the table from [Section 5.8.4.6.5](#).

The two stages are run-time configurable and adjustable during a G.7044 feature execution by firmware. In all other cases, these stages are statically configured by firmware on start-up.

### 5.8.10 ODTUjk Mux (ODTU\_MUX)

Figure 53 ODTU\_MUX Functional Diagram



All of the ODTU\_MUX blocks are multi-channel. The input channel configuration is a minimum one channel to a maximum of 96 channels. The appropriate output channel configuration is one channel since one output channel can have many input channels (maximum of 96 channels).

The ODTU\_MUX has two distinct functions:

- Maps LO ODUJs into a single ODTUjk container through appropriate the GMP or AMP mapping protocol
- Muxes several ODTUjk containers into a HO ODUk container, according to the Tributary Slot allocation in the HO. The HO ODUk must be framed through a mini-framer to insert the appropriate FAS and MFAS values. Once in HO form, the ODUk must be re-channelized appropriately on the DCI out port.

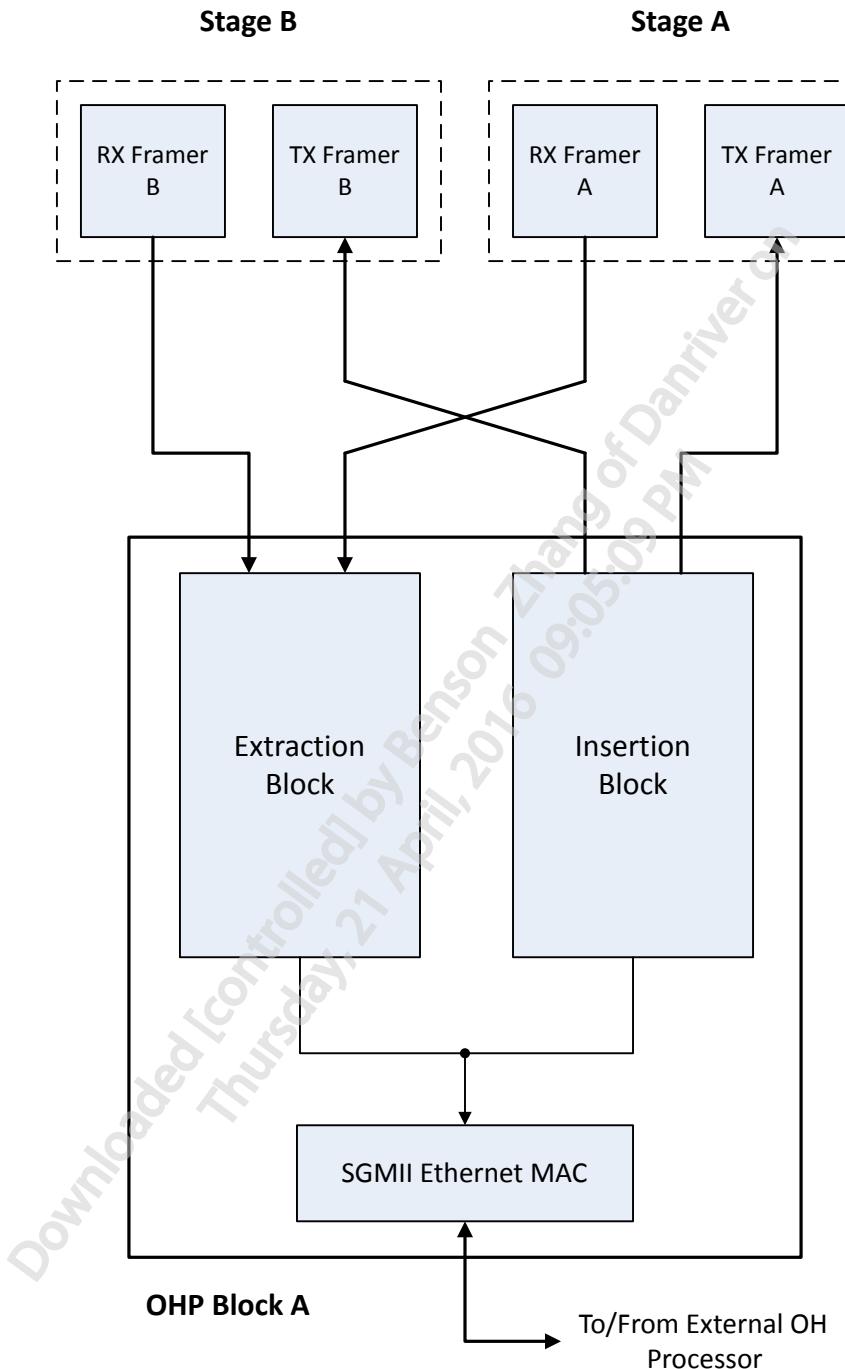
The ODTU\_MUX optionally can accept justification control (JC) from the packet monitor interface or use its internally generated nominal JC bytes to perform LO ODUj to ODTUjk mapping.

All combinations of mux from LO to HO are highlighted in the table from [Section 5.8.4.6.5](#).

The two stages can be run-time configurable and adjustable during a G.7044 feature execution by firmware or software. In all other cases these stages are statically configured by firmware on start-up.

### 5.8.11 ODU Overhead Block (ODU\_OHP)

Figure 54 ODU\_OHP Functional Block Diagram



The COREOTN subsystem provides three ODU\_OHP blocks.

The ODU\_OHP blocks coordinate extraction and insertion of overhead bytes between the framers and an external device. Overhead is extracted from each channel in the receive framer stage and inserted from each channel in the transmit framer stage. Each ODU\_OHP block is shared between two framer stages. The transmission medium for the overhead data is through a SGMII link that runs at a 1 GE full-duplex rate. Which two framers are connected to the ODU\_OHP block is configurable from the stages of framers in the COREOTN (OHP1 - Stage 1 and Stage N dedicated; OHP2 and OHP3 can be configurable between Stage 2, 3A, 3B and 4).

The extraction done by the ODU\_OHP block is before any frame/layer processing of the ODU but after the OTU layer is processed. The insertion done by the ODU\_OHP block is after the ODU layer is processed before going to the OTU layer.

The feature set of the ODU\_OHP block is as follows.

### 5.8.11.1 ODU\_OHP Features

Single OHP block is shared between two pairs of transmit and receive framers from two stages. See [Figure 54](#). These transmit and receive framers are served in a round-robin fashion with equal priority for access to the GE subsystem. [Figure 55](#) shows the overhead bytes that can be inserted or extracted for each OTUk channel.

**Figure 55 OTN Channel Overhead Data Frame**

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16				
1	FAS					MFAS	SM			GCC 0		RES (1,13)	RES (1,14)	JC4	SM					
2	RES (2,1)	RES (2,2)	PM&T CM	TCM ACT	TCM 6			TCM 5			TCM 4			FTFL	JC5					
3	TCM 3			TCM 2			TCM 1			PM			EXP (3,13)	EXP (3,14)	JC6					
4	GCC 1		GCC 2		APS / PCC / (4,5)	APS / PCC / (4,6)	APS / PCC / (4,7)	APS / PCC / (4,8)	RES (4,9)	RES (4,10)	RES (4,11)	RES (4,12)	RES (4,13)	RES (4,14)	PSI	OMFI				
SM / PM / TCM #																				
TTI BIP-8 BEI / BDI STAT / RES																				

For extraction, all overhead bytes (shown in white above) are extracted except BIP-8 from the PM/TCMs and JC1, JC2, and JC3. OTUk overhead (row 1, bytes 7-14) are only extracted on Stage 1 framers. The OHP block is configured in packed mode where only the bytes that marked for extraction are included in the data packet. In packed mode, the overhead marked for extraction is output in the data packet in the order it is received on the line. The total overhead extraction capacity is 40 bytes per frame. The 40-byte selection is handled through registers statically per channel.

For insertion, all overhead bytes (shown in white above) including FAS bytes are inserted except the BIP-8 bytes from the SM, PM, and TCM#, limited to any 40 bytes inserted out of 53 bytes (excluding MFAS).

OTUk overhead (row 1, bytes 1-14) including FAS is only inserted on Stage 1 and Stage N framers. Insertion Enable bytes are provided for the user to select which bytes to insert. The 40-byte selection is statically handled through registers on a per channel basis.

The 40-byte selection on both the insertion and extraction side per channel is independent. The selection allows 10 words to be chosen out of the 16 listed in [Table 40](#).

**Table 40 Overhead Byte Ordering**

Row 1	Word 0	{FAS, FAS, FAS, FAS}
	Word 1	{FAS, FAS, MFAS, 0x00}
	Word 2	{SM_TTI, SM_BW_INFO, GCC0_0, GCC0_1}
	Word 3	{RES, RES, JC4, 0x00}
Row 2	Word 4	{RES, RES, PM&TCM, TCM_ACT}
	Word 5	{0x00, 0x00, 0x00, 0x00}
	Word 6	{TCM6_TTI, TCM6_BW_INFO, TCM5_TTI, TCM5_BW_INFO}
	Word 7	{TCM4_TTI, TCM4_BW_INFO, FTFL, JC5}
Row 3	Word 8	{0x00, 0x00, 0x00, 0x00}
	Word 9	{TCM3_TTI, TCM3_BW_INFO, TCM2_TTI, TCM2_BW_INFO}
	Word 10	{TCM1_TTI, TCM1_BW_INFO, PM_TTI, PM_BW_INFO}
	Word 11	{EXP, EXP, JC6, 0x00}
Row 4	Word 12	{GCC1_0, GCC1_1, GCC2_0, GCC2_1}
	Word 13	{APS, APS, APS, APS}
	Word 14	{RES, RES, RES, RES}
	Word 15	{RES, RES, PSI, OMFI}

#### 5.8.11.1.1 Extraction Frame Structure

On the extract path, this 64 byte (16 words) overhead data is extracted by the receive framer and sent to the OHP block for processing. The extraction module in the OHP block receives and processes this data. It selects 10 words per frame based on per channel programmable extraction mask and stores it in the channel's FIFO. Internal status registers keep the record of accumulated data words for each channel from each receive framers. Later when 8 frames of overhead data (with each frame consisting of maximum 10 words) have been buffered for any given channel, they are transmitted to external overhead processor over SGMII Ethernet links. Note that 10 words have to be chosen or else extraction will not function. The extraction frame structure is the same for all data packets.

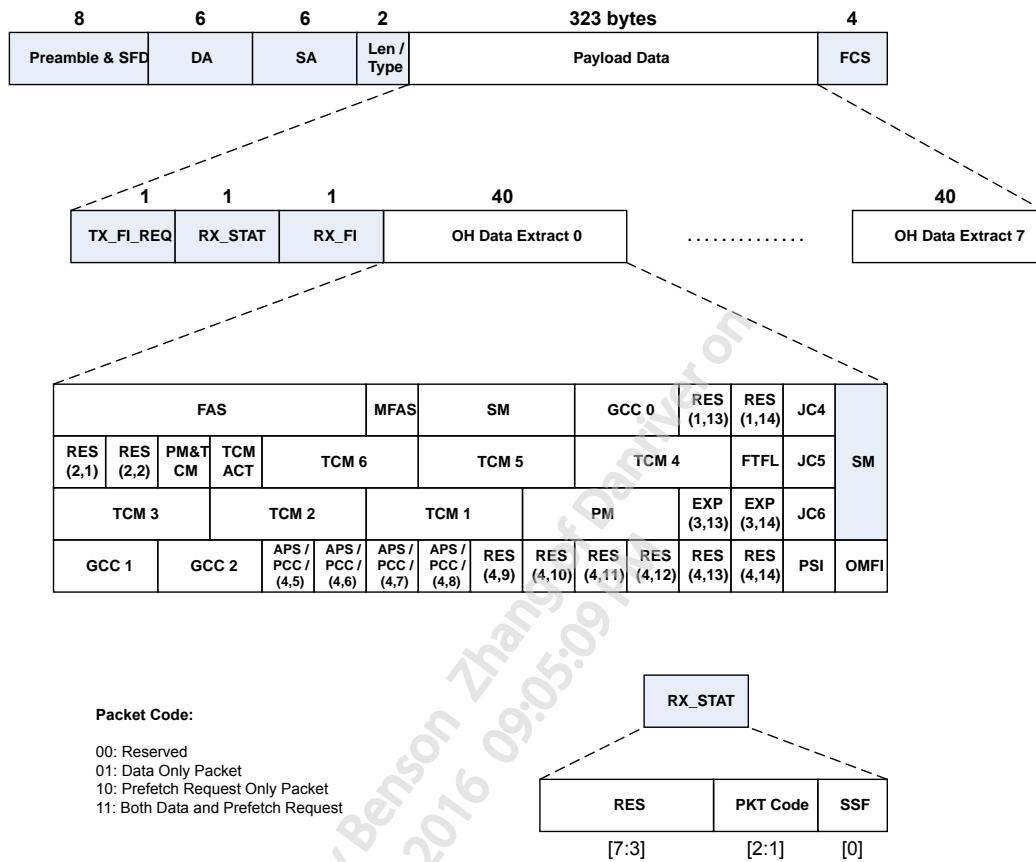
Any channel can request three types of services from the extraction controller:

- Data Prefetch Service Request: The extraction block sends Request-Only packets for a channel to prefetch 8 frames of insertion overhead data when insertion block has processed 8 frames of data for that channel (that is, the transmit framer has moved 8 frames ahead from the previous request).
- Data Transmission Service Request: When this channel has 8 frames of extracted overhead data in its FIFO to send out to external overhead processor (that is, the receive framer has given 8 frames worth of overhead to the ODU\_OHP).
- Both Data and Prefetch Requests: When a channel requires both prefetch data as well as it has 8 frames of extracted overhead data to send to external overhead processor.

The extraction scheduler is responsible for arbitrating the requests from multiple channels.

The structure of the Extraction Ethernet packet (transmit data from the device OHP SGMII port) is shown in [Figure 56](#).

Figure 56 Ethernet Packet for Extracted Overhead Data

**Note:**

- Preamble and SFD (7 + 1 Bytes): 7 bytes of Preamble bit pattern and a single SFD byte. These bytes are generated and inserted by the transmit Engine of the FEGE subsystem.
  - Destination Address (DA, 6 Bytes): 48-bit identifier consisting of the programmable, user-specified 32-bit DA\_UID, the 8-bit Framer ID (OHP input port) and the 8-bit Channel ID of the channel requesting the transmit service. This is not valid in the extraction packet.
  - Source Address (SA, 6 Bytes): 48-bit identifier consisting of programmable, user-specified 32-bit SA\_UID, the 8-bit Framer ID (OHP input port) and the 8-bit Channel ID of channel requesting the transmit service.
    - SA[47:16] = user-specific ID
    - SA[15:8] = framer stage number
    - SA[7:0] = channel ID
- Note that bit 47 is the MSB in the SA field. The Channel ID ranges from 0-95 and the Framer ID ranges from 0-5 (where 0 – stg1, 1 – stge2, 2 – stg3A, 3 – stg3B, 4 – stg4, 5- stgN).
- Length/Type (2 Bytes): Gives the length of the payload in the Ethernet MAC frame. For request only packets, it is the fixed decimal number 46. For Data Packets and combined Request+Data packets, it is the fixed decimal number 323.
  - FCS (4 Bytes): CRC Frame Check Sequence for error detection. It is calculated and appended by the transmit engine of the FEGE subsystem.

- Payload Data (Fixed 323 Bytes): Carries the extracted overhead data for any channel for 8 frames (OH\_DATA\_EXTRACT\_0 to OH\_DATA\_EXTRACT\_7). Each frame consists of 10 overhead words, therefore, the maximum fixed size of the overhead data is 80 words (320 bytes) for 8 frames. Three control bytes are appended to the overhead payload:
    - TX\_FI\_REQ [7:0]: The transmit Frame Indicator managed by the Insertion Block. This value is calculated by Insertion block for each channel by discarding three LSBs of the channel's current TX\_MFAS and adding the channel-specific Insertion Offset to it. This value is given as:
 

For any channel  $i$ ,

      1.  $TX\_FI = \{TX\_MFAS[7:3], 3'b000\}$ ;
      2.  $TX\_FI\_REQ = TX\_FI + ins\_offset[i]$ ;

For Data Only Packets, this field merely indicates the current location of the transmit framer for that channel to the external overhead processor. For Request Only packets and Data+Request packets, this value indicates the starting location of the 8 frames being requested from the external overhead processor.
- The recommended and driver programmed offsets are as follows:

Table 41 Channel Offset Values for Insertion Block

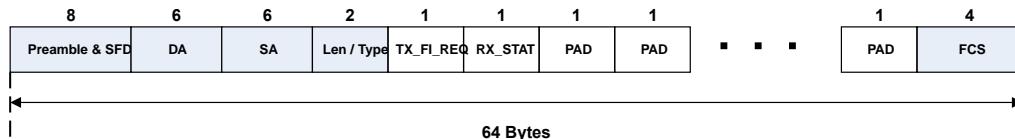
Channel Data Rate	96 Channel OHP Offset	12 Channel OHP Offset
ODU0	8	N/A
ODU1	16	N/A
ODU2	64	8
ODU3	256	32
ODU4	256	64

- RX\_FI [7:0]: RX Frame Indicator. As described before, the extraction block buffers 8 sequential frames for each channel before they are sent out to the external overhead processor. This field contains the RX\_FI value of the first of the eight sequential frames in the packet. All the frames in the packet have RX\_FI values, which increment sequentially starting from this value of RX\_FI. This field is valid only if packet contains extracted overhead data for 8 frames. It can start from any number from 0 to 255. Unlike TX\_FI\_REQ, it does not have to be a multiple of 8.
- RX\_STAT [7:0]: Status field. Carries the SSF Alarm and status information about the type of packet being sent to external overhead processor:
  - Bit [0]: Signal Fail indication (SSF Alarm) for given Channel (Active High).
  - Bit [2:1]: Packet Type Indicator. When:
    - 2'b00: Reserved
    - 2'b01: Data Only Packet
    - 2'b10: Prefetch Request Only Packet
    - 2'b11: Both Data and Prefetch Request
    - Bit [3:7]: Reserved bits. Set to '0'.
  - SSF: Server Signal Fail. Indicates that the ODU is currently receiving a Signal fail from Higher layer (that is, the ODU layer is receiving CI\_SSF). This means that the Overhead associated with this channel is unreliable when the SSF is asserted.

The structure of Request-Only Packets is given in [Figure 57](#). The unshaded blocks represent payload bytes.

This Request-Only packet is generated whenever insertion block completes processing of eight frames for any given channel. Calculation of the TX\_FI\_REQ is explained above.

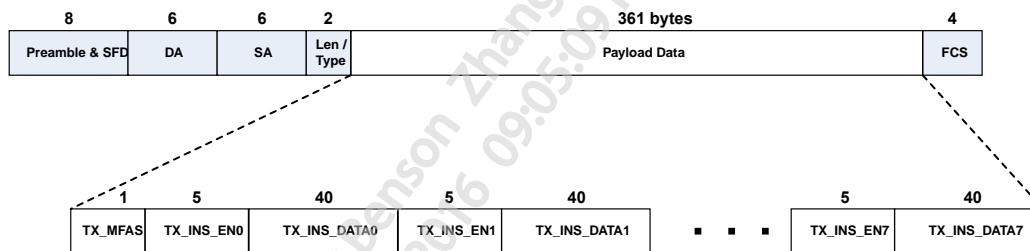
Figure 57 Request-Only Packet Structure



### 5.8.11.1.2 Insertion Frame Structure

When the external overhead processor (FPGA) receives request for overhead data for any given channel, it assembles 8 frames of overhead data for that particular channel and transmits them over a 1 Gbps SGMII link. The structure of the overhead data packet, comprising of 8 overhead frames, received from the external overhead processor is shown in [Figure 58](#). Each frame of insertion data is limited to 10 words (40 bytes). The same 10 as the extraction. Each byte of the frame is qualified by the corresponding mask bit. Therefore, all 40 bytes of the frame have corresponding 40 bits of an Insertion Field Mask as well, rendering the total size of each frame at 45 bytes. Thus the total payload size for 8 frames within the insertion packet is 360 bytes. A single byte TX\_MFAS value is appended to this payload. Their detail is given in [Figure 58](#).

Figure 58 Overhead Data Insertion Frame



#### Note:

- TX\_MFAS[7:0]: This is the TX\_MFAS value of the first frame of insertion overhead data and must be same as the TX\_FI\_REQ in the Request-Only or Data+Request packets. Refer to Section 3 on how it is used.
- TX\_INS\_EN[N][39:0]: This insertion mask corresponds to the TX\_INS\_DATA[N]. The data is always 40 bytes long and is organized from the global mask selection order. So, for instance, if we have words 0-4, 10-14 chosen as the 10 words, the order of insertion will be 0,1,2,3,4,10,11,12,13,14 in the TX\_INS\_DATA[N] always, regardless of TX\_INS\_EN[N]. TX\_INS\_EN[N][39:0] indicates which one is valid. TX\_INS\_EN[N][0] corresponds to the TX\_INS\_DATA[N][0] byte. Refer to Table 1 for the overhead bytes included in each insertion word.
- Preamble and SFD (7 + 1 Bytes): 7 bytes of Preamble bit pattern and a single SFD byte. These bytes are generated and inserted by the transmit engine of the FEGE subsystem.
- Destination Address (DA, 6 Bytes): 48-bit identifier consisting of the programmable, user-specified 32-bit DA\_UID, the 8-bit Framer ID (OHP input port) and the 8-bit Channel ID of the channel requesting the transmit service:
  - DA[47:16] = user-specific ID
  - DA[15:8] = framer stage number
  - DA[7:0] = channel ID

Note that bit 47 is the MSB in the SA field. The Channel ID ranges from 0-95 and the Framer ID ranges from 0-5 (where 0 – stg1, 1 – stge2, 2 – stg3A, 3 – stg3B, 4 – stg4, 5 – stgN)

- Source Address (SA, 6 Bytes): A 48-bit identifier consisting of the programmable, user-specified 32-bit SA\_UID, the 8-bit Framer ID (OHP input port) and the 8-bit Channel ID of the channel requesting the transmit service. This is not used in the insertion path.
- Length/Type (2 Bytes): This is always a fixed length of 361.
- FCS (4 Bytes): CRC Frame Check Sequence for error detection.

#### 5.8.11.1.3 GE MAC

The GE subsystem consists of a TX\_MAC, a RX\_MAC, a GE PCS layer and a packet status monitoring block. It handles the actual Ethernet frame transmit and receive functions using the PCS layer. The GE subsystem, in conjunction with the GE SERDES, supports a serial electrical interface to connect to industry standard 1000 Base-FX SFP optical transceivers.

The subsystem features:

- Support for 1000 Base-FX SFP optical transceivers using GE SERDES.
- A GE PCS supporting 8B/10B encoding/decoding as well as GE auto-negotiation.
- Support of an SGMII (clock-less) interface that allows a single SGMII-compliant optical module to support GE rates.
- Uni-directional Ethernet (IEEE 802.3 clause 66), which allows for continued uni-directional transmission of frames in the absence of a valid 'link up' indication. This can be configured on any of the Ethernet interfaces by turning off bi-directional protocols such as auto-negotiation, link OAM, timing, etc.)
- A GE channel that typically works on its own recovered clock in the receive direction and a common nominal transmit clock in the transmit direction.
- A GE MSTATX block to keep a log of important events seen by the transmit and receive MACs.
- A count of all 8B/10B line-code violations and running disparity errors, which is performed by the receive PCS block (L1RPP\_FEGE). These counts can all be read by an external CPU

#### 5.8.11.2 Response Time

The response to request latency is scalable with the amount of bandwidth employed in the OHP. The maximum bandwidth of overhead the OHP can support is two framers worth that are running +120G of bandwidth each of OTN data (96x2 ODU0 worth or 12x2 ODU2\*).

The following are the absolute times on the SGMII interface:

- The Extraction packet (req+ data, or only data) is 349 bytes, which is 2.8  $\mu$ s on the SGMII interface for transfers including 12 IPG.
- The Request only packet is 64 bytes which is 0.608  $\mu$ s on the SGMII interface for transfers including 12 IPG.
- The insertion packet is 399 bytes which is 3.2  $\mu$ s on the SGMII interface for transfers including 12 IPG.

Since the insertion packet is larger than the extraction packet or the request-only packet, the FPGA needs to store the requests coming from the extraction stream into a FIFO. The minimum depth of the request FIFO needs to be around 192 (regardless of channel combinations).

Internally, (inside the device), the insertion data is stored in an internal per-channel FIFO prior to being inserted into the ODU\_TFRM. The size of the FIFO scales according to the rate. The sizes of the various rates are as follows:

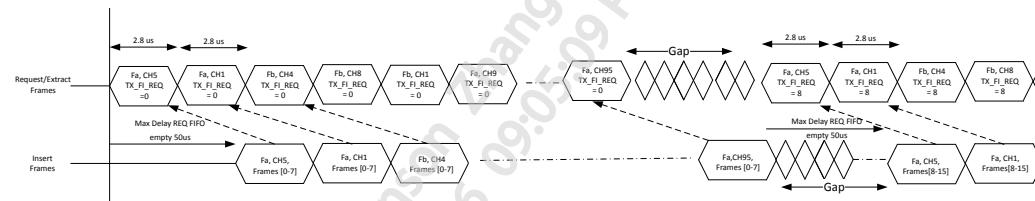
- ODU0 – 16 frames (For OHP96)
  - ODU1 – 32 frames (For OHP96)
  - ODU2 – 128 frames (For OHP96) and 16 frames (For OHP12 - Stage 1/N)
  - ODU3 – 512 frames (For OHP96) and 64 frames (For OHP12 - Stage 1/N)
  - ODU4 – 512 frames (For OHP96) and 128 frames (For OHP12 - Stage 1/N)

The request packets (TX\_FI sent) center this insertion FIFO. So for instance in an ODUO if this FIFO is less than 8 frames of data, then a request packet is queued to be sent to the external FPGA. For all ODUk rates, the center is around where the TX\_FI is set (which is half-way in the FIFO).

With that in mind you can roughly see that for a single ODU0 request to the absolute time, it needs data back inside the insertion FIFO is around  $800\mu\text{s}$  (8 frames). But, with multiple ODU0s, the user needs to optimize the turnaround latency such each channel can be serviced if there is a continuous burst of requests. The recommendation is to make sure that the insertion pipe is always full whenever the request FIFO in the external FPGA has data.

Figure 59 shows how the request and insertion interfaces can run if the external FPGA uses 50  $\mu$ s to prime the initial pipe (not that it is only initial pipe; but this ensures the pipe is always full).

**Figure 59 Example Showing Response Time Bandwidth**

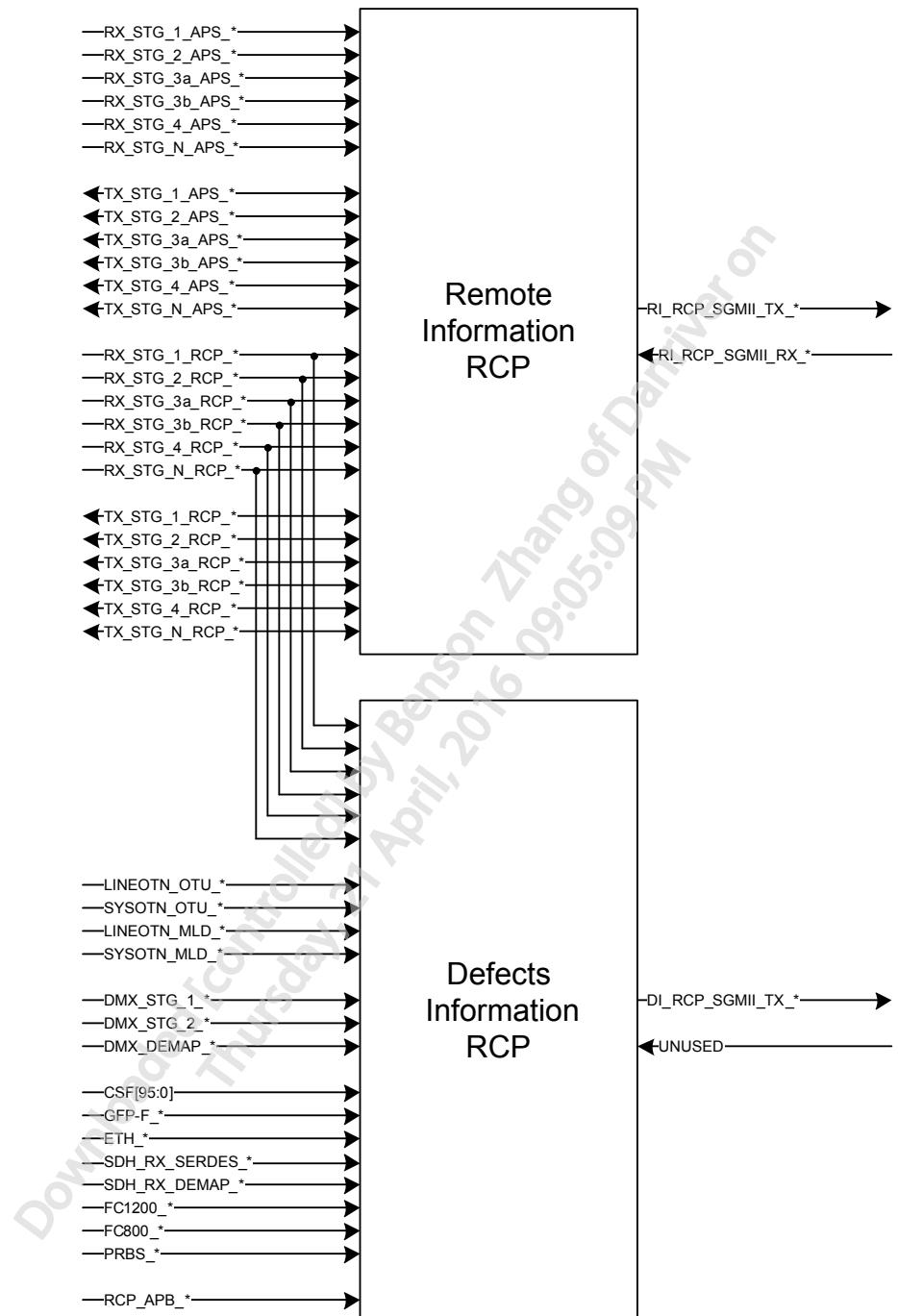


As you can see above that the insertion logic in the external device can take up to 50  $\mu$ s to prime the insertion upon receiving the first request, but after receiving the first request, consecutive insertions need to happen back-to-back without delays such that it always has the insertion SGMII full. The re-priming of the insertion pipe absorbs another 50  $\mu$ s as shown at the break point T1 in the diagram on the second set of request cycles.

**Note:** The 50  $\mu$ s priming latency is just an example. The hard specification is the overall turnaround time per ODUk channel, which is around 800  $\mu$ s. With multiple ODUks, the time would need to be managed.

### 5.8.12 ODU Ring Control Port Block (ODU\_RCP)

Figure 60 RCP Top Level Block Diagram



The RCP block has two interfaces: RI\_RCP(Remote Information Ring Control Port) and DI\_RCP (Defects Information Ring Control Port). These interfaces are used to communicate backward indications and defects/alarms off-chip.

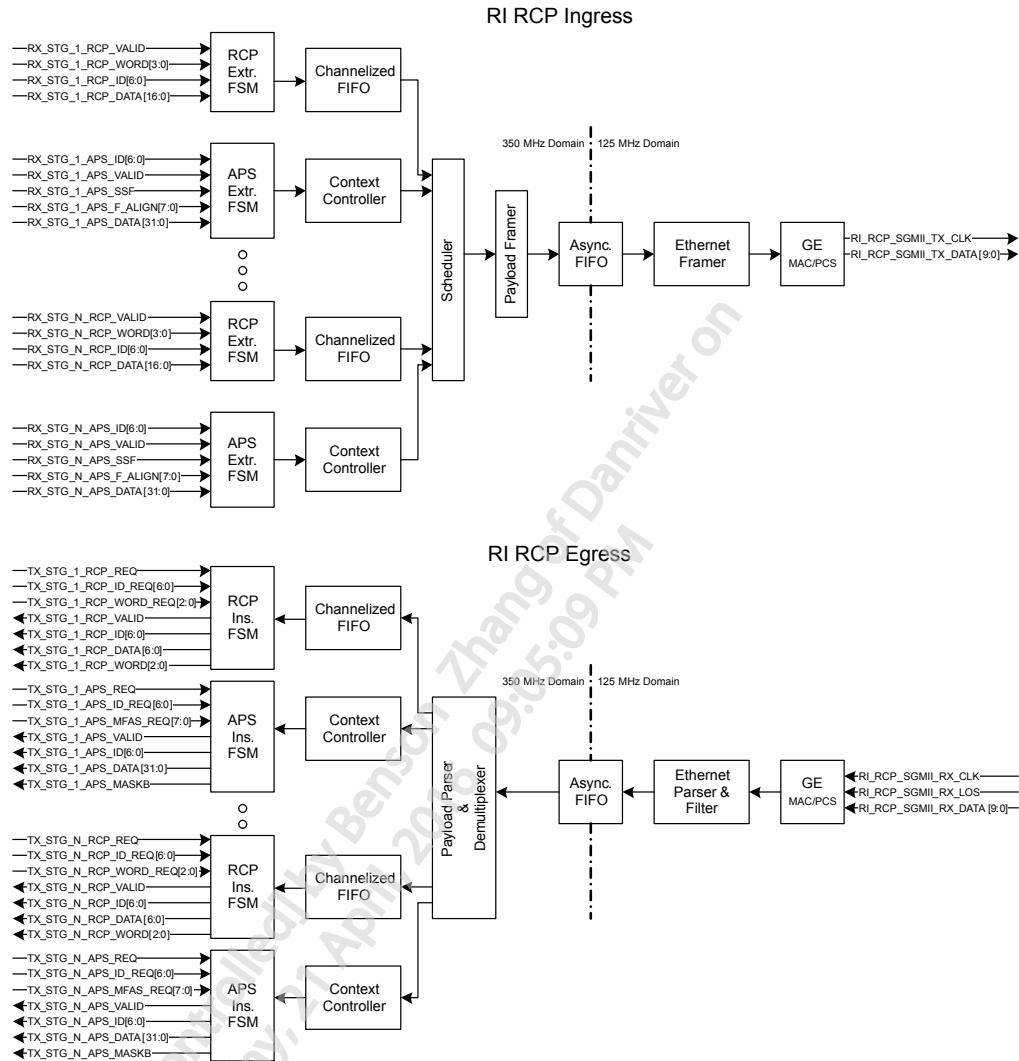
### 5.8.12.1 **Remote Information Ring Control Port**

The RI\_RCP supports communication of BDI/BEI/BIAE bits to and from up to 6 framer stages within the COREOTN subsystem. Within each framer stage there are 6 TCM layers and 1 PM layer per channel. Refer to the hardware specification document for detailed information about the framer stages supported by the device.

The Ethernet MAC SA field on transmit packets indicates the source framer stage and channel identification for each packet delivered onto the RI\_RCP interface. The Ethernet packet payload carries the 6 TCM and 1 PM fields per frame, and there are 8 frames worth of data in one Ethernet packet. To simplify the extraction and alarm processing, the BDI bit and BEI/BIAE nibble are byte aligned. Also included in the payload are four APS/PCC bytes from a single connection monitoring level. The level is user configurable. Details of the Ethernet frame format are described in [Section 5.8.12.1.7](#).

Downloaded [controlled] by Benson Zhang of Dayu  
Thursday, 21 April, 2016 09:05:09 PM

Figure 61 RI\_RCP Top Level Block Diagram



### 5.8.12.1.1 Features

- Full Duplex 1 Gbps MAC controller and SGMII PCS layer
- Ethernet frame assembly and disassembly by the Ethernet MAC block
- Full duplex APS/PCC, BDI/BEI/BIAE bits extraction and insertion
- Up to 96 ODUk channels per framer stage are supported

### 5.8.12.1.2 APS/PCC & RCP Extraction

- The BDI/BEI/BIAE bits from the TCMs and PM layers are extracted.
- The four APS/PCC bytes from a single connection monitoring level are extracted (one of ODUk Path, ODUk TCM1-6, or ODUk server layer trail where the server layer trail is an OTUk or HO ODUk). The monitoring level is user programmable per channel.

- Two channelized FIFOs are used to accumulate eight frames worth of RCP content (BDI/BEI/BIAE) and the four APS/PCC bytes from a single connection monitoring level. For every 8 ODUk frames a four byte APS/PCC word is received from the Rx framer stage.
- Per channel FIFO reset is supported. Remaining frames will be discarded when a channel is taken offline.
- The Ethernet interface throughput is higher than the source rate. The FIFO size supports the worst case scenario where all channels from all framer stages are ready to transmit at the same time.
- Only the BDI and BEI/BIAE bits from the fixed/floating TCM1-6 and PM layers in operational/monitor mode are sent, i.e. both the fixed and floating layer can be enabled but only one at a time can be in the operational/monitor mode.
- An APS/PCC Finite State Machine (FSM) monitors the sideband Multi-Frame Alignment Signal (MFAS) and captures the APS/PCC bytes from the ODUk Path, ODUk TCM1-6 or ODUk server layer trail.
- The APS/PCC FSM samples the Server Signal Fail (SSF) over the 8 ODUk frame window where the APS/PCC bytes are captured. The SSF status is reported in the Ethernet payload. The SSF status is sticky, i.e. it is recorded if set during the accumulation of the eight ODUk frames and cleared after the transfer with the APS/PCC word.

#### 5.8.12.1.3 APS/PCC & RCP Insertion

- The APS/PCC bytes are stored in the context memory (single word per channel). The last known APS/PCC value is always used in response.
- A single APS/PCC monitoring level is selected. The interface must insert the APS/PCC bytes in the selected level by monitoring the three LSB of the MFAS byte (ITU-T G.709/Y.1331 MFAS bits 6, 7 and 8). The insertion monitoring level is taken from the RI\_RCP packet payload which is set by the ingress port of the paired remote device.
- A user configurable per channel enable is provided to control the insertion of the APS/PCC bytes.
- The Rx framer Server Signal Fail (SSF) status bit transported inside the APS/PCC status byte of the RI\_RCP packet payload is not carried over to the Tx framer (i.e. the SSF field is not used by the APS/PCC & RCP insertion port).

#### 5.8.12.1.4 Extraction Scheduler & Payload Framer

- There are two levels of scheduler. The first level consists of 6 schedulers, one per Rx framer stage. It can be configured to implement a round-robin algorithm to provide fairness when servicing 96 ODU0 channels, or to implement a weighted round-robin when servicing a mix of high and low order ODUs. The calendars include a shadow copy to allow the schedulers to be reconfigured while traffic is flowing. The second level scheduler multiplexes the requests from the 6 framer stages using a round-robin algorithm.
- The Payload Framer block extracts from the channelized FIFO the compacted BDI bit and BEI/BIAE nibble from all TCM1-6 and PM layers. The format of the Ethernet frame is shown in [Figure 62](#). The Ethernet header is added in the Ethernet Framer module.
- The four APS/PCC bytes from the user selected connection monitoring level are pre-pended to the BDI and BEI/BIAE content from eight ODUk frames. The Server Signal Fail (SSF) state follows in a status byte.
- The Payload Framer block fills the reserved bits with zeroes.

#### 5.8.12.1.5 Payload Parser & De-Multiplexer

- The Tx framer stage and channel identification extracted from the Ethernet MAC DA field are used to select the destination FIFO.
- The Payload Parser block extracts the byte aligned BDI and BEI/BIAE bits per ODUk frame and stores the data in the channelized FIFO.
- The four APS/PCC bytes are extracted and stored in a channelized context memory. The latest APS/PCC word received will overwrite the memory content.

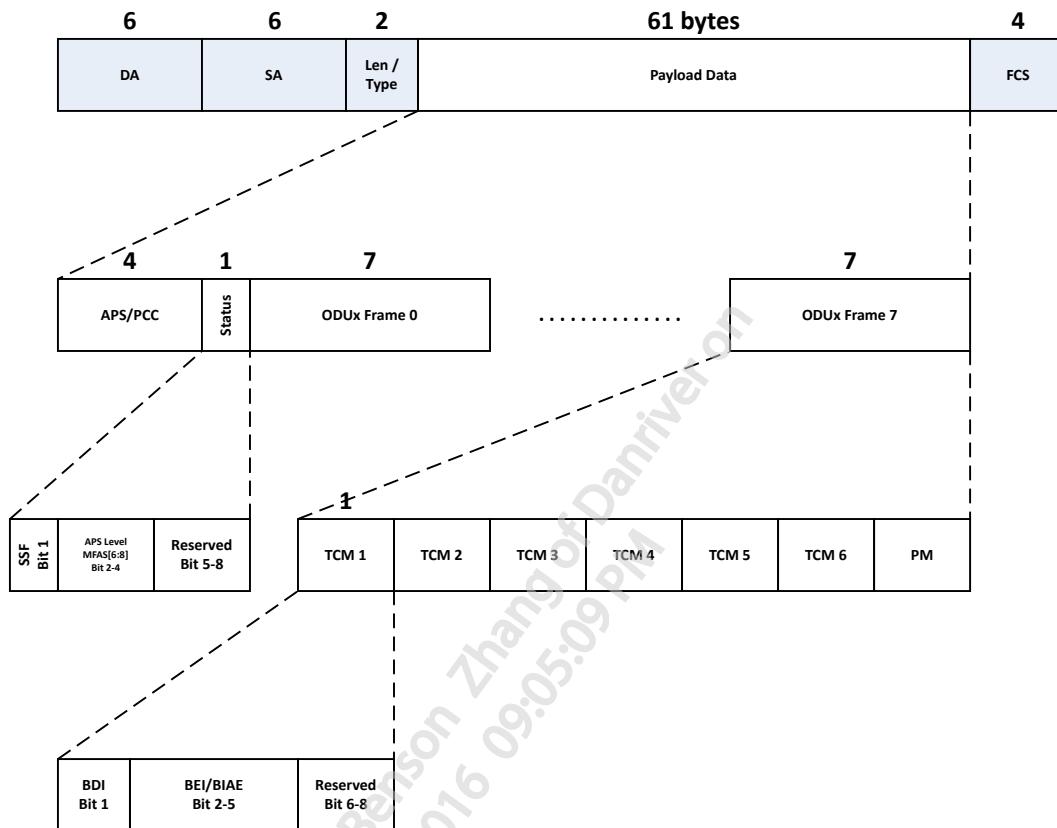
#### 5.8.12.1.6 Ethernet Framer, Parser & Filter Features

- Ethernet II frame format is used. The two byte field following the MAC SA is the EtherType. The value is user programmable. The original 802.3 format with a length field followed by the 802.2 LLC is not supported.
- The IFG, preamble, SFD and FCS are inserted/stripped by the GE MAC.
- Received Ethernet MAC DA is parsed and all frames not matching the local user identification (broadcast, flood, pause, LACP, etc.) are discarded.
- From the received Ethernet packets, the ODUk framer stage and channel fields encoded in the MAC DA are extracted and used to select Tx framer stage and channel.
- Any errors (8B/10B code violation, FCS error, etc.) flagged by the GE MAC/PCS will trigger a frame discard. The Ethernet MAC does not support back pressure from ODUk Tx framer stages. In the event of Rx FIFO overflow an interrupt is generated and the entire frame is discarded. In either cases, the previous memory content for the channel are used in response.
- For the transmit Ethernet packets, the MAC DA, MAC SA and EtherType fields are added. The MAC SA field is encoded with the local user, framer stage and channel identification. The local user identification is software configurable. The MAC DA is encoded with the result of a memory lookup using the source framer stage and channel identification. The lookup table can be configured by software. The EtherType value is software configurable.
- A shadow copy of the destination identification lookup memory is provided to allow reconfiguration while traffic is flowing.
- The RI\_RCP and DI\_RCP interfaces use the same Ethernet Framer module.

#### 5.8.12.1.7 Ethernet Frame Format

The RI\_RCP Ethernet format is shown in [Figure 62](#).

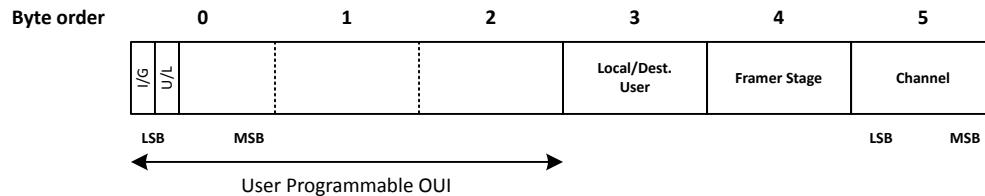
Figure 62 RI\_RCP Ethernet Frame Format



The 48-bit MAC source and destination address are encoded as shown in [Figure 63](#). The first three bytes are software programmable Organizationaly Unique Identifier (OUI). The fourth byte represent the software programmable user identification (a.k.a. the line card identification when a single DIGI device is used per card), the MAC SA is encoded with the source user identification while the MAC DA is encoded with the destination user identification. The last two bytes represent the framer stage and channel identification respectively. Both MAC SA and DA are encoded with the local framer stage (1, 2, 3a, 3b, 4, N) and channel identification (0-95). The framer stages 1, 2, 3a, 3b, 4 and N are numbered 0, 1, 2, 3, 4 and 5 respectively.

The byte order of the MAC address is big-endian but the bit order is little-endian as shown in [Figure 63](#). For example a MAC DA of 00-01-02-03-04-05 shows that the OUI value in the configuration register is 0x000102, the DIGI 120G device identification is 0x03, the framer stage is 4 and the channel number is 5. The first byte sent to the network is 0x00, followed by 0x01 then 0x02 and so on.

Figure 63 MAC Address Encoding



#### 5.8.12.1.8 RI\_RCP Ethernet Bandwidth Calculation

The bandwidth calculations apply to both ingress and egress directions. The Ethernet MAC is assumed to burst at Ethernet line rate and no shaping is done on the transmit traffic. The maximum bandwidth is 569 Mbps in the worst case scenario when all channels from all stages are used.

Table 42 RI\_RCP Ethernet Bandwidth Calculation

ODUx frame size =	TCM1 + TCM2 + TCM3 + TCM4 + TCM5 + TCM6 + PM					
	7 bytes					
ODUx frm / PDU =	8 frames					
APS/PCC + Status =	5 bytes					
Eth. Header + Overhead =	MAC DA + MAC SA + EtherType + FCS + IFG + Preamble/SoF					
	38 bytes					
Ethernet PDU =	Eth. Header + Overhead + APS/ACC + Status + Nb ODUx frames * ODUx frame size					
	99 bytes					
Eth. Frm Period / ODUx Chan. =	ODUx Chan. Period * Nb ODUx Frame / EthPDU					
Eth. BW / Framer Stage =	Nb Chan. * (Eth. PDU * 8 bit /bytes) / (ODUx Chan. Period * Nb ODUx Frame / EthPDU)					
Stage =	Stage 1	Stage 2	Stage 3a	Stage 3b	Stage 4	Stage N
Nb ODUx Channels =	12	96	96	96	96	1
ODUx Chan. Period =	1.22E-05	9.84E-05	9.84E-05	9.84E-05	9.84E-05	1.17E-06
Eth. Frm Period / ODUx Chan. =	9.75E-05	7.87E-04	7.87E-04	7.87E-04	7.87E-04	9.34E-06
Eth. BW / Framer Stage =	9.74E+07	9.66E+07	9.66E+07	9.66E+07	9.66E+07	8.48E+07
Total BW =	Stage 1 + Stage 2 +Stage 3a + Stage 3b + Stage 4 + Stage N Mbps					
	569	Mbps				

#### 5.8.12.1.8.1 GE Ethernet MAC Features

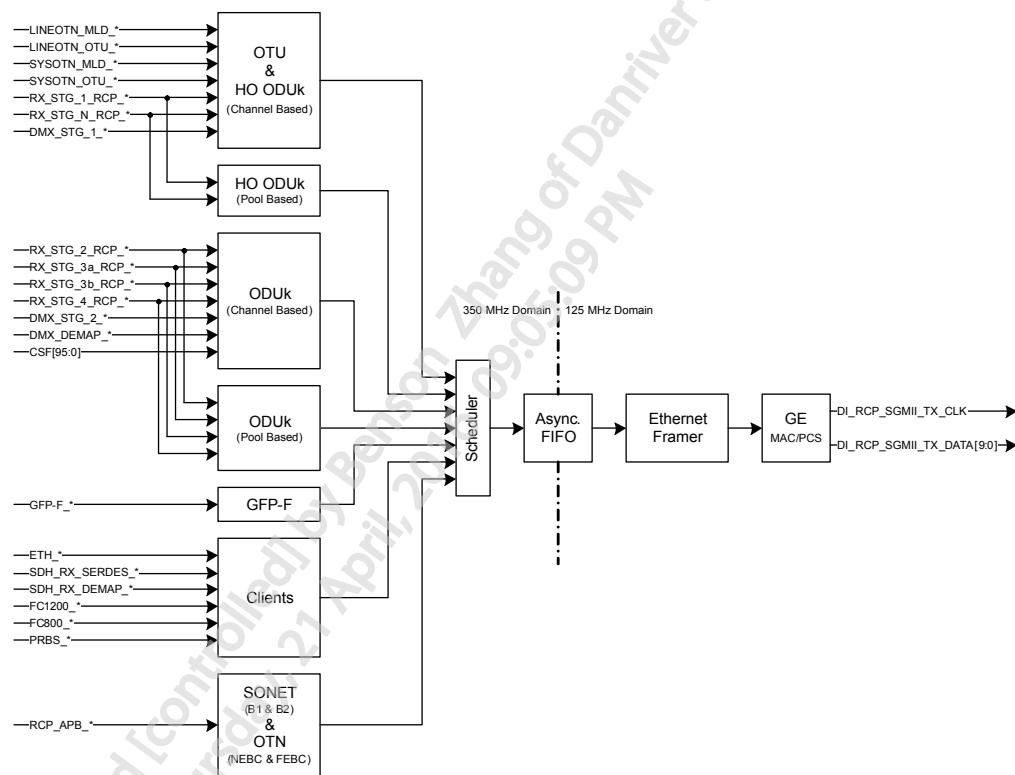
The GE Ethernet module consists of TX\_MAC, RX\_MAC, GE PCS layer and packet status monitoring blocks. The GE Ethernet module handles the actual Ethernet frames transmit and receive functions using PCS layer. The Ethernet module, in conjunction with the SERDES, supports a serial electrical interface to connect to industry standard Ethernet switch device.

- The GE PCS supports 8B/10B encoding/decoding as well as GE auto-negotiation.
- GE channel typically works on its own recovered clock in the receive direction and a common nominal transmit clock in the transmit direction.
- All 8B/10B line-code violations and running disparity errors are counted by the receive PCS block.

### 5.8.12.2 Defects Information Ring Control Port

The defects information module is aggregating defects from the Rx framer Stages 1, 2, 3a, 3b, 4, N, GFP-F mapper/demapper and clients interfaces (Ethernet system and line side, SDH Rx SERDES, SDH Rx demapper, PRBS, Fiber Channel 1200 and 800) to be transferred externally using a dedicated Gigabit Ethernet (SGMII) interface. The latched defects state is sent and cleared every 125us. The defects are mapped into different Ethernet PDUs (Protocol Data Unit) and identified at the Ethernet layer header using a different EtherType value for each source of defects. The SDH B1 & B2 & M0/M1 and OTN NEBC & FEBC error counts are also transferred every 5ms using the same interface. The primary goal is to enable fast protection switching decisions to be made by an external device (i.e. control CPU or FPGA).

Figure 64 DI\_RCP Top Level Block Diagram



### 5.8.12.2.1 Features

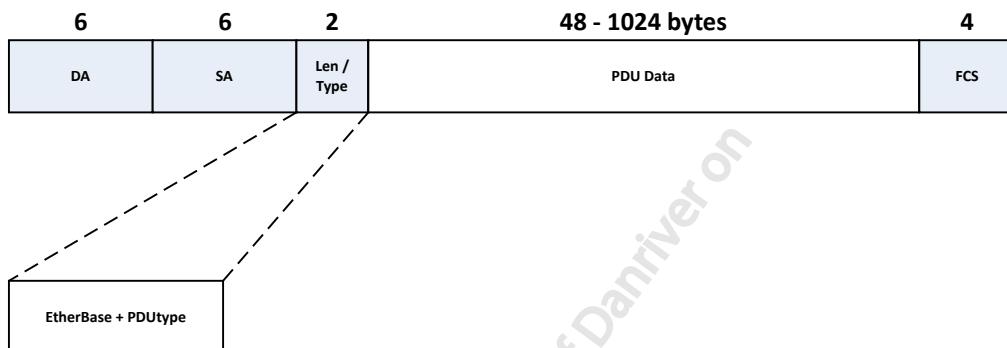
- Reports limited OTU, ODU (all stages), OPU and Client defects. The latched defects state is sent and cleared every 125us.
- The state of a defect is latched, i.e. a transient defect is recorded and the state is cleared after its transmission.
- Reports ODU (PM and TCM) NEBC and FEBC counts. The error counts are transferred and cleared every 5ms.
- Reports SONET/SDH B1, B2 and M0/M1 counts. The error counts are transferred every 5ms then cleared.
- The 5ms error counts reference clock is aligned to the edges of the 125us defects reference clocks.
- Transmit only interface. Received traffic is discarded.

- The defects modules fill reserved bits in the payload frame with zeroes.

### 5.8.12.2.2 Ethernet Frame Format

The DI\_RCP Ethernet frame format is shown in [Figure 65](#).

Figure 65 DI\_RCP Ethernet Frame Format



The Ethernet Framer module adds a common user programmable MAC SA/DA to all Ethernet packets. The different defect modules add the Ethernet Type header used for identification of the defect source. The Ethernet Type value is the sum of a user programmable EtherBase and a predefined PDUtype for each defect source.

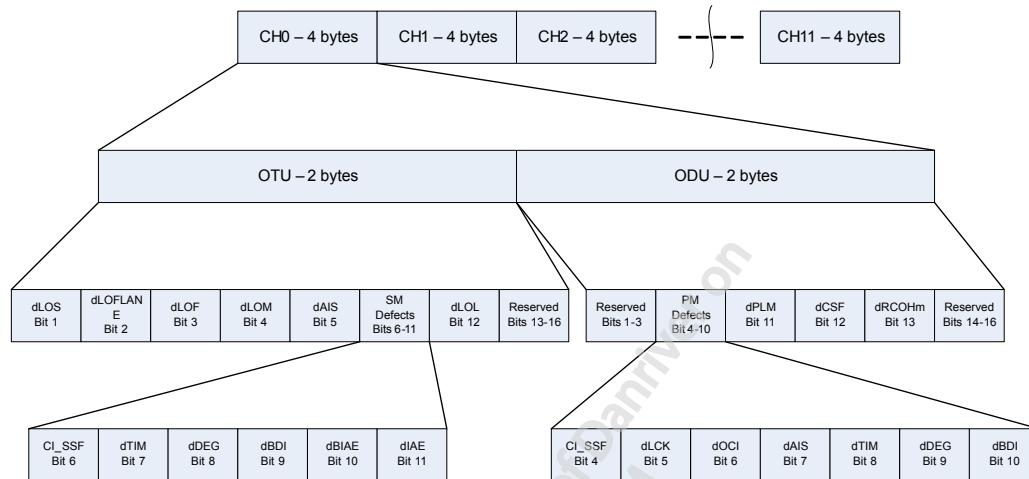
### 5.8.12.2.3 OTU and Channel-Based HO ODUk Defects

The OTU and channel based HO ODUk defects module aggregates defects from different sections of the DIGI device. The defects are framed in the PDU format shown in [Figure 66](#).

- Reports the following defects from OTN\_MLD: dLOS, dLOFLANE, dLOF, dLOM, dAIS, dLOL.
- Reports the following defects from OTU\_FRM: CI\_SSF, dTIM, dDEG, dBDI, dBIAE, dIAE.
- The OTN\_MLD & OTU\_FRM defects come from both the LINEOTN and SYSOTN subsystems.
- Reports the following defects from the HO Stage 1 and Stage N Rx framer: CI\_SSF, dLCK, dOCI, dAIS, dTIM, dDEG, dBDI.
- Reports the defects dPLM from Demux Stage 1 and Stage 2. The line side PDU uses the dPLM from demux stage 1. The system side PDU contains the dPLM from demux stage 2.
- Reports the dCSF defects from both the line and system side.
- Reports the defect dRCOHm for HO Stage 1 and Stage N. The defect state is configured through the register. This is only valid for channel 0.
- In PDU Type 1, dRCOHm for channel 1 and 2 are re-purposed to indicate dLOOFMI. Channel 1 dRCOHm is dLOOFMI bit from ODTU\_DMX Stage 1 (between framer stage 1 and 2). Channel 2 dRCOHm is dLOOFMI bit from ODTU\_DMX Stage 2 (between framer stage N and 3A in SYSOTN mode). Note that these bits are only valid during ODU4 operation. Note that the operation of transferring dLOOFMI from the DMX to the RCP is done via DIGI Firmware. Based on Firmware performance, the assertion and clear could be delayed by as much as 300µs beyond the 3 ms soak recommended by G.798.
- A common Ethernet PDU is used. The LINEOTN, HO Stage 1 and Demux Stage 1 defects are tagged with PDUtype equal to 1. The SYSOTN, HO Stage N and Demux Stage 2 defects are tagged with PDUtype equal to 2.
- All 12 defect channels from the LINEOTN and HO ODUk Stage 1 are active. Only channel 0 is active for SYSOTN and HO Stage N, the other channels are filled with zeroes.

- The payload structure is byte aligned and the payload size is 48 bytes.

Figure 66 OTU and Channel Based HO ODUk PDU

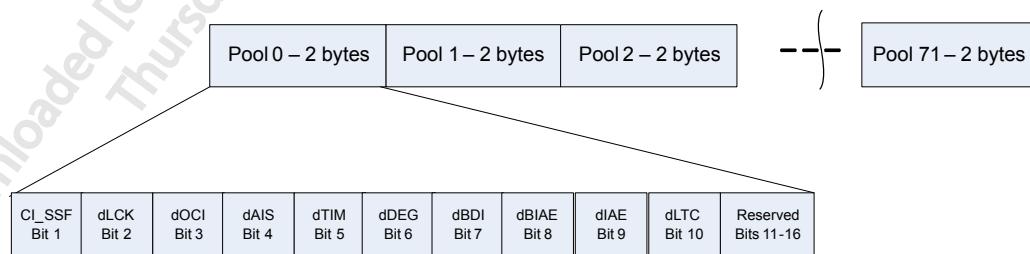


#### 5.8.12.2.3.1 Pool-Based HO ODUk Defects

The pool based HO ODUk defects module aggregates defects from Rx framer Stage 1 and Stage N. The defects are framed in the PDU format shown in [Figure 67](#).

- Reports the following defects from both Rx framer Stage 1 and Stage N: CI\_SSF, dLCK, dOCI, dAIS, dTIM, dDEG, dBDDI, dBIAE, dIAE, dLTC. Note that for PM-NIM layer dIAE, dBIAE and dLTC are invalid.
- A common PDU is used. The HO Stage 1 defects are tagged with PDUtype equal to 5. The HO Stage N defects are tagged with PDUtype equal to 6.
- The defects are pool based, i.e. a maximum of 72 pool entries can be selected to be transferred. The pool selection is made in the Rx framer stages. The defects module is using the pool number provided by the Rx framer stages as the address to store the defects (instead of the channel and layer number). Unused pool entries are filled with zeroes.
- The payload structure is byte aligned and the payload size is 144 bytes.

Figure 67 Pool-Based HO ODUk PDU

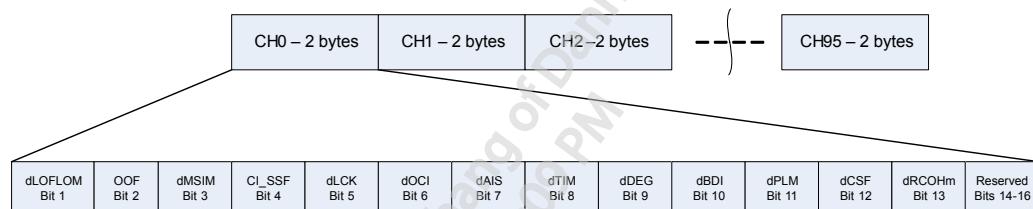


#### Channel-Based ODUk Defects

The channel based ODUk defects module aggregates defects from COREOTN subsystem. The defects are framed in the PDU format shown in [Figure 68](#).

- Reports the following defects from Rx framer stage 2, 3a, 3b and 4: dLOFLOM, OOF, dMSIM, CI\_SSF, dLCK, dOCI, dAIS, dTIM, dDEG, dBDI. Note that MSIM is not detected by stage 3b. Only the PM layer (word 0) is extracted from all 96 channels.
- A common PDU is used. The source of the defects is identified by using a different PDUtype. Stages 2/3a/3b/4 defects are tagged with a PDUtype equal to 9/10/11/12 respectively.
- The dCSF defect is carried but the source is different. For Rx framer stages 2 and 3a, it is from the dCSF vector. It is always zero for stage 3b. For stage 4, it is from the demapper.
- The dPLM defect is also carried but the source is different. For Rx framer stage 2, it is from the demux stage 2. It is always zero for stages 3a and 3b. Stage 4 is using the dPLM from the demapper.
- The dRCOHm defect for stage 2 is configured through the register. The defect status is cleared for stages 3a, 3b and 4.
- The payload structure is byte aligned and the payload size is 192 bytes.

Figure 68 Channel-Based ODUk PDU

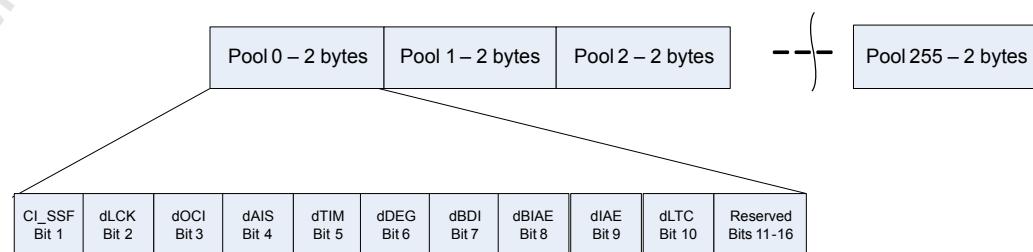


### Pool-Based ODUk Defects

The pool based ODUk defects module aggregates defects from COREOTN subsystem. The defects are framed in the PDU format shown in [Figure 69](#).

- Reports the following defects from Rx framer stage 2, 3a, 3b and 4: CI\_SSF, dLCK, dOCI, dAIS, dTIM, dDEG, dBDI, dBIAE, dIAE, dLTC. Note: for PM-NIM layer dIAE, dBIAE and dLTC are invalid.
- A common PDU is used. The source of the defects is identified by using a different PDUtype. Stages 2/3a/3b/4 defects are tagged with a PDUtype equal to 17/18/19/20 respectively.
- The defects are pool based, i.e. a maximum of 256 pool entries can be selected to be transferred. The pool selection is made in the Rx framer; the defects module is using the pool number provided by the Rx framer as the address to store the defects (instead of the channel and layer number).
- The pool can carry any layer with the exception of the PM layer which is transported in the channel based ODUk PDU.
- The payload structure is byte aligned and the payload size is 512 bytes.

Figure 69 Pool-Based ODUk PDU

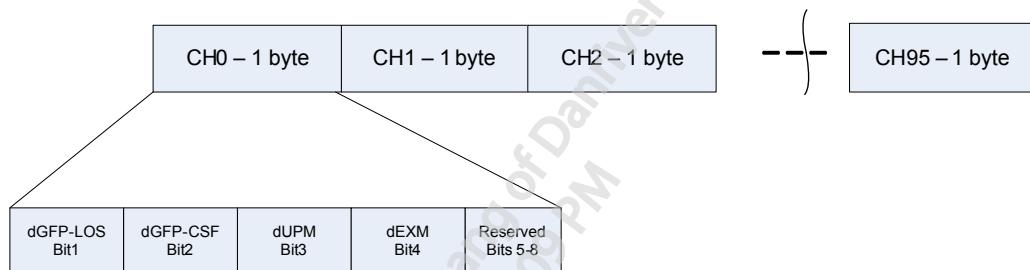


### GFP-F Mapper/Demapper Defects

The GFP-F mapper/demapper defects module accumulates defects coming from the GFP-F mapper/demapper block. The defects are framed in the PDU format shown in [Figure 70](#).

- Supports up to 96 channels.
- Defects reported: dGFP-LOS, dGFP-CSF, dUPM, dEXM.
- GFP-F payload is identified with PDUtype equal to 25.
- GFP-F payload structure is byte aligned and the payload size is 96 bytes
- dGFP-LOS corresponds to the dLFD defect from the GFP\_F demapper.
- dGFP-CSF is derived from dCSF-LOS (UPI=0x01) or dCSF-LOCS (UPI=0x02) or dCSF-FDI (UPI=0x04).

[Figure 70 GFP-F Mapper/Demapper PDU](#)

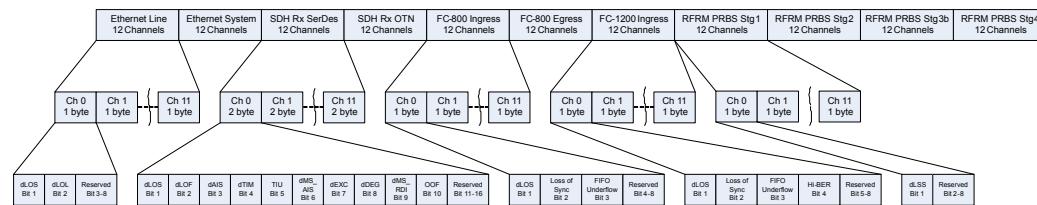


### Client Defects

The client defects module aggregates client defects from the Ethernet system and line side, SDH Rx SERDES, SDH Rx demapper, PRBS, Fiber Channel 1200 ingress, Fiber Channel 800 ingress and egress. The defects are grouped together into a single PDU as shown in [Figure 71](#).

- Reports the following defects from the Ethernet subsystem: dLOL, dLOS. All 12 Ethernet channels from both the line and system side are supported.
- Reports the following defects from the SDH subsystem: dLOS, dLOF, dAIS, dTIM, dMS\_AIS, dEXC, dDEG, dMS\_RDI, OOF, TIU. Twelve channels from both the Rx SERDES and Rx demapper from OTN direction are supported. On the Rx demapper from OTN direction, dLOS reports SSF (OTN Server Signal Fail). For 40G SONET, channels streams, {3, 7, 11} are used on the RCP defect reporting, not vs. {0, 4, 8}.
- Reports the following ingress defects from the FC1200 subsystem: dLOS, dLOSync, dHI-BER. Also reporting the dFIFO\_U\_Flow defect from FC1200 egress. All 12 channels are supported.
- Reports the following ingress and egress defects from the FC800 subsystem: dLOS, dLOSync, dFIFO\_U\_Flow. All 12 channels are supported.
- Reports the PRBS defect dLSS from the Rx framer stages 1, 2, 3b and 4. Twelve channels are supported.
- Client defects PDU is identified with PDUtype equal to 26.
- Client PDU structure is byte aligned and the payload size is 156 bytes.

Figure 71 Client Defects PDU



## SONET/SDH and OTN Performance Monitoring

The SONET/SDH and OTN performance monitoring module is used for the SONET/SDH B1 & B2 & M0/M1 and the OTN FEBC & NEBC error counts. Every 5ms the counts are transferred from different locations to the local memory using the DMA engine. The DMA transfers are configured by the embedded CPU and upon completion of the transfers it signals the performance monitoring module to start sending the counts over the DI\_RCP interface.

Different PDUs are needed to carry the different types of payload as shown in Figure 16 - 21. The source of the error counts is identified by using a different PDUtype. The different error counts, PDUtype, channel/pool number and payload size are shown in [Table 43](#).

**Table 43 SONET/SDH and OTN Performance Monitoring PDU Types**

Performance Monitoring PDU	PDUtype	Channel/Pool Number	Payload Size (bytes)
channel based HO ODUk Stage 1	3	12	48
channel based HO ODUk Stage N	4	1	48
pool based ODUk Stage 1	7	72	288
pool based ODUk Stage N	8	6	288
channel based ODUk Stage 2	13	96	384
channel based ODUk Stage 3a	14	96	384
channel based ODUk Stage 3b	15	96	384
channel based ODUk Stage 4	16	96	384
pool based ODUk Stage 2	21	256	1024
pool based ODUk Stage 3a	22	256	1024
pool based ODUk Stage 3b	23	256	1024
pool based ODUk Stage 4	24	256	1024
SONET/SDH	27	12	120
ODUK Stage 1/2/3b/4 PRBS	28	4	156

- The copied B1, B2, M0/M1, FEBC, NEBC and nTSE counts are cleared at the end of each transfer.
  - The error counts are byte aligned.
  - For the channel based HO ODUk Stage N performance monitoring counts, the PM layer of a single channel is supported. The channels 1-11 are all zeroes.
  - The 12 channels SONET/SDH performance monitoring can be configured to monitor a mix of ingress or egress channels. Each channel can be configured as ingress or egress direction independently. For example, all 12 channels could be defined as the ingress/mapping direction. They could also be configured as the egress/demapping direction, or a combination of both.

Figure 72 HO ODUk Stage 1 &amp; N Performance Monitoring PDU

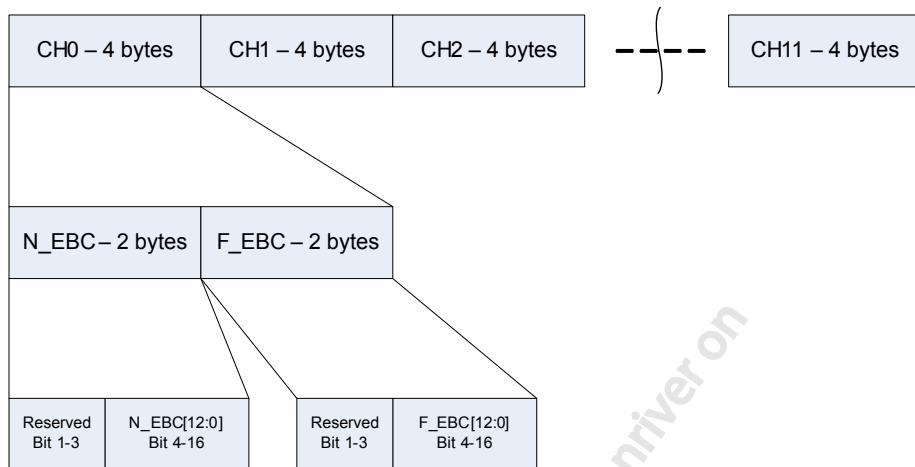


Figure 73 Pool-Based ODU Stage 1 &amp; N Performance Monitoring PDU

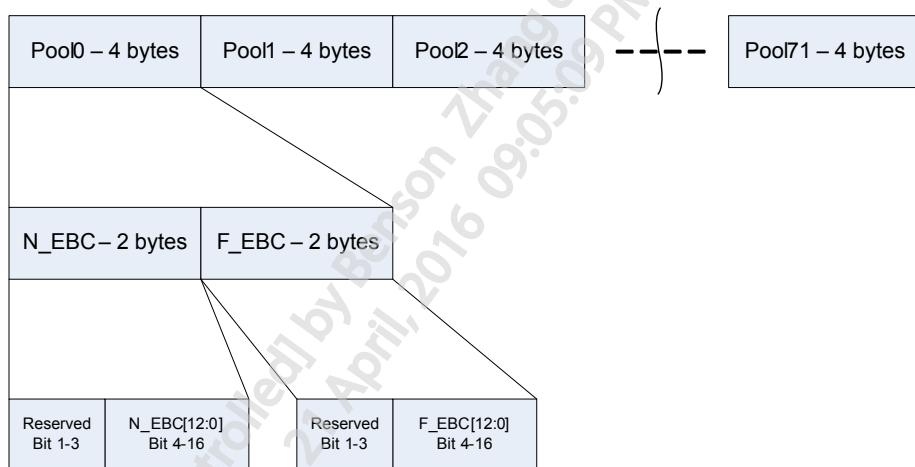


Figure 74 Channel Based ODU Stage 2/3a/3b/4 Performance Monitoring PDU

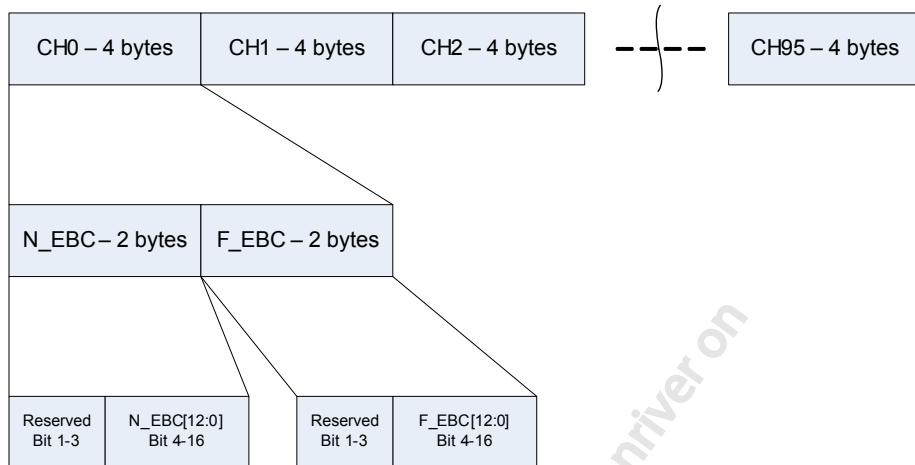


Figure 75 Pool Based ODU Stage 2/3a/3b/4 Performance Monitoring PDU

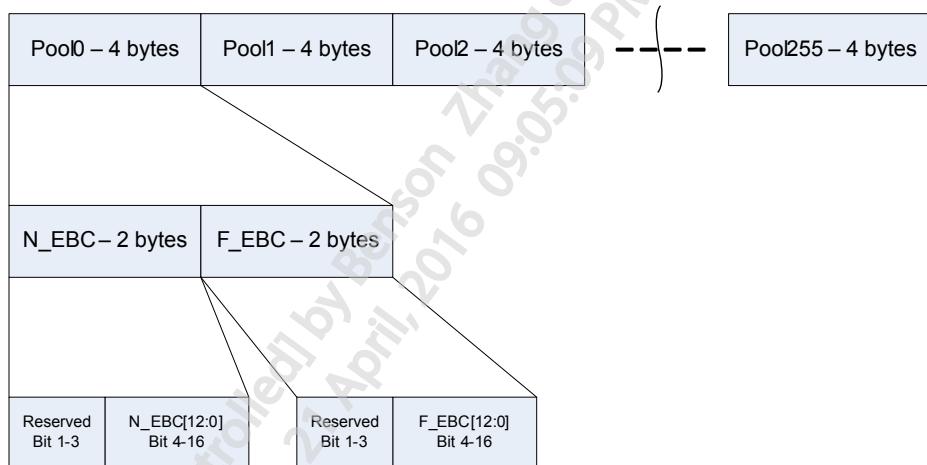


Figure 76 SONET/SDH Performance Monitoring PDU

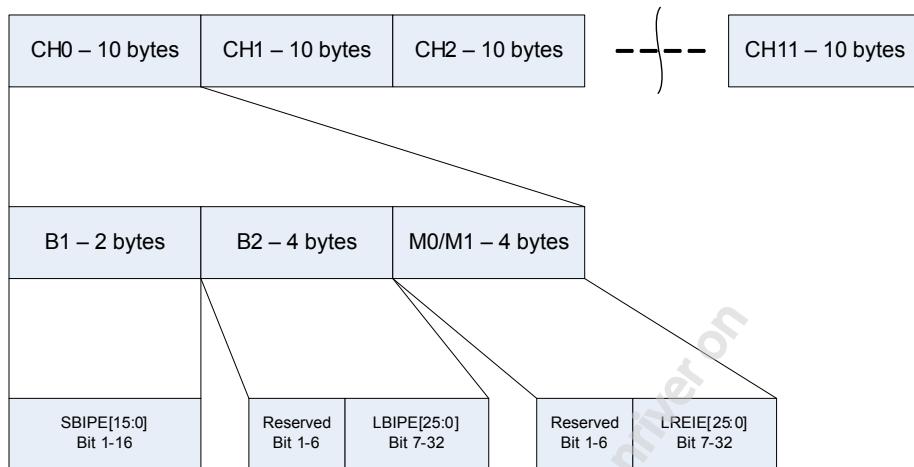
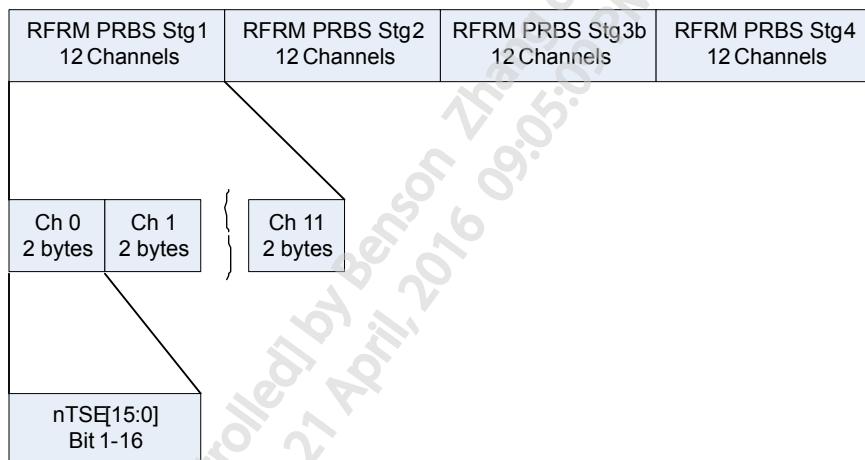


Figure 77 PRBS Performance Monitoring PDU



#### DI\_RCP Ethernet Bandwidth Calculation

The bandwidth calculations apply only to the Ethernet transmit direction. The Ethernet MAC is assumed to burst at Ethernet line rate and no shaping is done on the transmit traffic. The maximum bandwidth is 266 Mbps in the worst case scenario.

Table 44 DI\_RCP Ethernet Bandwidth Calculation - Defects

	Defects						Units
	OTU&HO ODUk	HO ODUk	Channel ODUk	Pool ODUk	GFP-F	Clients	
PDU Type	1,2	5,6	9,10,11,12	17,18,19,20	25	26	
Nb Sources	2	2	4	4	1	1	
PDU Length	48	144	192	512	96	156	bytes
Eth. Hdr. + Overhead	38	38	38	38	38	38	bytes

	Defects						Units
	OTU&HO ODUk	HO ODUk	Channel ODUk	Pool ODUk	GFP-F	Clients	
Ethernet Frame Length	86	182	230	550	134	194	bytes
Frame Period	1.25E-04	1.25E-04	1.25E-04	1.25E-04	1.25E-04	1.25E-04	seconds
BW/PDUtype	1.10E+07	2.33E+07	5.89E+07	1.41E+08	8.58E+06	1.24E+07	bits/sec.
Defects Total BW =	OUT & HO ODUk + HO ODUk + Channel ODUk + Pool ODUk + GFP-F + Clients						bits/sec.
	2.55E+08						bits/sec.

Table 45 DI\_RCP Ethernet Bandwidth Calculation - SONET/SDH B1 & B2 & M0/M1, OTN NEBC & FEBC Error Counts

	SONET/SDH B1 & B2 & M0/M1, OTN NEBC & FEBC Error Counts						Units
	Sonet/SDH	HO ODUk	Pool ODUk	Channel ODUk	Pool ODUk	PRBS	
PDU Type	27	3,4	7,8	13,14,15,16	21,22,23,24	28	
Nb Sources	1	2	2	4	4	1	
PDU Length	120	48	288	384	1024	96	bytes
Eth. Hdr. + Overhead	38	38	38	38	38	38	bytes
Ethernet Frame Length	158	86	326	422	1062	134	bytes
Frame Period	5.00E-03	5.00E-03	5.00E-03	5.00E-03	5.00E-03	5.00E-03	seconds
BW/PDUtype	2.53E+05	2.75E+05	1.04E+06	2.70E+06	6.80E+06	2.14E+05	bits/sec.
Error Counts Total BW =	SONET/SDH + HO ODUk + Pool ODUk + Channel ODUk + Pool ODUk + PRBS						bits/sec.
	1.13E+07						bits/sec.

- Total BW = Defects (Table 44) + Error Counts (Table 45): 266 Mbps

### 5.8.12.3 PMON Latch Clock

There are three PMON latch clocks (LCLK1, LCLK2 and LCLK3) inside COREOTN subsystem. LCLK is used to transfer the contents of PMON holding registers to shadow registers and clear counters for performance monitoring purposes.

The general clock cycle of these LCLKs can be in the following range:

- LCLK1 : 0 – 5 seconds (generally 1 second for hardware dDEG detect)
- LCLK2 : 0 – 10ms (generally 5ms range for DMA engine to transfer PMON error count to ODU\_RCP port) or asynchronous (triggered by software)
- LCLK3 : 125us (used only in ODU\_RCP for defect capturing and transfer)

LCLK1 is used by ODUk Rx framers to measure degradation over 1-5 seconds to assert dDEG defect. The dDEG defect is calculated for bursty mode as per G.798 standard specification. LCLK1 also captures NEBC and FEBC counts which are sized to saturate at a little over 1 second interval.

LCLK2 is used in the OTU Rx framer, ODUk Rx framers, SONET/SDH PMON and OTN-PRBS blocks to transfer PMON error counts to shadow registers. These error counts are sized exactly to capture 10ms worth of error counts and saturate. LCLK2 can also be selected in ODUk Rx framers to capture sticky defect and clear them.

LCLK3 is used in the ODU\_RCP block to capture sticky defects and to transfer to the ODU\_RCP interface every 125  $\mu$ s. If ODU\_RCP is enabled, user software will not be able to read these sticky defects through the PCIe interface because they are cleared by LCLK3 every 125  $\mu$ s.

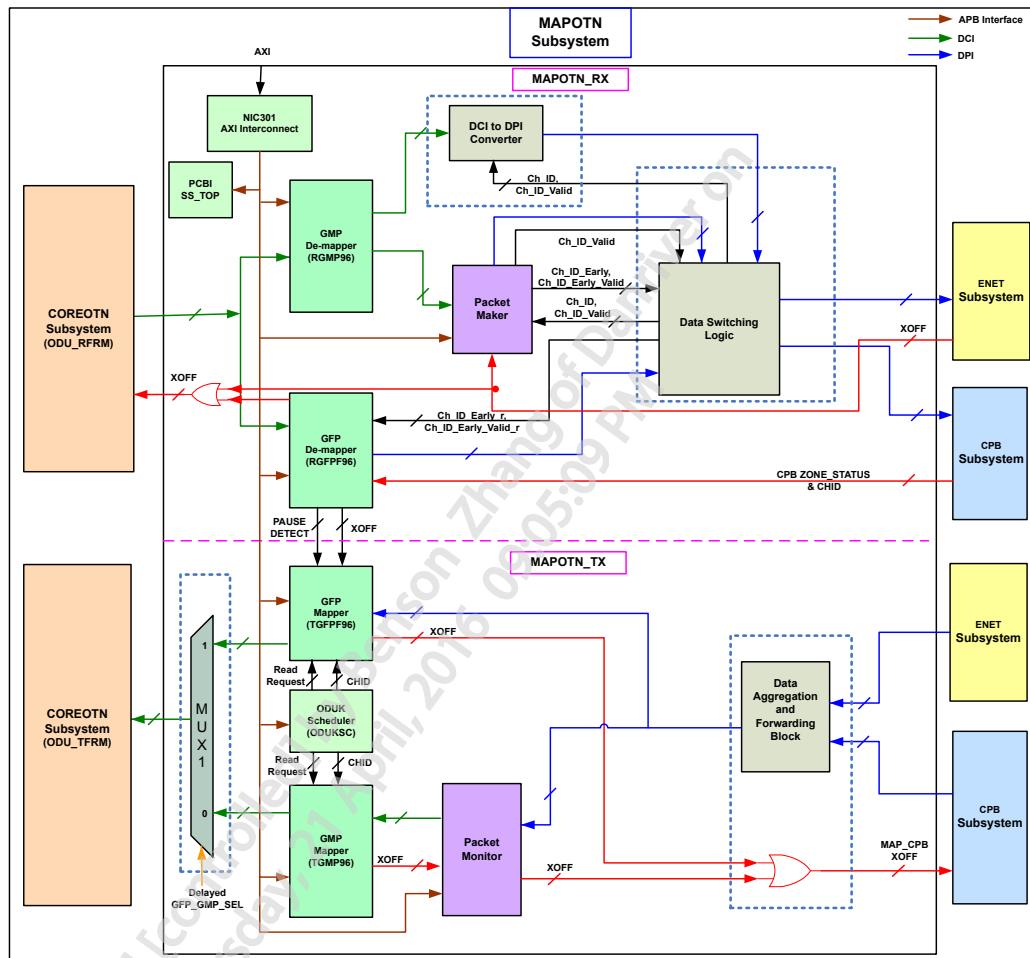
Configuration of the latch clocks is done by device firmware. Refer to the DIGI Device Family Programmer's Manual [\[4\]](#) for the supported LCLK use-cases for PMON collection, defect reporting, and the ring control port operation from a system software perspective.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:09 PM

## 5.9 OTN Mapper (MAPOTN) Subsystem

### 5.9.1 Block Diagram

Figure 78 MAPOTN Block Diagram



### 5.9.2 Overview

The MAPOTN is placed between the COREOTN, DCPB, and ENET(LINE) subsystems. Much of the terminology used in this section is defined in the subsections.

The MAPOTN is a mapping engine that maps various packet and CBR payloads into suitable ODUK containers for transport over OTN. The MAPOTN supports GFP-F, GMP, AMP and BMP mappings.

A packet client of any rate from the CPB can be GFP-F mapped and then placed into an ODUK container. The mapping into ODUK is based on the software programming of the ODUK scheduler.

The MAPOTN allows for the transport of a large number of protocols over an OTN network. 10GE, 40GE and 100GE Ethernet clients can be mapped into and extracted from OPUk streams using GFP or AMP/BMP/GMP mapping engines. The GMP engine also supports variable rate CBR client data mapped

into different OPUks and OPUflexs (CBR). The GMP mapper supports AMP, BMP or GMP mapping for any channel, based on user selection.

The MAPOTN is capable of processing up to 96 channels of packet and CBR traffic in both the transmit and receive direction. In addition to mapping and de-mapping, the MAPOTN performs filtering, performance monitoring, and scheduling of the packet and CBR traffic.

In the mapping direction, the CPB and ENET subsystems push data to the MAPOTN subsystem which is collected in DAFB ( Data Aggregation and Forwarding Block). This block multiplexes between the two sources before feeding the data to downstream mapping TSBs.

The client payload is inserted into the OPUk payload area. The mapping engines fill in all of the OPUk OH bytes, except for the PSI byte, where only the PT and CSF fields are filled. Only place holders are inserted for ODUk OH bytes except for the FAS and MFAS fields.

The GFP-F mapper can accept packet data from both the CPB and ENET subsystems, along with a corresponding channel ID. Along with the data, UPI field is also provided by the CPB and ENET subsystems. The GFP-F mapper inserts the PLI, cHEC, TYPE header, tHEC, optional Extension header, and optional Payload FCS with correct values and then maps the GFP-F frame into an OPUk payload area. If no client packets are available for transmission, GFP idle frames are transmitted between GFP client frames until the GFP mapper receives new packet data. The GFP-F mapper ensures a minimum of 1 GFP IDLE frame between each ordered set packet.

GMP mapper is the CBR mapping engine in MAPOTN. Inputs from DAFB is fed to a packet monitor, which removes the SOP/EOP delimiters, combines the packetized data, calculates the justification parameters and then feeds the packed CBR data to the GMP mapper. GMP mapper introduces stuff locations inside the OPUk container to accommodate for the rate differences between client signal and server container. Client mapping scheme is controlled by SW configuration.

In the de-mapping direction, the ODUk frames from the COREOTN ODUk Rx framer (ODU\_RFRM) are forwarded to the GFP or GMP de-mappers. Depending on the channel reset in the GFP and GMP de-mappers, the respective block consumes the data. In the (COREOTN) ODU\_RFRM to the CPB receive or the ENET(LINE) receive datapath, the de-mapping engines operate as a master and pushes the data towards the CPB or the ENET(LINE). The GFP and GMP de-mapper engines can accept data from the COREOTN ODU\_RFRM along with the channel number on every clock cycle. In the de-mapping direction, the engine de-maps the GFP or GMP frames from the OPU payload area. For the PSI byte overhead only, the PT and CSF fields are processed by the de-mapper engines.

The GFP-F de-mapper removes the ODU and OPU overhead bytes and extracts the GFP frame. It also removes the GFP encapsulation and then forwards the data to the CPB and/or the ENET(LINE) depending on per channel look-up via a DPI connection. Various checks like HEC and FCS are performed and the results are reported. The de-mapper optionally asserts a sideband "ERROR" signal for any FCS errors. Packets are dropped in case of any HEC errors in the GFP de-mapper and counted. Any received GFP IDLE frames are deleted by GFP de-mapper. The GFP de-mapper supports filtering of packets based on GFP header bytes in the incoming frame.

The GMP de-mapper removes the ODU and OPU overhead bytes, extracts the client CBR data from the incoming OPUk payload area and forwards the client CBR data to a DCI to DPI converter and/or packet maker depending on per channel look-up via the DCI. Data forwarded to the DCI to DPI converter goes to the ENET(LINE) subsystem (always 48-byte packets). The packet maker decides the size of the packets to be made, marks the SOP and EOP, and forwards the client data to the CPB via a DPI. The GMP de-mapper performs CRC checks and reports the status.

## 5.9.3 Features

### 5.9.3.1 Supported GFP and CBR Payload Mappings

Table 46 below describes the different types of supported GFP payload mappings.

Table 46 Supported GFP Mappings

Sr. No.	Client Data	Mapped Into	OPUk Mapping Scheme	Xcode Applicable	Reference to Standard	Comments
1	Any rate packet client (100K to 100G GFP-F mapped).	OPU0, OPU1, OPU1e, OPU1f, OPU2, OPU2e, OPU2f, OPU3, OPU3e2, OPU4, OPUflex (GFP)	GFP	No	G.709 Section 12.2.5 and 17.4 [28]	For the ODUflex (GFP) mapping, the OPUk containers will be generated at the rate specified by the user in the ODUK scheduler
2	10GE	Extended payload OPU2	GFP	Yes	G.7041 Section 7.9 [31], G.709 Section 17.4.1 [28]	The preamble and SFD are preserved along with the data for the mapping in the GFP frame. Ordered sets are mapped in separate GFP frames. This client mapping is defined in Gsup4 3 section 7.3 [27]
3	10GE	OPU2	GFP	No	G.7041 Section 7.1 [31], G.709 Section 17.4 [28]	This client mapping is defined in Gsup4 3 section 6.2 [27]
4	40GE	OPU3	GFP	No	G.7041 Section 7.1 [31], G.709 Section 17.4 [28]	40GE with GFP-F mapping into OTN should behave exactly the same as is described for 10GE in G.Sup43 clause 6.2.
5	100GE	OPU4	GFP	No	G.7041 Section 7.1 [31], G.709 Section 17.4 [28]	100GE with GFP-F mapping into OTN should behave exactly the same as is described for 10GE in G.Sup43 clause 6.2
6	FC-1200	OPU2e	GFP	Yes	G.709 Section 17.8.2 and Annex B [28]	Fixed Stuff, Note [1]

Table 47 below describes the different types of supported GMP payload mappings.

Table 47 Supported CBR (AMP/BMP/GMP) Mappings

Sr. No.	Client Data	Mapped Into	OPUk Mapping Scheme	Xcode Applicable	Reference to Standard	Comments
1	FC-800	OPU2	GMP	No	Non Standard ; use G709: 17.7.3 scheme [28]	
2	FC-800	OPUflex	BMP	No	G709: 17.9 [28].	Note[4]
3	FC-1200	OPU2f	BMP	No	Non Standard ; GSup43 7.1 [27]	Fixed Stuff, Note [1]
4	FC-1200	OPU1f	BMP	No	Non Standard ; GSup43 7.2 [27]	No fixed stuff columns
6	OC-192/STM-64	OPU2	AMP/BMP	No	G709: 17.2.2 [28]	Fixed Stuff, Note [1]
7	10GE	OPU1e	BMP	No	GSup43 7.2 [27]	No fixed stuff columns
8	10GE	OPU2e	BMP	No	GSup43 7.1 [27]	Fixed Stuff, Note [1]

Sr. No.	Client Data	Mapped Into	OPUk Map-ping Scheme	Xcode Ap-plicable	Reference to Standard	Comments
9	10G GDPS	OPU2e	GMP	No	Non Standard ; use G709: 17 .7.3 scheme [28]	Assuming the M, Pmserver values to be same as the nearest standard container, which is OPU2.
10	5G GDPS	OPU2	GMP	No	Non Standard ; use G709: 17 .7.3 scheme [28]	
11	4.9152G (CPRI- 5)	OPU2	GMP	No	Non Standard ; use G709: 17 .7.3 scheme, Annex D, App. VIII [28]	
12	9.8304G (CPRI -7)	OPU2	GMP	No	Non Standard ; use G709: 17 .7.3 scheme, Annex D, App. VIII [28]	
13	6.1440G (CPRI- 6)	OPU2	GMP	No	Non Standard ; use G709: 17 .7.3 scheme, Annex D, App. VIII [28]	
14	5G GDPS	OPUflex	BMP	No	G709: 17.9 [28]	Note[4]
15	4.9152G (CPRI- 5)	OPUflex	BMP	No	G709: 17.9 [28]	Note[4]
16	9.8304G (CPRI -7)	OPUflex	BMP	No	G709: 17.9 [28].	Note[4]
17	6.144G (CPRI- 6)	OPUflex	BMP	No	G709: 17.9 [28]	Note[4]
19	OC-768/ STM-256	OPU3	AMP/BMP	No	G709: 17.2.3 [28]	Fixed Stuff, Note [2]
21	40GE	OPU3	GMP	1027B transcoded	G709: 17.7.4.1, Annex B, App. VII, App. VIII [28]	
23	100GE	OPU4	GMP	No	G709: 17.7.5 [28]	Fixed Stuff, Note 4[3]
26	10G GDPS	OPUflex(CBR)	BMP	No	G709: 17.9 ,12.2.3 [28]	Note[4]
29	ODUkP_NULL	OPUk	GMP	No	G709: 12.2.4 [28]	

**Note:**

1. 16-byte fixed stuff column starting from byte 1905 in the ODUK frame.
2. 16-byte fixed stuff column appearing in two locations in the ODUK frame: bytes 1265 and 2545.
3. 8-byte fixed stuff column starting from byte 3817 in the ODUK frame.
4. Client replacement during signal fail , G709 17.9.1, is not in the scope of client mapper/de-mapper

Figure 79 MAPOTN Atomic Functions

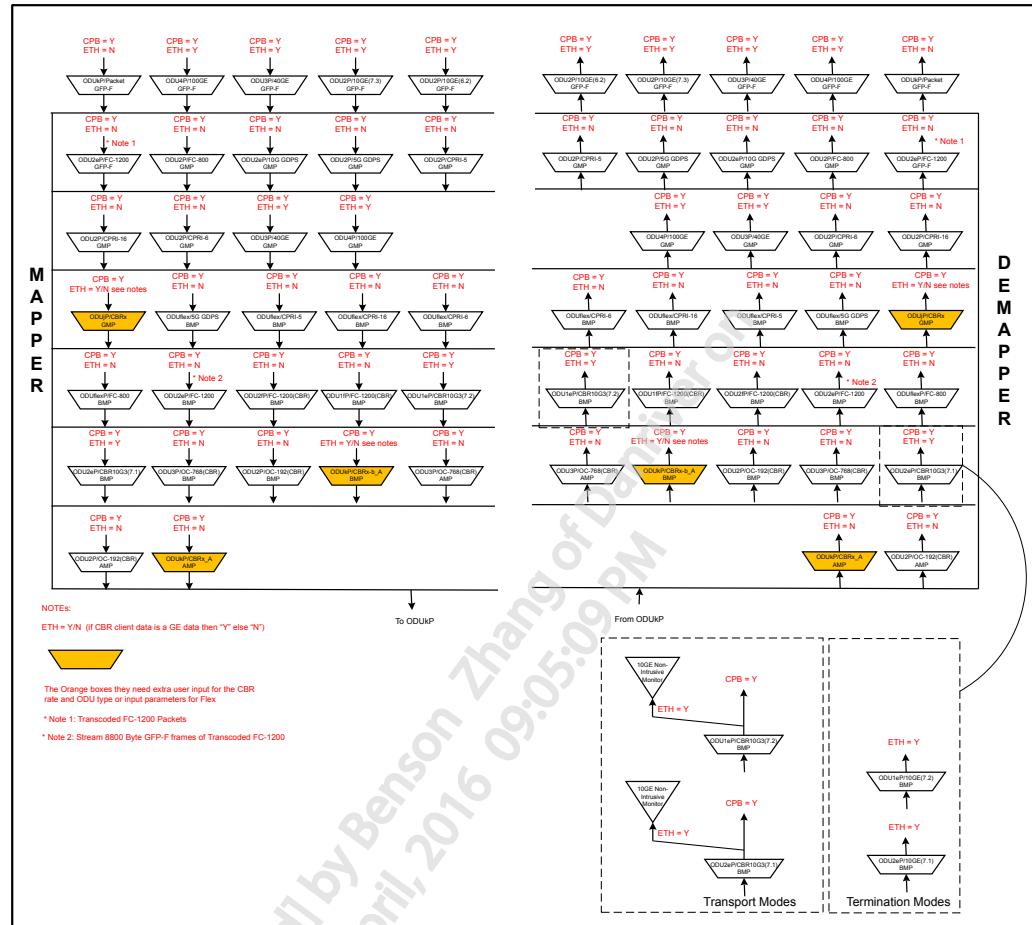


Figure 79 shows individual atomic functions for the mappings defined in Table 46 and Table 47.

### 5.9.3.2

### GFP Mapping and Interlaken Throughput Considerations for Packet Traffic

Packets transferred from the Interlaken or Ethernet interfaces (line or system) can be GFP-F mapped directly into an OPUk. The system throughput is a function of the number of bytes transferred over Interlaken, the throughput limit of the device interlaken processor, the bandwidth expansion as a result of bytes that are added to form the GFP frame, and the capacity of the OPUk.

Refer to Table 48 for the capacity of the Interlaken protocol as a function of the number of bytes transferred over the interface. The Bytes over Interlaken column defines the number of payload bytes. For example, the bytes over Interlaken column would define the size of an IP packet that is transferred over the Interlaken interface for GFP mapping. The number of bytes may optionally include header or trailer bytes, for example, a GFP extension header. The Interlaken Protocol Limit column defines the capacity of the Interlaken interface for typical lane configurations. The capacity of the Interlaken interface is a function of the BurstMax and BurstShort configuration of the device and includes overhead required for the 64/67 line coding, padding of bursts to a minimum size of BurstShort, padding of bursts to a multiple of 8B, and the meta frame efficiency. Examples provided are for a BurstMax of 256 and BurstShort of 32 bytes. The Interlaken processor of the device can achieve the throughput defined in the Interlaken processor limit column for a given packet size.

**Table 48** Interlaken Protocol and Device Interlaken Processor Throughput Versus Number of Bytes

Bytes over ILKN (B)	Interlaken Protocol Limit			ILKN Processor Limit (Gbps)
	24x6.25 or 12x12.5 (Gbps)	20x6.25 or 10x12.5 (Gbps)	16 x 10.3125 (Gbps)	
20	71.5	59.6	78.7	80.0
21	75.1	62.6	82.6	84.0
22	78.7	65.5	86.5	88.0
23	82.2	68.5	90.5	92.0
24	85.8	71.5	94.4	96.0
25	89.4	74.5	98.3	100.0
26	93.0	77.5	102.2	104.0
27	96.5	80.4	106.2	108.0
28	100.1	83.4	110.1	112.0
29	103.7	86.4	114.0	116.0
30	107.3	89.4	118.0	120.0
31	110.8	92.4	121.9	124.0
32	114.4	95.3	125.8	128.0
33	98.3	81.9	108.1	132.0
34	101.3	84.4	111.4	136.0
35	104.3	86.9	114.7	140.0
36	107.3	89.4	118.0	144.0
37	110.2	91.9	121.3	148.0
38	113.2	94.3	124.5	152.0
39	116.2	96.8	127.8	156.0
40	119.2	99.3	131.1	160.0
41	104.7	87.2	115.2	164.0
42	107.3	89.4	118.0	168.0
43	109.8	91.5	120.8	172.0
44	112.4	93.6	123.6	176.0
45	114.9	95.8	126.4	180.0
46	117.5	97.9	129.2	184.0
47	120.0	100.0	132.0	188.0
48	122.6	102.1	134.8	192.0
49	109.5	91.2	120.4	93.5
50	111.7	93.1	122.9	95.5
51	114.0	95.0	125.4	102.0
52	116.2	96.8	127.8	104.0
53	118.4	98.7	130.3	106.0
54	120.7	100.5	132.7	108.0
55	122.9	102.4	135.2	110.0
56	125.1	104.3	137.6	112.0
57	113.2	94.3	124.5	114.0
58	115.2	96.0	126.7	116.0
59	117.2	97.7	128.9	118.0
60	119.2	99.3	131.1	120.0
61	121.2	101.0	133.3	122.0

Interlaken Protocol Limit				
Bytes over ILKN (B)	24x6.25 or 12x12.5 (Gbps)	20x6.25 or 10x12.5 (Gbps)	16 x 10.3125 (Gbps)	ILKN Processor Limit (Gbps)
62	123.1	102.6	135.5	124.0
63	125.1	104.3	137.6	126.0
64	127.1	105.9	139.8	128.0
65	116.2	96.8	127.8	130.0
66	118.0	98.3	129.8	132.0
67	119.8	99.8	131.7	134.0
68	121.6	101.3	133.7	136.0
69	123.3	102.8	135.7	138.0
70	125.1	104.3	137.6	140.0
71	126.9	105.8	139.6	142.0
72	128.7	107.3	141.6	144.0
73	118.6	98.9	130.5	146.0
74	120.3	100.2	132.3	148.0
75	121.9	101.6	134.1	150.0
76	123.5	102.9	135.9	152.0
77	125.1	104.3	137.6	154.0
78	126.8	105.6	139.4	156.0
79	128.4	107.0	141.2	158.0
80	130.0	108.3	143.0	160.0
81	120.7	100.5	132.7	162.0
82	122.1	101.8	134.4	164.0
83	123.6	103.0	136.0	166.0
84	125.1	104.3	137.6	168.0
85	126.6	105.5	139.3	170.0
86	128.1	106.8	140.9	172.0
87	129.6	108.0	142.6	174.0
88	131.1	109.2	144.2	176.0
89	122.4	102.0	134.6	178.0
90	123.8	103.1	136.1	180.0
91	125.1	104.3	137.6	182.0
92	126.5	105.4	139.2	184.0
93	127.9	106.6	140.7	186.0
94	129.3	107.7	142.2	188.0
95	130.6	108.9	143.7	190.0
96	132.0	110.0	145.2	192.0
97	123.9	103.2	136.2	129.3
98	125.1	104.3	137.6	130.7
99	126.4	105.3	139.0	132.0
100	127.7	106.4	140.5	133.3
101	129.0	107.5	141.9	134.7
102	130.2	108.5	143.3	136.0
103	131.5	109.6	144.7	137.3
104	132.8	110.7	146.1	138.7

Interlaken Protocol Limit				
Bytes over ILKN (B)	24x6.25 or 12x12.5 (Gbps)	20x6.25 or 10x12.5 (Gbps)	16 x 10.3125 (Gbps)	ILKN Processor Limit (Gbps)
105	125.1	104.3	137.6	140.0
106	126.3	105.3	139.0	141.3
107	127.5	106.3	140.3	142.7
108	128.7	107.3	141.6	144.0
109	129.9	108.2	142.9	145.3
110	131.1	109.2	144.2	146.7
111	132.3	110.2	145.5	148.0
112	133.5	111.2	146.8	149.3
113	126.2	105.2	138.9	150.7
114	127.4	106.1	140.1	152.0
115	128.5	107.1	141.3	153.3
116	129.6	108.0	142.6	154.7
117	130.7	108.9	143.8	156.0
118	131.8	109.9	145.0	157.3
119	132.9	110.8	146.2	158.7
120	134.1	111.7	147.5	160.0
121	127.2	106.0	140.0	161.3
122	128.3	106.9	141.1	162.7
123	129.3	107.8	142.3	164.0
124	130.4	108.7	143.4	165.3
125	131.4	109.5	144.6	166.7
126	132.5	110.4	145.7	168.0
127	133.5	111.3	146.9	169.3
128	134.6	112.2	148.1	170.7
129	128.1	106.8	140.9	172.0
130	129.1	107.6	142.0	173.3
131	130.1	108.4	143.1	174.7
132	131.1	109.2	144.2	176.0
133	132.1	110.1	145.3	177.3
134	133.1	110.9	146.4	178.7
135	134.1	111.7	147.5	180.0
136	135.1	112.5	148.6	181.3
137	128.9	107.4	141.8	182.7
138	129.8	108.2	142.8	184.0
139	130.8	109.0	143.9	185.3
140	131.7	109.8	144.9	186.7
141	132.7	110.5	145.9	188.0
142	133.6	111.3	147.0	189.3
143	134.5	112.1	148.0	190.7
144	135.5	112.9	149.0	192.0
145	129.6	108.0	142.6	145.0
146	130.5	108.7	143.5	146.0
147	131.4	109.5	144.5	147.0

Interlaken Protocol Limit				
Bytes over ILKN (B)	24x6.25 or 12x12.5 (Gbps)	20x6.25 or 10x12.5 (Gbps)	16 x 10.3125 (Gbps)	ILKN Processor Limit (Gbps)
148	132.3	110.2	145.5	148.0
149	133.2	111.0	146.5	149.0
150	134.1	111.7	147.5	150.0
151	135.0	112.5	148.5	151.0
152	135.9	113.2	149.4	152.0
153	130.2	108.5	143.3	153.0
154	131.1	109.2	144.2	154.0
155	131.9	109.9	145.1	155.0
156	132.8	110.7	146.1	156.0
157	133.6	111.4	147.0	157.0
158	134.5	112.1	147.9	158.0
159	135.3	112.8	148.9	159.0
160	136.2	113.5	149.8	160.0
161	130.8	109.0	143.9	161.0
162	131.6	109.7	144.8	162.0
163	132.4	110.4	145.7	163.0
164	133.3	111.0	146.6	164.0
165	134.1	111.7	147.5	165.0
166	134.9	112.4	148.4	166.0
167	135.7	113.1	149.3	167.0
168	136.5	113.8	150.2	168.0
169	131.3	109.5	144.5	169.0
170	132.1	110.1	145.3	170.0
171	132.9	110.8	146.2	171.0
172	133.7	111.4	147.0	172.0
173	134.5	112.0	147.9	173.0
174	135.2	112.7	148.8	174.0
175	136.0	113.3	149.6	175.0
176	136.8	114.0	150.5	176.0
177	131.8	109.9	145.0	177.0
178	132.6	110.5	145.8	178.0
179	133.3	111.1	146.7	179.0
180	134.1	111.7	147.5	180.0
181	134.8	112.3	148.3	181.0
182	135.6	113.0	149.1	182.0
183	136.3	113.6	149.9	183.0
184	137.0	114.2	150.8	184.0
185	132.3	110.2	145.5	185.0
186	133.0	110.8	146.3	186.0
187	133.7	111.4	147.1	187.0
188	134.4	112.0	147.9	188.0
189	135.1	112.6	148.7	189.0
190	135.9	113.2	149.4	190.0

Interlaken Protocol Limit				
Bytes over ILKN (B)	24x6.25 or 12x12.5 (Gbps)	20x6.25 or 10x12.5 (Gbps)	16 x 10.3125 (Gbps)	ILKN Processor Limit (Gbps)
191	136.6	113.8	150.2	191.0
192	137.3	114.4	151.0	192.0
193	132.7	110.6	146.0	154.4
194	133.4	111.1	146.7	155.2
195	134.1	111.7	147.5	156.0
196	134.8	112.3	148.2	156.8
197	135.4	112.9	149.0	157.6
198	136.1	113.4	149.7	158.4
199	136.8	114.0	150.5	159.2
200	137.5	114.6	151.3	160.0
201	133.1	110.9	146.4	160.8
202	133.7	111.4	147.1	161.6
203	134.4	112.0	147.8	162.4
204	135.1	112.5	148.6	163.2
205	135.7	113.1	149.3	164.0
206	136.4	113.7	150.0	164.8
207	137.0	114.2	150.8	165.6
208	137.7	114.8	151.5	166.4
209	133.4	111.2	146.8	167.2
210	134.1	111.7	147.5	168.0
211	134.7	112.3	148.2	168.8
212	135.3	112.8	148.9	169.6
213	136.0	113.3	149.6	170.4
214	136.6	113.9	150.3	171.2
215	137.3	114.4	151.0	172.0
216	137.9	114.9	151.7	172.8
217	133.8	111.5	147.1	173.6
218	134.4	112.0	147.8	174.4
219	135.0	112.5	148.5	175.2
220	135.6	113.0	149.2	176.0
221	136.2	113.5	149.8	176.8
222	136.8	114.0	150.5	177.6
223	137.5	114.5	151.2	178.4
224	138.1	115.1	151.9	179.2
225	134.1	111.7	147.5	180.0
226	134.7	112.2	148.1	180.8
227	135.3	112.7	148.8	181.6
228	135.9	113.2	149.4	182.4
229	136.4	113.7	150.1	183.2
230	137.0	114.2	150.8	184.0
231	137.6	114.7	151.4	184.8
232	138.2	115.2	152.1	185.6
233	134.4	112.0	147.8	186.4

Interlaken Protocol Limit				
Bytes over ILKN (B)	24x6.25 or 12x12.5 (Gbps)	20x6.25 or 10x12.5 (Gbps)	16 x 10.3125 (Gbps)	ILKN Processor Limit (Gbps)
234	134.9	112.4	148.4	187.2
235	135.5	112.9	149.1	188.0
236	136.1	113.4	149.7	188.8
237	136.7	113.9	150.3	189.6
238	137.2	114.4	151.0	190.4
239	137.8	114.8	151.6	191.2
240	138.4	115.3	152.2	192.0
241	134.6	112.2	148.1	160.7
242	135.2	112.7	148.7	161.3
243	135.7	113.1	149.3	162.0
244	136.3	113.6	149.9	162.7
245	136.9	114.0	150.5	163.3
246	137.4	114.5	151.2	164.0
247	138.0	115.0	151.8	164.7
248	138.5	115.4	152.4	165.3
249	134.9	112.4	148.4	166.0
250	135.4	112.9	149.0	166.7
251	136.0	113.3	149.6	167.3
252	136.5	113.8	150.2	168.0
253	137.0	114.2	150.8	168.7
254	137.6	114.7	151.3	169.3
255	138.1	115.1	151.9	170.0
256	138.7	115.6	152.5	170.7
257	120.9	100.7	133.0	171.3
258	121.4	101.1	133.5	172.0
259	121.8	101.5	134.0	172.7
260	122.3	101.9	134.5	173.3
261	122.8	102.3	135.1	174.0
262	123.2	102.7	135.6	174.7
263	123.7	103.1	136.1	175.3
264	124.2	103.5	136.6	176.0
265	124.7	103.9	137.1	176.7
266	125.1	104.3	137.6	177.3
267	125.6	104.7	138.2	178.0
268	126.1	105.1	138.7	178.7
269	126.5	105.4	139.2	179.3
270	127.0	105.8	139.7	180.0
271	127.5	106.2	140.2	180.7
272	128.0	106.6	140.7	181.3
273	128.4	107.0	141.3	182.0
274	128.9	107.4	141.8	182.7
275	129.4	107.8	142.3	183.3
276	129.8	108.2	142.8	184.0

Interlaken Protocol Limit				
Bytes over ILKN (B)	24x6.25 or 12x12.5 (Gbps)	20x6.25 or 10x12.5 (Gbps)	16 x 10.3125 (Gbps)	ILKN Processor Limit (Gbps)
277	130.3	108.6	143.3	184.7
278	130.8	109.0	143.9	185.3
279	131.2	109.4	144.4	186.0
280	131.7	109.8	144.9	186.7
281	132.2	110.2	145.4	187.3
282	132.7	110.5	145.9	188.0
283	133.1	110.9	146.4	188.7
284	133.6	111.3	147.0	189.3
285	134.1	111.7	147.5	190.0
286	134.5	112.1	148.0	190.7
287	135.0	112.5	148.5	191.3
288	135.5	112.9	149.0	192.0
289	132.5	110.4	145.7	165.1
290	132.9	110.8	146.2	165.7
291	133.4	111.1	146.7	166.3
292	133.8	111.5	147.2	166.9
293	134.3	111.9	147.7	167.4
294	134.8	112.3	148.2	168.0
295	135.2	112.7	148.7	168.6
296	135.7	113.1	149.2	169.1
297	132.7	110.6	146.0	169.7
298	133.2	111.0	146.5	170.3
299	133.6	111.3	147.0	170.9
300	134.1	111.7	147.5	171.4
301	134.5	112.1	148.0	172.0
302	135.0	112.5	148.5	172.6
303	135.4	112.8	148.9	173.1
304	135.9	113.2	149.4	173.7
305	133.0	110.8	146.3	174.3
306	133.4	111.2	146.8	174.9
307	133.8	111.5	147.2	175.4
308	134.3	111.9	147.7	176.0
309	134.7	112.3	148.2	176.6
310	135.2	112.6	148.7	177.1
311	135.6	113.0	149.2	177.7
312	136.0	113.4	149.6	178.3
478	137.8	114.8	151.6	191.2
479	138.1	115.1	151.9	191.6
480	138.4	115.3	152.2	192.0
481	136.5	113.7	150.1	174.9
482	136.8	114.0	150.4	175.3
483	137.0	114.2	150.8	175.6
484	137.3	114.4	151.1	176.0

Interlaken Protocol Limit				
Bytes over ILKN (B)	24x6.25 or 12x12.5 (Gbps)	20x6.25 or 10x12.5 (Gbps)	16 x 10.3125 (Gbps)	ILKN Processor Limit (Gbps)
485	137.6	114.7	151.4	176.4
486	137.9	114.9	151.7	176.7
487	138.2	115.2	152.0	177.1
488	138.5	115.4	152.3	177.5
489	136.6	113.8	150.2	177.8
490	136.9	114.0	150.5	178.2
491	137.1	114.3	150.9	178.5
492	137.4	114.5	151.2	178.9
493	137.7	114.7	151.5	179.3
494	138.0	115.0	151.8	179.6
495	138.3	115.2	152.1	180.0
496	138.5	115.4	152.4	180.4
497	136.7	113.9	150.3	180.7
498	137.0	114.1	150.6	181.1
499	137.2	114.4	151.0	181.5
500	137.5	114.6	151.3	181.8
501	137.8	114.8	151.6	182.2
502	138.1	115.0	151.9	182.5
503	138.3	115.3	152.2	182.9
504	138.6	115.5	152.5	183.3
505	136.8	114.0	150.5	183.6
506	137.0	114.2	150.8	184.0
507	137.3	114.4	151.0	184.4
508	137.6	114.7	151.3	184.7
509	137.9	114.9	151.6	185.1
510	138.1	115.1	151.9	185.5
511	138.4	115.3	152.2	185.8
512	138.7	115.6	152.5	186.2
513	129.2	107.6	142.1	186.5
514	129.4	107.8	142.3	186.9
515	129.7	108.1	142.6	187.3
516	129.9	108.3	142.9	187.6
517	130.2	108.5	143.2	188.0
518	130.4	108.7	143.5	188.4
519	130.7	108.9	143.7	188.7
520	130.9	109.1	144.0	189.1
521	131.2	109.3	144.3	189.5
522	131.4	109.5	144.6	189.8
523	131.7	109.7	144.8	190.2
524	131.9	109.9	145.1	190.5
525	132.2	110.1	145.4	190.9
526	132.4	110.4	145.7	191.3
527	132.7	110.6	145.9	191.6

Interlaken Protocol Limit				
Bytes over ILKN (B)	24x6.25 or 12x12.5 (Gbps)	20x6.25 or 10x12.5 (Gbps)	16 x 10.3125 (Gbps)	ILKN Processor Limit (Gbps)
528	132.9	110.8	146.2	192.0
529	133.2	111.0	146.5	176.3
530	133.4	111.2	146.8	176.7

**Table 49** describes the total payload capacity of all channels at the mapper interface; an illustration that is useful when comparing mapper/OPUk bandwidth limitations against the aggregate bandwidth limitations of the Interlaken interface and Interlaken processor. The PMC device does not support the mapping of a single Interlaken channel into multiple OPUk flows.

**Table 49 OPUk Payload Capacity**

OPUk Configuration	Payload Capacity (Gbps)
1xOPU4	104.4
3xOPU3	120.5
12xOPU2	119.9
10xOPU2	99.953

Consider an example in which 20-byte packets received over a 12x12.5 Gbps Interlaken interface are GFP mapped into three OPU3 signals. Assume the GFP mapping adds a 4B core header, a 4B payload header with a null extension header, and a 4B FCS. From [Table 49](#), the aggregate payload capacity of 3xOPU3 is 120.5 Gbps. Only 20/32 of the OPUk payload capacity or 75.3 Gbps is usable due to the addition of the GFP headers and FCS when the 20-byte packets are GFP mapped. Similarly, the capacity of the device's Interlaken processor is 80 Gbps, of which 75.3 Gbps is needed. From [Table 48](#), the capacity of the Interlaken protocol is 71.5 Gbps. The Interlaken protocol does not have sufficient capacity to carry 20B packets at full throughput for this application. In this example, full throughput is limited to 71.5 Gbps for 20B packets received over Interlaken.

Bandwidth expansion will also occur when a continuous stream of 4-byte Ethernet ordered sets is generated and GFP-F mapped as per clause 17.4.1 of G.709 and clause 7.9 of G.7041 (the mapping formerly defined by GSUp43 clause 7.3). In this scenario, ordered sets are not received from the Interlaken Interface, they are generated by the ENET subsystem. The MAPOTN subsystem ensures a minimum of 1 idle frame is generated between each ordered set GFP encapsulated frame. The ENET subsystem limits the bandwidth expansion by generating ordered sets at a rate lower than the payload capacity. On average, the MAPOTN will insert 5-6 idle frames for every ordered set frame generated by the ENET subsystem.

### 5.9.3.3 BMP OPU3e2 Mapping From Interlaken

Typically, PDUs received from the Interlaken interface that are mapped into an OPU3e2 are first mapped into an OPU2e, as per clause 17.2.4 of G.709. 4xODU2e is then mapped into an OPU3e2. The mapping datapath is as follows:

IF\_ILKN(PIF)/Rx -> SIFD(Pkt)/Rx -> CPB(Pkt-OPU2e) -> ENET\_LINE(Pkt-OPU2e)/Tx -> MAPOTN(BMP OPU2e)/Tx -> COREOTN(ODU2e)/TxClient -> ODUkSW(ODU2e)/TxClient -> Mux2(ODU2e) -> COREOTN(ODU3e2)/TxLine -> LINEOTN(OTU3e2)/Tx -> IF\_L\_OTU3/Tx

The PCS is added by the ENET\_LINE block prior to the OPU2e BMP mapping, ensuring that a minimum inter packet gap is received. The PDUs are mapped at a nominal 10 GbE rate and the CPB will backpressure the Interlaken interface if the rate at which PDUs arrive exceeds the nominal 10 GbE rate.

#### 5.9.3.4 Baseline Performance for GFP Mapped Small Packets

The throughput of the device GFP mapper and demapper that can be achieved for a continuous stream of packets is a function of the following:

- The packet size
- The capacity of the OPUk
- The number of GFP overhead bytes added
- The GFP mapping and demapping throughput of the device

The GFP overhead typically consists of the core header, payload header with a null extension header, and the payload FCS for a total of 12 bytes. [Table 50](#) summarizes the packet sizes where full throughput cannot be achieved.

**Table 50** Baseline Small Packet Sustained Throughput for the GFP Mapper and Demapper

OPUk Configuration	Packet Size (Bytes)	GFP Frame Size (Bytes)	OPUk Capacity
OPU0	1 - 23	13 - 35	less than 100%
OPU0	24 - 36	36 - 48	100%
OPU0	37 - 49	49 - 61	81 - 99%
OPU0	50 - MTU	62 - (MTU+12)	100%
OPU1	1 - 23	13 - 35	less than 100%
OPU1	24 - 36	36 - 48	100%
OPU1	37 - 49	49 - 61	81-99%
OPU1	50 - 59	62 - 71	98 - 99%
OPU1	60 - MTU	72 - (MTU+12)	100%
OPU2	1 - 23	13 - 35	less than 100%
OPU2	24 - 36	36 - 48	100%
OPU2	37 - 49	49 - 61	81 - 99%
OPU2	50 - MTU	62 - (MTU + 12)	100%
OPU3	1 - 23	13 - 35	less than 100%
OPU3	24 - 36	36 - 48	100%
OPU3	37 - 49	49 - 61	81 - 99%
OPU3	50 - MTU	62 - (MTU + 12)	100%
OPU4	1 - 23	13 - 35	less than 100%
OPU4	24 - 36	36 - 48	100%
OPU4	37 - 49	49 - 61	81 - 99%
OPU4	50 - MTU	62 - (MTU + 12)	100%

Packets are dropped when maximum sustained throughput is exceeded. Packets are dropped and counted (UNPACK\_DROP\_FRM\_CNT) by the receive GFP demapper block (RGFP) in the GFP demapping direction. Packets are dropped and counted (CH\_DROP\_CNT\_VAL) by the central packet buffer (CPB) in the GFP mapping direction.

## 5.9.4 Top-level Interface Support Blocks

The top-level interface support blocks are shown within a dotted line box in [Figure 78](#).

### 5.9.4.1 Mapping Direction Interface

The MAPOTN has two DPI interfaces, one driven by the CPB subsystem and the other driven by the ENET(LINE) subsystem. On the DPI bus fed to the MAPOTN, the CPB's egress scheduler contains a 96-entry calendar with:

- Entries that deliver data directly to the MAPOTN using channel numbers in the range of [95:0], which corresponds to the ODUk channel number into which client streams are inserted.
- Entries that deliver data indirectly to the MAPOTN via the ENET subsystem. Configuration guarantees that ENET and CPB uses different channel numbers.

As the CPB and ENET subsystem can push in data in the same cycle, it is required to provide a multiplexing scheme between the two. If CPB pushes valid data in the current cycle, it is forwarded to mapping TSBs. Incoming ENET data, provided it is valid, is collected in a communal FIFO and is sent to mappers only if there is no CPB scheduling in the same cycle. If there is no valid data from either of these sources for a cycle, DPI cycle valid indications are forced to zero on inputs to mapping TSBs.

Data to the COREOTN ODUk TX framer (ODU\_TFRM) is transmitted in a TDM format from the GFP or GMP mappers based on the channel scheduling by ODUkSC. The channel ID transmitted to the COREOTN ODU\_TFRM is the same as what is received from the CPB or the ENET(LINE) subsystems.

### 5.9.4.2 De-mapping Direction Interface

In the de-mapping direction, data from the COREOTN ODUk Rx framer (ODU\_RFRM) is fed to the GMP and GFP de-mappers.

The GMP de-mapper accepts the data from the COREOTN ODU\_RFRM interface depending on its per channel reset configuration. It extracts the OPUk payload data bytes and then feeds them to the packet maker and then to the CPB subsystem or to the DCI to DPI converter and then to the ENET subsystem.

The packet maker receives the data from the GMP de-mapper that is to be fed to the CPB. It schedules the channel based on 96 calendar entries. The 'Data Switching Logic' switches the data using the channel ID generated by the packet maker. The data read from the GFP de-mapper, packet maker and DCI to DPI converter is then multiplexed and fed to the CPB and/or ENET(LINE) subsystems via a DPI connection based on user programming.

The channel ID transmitted to the CPB or the ENET(LINE) subsystem is the same as received from the COREOTN ODU\_RFRM.

## 5.9.5 ODUk Scheduler (ODUkSC)

### 5.9.5.1 Overview

The MAPOTN ODUkSC (ODUk scheduler) is used to schedule traffic in the mapping direction from the CPB/ENET subsystems to the ODU\_TFRM. The scheduler maintains a calendar with a list of 96 entries, which are to be programmed by software to determine, for a given calendar entry, which data is output for which ODUk container. The primary purpose of the ODUkSC is to schedule nominal rate ODUk container rates for GMP/AMP/BMP and GFP-mapped clients.

To support the different traffic rates for different types of ODUk containers, a credit generation mechanism is implemented to decide the scheduling. Data is transferred in a particular time-slot only if the time slot has a credit associated with it. The credit generator uses a modulo M/N counter that can effectively generate any  $M/N * 1.39968$  Gbps rate, where M and N are integer values programmed through software. Different ODUk containers have different number of entries in the calendar and each entry has software-programmed M and N values for the container.

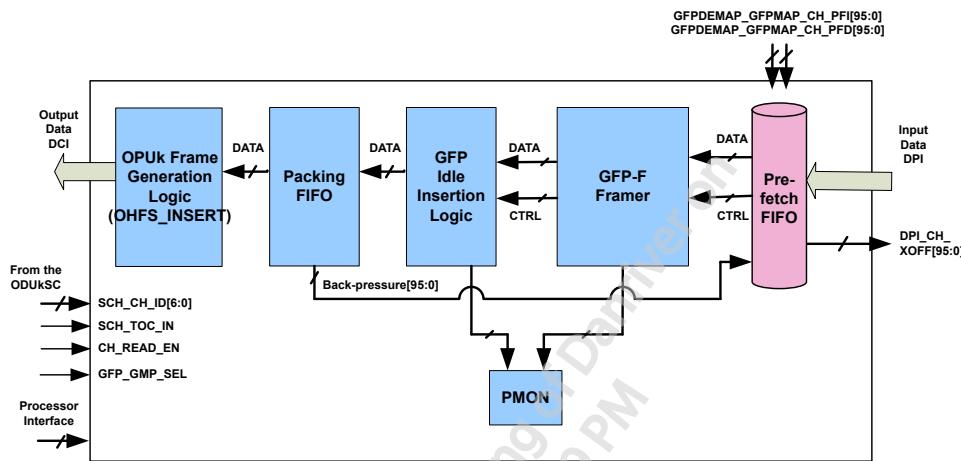
### 5.9.5.2 Features

- Maintains 96 channel entry software programmable calendars.
- Capability to schedule any channel with any rate range ODUflex (GFP) @ 5.21421 Mbps (1/228 \* 1.39969 Gbps, M=1 and N=228) to OPU4 with the accuracy of +/- 16 ppm to support AMP/BMP/GMP. 28-bit counters are implemented for M and N. For different values of M and N the large granularity of rates can be achieved.
- Rate is adjusted through software-programmable values of M and N on a per channel basis.
- Supports ODUk streams with rate controlled by the client. For example, for the FC-1200 traffic mapping via BMP, credits are generated at higher rate. If the data FIFO is found empty, the channel is not scheduled and thus scheduling follows the client data rate.
- In GFP mode of operation, even if the data FIFO is empty, the channel is scheduled and the GFP engine will insert GFP IDLES.
- Generates the credits based on user programmable rates to schedule requests.
- Provides a software-programmable rate change mechanism. Rate changes are done in a finite amount of user programmable time after software programs new M and N values by ramping-up or ramping-down the current rate to desired rate.

## 5.9.6 GFP-F Mapper

### 5.9.6.1 Block Diagram

Figure 80 GFP-F Mapper Block Diagram



### 5.9.6.2 Overview

The GFP-F mapper provides a generic mechanism for mapping any rate of channelized packet client traffic to the appropriate OPUk container. Data from the CPB or the ENET(LINE) subsystems is multiplexed and received at the input interface of the GFP-F mapper.

Input data is stored in the Prefetch FIFO. The Prefetch FIFO provides the back-pressure to the input when the data incoming rate is more than data outgoing rate and the Prefetch FIFO for a channel is about to become full. If PAUSE or CMF frame insertion is required, the Prefetch FIFO is not read, otherwise whenever there is data in the Prefetch FIFO and the downstream Packing FIFO is not backpressuring, it is read out.

After the Prefetch FIFO data is read, it is passed to the GFP-F framer, which performs GFP header insertion, payload scrambling (optionally, if enabled), FCS computation (optionally) thus creating a GFP-F frame. The GFP-F framer output is fed to the GFP IDLE insertion logic, which inserts GFP IDLE frames at frame boundaries to rate adapt to the server ODUK container rate when insufficient incoming data is received.

The GFP frames output by the framer are processed by packing logic which packs the variable length frame words into words of fixed length (384 bits) and stores them in a Packing FIFO.

When the data is requested by the ODUkSC (ODUk scheduler), data from the Packing FIFO is read and the ODUK frames are constructed as per the selected mapping. The OPUk OH columns are filled with appropriate values by the OHFS\_INSERT block and an ODUK frame is output on a DCI bus.

The PMON maintains counters for performance statistics.

### 5.9.6.3 Features

- Accepts variable length frames on a DPI slave interface operating at up to 500MHz, receiving data on up to 96 channels, each channel occupying from 1 (ODU0) to 80 (ODU4) time-slots. Channels may be assigned to time-slots in any arrangement, bursty or evenly spaced.
- Outputs fixed 48-octet words on a DCI interface operating at up to 349.92MHz according to an external calendar schedule input. The calendar may contain up to 96 channels, each channel occupying from 1 (ODU0) to 80 (ODU4) time-slots. Channels may be assigned to time-slots in any arrangement, bursty or evenly spaced and need not be related to the calendar used by the DPI interface.
- Channel bandwidth and time-slot allocation may be changed on-the-fly for ODUflex applications.
- Encapsulates any rate channelized client data according to the frame-mapped GFP protocol as described in ITU-T G.7041/Y.1303 [31]
- Performs rate adaptation via GFP Idle frame insertion. The number of GFP Idle Frames inserted is based on the incoming client packet rate and the outgoing server ODUk container rate.
- Performs insertion of GFP-F Core Header and Type Header into client frames. Space must be reserved for these headers in the form of SOP Hole bytes on the incoming DPI interface.
- Accepts out of band PLI[15:0] field on the DPI, to insert into GFP-F frames.
- Optionally inserts UPI from a configuration register or from the DPI\_TYPE[15:8] sideband bus.
- Provides optional GFP-F Extension header insertion into the client data frames. Supports insertion of 0-60B from internal registers or 4B from incoming inband signal.
- Optionally applies DC-balancing Barker-like code on the core header.
- Provides the option to scramble ( $x^{43} + 1$ ) the GFP payload area.
- Provides optional payload frame check sequence (pFCS) generation for client data using CRC-32 ISO-13239. Space must be reserved for the FCS in the form of EOP Hole bytes on the incoming DPI interface.
- Provides an option to calculate and insert Ethernet FCS (eFCS) for client data using CRC-32 ISO-13239. Only one of either pFCS or eFCS can be generated. It is recommended that at least one option be enabled unless an FCS has been appended to the frame by upstream logic. Note that the PFI bit in the GFP-F type header is only set to 1 if the option to generate pFCS is selected.
- Supports optional detection of Ethernet Ordered Set frames by recognizing a user-programmable UPI value. If FCS generation is enabled, the EOP hole bytes provided on the incoming DPI interface for FCS are removed for Ordered Set frames and no FCS is appended. Note that because the UPI setting for ordered sets is global, the device cannot simultaneously support current and legacy UPI codes for this feature. eFCS insertion in the GFP mapper in Gsup43 7.3 mode is not typically required since the upstream MAC is generally responsible for its insertion, so this limitation may not apply.
- Supports insertion of firmware formatted GFP Client Management Frames (CMF) / Management Communications frames (MCF) of a size up to 128 bytes per channel into the GFP-F framed traffic. The insertion priority between management and PAUSE frames can be set per-channel. Insertion priority between client data and pause/management frames can also be set per channel.
- Provides insertion of Ethernet PAUSE frames (IEEE 802.3 Annex 31B) into GFP-F framed traffic for the channels carrying CPB traffic, triggered by either GFP de-mapper or firmware. Pause frames have a fixed length of 72 bytes (including all headers and FCS field).
- Provides per-channel interrupts indicating that the transmission of inserted PAUSE and CMF/MCF frames has completed.
- Performs optional FCS corruption (inversion) when an errored frame (indicated by missing SOP, missing EOP or DPI\_ERR assertion) is received on the DPI interface.

- Provides per-channel performance counters for GFP client data frames, GFP client data bytes, GFP idle frames, PAUSE frames, CMF/MCFs, and errored frames transmitted in the output stream.
- Maintains per-channel counts of client data frames having a length greater or less than a user-programmed value.
- Performance count values are transferred from active to shadow memory prior to being read by software. Memory transfers may be initiated by software trigger or external clock.
- Generates channel-based XOFF requests to the upstream CPB based on Prefetch FIFO fill levels.
- Maps incoming client data into the OPUk payload area of an ODUk frame.
- Supports mapping of input 10GE LAN information transcoded using ITU-T G.sup43 section 7.3 [27] (now G.7041 section 7.9 [31]) into an extended OPU2 payload (G.709 section 17.4.1. [28]). Note the first 10 GE packet received after initial provisioning of a G.7041 section 7.9 (GSup43 section 7.3) datapath will be dropped.
- Supports ITU-T G.sup43 section 6.2 [27] mapping into an OPU2 payload.
- Supports appending of a minimum 1 GFP-Idle frame between two Ethernet ordered set frames. Optionally can append 4 GFP-Idle frames between ordered sets for GSup43 7.3 mapping.
- Supports GFP-F mapping of 40GE clients into OPU3 and 100GE clients into OPU4.
- Inserts FAS and MFAS into the ODUk frame.
- Inserts PSI byte with PT field as provided by software configuration.
- Supports software setting/clearing of the CSF bit in the PSI byte. No support is provided, however, for automatic setting of this bit by hardware in response to detected error conditions.
- Supports software setting/clearing of the NCS bit in the ODUflex container.
- Reserved (RES) fields within overhead bytes are set to 0 except those within the Resize Overhead Configuration (RCOH) which are programmable by the user.
- Supports the minimum GFP-F control frame size of 4 bytes and the full range of client frame lengths (9 to 65535 bytes). The maximum frame size which can be accepted from the ENET subsystem is 16384 bytes.

#### 5.9.6.4 Prefetch FIFO

The Prefetch FIFO stores the incoming frames from the CPB or ENET subsystems. When PAUSE/CMF frame insertion is enabled by software, reading of the data from the Prefetch FIFO is stopped at the frame boundaries while the PAUSE/CMF frame is inserted. For channels mapped from the CPB subsystem, the Prefetch FIFO generates per-channel back pressure based on the FIFO occupancy if the incoming data rate is greater than the outgoing data rate.

If a PAUSE frame is received by the receive path GFP de-mapper, an indication is sent to the GFP-F mapper to stop transmitting frames. The GFP de-mapper may also request the GFP mapper to transmit pause frames at regular intervals if it detects that the downstream receive FIFO is becoming full.

##### 5.9.6.4.1 PAUSE Frame Insertion

Four buffers are used to hold the user-defined Ethernet PAUSE frames. These buffers are shared between all channels. Pause frames are transmitted either by firmware or automatically by request of the GFP de-mapper.

For each GFP channel, the user selects one of the four user-programmed PAUSE frames to be sent as an XOFF request and another to be used as the XON request. Each PAUSE frame template also has an associated retransmission interval. The retransmission interval and the pause quanta value in XOFF templates are set in units of 512ns. The GFP mapper automatically sends the XON PAUSE frame if the GFP de-mapper XOFF request is de-asserted before the PAUSE timer expiration, allowing the remote

device to quickly resuming data transmission. If the PAUSE timer expires, another PAUSE XOFF frame is sent at frame boundary.

Note: XOFF/XON PAUSE can be defined as:

- XOFF PAUSE: requests the far-end to stop data transmission for a specified interval.
- XON PAUSE: requests the far-end to resume data transmission (by specifying a pause interval of 0).

The PAUSE frames stored in the buffers must be fully formatted and delineated by firmware. They must include all GFP-F headers (including core header, scrambled with Barker Code if required) and a properly calculated FCS field. No framing operations are performed on PAUSE frames by the GFP framer other than payload scrambling. The length of PAUSE frames is fixed at 72 bytes, including all headers and the FCS field.

#### 5.9.6.4.2 Management Frame Insertion

For management frame insertion, 128 bytes of storage is allocated per channel for either a:

- Client Management Frame (CMF): The CMF is used to indicate a client signal fail (CSF), forward defect indication (FDI), reverse defect indication (RDI) and defect clear indication (DCI).
- Management Communication Frame (MCF): The MCF provides an in-band management communications channel capability, but not yet defined by the G.7041 standard.

The insertion of the MCF/CMF frames is initiated by firmware. It is the responsibility of firmware to transmit all CMF frames required in response to alarms detected elsewhere in the system.

The management frames stored in the per-channel management frame memory must be fully formatted and delineated by firmware. They must include all GFP-F headers (including core header, scrambled with Barker Code if required) and a properly calculated FCS field. No framing operations are performed on management frames by the GFP framer other than payload scrambling.

#### 5.9.6.4.3 De-mapper-based XOFF to CPB

Whenever a PAUSE indication is received from the GFP de-mapper, the prefetch FIFO stops reading out client data traffic at the next frame boundary and only IDLE, PAUSE, CMF or MCF frames can be sent until the PAUSE indication is de-asserted. The prefetch FIFO will consequently fill up as frames continue to be received and back-pressure will be applied toward the CPB subsystem when the FIFO fill level reaches the XOFF threshold.

#### 5.9.6.5 GFP-F Framer

The GFP-F framer is a time-sliced processor encapsulating channelized client data into GFP-F frames. The client data is read from the prefetch FIFO and passed to the GFP-F framer. The GFP-F framer requires the client data frame to have correctly sized holes for the GFP-F core header, Type header, (optional) Extension header and (optional) pFCS/eFCS field and a correct PLI value in the first 2 octets which takes account of the holes. The framer then fills the holes with the correct values. The client data frame is checked for protocol errors (incorrect SOPs/EOPs) and for length errors (SOP/EOP position do not match PLI) while calculating the FCS. If an error is detected, the framer can (optionally) apply a corrupted (inverted) FCS to the frame.

The main features of the framer are:

- Calculates cHEC for the core header and optionally applies Barker code DC scrambling to the same.

- Constructs a Type Header based on per-channel register settings with the option to set the UPI field to a value received from the DPI bus sideband signals DPI.TYPE[15:8]. Calculates and applies the tHEC.
  - Optionally constructs a 0-60B extension header (including pre-computed eHEC) on a per-channel register settings or transmits a 4B extension header (including pre-computed eHEC) received inband. The eHEC is not computed by the GFP-F mapper.
  - Optionally, computes either a payload FCS (pFCS) as defined in ITU-T G.7041[31] or an Ethernet FCS (eFCS).
  - Optionally, scrambles the payload area using the  $x^{43}+1$  self-synchronizing scrambler.
- The GFP-F frame sub-blocks are described as follows.

#### 5.9.6.5.1 cHEC and tHEC Calculation

The Core and Type Header HEC fields are both calculated by generating a CRC using the polynomial  $x^{16} + x^{12} + x^5 + 1$ .

#### 5.9.6.5.2 Core Header Scrambler

The GFP frame core header and payload area are scrambled separately. The core header is scrambled for DC balance by an exclusive-OR operation (modulo 2 addition) with 0xB6AB31E0. This number is the maximum transition, minimum side lobe, Barker-like sequence of length 32. The scrambling of the GFP core header improves the robustness of the GFP frame delineation procedure and provides a sufficient number of 0-1 and 1-0 transitions during idle transmission periods.

#### 5.9.6.5.3 FCS Generation Logic

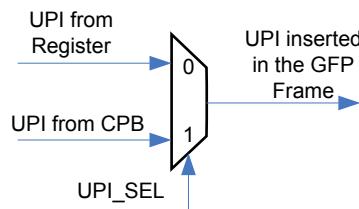
Either a pFCS or eFCS may be calculated over the GFP-F Payload area using CRC-32.ISO-13239. The pFCS and eFCS generating polynomial is the same:  $G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ , with an initial value of all 1s. Both FCS types transmit the 1s complement of the remainder of the polynomial division. They differ just in bit ordering within octets:

- For pFCS, octets are processed from MSB to LSB. For eFCS, octets are processed from LSB to MSB.
- For pFCS, the (1s complement of the)  $x^{31}$  coefficient of the CRC is mapped to the MSB of the first octet of the FCS field and the  $x^0$  coefficient is mapped to the LSB of the fourth octet of the FCS field. For eFCS, the (1s complement of the)  $x^{31}$  coefficient of the CRC is mapped to the LSB of the first octet of the FCS field and the  $x^0$  coefficient is mapped to the MSB of the fourth octet of the FCS field.

#### 5.9.6.5.4 Type Header UPI Byte Insertion Logic

The UPI byte can be inserted from the value received from the CPB or ENET subsystems or can be inserted from a user-programmed register on a per channel basis.

Figure 81 UPI Byte Insertion in the Type Header



#### 5.9.6.5.5 Payload Scrambler

Scrambling of the GFP-F payload area is required to provide security against payload information replicating the scrambling word (or its inverse) from a frame synchronous scrambler such as those used in an OTN frame. Scrambling is performed using a  $x^{43} + 1$  self-synchronous scrambler, as described in ITU-T G.7041/Y.1303 [31].

#### 5.9.6.6 GFP Idle Frame Insertion Logic

As part of rate adaptation, to meet the per-channel traffic rate, the GFP Idle Insertion Logic inserts (optionally scrambled) GFP Idle frames depending on the occupancy of the downstream packing FIFO. This is required when there is no client data available or when the effective incoming data rate is reduced because too many small packets are being received on the DPI interface, reducing the latter's throughput.

If an Ethernet Ordered Set frame is detected (identified by the UPI field on the DPI bus sideband signals matching a global, programmable register value), the Idle Insertion Logic will insert a minimum of 1 idle frame between the Ordered Set frame and the following frame. Note that because the UPI setting for ordered sets is global, the device cannot simultaneously support current and legacy UPI codes for this feature.

#### 5.9.6.7 OPUk Frame Generation Logic

The OHFS\_INSERT block receives OPUk payload bytes per channel, inserts octets for the Frame Alignment, OTUk, OPUk and ODUk overheads, generates the OPUk overhead and outputs the ODUk frame on the DCI interface. Note that many of the features of this block (e.g. fixed stuff insertion) are not applicable to GFP-F framed mappings.

The following overhead fields are generated:

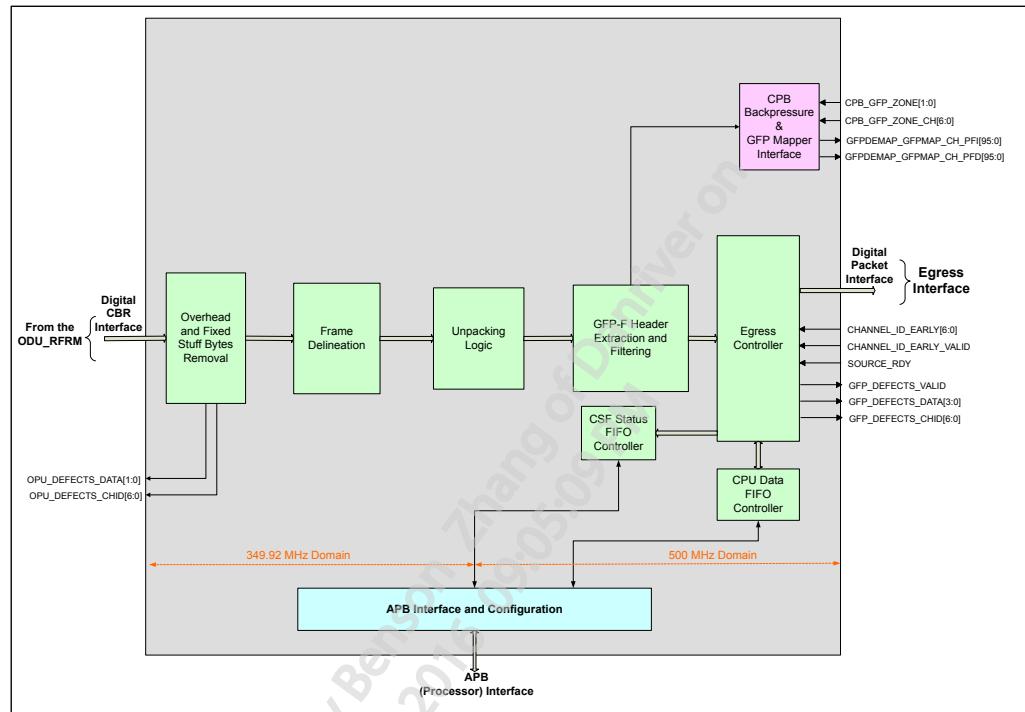
- FAS and MFAS bits in the Frame Alignment overhead
- PSI octet in the OPUk overhead
- Resize Control Overhead (RCOH) octets in the OPUk overhead, for Hitless Adjustment of ODU-flex(GFP).

The PSI octet in the OPUk overhead carries the CSF (Client Signal Failure) bit. This bit is set and cleared by firmware writing to a register bit. The GFP-F framer does not support any automatic setting of this bit in response to detected error conditions.

## 5.9.7 GFP-F De-mapper

### 5.9.7.1 Block Diagram

Figure 82 GFP-F De-mapper Block Diagram



### 5.9.7.2 Overview

The GFP-F de-mapper is a time-sliced processor capable of processing multiple channels. When traffic is fed from the multi-channel ODUk receive framer (ODU0, ODU1, ODU1e, ODU1f, ODU2, ODU2e, ODU2f, ODU3, ODU3e2, ODU4, ODUflex), the client data is extracted from the OPUk frame and forwarded to the CPB or the ENET subsystems.

The GFP-F de-mapper can receive the following categories of client data:

- Any packet rate client data mapped into OPU0, OPU1, OPU1e, OPU1f, OPU2, OPU2e, OPU2f, OPU3, OPU3e2, OPU4, and OPUflex, 10GE data mapped in OPU2, 40GE data mapped in OPU3, and 100GE mapped in OPU4.
- 10G Ethernet mapped into extended payload OPU2
- FC-1200 mapped into OPU2e

Note: 'Extended OPU2' is different from 'OPU2e'.

In the case of a GFP client-mapped in extended OPU2, the rate is the same as standard OPU2, but the OPU overhead area is used for the GFP client bytes except for the PSI byte. However, for OPU2e, the OPU overhead area is reserved, but the OPU2e is over-clocked and hence the rate of OPU2e is more than OPU2.

See Section 7 of ITU-T G709 [28] for the exact rates.

Initially, the GFP frames are extracted from the incoming ODU frame and frame delineation is performed based on the programmable number of validated core headers. Once frame synchronization is achieved, the GFP frame encapsulation is removed from the underlying client data frame. The GFP de-encapsulation process is made flexible by allowing HEC values to be ignored, verified, or verified with active single-bit error correction. Multiple programmable octet matching features allow user-specified frames to be either passed to the ENET or CPB interface or to the per-channel microprocessor-accessible storage or discarded. GFP Idle frames and control frames, having PLI less than 4, are discarded. It also supports payload FCS checking and removing FCS bytes.

The GFP-F de-mapper also provides a feature to check CMF frames. UPI values of frames identified as CMF by the filter are written into a software accessible CSF Status FIFO. It also provides a feature to capture the full CMF frames in CPU data FIFO which can be read by FW/SW.

The GFP-F de-mapper also supports Ethernet FCS checking and FCS byte removal. In case of an Ethernet FCS check, UPI is also checked and if it is an ordered set (defined by a global, configurable UPI field whose default value is 0x14) then it is interpreted that FCS is absent in the current frame. Note that because the UPI setting for ordered sets is global, the device cannot simultaneously support current and legacy UPI codes for this feature. eFCS checking and stripping in the GFP demapper in Gsup43 7.3 mode is not typically required. For packet termination, eFCS checking is done in an in-line MAC. For monitoring, a look-aside RX MAC is available to gather FCS-related statistics.

During GFP de-encapsulation, PAUSE frames are extracted and optionally forwarded to the CPB. In PAUSE extraction mode, the PAUSE time is extracted and passed to the CPU via status registers. During PAUSE time, per channel indication is given to the GFP mapper to stop data traffic for the channels which have received PAUSE frames. This functionality supports 802.3 link pause.

### 5.9.7.3 Features

- Extracts GFP frames from an incoming OPUk frame based on different OPUk mappings used and provides extracted GFP frames to the GFP-F de-mapper engine.
- De-encapsulation of incoming GFP-F frame payload as described in ITU-T G.7041// Y.1303 [31]
  - Core, type and extension headers are decoded by this block.
- Supports up to 96 independent time division multiplexed channels of encapsulated client frames.
- Provides received GFP frame synchronization with a user-programmable number of consecutively detected good core headers.
- Performs multi-bit error detection with optional single-bit error correction on the GFP frame core header only in the “SYNC” state. Uncorrected cHEC errors move the state machine to out of sync and all the frame data is dropped in de-mapper.
- Optionally removes DC-balancing Barker-like code from the core header.
- Provides multi-bit error detection with optional single-bit error correction on the GFP frame type header. Uncorrected frames are discarded. Type header fields (PTI, PFI, EXI and UPI) are accepted only after type check is complete.
- Provides multi-bit error detection and single-bit error correction on the GFP frame linear Extension Header. Single bit error correction is not supported for ring extension header. Corrupt frames are discarded and counted.
- Optionally de-scrambles the GFP payload area according to the generating polynomial  $G(x) = x^{43}+1$ .
- Supports discarding of GFP client data frames with client payload area greater than programmed maximum length.
- Supports discarding of GFP client data frames with client payload area less than programmed minimum length.

- Supports monitoring of client frame length in user-defined ranges. This monitoring is done after GFP de-encapsulation process.
- Terminates and discards GFP Idle frames (PLI = 0).
- Terminates and discards GFP Control frames with PLI = 1, 2, and 3, which are assumed to be composed of 4 bytes always (that is only core header). When the frames with PLI =1, 2 and 3 are received the interrupt is raised for that channel.
- Extracts the UPI field from the extracted GFP frame and provides to CPB/ENET as “TYPE” sideband signal.
- Provides programmable extension header size definitions (0 to 60 bytes) based on EXI values, with interrupt generation upon mismatch.
- Provides programmable type header matching filters with programmable frame disposition options to discard or forward frames to egress interface or CPU FIFO or CSF FIFO.
- Provides extension header type filter with programmable disposition options to discard the frame, forward to CPU or pass for further processing.
- Offers one programmable index (range from 0 to 64) and maskable 4-byte window generic filter with programmable frame disposition options to discard or forward frames to either egress interface or CPU.
- Provides a status register to report PTI/UPI and Extension header type filter failure. It captures the type header and channel ID for which the filter has failed. It is a global status register which can be configured to capture filter failure for a particular channel or any channel.
- Optionally checks the incoming payload for CRC/FCS errors (GFP FCS/Ethernet FCS). In case of HEC errors, the frame is dropped while in case of FCS errors the receive GFP-F de-mapper asserts the side band "DROP" signal to the CPB and CPB will in turn drop the frame. There is a user-configurable bit to decide whether the DROP is to be asserted or not. If it is configured not to DROP then an ERROR indication is given downstream for FCS error-ed frames. The FCS can optionally be stripped while extracting the payload.  
Checking of FCS and reporting error is supported irrespective of whether stripping of FCS is enabled or disabled. To strip the FCS bytes, checking of FCS should be enabled.
- Supports Ethernet FCS calculation and checking after skipping a programmed number of (preamble +SFD) bytes. The programmable skip feature is to support Ethernet client mapping in extended OPU2 (G.Sup43 7.3 [[27](#)]) where prior to GFP encapsulation, Ethernet frame preamble and SFD are preserved but the FCS is calculated over Ethernet data frame. Ordered set frames are skipped for Ethernet FCS checks. (UPI value for Ordered set is globally configurable with default value 0x14)  
Note that because the UPI setting for ordered sets is global, the device cannot simultaneously support current and legacy UPI codes for this feature. eFCS checking in the GFP demapper in Gsup43 7.3 mode is not typically required. For packet termination, eFCS checking is done in an in-line MAC. For monitoring, a look-aside RX MAC is available to gather FCS-related statistics.  
Ethernet FCS check is done after minimum and maximum frame length check.
- Supports a separate programming option to remove Ethernet FCS or GFP frame pFCS. pFCS check is performed only if the PFI bit is set in incoming GFP frame
- Detects PAUSE frames and extracts PAUSE time or just forward PAUSE frames based on configuration. Implements the PAUSE down counters and based on per channel PAUSE indication received from the GFP-F de-mapper, the GFP mapper stops transmitting client frames at the next frame boundary. During this condition, the GFP mapper provides GFP IDLE frames upon request from the ODUk scheduler. GFP IDLE frames are inserted only on frame boundary.
- In the de-mapping direction depending upon CPB receive FIFO zone status and user configured threshold, provides indication to GFP mapper for insertion of the PAUSE frames. The PAUSE frame

insertion can optionally be disabled for any channel. PAUSE frame insertion feature should not be used for the channels routed through ENET.

- Core headers for de-mapped GFP frames are always removed. Provides programmable options for the removal of the Type Header and the Extension Header from GFP frames. Two configurable bits per channel will be used to control the header removal.
- Provides 16-bit Length of the client frames at egress interface. This length gives the number of bytes from SOP to EOP driven on egress interface.
- Implements programmable definition of GFP Client Signal Failure (CSF) frames with CPU interrupt upon reception. The UPI of the received GFP CMF frame is used for the identification of CSF frames.
- Provides dCSF defect clearance for below conditions
  - On receiving a good client data frame, dCSF is cleared.
  - On receiving a defect clear indication (DCI), dCSF is cleared.
  - On seeing a per channel configuration bit (dCSF\_DEASSERT) asserted, dCSF is cleared (this configuration is supposed to be maintained by firmware and is the indication of time since the last corresponding CMF frame is received).
- Provides dedicated hardware support to check CSF frame/defect indication and capture their UPI into FIFO, which can be read out through a microprocessor interface. (See [Section 5.9.7.8.1](#) for details.) The CSF Status FIFO is common for all channels and has a depth of 96. Each entry has only extracted UPI and corresponding channel ID. (Note that if the status FIFO is used to capture MCF frames, then it is not possible to distinguish MCF from CMF.)
- Provides a dedicated interrupt output for PAUSE / CMF / CSF clear interrupts. This dedicated interrupt output is intended to be used by firmware. Whenever the client management frame is received, a change interrupt will be asserted and the extracted UPI field along with the channel number is pushed in dedicated CSF Status FIFO.
- Supports per channel back-pressure from the GFP-F de-mapper to the ODUk receive framer
- Supports information transcoded 10GE LAN de-mapping from an extended payload OPU2 ([Table 46](#) ).
- Supports G.Sup43 7.3 and G.Sup43 6.2 [\[27\]](#) de-mapping from ODU2 ([Table 46](#) ).
- Supports de-mapping of GFP-F mapped and transcoded FC-1200 payload from incoming ODU2e.
  - For FC1200 traffic, if the filter results to sending the payload on egress interface then instead of sending complete GFP frame payload area as a single packet, multiple packets of 48 bytes are sent with last payload bytes carrying the original EOP value.
- Supports 40GE and 100GE GFP-F de-mapping from OPU3 and OPU4 respectively ([Table 46](#) ).
- Provides OPU OH layer defects (OPU.dCSF, OPU.dPLM) on a per channel basis at the egress interface.
- Provides GFP defects (dLFD, dUPM, dEXM and dCSF) on a per channel basis at the egress interface. Each of these defects can be masked on a per channel basis.
- Generates Signal fail (SF) for received server layer signal fail (SSF received), payload type (PT) mismatch (dPLM), loss of frame delineation (dLFD), client signal fail (CSF), extended header identifier mismatch (dEXM) and user payload mismatch (dUPM).
- Supports extraction of OPUflex RCOH header.
- Process minimum GFP control frame of 4 bytes and minimum GFP data frame of 9 bytes.
- Supports maximum GFP frame of 65535 bytes.
- Provides per-channel configuration, control, status and performance monitoring registers.
- Provides performance monitor status counters per channel. All the counters saturate after reaching their maximum value. All the PMONs are 1 sec counters.

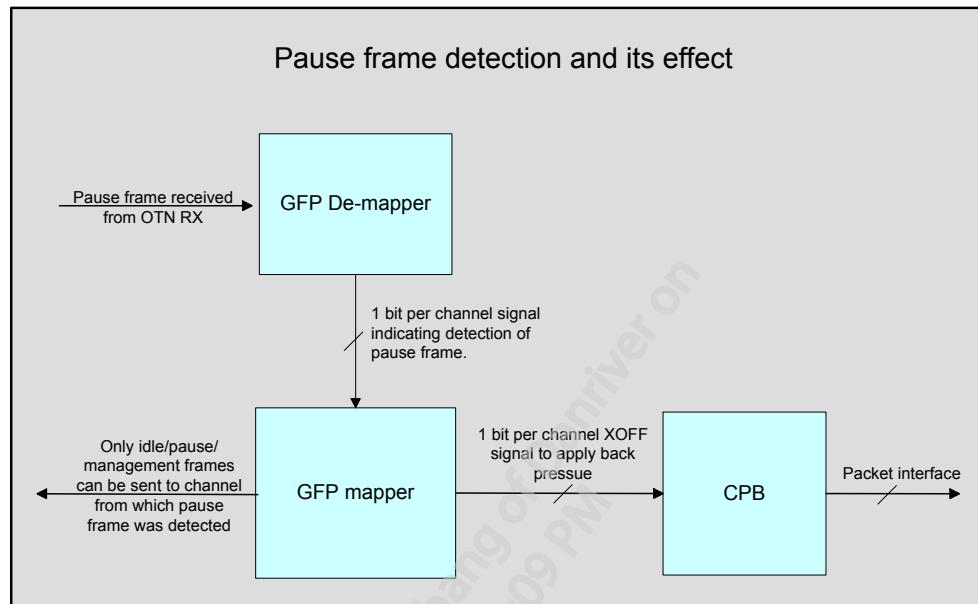
Table 51 GFP De-mapper PMON Status Counters

GFP Counters	Ethernet Counters
<ul style="list-style-type: none"><li>Received Idle frames.</li><li>Received non-idle control frames (i.e. PLI = 1, 2, 3).</li><li>Received bytes.</li><li>Received frames.</li><li>Packets dropped due to overabundance of small packets</li><li>Unpacked number of GFP frames to be processed by GFP sink processes.</li><li>Uncorrectable cHEC errors.</li><li>Uncorrectable tHEC errors.</li><li>Uncorrectable eHEC errors or extension header type filter mismatches.</li><li>Corrected cHEC errors.</li><li>Corrected HEC errors (tHEC or eHEC).</li><li>Received CSF frames.</li><li>Frames with pFCS error. Option of counting only FCS errored User Data Frames (PTI = 000)</li><li>PTI / UPI filter mismatches.</li><li>Generic filter mismatches.</li><li>CPU FIFO overflows.</li><li>Egress frames.</li><li>Number of frames dropped due to filter results.</li><li>Maximum frame length violations.</li><li>Minimum frame length violations.</li><li>Frame length statistic (0 to 6).</li></ul>	<ul style="list-style-type: none"><li>Frames with Ethernet FCS error.</li><li>Received number of PAUSE frames which are filtered.</li><li>Received number of PAUSE frames, which are forwarded transparently.</li></ul>

#### 5.9.7.3.1 PAUSE Frame Detection Feature

Note: This feature detects PAUSE frame described in IEEE 802.3 Annex 31B.

Figure 83 PAUSE Frame Detection and Its Effect



On receiving and detecting a PAUSE frame for a channel, the GFP-F de-mapper will extract the PAUSE time. Pause frames detected can be dropped or transparently forwarded ahead depending on a per channel configuration. A PMON is maintained for the number of pause frames received. The GFP-F de-mapper maintains a counter per channel which is loaded with the extracted pause time value. For a non-zero value of this counter, the corresponding channel bit going to the GFP mapper is asserted. On seeing this bit asserted, the GFP mapper stops the data traffic for this channel (at frame boundary) and starts sending idle frames for the corresponding channel instead of data frames upon request from the ODUK scheduler. The PAUSE timer in the GFP de-mapper decrements independently of the state of the GFP mapper. As a result, the value of the PAUSE time received in a PAUSE frame must take into consideration the data in transit in the network as well as a maximum size GFP frame that could be in progress in the mapper. When the de-mapper receives the PAUSE frame with PAUSE time equal to zero or when the PAUSE timer in the GFP-F de-mapper expires, the PAUSE indication to the GFP mapper is de-asserted and that can be used by the GFP mapper to start sending data again.

During the pause time, the GFP-F mapper can still send the PAUSE or CMF frames at frame boundary if triggered.

#### 5.9.7.3.2 Signal Fail Handling

- The events contributing to the signal fail indication are
  - Signal fail received from higher layers: server signal fail (SSF) and signal fail (SF)
  - PT mismatch event (OPU.dPLM)
  - OPU OH layer client signal fail event (OPU.dCSF)
  - Loss of GFP frame delineation (dLFD)
  - GFP client signal fail event (dCSF)
  - EXI mismatch event (dEXM)

- UPI mismatch event (dUPM)
- There is a per channel interrupt for each of these defects.
- There is a per channel configuration to squelch data when the server layer signal fail is seen asserted.
- Apart from these there are two defect ports given out with scheduled channel ID.
  - OPU layer defects : OPU.dPLM and OPU.dCSF driven with corresponding scheduled channel ID.
  - GFP layer defects : dLFD, dCSF, dEXM and dUPM driven with corresponding scheduled channel ID.

#### 5.9.7.4 Overhead and Fixed Stuff Bytes Removal Block (OHFS\_REMOVE)

This block in the GFP de-mapper:

- Extracts the OPUk payload out of the ODUk received.
- Validates PT information with the user-programmed values.
- Extracts the CSF bit from the PSI byte in OPU overhead.
- Removes ODUk overhead (Column 1 to 14 of ODUk frame).
- Optionally, extracts data from OPUk header depending on configuration.
- Optionally, removes only the PSI byte (OPUk header) depending on configuration.
- Removes fixed stuff depending on configuration.
- Supports extraction of header bytes for G.7044 processing.

#### 5.9.7.5 Frame Delineation

The frame delineation block identifies the GFP frame boundary by validating the core header bytes of the incoming stream. Frame synchronization is achieved when a correct cHEC is calculated over the PLI field of a programmable number (delta) of consecutive GFP frames. Frame delineation supports:

- DC balancing on core header bytes.
- Single bit correction on the core header bytes is performed while the frame delineation is in SYNC state.
- Deletion of the GFP control frames (PLI = 0, 1, 2, 3). Interrupt is raised and a counter is maintained for idle frames (PLI=0) and non-idle control frames (PLI=1, 2, 3).

As hunting starts at the 48-byte boundary, frame delineation FSM is not strictly in accordance with description in G.7041 [31].

Supports all possible values of PLI (16-bit).

There can be a maximum 48 bytes of data in one clock cycle. Hence, it is possible that the state machine moves from a HUNT to SYNC state in a single 48-byte chunk. Considering the worst case with minimum size of frame to be 4 bytes (idle frame) and delta = 1, the state machine can potentially come to SYNC and lose it multiple times within 48 bytes. In the event it loses SYNC in between, at any given data transfer, the delineation algorithm starts checking for next good core header from the first byte of next data transfer.

The state machine in the G.7041 specification describes processing octet by octet and when a bad core header is received, hunting is started from the next octet. But in this design, since the data width is 48 octets, when a bad core header is detected in between, the rest of the octets are discarded and hunting starts from the first octet in the subsequent data transfer. So, for a state transition, PRESYNC  $\Rightarrow$  HUNT and SYNC  $\Rightarrow$  HUNT, where the rest of the bytes in the current data cycle are discarded (till 45th byte of current data transfer) and hunting starts by considering ending of assumed core header

as first byte of the subsequent data cycle (considering history of 3 bytes from previous data transfer). Delta (number of frames required to go from PRESYNC to SYNC state) is configurable and can have values 1, 2, 3.

#### 5.9.7.6 Unpacking Logic

The data from frame delineation can potentially have more than one frame in it. The data is initially transferred to Digital Packet interface clock domain by the communal asynchronous FIFO. The unpacking logic unpacks the data and provides a maximum of one GFP frame per cycle to next sub block.

This block also provides back-pressure when the internal FIFO nears full condition. In case of consecutive small size frames being de-mapped (PLI of GFP frame less than 30 or between 49 to 63 bytes), it can drop these small size GFP frames. One of the cases when this can occur is for 49-byte packets on Digital Packet Interface which takes two cycles, wasting 47 bytes worth of data opportunity in second cycle.

#### 5.9.7.7 GFP-F Header Extractor and Filtering Block

This GFP de-mapper block:

- De-scrambles the payload area of the received frame using self synchronous polynomial ( $x^{43}+1$ ).
- Extracts the type header, performs a tHEC check if enabled, and also supports single bit error correction in tHEC.
- Extracts the extension header, performs an eHEC check if enabled, and also supports single bit error correction for linear extension header.

**Note:** In case of a non-linear extension header, the length of extension header is derived from the look up configuration, which uses the extracted EXI value from the Type header for look up. Supports extension header length up to 60 bytes.

- Extracts four bytes from the incoming frame at an offset programmed in generic filter for later use in filtering.
- Performs filtering on the incoming frame:
  - Supports 10 sets of identical filters per channel
  - Each filter can act as type header filter or generic filter. For type header filter, PTI, UPI and EXI are used to match values in filter set. Each has its own programmable mask and comparison operator. For generic filter, there is a common set of configuration of generic offset, mask and value.
  - Depending on filter match, generates defect events (dEXM, dUPM, generic mismatch) for a given frame.
  - Depending on filter match, supports dCSF event generation.
  - Each filter supports four sets of actions: forward to egress interface, forward to CPU interface, drop the frame, drop the frame and store the UPI in the CSF Status FIFO to be read by the CPU.
  - These filter results are ignored for identified pause frames
- If enabled, performs pFCS or Ethernet FCS checking.

Ethernet FCS checking is done after frame length checking process.

Supports Ethernet FCS checking after skipping the programmable number of preamble and SFD bytes.

On FCS error (pFCS/eFCS), an interrupt is given and a 1 second PMON counter is maintained.

When pFCS/eFCS checking is enabled, any packet that fails the check must be flagged with either a 'drop' or an 'error' indication (depending on a per-channel configuration bit). Packets marked

for 'drop' will subsequently be discarded by the DCPB. Packets marked as 'error' that are sent to the COPI (PIF) will result in the CRC being intentionally corrupted. It is not possible to detect an error in the RGFP and not flag the packet one way or the other. Thus non-intrusive monitoring is not supported.

pFCS checking is done only when PFI bit in type header of received frame is seen asserted and FCS checking is enabled.

Support separate control for removal of FCS bytes after checking for pFCS or eFCS.

- Supports PAUSE frame filtering and the transparent forwarding of received PAUSE frames to the egress interface.
  - PAUSE time is extracted into status register and also loaded into the down counter. The GFP mapper is given a per channel signal of PAUSE frame detected as long as this down counter is non-zero. The GFP mapper, subsequently, on seeing this signal asserted, stops the data traffic on the given channel.
  - Supports the discarding of client data frame if its 'client payload length' (as defined in Figure 6-1 of G.7041 [31]) is greater than the 'maximum frame length' configuration or less than the 'minimum frame length' configuration.
  - Collects the statistics for number of frames having client payload area length in the configured ranges. There are seven set of counters, each having corresponding configuration for ranges.

#### 5.9.7.8 Egress Controller

The egress controller does the data stripping and compaction of data. The core header is removed from the frame. Type and extension headers are optionally removed from the frame. Depending on the results of filters and events from the rest of the header processing blocks, the frame is forwarded to the egress interface and/or CPU or discarded and/or a UPI is stored in a CSF Status FIFO.

If the frame has to be forwarded to the egress interface, it is written into per channel output FIFO. When a read request input from the packet maker (the subsystem MPMA block) is received, the data for a given channel ID is read out from this per channel FIFO and given out on the DPI towards the CPB/ENET subsystem.

Egress controller also maintains the GFP defects detected.

- dLFD (Loss of frame delineation): It keeps accumulating the dLFD indication received from the frame delineation block.
- dCSF (GFP client signal fail): It asserts this when there is a corresponding CMF frame received with that defect as programmed in the filters. The clearance is done on following events.
  - Good client data frame is received.
  - A CMF frame with UPI = DCI (defect clear indication) is received.
  - FW stores the time since last corresponding defect was received and asserts a configuration bit to clear this defect indication if no such CMF frame is received in  $N*1000$ ms. On observing this configuration bit asserted, hardware clears the defect indication.
- dEXM (EXI value mismatch): Based on the filter results and comparison with expected EXI value by extension header length process, dEXM defect event is maintained.  
In case of dEXM event, the status of type header and corresponding channel id is also maintained. It is a global status register which can be configured to capture filter failure for a particular channel or any channel.
- dUPM (UPI value mismatch): Based on the filter results, dUPM defect event is maintained.

In case of dUPM event, the status of type header and corresponding channel id is also maintained. It is a global status register which can be configured to capture filter failure for a particular channel or any channel.

The egress controller also updates the performance monitoring registers.

#### 5.9.7.8.1 **CSF Status FIFO Controller**

If the filter result includes the action of passing the frame to the CSF Status FIFO, the extracted UPI value along with channel ID is pushed into a FIFO. If no other action specifies frame's destination as the egress interface or the CPU, then the frame is discarded. However, if the frame has any other destination, then the CSF status FIFO is updated and the frame also goes to its determined destination.

The CPU can read this status FIFO through direct register access. A data available interrupt is provided when the occupancy of this FIFO crosses the programmed threshold value. An overflow interrupt is also provided if a CMF status is dropped because this FIFO has been full.

#### 5.9.7.8.2 **CPU Data FIFO Controller**

If the filter result includes the action of passing the frame to the CPU Data FIFO, the received frame, after optional stripping of type and/or extension header, is sent to the CPU Data FIFO, which is read by the CPU through a PCBI.

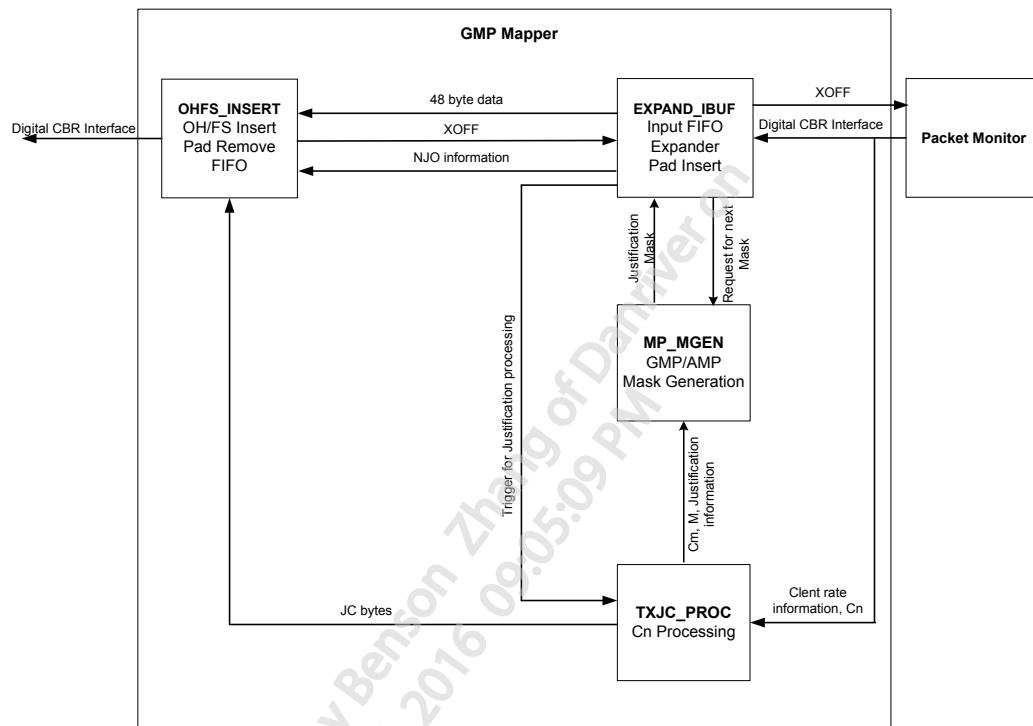
This CPU Data FIFO has four buffers per channel. Each buffer can, at the maximum, store one frame, but one frame can optionally occupy more than one buffer (if available). The size of each buffer is 144 bytes. Whenever any frame is written to this FIFO, a data available interrupt is given after a buffer is ready to be read. An overflow interrupt is also provided if any bytes are dropped because the buffers are full. There is also a programmable option which restricts storing any frame in only one buffer and if the length of frame is greater than buffer size, it is truncated and an overflow indication is given.

Wherever there is FCS error or an overflow condition, an error indication is given within the status of corresponding buffer. Each channel buffer has a software reset which resets the status register and data FIFO pointers for corresponding channel.

## 5.9.8 GMP Mapper

### 5.9.8.1 Block Diagram

Figure 84 GMP Mapper Block Diagram



### 5.9.8.2 Overview

The GMP mapper maps CBR clients over OPUk ( $k = 1, 2, 3, 4, 1e, 1f, 2e, 2f$ ) containers. It supports mapping of CBR traffic over 96 time division multiplexed (TDM) channels using any one of GMP, AMP or BMP mapping techniques. It also supports ODUflex technique to map clients that do not fit into OTN standard containers using GMP or AMP.

The GMP mapper contains the following blocks:

- The EXPAND\_IBUF block expands incoming CBR data to match the OPUk container size. The block aligns and sends the justified data stream to the egress data bus.
- The MP\_MGEN block generates AMP/BMP/GMP mask patterns to qualify OPUk payloads.
- The TXJC\_PROC block is a transmit justification processor. It generates justification information for the AMP, BMP and GMP mapping types.
- The OHFS\_INSERT block is an ODUk Overhead inserter. It receives OPUk payload bytes per channel, inserts FAS, MFAS, OPUk overheads and fixed stuff bytes, and sends the ODUk frame at the output.

### 5.9.8.3 Features

#### 5.9.8.3.1 Feature Overview

- Processes up to 96 channels in TDM with per channel enable and disable feature.
- Maps various clients over OPUk frames with necessary justification techniques like BMP, AMP or GMP as configured per channel.
- Per channel option to configure server size to indicate the rate of the OPUk container.
- Inserts Frame Alignment Overhead into the outgoing frames.
- Inserts OPUk overhead.
- Supports storing of up to 4 Cn values per channel from packet monitor (MPMO) within a frame duration.
- Derives justification information from Cn received from the packet monitor (MPMO).
- Supports mechanism of storing Cn residue per channel up to 8 bit values during an AMP technique.
- Optionally switches to previous Cn or a programmed golden Cn per channel on the event of a signal failure.
- Optionally inserts Fixed Stuff columns in programmed locations.
- Supports JC bytes encoding for AMP/GMP techniques.
- Supports CRC calculation over GMP Cm and CnD values.
- Applies back-pressure to upstream blocks at the ingress interface to avoid FIFO overflow in the datapath.
- Sends MFAS values over a sideband signal to downstream.
- Optionally inserts CSF into the OPUk OH PSI bytes.
- Optionally transmits NULL ODUk frames at the egress.

#### 5.9.8.3.2 Processing of 96 Channels in TDM

In the GMP mapper, the required channels can be enabled or disabled as required with the help of channel reset which is a configurable option.

#### 5.9.8.3.3 Mapping of Clients over OPUk Containers

The GMP mapper supports the mapping of various client types into different OPUk containers using any one of AMP/BMP/GMP techniques based on per channel configuration.

Table 47 lists all the possible mappings supported in the block.

#### 5.9.8.3.4 Frame Alignment Overhead Insertion

In the ODUK frame, columns 1 to 7 of row 1 are filled with FAS and MFAS bytes:

- FAS is 6 bytes: 0xF6F6F6\_282828.
- MFAS is the 8 bit counter value which increments by 1 in each frame. Count value moves from 0 to 255 and then rolls over.

#### 5.9.8.3.5 OPUk Overhead Insertion

The GMP mapper inserts OPUk overhead in columns 15 and 16 of the OTN frame:

- For BMP mapping, the JC bytes are reserved and PSI[2] is filled with PT byte.

- For AMP mapping, the specific JC encodings are performed to indicate an NJO or PJO event during mapping.
- For GMP mapping, the specific JC encodings are performed for Cm and CnD along with their CRC.

#### 5.9.8.3.6 PT Byte Insertion in OPU Overhead

The GMP mapper inserts a PT byte in the OPUk overhead in column 15 of the OTN frame based on user-programming on per channel basis.

#### 5.9.8.3.7 Justification Sideband from Packet Monitor

The packet monitor sends Justification Information (Cn) for each frame to the mapper but is not synchronized to the actual frame duration. The TGMP96 justification interface in the TXJC\_PROC block provides storage of 4 Cn values per channel in a FIFO. Only one Cn is read out and used at a time for justification processing per frame for a given channel.

- GMP Justification Value
  - $Cm = \text{INT}((Cn + \text{Prev\_CnD})/M))$
  - $CnD = (Cn + \text{Prev\_CnD}) \bmod M$
- AMP Justification Value (Cn) = ODUk size + NJO byte - PJO stuffs + residue NJO Bytes - unused previous PJO Stuffs
- TXJC\_PROC block in the GMP mapper uses Cn for:
  - AMP mapping: TXJC\_PROC uses the available stuff/byte (NJO/PJO) locations and any excess Cn residue will be added to the next justification Cn value for processing. The maximum residue expected is of 10 bits value.
  - GMP mapping: TXJC\_PROC calculates Cm, CnD from incoming Cn. If the Cm value turns out to be beyond +2 or -2 range from previous Cm, it is considered as new Cm and inserted into the JC bytes without any further encoding.

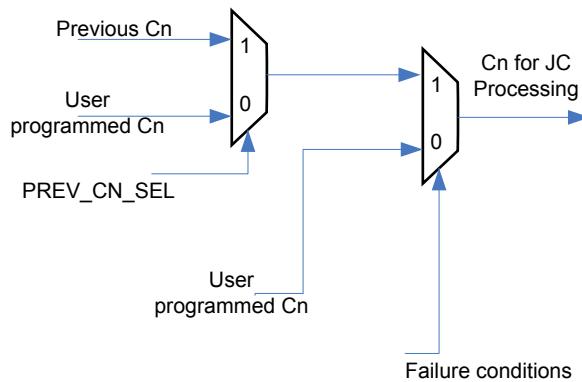
#### 5.9.8.3.8 Cn Selection

The GMP mapper provides storage for a user-programmed golden Cn value per channel, which is also configurable by the user. There is a Cn ramp generation mechanism to provide zero ppm variation for this user-programmed golden Cn.

For normal operation, the Cn received from packet monitor is used.

During ramp-up, if Cn is not available, there is an option to choose Cn dynamically either from the user-programmed Cn or a previous Cn. Based on the user-selected Cn, JC bytes would be calculated and data will be mapped.

Figure 85 Cn Selection for JC Byte Calculation



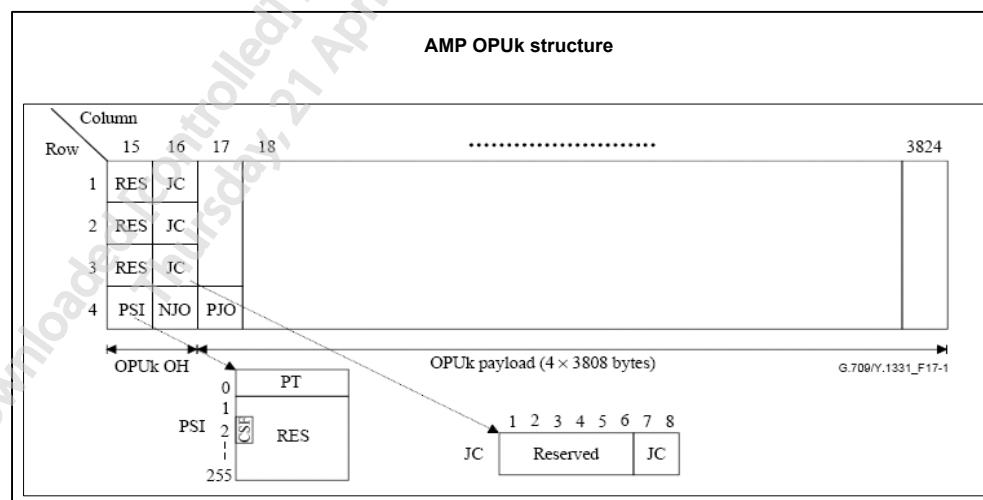
#### 5.9.8.3.9 Fixed Stuff Insertion

The GMP mapper provides fixed stuff insertion into user-programmed columns. There are two modes of stuff insertion allowed: 8 bytes and 16 bytes. There is a configurable option per channel to enable or disable stuff column insertion into the stream.

#### 5.9.8.3.10 Support for AMP Justification

The asynchronous mapping generates the JC bytes and inserts them in the OH locations according to Figure 86. NJO and PJO bytes are inserted in appropriate locations defined by the ITU-T G.709 standard for a given type of OPUk container.

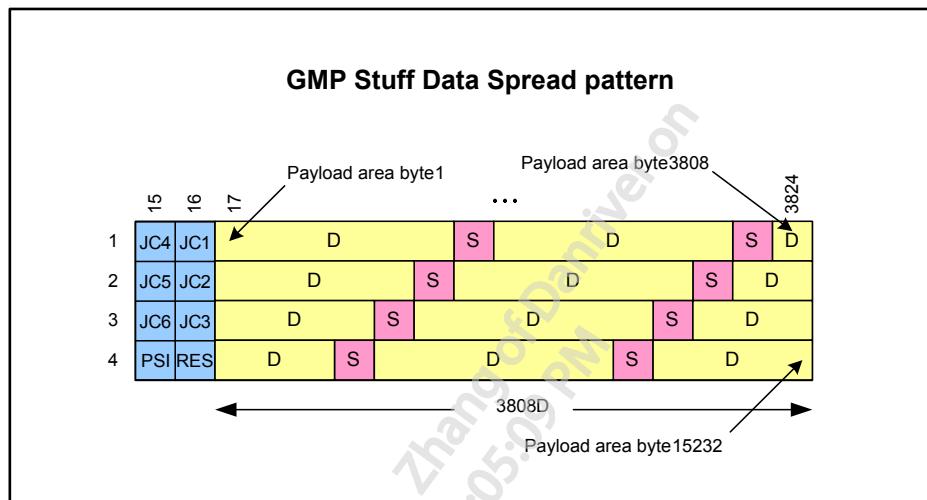
Figure 86 OPUk Frame Structure for AMP Mapping



### 5.9.8.3.11 GMP Data and Stuff Insertion

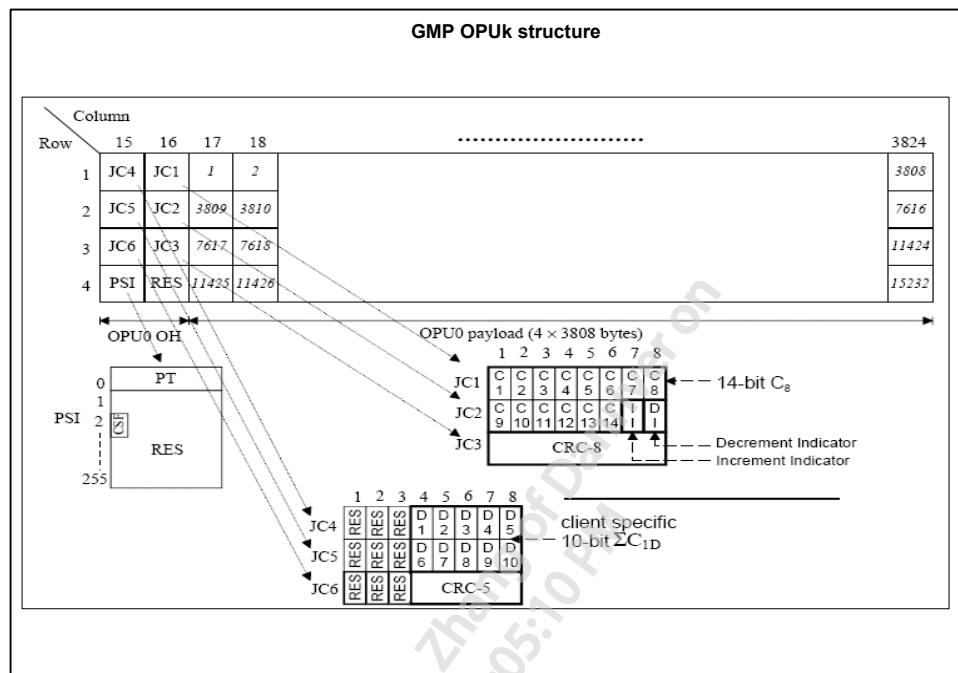
Unlike the fixed column stuff method used for AMP and BMP mapping, GMP makes use of an algorithm that ensures uniform spread of data and stuff bytes.

Figure 87 GMP Data/Stuff Spread Pattern



For the overhead information location for GMP mapping, see ITU-T G.709/Y.1331 (12/2009), Figure 17-12.

Figure 88 OPUk Structure for GMP Mapping



The following features are supported by the GMP mapper:

- Payload mapping with 8-bit timing granularity (Cn with n=8).  
Note: Payloads with 1-bit timing granularity are not supported (Cn with n= 1)
- CnD generation.
- Supports a wide range client rates for GMP mapping with M=1 to M=80 in different rate ODUK containers.

### 5.9.8.3.12 GMP Cm Insertion

The GMP mapper encodes the Cm values into the stream using the following guidelines.

The bit inversion patterns apply to the current Cm(t) value, prior to the increment or decrement operation that is signaled by the inversion pattern. The incremented or decremented Cm(t) value becomes the base value for the next GMP overhead transmission:

- When the value of the Cm(t) is incremented with +1 or +2, a subset of the Ci bits as specified in [Table 52](#), is inverted and the increment indicator (II) bit is set to 1.
- When the value of the Cm(t) is decremented with -1 or -2, a subset of Ci bits as specified in [Table 52](#), is inverted and the decrement indicator (DI) bit is set to 1.
- When the value of Cm(t) is changed with a value larger than +2 or -2, both the II and DI bits are set to 1 and the Ci bits contain the new Cm(t) value.
- When the value of Cm(t) is unchanged, both the II and DI bits are set to 0.

**Table 52** Cm(t) Increment and Decrement Indicator Patterns

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	II	DI	Change
U	U	U	U	U	U	U	U	U	U	U	U	U	U	0	0	0
I	U	I	U	I	U	I	U	I	U	I	U	I	U	1	0	+1
U	I	U	I	U	I	U	I	U	I	U	I	U	I	0	1	-1
U	I	I	U	U	I	I	U	U	I	I	U	U	I	1	0	+2
I	U	U	I	U	U	I	U	U	I	U	U	I	U	0	1	-2
Binary value													1	1	More than +2/-2	

Note: I -> indicates inverted Ci bit ; U -> indicates unchanged Ci bit

### 5.9.8.3.13 BMP Mapping Support

The GMP mapper supports mapping of CBR clients into a ODUK container using BMP techniques. In this mode, the client rate exactly matches with the container (server) rate and thus there is no byte justification needed. The ODUkSC is programmed in such a way that the scheduled client rate matches or is a little higher than the configured server rate.

### 5.9.8.3.14 Back Pressure Support

The GMP mapper does not accept any back pressure from downstream blocks (that is, from the transmit framer) because the mapper provides output based on the ODUkSC's requests. Downstream blocks are also expected to operate on the same rate as GMP mapper.

The GMP mapper can back pressure the packet monitor when its FIFOs goes above a fill threshold. The fill thresholds are configurable per channel for a given rate of container.

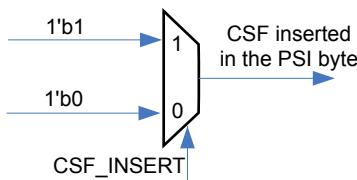
### 5.9.8.3.15 NULL Frame Insertion

The GMP mapper provides an option to insert NULL frames into the stream per channel. When a channel is configured for NULL frame insertion, the upstream blocks are not supposed to send any data.

A channel can be configured for either normal frame processing or NULL frame processing. A mix of NULL and normal frames is not allowed.

### 5.9.8.3.16 Handling Signal Failure Conditions

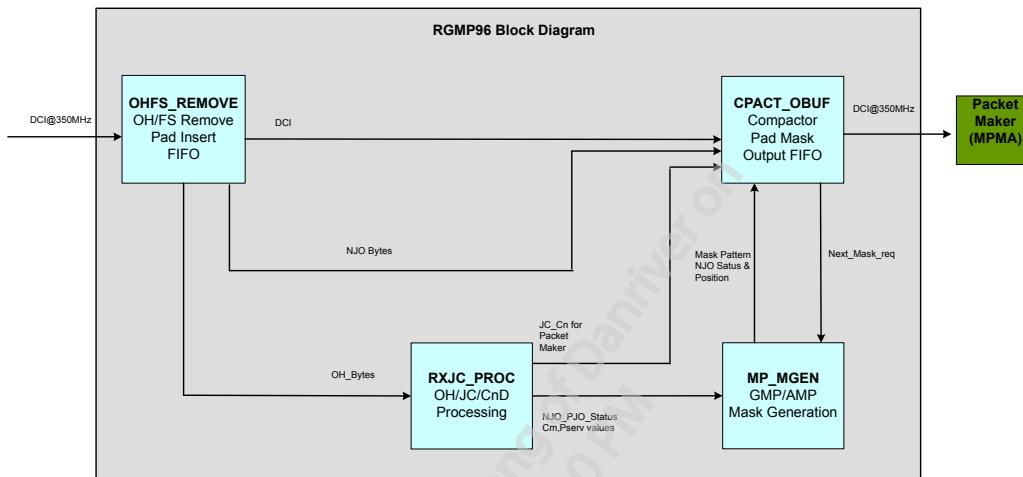
Signal failure conditions are handled by Software. There is a configurable bit in the OHFS\_INSERT, which allows the OHFS\_INSERT block to optionally inserts the CSF bit into the PSI overhead byte.

**Figure 89** Handling a Signal Failure

## 5.9.9 GMP De-mapper

### 5.9.9.1 Block Diagram

Figure 90 GMP De-mapper Block Diagram



### 5.9.9.2 Overview

The GMP de-mapper is a CBR de-mapping engine. It handles various CBR traffic, where  $2.488G \leq \text{CBR client} \leq 104.343G$ . The GMP de-mapper supports time-sliced processing with payloads being mapped into individual channels. The de-mapper supports a maximum of 96 channels, with each occupying the processing bandwidth based on the respective payload rates, and these channels add up to an aggregate rate of 120G. The GMP de-mapper follows the DCI protocol at the ingress and egress paths. The de-mapper removes stuff and overhead locations in the process of extracting the payload, which in turn results in a reduction of data rate as the payload moves downstream.

The GMP de-mapper blocks include:

- OHFS\_REMOVE to extract OPUk payload & overhead from an ODUk.
- RXJC\_PROC to process the justification control bytes and to generate CnD, Cn and CM values.
- MP\_MGEN to generate AMP/BMP/GMP mask patterns to qualify OPUk payload.
- CPACT\_OBUF to qualify and compact the incoming CBR data based on standard/custom stuff patterns.

### 5.9.9.3 Features

- Supports de-encapsulation of various data rates from ODUk frames with justification overhead as described in ITU-T G.709.
- Supports ODUk overhead removal.
- Supports OPUk overhead extraction (see [Section 5.9.9.3.2](#)).
- Supports removal of fixed stuff columns (programmable feature) (see [Section 5.9.9.3.4](#)).
- Supports user-programmable fixed stuff location & type (8 bytes and 16 bytes).
- Supports CBR payload de-mapping, in the range of  $2.488G$  to  $104.343G$ , from different OPUk and OPUflex containers.

- Supports up to 96 channels, with a maximum individual rate of 104.343G.
- Supports PT (Payload Type) acceptance and validity check (dPLM - defect Payload Mismatch).
- Supports payload extraction based on AMP/BMP methods (see [Section 5.9.9.3.3](#)).
- Supports payload extraction based on GMP scheme (see [Section 5.9.9.3.5](#)).
- Achieves synchronization with source using Finite State Machine (FSM) specified in G-709 (see [Section 5.9.9.3.8](#)).
- Checks CRC8 over Cm bytes (JC1 to JC3) and CRC5 over CnD bytes (JC4 to JC6) (see [Section 5.9.9.3.6](#)).
- Supports Cm extraction scheme capable of extracting the correct value in spite of CRC errors (see [Section 5.9.9.3.7](#)).
- Stores extracted Cm values on a per channel basis in a RAM for S/W access.
- Supports programmable de-mapping scheme, whether AMP/BMP/GMP, on per channel basis.
- Supports NULL client for test purpose ( dPLM check is performed on the client, while discarding the payload).

[Table 47](#) lists all the possible mappings supported in the block.

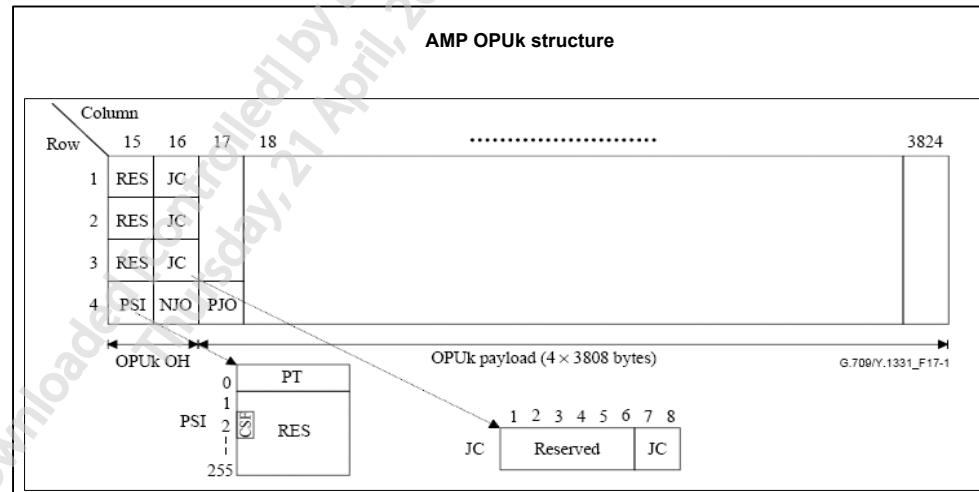
### 5.9.9.3.1 ODUk Overhead Removal

The ODUk overhead (columns 1 to 14 in the four rows) is removed as part of the de mapping process.

### 5.9.9.3.2 OPUk Overhead Extraction (AMP/BMP)

The OPUk structure for CBR mapping (AMP/BMP) is shown in [Figure 91](#). Columns 15 and 16 contain the OPUk-specific overhead information.

[Figure 91](#) OPUk Frame Structure for AMP/BMP Mapping



Note: [Figure 91](#) is from [\[28\]](#) (Figure 17.1)

The following features are offered by the GMP de-mapper:

- Extraction of OH bytes in columns 15 and 16 based on the mapping type (AMP and BMP).
- Extraction of JC bytes to detect NJO/PJO status.
- Checking of the PSI field containing the payload type. If the extracted payload type from the PSI field does not match the expected one configured by software, an interrupt is raised (dPLM).
- If the CSF bit is set in the incoming stream, an interrupt is raised (dCSF)
- dPLM and dCSF statuses are made available at the block output via a port that works in a Time Division Multiplexed (TDM) manner.

#### 5.9.9.3.3 Support for AMP/BMP Justification

In asynchronous mapping mode (AMP), the mapper generates the JC bytes and insert them in the OH locations. The de-mapping process interprets the JC bytes according to [Table 53](#). A majority vote (two out of three) is used to make the justification decision in the de-mapping process to protect against an error in one of the three JC bytes.

The AMP/BMP selection is based on software programming. Based on [Table 53](#), a decision is made on either positive or negative justification for the AMP mapping.

**Table 53 JC, NJO and PJO Interpretation**

JC Bits (7,8)	NJO	PJO
00	Justification byte	Data byte
01	Data byte	Data byte
10(Note)	Justification byte	Data byte
11	Justification byte	Justification byte

**Note:**

1. A mapper circuit does not generate this code. Due to bit errors a de-mapper circuit might receive this code.
2. [Table 53](#) is referred from [\[28\]](#) (Table 17-3).

If the JC bytes do not offer a majority decision, the GMP de-mapper treats NJO as stuff and PJO as data, and continues processing. Along with an interrupt, the PMON counter is updated to track this event.

In bit-synchronous mode (BMP), JC bytes are ignored by the de-mapper (NJO is ignored and PJO is interpreted as data).

#### 5.9.9.3.4 Removal of Fixed Stuff Columns

The GMP de-mapper supports the removal of fixed stuff columns present in the data stream. The location of fixed stuff (FS) columns is configured by software. The AMP/BMP mapping of the CBR10G into OPU2 introduces one column of fixed stuff bytes, with a width of 16 bytes, at position 1905. The AMP/BMP mapping of CBR40G into OPU3 introduces two columns of fixed stuff bytes, each with a width of 16 bytes, at positions 1265 and 2545. While mapping CBR data ~104G into OPU4 using GMP, there is fixed stuff column at the end of the frame. The stuff column has a width of 8 bytes from positions 3817 to 3824.

The different stuff column layouts are shown in [Figure 92](#), [Figure 93](#) and [Figure 94](#).

Figure 92 ODU2/ODU2e/ODU2f Fixed Stuff Column (AMP/BMP) Modes

ODU2 Fixed stuff columns									
Row #							Column #		
	15	16	17	.....	1904	1905	.....	1920	.....
1	RES	RES	RES		118 × 16D	16FS		119 × 16D	
2	RES	RES	RES		118 × 16D	16FS		119 × 16D	
3	RES	RES	RES		118 × 16D	16FS		119 × 16D	
4	PSI	RES	RES		15D + 117 × 16D	16FS		119 × 16D	
	NJO	JC							
	PJO								

G.709/Y.1331\_F17-3

Note: Figure 92 is referenced from [28] (Figure 17-3).

Note: There are no Fixed Stuff columns for ODU1e/1f mappings.

Figure 93 ODU3 Fixed Stuff Columns (AMP/BMP) Modes

ODU3 Fixed stuff columns										
Row #							Column #			
	15	16	17	.....	1264	1265	.....	1280	1281	.....
1	RES	RES	RES		78 × 16D	16FS		79 × 16D	16FS	79 × 16D
2	RES	RES	RES		78 × 16D	16FS		79 × 16D	16FS	79 × 16D
3	RES	RES	RES		78 × 16D	16FS		79 × 16D	16FS	79 × 16D
4	PSI	RES	RES		15D + 77 × 16D	16FS		79 × 16D	16FS	79 × 16D
	NJO	JC								
	PJO									

G.709/Y.1331\_F17-4

Note:

1. FS indicates Fixed Stuff.
2. Figure 93 is referenced from [28] (Figure 17-4).

Figure 94 ODU4 Fixed Stuff Column for GMP

ODU4 GMP Fixed stuff columns													
Row	Column	15	16	17	56	57	96	97	3776	3816	3817	3824	3825
1	JC4	JC1	1		1	1	1	2	48		48		
2	JC5	JC2	48		48	49	49	49		95		95	
3	JC6	JC3	96		96	96	96	97		143		143	
4	PSI	RES	143		143	144	144	144		190		190	

Note: Figure 94 is referenced from [28] (Figure 17-16).

#### 5.9.9.3.5 GMP: Data and Stuff Separation

The following features are supported in the GMP de-mapper:

- Supports payload with 8-bit timing granularity (Cn with n=8 )
- Supports client de-mapping from OPU2 containers with M value =8
- Supports client de-mapping from OPU2e containers with M value =8,9
- Supports client de-mapping from OPU3 containers with M value=32
- Supports client de-mapping from OPU4 containers with M value=80
- The derived CnD value is not used in the de-mapping process. However, this is an important parameter to deduce the client rate when the CBR client is driven on an interface like XLAUI, CAUI or XFI. The parameter is passed downstream for use in clock rate capturing blocks in the device (that is, the packet monitor). The packet monitor will use the Cn and CnD to control the client data rate. If the extracted CnD value goes beyond the maximum, that is, the M value for the container, an interrupt is raised.
- The Cm, CnD parameters extracted in one frame will be used in the next frame

Note: The transient PLL phase information CnD can be inserted with n=8 or n=1 (in terms of bytes or in bits). Support for byte granularity is mandated by the standard but the bit granularity support is implementation-specific. There is no support for client payloads with bit level granularity.

#### 5.9.9.3.6 GMP: CRC Check on Justification Bytes

The GMP de-mapper performs a CRC check on the justification fields. The 8-bit CRC over the JC1 and JC2s field is put in the JC3 byte. The JC4, JC5 and JC6 bytes consist of:

- a 10-bit CnD field (bits D1, D2, .., D10).
- a 5-bit CRC-5 field, which contains an error check code over bits 4 to 8 in JC4 and JC5.

The de-mapper supports parallel CRC logic as specified in Appendix IX of [28].

Annex D.3 of [28] lists the methods to recover Cm in spite of having CRC errors. There are still scenarios that lead to a loss of synchronization with respect to the source. The standard requires that the synchronization search be re-initiated while the incoming frame is processed with the previous value of Cm until resync is achieved.

In the event of a CRC error in the CnD, GMP de-mapper uses zero.

An interrupt is raised in the event of a CRC error in either the Cm or CnD.

#### 5.9.9.3.7 GMP: Cm Extraction

The GMP de-mapper is responsible for being synchronized to the source. Once synchronization is achieved, based on the CRC status, the block derives the Cm value for the next frame.

The mapper encodes the Cm and CnD values into the stream using the following guidelines. The reverse of this logic is used in the GMP de-mapper side.

The bit inversion patterns apply to the current Cm(t) value, prior to the increment or decrement operation that is signalled by the inversion pattern. The incremented or decremented Cm(t) value becomes the base value for the next GMP overhead transmission:

- When the value of the Cm(t) is incremented with +1 or +2, a subset of the Ci bits, as specified in the below table, is inverted and the increment indicator (II) bit is set to 1.
- When the value of the Cm(t) is decremented with -1 or -2, a subset of Ci, as specified in the below table, is inverted and the decrement indicator (DI) bit is set to 1.
- When the value of Cm(t) is changed with a value larger than +2 or -2, both the II and DI bits are set to 1 and the Ci bits contain the new Cm(t) value.
- When the value of Cm(t) is unchanged, both the II and DI bits are set to 0.

Table 54 Cm(t) Increment and Decrement Indicator Patterns

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	II	DI	Change			
U	U	U	U	U	U	U	U	U	U	U	U	U	U	0	0	0			
I	U	I	U	I	U	I	U	I	U	I	U	I	U	1	0	+1			
U	I	U	I	U	I	U	I	U	I	U	I	U	I	0	1	-1			
U	I	I	U	U	I	I	U	U	I	I	U	U	I	1	0	+2			
I	U	U	I	U	U	I	U	U	I	U	U	I	U	0	1	-2			
binary value										1	1	More than +2/-2							
Note:																			
1. "I" indicates inverted Ci bit ; "U" indicates unchanged Ci bit																			
2. Table 54 is referenced from [28] (Appendix D Table D-2).																			

The extracted Cm value is checked against a user-programmable range. The minimum and maximum value for Cm is user-configurable on a per channel basis. An interrupt is raised if the extracted Cm value falls outside this range or if the CnD value goes above the permissible limit (M value). The extracted Cm value is also captured in a RAM for firmware access. If the extracted CnD value goes above the limit, zero is used instead.

#### 5.9.9.3.8 GMP: Synchronization with Source

The state machine shown in Appendix D of [28] is used to achieve synchronization with the source. When the GMP sink has synchronized its Cm(t) value, the received JC1-JC3 octets are processed as per Section 5.9.9.3.7.

The following features are supported as part of the synchronization stage:

- At the very beginning, incoming data is ignored by the de-mapper (data is processed with Cm=0) until it finds the synchronization for the specified channel.

- Sync hunting logic can come to SYNC within two frames, the worst case scenario provided no bad CRC patterns are received during the sync hunt process.
- FSM remains in HUNT state (HUNT\_A to HUNT\_F) only for a single cycle. If it does not receive a valid pattern to move to the SYNC state, the FSM goes back to a 'START\_HUNT' state.
- Ability to come to sync within a single frame, which is the best case scenario.
- An interrupt is raised if the FSM goes out of sync and a PMON counter is updated to track this event.

#### 5.9.9.3.9 GMP Stuff Extraction Block

The following equation is used to determine the GMP data and stuff byte locations in the ODUK frame. The variable 'j' moves in the range of 1 to Pserver.

- Client data (D) if  $(j \times Cm(t)) \bmod Pserver < Cm(t)$
- Stuff (S) if  $(j \times Cm(t)) \bmod Pserver \geq Cm(t)$

The data granularity changes based on the container rate. The data granularity for different container rates is as given below:

- For OPU2 and OPU2e, data unit granularity is 64 bits (8 bytes).
- For OPU3 data unit granularity is 256 bits (32 bytes).
- For OPU4, data unit granularity is 640 bits (80 bytes).

For example, in the case of OPU2, 6 data units ( $6=48B/8B$ ) need to be processed in a single cycle. In the case of OPU4, one data unit could be spread across three cycles of processing clock as an  $80B > 48B$  bus width.

The parameter 'Pserver' indicates the number of possible locations in the container. The variable 'j' tracks the current position in the container. The range of 'Pserver' and 'j' for different container rates is given below:

- For OPU2 and OPU2e,  $P(server)$  is 1904.
- For OPU3,  $P(server)$  is 476.
- For OPU4,  $P(server)$  is 190.

#### 5.9.9.3.10 Signal Fail Handling

As part of signal fail handling, a SSF is (optionally) created within the GMP de-mapper due to dPLM and dCSF.

The GMP de-mapped data can be fed to either the CPB or ENET subsystems. The ODUK receive framer can assert a signal fail condition as part of the DCI protocol, via the DCI\_SSF and DCI\_SF signals.

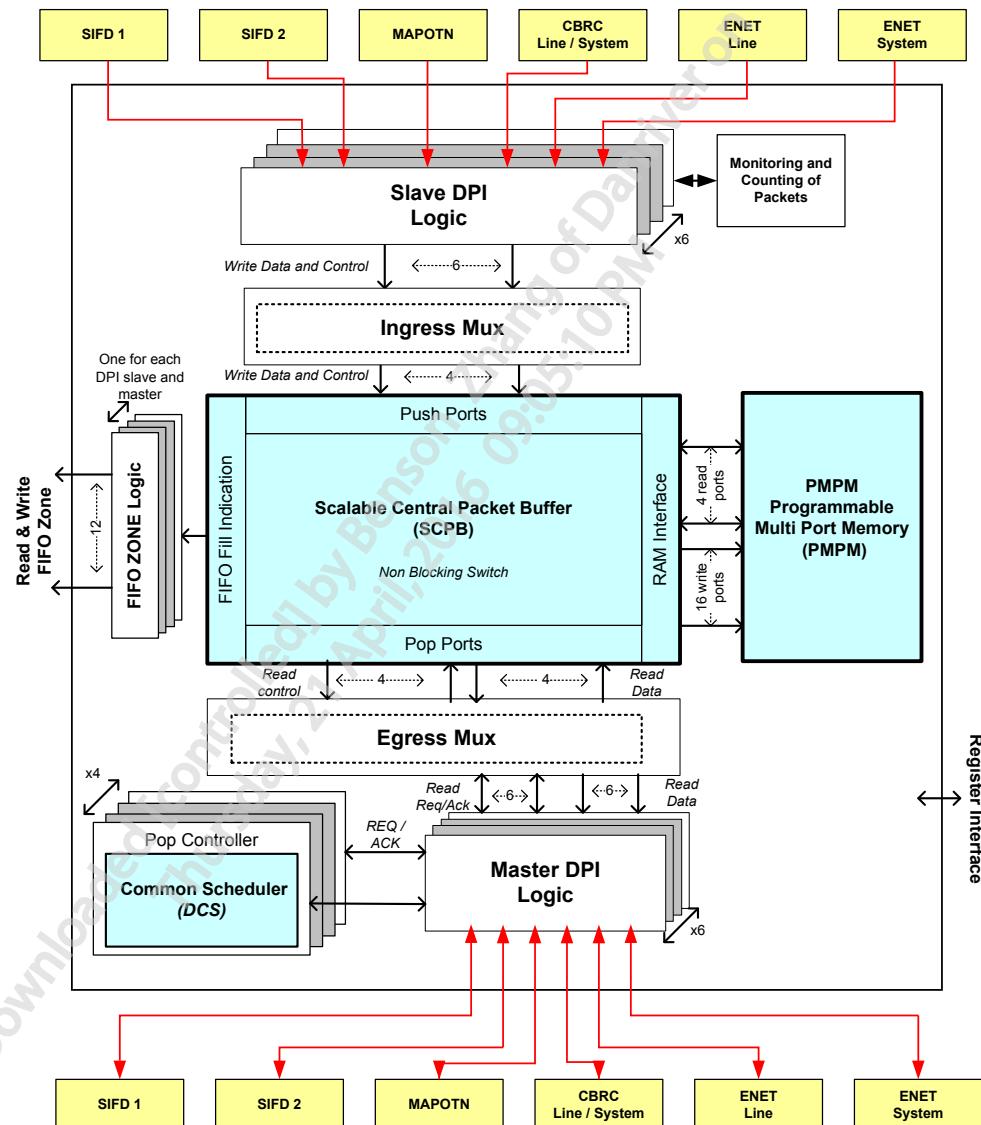
During an SSF, de-mapper shuts down the traffic.

## 5.10 Central Packet Buffer (CPB) Subsystem

### 5.10.1 Overview

The CPB (Central Packet Buffer) subsystem is a central repository for buffering of packets between source and destination clients. It is a store and forward, non-blocking, FIFO-based, bi-directional packet switch. Clients communicate to the CPB through a channelized packet interface called the DPI, each client having a dedicated DPI port. See [Figure 95](#) for the block diagram.

[Figure 95](#) CPB Block Diagram



Source clients generate channel associated packet data and send the packets to the CPB by mastering a slave DPI port. Destination clients receive channel associated packet data whenever the CPB schedules

a channel on a master DPI port. Incoming source channels are directly associated to internal queues and internal queues are mapped to outgoing destination channels.

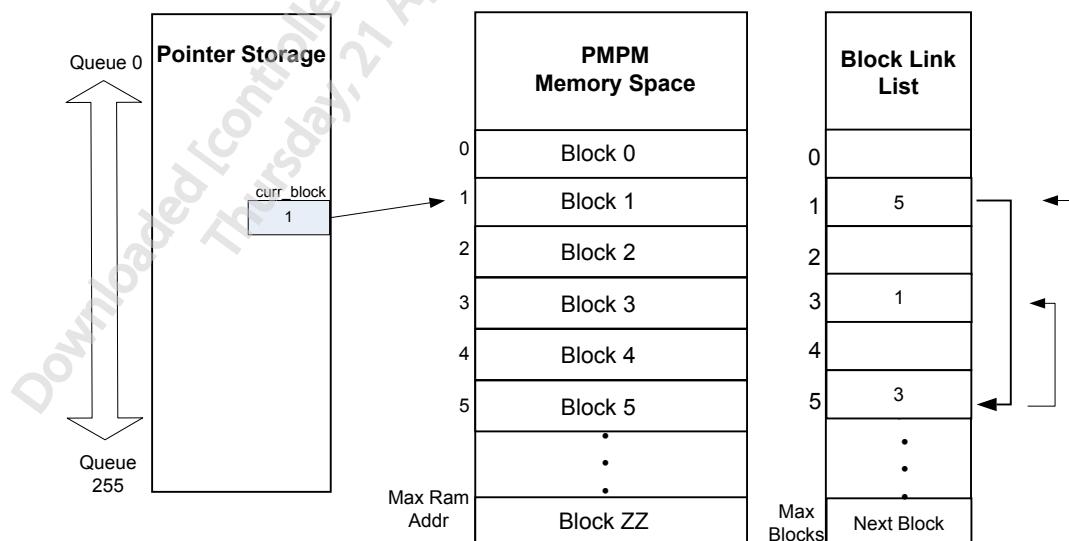
Packets flow through the CPB as follows. Packets arrive on the slave DPI ports. The ingress mux block receives a packet from a specific channel slave DPI port and routes it to one of four ingress ports of the SCPB (Scalable Central Packet Buffer). Within the SCPB, the packet is transmitted to four queue systems with each queue system accepting only those packets that match a user configured source channel and port number. Broadcast (duplication of packets) is achieved by configuring more than one queue system to accept the source channel. Broadcast is typically limited to 2-way broadcast. However, to allow the support of a unidirectional "mirror" port, triple-broadcast is supported in a limited fashion.

Schedulers within the pop controllers, associated with each of the four egress ports of the SCPB, are responsible for reading packets from queues. A scheduler selects the queue to be serviced based on weighted round robin and deficit round robin algorithms. A queue is eligible for servicing when at least one packet is stored within the queue. Once selected, the packet data is retrieved from the queue and forwarded to the egress mux. The scheduler is configured to associate a queue to an output destination channel and provides the egress mux with this information. The egress mux steers the packet to the appropriate master DPI port based on the channel number provided by the scheduler and on the port mux configuration of the egress mux.

Errored packets received on a slave DPI port flow through to the destination with an error indication aligned to the start of packet. All packets marked as errored by the MAPOTN subsystem are dropped by the CPB but counted by the MAPOTN. All errored packets due to overflow of a CPB queue are dropped and counted by the CPB.

The CPB contains 32 pages of memory with each page 192 KB. Each page contains 16 blocks for a block size of 12 KB. Queues are formed by first allocating pages to an input port / output port pair. A link list of blocks allocated from pages assigned to a port pair forms a queue as illustrated in [Figure 96](#). A queue can include blocks from multiple pages provided those pages are assigned to the input port / output port pair. The number of blocks allocated to a queue can be hitlessly reconfigured.

[Figure 96](#) CPB Queue Linked List of Page Blocks



The CPB subsystem features are:

- Store-Forward packet switch architecture
- 6 MB of storage for packet queuing
- Slave DPI ports for receiving packets from source clients
- Master DPI ports for pushing packets to destination clients
- Configurable for insertion of zeroed data (hole) aligned to the start of packet for header content populated by the SIFD as noted in [Section 5.11.4.5.6](#)
- Supports burst mode DPI transactions for the Interlaken interfaces (SIFD1 and SIFD2)
- Supports the transmission of a sideband "TYPE" indication that accompanies each packet transiting the packet buffer.
- Dropping of client marked packets. Counting of the dropped packets is done by the subsystem that marked them (MAPOTN, SIFD, CBRC, ENET(LINE) or ENET(SYS)).
- Interface supports reception of packets up to 12 KB in length
- Supports packet and segment mode of operation. The Interlaken interfaces (SIFD1 and SIFD2) operate either in segment or packet modes. The MAPOTN and COREOTN interfaces operate in segment mode. For the Interlaken interfaces, segment and packet modes can be independently configured in the add and drop direction.
- Channel, Link XOFF input for back-pressuring the CPB schedulers
- CPB scheduler supports channel XOFF on packet, word or burst boundary
- CPB scheduler supports link XOFF on packet, word or burst boundary
- Broadcast such that packets from a source channel can be copied to multiple queues. Broadcast requires additional queues due to the duplication of packets but does not consume egress bandwidth. The additional broadcast queues need to be accounted for when allocating queues from the available CPB storage space.
- FIFO zone indication for both slave and master DPI to monitor the relative fill level of the channel queues
- Dynamic and hitless resizing of channel queues
- Diagnostic PMON on slave DPI ports

### 5.10.2 Slave DPI Block and Performance Counters

The slave DPI block is replicated for each slave DPI port. The block provides DPI performance counters for monitoring activity on the DPI. For each DPI port, there are four independent DPI performance counters. These can be configured to count client marked dropped packets, good packets or all packets. Further filtering can be performed such that only packets associated with select channels are counted.

### 5.10.3 Sideband DPI "Type"

The CPB supports the transmission of a sideband "TYPE" indication that accompanies each packet transiting the packet buffer. The CPB does not interpret the content of the DPI TYPE field and is therefore agnostic to its usage. At a device level, the DPI TYPE bus is used for GFP-F mapped clients whose content can change dynamically on a packet-by-packet basis. This indication is transported in the UPI field of the GFP Type Header and is either terminated by the ingress/egress client interfaces or conveyed across Interlaken to a Network Processor. An example use of this functionality would be to distinguish between IPv4 and IPv6 for GFP-mapped IP packets over Interlaken. GSup43 7.3-mapped datapaths (ENET subsystem to GFP mapper) can also use this feature to differentiate between packets and ordered sets.

Table 55 DPI\_TYPE Use

Source	Destination	DPI_TYPE [Field]	Usage
SIFD (PIF)	MAPOTN (TGFPF96)	[15:8]	GFP UPI[7:0]
ENET120 (GSup437.3)	MAPOTN (TGFPF96)	[15:8]	GFP UPI[7:0]
MAPOTN (RGFPF96)	SIFD (PIF)	[15:8]	GFP UPI[7:0]
MAPOTN (RGFPF96)	ENET120 (GSup437.3)	[15:8]	GFP UPI[7:0]

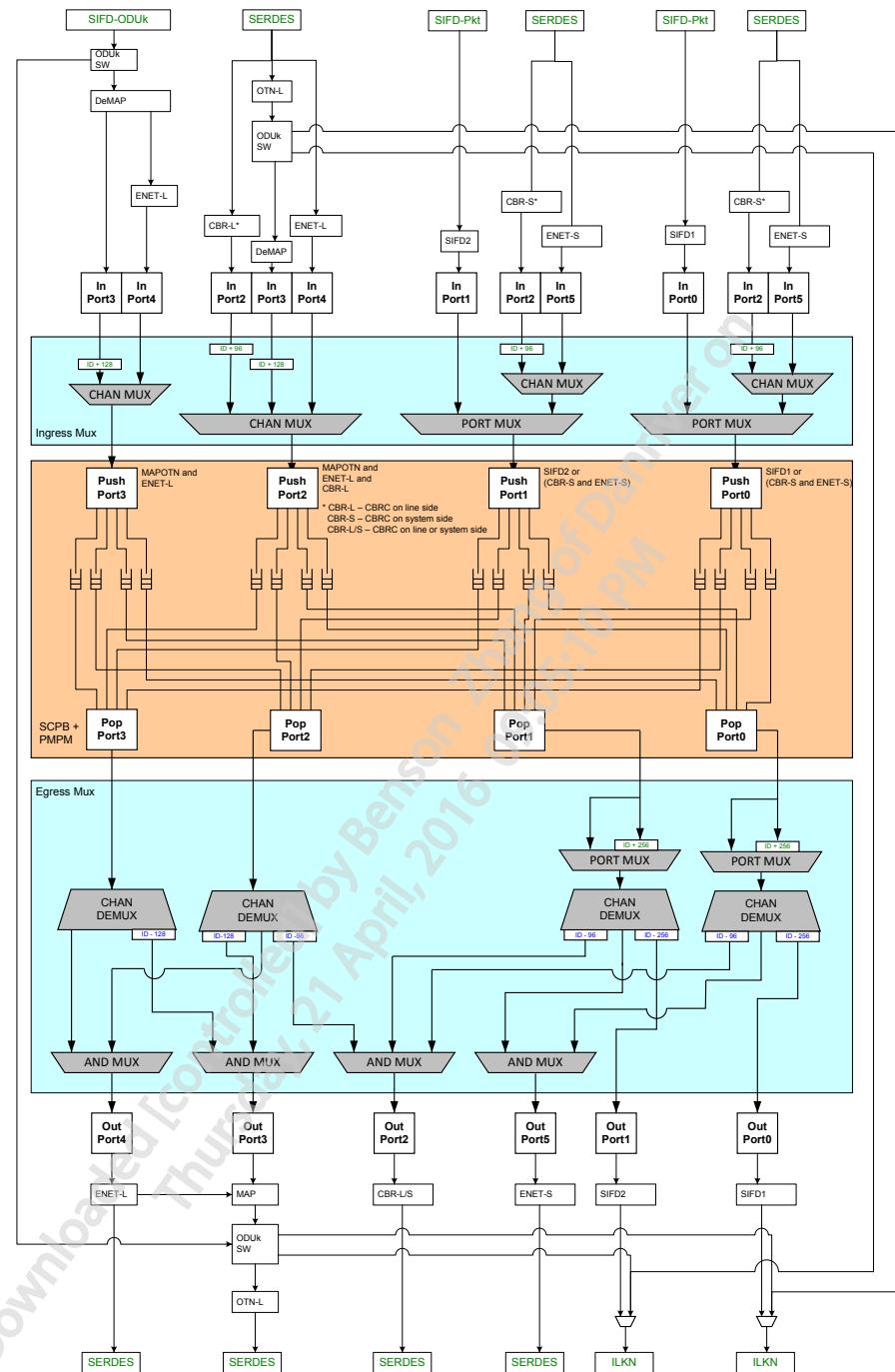
Note: All other bits of DPI\_TYPE[15:0] not defined in the above table are reserved for future use.

#### 5.10.4 Ingress and Egress Mux

The SCPB core provides four push ports for ingress packets and four pop ports for egress packets, as illustrated in [Figure 95](#). The ingress mux selects packets from the following subsystem sources (SIFD1, SIFD2, MAPOTN, CRBC line or system, ENET line and ENET system) and routes them to one of the four push ports. Similarly, the egress mux selects data from the four pop ports and routes them to one of the subsystem destinations. The ingress mux and egress mux provide the flexibility required to implement the supported datapath through the CPB given restrictions imposed by shared SERDES on the line and system interfaces.

[Figure 97](#) illustrates supported data routing options for the SCPB ports via the ingress and egress muxes. Data originates and terminates at the device line / system client SERDES or Interlaken interfaces via other device subsystems as shown in the diagram. Queue systems in the Programmable Multi-port Memory (PMPM) provide the ingress / egress port pair connectivity required to route data from any of the four SCPB input ports to any of the four SCPB egress ports.

Figure 97 SCPB Port Data Routing Options



The ingress mux selects packets using a combination of port and channel muxes. A port mux selects all channels for either of its DPI sources. A channel mux allows a mix of channels to be selected from either DPI source.

The ingress mux block receives six slave DPI streams from the slave DPI logic and aggregates these down to four streams before forwarding to the push interface of the SCPB:

- The SCPB Push Port 0 can be driven by channels from the SIFD1 slave DPI port or a mix of channels from the slave DPI ports of the CBRC and ENET subsystems. SIFD1 traffic cannot be mixed with CBRC or ENET system traffic. Only channels configured for SCPB Port P0 are forwarded; all other channels are ignored.
- The SCPB Push Port 1 can be driven by channels from the SIFD2 slave DPI port or a mix of channels from the slave DPI ports of the CBRC and ENET subsystems. SIFD2 traffic cannot be mixed with CBRC or ENET system traffic. Only channels configured for SCPB Port P1 are forwarded; all other channels are ignored.
- The SCPB Push Port 2 can be driven by mix of channels from the MAPOTN, CBRC, and ENET(LINE) subsystems. Only channels configured for SCPB Port P2 are forwarded; all other channels are ignored.
- The SCPB Push Port 3 can be driven by mix of channels from the MAPOTN, ENET(LINE) subsystems. Only channels configured for SCPB Port P3 are forwarded; all other channels are ignored.

The channel muxes in the ingress mux combine multiple channelized input streams into a single output stream. The input streams are simply ANDed together; the expectation being that only one stream is driving valid data during a timeslot. The channel mux requires the upstream schedulers of the source data to be programmed such that for any timeslot only one scheduler is transmitting data. When timeslot collisions occur, the channel mux will assert an interrupt and drop the data.

The egress mux selects data using a combination of port muxes, AND muxes, and channel demuxes. The port mux operates as per the port mux instances located in the ingress mux. The channel demux selects specific channels from a channelized input stream and directs packets from the selected channels to a single output. The AND mux combines multiple channelized input streams into a single output stream. The AND mux in the egress mux logic requires the upstream schedulers (the DCS instances) to be programmed such that for any timeslot only one stream is transmitting data. When timeslot collisions occur, the AND mux will assert an interrupt and drop the data.

The ingress and egress muxing options limit the SCPB ports that are simultaneously available to the device subsystems. For example: When using ENET System, the ingress and egress port muxes limit the CPB to 1 Interlaken interface being available. Similarly when ENET Line and ENET System are used, the CPB is limited to 1 Interlaken. Other restrictions may apply as a result of the port muxing.

**Table 56** provides a summary of the intended data paths through the egress mux for typical device applications. Note the DCS column of the table is referring to the scheduler associated with an egress port of the SCPB, and therefore, is equivalent to the SCPB pop port number. See [Section 5.10.5](#) for an explanation of the offsets that are added to or subtracted from DPI channel numbers.

**Table 56 DCS Applications**

Destination Summary	DCS Number	Destination <sup>2</sup>	Sources <sup>2</sup>	Applications
System-side Ethernet, CBRC when located on the system side, or SIFD when forwarding packet traffic	0 - 1	ENET_SYS	IF_L_xGE -> *ENET_LINE -> CPB IF_ILKN(PKT) -> SIFD -> CPB	ENET OEO Line Card <sup>1</sup>
		CBRC_SYS	IF_L_OTUn -> LINEOTN -> ODUkSW -> MAPOTN -> CPB	Transponder
		ENET_SYS	IF_ILKN(CBRI) -> SIFD -> ODUkSW -> MAPOTN -> CPB	Line Card <sup>1</sup>
		SIFD (Pkt)	IF_ILKN(CBRI) -> SIFD -> ODUkSW -> MAPOTN -> ENET_LINE -> CPB IF_ILKN(CBRI) -> SIFD -> ODUkSW -> MAPOTN -> CPB IF_L_10\40\100GE -> ENET_LINE -> CPB IF_L_OTUn -> LINEOTN -> ODUkSW -> MAPOTN -> ENET_LINE -> CPB IF_L_OTUn -> LINEOTN -> ODUkSW -> MAPOTN -> CPB IF_ILKN(CBRI) -> SIFD -> ODUkSW -> MAPOTN -> CPB IF_L_10\40\100GE -> ENET_LINE -> CPB IF_L_OTUn -> LINEOTN -> ODUkSW -> MAPOTN -> ENET_LINE -> CPB IF_L_OTUn -> LINEOTN -> ODUkSW -> MAPOTN -> CPB	Portless Server Card Line Card <sup>1</sup>
			IF_S_10\40\100GE -> ENET_SYS	Ported Server Card Line Card <sup>1</sup>

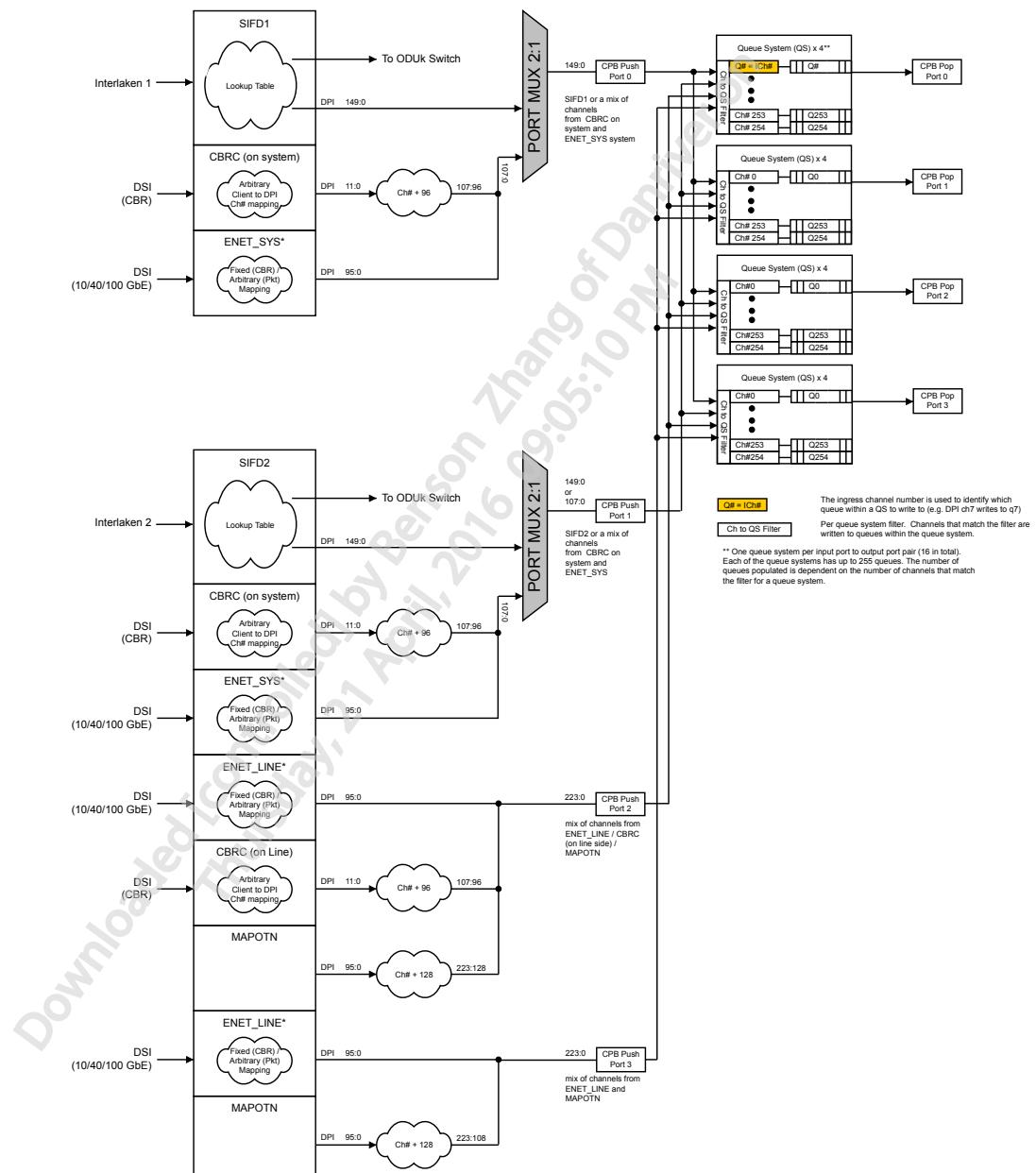
Downloaded [controlled] by Benson Zhang  
 Thursday, 21 April, 2016 09:00:00

Destination Summary	DCS Number	Destination <sup>2</sup>	Sources <sup>2</sup>	Applications
Traffic direct to SERDES	2	ENET_LINE	IL_S_xGE -> ENET_SYS -> CPB	ENET OEO
		ENET_LINE -> IF_L10\40\100GE	IF_ILKN(CBRI) -> SIFD -> ODUkSW -> MAPOTN -> CPB	Other to Packet / OTN to line-side SERDES
		CBRC_LINE -> IF_L_CBR	IF_ILKN(CBRI) -> SIFD -> ODUkSW -> MAPOTN -> CPB	
		ENET_LINE -> IF_L10\40\100GE	IF_ILKN(PKT) -> SIFD -> CPB	
		MAPOTN -> ODUkSW -> LINE-EOTN -> IF_L_OTUn	IF_ILKN(PKT) -> SIFD -> CPB	
		ENET_LINE -> MAPOTN -> ODUkSW -> LINE-EOTN -> IF_L_OTUn	IF_ILKN(PKT) -> SIFD -> CPB	
		MAPOTN -> ODUkSW -> LINE-EOTN -> IF_L_OTUn	IF_S10\40\100GE -> ENET_SYS -> CPB	
		MAPOTN -> ODUkSW -> LINE-EOTN -> IF_L_OTUn	IF_L10\40\100GE -> ENET_LINE -> CPB	
		MAPOTN -> ODUkSW -> LINE-EOTN -> IF_L_OTUn	IF_S_CBR -> CBRC -> CPB	
		MAPOTN -> ODUkSW -> IF_L_ILKN(CBRI)	IF_L_CBR -> CBRC -> CPB	Line Card
Traffic to back-plane via ODUk-SW	3	MAPOTN -> ODUkSW -> IF_L_ILKN(CBRI)	IF_L10\40\100GE -> ENET_LINE -> CPB	
		MAPOTN -> ODUkSW -> IF_L_ILKN(CBRI)	IF_S10\40\100GE -> ENET_SYS -> CPB	
		ENET_LINE -> MAPOTN -> ODUkSW -> IF_L_ILKN(CBRI)	IF_ILKN(PKT) -> SIFD -> CPB	
		MAPOTN -> ODUkSW -> IF_L_ILKN(CBRI)	IF_ILKN(PKT) -> SIFD -> CPB	
		MAPOTN -> ODUkSW -> IF_L_ILKN(CBRI)	IF_ILKN(PKT) -> SIFD -> CPB	
<p><b>Note:</b></p> <ol style="list-style-type: none"> <li>1. Using System-side instead of Line-side.</li> <li>2. See the device Product Overview <a href="#">[1]</a> for datapath details.</li> </ol>				

## 5.10.5 Channel Mapping

Figure 98 illustrates the channel mapping through the ingress mux to queues within the PMPM. DPI channel numbers for packets arriving from the Interlaken interface are derived from the channel number in the burst control word. Lookup tables use channel numbers from the Interlaken BCW to direct packets arriving on the Interlaken interface to the CPB or ODUk switch, as described in [Section 5.11.4.5.4](#).

Figure 98 CPB Ingress Channel Mapping

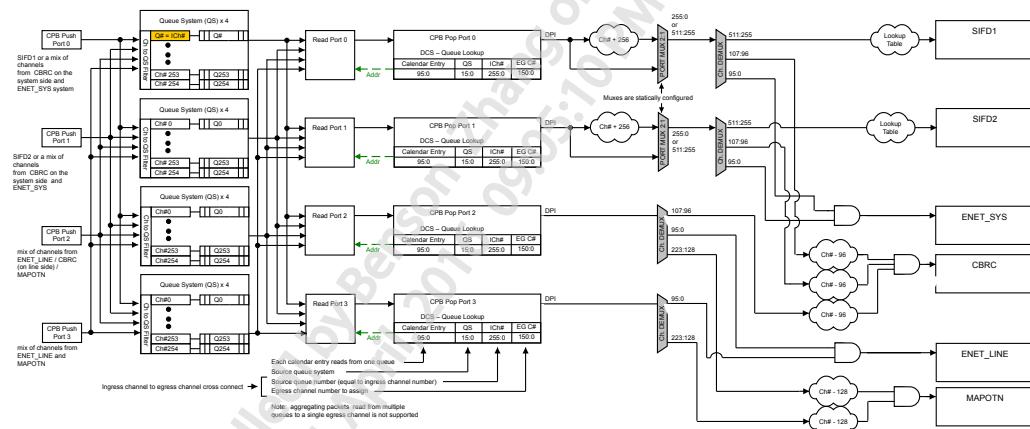


Port muxes in [Figure 98](#) select a DPI port to forward to a CPB push port. The port muxes are statically configured by software. Channel numbers from the CBRC and MAPOTN subsystems are offset by 96 and 128, respectively to allow their DPI to be merged with the ENET(LINE) or ENET(SYS) DPI. CPB push port 0 receives either SIFD1 packets or a mix of channels from the CBRC located on the system side and ENET(SYS) subsystems. CPB push port 1 receives either SIFD2 packets or a mix of channels from the CBRC located on the system side and ENET(SYS) subsystems. CPB push port 2 receives packets from a mix of channels from the ENET(LINE) subsystem, MAPOTN subsystem, or CBRC subsystem when located on the line side. CPB push port 3 receives packets from a mix of channels from the ENET(LINE) subsystem, MAPOTN subsystem, or CBRC located on the line side.

Each CPB push port directs the DPI bus to four queue systems, one queue system per CPB pop port. A total of 16 queue systems are supported. A user-configured filter selects which channels to write to each queue system. The channel number is used to specify the queue number to populate, which creates a one-to-one correspondence between the ingress channel number and the queue number.

[Figure 99](#) illustrates the channel mapping from the read side pop ports of the queue systems through to the device subsystem DPI ingress ports.

**Figure 99 CPB Egress Channel Mapping**



Schedulers associated with each pop port select the egress packets based on the user-configured source queue and system ingress channel numbers. The ingress channel number is equivalent to the queue number within a queue system. The DPIs for pop ports 1 and 2 are duplicated to allow the channel numbers to be offset by 256 for one copy of the traffic. Statically-configured port muxes then allow packets to be directed to either the SIFD interfaces, where the 256 channel offset is removed, or to the CBRC and ENET(SYS) subsystems.

The channel demux blocks of [Figure 99](#) filter packets based on a channel number range. Channels in the range 223:128 are directed to the MAPOTN subsystem, channels in the range of 107 to 96 are directed to the CBRC subsystem, and channels in the range 95:0 are directed to the ENET(LINE) or ENET(SYS) system. The offsets of 128 and 96 are subtracted from the channel numbers for DPIs destined to the MAPOTN and CBRC subsystems, respectively. The “AND” functions allow a subsystem to receive a mix of packets that originate from different queue systems.

### 5.10.6 Scalable Central Packet Buffer (SCPB)

The SCPB is a multi-port, non blocking storage switch for queuing of packet data to a queue system where a queue system is associated with a port pair from one of the four SCPB ingress ports to one of the four SCPB egress ports. The SCPB supports a total of 16 queue systems in the PMPM.

Packets are enqueued to a queue system by making push requests on a push port. Data on the push interface along with an associated queue ID is presented to each of the attached queue systems. Individual queue system are programmed, on a per queue basis, to accept or reject data from a particular push interface. The queue storage memory is not contained within the SCPB, rather a multi-port memory interface is provided for connection to the PMPM.

Packet are dequeued from a queue system by making pop requests to a data queue. The pop request is accompanied by queue ID and associates the request with a particular queue system. The targeted queue system will retrieve the queues read pointer, read the packet from the queue storage memory, and advance the read pointer.

The SCPB features include:

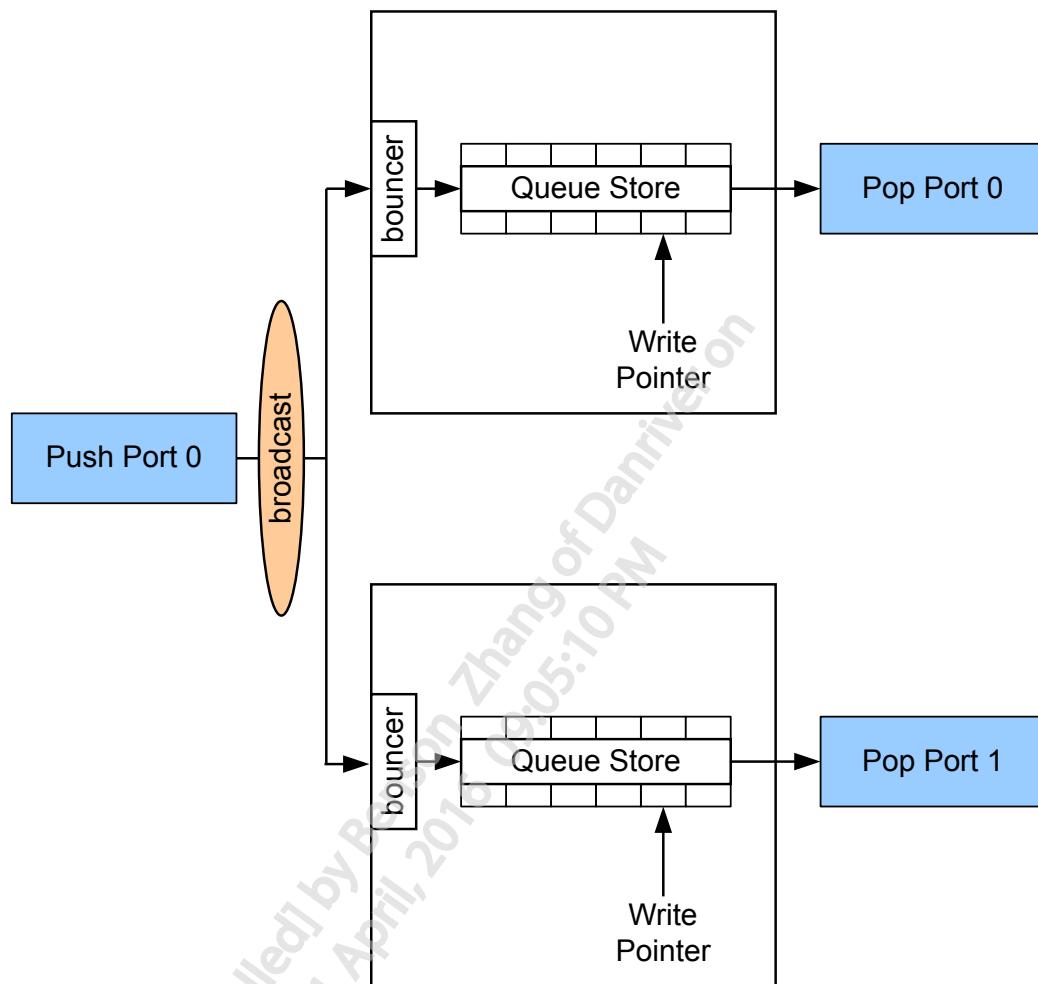
- Four push ports for enqueue of packet data
- Four pop ports for dequeue of packet data and packet length
- Queue zone interface for each Push and Pop port with programmable thresholds for monitoring of queue fill levels
- Support packets up to 12 KB in length
- Optional dropping or truncating of over-sized packets. Dropped packets caused by queue fill are counted.
- Provides an independent queue system (16 in total) connecting each push port and pop port
- Each queue system supports a configurable number of queues; up to 256
- Broadcast such that a single push can reach multiple queue systems
- Hitless resizing of individual queues via register interface

### 5.10.7 Broadcast of Packet Data

The CPB provides broadcast of packet data to multiple queues. The broadcast feature allows packets from a source channel to target multiple DPI master ports. The CPB does not support multicast. Note broadcast causes storage expansion as the data is replicated for each target DPI master port.

For broadcast, a SCPB push port receives a packet and forwards to multiple queues system. This is done by programming a queues system to accept packets marked with a queue ID from a specific push port. A packet is broadcast when multiple queue systems are configured to accept the packet. This feature leads to storage expansion as the packet is duplicated in the queue storage.

Figure 100 Broadcast



### 5.10.8 Memory Allocation

The SCPB is attached to a shared memory for storage of packet data. This shared memory is segmented into pages of memory, and the pages are segmented into blocks of memory. The SCPB provides configurable page table and block table for allocation of memory. The page table is used to allocate a page of memory to a push and pop port pair. The block table is used to allocate a block of memory to a particular queue within the port pair.

### 5.10.9 Zone Interface and Flow Control

The CPB allows destination clients to back pressure the scheduler driving the master DPI port. Destination clients can issue link XOFF or channel XOFF.

The CPB can be configured to enter the channel XOFF state, as signaled by the destination client, on a segment, burst or packet boundary. Once a channel has been requested to enter the XOFF state and after waiting for the appropriate transmission boundary, the CPB will associate the channel to an internal queue and the scheduler will cease to request data from that queue.

The CPB can be configured to enter the link XOFF state, as signaled by the destination client, on a segment, burst or packet boundary. Once a link has been requested to enter the XOFF state and after waiting for the appropriate transmission boundary, the CPB will instruct the scheduler not to schedule any queues associated to the link. A link can contain all the channels of a master DPI port, or a grouping of channels.

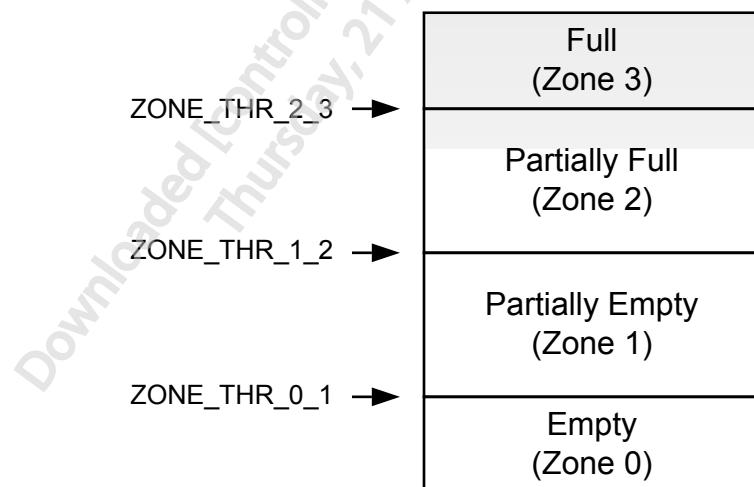
The SCPB zone interface is used to transmit the relative queue fill level when compared against a programmable threshold. Each queue has three programmable thresholds (48-byte granularity) which creates four queue zones: Full, Partially Full, Partially Empty, Empty. The SCPB provides a zone interface for each pop interface and a zone interface for each push interface. Zone signaling includes a zone 'level' and a Zone ID. The Zone ID represents the queue ID for which the 'level' applies to. For a pop zone interface, the SCPB will send out a Zone update associated to a queue whenever the queue being popped is crossing into a new zone, else the SCPB will round robin through each of the enabled queues and transmit the zone for each. Similarly for a push zone interface, the SCPB will send out a zone update associated to a queue whenever the queue being pushed is crossing into a new zone, else the SCPB will round robin through each of the enabled queues and transmit the zone for each.

When performing a broadcast push operation, multiple queues can simultaneously be crossing into a new zone. In this scenario a primary queue is identified through register interface configuration. Only the zone of the primary queue is updated on the push zone interface, all associated broadcast queue zones are ignored.

The zones supported are: Zone 0 "Empty", Zone 1 "Partially Empty", Zone 2 "Partially Full", Zone 3 "Full".

- A queue is in Zone 0 when the queue fill level contains `ZONE_THR_0_1` or fewer entries
- A queue is in Zone 1 when the queue fill level contains  $(ZONE\_THR\_0\_1 + 1)$  to `ZONE_THR_1_2` entries
- A queue is in Zone 2 when the queue fill level contains  $(ZONE\_THR\_1\_2 + 1)$  to `ZONE_THR_2_3` entries
- A queue is in Zone 3 when the queue fill level contains  $(ZONE\_THR\_2\_3 + 1)$  or greater entries

Figure 101 FIFO Zone Indication



The zone thresholds are checked from highest to lowest. For example if the fill level exceeds ZONE\_THR\_2\_3, the queue is considered to be in "Zone 3" regardless of the ZONE\_THR\_1\_2 value being less, equal or greater than ZONE\_THR\_2\_3.

### 5.10.10 Channel XOFF to FIFO Zone Cut-through

The CPB provides channel XOFF to FIFO zone cut through. The zone cut through allows an external network processor connected to the Interlaken interface to perform the buffering required for reach extension should the amount of buffering provided by the CPB be insufficient for a given application.

Upon assertion of channel XOFF on a master DPI port, the CPB can be programmed to force a FIFO full condition on the zone interface of the associated slave DPI port. The CPB can be programmed for this behavior on a per channel basis. When forcing a FIFO full condition via channel XOFF, the CPB will continue to schedule packets for the XOFF channel. In other words, the XOFF indication in this mode does not impede the data flow through the CPB. The queue will continue to accept packets and update the fill level. Packets will not be dropped and counted until the queue reaches the queue full threshold.

### 5.10.11 Dropping and Counting of Packets

Packets can be dropped under the following three conditions:

1. A packet is over-sized.
2. A packet is marked by the client and a drop indication asserted at the end of packet.
3. A packet is received and the target queue is Full.

Packets are counted when packets dropped because the queue is in the Full. Each queue has an associated 32-bit saturating packet full drop counter. The counter is incremented each time a packet is dropped due to queue full. The counter is a saturating counter that can be read and cleared using device registers.

### 5.10.12 Dynamic Resizing of Queues

The SCPB queue system allows for on-the-fly resizing of queues without affecting the data currently stored within the queue. Resizing does not conflict with incoming or outgoing queue data. Users can select to increase or decrease the queue size through register interface control registers.

When deallocating a block of memory from a queue, the user must first decrease the queue end address such that the memory block is unused and further mark the page table entry associated to the block with a NULL value.

When allocating a block of memory to a queue, the user must first locate a unallocated block by searching the page table, once a NULL entry has been found the user can program the table to associate the block to a queue. Once the page table is programmed the user must increase the queue end address, effectively growing the queue.

### 5.10.13 Dynamic Remapping of Memory Allocation

The SCPB supports a feature whereby the allocation of a block of memory to a queue can be changed on the fly and in hitless fashion. This feature allows the user to swap one block for another, and provides the capability to migrate blocks from one page to another. The ability to migrate blocks to a new page is important as it allows the user to potentially free up a partially used memory page.

Within the block table, blocks are allocated to a queue. To remap a block, the user must select a currently unallocated block and program the block table with the same values as the allocated block to

be remapped. The users then makes a request through the configuration register, and the SCPB indicates when the operation is complete and successful. For the remap to complete successfully, the queue fill must be less than full - block size.

### 5.10.14 Digital Common Scheduler and Pop Controller

The pop control block is responsible for generating dequeue requests to a pop interface of the SCPB. There is one pop control block for each SCPB pop interface. The pop control block contains a DCS (Digital Common Scheduler). At each cycle the pop controller selects a queue via one of the DCS supported arbitration policies and issues a dequeue request with a queue ID to the SCPB. With each dequeue request, the SCPB also outputs a channel ID for the downstream egress mux to steer the data to the appropriate DPI port and to signal the DPI.

The DCS is a scalable calendar-based scheduler. It is responsible for arbitrating traffic from up to 255 queues. When a particular queue is selected to transmit its data depends on how the calendar is programmed, whether the link and channel are enabled and of course if there is data available to send.

The DCS is a weighted round robin (WRR) scheduler which blindly selects queues in the sequence programmed within the calendar pages. The DCS can be configured to select the queues in two modes: segment (one clock cycle at a time) or packet (as many clock cycles as required to complete the packet), and both of these can have burst enabled (so the queue is selected in multi-cycle bursts). The DCS supports the following features.

- Supports up to 255 queues
- Weighted round robin scheduling using a 256 entry calendar memory
- Queue to Channel mapping for WRR queues
- Mapping of multiple queues to a single channel is supported, but not multiple channels to a single queue
- Support for segment and packet modes, each of which can be configured to be in burst mode
- Channel and Link XOFF support
- Synchronization of WRR calendar via external trigger
- Duplicate Calendar for on the fly scheduler reconfiguration

The burst / segment / packet configuration for the DCS associated with each pop port is as follows:

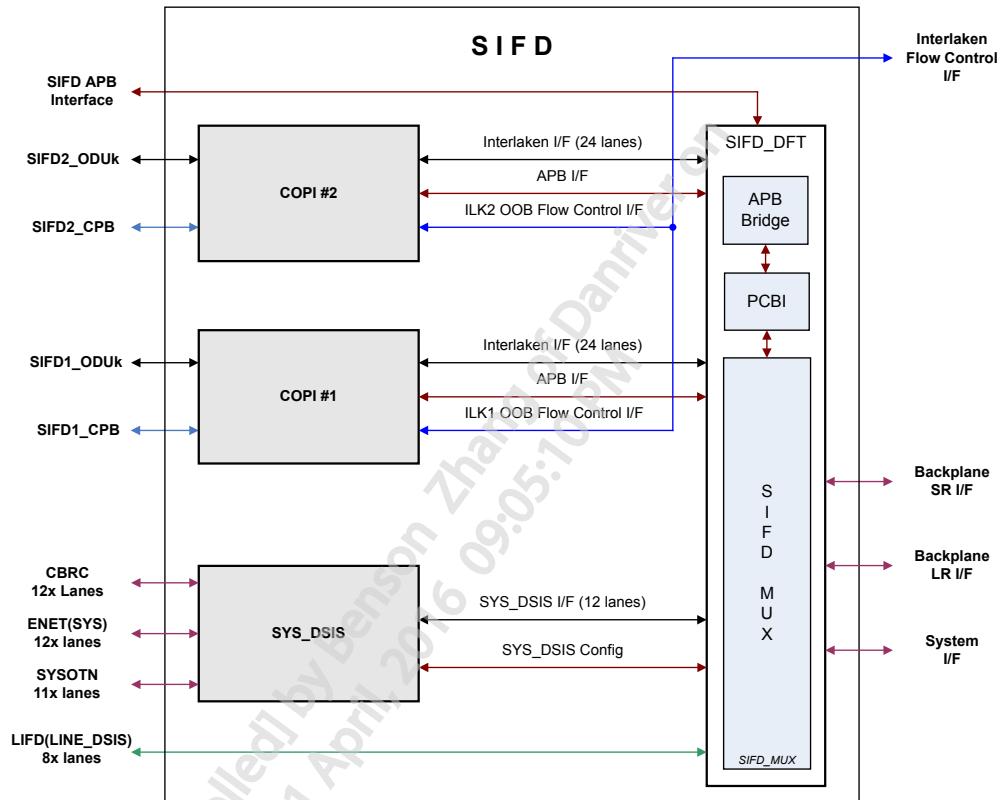
- DCS0/1: burst / segment (Interlaken segment) or burst / packet (Interlaken packet)
- DCS0/1 to CBRC/ENET\_LINE: non-burst segment
- DCS2: always segment non-burst
- DCS3: always segment non-burst

Note: Ethernet is packet-contiguous per link but links are interleaved on segment basis.

## 5.11 System Interface (SIFD)

### 5.11.1 Block Diagram

Figure 102 SIFD Block Diagram



### 5.11.2 Overview

The SIFD transports packets to/from the CPB (Central Packet Buffer) and the ODUk switch over an Interlaken interface. The SIFD contains two identical COPI (Combined OTN and Packet Interface) blocks. One side of the COPI block connects one port of the CPB and one port of the ODUk switch, while the other side connects to an Interlaken interface.

The Interlaken protocol is implemented through the Interlaken block. On the transmit side, packets received from the CPB and the ODUk switch are sent to the Interlaken block. The block applies the Interlaken protocol and stripes the data across a software configured number of SERDES lanes. The Interlaken block provides an interface that allows it to be connected to SERDES blocks outside the SIFD. On the receive side, the data received from SERDES is decoded and checked for protocol compliance. The data is aggregated based on the channel number and sent to either the CPB or the ODUk switch. Interlaken channel numbers greater than programmable channel number (default = 150) are destined for ODUk switch. A programmable channel offset is subtracted from the Interlaken channel ID before sending the channel ID to the ODUk switch. The Interlaken block supports a maximum of 24 SERDES lanes and interfaces to 3.125/6.25/10.3125/11.5/12.5 Gbps rate SERDES.

The interface to the CPB and the ODUk switch is through a DPI. The CPB and ODUk switch push packets into the SIFD in short bursts. The burst sizes are identically configured in the SIFD, CPB and the ODUk switch by software.

Based on the signaling during the start of packet, the SIFD can prepend CPB-sourced packets with a 4-byte optional header in the transmit direction (from the device to an external device). The optional header carries packet type information (TYPE[15:8]) provided by the CPB specifying a UPI field from the GFP de-mapper. The UPI field is used to identify ordered sets versus data for the GSup43 7.3 mapping of packets. This mode of operation also enables the remote packet source to specify the UPI field for packets entering the GFP mapper. Conversely, the UPI value in the TYPE[15:8] field allows the remote packet sink to determine the GFP UPI associated with each de-mapped packet. In the receive direction (from an external device to the device), the fourth byte of a packet is extracted and transmitted on the TYPE[15:8] field as a sideband signal to the CPB.

The SIFD can be operated in either packet mode or segment mode, configurable for each DPI port. In packet mode, the CPB and ODUk switch sends a complete packet on a channel before switching to another channel. In segment mode, a packet is divided into a small programmable burst by the CPB and the ODUk switch. Then, the CPB and ODUk switch sends a complete burst on the channel before switching to another channel. The SIFD always prioritizes the packets received from the ODUk switch over packets from the CPB when transmitting out through the Interlaken block. On the receive side, the SIFD simply sends the data it receives from the Interlaken interface. It does not create packets or burst out of the received data. Other interfacing devices are responsible to honor packet or segment mode accordingly.

**Note:** When the CPB traffic is in segment mode and the ODUk traffic is in packet mode, the packet mode channel change error interrupt (RX\_PKT\_CH\_CHANGE\_ERR\_INT) interrupt for the Interlaken interface will incorrectly fire and should be ignored. Additionally, when the Interlaken burst short is set to 32B, spurious activity will be generated on the RX\_PKT\_CH\_CHANGE\_ERR\_INT interrupt. This interrupt should be ignored under these conditions.

The SIFD implements in-band and out-of-band flow control mechanisms on the Interlaken interface. However only one of the flow control mechanisms is active at a time. Selection of the flow control mechanisms is programmable. Both flow control mechanisms are calendar-based and support flow control on a channel-by-channel basis or the entire interface.

On the receive side, the SIFD does not have a per-channel buffer. Any channel level flow control from the CPB and ODUk switch is transmitted over the Interlaken interface and the CPB and the ODUk switch must absorb the data received during the flow control response time. Any link-level flow control from the CPB and ODUk is responded to by the SIFD within 20 ns. Flow control is transmitted over the Interlaken interface. The data received during the flow control response time is absorbed by the SIFD.

On the transmit side, the SIFD does not have any per-channel buffers. Any channel level flow control from the Interlaken interface is passed on to the CPB and the ODUk switch depending on the Interlaken channel number. The ODUk switch and CPB are expected to stop on the packet boundary for channel level flow control to avoid any head of line blocking. Link level flow control from the Interlaken interface is responded to by the SIFD and subsequently passed on to the CPB and ODUk switch depending on the FIFO threshold.

The SIFD\_MUX block allows the COPI blocks (inside the SIFD), and the CBRC, ENET(SYS) blocks, LINEOTN block, and the SYSOTN (outside of the SIFD) to be connected to the backplane LR (40-lane SERDES), system (16-lane SERDES), and backplane SR (8-lane SERDES) blocks. The SIFD provides a serial interface (DSI) to the system, backplane SR, and backplane LR SERDES. Similarly, DSI ports are provided to the ENET(SYS) and CBRC subsystems. The SIFD also connects the SYSOTN interface to be transmitted over the SERDES.

The SIFD implements an ACB (Adaptive Clock Block) mux block that allows the ACB clock control signals driven by the CBRC and the ENET(SYS) subsystems to be connected to the ACBs associated with the system SERDES blocks.

The SIFD provides an AMBA APB (Advanced Peripheral Bus) interface to allow it to be configured, controlled and monitored by other masters in the device such as the internal CPU.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:10 PM

## 5.11.3 Features

### 5.11.3.1 Feature Overview

- Supports a common packet interface transporting packets to/from the CPB and the ODUK switch over an Interlaken interface.
- Supports two Interlaken interfaces.
- Supports fixed association between the Interlaken interfaces to the CPB and ODUK switch through a pair of DPIS.
- Supports buffering to handle rate mismatch between the Interlaken transfer rates and the transfer rate across the CPB/ ODUK switch and the SIFD interface.
- Provides a transmit feature (CPB and ODUK  $\Rightarrow$  Interlaken):
  - Checks for channel number associated with the packets received from the ODUK switch and the CPB interface to be within the programmable range. Any packets that are out of range will be dropped and counted.
  - Assigns a programmable offset to the channel number associated with the packets received from the ODUK switch and the CPB interface before sending to the Interlaken interfaces.
  - Optionally prepends the packet with the { 24'b0, TYPE[15:8] } field provided by the CPB. TYPE[15:8] carries UPI field from the GFP de-mapper.
  - Provides absolute priority to packets from the ODUK interface over CPB interface.
  - Supports segment mode or packet mode on both the CPB and the ODUK switch interfaces. The modes can be selected independently for either the CPB or ODUK switch.
  - Provides channel level and link level flow control on the Zone interface and transmits over the Interlaken interface using an in-band or out-of-band flow control mechanism.
- Includes the following receive features (Interlaken  $\Rightarrow$  CPB & ODUK):
  - Packets received on an Interlaken interface are sent to CPB or the ODUK switch based on the incoming Interlaken channel number. Packets on channel numbers greater than the programmable channel number are sent to ODUK switch.
  - The SIFD subtracts a programmable offset from the Interlaken channel number before sending to the CPB and ODUK switch interface.
  - Extracts the fourth byte of the packet and communicates to the CPB via TYPE[15:8] (used by the GFP mapper as an optional source for the UPI).
  - Provides Interlaken interfaces that:
    - Conform to Interlaken Protocol Definition version 1.2 [25]
    - Complies with relevant sub-clauses of Interlaken Interoperability specification version 1.6 [26]
    - Includes a software-configurable number of SERDES lanes to a maximum of 24 lanes.
    - Supports a connection to SERDES at 3.125/6.25/10.3125/11.5/12.5 Gbps rates.
    - Provides support for a maximum of 254 channels.
    - Supports segment mode or packet mode.
    - Supports Burst Max parameters of 128/192/256 bytes.
    - Supports Burst Short parameters of 32 or 64 bytes.
    - Provides In-band and Out-of-Band flow control (only one can be active at a time).
    - Provides a 300-entry flow control calendar. Some entries are used to convey link level flow control information other than the 256 channels.
    - Collects statistics on one selected Interlaken channel.
    - Collects statistics per SERDES lane.

- Does not support a latch clock (LCLK) snapshot for the per selected channel and per SERDES lane statistics.
- Provides loopback support on the Interlaken OOB flow control interface.
- Provides transmit to receive loopback and receive to transmit loopback paths on data.
- Supports PRBS test pattern generation and checking on Interlaken lanes (31/23/7 or programmable polynomial).
- Supports arbitrary lane-reordering.
- Provides configurable error injection on transmit for improved testability.
- Requires non-eop bursts to be aligned to 32-byte boundaries.
- Muxes SERDES interfaces of the two Interlaken interfaces, ENET(SYS), LINEOTN, SYSOTN and CBRC to the backplane LR, system and backplane SR SERDES blocks:
  - Provides static re-ordering of the ENET(SYS) lanes, allowing any lane from the ENET(SYS) to connect to lanes [11:0] of the system SERDES.
  - Provides static re-ordering of the CBRC interface and connection of the selected lanes to lanes [11:0] of the system SERDES.
- Uses SYS\_DSIS to mux ACB clock control signals driven by the CBRC, SYSOTN and ENET(SYS) subsystems to the ACBs associated with lanes [11:0] of the system SERDES.

The Interlaken standard allows for the size of non-EOP bursts to be a multiple of 8-bytes. The receive interface of the PMC device Interlaken interface requires non-EOP bursts to be aligned on 32-byte boundaries. In addition, the RX\_PKT\_CH\_CHANGE\_ERR\_INT\_IO interrupt will incorrectly fire with 32-byte non EOP packets are received from Interlaken and the interrupt should be disabled if the mate device is expected to send non EOP bursts that are 32-bytes.

The device does not support receiving packets under the following scenarios:

1. The first burst carrying the start of the packet to be transferred is *48-bytes and not the end of the packet*.
2. The burst short size is configured for 32B.
3. Header stripping in the CPB is used.

Header stripping in the DCPB is used to remove the 4-byte COPI TYPE header in the fabric egress direction; i.e. when the UPI value is passed in-band across the Interlaken.

**Note:**

1. In the GFP mapper direction, the COPI TYPE header does not need to be stripped since the CPB must add a net 8-byte hole to the packet. The existing COPI TYPE header is used as 4-bytes of the hole and the egress of the DCPB adds an additional 4-bytes. This is the required configuration when the in-band UPI is needed for inclusion in the GFP frame.
2. In the GFP mapper direction, if COPI TYPE header is present, but the UPI value is not used, then header stripping may be enabled.
3. In the ENET MAC direction, COPI TYPE header information is not used. In the event that the COPI TYPE is provided, it must be stripped at the DCPB.
4. GSUP43 7.3 encoded packets from the PIF to a 10GE port via the EPMM are not supported by the device.

This is a corner case that is not expected to occur when the device is receiving packets that are scheduled using the Interlaken enhanced scheduling algorithm but should be considered for applications that use different scheduling algorithms.

Devices that implement the optional scheduling enhancement defined in the Interlaken Protocol Definition v1.2 will inter-operate with the PMC-Sierra device since BurstMin is defined as a multiple

of 32-bytes. Note that the enhanced scheduling algorithm is not supported when transmitting burst to a mate device. Receiving packets that are greater than one burst in size and are segmented into bursts using the Interlaken enhanced scheduling by a mate device is supported.

External devices that do not support transmission of bursts using the Interlaken enhanced scheduling algorithm are likely to always send non-EOP bursts that are BurstMax in size and will inter-operate with the PMC-Sierra device.

Devices that implement a non-standard enhanced scheduling algorithm need to consider the 32-byte boundary restriction of non-EOP bursts for the PMC-Sierra device.

**Note:**

The device requires link level flow control per the Interlaken Protocol Definition, Revision 1.2. Link level flow control may occur for packet traffic when packets less than 65 bytes are received due to the baseline small packet throughput of the device. The communal FIFO used to buffer packets from all channels is sized such that one calendar entry per burst control word must be used to report link level flow control.

### 5.11.3.2 Statistics

The SIFD subsystem allow statistics to be collected on a single Interlaken channel on each Interlaken interface and on each SERDES lane on each Interlaken interface. The statistics collection is implemented in the COPI. The counters that count byte level statistics are 56-bits wide, the counters that count packet level statistics are 48-bits wide and the counters that count error events are 24-bits wide.

Statistics collected on one selected Interlaken channel (per Interlaken interface) are:

- Number of packets received
- Number of bytes received
- Number of packets transmitted
- Number of bytes transmitted
- Number of packets that are errored (i.e. Bad CRC, FIFO overflow, ERR bit set, etc.)
- Number of packets dropped due to receive FIFO overflow
- Number of bursts with a detected CRC error
- Number of errors detected on the out-of-band flow control interface
- Number of bursts received longer than the BurstMax parameter
- Number of bursts smaller than the programmed BurstMin and BurstShort parameter

Statistics collected on each SERDES lane of the Interlaken interfaces are:

- Number of alignment sequences received in error (i.e., those that violate the current alignment)
- Number of times alignment was lost (after four consecutive alignment sequence errors)
- Number of times a lane lost word boundary synchronization
- Number of times a lane lost clock-data-recovery
- Number of errors in the lane CRC
- Number of words received with Control Word framing that don't match one of the defined Control Words

Statistics are maintained for packets dropped in the SIFD:

- 32-bit packet counters provided separately for CPB and ODUk interface. Provided separately for transmit and receive paths.

### 5.11.3.3 Low-power Mode

The COPI block can be put in low power mode. In low power mode, the block is not functional except for the APB interface. The block must be kept in reset when in low power mode. The clocks in the block are disabled (except for the APB interface) and the RAMs are put in low power mode. The COPI block also allows the receive buffers for ODUk switch or the CPB to be put in low power mode.

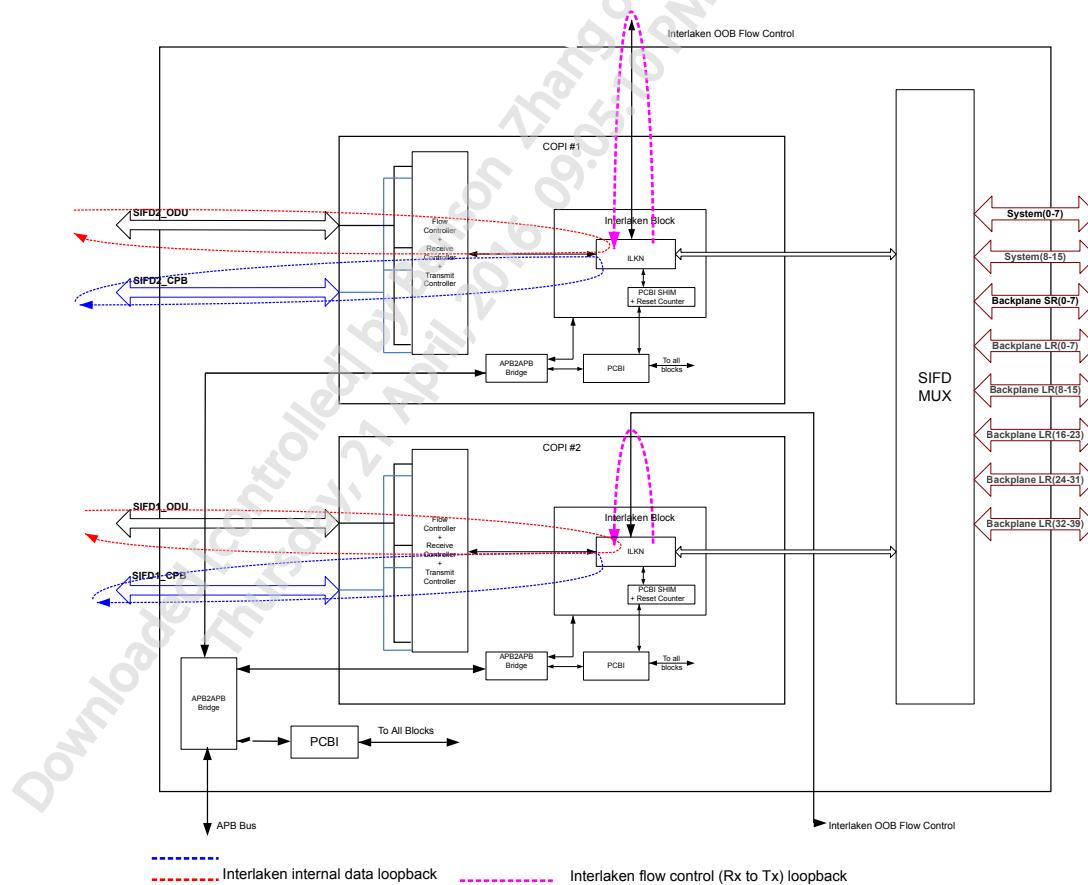
### 5.11.3.4 Test Features

#### 5.11.3.4.1 Interlaken Internal (Transmit to Receive) Data Loopback

The data sent by the CPB and ODUk switch is looped in the Interlaken block back to the CPB and ODUk switch. The data is also sent by the Interlaken block on the SERDES interface. Software enables this loopback by programming a register in the Interlaken block. The data can optionally not be sent to the SERDES interface by programming another register in the Interlaken block.

These loopbacks are shown in [Figure 103](#) below.

**Figure 103** Interlaken Internal Loopback/Flow Control Tx-Rx Loopback

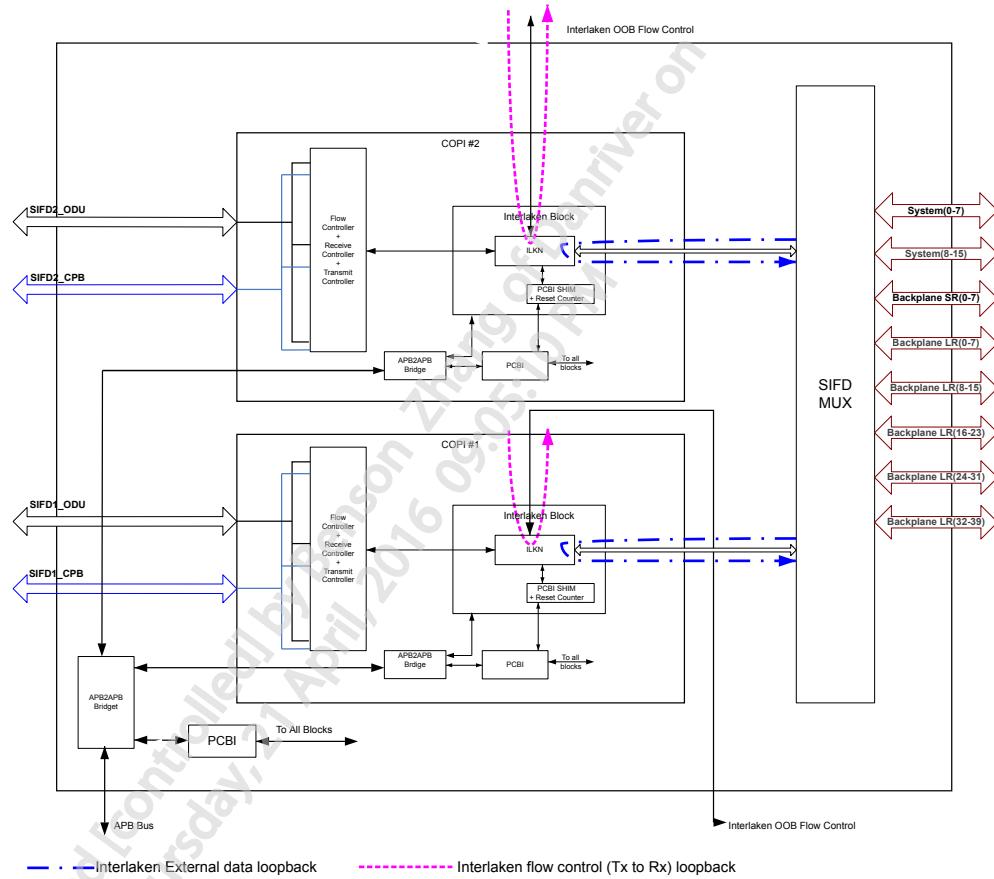


### 5.11.3.4.2 Interlaken External (Receive to Transmit) Data Loopback

The data received from the SERDES lanes are sent back on the SERDES lanes regardless of any error checking. The data is also sent to CPB and ODUK switch with normal error checking and handling. Software enables this loopback by programming a register in the Interlaken block. The data can optionally not be sent to CPB and ODUK switch by programming another register in the Interlaken block.

These loopbacks are shown in Figure 104 below.

Figure 104 Interlaken External Loopback/Flow Control Rx-Tx Loopback



### 5.11.3.4.3 Interlaken Flow Control (Transmit to Receive) Loopback

The flow control that is sent out of the Interlaken interface is exactly what was received from the Interlaken interface, regardless of any error checking, programmable calendar, or other configuration.

This loopback is used in conjunction with the Interlaken External data loopback. The flow control (Tx to Rx) loopback allows the receive side the remote receiver to throttle the remote transmitter. This prevents receiver overflows and therefore allow the Interlaken External data loopback to work successfully.

Software enables this loopback by programming a register in the Interlaken block. In addition to being sent to the Interlaken interface, the received Out-of-Band Flow Control is also sent through to the user

with normal error checking and calendar processing. This path can be disabled by programming another register in the Interlaken block. These loopbacks are shown in [Section 5.11.3.4.1](#).

#### **5.11.3.4.4 Interlaken Flow Control (Receive to Transmit) Loopback**

The flow control generated by the SIFD towards Interlaken interface is looped back to the SIFD. Software enables this loopback by programming a register in the Interlaken block. These loopbacks are shown in [Section 5.11.3.4.2](#).

#### **5.11.3.4.5 Test Pattern Generation**

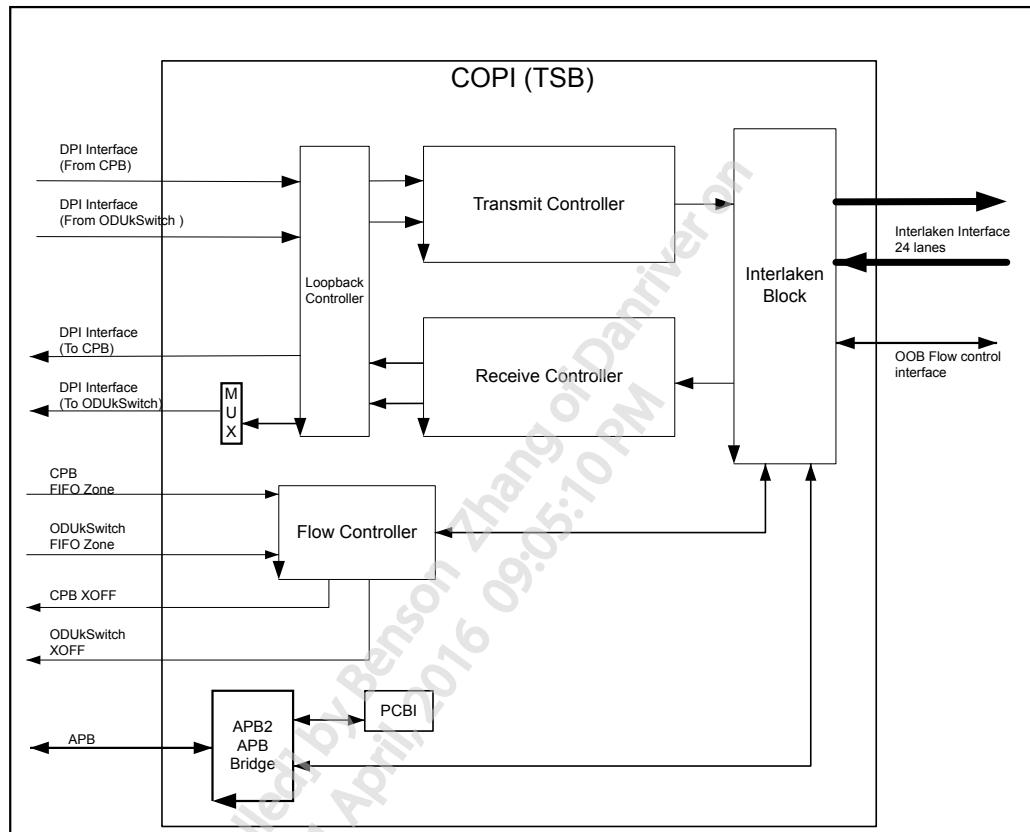
The SIFD supports test pattern generation and reception on the Interlaken lanes as required by the Interlaken protocol. The SIFD implements a PRBS 31/23/7 generator and checker and a programmable test pattern generator and checker. The programmable test pattern generator implements registers that contain the patterns (A, B, and C) and three registers REPA, REPB, REPC contain the number of repeats for Patterns A, B, and C. The programmable test pattern generator creates a bit stream of Pattern A for REPA times, followed by Pattern B for REPB times, followed by Pattern C for REPC times, and then back to Pattern A to repeat the entire sequence. Either PRBS 31/23/7 pattern generator or the programmable test pattern generator can be used at a time. The selected pattern generator and the pattern apply to all Interlaken lanes. On the receive side, the SIFD checks for the errors in the received pattern and updates error counters on errors.

Downloaded [controlled] by Benson Zhang  
Thursday, 21 April, 2016 09:05:10 PM

## 5.11.4 Combined OTN and Packet Interface (COPI) Block

### 5.11.4.1 Block Diagram

Figure 105 COPI Block Diagram



### 5.11.4.2 Overview

The COPI (Combined OTN and Packet Interface) transports packets to/from the CPB and the ODUK switch over an Interlaken interface. One side of the COPI block connects to one port of the CPB and one port of the ODUK switch, while the other side connects to an Interlaken interface. The COPI transports packets from two DPIS (from the CPB and ODUK switch) over a single Interlaken interface.

The Interlaken protocol is implemented through an ILKN120 component instantiated within an Interlaken block. On the transmit side, packets received from the CPB and the ODUK switch are sent to the Interlaken block. The block applies the Interlaken protocol and stripes the data across a software configured number of SERDES lanes. The Interlaken block provides an interface that allows it to be connected to SERDES blocks outside the COPI. On the receive side, the data received from the SERDES is decoded and checked for protocol compliance. The input data in the Interlaken core is aggregated based on the channel number and sent to the CPB or the ODUK switch. The Interlaken block supports a maximum of 24 SERDES lanes and interfaces to a 3.125/6.25/10.3125/11.5/12.5 Gbps rate SERDES. The Interlaken protocol is applied on the data sent to the ILKN120 by the COPI logic and sent out on the SERDES interface. Data is extracted from the SERDES stream and passed out of the ILKN120 to be processed by the COPI logic. The interface to the ILKN120 is 2x256 bits wide. In the transmit direction, the data received

from the DPI is re-formatted as required by the ILKN120. The data sent by the ILKN120 is re-formatted as required on the DPI.

The interface to the CPB and the ODUk switch is through a packet interface (DPI).

In the transmit path COPI performs channel remapping, performed using Arbitrary Channel Lookup Table to transfer the packet/burst with the required reconfigured channel number. In the received path Arbitrary Channel Mapping Table is used to reconfigure the channel number of the packet/burst transferred.

The COPI can be operated in packet mode or segment mode. These modes are configurable for each DPI port. In packet mode, the CPB and ODUk switch sends a complete packet (in terms of a small burst) on a channel before switching to another channel. In segment mode, the CPB and ODUk switch sends a burst on a channel before switching to another channel. The COPI always prioritizes the packets received from the ODUk switch over packets from the CPB when transmitting out through the Interlaken block. On the receive side, the COPI demux data is based on the channel ID and is sent to the CPB or the ODUk.

**Note:** When the CPB traffic is in segment mode and the ODUk traffic is in packet mode, the packet mode channel change error interrupt (RX\_PKT\_CH\_CHANGE\_ERR\_INT) interrupt for the Interlaken interface will incorrectly fire and should be ignored. Additionally, when the Interlaken burst short is set to 32B, spurious activity will be generated on the RX\_PKT\_CH\_CHANGE\_ERR\_INT interrupt. This interrupt should be ignored under these conditions.

To support in-band communication to the system side packet fabric, at the start of packet, the COPI can replace the first 4 Bytes of the packet with packet type information provided by the CPB. In the receive direction, the COPI extracts the same from packet and forwards to the DPI type field. For the ODUk Switch transmit path, COPI optionally inserts 4Bytes+2Bytes user defined per channel values at the packet header. In the receive ODUk Switch path, COPI extracts the same and compares with the per channel value received from the lookup table.

The COPI implements in-band and out-of-band flow control mechanisms on the Interlaken interface independent of Tx/Rx Paths. Both channel and link level flow control mechanisms exist on the CPB and ODUk switch interfaces. These allow flow control on a channel-by-channel basis or allows the entire interface to be flow controlled. Flow control information sent by the CPB and ODUk switch is processed and formatted as required by the ILKN120. The flow control information sent by the ILKN120 is forwarded to the CPB and ODUk switch. FIFO conditions within the COPI also trigger flow control information to the ILKN120, CPB and ODUk switch. The COPI has an APB interface for configuration and control.

#### 5.11.4.2.1 Feature Overview

- Includes a common packet interface for transporting packets to/from the CPB and ODUk switch over an Interlaken interface
- Includes common Interlaken interfaces to support up to 24 configurable lanes
- Supports two DPI ports: one connected to CPB and the other one connected to the ODUk switch
- Supports channel remapping for the packet/segment transferred.
- Supports 6B(2B+4B) header insertion in ODUk transmit path of Interlaken interface.
- Supports 6B(2B+4B) header extraction in ODUk receive path of Interlaken interface.
- Optionally, enables replacement of first 4 packet bytes with the TYPE field provided by the CPB and 3 bytes of zeros
- The CPB and ODUk switch interface can be independently configured to operate in segment mode or packet mode (software configurable)

- Prioritizes packets from ODUk switch over packets received from CPB for transport over the Interlaken interface
- Provides buffering to handle interface level rate mismatches between the Interlaken transfer rates and the transfer rate across the CPB/ODUk switch interfaces, and to handle the Link XOFF back pressure received from Interlaken Flow Control Interface.
- Supports generation of flow control on an Interlaken interface:
  - Channel-level flow controls derived from 'FIFO ZONE' from the CPB and ODUk switch are forwarded to the Interlaken interface
  - Link-level flow control is generated based on the FIFO levels in the COPI
  - Supports inband and out-of-band flow control using a 300-entry programmable calendar. Inband and out-of-band flow control cannot be used concurrently
  - Supports 100MHz OOB interface
- Provides generation of flow control to the CPB and ODUk switch:
  - Channel-level flow control is generated based on the channel level flow control information received on the Interlaken interface
  - Link-level flow control is generated based on link level flow control received on Interlaken and FIFO levels in the COPI

#### 5.11.4.2.2 Low-power Mode

The COPI block can be put in low-power mode. In low-power mode, the block is not functional except for the PCBI. The block can be kept in reset when in low-power mode. The clocks in the blocks are disabled (except for the PCBI) and the RAMs are put in low-power mode. The COPI block also allows the receive buffers for the ODUk switch or the CPB to be put in low-power mode.

#### 5.11.4.3 APB Interconnect

The APB2APB bridge connects to the APB interface entering the COPI and provides bridge used to:

- Connect to the PCBI for local register access
- Connect to the Interlaken core access

#### 5.11.4.4 PCBI

The PCBI has an APB interface and implements the following registers that are accessible to software:

- A register to generate reset to the core logic of ILKN120, transmit section of the ILKN120 and the receive section of the ILKN120
- A register to specify values transmitted on the multi-use field of the Interlaken control word. Note that the use of the MU field for flow control or address extension is *not* supported.
- A control register to force flow control to all Interlaken channels to XON.
- A status register to capture the values received on the multi-use field of the Interlaken control word.
- Registers to specify the FIFO thresholds for various FIFOs.
- Register to specify zone FLOP initialization values.

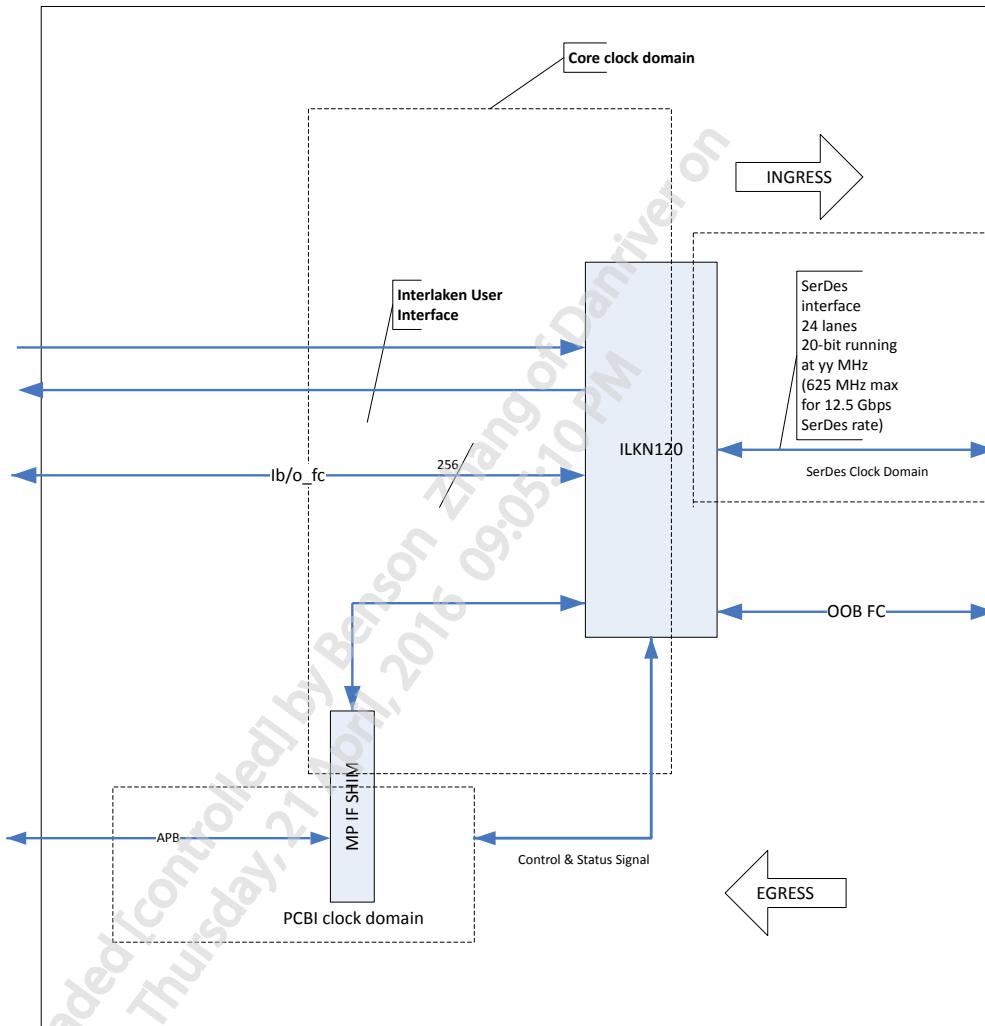
The PCBI also combines the various sources of interrupts provided by the Interlaken block into a single interrupt output.

### 5.11.4.5 Interlaken Block

The Interlaken block implements the ILKN120 and PCBI SHIM.

#### 5.11.4.5.1 Interlaken Block Diagram

Figure 106 Interlaken Block Diagram



#### 5.11.4.5.2 Overview

Features of the ILKN120 are provided in [Section 5.11.4.5.3](#).

In the fabric ingress (transmit) direction, the ILKN120 is used to wrap the data as per Interlaken protocol. It sends and accepts both in-band and out-of-band flow control information and performs clock domain crossing across the core clock and each of the SERDES clocks. The data sent to the ILKN120 is encoded using 67/64 encoding. The encoded data is striped over the configured number of SERDES lanes.

In the fabric egress (receive) direction, the ILKN120 is used to accept data from all the SERDES lanes. The received data is decoded per the Interlaken specification. The Interlaken protocol is enforced on

the incoming data stream. Errors are reported for protocol violation, and appropriate error counters are updated. The decoded data is passed over the user interface.

#### 5.11.4.5.3 Features

- Conforms with Interlaken Protocol Definition Rev 1.2
- Conforms to Interlaken Interoperability Recommendation Spec, Rev 1.6
- Software configurable number of SERDES lanes to a maximum of 24 lanes
- Supports connection to SERDES at 3.125/6.25/10.3125/11.5/12.5 Gbps rates
  - Maximum 24 lanes at 3.125 Gbps
  - Maximum 24 lanes at 6.25 Gbps
  - Maximum 16 lanes at 10.3125 Gbps
  - Maximum 12 lanes at 11.5 Gbps
  - Maximum 12 lanes at 12.5 Gbps
- Ability to disable SERDES lanes
- Supports arbitrary lane reordering for the enabled lanes:
  - The lane sequence will be specified through register for Rx and Tx separately. For each logical Interlaken lane, the physical SERDES lane is selected through programmable register.
- Supports up to 256 channels
- Supports packet mode and segment mode
- Provides a programmable Burst Max parameter (valid values are 128, 192, 256 bytes)
- Provides a programmable Burst Short parameter (valid values are 32 and 64 bytes)
- Provides a programmable Burst Min parameter (valid values are 32 and 64 bytes, meeting the requirements of  $\text{BurstMin} \leq \text{BurstMax}/2$  and  $\text{BurstMin} \geq \text{BurstShort}$ )
- Provides a programmable MetaFrameLength parameter value of 2048, 1024 and 4096 words
- Provides error detection on user interface to detect illegal burst sizes and other errors
- Provides a single channel (programmable) statistics counter
- Provides per-SERDES lane statistics counters
- Supports both in-band and out-of-band flow control
- Provides a programmable 300-entry calendar for flow control
- Provides transmit to receive loopback paths on both data and flow control
- Provides receive to transmit loopback paths on both data and flow control
- Provides configurable error injection on transmit for improved testability
- Supports PRBS and test pattern generation

#### 5.11.4.5.4 Interlaken Channel Mapping

Table 57 shows recommended channel numbering in the Interlaken burst control word. Table 58 illustrates recommended numbering for channels within a range. Note that channels 254 and 255 are reserved.

Table 57 Interlaken Channel Mapping at the COPI Interface Level

Channel ID	Source
149:0 (Programmable range)	CPB
253:150 (Programmable range)	ODUk switch
254	Reserved (used in flow control calendar to signal link level flow control)

Channel ID	Source
255	Reserved(Null channel)

Table 58 Channel Information Details

Channel ID	Source
47:0	Ethernet ports (up to 12 channels used)
143:48	GFP channels
144	On-chip microprocessor
145	Diagnostic port (data client)
149:146	Reserved - CPB
245:150	96x line side Rx ODUk/96x mapper ODUk channels
246	Diagnostic port (ODUk switch client)
253:247	Reserved (ODUk switch)
254	Reserved (used in flow control calendar to signal link level flow control)
255	Reserved (null channel)

In the fabric egress direction (external device to PMC device), a 256 entry lookup table allows arbitrary assignment of an Interlaken packet to any channel in the ODUk switch or CPB switch. The table is indexed by the BCW channel number, with each entry consisting of an 8-bit destination channel and a 1-bit destination buffer (either ODUk switch or CPB).

In the fabric ingress direction (PMC device to external device), a 104 word by 8-bit lookup table for ODUk sourced packets allows arbitrary assignment of BCW channel numbers for traffic originating from an ODUk queue. Similarly, a 150 word x 8 bit lookup table for CPB sourced packets allows arbitrary assignment of BCW channel numbers for packets originating from a CPB queue.

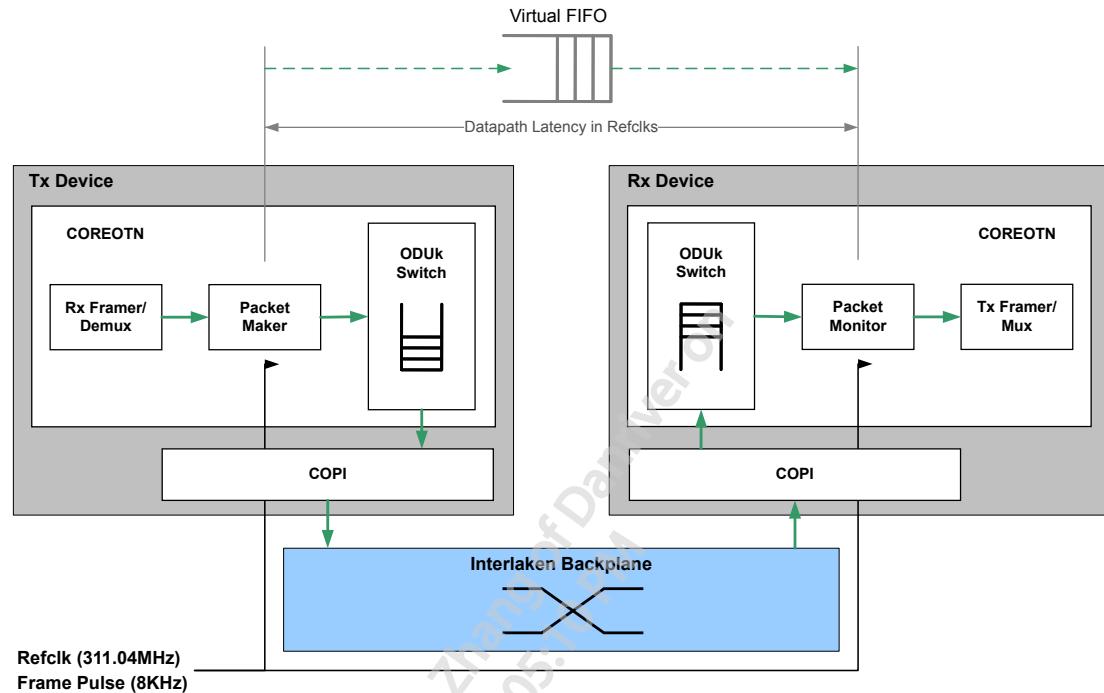
#### 5.11.4.5.5 Latency Management Across an Interlaken Based Fabric

When data is sent across an Interlaken based fabric, there is packet delay variation (PDV) that must be accounted for in the sizing of datapath buffers and in the centering of these buffers. We must ensure that sufficient data is stored in the ODUk switch of the receive device to accommodate this variation without buffers under-running or overflowing. The datapath region between the Packet Maker in the transmit device and the Packet Monitor in the receive device can be thought of as a virtual buffer. For ODUk data sent from the transmit device COREOTN, the datapath latency in this virtual buffer is monitored and adjusted at the receive device to ensure that data integrity is maintained.

In order to monitor datapath latency, a synchronized 8 kHz frame pulse must be delivered to both the transmit device and the receive device. This frame pulse synchronizes 311.04 MHz counters used to determine the datapath latency through the use of timestamps inserted into the datastream. These frame pulses should be phase aligned within the window as defined in the [Section 5.16.1.6](#) description.

For data sent from the device COREOTN, prior to the ODUk switch, the stream data is packetized and a header including timestamp is applied. This packetization and timestamping occurs in the Packet Maker as shown in [Figure 107](#). In the receive device on the far end of the fabric, following the switch, this timestamp is compared against the local timestamp counter and the datapath latency is determined. By dividing this latency with the known stream data rate, the amount of data causing this latency is determined. This is the amount of data between the point at which the timestamp was applied in the transmit device, and this point in the receive device where the timestamp is compared. This timestamp comparison occurs in the Packet Monitor as shown in [Figure 107](#).

Figure 107 Definition of the Virtual FIFO: Transmit Device -&gt; Interlaken -&gt; Receive Device

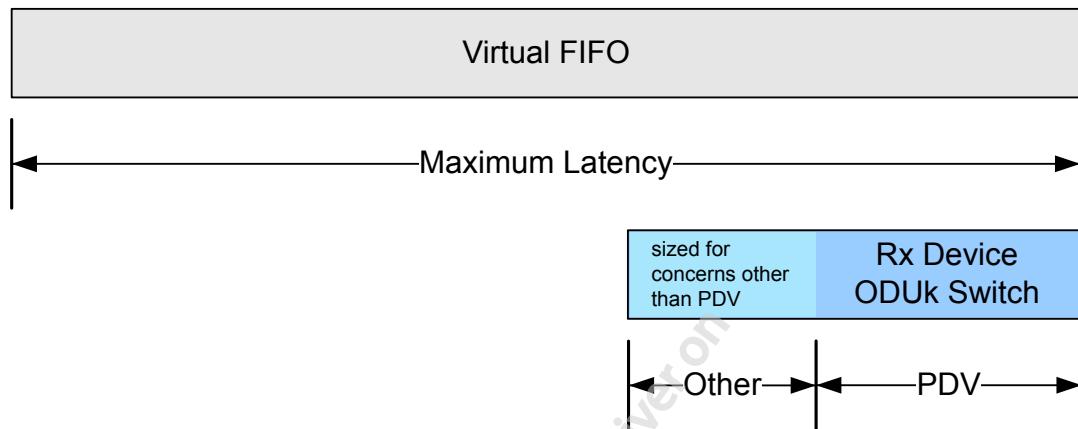


On datapath startup in the receive device, the latency is measured and the datapath held back until aged to the desired latency. At this point the datapath is released and the amount of data between the transmit and receive devices is as desired based on the latency.

There are two latency parameters that need to be understood in order to properly control fabric datapath: the maximum datapath latency and the PDV. The maximum datapath latency is the maximum time as measured by packet timestamps, that a packet can take to transition from the transmit device to the receive device. As mentioned above and shown in Figure 107, the datapath between the transmit device and the receive device can be thought of as a virtual buffer. This virtual buffer has a data capacity equal to the aggregate buffering along the path, and must be sized to allow for the expected PDV across the fabric. The maximum latency to which the datapath is held back should be set at the limit of the virtual buffer.

The receive device ODUK switch depth should be set to source the expected PDV worth of data, as if it were the only buffer in the datapath. The relationship between the virtual buffer, maximum latency, PDV, and the ODUK switch depth is shown in Figure 108. PDV is only one of the considerations used to size the receive device ODUK switch; other considerations include datapath XOFF and ON latency. Refer to Section 5.21 of this document for additional ODUK switch FIFO sizing requirements.

Figure 108 Sizing Relationship Between the Virtual FIFO and the Receive Device ODUk Switch Buffer



When configuring the ODUk switch depth and datapath holdoff parameters, maximum datapath latency is broken down into various subcomponents. The maximum latency between the transmit device MPMA and the receive device ODUk switch ingress is referred to as latency\_s. When accounting for PDV in the receive device ODUk switch depth, the depth requirement due to PDV is accounted for in what we refer to as the halt buffer. A portion of the additional depth allocation required in the receive device ODUk switch is due to datapath transients in the receive, and is referred to as the hysteresis buffer depth. Two of the components which are used when sizing the receive device ODUk switch depth are the PDV dictated halt buffer and the receive device datapath dictated hysteresis buffer. Additionally, the latency\_s, halt buffer and hysteresis buffer form components of the maximum datapath latency. Please refer to the Programmers Manual [4] for detailed guidance on the API specification and its configuration.

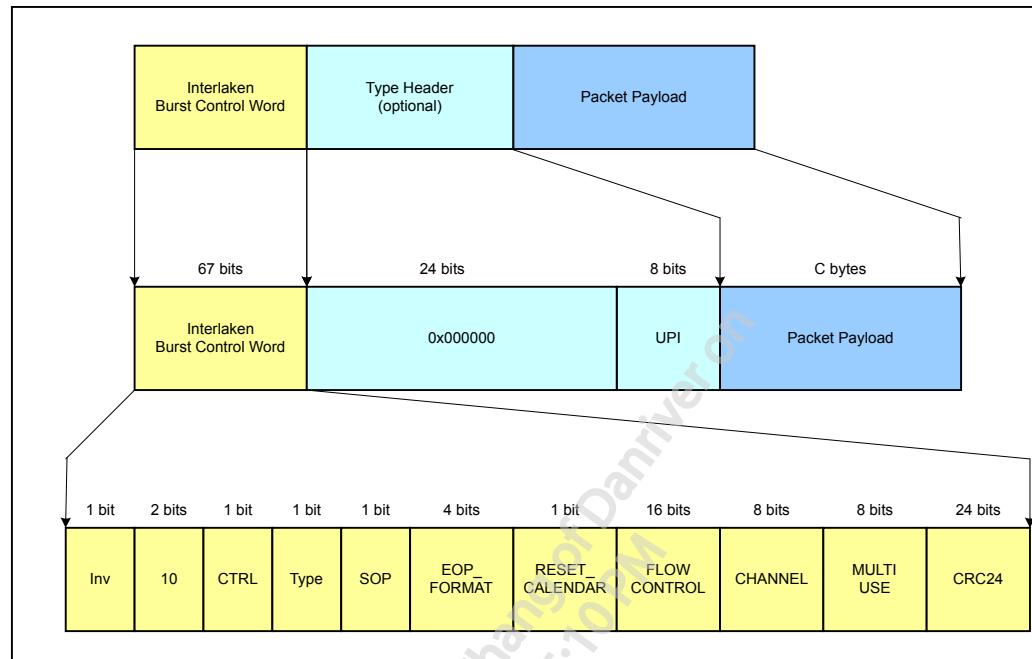
#### 5.11.4.5.6 PIF (CPB) Packet Format over the Interlaken Interface

Figure 109 shows the transfer format of packets from the CPB interface over the Interlaken interface.

To support GFP mapping of Interlaken packets, the SIFD can be configured to enable the insertion and extraction of a “type” header in packets to/from the Interlaken interface. This 32-bit type header contains a byte representing the UPI value carried in the GFP payload header, when packets are destined for GFP mapping. This mode of operation enables the remote packet source to specify the UPI field for packets entering the GFP mapper. Conversely, it allows the remote packet sink to determine the GFP UPI associated with each demapped packet. The header insertion can be enabled and disabled per channel.

The TYPE insertion is performed when dpi\_sop\_hole is set for a packet and when insertion enable register bit is set. Here the UPI field will be placed at the byte location 4 of SOP packet. In the egress path when enabled, the UPI field will be extracted and forwarded to DPI[15:8] field.

Figure 109 PIF Format Over Interlaken



[Table 59](#) and [Table 60](#) enumerate the possible sources and destinations for packet traffic to/from the Interlaken (PIF) interface. Multiple packet payload formats are available for each datapath as shown in the "Payload" column heading. For each datapath/payload, the PMC device can support optional header configurations enabling the transport of GFP UPI and linear extension headers with a Network Processor. Not all combinations are supported by the device as denoted by the "Support" column in the tables below.

Table 59 Ingress Packet Traffic (Device to Network Processor)

Source	Payload	COPI Hdr	GFP Type Hdr	GFP Ext Hdr	Support	Interlaken Packet Format
EMAC	ENET (Pkt)	No	N/A	N/A	Yes	[ILKN BCW] [ENET Pkt]
EPMM	ENET (PP-OS)	Yes	N/A	N/A	No	N/A
MAPOTN->	ENET (Pkt)	No	No	No	Yes	[ILKN BCW] [ENET Pkt]
EMAC						
MAPOTN->	ENET (Pkt)	No	Yes	No	No	N/A
EMAC						
MAPOTN->	ENET (Pkt)	No	Yes	Yes	No	N/A
EMAC						
RGFPF	ENET (Pkt)	No	No	No	Yes	[ILKN BCW] [ENET Pkt]
RGFPF	ENET (Pkt)	Yes	No	No	Yes	[ILKN BCW] [COPI Hdr] [ENET Pkt]
RGFPF	ENET (Pkt)	No	Yes	No	Yes	[ILKN BCW] [GFP Type Hdr] [ENET Pkt]
RGFPF	ENET (Pkt)	No	Yes	Yes	Yes	[ILKN BCW] [GFP Type Hdr] [GFP Ext Hdr] [ENET Pkt]

Source	Payload	COPI Hdr	GFP Type Hdr	GFP Ext Hdr	Support	Interlaken Packet Format
RGFPF	ENET (PP-OS)	Yes	No	No	Partial - No egress PMON	[ILKN BCW] [COPI Hdr] [ENET PP-OS]
RGFPF	Non-ENET	No	No	No	Yes	[ILKN BCW] [Pkt]
RGFPF	Non-ENET	Yes	No	No	Yes	[ILKN BCW] [COPI Hdr] [Pkt]
RGFPF	Non-ENET	No	Yes	No	Yes	[ILKN BCW] [GFP Type Hdr] [Pkt]
RGFPF	Non-ENET	No	Yes	Yes	Yes	[ILKN BCW] [GFP Type Hdr] [GFP Ext Hdr] [Pkt]

Table 60 Egress Packet Traffic (Network Processor to Device)

Source	Payload	COPI Hdr	GFP Type Hdr	GFP Ext Hdr	Support	Interlaken Packet Format
EMAC	ENET (Pkt)	No	N/A	N/A	Yes	[ILKN BCW] [ENET Pkt]
EPMM	ENET (PP-OS)	Yes	N/A	N/A	No	N/A
MAPOTN-EMAC	ENET (Pkt)	No	No	No	Yes	[ILKN BCW] [ENET Pkt]
TGFPF	ENET (Pkt)	No	No	No	Yes	[ILKN BCW] [ENET Pkt]
TGFPF	ENET (Pkt)	No	No	Yes	Yes	[ILKN BCW] [GFP Ext Hdr] [ENET Pkt]
TGFPF	ENET (Pkt)	Yes	No	No	Partial - No egress PMON	[ILKN BCW] [COPI Hdr] [ENET Pkt]
TGFPF	ENET (PP-OS)	Yes	No	No	Yes	[ILKN BCW] [COPI Hdr] [ENET PP-OS]
TGFPF	Non-ENET	No	No	No	Yes	[ILKN BCW] [Pkt]
TGFPF	Non-ENET	No	No	Yes	Yes	[ILKN BCW] [GFP Ext Hdr] [Pkt]
TGFPF	Non-ENET	Yes	No	No	Yes	[ILKN BCW] [COPI Hdr] [Pkt]

See below for the [ILKN BCW], [COPI Hdr], [GFP Type Hdr], [GFP Ext Hdr], [ENET Pkt], and [ENET PP-OS] header formats.

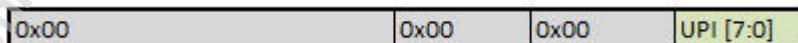
### [ILKN BCW]

Interlaken Burst Control Word

67-bit word per Interlaken Protocol specification

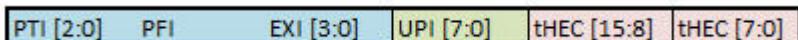
### [COPI Hdr]

COPI UPI Type Header



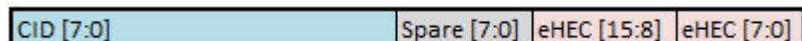
### [GFP Type Hdr]

GFP Payload Header



### [GFP Ext Hdr]

Linear Extension Header



**[ENET Pkt]**

Ethernet packet framed PDU per ITU-T G.7041 clause 7.1

**[ENET PP-OS]**

(Preamble + Packet) per ITU-T G.7041 clause 7.9.2.1

(Ordered Set) per ITU-T G.7041 clause 7.9.2.2

#### 5.11.4.5.7 CBRI (ODUkSW) Packet Format over the Interlaken Interface

Figure 110 shows the transfer format used to convey ODUk streams over the Constant Bit Rate Interface (CBRI). Two optional user configurable fields are provided:

- The first field (user specific overhead "A") is 4 bytes in length. When enabled, it is compared to a 4B lookup result indexed by the channel ID resolved from use of the 2B header if mode 0 is enabled or the Interlaken burst control word channel number if the 2B header mode 0 is not enabled. An interrupt is generated if a comparison mismatch occurs but the packet is not dropped and the 4B part B header value is not captured, similar to mode 1 of the 2B part B header.
- The second field (user-specified overhead "B") is 2 bytes in length. When enabled, it operates in mode 0 or mode 1.

In mode 0, the 2B value is used as an index to a CAM of 104 user-defined 16-bit values for the purpose of defining the destination channel in the ODUk switch. A destination ODUk switch channel ID is associated with each CAM entry. A match of the 2B header value to the CAM entries will result in the packet being forwarded to the ODUk switch as per the CAM lookup result. A CAM lookup failure will result in capture of the 2B header value that failed the lookup, the generation of an interrupt, and drop of the ODUk packet.

**Note:** The design intent is for the CAM to be statically provisioned prior to receiving traffic. Dynamic provisioning is supported provided a single bit at the reserved bit offset within the 16-bit header is static and always set to 0b0 or 0b1. Two options are supported for configuring the destination channel lookup table in 2B header mode 0:

- Statically provision the CAM prior to receiving traffic.
- Dynamically provision the CAM but ensure 1-bit of the 16-bit header is always static. Dynamic provisioning will reduce the addressing range of the 16-bit header due to the need to reserve one bit for a static one or zero value. Any of the 16-bits in the 2-byte header can be used for this purpose.

In mode 1, the 2B header value is compared to the 2B lookup result indexed by the burst control word channel number. If there is a compare mismatch an interrupt is generated, the packet is not dropped, and the lookup index is not captured.

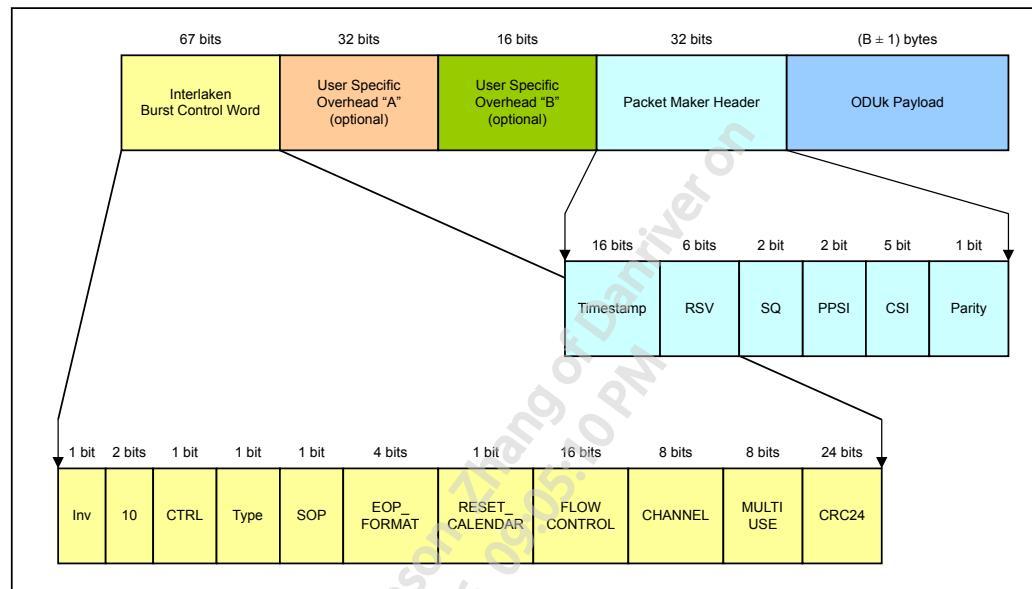
User-specific headers are inserted per user payload packet and not per the Interlaken burst word. Therefore, if the CBRI packet size is less than the configured max\_burst size of the Interlaken, the custom headers will appear at the start of every Interlaken burst word. CBRI packet sizes exceeding the max\_burst size will result in the optional header appearing with Interlaken burst words corresponding to the start of packet (SOP=1). Lookup of the correct destination queue for the second burst of packet will fail since the optional headers are used to address the correct destination queue and the optional header is not present in the second burst. Therefore, use of the optional headers is not supported for nominal sized ODUk packets greater than max\_burst bytes in length.

Note that nominal sized packets greater than 256 bytes are used when interfacing to 512 byte fabrics. Note also that use of the optional 6B header is only supported for ODUk traffic. The optional header is not supported for Interlaken packets to or from the CPB.

Use of the Interlaken interface is dependent on the nominal ODUk payload size and the bandwidth expansion required for the user specific and the packet maker header. Consider the aggregate bandwidth

of the ODUk signals, the nominal ODUk packet sizes used, the number and rate of the Interlaken lanes, and the headers that are added when determining the throughput that can be achieved for the Interlaken interfaces. Small packet sizes and insertion of all the optional headers will limit the throughput that can be achieved over Interlaken. Limited Interlaken throughput, may in turn, limit the number of line and client interfaces that can be supported for a given application.

Figure 110 CBRI Packet Format Over Interlaken



#### 5.11.4.5.8 Interlaken Throughput Considerations for OTN over Packet Fabrics (CBRI)

The device implements an industry-standard Interlaken interface for transferring packet data or packetized ODUk signals to external devices. The Interlaken interface supports a configurable BurstMax value of 128, 192, or 256 and a configurable BurstShort value of 32 or 64. The throughput of the Interface depends on the packet sizes that are transferred, the number of lanes used, the rate of the lanes and the burst max/short configuration.

The device generates nominally B sized packets that vary by +/- 1 byte. A mandatory 4-byte packet maker monitor header is added to the "B" sized ODUk packet. Bandwidth expansion will occur when the optional A and B headers are added (6-bytes). Utilization of the interface suffers when using small packets that are required to support 128-byte fabrics since the headers and interlaken overhead represent a larger percentage of the total number of bytes that are transferred. Supporting a 512-byte fabric requires using B packet sizes that require two Interlaken bursts to transmit, resulting in additional overhead for burst control words. Note that packet sizes are typically selected to be near 512-bytes. The second burst will be greater than BurstShort in length and the reduction in Interlaken utilization for sending two bursts will be minimal since additional idle words are not needed to pad the length of the second burst to BurstShort. Optional ODUk packet headers are not supported for 512B fabrics.

Each application should run the *CBRI Utilization Calculator*<sup>[5]</sup> to ensure sufficient bandwidth is allocated. For example, 12xODU2f can exceed the bandwidth of a 150G Interlaken interface at some OIF packet sizes.

### 5.11.4.5.9 Arbitrary Channel Mapping

The Interlaken interface supports arbitrary mapping of channel numbers in the Interlaken BCW or optional headers to internal CPB / ODUk switch channel numbers. Dynamic update of mapping is supported with the following restrictions for the header modes.

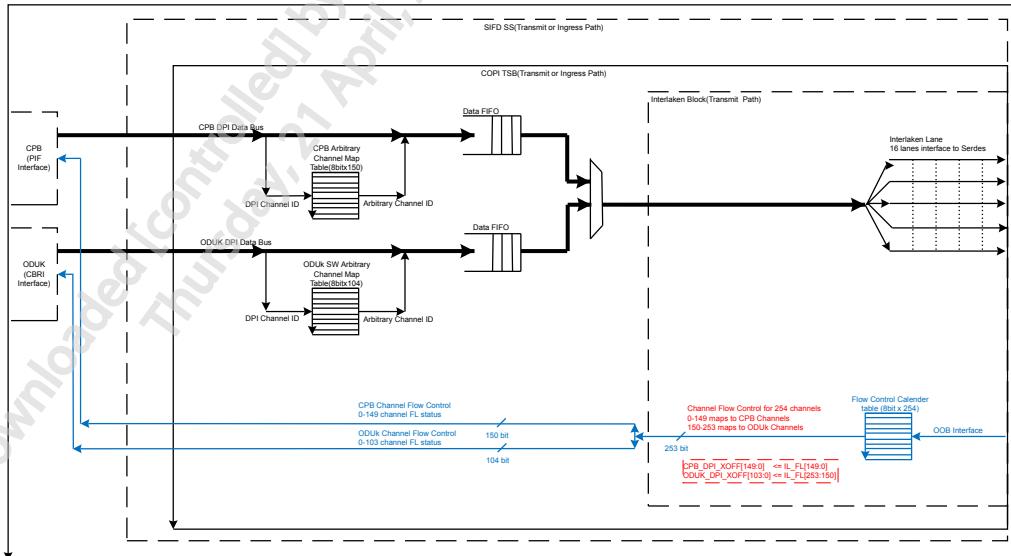
1. When optional headers are not used, arbitrary mapping from the BCW channel to the ODUk Switch or CPB channel is supported provided there is no data being received on the channels being changed.
2. 2B mode 1. Not supported for updates of an active channel as per (1).
3. 2B mode 1 + 4B. Not supported for updates of an active channel as per (1).
4. 2B mode 0 without 4B header:
  - If a full 16-bit address width is required, dynamic update to the address mapping is not supported. The 2B CAM must be provisioned prior to receiving any traffic on any channel.
  - Dynamic provisioning can be supported if a single bit, at reserved bit offset within the 16-bit header, is static and always set to 0b0 or 0b1. Any of the 16-bits in the 2-byte header can be used for this purpose.
5. 2B mode 0 + 4B. Same restriction as (4). Only static reprovisioning of the 4B LUT prior to receiving any traffic is supported. Dynamic reprovisioning is not supported under any condition.

Note that the device does not provide any means of disabling an Interlaken channel to allow the dynamic update in items 1 through 3. The upstream device must not send data to the channel that is to be updated. Failure to prevent this could result in the disruption of unrelated channels.

The features associated with the Packet Maker Header fields are described in [Table 25](#).

### 5.11.4.5.10 Flow Control Arbitrary Channel Map

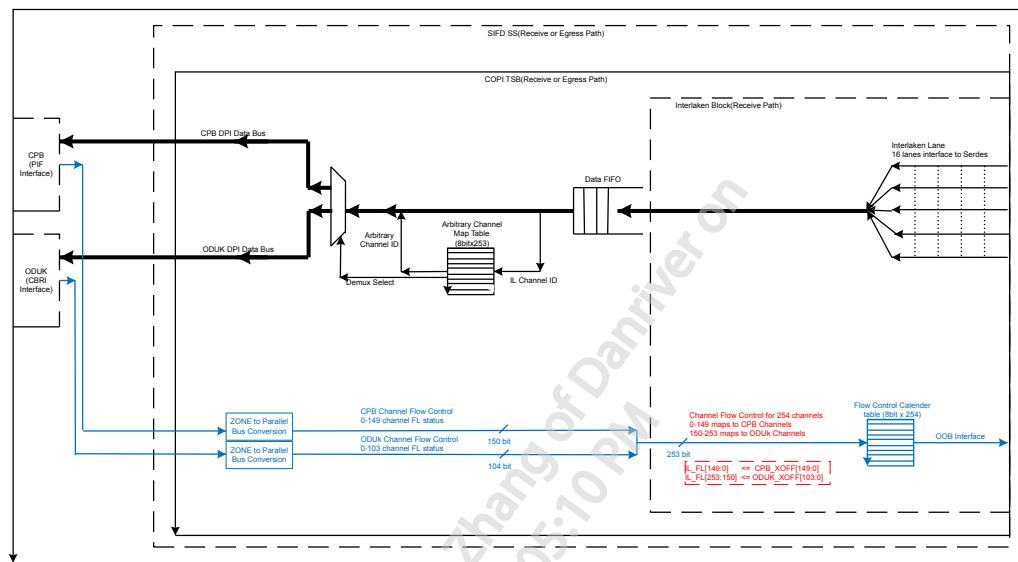
[Figure 111 Transmit Flow Control Arbitrary Channel Mapping](#)



For the requirement of Arbitrary Channel mapping, the CPB Arbitrary Channel Map table and ODUK SW Arbitrary Channel Map table need to be programmed in order to map the channel for the data path. For the Flow Control, the Calender table need to be programmed for Arbitrary Channel Mapping.

The Flow Control Calender is common for CPB and ODUk Path. Output IL\_FL[149:0] maps to CPB\_DPI\_XOFF[149:0] and IL\_FL[253:150] maps to ODUK\_DPI\_XOFF[103:0]. The flow control calender needs to be programmed accordingly.

Figure 112 Receive Flow Control Arbitrary Channel Mapping



For the requirement of Arbitrary Channel mapping, the Arbitrary Channel Map table needs to be programmed in order to map the channel for the data path. For Flow Control, the Calender table needs to be programmed for Arbitrary Channel Mapping. The Flow Control Calender is common for CPB and ODUk Path. CPB\_XOFF[149:0] maps to IL\_FL[149:0] and ODUK\_XOFF[103:0] maps to IL\_FL[253:150]. The flow control calender needs to be programmed accordingly.

#### 5.11.4.5.11 2B/4B/6B Header Insertion/Extraction

The header insertion/extraction block provides an option to replace the first 2 bytes or 4 bytes or 6 bytes of packets transmitted from ODUK SW interface. In the receive direction, the first 2/4/6 bytes of packets received can be compared to user configured header values.

In both receive and transmit direction a 104 deep lookup table is maintained, one each for 2 bytes and 4 bytes. The per channel (DPI channel) user defined values can be configured in these lookup tables. [Table 61](#) describes the modes of operations supported.

Table 61 Header Insertion Operating Modes

Mode of operation	Operation
No header insertion enabled	Transmitted packets are un-modified
2 Byte header insertion enabled	First 2 bytes of packets are replaced with 2 Bytes per channel user defined value from 104 deep lookup table
4 Byte header insertion enabled	First 4 bytes of packets are replaced with 4 Bytes per channel user defined value from 104 deep lookup table
6 Byte header insertion enabled	First 6 bytes of packets are replaced with 2 Bytes and 4 bytes per channel user defined values from 104 deep lookup tables

Not all packets transmitted are modified when 2/4/6 bytes insertion is enabled; only the packets with SOP\_HOLE indication gets modified. ODUK switch will keep 2/4/6 bytes hole at the start of the packet and assert SOP\_HOLE indication for all packets which needs to be modified, in real operating scenario. [Figure 113](#) and [Figure 114](#) provide further details.

**Figure 113** Arbitrary Channel Map and 6B Insertion Block

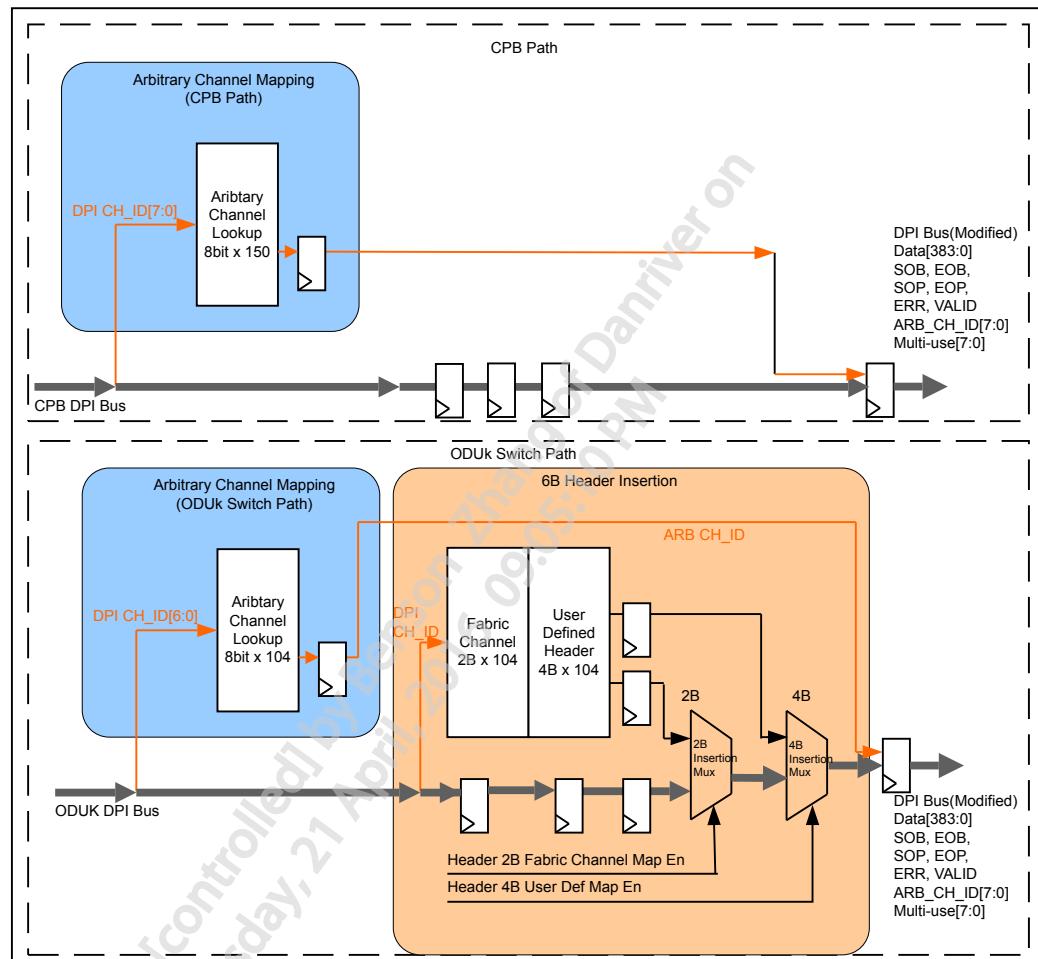
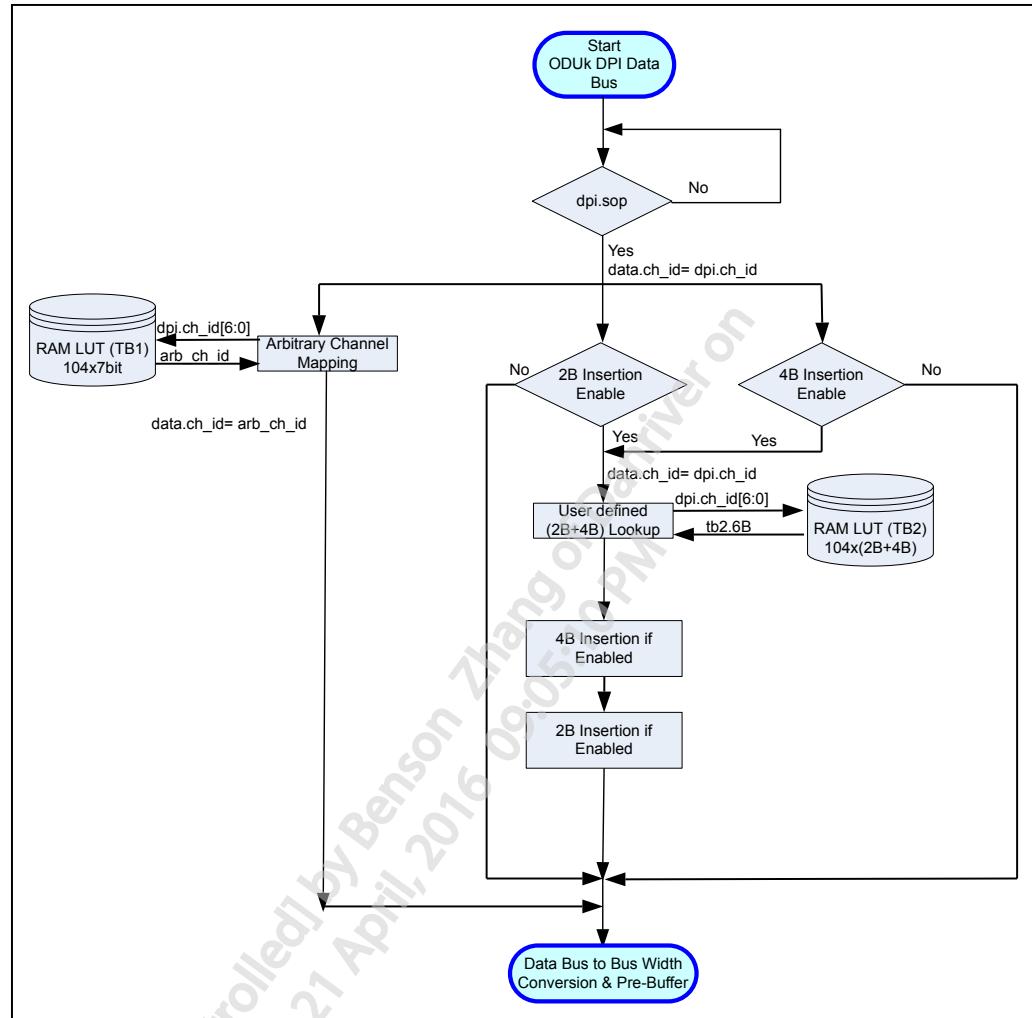


Figure 114 Tx Channel Map &amp; 6B Insertion Block Flow Chart



In the receive direction, the first 2/4/6 bytes of packets destined to ODUK SW interface can be compared with a per channel user configured values programmed in 104 deep 2byte/4byte lookup tables. The usage of the lookup table is similar to the transmit direction. If the lookup fails, an error interrupt is generated.

The 2 byte lookup table, instead of being used as a header comparison lookup table, can also be used to derive the DPI channel ID, when enabled. This special mode (called fabric channel lookup mode) should only be enabled when interlaken is operating in packet mode (not in segment mode). In fabric channel lookup mode, channel ID derived from 2B header lookup is used for all further processing including the 4 byte header lookup. If the 2 byte header lookup fails in fabric channel lookup mode, the packet is marked for dropping. [Figure 115](#) block diagram and [Figure 116](#) flow chart provides more details.

Figure 115 Arbitrary Channel Map and Header Extraction Block

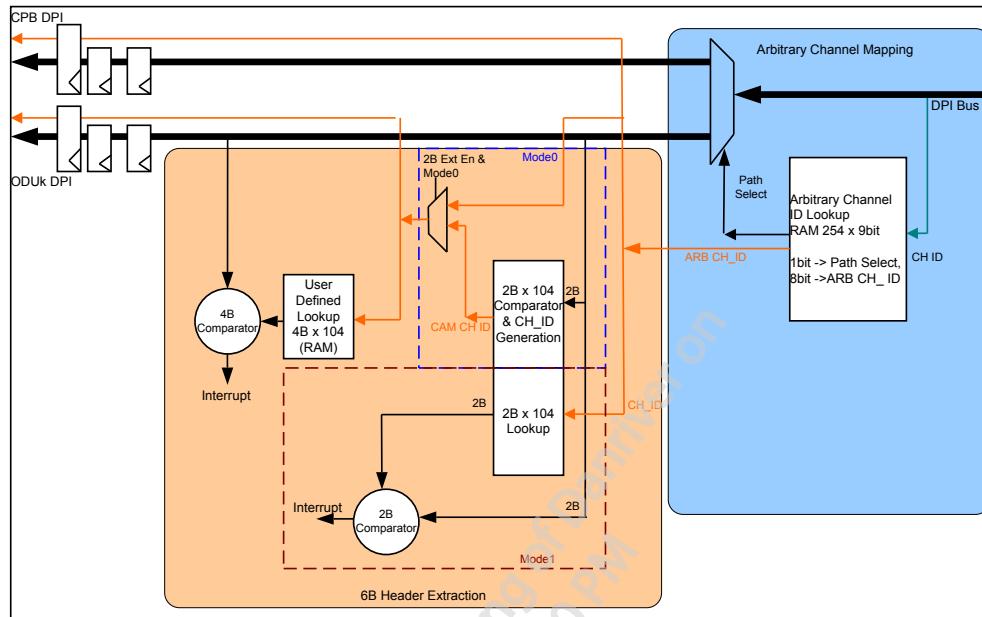
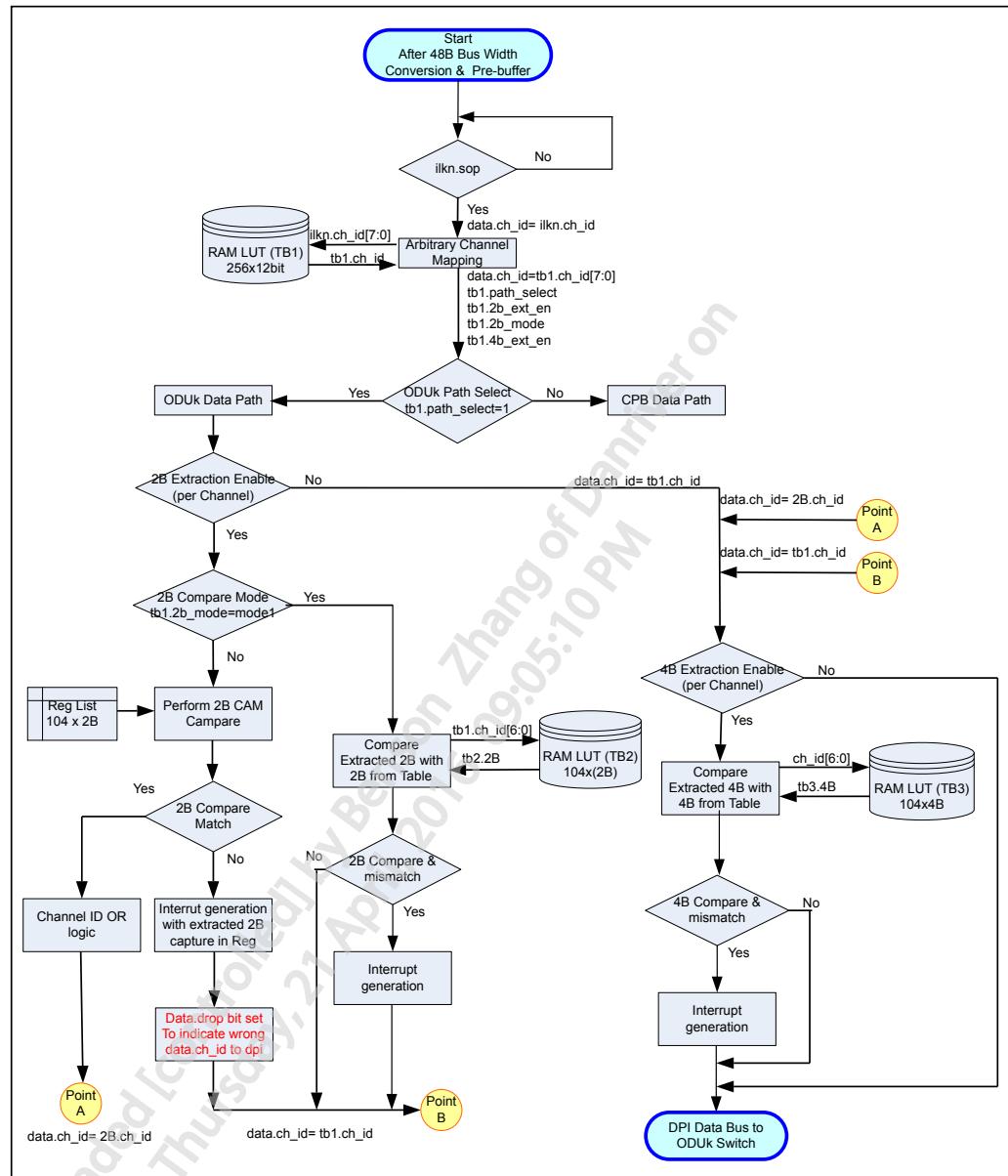


Figure 116 Rx Arbitrary Channel Map Flow Chart



#### 5.11.4.5.12 Spurious Packet When Re-provisioning

De-provisioning an Interlaken channel in the SIFD block in the fabric egress direction (external device to device direction) will result in a residual word being stored in the ODUk Switch. Upon re-provisioning the channel, the residual word will result in the transmission of a spurious packet. The impact of the spurious packet will be as follows:

- For the SIFD to ODUk switch datapath, there will be no impact due to the time required to re-frame to the ODUk signal after re-assembly of packets and recover the new rate of the ODUk.

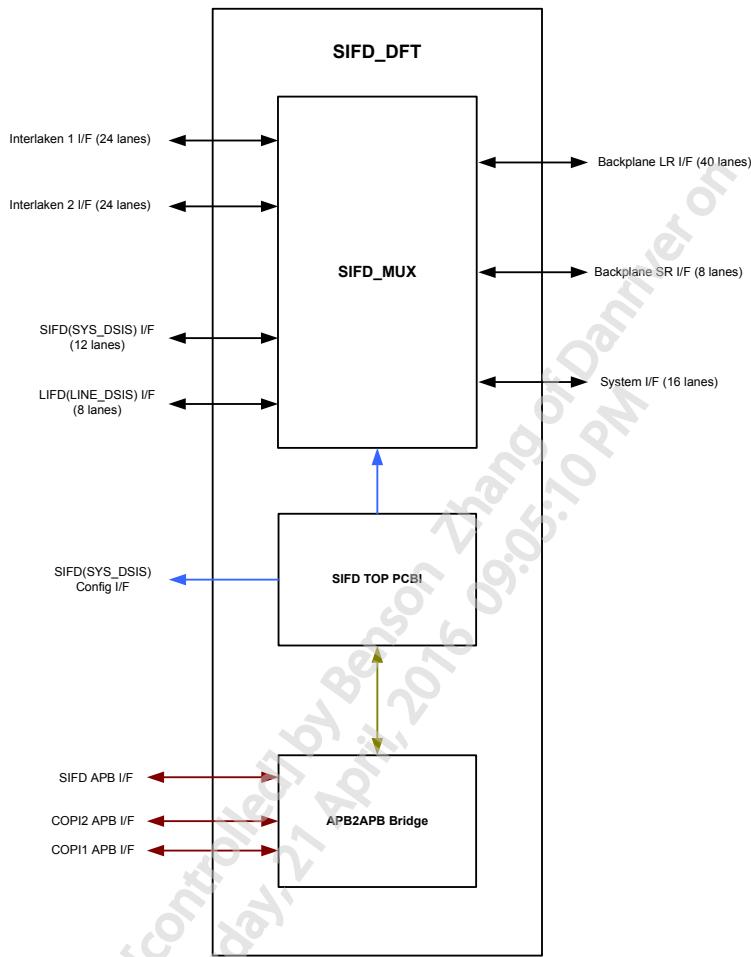
- For the SIFD to DCPB to the GFP mapper datapath, downstream nodes will receive one GFP-F encapsulated packet with a valid GFP layer. The packet that is encapsulated will be corrupt, for example, it may have a badly formed MAC address if the far end is expecting Ethernet packets.
- For the SIFD to DCPB to ENET line datapath, a corrupt packet will be sent to the line and received by the far end device.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:10 PM

## 5.11.5 SIFD\_DFT Block

### 5.11.5.1 Block Diagram

Figure 117 SIFD\_DFT Block Diagram



### 5.11.5.2 Functional Description

#### 5.11.5.2.1 Overview

The SIFD\_DFT incorporates all the top-level logic in SIFD subsystem. It includes an APB interconnect/bridge, top-level PCBI blocks, and the SIFD\_MUX. The SIFD\_MUX allows various client interfaces to share the system-side SERDES.

The SIFD\_DFT also generates and provides hierarchical reset pins to other SIFD subsystem blocks.

#### 5.11.5.2.2 Features

- Allows all the client side interfaces: Interlaken 1, Interlaken 2, and SYS\_DSIS (CBRC, ENET(SYS), SYSOTN) to be connected to the backplane LR, backplane SR and system SERDES

- Provides eight system SERDES lanes to support two 40G CFP line-side (LINEOTN) interfaces
- Provides APB master interfaces to two instances of the COPI
- Implements all the configurations required by the System Digital Interface Switch Interface (SYS\_DSIS)
- Provides all SERDES-side client-interface clocks to the COPI, SYS\_DSIS, and LINEOTN
- Performs data, LOS, and ACB signal multiplexing between client interfaces based on software configurations before connecting to the SERDES lanes
- Provides option (based on software configurations) to disable and invert the SERDES LOS pins before feeding the client interfaces
- Combines a system SERDES LOS and optical module LOS for the SYS\_DSIS

#### 5.11.5.2.3 APB Interconnect

The APB2APB bridge connects to the APB interface entering the SIFD and provides bridge to connect to three master interfaces. One of these interfaces is the APB master interface used to connect to the PCBI block for local register access. The other interfaces are for the APB master interface to connect to the two instances of the COPI.

#### 5.11.5.2.4 PCBI

The PCBI has an APB interface and implements registers that are accessible to software.

The PCBI includes:

- A register to generate resets to the different blocks in the SIFD
- Registers to generate low-power mode inputs to the different blocks in the SIFD
- Mux selects to the SIFD\_MUX block
- SYS\_DSIS configurations
- Clock gating controls to the SIFD\_MUX block
- Deskew inputs to the SIFD\_MUX block

The PCBI also combines the interrupts provided by the different blocks into a single set of interrupts.

#### 5.11.5.2.5 SIFD\_MUX

The SIFD\_MUX block is used to connect the Interlaken interfaces, SYS\_DSIS (CBRC, ENET(SYS) or SYSOTN) and LIFD\_MUX blocks to the backplane LR, system, and backplane SR SERDES blocks. The multiplexer is configured by software to choose one of the possible sources of data for each SERDES lane. See [Section 14.1](#) for the amount of deskew supported by the device. The SIFD\_MUX supports per receive/transmit SERDES lane deskew functionality to match latency of various SERDES lanes used by the same client interface (based on software configuration). The data from each SERDES lane is directed to the correct destination based on the software configuration. The SIFD\_MUX receives ACB clock control signals only from the SYS\_DSIS and the LIFD\_MUX client interfaces. SIFD\_MUX selects the appropriate ACB signals based on software configuration and drives the corresponding system SERDES interface. The SIFD\_MUX block can also squelch clocks based on software configuration.

The backplane LR SERDES are Titan-8 SERDES that comply with the CEI-6G LR link protocol from 2.488 Gbps to 6.4 Gbps. The backplane SR SERDES are Centaur-8 SERDES supporting transfer rates from 3.125 Gbps to 6.25 Gbps. The system SERDES are Saphyre-16 SERDES supporting transfer rates from 2.488 Gbps to 12.5 Gbps.

The SERDES blocks provide a clock in both the transmit and receive direction.

[Figure 118](#) shows the multiplexing implemented in the SIFD\_MUX block for the transmit direction. The data received from the SERDES is similarly sent to the appropriate interface as per the SIFD\_MUX configuration.

SERDES lane use by different interfaces is shown in [Table 62](#).

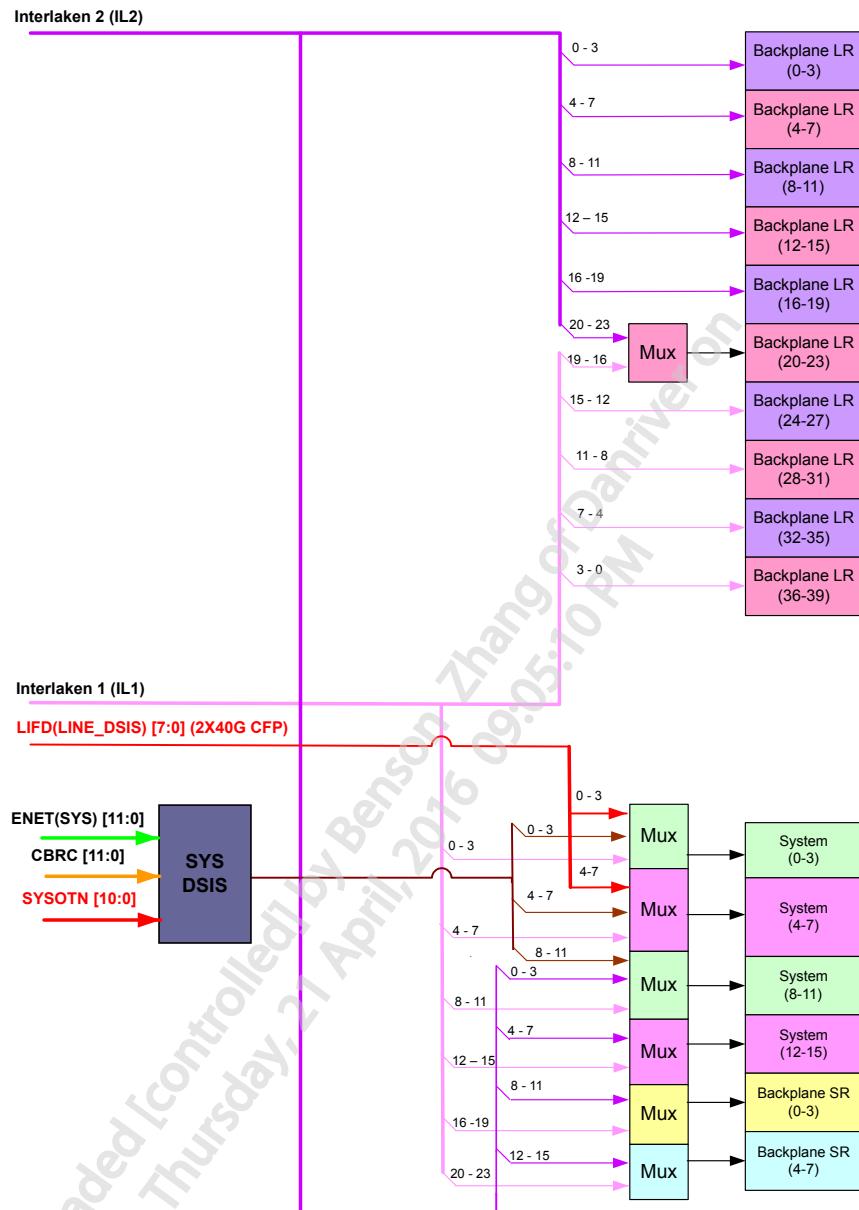
**Table 62 SERDES Usage Per Interface**

Interface	SERDES Lanes Used
Interlaken 1 (Lanes 15:0)	System lanes (15:0) Backplane LR lanes (24:39)
Interlaken 1 (Lanes 19:16)	Backplane SR lanes (3:0) Backplane LR lanes (20:23)
Interlaken 1 (Lanes 23:20)	Backplane SR lanes (7:4)
Interlaken 2 (Lanes 7:0)	System lanes (15:8) Backplane LR lanes (7:0)
Interlaken 2 (Lanes 15:8)	Backplane SR lanes (7:0) Backplane LR lanes (15:8)
Interlaken 2 (Lanes 23:16)	Backplane LR lanes (23:16)
SYS_DSIS Interface (Lanes 11:0) [ENET(SYS), SYSOTN, CBRC]	System lanes (11:0)

**Note:**

1. When Interlaken 2 is greater than 20 lanes, all T-8 SERDES must run at the same rate.
2. The eight Backplane SR(C-8) lanes cannot be split between the two Interlaken interfaces

Figure 118 PM5440 SIFD\_MUX System-side SERDES Muxing



Note that SERDES lanes share CSUs. This puts restrictions on the frequency of operation of one client interface when another client is using a SERDES lane that shares the same CSU. For example, if Interlaken 2 is using 24 backplane LR SERDES lanes and Interlaken 1 is using the remaining 16 backplane LR SERDES lanes, then Interlaken 1 must operate at the same SERDES rate as Interlaken 2 since the backplane LR SERDES lanes [29:20] share the same CSU. Also note that the eight Backplane SR(C-8) lanes cannot be split between the two Interlaken interfaces.

The Table 63 provides information on the sharing of CSUs by SERDES lane groups.

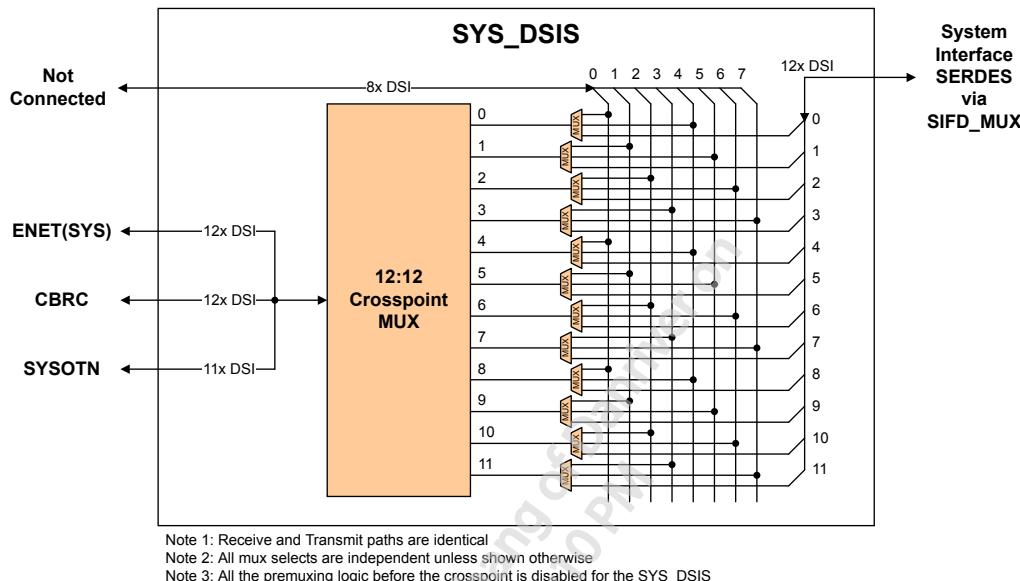
**Table 63** System-side SERDES CSU Information

SERDES Type	Information on CSU Driving Each Lane
Backplane LR SERDES	Backplane LR SERDES lanes [19:0] share one CSU while lanes [39:20] share another CSU.
Backplane SR SERDES	All backplane SR lanes share the same CSU.
System SERDES	Each system SERDES lane has a separate CSU. Each lane can operate at independent rates.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:10 PM

### 5.11.6 System Digital Serial Interface Switch (SYS\_DSIS)

Figure 119 SYS\_DSIS Block Diagram

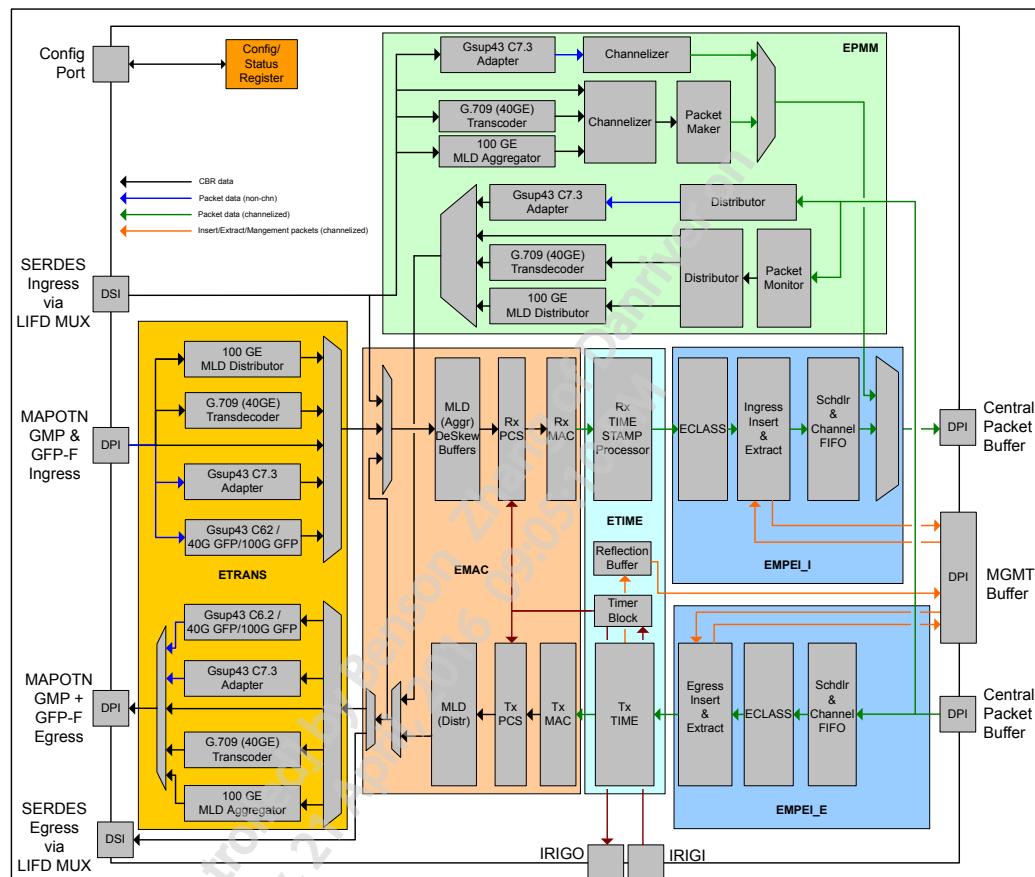


The SYS\_DSIS (System Digital Serial Interface Switch) allows the 12 System Interface SERDES to be shared by the ENET(SYS), CBRC, and SYSOTN subsystems. It also provides the flexibility of assigning each of the 12 System Interface SERDES lane to any of the subsystem DSIs using a 12x12 crosspoint mux. Note that one output of the crosspoint mux is not connected to the SYSOTN subsystem. The SYS\_DSIS is identical to the LINE\_DSIS except that the pre-muxing after the crosspoint mux is disabled. The DSIS also allows the ACB (Adaptive Clock Block) outputs from these three different subsystems to be selected and driven to the ACB in each of the 12 System Interface SERDES.

## 5.12 Ethernet Line-side (ENET(LINE)) Subsystem

### 5.12.1 Block Diagram

Figure 120 ENET(LINE) Block Diagram



### 5.12.2 Overview

The ENET subsystem provides centralized services to the device's Ethernet clients and also to Ethernet over OTN. The ingress/receive path is defined to be in the direction from the SERDES/de-mapper to the external packet buffer and the egress/transmit path is from the external packet buffer to the SERDES/mapper.

The ENET(LINE) subsystem contains the major blocks described in [Table 64](#).

Table 64 ENET(LINE) Subsystem Block Summary

Block	Overview
Ethernet Transcoder (ETRANS)	<p>The ETRANS block performs the required translation (by means of encoding and decoding) of 10GE/40GE/100GE traffic, in such a way that the traffic can be mapped to OPU2/OPU3/OPU4 OTN containers.</p> <p>Two broad mappings are supported by the ETRANS block:</p> <ul style="list-style-type: none"> <li>• Payload mapping (GFP-F) mapping, where Ethernet MAC frames are mapped into the OTN container with an 8-byte header. The GFP mappings are done according to Clauses 6.2 and 7.3 of [27] (for 10GE) and Clause 7.1 of [31] (for 40GE/100GE)</li> <li>• Constant bit-rate (GMP/BMP) mapping, where a serial stream is mapped directly into the OTN container. The GMP/BMP mappings are done according to Sections 17.1 and 7.2 of [27] (for 10GE) as well as Annexes B, E and F of [28] (for 40GE and 100GE).</li> </ul> <p>The ETRANS acts as a bridge between the PCS layer (EMAC block) and the OTN mapper/de-mapper. The "PCS" interface is composed of PMA lanes, which are configurable to be independent 10GE links or as part of 40GE or 100GE links.</p> <p>The mapper/de-mapper interface consists of a TDM interface, where all channels are expressed on a common DPI bus.</p>
Ethernet PCS/MAC (EMAC)	<p>The EMAC block implements MLD, PCS and MAC layers for 10GE/40GE/100GE interfaces. The block is channelized and can be configured to support twelve 10GE interfaces, three 40GE interfaces, one 100GE interface and several combined rate modes to form an aggregate of 120 Gbps.</p> <p>An MLD (Multi-Lane Distribution) function in the PCS layer aggregates parallel 10G SERDES interfaces to implement 40 and 100 Gbps Ethernet according to the IEEE P802.3ba specification [24].</p> <p>The PCS layer supports twelve 10GE PCS streams, three 40GE MLD PCS streams, one 100GE MLD PCS stream or any legal combination up to an aggregate of 120 Gbps. Each interface implements the industry standard 64B/66B encoding/decoding [23].</p> <p>The 10GE/40GE/100GE MAC supports Ethernet PAUSE frame functionality (see Annex 31B in [23]).</p> <p>The EMAC is also responsible for accurate time stamp capture on both outgoing and incoming packets to/from the line interface. The time stamp logic provides support for the IEEE 1588v2 timing protocol [22], including both one-step and two-step clock modes.</p>

Block	Overview
Ethernet Timing Shim (ETIME)	<p>The ETIME block contains much of the support logic required for PTP, including the free-running and TOD counters along with external IRIG interfaces.</p> <p>In 2-step clock mode, the time stamp for an outgoing packet is disassociated with the packet transfer so the block provides the necessary logic to realign the time stamp with a duplicate of the original packet for later processing by a CPU. Accurate timing adjustments by firmware are possible as a result of the hardware time stamp capture logic. A series of header/time stamps are inserted/extracted to assist firmware in correlating the time stamps and the associated data packets on both the transmit and receive path.</p> <p>The ETIME block supports PTP Boundary Clock, Peer-to Peer Transparent Clock, and End-to-End Transparent Clock modes.</p> <p>The ENET block can also support Synchronous Ethernet (SyncE) Timing, as defined in G.8261 [37], G.8264 [39] and G.8262 [38], although this function does not involve the ETIME block.</p>
Ethernet Ingress Management Packet Extract / Insert (EMPEI_I)	<p>The EMPEI_I block performs packet identification and supports insertion and extraction of management and timing packets from the channelized ingress data path. Inserted and extracted packet types can include Ethernet Link OAM, LACP, Management VLAN frames and Timing messages (SyncE and PTP), although the block is not limited to these types. Based on their classification, the EMPEI_I redirects/forwards the packets to their intended destination. Extracted packets are sent to the Management switch on a dedicated interface. The EMPEI_I also provides the means to perform management packet insertion into the channelized data path.</p> <p>Packet Identification is performed by a flexible channelized Ethernet Classifier (ECLASS), which can inspect the first 88 bytes of all incoming packets in order to flag any targeted packet types.</p> <p>Scheduling towards the downstream packet buffer between the inserted management packets and ingress packets is done by a Multi-level Scheduler with scheduling based on a fixed or configurable weighted round robin algorithm.</p>
Ethernet Egress Management Packet Extract / Insert (EMPEI_E)	<p>The EMPEI_E block performs packet identification and supports insertion and extraction of management and timing packets from the channelized egress data path. Inserted and extracted packet types can include Ethernet Link OAM, LACP, Management VLAN frames and Timing messages (SyncE and PTP), although the block is not limited to these types.</p> <p>Based on their classification, the EMPEI_E can extract the packets to the management switch. The EMPEI_E also provides a means to perform management packet insertion into the channelized egress data path.</p> <p>Packet Identification is performed by a flexible channelized Ethernet Classifier (ECLASS), which can inspect the first 88 bytes of all incoming packets in order to flag any targeted packet types.</p> <p>Scheduling towards the transmit MAC between the Management switch and the upstream packet buffer is done by a Multi-level Scheduler with scheduling based on a deficit-round-robin algorithm.</p>

Block	Overview
Ethernet Packet Classifier (ECLASS)	<p>The ECLASS block performs packet classification by inspecting the first 88 bytes of all incoming packets. The classification engine is fully programmable and can be used to concurrently search for any packet types that are identifiable by their first 88 bytes. Although the ECLASS has finite classification resources, it has been conceived to be able to classify all of the following packet types concurrently: Ethernet Link OAM, LACP, Management VLAN, SyncE, PTP (event and non-event), PTP over UDP over IPv4 (event and non-event), and PTP over UDP over IPv6 (event &amp; non event). Also, the ECLASS has enough spare classification resources to support future expansion and to support future protocols.</p> <p>The ECLASS operates on a channelized basis, and each channel can be configured to flag different packet types. Different destination addresses, VLAN IDs and other criteria can be searched for on each channel.</p> <p>The data output from the ECLASS is identical to the input, but includes a side-band classification result that can be used in downstream routing and scheduling.</p>
Ethernet Packet Maker/Monitor (EPMM)	<p>The EPMM block provides transparent data paths that bypass the Ethernet MACs. In the ingress direction, data from the SERDES links is mapped into packets towards the downstream packet buffer. In the egress direction, packets from the upstream packet buffer are mapped towards SERDES links.</p> <p>Two types of transparent data paths are provided:</p> <ul style="list-style-type: none"> <li>• Payload transparent (for 10GE paths only)</li> </ul> <p>The payload transparent paths for 10GE channels are payload mapped according to GSUp43 Clause 7.3 (see Section 7.1 of <a href="#">[31]</a>). In the ingress direction, the 10GE streams are de-scrambled, 64B/66B-decoded, and any inter-packet gap is removed, leaving only Ethernet frames (including the preamble) and ordered sets. The reverse operation is done in the egress direction.</p> <ul style="list-style-type: none"> <li>• Bit-transparent (for 10GE/40GE/100GE paths)</li> </ul> <p>Bit-transparent paths are transcoded/transdecoded according to ITU-T GSUp43 Clause 7.1 (see Section 7.1 of <a href="#">[27]</a>) for 10GE paths, according to Section 17.7.4.1 and Annexes B, E and F of <a href="#">[28]</a> for 40GE paths, and according to Section 17.7.5.1 and annex E of <a href="#">[28]</a> for 100GE paths.</p> <p>In the ingress direction, the transcoded data channels are passed through a packet maker, which slices each channel into arbitrarily-sized artificial packets. The size of these artificial packets can be modulated to be nominal +/- 1 bytes in order to encode the rate of the incoming SERDES client. The nominal size (B) of the artificial packets is programmable.</p> <p>In the egress direction, bit transparent channels pass through a packet monitor block, which reforms the serial streams from the artificial packets. In addition, the packet monitor block recovers timing information encoded in the artificial packets' size in order to control the rate of the outgoing SERDES client. The output of the packet monitor is then transdecoded in order to deliver data to the egress SERDES links.</p>

## 5.12.3 Features

This section describes the main features of the ENET as well as features not clearly described from the standards or the requirements.

### 5.12.3.1 Supported Rates

The ENET supports any of the following combination up to an aggregate of 120 Gbps:

- Up to twelve 10 Gbps Ethernet links
- Up to three 40 Gbps Ethernet links
- Up to two 40 Gbps and up to four 10GE links
- One 40 Gbps and up to eight 10GE links
- One 100GE Ethernet link

### 5.12.3.2 Ingress Data Path Flow

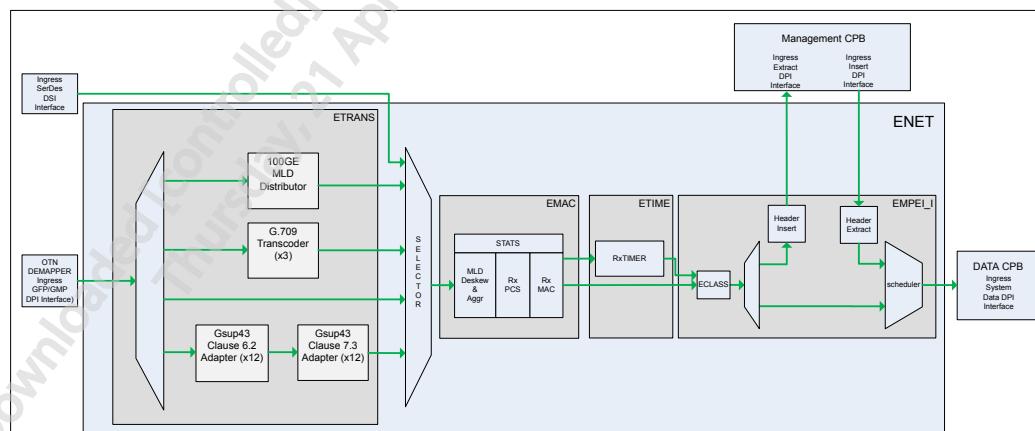
The ingress data path has two internal paths as described in the following subsections:

- From the SERDES or the de-mapper to an external packet buffer or processor interface via the Rx PCS and Rx MAC
- From the SERDES to an external packet buffer via the packet maker (EPMM)

#### 5.12.3.2.1 Through the Rx PCS and Rx MAC

This section describes the path taken by the ingress packet from the SERDES or de-mapper for the Ethernet clients from the OTN to either the external packet buffer/CPU/external management processor through the Rx PCS and Rx MAC. The figure below shows the ingress data path through the Rx PCS and Rx MAC. In this path, the Ethernet protocol is terminated in the Rx PCS and Rx MAC layer.

Figure 121 The Termination Mode Ingress Data Path



The source for ingress data path is either the SERDES or the OTN de-mapper. In the former case, the PMA lines directly connected to the Rx PCS layer through the static selector. In the latter case, traffic from the de-mapper is a channelized input on a 384-bit wide DPI data bus that can carry GFP-F or GMP mapped traffic on a channelized basis. The DPI is time division multiplexed over a maximum of 12 channels. On the DPI, the SOPs are expected to be MSB-aligned, which results in an expansion of bandwidth for GFP-F. Thus, the interface to the de-mapper is over-clocked. The frames go through

transdecoders or MLD distributors before being transformed into PMA lanes to the Rx PCS layer through a static selector.

The 10GE, when mapped into GFP-F like frames in an extended-payload OPU2 container as explained in ITU-T Gsup43 Clause 7.3 (see Section 7.1 of [31]), is de-mapped and shared as a packet to the ENET. These packets are sent on an over-clocked interface. This along with the format of OPU2 mapping in ODU2 results in non-continuous packets.

The GSup43 Clause 7.3 adapter logic for each of the 12 links has a Tx Formatter, which simply takes the signals on the GSUP43's line interface and, based on the GFP payload header UPI field, identifies the type of information as a data packet or an ordered set. It also identifies characteristics of these signals such as their SOP and EOP positions. This information is passed to the subsequent blocks and the performance monitor counters. On the ingress data path, the Gsup43 block controls the inter packet gap (IPG) between packets and ordered sets based on the fill level of its upstream packet buffer. By default, an average of 12 bytes of IPG are inserted between packets and ordered sets. The 64B/66B encoder inside the adaptation logic implements all the required functionality of the 10GE PCS (Physical Coding Sublayer) as specified in the IEEE 802.3 specification. It performs frame delineation on frames, scrambling, and an 64B/66B encode function on the scrambled stream. In IEEE mode, the 64B66B input stream consists of standard Ethernet (Preamble/SFD, MAC Header, Data, Pad and CRC) frames, ordered sets, and inter-packet gaps.

The 10GE, when mapped into GFP-F-like frames in the OPU2 container as defined in ITU-T Gsup43 Clause 6.2 (see Section 6.2 of [27]), is de-mapped and shared as packets to the ENET. The Gsup43 Clause 7.3-mapped and Gsup43 Clause 6.2-mapped data from the de-mapper shares the same DPI. The GSup43 Clause 6.2 adaptation logic for each of the 12 links prepends preamble and SFD to each packet as well as insert IPG. The data stream then flows through the GSup43 Clause 7.3 block and is passed to the EMAC as a 64b66b-encoded PCS stream.

40GE and 100GE can also be GFP-F-mapped. In much the same way as 10GE can be mapped into an OPU2 as defined in ITU-T Gsup43 Clause 6.2, 40GE can be mapped into an OPU3, and 100GE can be mapped into an OPU4 according to ITU-T G.7041 Clause 7.1 (see Section 7.1 of [31]). The GFP-F frames have preamble, SFD and IPG inserted. The data stream then passes through the 40G/100G blocks in order to be properly encoded (PCS and MLD distribution according to [24]). The encoded and distributed stream is then passed onto the EMAC as PMA lanes (4 lanes for 40G and 10 lanes for 100G ).

The ENET also supports GMP/BMP-mapping of Ethernet, otherwise referred-to as bit-transparent mapping.

For 10GE traffic in bit transparent mode as defined in ITU-T Gsup43 Clause 7.1 (see Section 7.1 of [27]) the de-mapped 10GE stream is translated to 20-bit wide PMA lanes and forwarded to the PCS layer.

40GE traffic, which is mapped to OPU3 container, is routed through the G.709 1027B transdecode logic. The G709's transdecode logic will convert the 1027-bit data into a stream of 64/66B and the MLD logic inside the transdecode block, which distributes this 64/66B stream as four streams of 66 bits as specified in IEEE 802.3ba [24], along with the insertion of alignment markers. The output is a 66-bit data bus, which is converted to a 20-bit-by-32-bit-to-20-bit converter and forwarded to the selector, then routed to the PCS layer of the EMAC through the selector.

For 100GE traffic, when mapped to OPU4 container, the MLD logic identifies the marker, aligns the markers and distributes the 64/66B blocks into ten 66-bit wide data buses along with the insertion of an alignment marker as per IEEE P802.3ba [24]. The output is a 66-bit data bus, which is converted to a 20-bit-by-32-bit-to-20-bit converter and forwarded to the selector, then routed to the PCS layer of the EMAC through the selector.

An MLD function in the PCS layer aggregates the parallel 10G SERDES interfaces to implement 40GE and 100GE according to the IEEE P802.3ba specification. The PCS layer supports twelve 10GE PCS streams, three 40GE MLD PCS streams, one 100GE MLD PCS stream or most combinations up to an aggregate of 120 Gbps (100GE +2x 10GE is an invalid combination). Each interface implements the industry standard 64B/66B encoding/decoding as specified in Clause 89 of IEEE 802.3ba [24].

The 10GE/40GE/100GE MAC supports Ethernet PAUSE frame functionality (see Annex 31B in [23]).

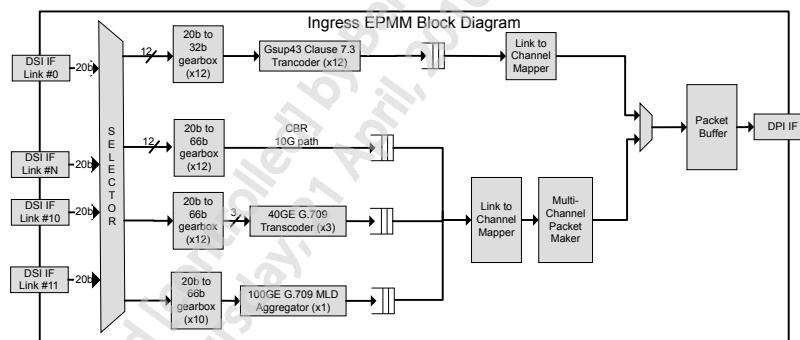
The data path to and from the EMAC is channelized and the single wide data bus carries data for all the links in a TDM fashion.

A channelized block of common Ethernet functions provides identification, insertion and extraction of any Ethernet frame type, including Link OAM, Timing, LACP, and Management VLAN frames. Ethernet Type identification is performed by a channelized Ethernet Classifier that parses the data, looking for user-configured pattern matches on any of the first 96 bytes of the non-tagged Ethernet frame (92 bytes for tagged and 88 bytes for stacked VLAN Ethernet frames), which, in this case, starts at the DA field. The bytes prior to the DA field are removed by the receive Ethernet MAC. The Ethernet Classifier can be configured to inspect and flag many packet types using combinations of patterns with individual masking options for each bit. The classified packets are forwarded to the external management packet buffer (ingress Extract interface) or to the CPB (ingress Data interface).

### 5.12.3.2.2 Through the Packet Maker

The ENET subsystem supports direct mapping of 64b/66b data streams from the SERDES to the external packet buffer through the channelized packet maker (EPMM). The figure below shows the ingress data path through the packet maker.

**Figure 122 The Transparent Mode Ingress Data Path**



There are two transparent path modes available:

- Payload transparent mode for 10GE channels
- Bit transparent mode for 10GE/40GE/100GE channels

In payload transparent mode (which is only available for 10GE channels), the 10GE bit stream is converted to a 32-bit wide data bus and passed on the G.Sup43 clause 7.3 Adapter (see Section 7.1 of [31]), which converts the serial stream into Ethernet frames. The Gsup43 clause 7.3 adapter finds the 64B/66B frame position, implements a 64/66-bit Control Code encoder function for the IEEE 802.3 (see Clause 49 of [23]) Ethernet MAC layer at 10GE link speeds, descrambles the incoming data stream with a 1+ X39 + X58 Self-Synchronizing Scrambler, implements a 33:32 rate adaptation function, and removes IPG from data stream before passing the packet data downstream. This block also differentiates between packet data and ordered set control codes, marks SOP and EOP for packet data and ordered

set control codes according to Gsup43 Clause 7.3 [27]. The output of the GSup43 Clase 7.3 adaptation block is channelized to meet the requirement of the DPI.

The ENET subsystem also supports mapping of Ethernet traffic in bit transparent mode for 10GE, 40GE and 100GE channels:

- For bit-transparent mapping of 10GE, as defined in ITU-T Gsup43 Clauses 7.1 and 7.2 (see Sections 7.1 and 7.2 of [27]), data from up to 12 links is channelized and forwarded to the packet maker. The Ethernet packet maker gathers the consecutive bytes of data into artificial packets, which are written into a channelized FIFO before being sent downstream on the DPI. Artificial packets are constructed by inserting SOP and EOP along with the necessary control signals. The size of the artificial packets can be modulated to be C-1, C, or C+1 bytes in length to encode the rate of the incoming client. The nominal size (C) of the artificial packet and the observation period (T) for relative rate measure are software configurable.
- For 40GE channels, the PMA data is received on four links, each 20 bits wide. The Rx PCS block finds the 64/66B frame positions and alignment markers on each of these links. It aligns the 64/66B frames from the four links into a single stream (see Clauses 82 and 83 of [24]). This block converts the 64/66B frame into 1024/1027 bit frames (see Section 17.7.4.1 and Annexes B, E and F of [28]) before forwarding the channelized packet maker, which converts the transcoded 40GE data stream into artificial packets before they are time division multiplexed and sent downstream on the DPI towards a CPB.
- For the 100GE channel, the PMA data is received on 10 links, each 20 bits wide. These 10 links are converted to 20 PMA lanes by segregating odd and even bits (Refer to Clauses 23 and 83 of [24]). The Rx PCS/MLD block converts these 20 PMA lanes into a single stream of 64/66B data, according to Section 17.7.5.1 and Annex E of [28]. This stream is sent to packet maker to convert this 100GE data stream into artificial packets, which are forwarded to external packet buffer before being time division multiplexed and sent downstream on the DPI towards the CPB.

### 5.12.3.3 Egress Data Path Flow

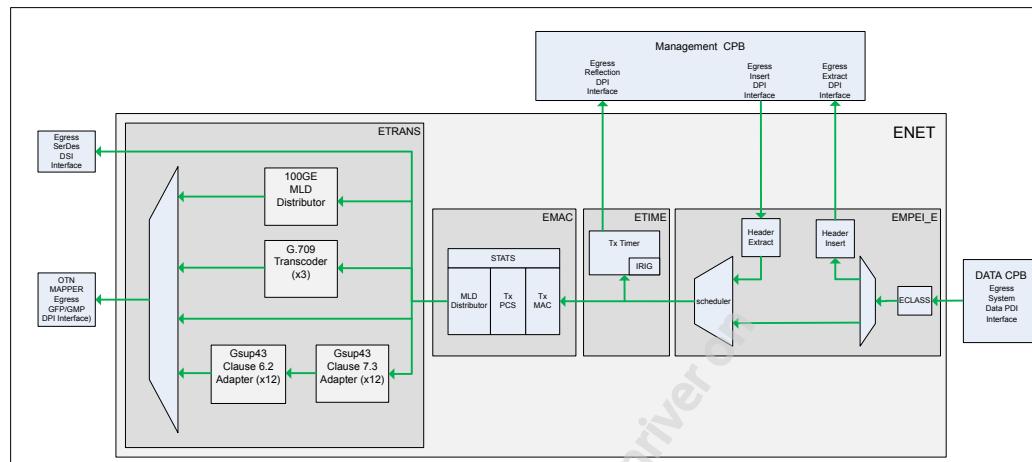
The egress data path has two internal paths as described in the following subsections:

- From the external packet buffer/processor interface to the SERDES or mapper via the Tx PCS/Tx MAC
- From the external packet buffer to SERDES via the packet moniter (EPMM)

#### 5.12.3.3.1 Through the Tx MAC and Tx PCS

This section describes the path taken by egress packets from the CPB to the SERDES for Ethernet clients or Ethernet packets to be mapped into the OTN through the Tx MAC and Tx PCS. The figure below shows the egress data path.

Figure 123 The Termination Mode Egress Data Path



The source to egress data path can be either Ethernet packets from the CPB or inserted packets (timing, OAM, etc.) from either the CPU or an external management processor. The header information is extracted from the packets received from the CPU/external management processor before being scheduled for transmission. These packets from the external buffer/CPU/external management processor are scheduled for transmission after channel-to-link mapping is done. The classification unit parses packets from the external packet buffer and generates an output tag, which identifies the type of packet. Based on the classification tag, the packets may be either forwarded to the Tx MAC or to the CPU or external management processor. Inside the EMAC, the Ethernet packets are formed by adding the preamble, SFD and FCS fields along with any required pad fields before forwarding them to the PCS layer. If the packet is a timing event packet, the MAC can additionally update the Time stamp or Correction field before framing the Ethernet packet.

In the Tx PCS layer, the PCS function implements all required functionality of the 10GE PCS sublayer interface as specified in IEEE 802.3 (see Clause 49 of [23]) and the 40GE/100GE PCS as specified in IEEE P802.3ba (see Clause 89 of [24]). This block implements a 64/66-bit encoder function along with control code mapping function along with de-scrambling. The block also performs optional insertion and extraction of ordered sets and Virtual Lane Alignment marker insertion for 40G/100G (MLD) along with round robin block distribution.

Each data link is driven to the SERDES or passed on to the Transcode block to enable successful mapping into OPU2/OPU3/OPU4 containers.

There are several transcoding options available. 10GE links can be adapted towards GFP-F according to Gsup43 Clause 7.3 (see Section 7.1 of [31]). The Gsup43 Clause 7.3 adapter block finds the 64B/66B frame position, implements a 64/66-bit Control Code encoder function for the IEEE 802.3 Ethernet MAC layer at 10GE link speeds, descrambles the incoming data stream with a 1+ X39 + X58 Self-Synchronizing Scrambler, which bypasses the 2-bit sync header and removes IPG from data stream. The block will finally mark SOP and EOP for packet data and ordered set control codes before the data is channelized onto the DPI bus towards the GFP-F mapper.

10GE, 40GE and 100GE can also be mapped into mapped GFP-F as defined in ITU-T Gsup43 Clause 6.2 (see Section 6.2 of [27]) for 10GE, and according to ITU-T G.7041 Clause 7.1 (see Section 7.1 of [31]) for 40GE and 100GE. For these paths, the Ethernet payload is tapped after the Tx MAC, the preamble and SFD are stripped, and ordered sets are discarded. The resulting data path is channelized on to the same DPI bus towards the GFP-F mapper.

The ENET also supports GMP/BMP mapping of 10GE traffic in bit transparent mode as defined in ITU-T Gsup43 Clauses 7.1 and 7.2 (see Sections 7.1 and 7.2 of [27]). In this mode, CBR data is packed into 384-bit artificial packets so that the data stream can be channelized along with the GFP-F channels and sent downstream to the OTN mapper on a common DPI bus.

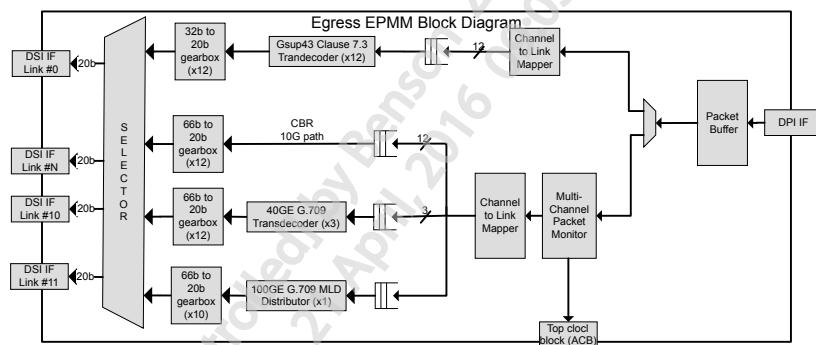
For 40GE GMP mapping, the 64B/66B Tx PCS block finds 64/66B frame positions and alignment markers. It aligns the 64/66B frames from the four links into a single stream. This block converts the 64/66B frame into 1024/1027-bit frames, as described in Appendix VIII of G.709 [28] before forwarding these to the OPU3 mapper. In this mode, the CBR data is packed into 384-bit artificial packets so that the data stream can be channelized along with the GFP-F channels and sent downstream to the OTN mapper on a common DPI bus.

For 100GE GMP mapping, the 64B/66B Tx PCS block finds 64/66B frame positions and alignment markers. It aligns the 64/66B frames from 10 links into a single stream and forwards these to the OPU4 mapper. Again, the CBR data is packed into 384-bit artificial packets so that the data stream can be sent downstream to the OTN mapper on a common DPI bus.

### 5.12.3.3.2 Through the Packet Monitor

The ENET subsystem supports direct mapping of packetized data from an external packet buffer to the SERDES as a 64b/66b data stream through the channelized packet monitor (EPMM block). The figure below shows the egress data path through the packet monitor.

Figure 124 The Egress Data Path Through the Packet Monitor



Two types of transparent paths are offered:

- Payload-transparent mapping for 10GE links
- Bit-transparent mapping for 10GE/40GE/100GE links

For 10GE paths, a payload-transparent mapping is available where packets are passed through the Gsup43 clause 7.3 de-adapter (see Section 7.1 of [31]). The Gsup43 clause 7.3 de-adapter for each link accepts packets and ordered sets from an upstream packet buffer, inserts an appropriate IPG and performs the 10GE PCS as specified in the IEEE 802.3 specification before passing the stream towards the SERDES links. By default, an average of 12 bytes of IPG is inserted between the packets and ordered sets. The output is pushed out on a 32-bit data bus and is converted to 20-bit before forwarding to the transmit SERDES.

A bit-transparent mapping is also available for 10GE/40GE/100GE links:

- For 10GE, Ethernet traffic is mapped as defined in ITU-T Gsup43 Clause 7.1 (see Section 7.1 of [27]). In this mode, the artificial packets are read from the external packet buffer and a 64b66b data

stream is reconstructed by the packet monitor for each of the 12 links. The 66-bit-to-20-bit converter converts the 66-bit data to 20 bits and push it to the SERDES.

- For 40GE, artificial packets are read from the external packet buffer and the packet monitor reconstructs the 1027B-encoded data stream. This data stream is passed through the G.709 1027B transdecoder logic, which converts the 1027-bit data into a stream of 64/66B-encoded blocks according to Section 17.7.4.1 and Annexes B, E and F of [28]. The MLD logic inside the transdecoder distributes this 64/66B stream as four streams of 66 bits as specified in Clauses 82 and 83 of [24]. The data is then pushed to the SERDES lanes over a 20-bit DS1.
- For 100GE, artificial packets are read from the external packet buffer and a 64b66b data stream is reconstructed by the packet monitor. The 64b66b data stream is distributed to 20 PCS lanes and, from these lanes, 10 serial lanes are derived. Refer to Section 17.7.5.1 of [28] and Clauses 82 and 83 of [24]. The data is then pushed to the SERDES lanes after being gearboxed to the appropriate data width (20 bits).

#### 5.12.3.4 Ethernet Over OTN

The ETRANS block services Ethernet clients to carry them over OTN. This section describes the mechanism provided by the ETRANS to enable mapping of 10GE/40GE/100GE client signals into OPU2/OPU1e/OPU2e/OPU3/OPU4, which can fit into the payload cavity of ODU2/ODU1e/ODU2e/ODU3/ODU4 or OTU2/OTU1e/OTU2e/OTU3/OTU4 of an OTN frame.

##### 5.12.3.4.1 Transporting 10GE/40GE/100GE Over OTN Using GFP-F Encapsulation

This section explains the mapping mechanism used to transport a 10G/40G/100G BASE-R (LAN PHY) Ethernet payload only into OPU2/OPU3/OPU4 using the GFP-F encapsulation mechanism. A payload transparent mapping can be performed by terminating the 64B/66B line code, the preamble, SFD, and IPG control characters present in an Ethernet packet defined per IEEE 802.3 and then applying GFP-F framing as described under Clause 7.1 of [31] and Clause 6.2 of [27].

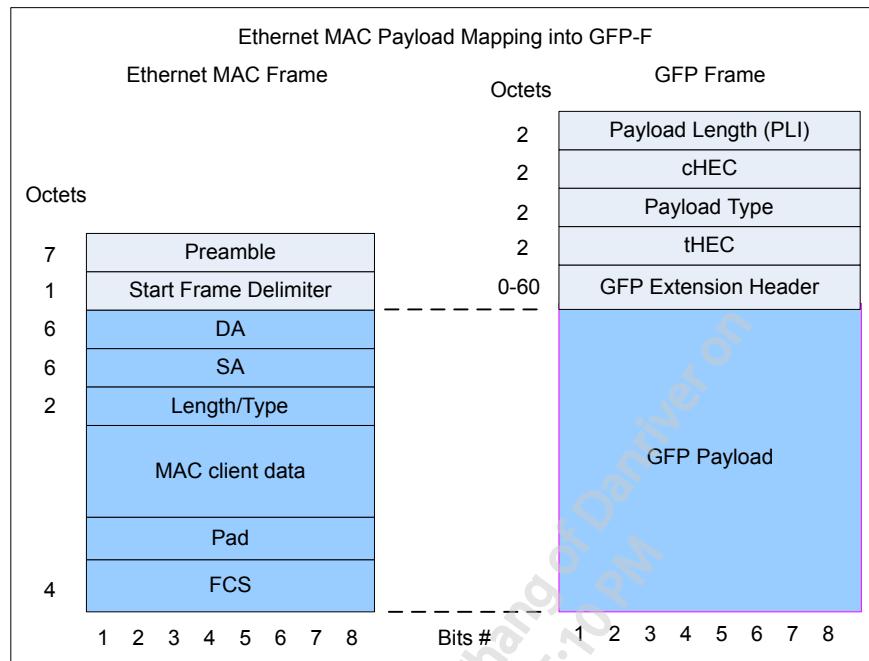
To transport the 10GE/40GE/100GE packet data over OTN using GFP-F encapsulation, the ETRANS performs the following functions:

1. Strips 7-bytes of preamble, 1-byte SFD and IPG control characters on each received Ethernet packet.
2. Transmits the payload portion of data carried in the Ethernet packet (starting from the DA to FCS) to the GFP mapper with 8-bytes of hole at the SOP.
3. Encodes and transmits an 8-bit UPI value into the TYPE[15:8] fields to indicate to the GFP mapper about the type of packet.
4. Transmits a 2-byte packet length value to the GFP mapper through a customized DPI.

The valid UPI value for this kind of frame-mapped Ethernet is 8'b0000\_0001 as defined in Table 6.3 of [31].

Figure 125 below shows the relationship of encapsulating the Ethernet MAC frames using the GFP-F mechanism.

Figure 125 Ethernet Client Data Frame Mapping into GFP-F



#### 5.12.3.4.2 Transporting 10GE Over OTN

The maximum rate of the OTU2 signal is less than the rate of the 10GE LAN signal, which is 10.3125 Gbps after 64/66B encoding. There are various methods that have been proposed to meet this challenge. The most popular methods are documented in the ITU-T supplementary document G.sup43 [27]. The mapping methods are explained in detail as follows.

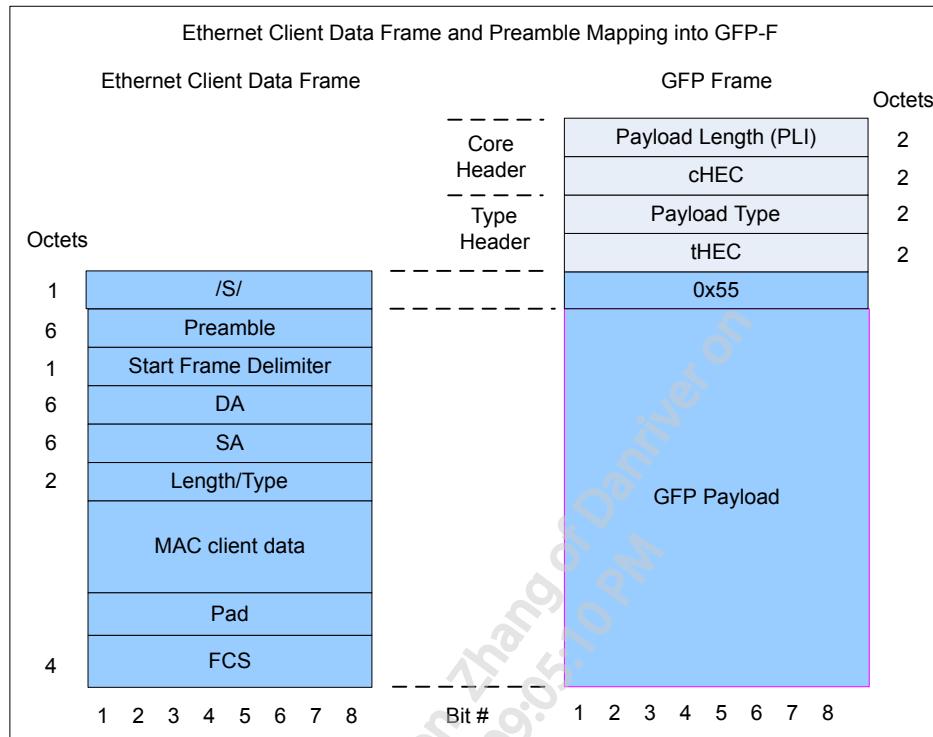
##### 5.12.3.4.2.1 Transporting 10GE with Preamble Over OTN Using GFP-F Encapsulation

This kind of framing mechanism is defined for applications where there is a requirement to preserve the Ethernet frame preamble and ordered set information when the 10GBASE-R signal is carried by GFP-F in addition to the Ethernet payload data. The mapping of both Ethernet client data frames and ordered sets into GFP-F frames is described in this framing mechanism described in Clause 7.9 of [31] and Clause 7.3 of [27]. The GFP-F uses an UPI field to indicate whether the GFP-F framing is done for Ethernet data or ordered sets.

This mapping includes the Ethernet frame preamble information in the GFP-F OPU2 extended payload area along with the client Ethernet data frame as shown in below Figure 126. As specified in Clause 46.2 of IEEE 802.3, the preamble consists of seven octets beginning with the /S/ (start) control character and followed by the SFD (start of frame delimiter) character. Since the /S/ control character is always present at the beginning of the preamble, as shown in below Figure 126, it is encoded to a fixed value of 0x55 when it is transmitted to the GFP-F mapper for GFP-F framing.

To enable this kind of framing, the ETRANS detects and transmits preamble, SFD and Ethernet payload to the GFP-F mapper. It also encodes and transmits a valid UPI value which is 8'b0001\_0011 as defined in Table 6.3 of [31]. The ETRANS strips the control codes such as idles, error and reserved codes. It also transmits the packet length value to GFP mapper. The ETRANS uses a customized DPI to transmit data and corresponding control signals to the GFP mapper present in OTN mapper subsystem.

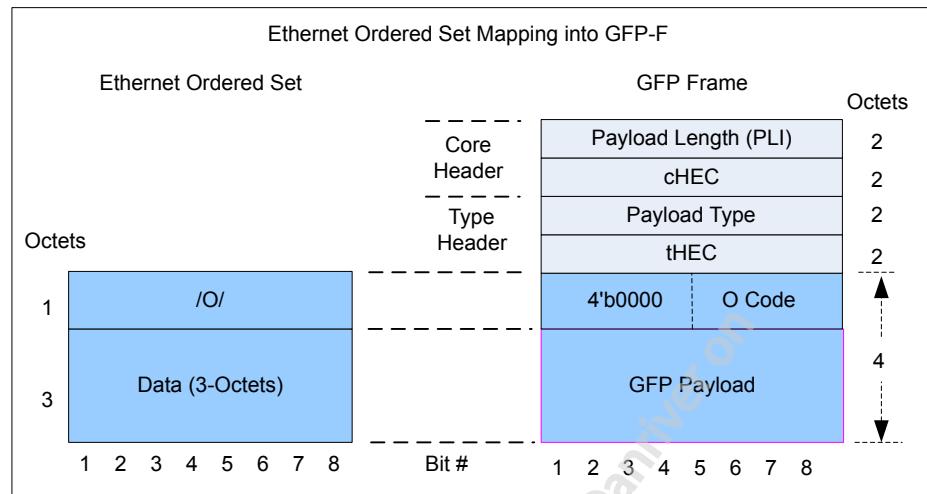
Figure 126 Ethernet Client Data Frame and Preamble Mapping into GFP-F



This kind of mapping also preserves the 10GE ordered set as defined in Clause 49.2.4 of IEEE 802.3. An ordered set consists of four octets, beginning with a special character (/O/) followed by three data octets. Each ordered set is mapped into its own GFP-F frame as shown below in [Figure 127](#). The first octet of the ordered set has the four most significant bits set to all-zeros and the four least significant bits equal to the /O/ code. This way both sequence ordered sets (O code = 0000) and signal ordered sets (O code = 1111) can be transferred. The next three octets contain the three data bytes of the ordered set. The valid UPI value, for this kind of frame-mapped 64B/66B encoded Ethernet ordered set information, is 8'b0001 0100 as defined in Table 6.3 of [\[31\]](#).

In this kind of mapping the control codes such as idles, error and reserved codes are not mapped into GFP-F framing.

Figure 127 Ethernet Ordered Set Mapping into GFP-F

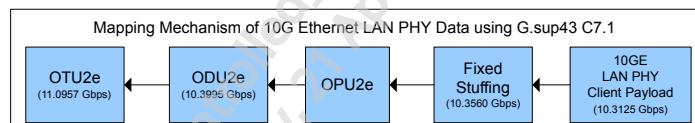


#### 5.12.3.4.2.2 Transporting Constant Bit Transparent 10GBASE-R Over OTN Using Over-clocking

This section explains the mapping of a 10G CBR Ethernet client signal into OPU2e or OPU1e by increasing or over-clocking the transmission clock rate at a nominal bit rate of 11.0957 Gbps (OPU2e) or 11.0491 Gbps (OPU1e) to accommodate the bit transparent mapping of the 10GBASE-R signal, thus allowing the full 10G LAN PHY data to be transparently transmitted to the OPU2e/OPU1e payload cavity as defined in Clauses 7.1 and 7.2 of [27].

Since over-clocking is used, this kind of mapping is designated as OPU2e and OPU1e, where 'e' stands for extended mapping by over-clocking. Figure 128 below illustrates this bit transparent mapping mechanism.

Figure 128 Bit Transparent Mapping of a 10GBASE-R Signal into OPU2e



#### 5.12.3.4.3 Transporting 40GE Over OTN

This section explains the mechanism used to map and transport 40GE client signals into OPU3 payload cavity of an ODU3 or OTU3 frame.

The 40GE is transported by using four 10G links, with multi-line distribution (MLD) on 64/66B blocks as specified in Clause 82 of [24]. The 40GE, effectively uses the 64B/66B encoding as in 10GE, which results in a 41.25 Gbps serial rate. Since the OPU3 payload rate is 40.150519 Gbps for transport over ODU3, the 64B/66B blocks are transcoded into a more efficient 1024B/1027B block code, which results in a client signal rate of 40.117088 Gbps. The resulting 40.117088 Gbps stream is mapped into a standard-rate OPU3 using the Generic Mapping Procedure (GMP). Thus it is easier to achieve full character-level and timing transparency without using an over clocked ODU3 for 40GE.

To enable mapping of 40GE over OTN, the ETRANS performs the following functions:

1. Performs 64B/66B block encoding/decoding.
2. Encodes 512b/513 blocks using eight 64B/66B block and framing.

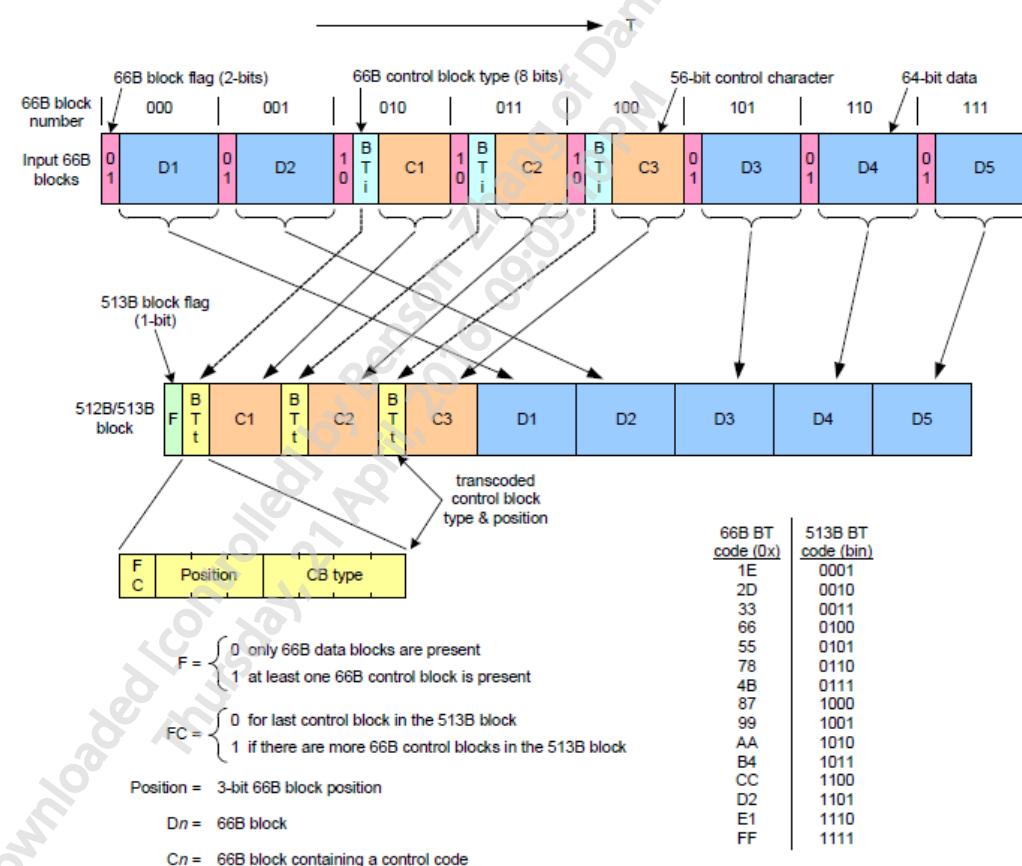
3. Encodes two 512b/513b blocks, to form 1024b/1027b blocks.
4. Performs a lane deskew and alignment function.

The 1024B/1027B block code is constructed as a concatenation of two 512B/513B block codes with an additional synchronization bit added as a parity check over the flag bits of the two 512B/513B blocks.

The 512B/513B block code is an extension of the transcoding technique described in Annex-B of [28], used to create the 64B/66B block code. The 513B block flag bit indicates whether any 66B control characters are present in the block. The control characters are moved to the beginning of the block, with each preceded by fields that indicate the control character's original location within the input stream, a compact encoding of the control character type, and a flag continuation (FC) bit to indicate whether this is the last control character in that 513B block.

The 512B/513B block construction is illustrated below in [Figure 129](#).

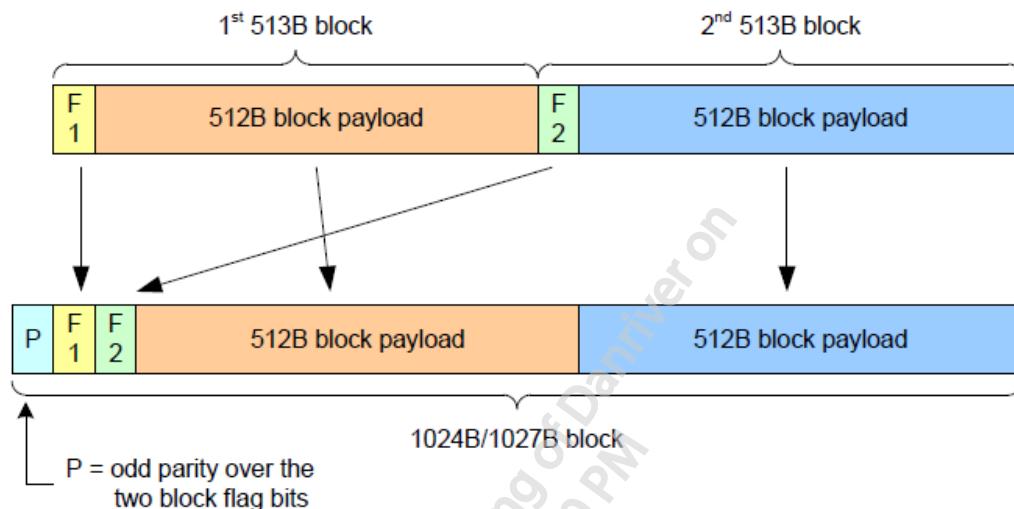
**Figure 129 64/66B Block to 512B/513B Block Conversion**



In general, the bandwidth efficiency of block codes is increased by reducing the amount of redundant information contained within the codes, which in turn increases their potential vulnerability to undetectable bit errors. Various techniques to detect corrupted 512B/513B codes are described in Annex B of [28]. An error that corrupts the 513B flag bit can lead to the corruption of a substantial amount of data, which can be difficult to detect. The OPU3 bandwidth is not sufficient to use a pair of flag bits per block as is done with 64B/66B. Consequently, the structure of [Figure 130](#) below was adopted. Two

513B blocks are concatenated such that their flag bits are moved to the beginning of concatenated pair with an odd parity bit added to cover the flag bits.

Figure 130 512B/513B to 1024B/1027B Block Construction



As per the definition in Clause 83 of IEEE 802.3ba [24], the 40GE interface is a parallel interface using four 10GE parallel lanes with each lane at a rate of 10 Gbps. Each lane uses 64B/66B characters and includes a periodic special 66b control block. This special control block serves as a lane marker to identify the lane and is also used by the receiver for timing deskew between the lanes. For transmission over ODU3, the characters from the four lanes are reassembled into their proper order in a serial stream. The lane marker control blocks are preserved in their proper locations by mapping them into the 1024B/1027B blocks in the same manner as other 66B control blocks. The ODU3 receiver can then directly de-interleave the characters into a parallel interface with the lane markers again in their proper positions.

Refer to Section 17.7.4.1 and Annexes B, E and F in [28] for a complete overview of the 40GE transcode & transdecode function.

#### 5.12.3.4.4 Transporting 100GE Over OTN

This section explains the mechanism used to map and transport 100GE client signals into the OPU4 payload cavity of an ODU4 or OTU4 frame.

The OTU4 rate is around 111.809 Gbps and has an advantage of compatibility with the 100GE rate that includes the hope that OTU4 will become the preferred method for serial transmission of 100GE whenever FEC is required. There is no transcode logic required to reduce the data rate requirement to support the 100GE. 100GBASE-R clients are parallel interfaces and are comprised of 20 PCS lanes. If the number of physical lanes on the interface is fewer than the number of PCS lanes, the appropriate number of PCS lanes are bit-multiplexed onto each physical lane of the interface. Each PCS lane consists of 64B/66B encoded data with a PCS lane alignment marker inserted on each lane once per "16383" 66b blocks. The PCS lane alignment markers themselves are formed from a known pattern that is defined to be balanced and with many transitions and therefore scrambling is not necessary for the alignment markers.

Client framing recovery consists of the following:

- Segregates the 10 PCS links into 20 PMA lanes by bit-demultiplexing each link

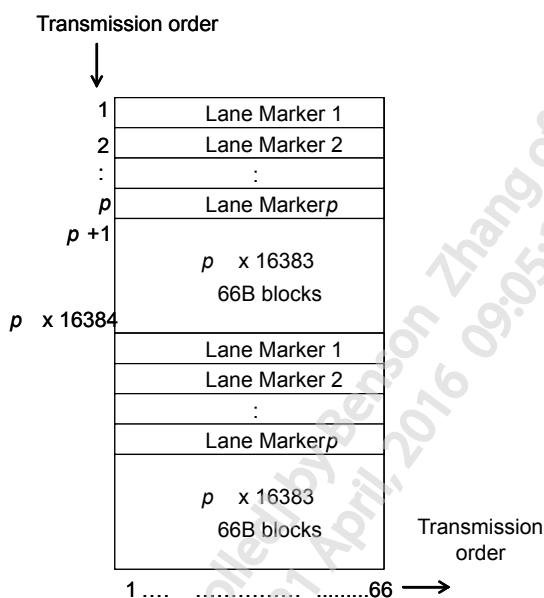
- Recovers the 64b66b block lock as per IEEE 802.3ba D2.2
- Recovers lane alignment marker framing on each PCS lane as per IEEE 802.3ba D2.2
- Deskews and reorders the PCS lanes into a serialized stream of 66B blocks (including lane alignment markers)

To enable mapping of 100GE over OTN, the ETRANS performs the following functions:

- 64B/66B block lock/synchronization
- 64B/66B block encoding/decoding
- PCS lane deskew and alignment, etc.

[Figure 131](#) below illustrates the ordering of 66B blocks after the completion of this process for an interface with ' $p$ ' PCS lanes.

[Figure 131 100GE Transmission Order](#)



For a complete overview of the 100GE mapping and demapping to/from ODU4, refer to Section 17.7.5.1 and Annex E of [\[28\]](#).

### 5.12.3.5 Flow Control

As defined in Annex 31B of 802.3 [\[23\]](#), the ENET subsystem supports link-based flow control (link pause).

#### 5.12.3.5.1 Link-based Flow Control

All full-duplex physical ports support link-level pause frames (see Annex 31B IEEE 802.3 [\[23\]](#)), which is used to back pressure the remote link transmitter in the event that the receive path is being overrun. The PAUSE operation is used to inhibit transmission of data frames for a specified period of time. The PAUSE command used to inform the remote transmitter that it is required to pause the transmission of packets until the time specified in the PAUSE frames (16-bit PAUSE quanta with a non-zero value) or until a resume command is received with a PAUSE quanta of 0. The EMAC is responsible for the

automatic generation and termination of Link PAUSE frames. The globally assigned 48-bit multi cast address, 01-80-C2-00-00-01, has been reserved for use with MAC Control PAUSE frames.

**Table 65** below shows the MAC PAUSE Control frame format.

**Table 65 MAC Link Pause Control Frame**

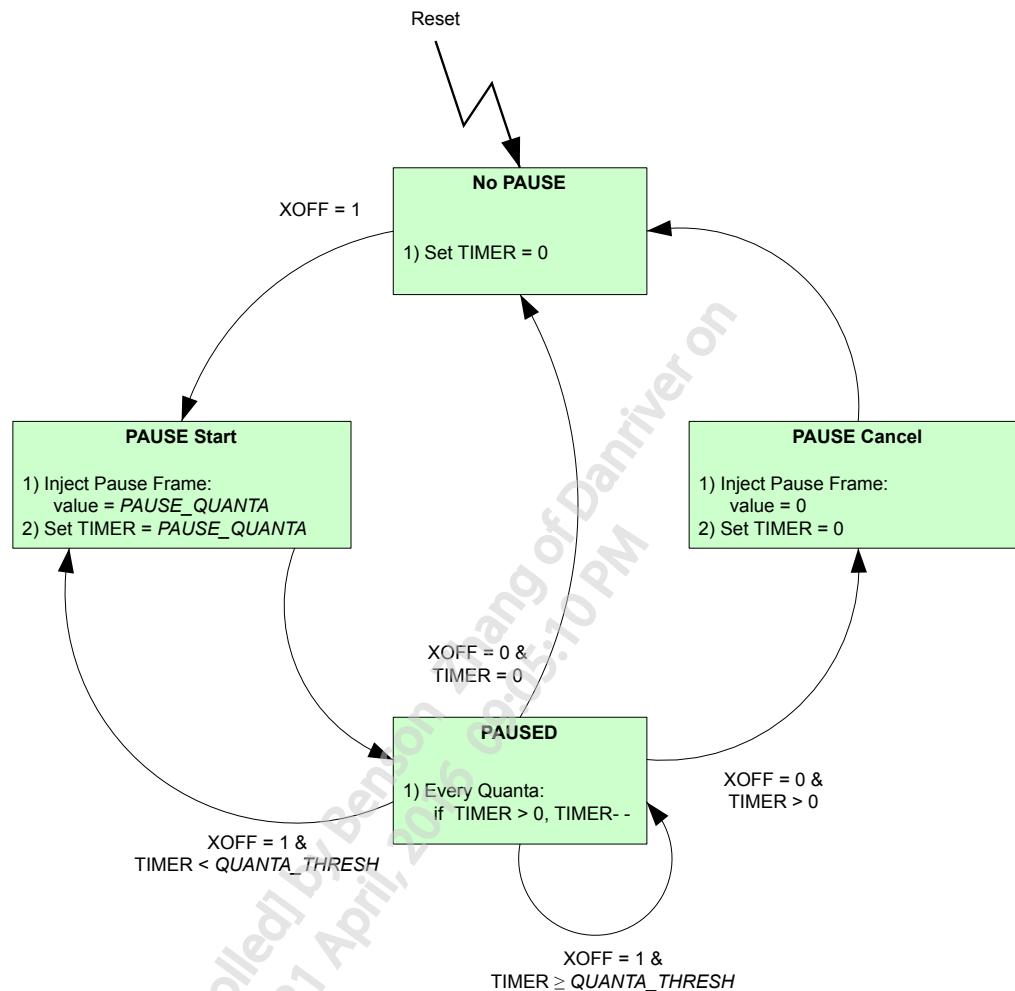
Number of Octets	Description
7 Octets	Preamble( All x55s)
1 Octet	SFD ( xD5 )
6 Octets	Destination Address(x01-80-C2-00-00-01)
6 Octets	Source Address
2 Octets	Type (x8808)
2 Octets	Opcode(x0001)
2 Octets	Pause Quanta
42 Octets	Pad ( All x00s
4 Octets	FCS

The PAUSE quanta is a 2-octet, unsigned integer containing the length of time for which the receiving station is requested to inhibit data frame transmission. The field is transmitted most-significant octet first and least-significant octet second. The PAUSE quanta is defined in terms of 512 bit times. The PAUSE quanta value in the PAUSE frame has a range from 0 to 65535.

On reception of a PAUSE control frame on a particular link with non-zero PAUSE quanta, a link-specific PAUSE timer, which is initially zero is preset to a PAUSE quanta value and the transmission of packets on that particular link are paused, after the completion of the ongoing packet transmission. The PAUSE timers are continually decremented until zero. Once the counter value is zero or a PAUSE control frame on the paused link with a zero PAUSE quanta is received, the transmission is restarted on the particular link. However in the meantime, if another PAUSE control frame is received with new non-zero value, then the timer is preset to the new PAUSE quanta.

**Figure 132** below shows the link pause operation per link. XOFF indicates transmission is paused and XON means transmission is turned on.

Figure 132 Pause Timer State Machine

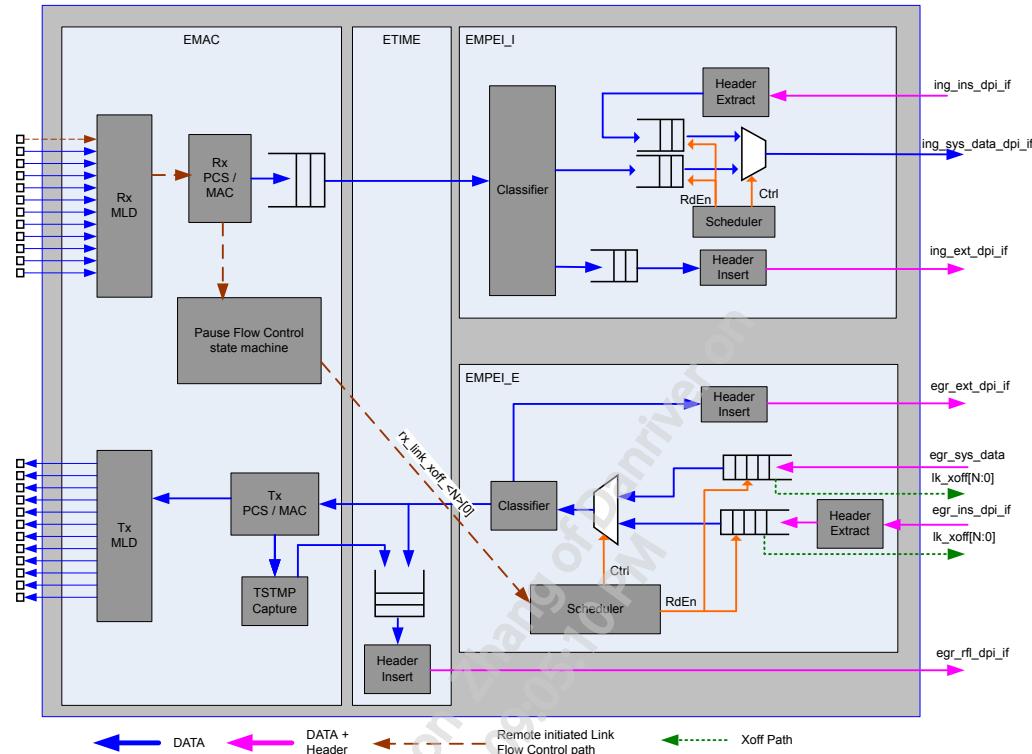


The link pause can be either initiated by remote transmitter, when the remote side finds its resources are being depleted or by the ENET, when it receives a back pressure signal from the CPB, indicating that CPB resources are being depleted.

#### 5.12.3.5.1.1 Link Pause by Remote Entity

When the remote side starts lacking buffer space, before the transmitted packet is dropped by the remote link receiver, the remote transmitter sends a PAUSE control frame. Figure 133 below shows the flow of the link-based PAUSE control frame.

Figure 133 Remote Link Pause Control Information Flow



When a link PAUSE frame is received on a particular link, the quanta time is extracted by the EMAC's Rx MAC. This PAUSE quanta registered and loaded into the PAUSE flow control state machine similar to the one shown in Figure 132 above. The PAUSE controller state machine waits for the transmitter to complete any ongoing frame transmission and then enters a state where it does not transmit any user frames from the transmit FIFO from the EMPEI and CPB for that particular link except MAC control frames. When the transmitter has reached its PAUSE state, the timer starts to decrement. When the timer reaches 0 or a resume command is received (a PAUSE frame with quanta value of 0), the transmitter resumes normal transmission of frames for that link. The processing and responding to a received link PAUSE control frame is automatically handled within the EMAC block without any intervention or support from other blocks.

When a link-based PAUSE control frame is received on a particular link, the frame is analyzed by the EMAC, the corresponding internal PAUSE quanta counters inside the EMAC are set with the quanta values extracted from the received link-based PAUSE frame and start decrementing. The EMAC provides a signal to indicate the status of the PAUSE state on per link basis that is being fed to the EMPEI. The EMPEI relays it to the central packet buffer (DCPB), which stops the traffic.

The reaction time to pause frames varies according to the link rate. Below, the reaction times (time from the reception of a PAUSE frame to traffic stop) and "skid times" (number of packets that can be transmitted after receiving a link PAUSE and before traffic stop) are listed. Note that the numbers here do not take into account the time to transmit any large packet. The numbers are meant to indicate the worst-case reaction time of the ENET hardware. In all cases, the dominant term in the equation is the time to empty the data in the pipeline between the DCPB and the Tx Line. The numbers below assume that the FIFOs and pipeline from the DCPB out to the line are completely full at the time of the pause frame reception.

**Table 66** Pause Reaction

Rate	max pause reaction time (after the reception of a pause, no new Start-of-Packet is transmitted after the reaction time below).	PAUSE skid (# of min-sized packets sent before traffic stop)
10 Gbps	1600 ns	26
40 Gbps	1500 ns	95
100 Gbps	1400 ns	225

**5.12.3.5.1.2** **Link Pause by Local Entity**

Whenever receive resources are depleted, the ENET can send a PAUSE frame on the link to stop the remote side from transmitting further data on the link except MAC control frames.

This is done based on the status of the external central buffer to which the received packets are sent. After transmitting a frame, the MAC samples a fill level status input and determines if a link PAUSE control frame should be immediately scheduled. The following scenarios at the time for PAUSE control frame transmission are possible:

- A transmit PAUSE is not in progress and the transmit XOFF bit is set. In this case, a new PAUSE control frame is sent with the programmed quanta time value
- A transmit PAUSE is already in progress for this link, but the transmit XOFF is reset. In this case, a PAUSE frame with a quanta time = 0 will be sent.
- A transmit PAUSE is already in progress for the link and the transmit XOFF bit is still set. In this case:
  - If the transmit quanta timer is between its maximum value and the threshold value, no new PAUSE update is needed, so transmission of next packet is resumed.
  - If the transmit quanta timer is below the threshold, a new PAUSE control frame is sent with the programmed quanta time value.

The following figure shows the flow for the generation of transmit PAUSE control frame.

The packets from a link are mapped or routed to one of the eight channels in the external packet buffer dedicated for a link. The status for each of these channels is indicated through 2 bits per channel. Thus when the resource reaches the near full state for any of the one or more channels associated with a link, the transmit XOFF is asserted.

**Receive Operation with Local Link Pause**

The following section describes the flow of packets on an ingress link configured for a link PAUSE operation:

1. Packets received at the link and after the PCS and MAC are written to the RxFIFO. In normal circumstances, assuming that the read operation is at a much higher rate compared to the link speed, the RxFIFO will not overflow or cross the upper threshold.
2. Packets are read from the MAC RxFIFO through the ETIME by the EMPEI block after being scheduled by a round-robin scheduler.
3. Packets are classified to determine type (PTP, Mgmt, etc.) and are written to the inline RxFIFO with sideband signals: link number (4-bit).
4. After some delay through the inline RxFIFO, the receive scheduler will schedule the packet to be written into the CPB.
5. Within the external packet buffer, the depth of indexed channel number queue is used to produce near full indication for all queues via a parallel bus. If the queue is completely full then the packet

will be tail-dropped. This full queue overflow condition should be averted by the link PAUSE mechanism.

6. A Near Full flag from the external packet buffer is checked for all queues/channels in parallel and converted to an almost full flag of a channel to the link.
7. Based on the near full flag status of a link, the Tx MAC is informed that it needs to send the link PAUSE control frame. At the time for the PAUSE control frame transmission, the following scenarios are possible:
  - a. A transmit PAUSE is not in progress and the transmit XOFF bit is set. In this case, a new PAUSE control frame is sent with the programmed quanta time value.
  - b. A transmit PAUSE is already in progress for this link, but the transmit XOFF is cleared. In this case, a PAUSE frame with quanta Time = 0 will be sent.
  - c. A transmit PAUSE is already in progress for the link and the transmit XOFF bit is still set. In this case:
    - i. If the transmit quanta timer is between its maximum value and the threshold value, no new PAUSE update is needed, transmission of the next application is resumed.
    - ii. If the transmit quanta timer is below the threshold, a new PAUSE control frame is sent with the programmed quanta time value.

#### 5.12.3.5.2 Firmware Generation of Link Pause

The ENET subsystem can also generate link PAUSE according to registers that are controllable by firmware. In this mode (selectable on a per-link basis), a link PAUSE XOFF duration and a link PAUSE XON duration can be programmed. When the link PAUSE XOFF timer is active, the Tx MAC sends link PAUSE frames thereby halting transmission at the remote end. When the link PAUSE XOFF timer expires, the Tx MAC sends a link PAUSE frame with the PAUSE quanta = 0, thereby re-enabling data transmission at the remote end. In this way, a periodic link PAUSE can be sent that effectively limits the overall rate of transmission of the remote entity, but that is not controlled by the local device's receive buffer fill level.

A more crude way of sending a link PAUSE is available through firmware-controlled registers. In this mode (enabled on a per-link basis), firmware can pause a link with a level-sensitive ON/OFF bit. Setting the bit causes the Tx MAC to send pause frames indefinitely, thereby disabling transmission on a link. Only when the ON/OFF bit is cleared will the Tx MAC sends a link PAUSE frame with PAUSE quanta = 0 allowing data transmission at the far end.

#### 5.12.3.6 Packet Classification

The Ethernet Packet Classifier (ECLASS) block performs the classification of Ethernet packets. The channelized packet classifier is capable of processing first 88 bytes of all Ethernet packets at the line rate, with channelized data arriving in a time-sliced manner. The packet classifier parses and matches any of the first 88-bytes of Ethernet frame to user programmable values configured by firmware. Based on the configured patterns, the packet classification engine identifies variety of packets and encapsulation formats as mentioned in below. The bytes prior to the DA field namely preamble, SFD or custom preamble are removed by the EMAC in the ingress path and by a Header Extraction block in the egress path. The ECLASS has enough classification resources to concurrently identify the following packet types:

- Ethernet Link-OAM
- Ethernet SyncE ESSM
- Ethernet LACP
- Ethernet Marker Protocol

- Management VLAN (with specific configurable TPID and VLANID which are not expected to be mixed with the other types of packets)
- PTP over Ethernet packet (without any Destination Address check)
- PTP over Ethernet packet (with a Destination Address check) (Unicast DA, not (Unicast DA), and Multicast DA all flagged)
- PTP over UDP over IPv4 over Ethernet (without any Destination Address check)
- PTP over UDP over IPv4 over Ethernet (with Destination Address check) (Unicast DA, not (Unicast DA), and Multicast DA all flagged)
- PTP over UDP over IPv6 over Ethernet (without any Destination Address Check)
- PTP over UDP over IPv6 over Ethernet (with Destination Address Check) (Unicast DA, not (Unicast DA), and Multicast DA all flagged)
- IPv4 ARP
- IPv6 NDP

In addition, the ECLASS has hardware to classify VLAN Tags (Inner VLAN Tag or Outer VLAN Tag or both) separately from the rest of the frame. A pre-classifier strips the VLAN header before feeding the rest of the packet to the 'general classifier'. In this way, the general classifier will see the packets as non-tagged packets, allowing the limited resources of the classifier to cover both tagged and non-tagged packets.

The ECLASS has a limited number of configurable filters to allow classification beyond the above-stated packet types. However, these configurable filters can only target the first 88 bytes of a packet, not beyond.

Each classification process uses various types of filters. As a result of classification, the packet classifier provides a unique output called "Classifier Tag" after a fixed latency. If packets did not match with any of the configured patterns, then a "No Match" signal is asserted.

The Packet Classifier output results are aligned with the SOP of the packet. The Classifier Tag is used by the packet distributor to decide where to route the received Ethernet packet. The Ethernet packet along with a given Tag can be routed to more than one destination, based on the configuration.

### 5.12.3.7 Management Port and Management Path Flows

The ENET subsystem, the Management port, and the packet cache (Internal CPU) communicate via the Management CPB (MCPB). The figures below shows the possible management insert and extract path flows for packets to and from the Management port (MGMT\_GE), the packet cache and the ENET subsystem.

Figure 134 ENET Management Paths to/from MGMT\_GE

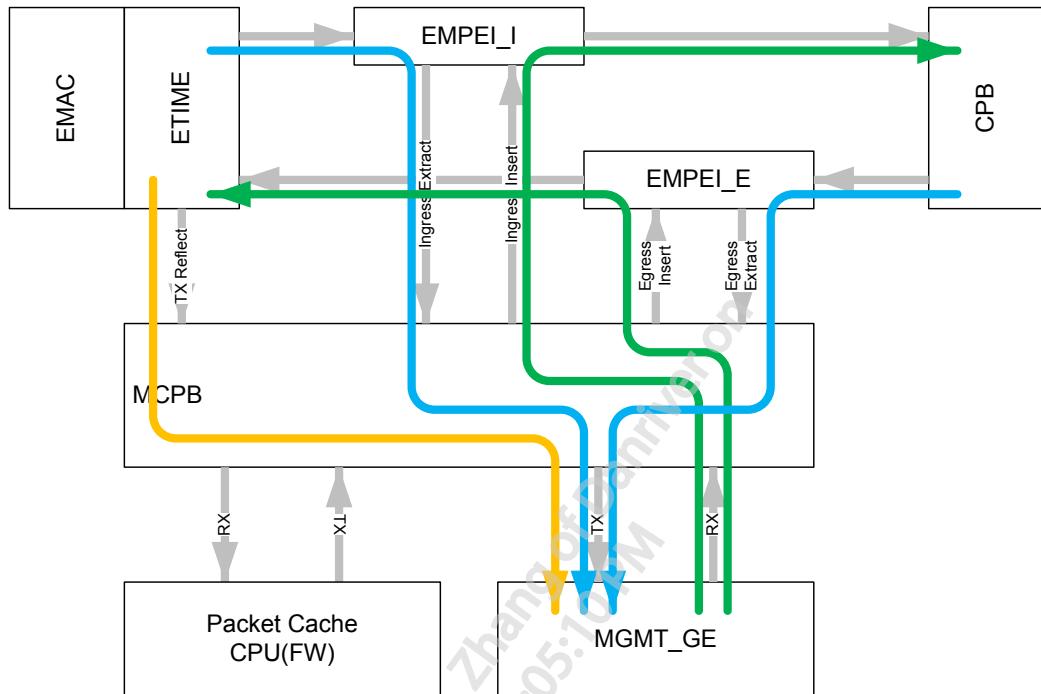


Figure 135 ENET Management Paths to/from CPU PKT\_CACHE

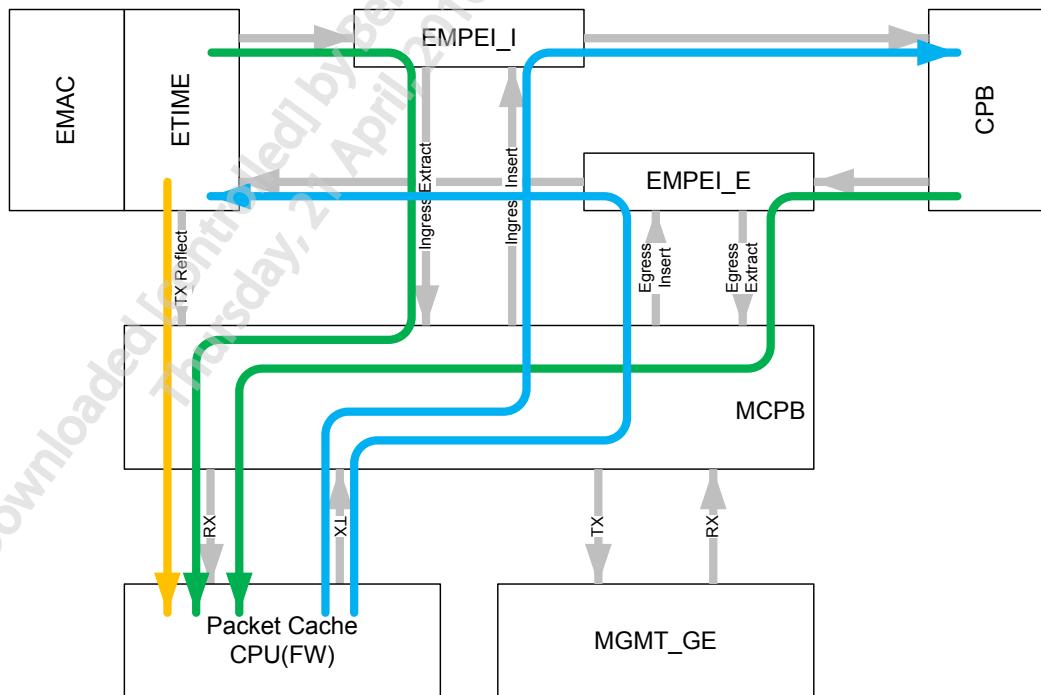
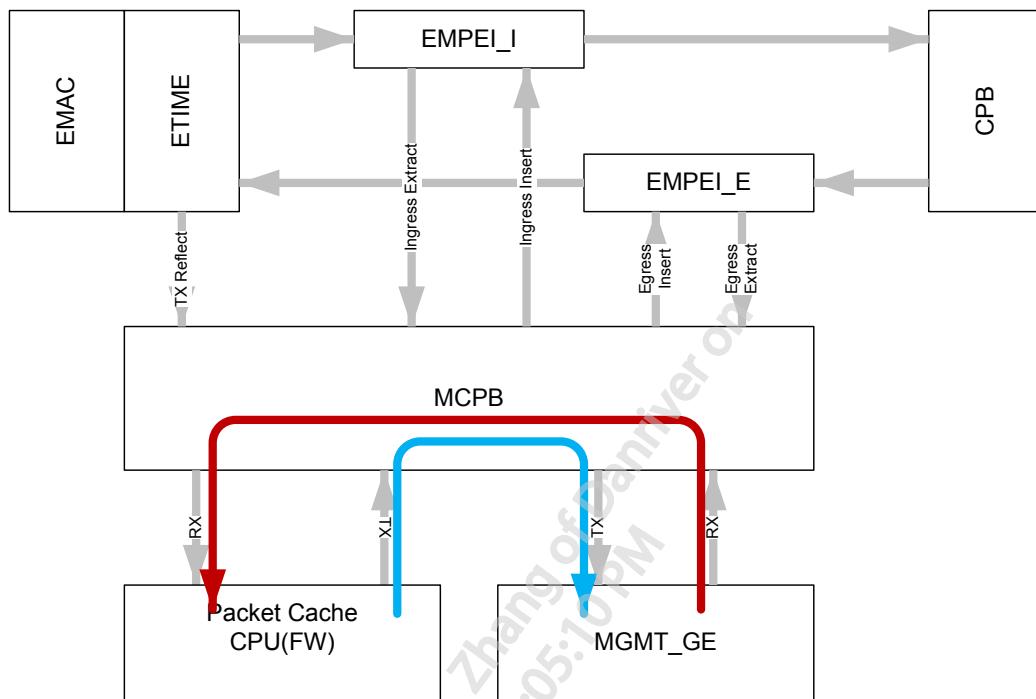


Figure 136 Management Paths between CPU PKT\_CACHE and MGMT\_GE



Although insert and extract connections to and from the MCPB exist from both line and system instances of the ENET, both cannot be supported simultaneously. Only the insert and extract paths for either the line side ENET or the system-side ENET can be enabled at any one time. This implies that any protocols requiring insertion and extraction, or requiring packet interaction with the CPU or external management port cannot be supported on both line and system sides concurrently. Such protocols include PTP, SyncE, LinkOAM, LACP, and others.

Insert and extract flows for each block are fed into or out from the MCPB switch. The MCPB provides a single Insert and Extract DPI port to/from each block. All packets inserted into the MCPB via the Insert Port, must provide a destination channel ID so that the MCPB can place the packet in the appropriate destination queue. The MCPB has a pool of queues that can be assigned to each block's insert interface and follows a strict scheduling as configured for that interface.

In the egress Extract path of the ENET, the destination of the extracted packet is configurable based on the packet type and the VLANs received. Possible destinations of extracted packets are:

- Management GE subsystem
- CPU subsystem

In the ingress insert path of the ENET, possible sources of inserted packets are as follows.

- Management GE subsystem
- CPU subsystem

In the egress Insert path of the ENET, inserted packets are dissected to remove a proprietary VLAN header carrying time-stamp control information and reflection control information. Packets are issued onto the transmit path and may be time-stamped (one step) or reflected with critical time stamp information (two step). Possible sources of inserted packets are:

- Management GE subsystem
- CPU subsystem

In the egress Reflection path, the destination of the reflected packet is configurable. Possible destinations are:

- Management GE subsystem
- CPU subsystem

The MCPB is designed to have enough buffering for a maximum packet size of 2048 bytes. Beyond that, the MCPB queues may overflow and the bandwidth may be compromised.

#### 5.12.3.7.1 Header Insertion and Extraction

Each packet to/from the management switch has an additional header (VLAN tag plus additional bytes) applied to it, forming an internal "augmented frame". The header carries critical timing information about the packet, control or identification information about the packets between the ENET and either the Management Port or the Internal CPU.

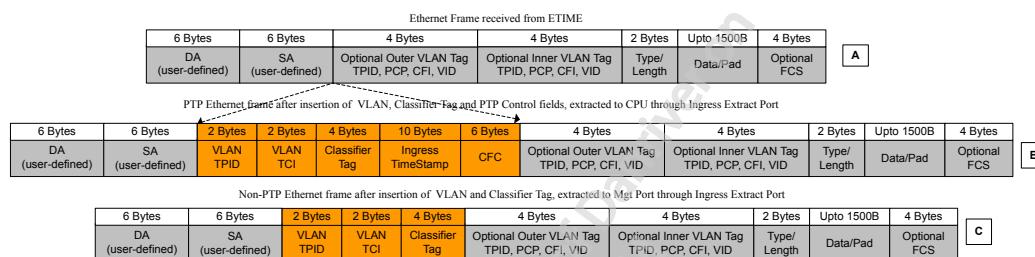
The Extraction Path (ingress extract, egress extract or reflection port) adds a proprietary VLAN Tag, time stamp information and packet classification information. The PMC Header format, via the TS\_INCLUDE control, will include or exclude timestamp information. The EMPEI Ingress TSB makes use of both PMC Header possibilities, while the EMPEI Egress TSB only issues packets without timestamp information. The ETIME TSB (reflection path) always includes the timestamp, but will always zero the classification tag. All PMC Header bit fields are described below.

1. 2-bytes VLAN TPID
  - Set to 0x8100.
2. 2-bytes VLAN TCI
  - Bits (15:13) VLAN State
    - 0 = No VLAN Tag in received Ethernet Packet
    - 1 = One VLAN Tag in received Ethernet Packet
    - 2 = Double VLAN Tag in received Ethernet Packet
    - 3-7 = Reserved
  - Bit (12) Reserved = 0
  - Bits (11:8) Source Link ID
  - Bits (7:4) Source ID
    - 0 = Line Side Ingress (EMPEI Ingress TSB)
    - 1 = Line Side Egress (EMPEI Egress TSB)
    - 2 = Line Side Reflect (ETIME TSB)
    - 4 = System Side Ingress (EMPEI Ingress TSB)
    - 5 = System Side Egress (EMPEI Egress TSB)
    - 6 = System Side Reflect (ETIME TSB)
  - Bits (3:1) Page Number (EMPEI Ingress TSB)
  - Bits (3:1) Reserved (EMPEI Egress TSB)
  - Bits (3:1) Sequence ID (ETIME TSB)
  - Bit (0) Timestamp Included
    - 0 = ToD and CFC not present in forwarded packet
    - 1 = ToD and CFC present in forwarded packet
3. 4-bytes Classifier Tag

4. 10-bytes Time of Day Timestamp (Inserted only if VLAN\_TCI[0] is 1)
5. 6-bytes Correction Field Counter Timestamp (Inserted only if VLAN\_TCI[0] is 1).

The following [Figure 137](#), depicts the input frame format, Frame A, and possible packet output formats from the Header Insertion block. Frame B is output if the Timestamp Included bit is set in the VLAN TCI bit 0, while Frame C is output when that bit is 0. All frames are depicted with Inner and Outer VLAN tags as well as FCS, although these fields may or may not be present in the packet. The VLAN state information, provided in the PMC Header, provides the processor a quick check to identify if, and how many, VLAN tags are present.

**Figure 137** Frame format through Header Insertion Block



The Insert Path (Header Extraction) removes the proprietary VLAN Tag from the packet.

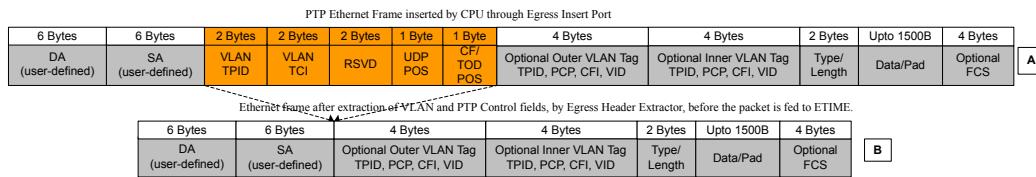
For inserted packets, the format differs for ingress insert and egress insert. This is because for the egress insert, instructions on what to do with the packet are included (timestamp instructions, FCS calculation instructions). Meanwhile packets inserted into the ingress datapath (towards the Central Packet Buffer) have no such instructions.

In the Egress direction, the Header Extraction block extracts the following PMC Header information from the received packet:

1. 2-bytes VLAN TPID
2. 2-bytes VLAN TCI
  - Bits (15:14) VLAN State
    - 0 = No VLAN Tag in received Ethernet Packet
    - 1 = One VLAN Tag in received Ethernet Packet
    - 2 = Double VLAN Tag in received Ethernet Packet
  - Bits (13:8) Not Used, set to 0
  - Bit (7): REFLECT\_EN
  - Bit (6): TOD\_UPDATE\_EN
  - Bit (5): CF\_UPDATE\_EN
  - Bit (4): UDP\_CHKSUM\_UPDATE\_EN
  - Bit (3): FCS\_CALC\_EN
  - Bits (2:0): Sequence ID
3. 2-bytes Reserved
4. 1-byte UDP Position
5. 1-byte Correction Field (CF) or TOD Position

The below [Figure 138](#), shows the frame format through Egress header extraction block, where Frame A depicts the received frame, while Frame B depicts the outgoing packet with the PMC Header removed. The frame may be received with, or without, optional VLAN Tags and FCS.

Figure 138 Frame Format through Egress Header Extraction block

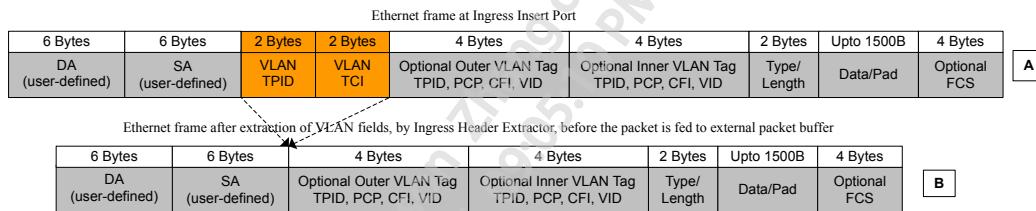


Note: The Egress Header extraction block, after the extraction of above mentioned control header fields, will not recompute, nor will it append an FCS field to the outgoing packet.

In the Ingress direction, the Header Extraction block extracts the following PMC Header information from the received packet:

1. 2-bytes VLAN TPID
2. 2-bytes VLAN TCI .
  - Bits (15:0) Reserved

Figure 139 Frame format through Ingress Header Extraction Block



In the above [Figure 139](#), Frame format -A shows the Ethernet Packet inserted through Ingress Insert port, whereas Frame format-B shows the output of Ingress Header Extraction block (VLAN Tag extracted) before the packet is forwarded to DCPB.

### 5.12.3.7.2 Management Port (MGMT\_GE)

A single serial link with an SFP SGMII is dedicated as a Management Ethernet interface. This interface supports 1000 Mbps full-duplex Ethernet with PAUSE frame support. The Ethernet MGMT\_GE is a separate block outside of the ENET that is connected via a external packet buffer elsewhere in the device.

The MGMT\_GE is reserved for insertion / extraction of "management packets" that may include, but are not limited to LACP, Management VLAN and PTP Timing packets. These packets are optionally extracted from each of the 10GE/40GE/100GE ports and reported on this Management Ethernet port.

All packets that pass through the MGMT\_GE must contain an extra VLAN tag to identify their source (for transmit packets) or destination (for receive packets). All source-destination pairs in the MCPB are identifiable with a unique queue number. Upon receiving a packet from the MCPB, the MGMT\_GE performs a mapping between queue number and VLAN TCID. In this way, an external CPU, by performing a lookup on the received VLAN TCID, can determine the source of the packet (CPU, ENET ingress extract, ENET egress extract, as well as link number).

In the same way, all packets from an external CPU must contain an extra VLAN tag from which the MGMT\_GE can determine the packet's intended destination. Based on the VLAN TCID of the incoming packet, the MGMT\_GE sends the packet to the proper MCPB queue, and ultimately to the packets in-

tended destination. Destinations include the CPU (PKT cache), ENET ingress insert, and ENET egress insert.

Packets to be transmitted by the 10GE/40GE/100GE ports are inserted via the Management port. The insertion and extraction of the augmented frame is performed by the various blocks within the EMPEI.

Apart from the data rate of the link, other functional differences exist on the Management port. Specifically, no arrival and departure time stamping (for PTP) is available on the Management port. This port is meant simply as a pipe between the local device and an external processor.

### 5.12.3.8 System Queues and Data Ports

The system interface to the ENET provides a number of channelized, time-sliced data ports that are used to access the insert and extract queues implemented within the EMPEI\_I and EMPEI\_E blocks and well as the Reflection port of the ETIME. The various ports are:

- Ingress Extract DPI: EMPEI\_I to the MCPB
- Ingress Insert DPI: MCPB to the EMPEI\_I
- Ingress System Data DPI: EMPEI\_I to the Data CPB
- Egress Extract DPI: EMPEI\_E to the MCPB
- Egress Insert DPI: MCPB to the EMPEI\_E
- Egress System Data DPI: Data CPB to the EMPEI\_E
- Egress Reflection: DPI

Each system port conveys packet information to/from the physical links operating in the system. One physical link may be mappable to multiple Data CPB channels or multiple MCPB channels, while the inverse is also true, multiple Data CPB channels may be mapped to one physical link and multiple MCPB channels may be mapped to one physical link. [Table 67](#) below identifies the number of channels per physical link.

**Table 67 Insert / Extract Queue to Port Mapping**

Port	Direction	Number of Channels	Queues Represented
ing_sys_data_dpi_if	Output	1	Ingress System Data Queue
ing_ext_dpi_if	Output	2	Ingress Extract Queue. CPU and Management Port are possible destinations.
ing_ins_dpi_if	Input	2	Ingress Insert Queue. CPU and Management Port are sources of inserted packets.
egr_sys_data_dpi_if	Input	1	Egress System Data Queue
egr_ins_dpi_if	Input	2	Egress Insert Queue. CPU and Management Port are sources of inserted packets.
egr_ext_dpi_if	Output	2	Egress Extract Queue. CPU and Management Port are possible destinations.
egr_ref_dpi_if	Output	2	Egress Reflection Queue. CPU and Management Port are possible destinations..

For flows into the ENET, Insert flow and Egress System Data flow, scheduling of packets from the various channels is accomplished by the respective CPB. For example, in the Egress System Data Flow, eight possible channels are scheduled onto a single DPI. The Data CPB schedules based on channel configuration. As scheduling is done at the source, the ENET queues accepting this scheduled data are extremely shallow, and the ENET is responsible for combining the inserted packets with the system

data packets. When a path is stalled, the queue quickly fills, indirectly stalling the source CPB. This ensures that the CPBs schedule data per channel configuration.

### 5.12.3.9 Performance Monitoring and Statistics

The ENET provides an independent set of statistic counters for each of the available physical links. Events in the channelized MAC core are captured on both receive and transmit which are in turn conveyed to a channelized statistics counter block.

The following statistics classes are supported:

- IEEE 802.3 clause 30: Basic, Mandatory, Recommended packages.
- RFC 2665 (MIB), RFC 3635 (updated for SNMP)
- RFC 2819 (RMON)

As seen below, there is a much richer set of performance monitoring counters available in the Ingress (Receive) direction than there is in the Egress (Transmit) direction. In some cases, performing a transmit-to-receive loopback in order to perform performance monitoring at the Receive MAC may be desirable. Loopbacks are available for delivering transmit streams back to the RX MAC:

- EMAC Transmit to EMAC Receive (loopback from the ENET MAC-path)
- EPMM Transmit to EMAC Receive (loopback from the ENET transparent path)
- Device top-level loopbacks can also be used to deliver transmit Ethernet streams to any ENET Rx port, potentially on a separate instance of the ENET subsystem.

#### 5.12.3.9.1 Ingress Receive Statistics

Table 68 RX MAC Stats: IEEE 802.3 oMacEntity and oPauseEntity Managed Object

Object	Description
aAlignmentErrors	Frame received with a start-of-frame delimiter (SFD) mismatch.
aMACControlFramesReceived	Valid control frame received.
aPAUSEMACCtrlFramesReceived	Valid pause frame received.
aFrameTooLongErrors	Frame received exceeded the maximum length programmed in register FRM_LENGTH.
aInRangeLengthErrors	A count of frames with a length/type field value between 'd4 6 (VLAN: 'd42) and less than 'd1536 (0x600), that does not match the number of payload data octets received.
aFramesReceivedOK	Frame received without error (including pause frames).
aFrameCheckSequenceErrors	CRC-32 Error is detected but the frame is otherwise of correct length. According to RFC2665, fragment and undersize packets are not counted as CRC error packets.
VLANReceivedOK	VLAN frame received without error.

Table 69 RX MAC Stats: IETF MIB (MIB-II) Objects

Object	Description
ifInOctets	All octets received except preamble (i.e. Header, Payload, Padding and FCS) for all valid frames and valid pause frames received.

Object	Description
ifInErrors	Number of frames received with error: <ul style="list-style-type: none"> <li>• FIFO Overflow Error</li> <li>• FCS Error (except for fragment and undersize packets)</li> <li>• Frame Too Long Error</li> <li>• SFD mismatch (40 and 100G only)</li> <li>• Dedicated Error Code (0xFE, not a code error) was received</li> </ul>
ifInUcastPkts	Incremented with each valid frame received on the Receive FIFO interface and bit 0 of the destination address was '0'.
ifInMulticastPkts	Incremented with each valid frame received on the Receive FIFO interface and bit 0 of the destination address was '1' but not the broadcast address (all bits set to '1'). Pause frames are not counted.
ifInBroadcastPkts	Incremented with each valid frame received on the Receive FIFO interface and all bits of the destination address were set to '1'.

Table 70 RX MAC Stats: IETF RMON MIB Objects

Object	Description
etherStatsDropEvents	Counts the number of dropped packets due to internal errors of the MAC Client. Occurs when a Receive FIFO overflow condition persists. In 10G only, alignment (SFD) errors are also counted as dropped packets.
etherStatsOctets	Total number of octets received. Good and bad packets.
etherStatsPkts	Total number of packets received. Good and bad packets.
etherStatsUndersizePkts	Total number of packets that were less than 64 octets long with a good FCS. Note: Undersized packets are not delivered to the FIFO interface.
etherStatsPkts64Octets	Incremented when a packet of 64 octets length is received (good and bad frames are counted).
etherStatsPkts65to127Octets	Frames (good and bad) with 65 to 127 octets.
etherStatsPkts128to255Octets	Frames (good and bad) with 128 to 255 octets.
etherStatsPkts256to511Octets	Frames (good and bad) with 256 to 511 octets.
etherStatsPkts512to1023Octets	Frames (good and bad) with 512 to 1023 octets.
etherStatsPkts1024to1518Octets	Frames (good and bad) with 1024 to 1518 octets.
etherStatsPkts1519toMaxOctets	Proprietary RMON extension counter that counts the number of frames with 1519 bytes to the maximum length programmed in register FRM_LENGTH.
etherStatsOversizePkts	Total number of packets longer than the valid maximum length programmed in register FRM_LENGTH (excluding framing bits, but including FCS octets), and with a good Frame Check Sequence.
etherStatsJabbers	Total number of packets longer than the valid maximum length programmed in register FRM_LENGTH (excluding framing bits, but including FCS octets), and with a bad Frame Check Sequence.

Object	Description
etherStatsFragments	Total number of packets that were less than 64 octets long with a wrong FCS.  Note: Fragments are not delivered to the FIFO interface.

### 5.12.3.9.2 Egress Transmit Statistics

Table 71 TX MAC Stats: IEEE 802.3 oMacEntity and oPauseEntity Managed Object

Object	Description
aPAUSEMACCtrlFramesTransmitted	Valid pause frame transmitted.  Note: This object is not implemented.
aFramesTransmittedOK	Frame transmitted without error (including pause frames).
VLANTransmittedOK	VLAN frame transmitted without error.

Table 72 TX MAC Stats: IETF MIB (MIB-II) Objects

Object	Description
ifOutOctets	All octets transmitted except preamble (i.e. Header, Payload, Padding and FCS) for all valid frames and valid pause frames transmitted.
ifOutErrors	Number of frames transmitted with error: <ul style="list-style-type: none"> <li>• FIFO Overflow Error</li> <li>• FIFO Underflow Error</li> <li>• User application defined error (ff_tx_err asserted together with ff_tx_eop)</li> </ul>
ifOutUcastPkts	Incremented with each frame written to the FIFO interface and bit 0 of the destination address set to '0'.
ifOutMulticastPkts	Incremented with each frame written to the FIFO interface and bit 0 of the destination address set to '1' but not the broadcast address (all bits set to '1').
ifOutBroadcastPkts	Incremented with each frame written to the FIFO interface and all bits of the destination address set to '1'.

### 5.12.3.10 ENET PTP Support

The ENET subsystem supports PTP (Precision Timing Protocol) defined packet-based timing recovery and distribution using the mechanisms described in the IEEE 1588v2 standard. The PTP protocol allows clock frequency and TOD information to be recovered from a timing master. The ENET subsystem provides the hardware required to support boundary clock, peer-to-peer and end-to-end transparent clock modes, as defined in the IEEE 1588v2 standard. Some of these hardware functions include: time stamp generation, on-the-fly correction field update value calculation and support of a two-step clock mode by storing the two-step packet and returning it to the central PTP processing engine with a header containing the associated egress time stamp. The timing packets and their associated time stamps can be routed to either the on-chip CPU or an external timing processor. The ENET also supports an external PTP processor via the external management processor interface, thus the processing CPU can be either on the chip or external to the chip.

The ingress path has hardware logic that time stamps all incoming packets and corrects for any link delay between the device pins and the time stamping point. This hardware assist in capturing the event time stamp is necessary for the CPU to get accurate values.

Similarly the egress path time stamps outgoing packets when requested and returns the time stamp to upstream logic or inserts the time stamp into the packet depending on the requested mode of operation. The returned (reflected) time stamp value is used to support the PTP's 2-step clock mode operation. The inserted time stamp is used to support one step operation where ENET is required to insert on-the-fly an updated time stamp or correction field value. The ENET also supports FCS field insertion and UDP checksum update field correction based on need.

This PTP clock is generated by a DCSU (Digital Clock Synthesis Unit) and is tuned so its 125 MHz is synchronized to the selected PTP master clock. This PTP clock is provided on an output port, which can then be routed to a device pin. The PTP master clock has several possible origins:

- An input port clocked by any physical layer recovered clock inside the device
- An input port clocked by the local system clock

If a physical clock inside the device is selected as the PTP clock's master, then only synchronization (frequency synchronization, not TOD synchronization) can be achieved.

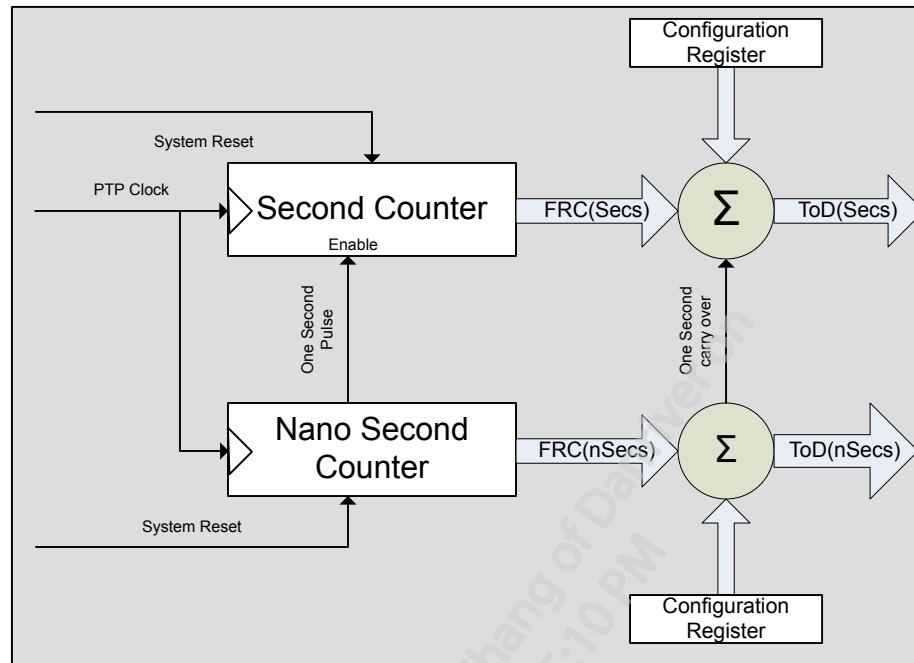
The ENET supports an ingress and egress TOD or IRIG-B 800x interface that can be used to transfer TOD and a clock frequency across a backplane to IRIG slaves. The ENET can be an IRIG slave receiving a desired TOD from an external IRIG master, act as an IRIG master sending a desired time-of-day (TOD) to another IRIG slave, or both simultaneously. The IRIGI input interface performs the slave portion of the duties and is responsible for receiving the current TOD from a timing master and making this information available to the timing processor through readable registers. The IRIGO output interface performs the master portion of the duties and is responsible for forwarding the local TOD to an IRIG slave. The IRIGO and IRIGI blocks are expected to operate at 125 MHz.

#### 5.12.3.10.1 TOD and Correction Field Counter

The ENET is designed to fully support PTP. In PTP, the protocol time stamp is defined as an 80-bit field with the upper 48 bits representing the integer seconds and the lower 32 bits representing the nano-seconds. A local FRC (Free-Running Counter) and a programmable offset are used to produce a TOD value that is used for time stamping.

There are two parts of the counter block. The first half is a counter that free-runs from 0 after reset is de asserted. This free-running counter is expected to be used for long-term frequency matching. The other is a TOD counter that is calculated by a programmable offset to the free running count value by the timing processor. The desired TOD is determined through a PTP or IRIG mechanism.

Figure 140 Generation of TOD from the FRC



This free-running counter is configured to increment by a configurable amount with each clock cycle, depending on its mode of operation and the frequency of its clock (normally 125 MHz for Ethernet based systems).

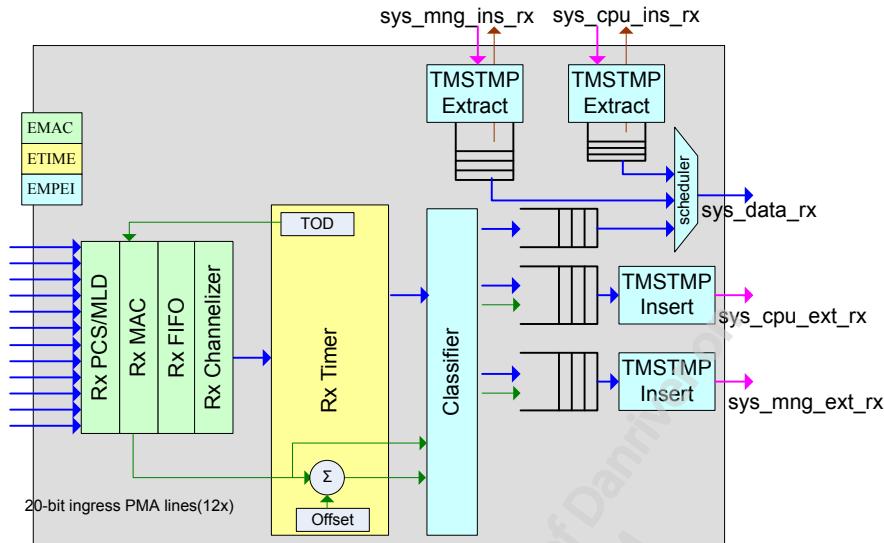
In addition to the TOD value, a 48-bit Correction Field Counter (CFC) is provided. This value can be used to simplify the arithmetic for correction field (CF) updates. Using the correction field counter, the final correction field (CF) value is calculated using the following technique:  $CF(\text{outgoing}) = CF(\text{incoming}) - CFC(\text{arrival}) + CFC(\text{departure})$ . The arrival CFC is subtracted by the internal or external timing processor and the departure CFC is added on the fly at egress time for one-step mechanisms and by the timing processor in 2-step mechanisms.

In order to support multiple links with varying latencies, link-specific receive and transmit offset registers are available to generate the specific TOD and correction field counter values that will be used from the free running count values.

#### 5.12.3.10.2 Timestamping on the Receive Path

All incoming packets have their arrival time stamps recorded by the EMAC, but only messages identified as PTP will include that time stamp in an internal header prepended to the packet when the packet is extracted for processing by the local or external CPU. The EMAC will record the TOD and CFC values as it detects the SOP on the ingress path using ENET provided running TOD and correction field values. The discrepancy between the time stamp sampling and the arrival of the packet on the line is compensated for by the ENET using link-specific offset registers. Although only the TOD values are forwarded to the PTP processor, knowing the programmed offset values, the processor can regenerate the FRC values. [Figure 141](#) below shows the data path during capture of time stamp during ingress of packet.

Figure 141 Time Stamping on the Receive Path



### 5.12.3.10.3 Time Stamping on the Transmit Path

#### 5.12.3.10.3.1 Transmit: One-step Clock

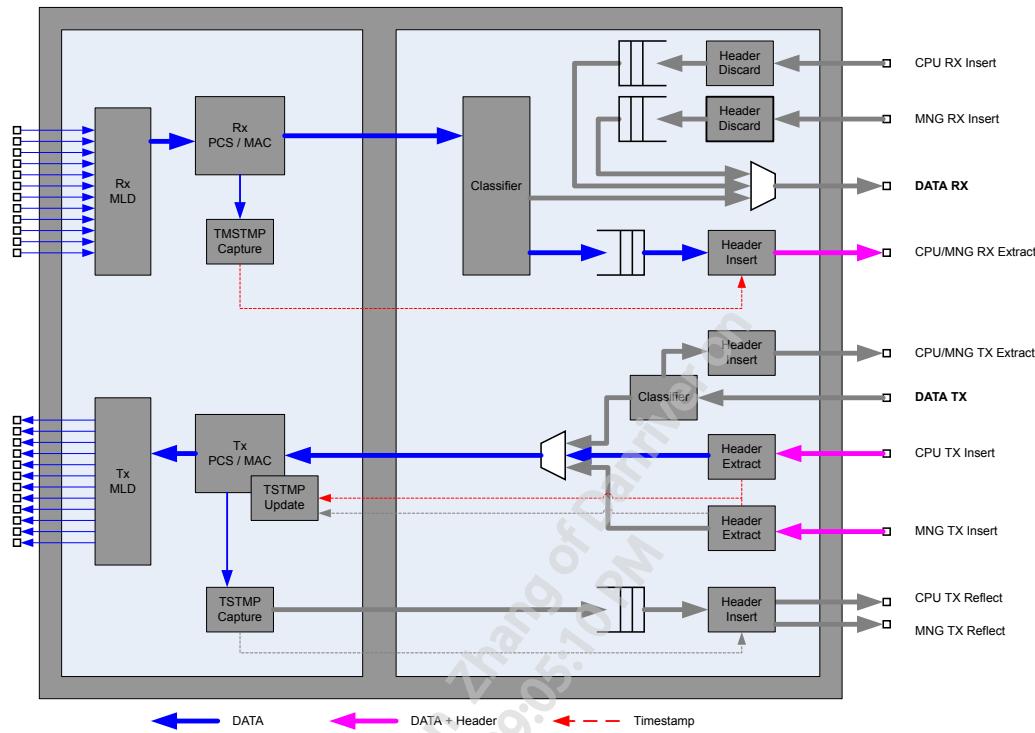
The ENET supports both one-step and two-step timing, as defined in the IEEE 1588 PTP protocol. In the one-step clock, the MAC inserts the provided time stamp or updated correction field information and updates the UDP checksum update field, if required, into the associated PTP event message thereby reducing the involvement of firmware in the process. The ENET classifier identifies the tasks required for each PTP event packet. The tasks include whether or not to perform a one-step TOD or CF update, the position of field to be updated, and whether or not to reflect the packet with a time stamp. The MAC will also insert an FCS (taking into account the modified PTP fields) if requested.

The ETIME block is responsible for extracting the correction field information (in a one-step mechanism) for each PTP event packets while maintaining an accurate correction field count value on a per link basis until the correction field is reflected or inserted. This functionality is required because due to the segmented nature of the device, multiple independent links can be time stamped simultaneously.

Note that Delay\_Req messages require a two-step mechanism irrespective of whether Sync messages are processed in one-step or two-step fashion. The departure time of the Delay\_Req message must be recorded and sent to the PTP processing engine.

Figure 142 below shows the block diagram for the transmit one-step clock mode data path.

Figure 142 Data Path for PTP in One-step Mode



PTP packet types that require timing capture / update are:

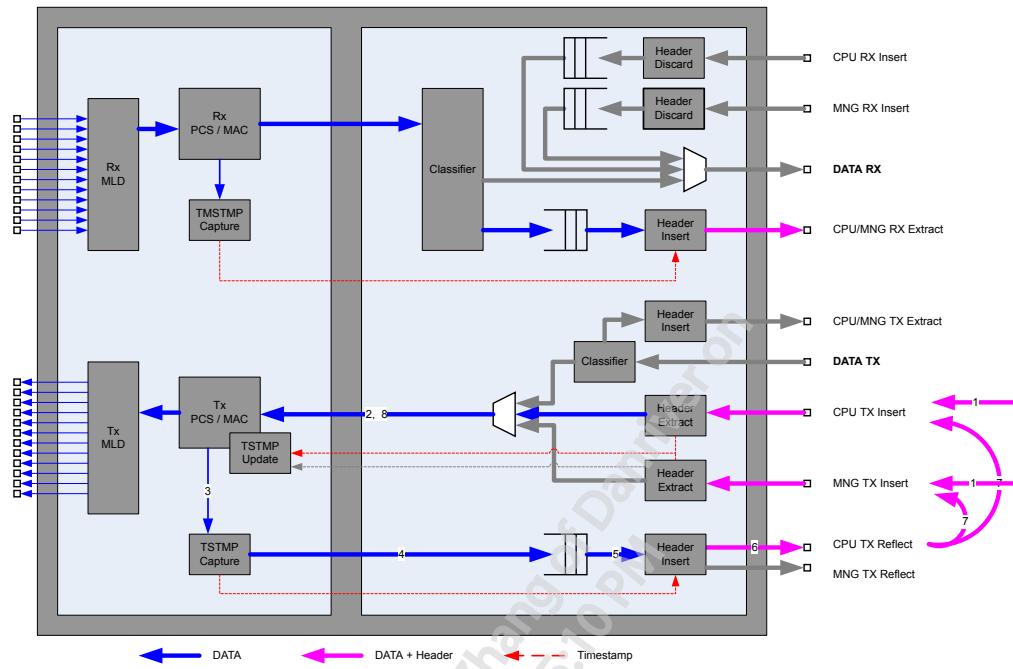
- Sync
- Delay\_Req
- Pdelay\_Req
- Pdelay\_Resp

#### 5.12.3.10.3.2 Transmit: Two-step Clock

For two-step processing, packets scheduled for transmit are flagged as requiring timing capture (PTP Sync, Delay\_Req, Pdelay\_Req and Pdelay\_Resp packets). When a packet is identified as one scheduled for a two-step mechanism, a copy of it is stored in the ETIME while awaiting a returned time stamp. As these packets pass through the channelized MAC (EMAC) transmit data path, the captured TOD and CFC is recorded and a copy of the outgoing packet is reflected back to the ETIME, associated with the stored packet, and sent to the internal or external timing processor for further processing. The time stamp is prepended to the packet in an inserted header prior to transmission.

Figure 143 below shows the block diagram for the transmit two-step clock mode data path:

Figure 143 Data Path for PTP in Two-step Mode



PTP event packet types that may require two-step processing include:

- Sync
- Delay\_Req
- Pdelay\_Req
- Pdelay\_Resp

Note that Delay\_Req messages require a two-step mechanism irrespective of whether Sync messages are processed in one-step or two-step fashion. The departure time of the Delay\_Req message must be recorded and sent to the PTP processing engine.

The PTP follow-up packet types generated are:

- Follow\_Up
- Pdelay\_Resp\_Follow\_Up

#### 5.12.3.10.4 TOD Offset and Frequency Correction

The device local TOD and Frequency needs to be in sync with the Clock Master. The device can achieve this synchronization in different modes:

- When PTP algorithms run on an internal timing processor core, that is when the internal processor terminates the PTP messages (boundary clock) or performs correction field updates (transparent clock), the internal processor has all the information necessary to adjust the TOD or to correct the frequency of the local clock.
- When PTP algorithms run on an external processor, the PTP messages are terminated by the external processor (boundary clock) or the correction field updates are performed by that external processor (transparent clock). In such a cases, the local processor can still access the necessary information for TOD offset updates and frequency correction. The local processor can snoop the

PTP packets destined for the external processor, thereby extracting all the information relevant to the TOD offset calculation. Alternatively, the external and internal processors can communicate directly, so that the internal processor has access to all the relevant information to adjust the TOD or to adjust the local clock.

#### 5.12.3.10.5 TC PTP One Step Requirement

The Precision Timing Protocol (PTP) enables precise synchronization of clock in a packet-based communication system. Through this clock synchronization both Time-Of-Day and frequency alignment to a master reference can be achieved. Annex-F of IEEE Std 1588™-2008 specifies the Transport of PTP over IEEE 802.3 /Ethernet. The first octet of the PTP message occupies the first octet of the MAC Client Data field.

Table 73 shows the Frame format of PTP over Ethernet.

Table 73 PTP Over Ethernet Frame Format

Octets	Field	Value	Type
6	Destination Address	0x011B 1900 0000 or Uni-cast DA (Sync, Delay_Req, General (other than Pdelay_Resp_Follow_Up)) 0x0180 C200 000E or Uni-cast P2P DA (PDelay_Req, PDelay_Resp, Pdelay_Resp_Follow_Up)	Ethernet Header
6	Source Address	-	
2	Length/Type	0x88F7	
1	Transport specific, message Type	Transport specific = 000X	PTP Message
1	Reserved, versionPTP	0x2 (optional) alternate values possible	
2	messageLength	-	
1	domainNumber	-	
1	reserved	-	
2	flagField	-	
8	correctionField	-	
4	reserved	Field used to store arrival time of packet	
10	sourcePortIdentity	-	
2	sequenceId	-	
1	controlField	-	
1	logMessageInterval	-	
10	originTimestamp	-	

A mechanism to transfer timing information between the ingress port and the egress port of a transparent clock has not been standardized in IEEE 1588. The PMC device supports a mode where the reserved field is used to record the arrival time of packets for interfacing with non-PMC devices. F/W receives and generates packets with the arrival timestamp in the RESERVED field. It is assumed that the PMC and the non-PMC device are aligned using only a 1 pulse-per-second input signal so their frequencies and sub-second timestamp counter values are the same.

## System Requirements

1. The non-PMC timestamp format is 30-bits and only counts up to 999,999,999 ns.
2. The maximum residence time of a PTP packet within the transparent clock must be significantly less than 1 second.
3. Within each PMC device, a sub-second timestamp counter (TSTMP\_CNTR) must be aligned to the other devices in the system by way of a 1PPS event signal generated by an external device and provided to the PMC device using the ONE\_PPS\_I input. Also note the TOD input interfaces are not used for this application. Because only the sub-second timestamp counter value needs to be aligned, the 1 PPS input signal is sufficient and the clock and data signals of the TOD input interface are not necessary. Because of this simplification, the TOD clock and data signals would not be used for this application.
4. The integer second timestamp counter values within each PMC device free-run. The value is used in intermediate calculations, but will eventually be subtracted from itself so the actual value is irrelevant.

## Firmware translation procedures:

From the 30-bit timestamp format to the PMC timestamp format:

1. FW receives the PTP packet with the 30-bit nanoseconds arrival time in the RESERVED field
2. FW polls its TOD TSTMP\_CNTR to see its current time (seconds and nanoseconds).
3. Knowing that the PTP packet should have arrived no more than N milliseconds ago (where  $N \ll 1000$ ), FW will pick the integer second value that best supports this behavior. The 80-bit TOD value will consist of this integer second value, "00", and the 30-bit nanosecond value from the RESERVED field of the PTP packet. This resulting TOD value corresponds to the TOD arrival time in the timing domain of this PMC device.
4. FW transforms this new 80-bit TOD arrival timestamp into a 48-bit correction\_field arrival timestamp format by doing a mod-48 function on the integer seconds value of the TOD timestamp and adding the result to the nanoseconds value from this TOD timestamp. This now corresponds to the 48-bit nanoseconds arrival time, in the timing domain of this PMC device.
5. The 48-bit correction\_field arrival timestamp is subtracted from the correction\_field of the PTP packet. Now this PTP packet is of the format that the PMC device hardware expects.
6. FW sends the modified PTP packet to the PMC device's egress hardware, where the 48-bit CFC departure time will be added to the correctionField correctly reflecting the packets residence time.

The translation procedure from the PMC 80-bit TOD timestamp format to the 30-bit timestamp format is as follows:

1. FW receives the PTP packet along with its 80-bit TOD arrival timestamp
2. The least significant 30-bits of this arrival timestamp are written into the RESERVED field of the packet. Bits [31:30] of the RESERVED field are set to 'b00'.
3. FW puts the modified PTP packet back into the datapath for transparent clock operation. With the alignment to 1PPS, and because the PTP packet must have arrived much less than 1 second ago, it is not necessary to consider rollover problems. There is never any ambiguity finding the correct integer second as shown in the examples below.

Example 1: timestamp in RESERVED field = 999,999,900ns PMC current TOD = 10 sec + 100ns Integer second options:

- If the packet arrived at TOD = 8 sec + 999,999,900ns, then the residence time up to now is 1.000 000 200 seconds, which is too big so this cannot be true.

- If the packet arrived at  $TOD = 9 \text{ sec} + 999,999,900\text{ns}$ , then the residence time up to now is approximately  $1,000,000,100\text{ns} - 999,999,900\text{ns} = 200\text{ns}$ . This is the only possible answer.
- If the packet arrived at  $TOD = 10 \text{ sec} + 999,999,900\text{ns}$ , then the residence time up to now is negative, so this cannot be true.

Example 2: timestamp in RESERVED field = 999,999,900ns PMC current TOD = 9 sec + 999,999,999ns Integer second options:

- If the packet arrived at  $TOD = 8 \text{ sec} + 999,999,900\text{ns}$ , then the residence time up to now is 1.000 099 seconds, which is too big, so this cannot be true.
- If the packet arrived at  $TOD = 9 \text{ sec} + 999,999,900\text{ns}$ , then the residence time up to now is approximately  $999,999,999\text{ns} - 999,999,900\text{ns} = 99\text{ns}$ . This is the only possible answer.
- If the packet arrived at  $TOD = 10 \text{ sec} + 999,999,900\text{ns}$ , then the residence time up to now is negative, so this cannot be true.

Example 3: timestamp in RESERVED field = 999,999ns PMC current TOD = 10 sec + 2,000,000ns Integer second options:

- If the packet arrived at  $TOD = 9 \text{ sec} + 000,999,999\text{ns}$ , then the residence time up to now is 1.001 000 seconds, which is too big, so this cannot be true.
- If the packet arrived at  $TOD = 10 \text{ sec} + 000,999,999\text{ns}$ , then the residence time up to now is approximately  $2,000,000\text{ns} - 999,999\text{ns} = 1,000,001\text{ns}$ . This is the only possible answer.
- If the packet arrived at  $TOD = 11 \text{ sec} + 000,999,999\text{ns}$ , then the residence time up to now is negative, so this cannot be true.

Note that if the max residence time limit was allowed to approach 1 second, then there would be ambiguity in choosing between the first and second integer second options in the above examples because of the timing uncertainties inherent in FW operations.

### 5.12.3.11 Timing: SyncE Support

Like traditional SONET/SDH links, SyncE links carry their data with a stratum traceable clock and with tight jitter and wander constraints. The quality of each SyncE link is communicated through Ethernet ESMC messages as defined in G.8264. The Ethernet subsystem extracts these Ethernet SSM messages and allows the application to select which clock to use as the timing reference for its transmit data streams. The ENET's classifier is responsible for identifying the ESMC messages.

The clock used as the timing reference can be selected from any of the physically recovered clocks from the received data streams or from the PTP derived clock (described in the following section). Logic outside of the ENET subsystem would typically route this recovered clock to an output pin of the device, which can then be distributed as a timing reference to any or all devices in a system.

### 5.12.3.12 IRIG Interface

The ETIME supports an interface which can be used to transfer time of day and a clock frequency across a backplane to timing slaves which operates in either a custom PMC time-of-day or an IRIG-B 800x mode. The ETIME can be an IRIG slave receiving a desired time-of-day from an external IRIG master, act as an IRIG master sending a desired time-of-day to another IRIG slave, or both simultaneously. The IRIG input interface performs the slave portion of the duties and is responsible for receiving the current time of day from a timing master and making this information available to the timing processor through readable registers. The IRIG output interface performs the master portion of the duties and is responsible for forwarding the local Time of Day to an IRIG slave. The IRIG output and IRIG input blocks operate using the PTP clock, nominally 125 MHz.

When the IRIG interface operates in IRIG-B B00x mode with a PWM encoded data stream, the encoded data stream provides a “frame” position which indicates the occurrence of an integer seconds increment event and data which provides the integer seconds value that corresponds to the integer seconds value after the increment event. The ToD interface consists of a clock, a 1PPS pulse and a serial data signal. The serial data sent over the data line indicates a 48-bit seconds value for both input and output interfaces, and optionally a 32-bit nanoseconds value for the IRIG input interface only. The rising edge of the 1PPS signal indicates the moment that the captured input/output time is valid. All interfaces provide data at 100 Hz.

There are two distinct modes that the IRIG block can be programmed to work in:

- TOD Interface mode with a distinct 1PPS signal and ToD serial interface
- IRIG-B B00x mode with the 1PPS and ToD information encoded on a pulse width modulated bit stream.

#### 5.12.3.12.1 IRIG Input Interface

The IRIG input interface can be used by an external master to manually control the frequency and phase (TOD) of the local clock. The IRIG input interface can be used as a substitute to time-of-day programming over PTP. When active, the IRIG input block forwards a 1PPS signal to the ETIME Timestamp capture circuitry to capture the FRC\_Tod and TOD values when it detects a 1PPS event at its input. The 1PPS input signal is sampled in the MAC clock domain to provide increased accuracy with respect to the PTP clock. The IRIG input desired time of day is a seconds only value (48 bits) in IRIG B00x mode, and 48-bit seconds and optionally 32-bit nanoseconds in TOD mode. This desired time of day arrives following the 1PPS input strobe. This value is stored in a register and used by firmware to modify the local time of day and IRIG output TOD value to be broadcasted and the local TOD offset, as applicable. The Timing processor can read the sampled local TOD and the TS\_PAGE, to determine what the current error of the local time of day is and apply corrections, as appropriate to the TOD offset. If everything is properly tuned, the TSTMP\_CFC\_CNTR's seconds value will have the same value as the incoming IRIG input desired time of day value and rollover precisely when the incoming IRIG input 1PPS signal arrives, assuming a zero nanoseconds value (when applicable). Since the ETIME samples the incoming 1PPS signal using the MAC clock, the IRIG input Time of Day accuracy is +/- 2ns, assuming a 652 MHz MAC clock.

The IRIG input can operate in one of two modes:

- TOD interface with Clock, Data, and 1PPS strobe. Optionally, the Clock and Data ports can be ignored, using only the 1PPS strobe for timing.
- Pulse-width modulated IRIG-B B00x, where x is typically 0, but can be 0 to 7 based on F/W support

For the IRIG TOD interface, the 1PPS signal is a 1Hz signal. For the IRIG B00x interface, the 1PPS signal is embedded in its data stream.

#### 5.12.3.12.2 IRIG Output Interface

The IRIG output interface is a block that can be used to present the TOD of the local clock to an external device. However, since the IRIG output block operates on the PTP\_clk, the output timing synchronization is accurate to +/-8ns. Based on this, the preferred mode of operation of the IRIG block is as a timing slave. The ETIME detects an upcoming rollover event at RX\_Tod\_nsec in the MAC clock domain and uses this signal to generate a 1PPS signal in the PTP\_CLK domain, and captures the local seconds TOD following the rollover event. The 1PPS signal and local seconds time of day value is used to broadcast the local time of day information to timing slaves. The IRIG output can operate in one of two modes:

- Clock, Data, and 1PPS Strobe
- Pulse-width modulated IRIG B B00y where y is typically 3, but can be 0 to 7 based on F/W support

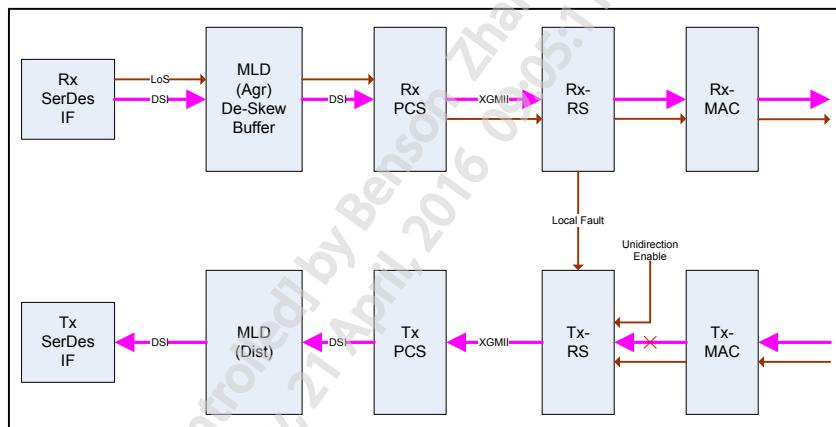
### 5.12.3.13 Signal Fault Handling and Consequential Actions

The ENET subsystem is capable of detecting faults that render a link unreliable for communication. Upon recognition of a signal fault condition, the consequential actions vary depending on the type of fault and the datapath in question. This section describes the various signal fault conditions in different modes and the expected behavior of the ENET. The ENET can be operated in Ethernet termination mode or in transparent mode (payload transparent or bit-transparent). In each of these cases the identification of signal fail condition and the response to the condition varies.

#### 5.12.3.13.1 Ethernet Client: MAC Termination Mode

The MLD Aggregator, Rx PCS and Rx Reconvergence Sublayer (RS) layer, along with the external SERDES are capable of detecting faults that render a link unreliable for communication. This condition is called link fault or the signal fail condition. [Figure 144](#) below shows the data path when a link in the ENET subsystem is configured to termination mode and interfaced with SERDES. This is applicable to both line side and system side.

[Figure 144](#) Signal Fail Condition in Ethernet Termination Mode with SERDES Interface



The RxMLD/Rx PCS/RxRS/Rx MAC blocks detect the following defects, all of which are on a per channel (link) basis. Each of these defects can trigger interrupts at defect detection and defect removal:

- LOS (via the DSI LOS pin)
- Loss of 66-bit synchronization (per SERDES lane)
- Loss of lane alignment (for 40G/100G links only)
- Local fault (LF) detection
- Remote fault (RF) detection
- High BER (per SERDES lane)

For the receive terminated path, beyond the interrupts, there is no consequential action in the receive direction. The Rx MAC will have no data to provide downstream.

Consequential action in the Tx MAC/TxRS/Tx PCS/TxMLD can be automatic or through register access. Automatic consequential actions occur in response to received defects, and only happen when the

MAC/PCS is set for bidirectional mode. When the unidirection\_enable bit is set to false (that is, the block operates in bi-directional mode), the following automatic consequential actions are enabled:

- When local faults are detected by the RxRS, Remote Faults (RF) are continuously transmitted towards the transmit SERDES. When the RS no longer receives LF ordered sets, the transmitter can return to sending MAC frames.
- When remote faults are detected by the RxRS, idle blocks are continuously transmitted towards the transmit SERDES. When the RS no longer receives RF ordered sets, the transmitter can return to sending MAC frames.

Consequential actions in the transmit direction can also be triggered via register access (typically in response to a defect elsewhere). The following consequential actions are possible, and can be enabled on a per-link basis:

- Insert Local Fault (LF) Ordered Sets
- Insert Remote Fault (RF) Ordered Sets

#### 5.12.3.13.2 Ethernet Client: Payload Transparent Mode

In Payload transparent mode, the Ethernet packets are not terminated in the Rx MAC. Instead, the Ethernet data is processed by a GSUP43 Clause 7.3 processor that passes the decoded packet data and ordered sets downstream to a packet buffer.

In 10GE Payload transparent mode, several fault types are identifiable on the receive data stream, each of which can trigger an interrupt:

- LOS (via the DSI LOS input)
- Loss of 66-bit synchronization
- Local fault (LF) detection
- Remote fault (RF) detection
- High BER

Upon detection of any of these fault conditions, the payload processor (GSUP43 Clause 7.3 processor) ceases to send any MAC data downstream, except in the case of uni-directional Ethernet. Typically, the OTN mapper would perform consequential actions based on the above defects. There are also several actions possible towards the downstream egress client interface, although their use is not mandated by the standards.

- In the case of LOS or loss of 66-bit synchronization, the data stream can be overwritten with LF ordered sets.
- In the case of LF or RF ordered set reception, the ordered sets are forwarded downstream via hardware. Note that 11/12 ordered sets are deleted prior to mapping, which can lead to the alarm not being declared if multiple mapper/demapper nodes are used or if the incoming fault density is low at the client interface.
- In the case of LF or RF ordered set reception using firmware, the ordered sets are regenerated and forwarded downstream. This ensures that each node will have the minimum density, as specified by the standard, to declare the LF/RF alarm downstream. Note that any data packets that are accompanying the LF/RF ordered sets will be overwritten, affecting IEEE 802.3 clause 66 unidirectional Ethernet.
- Note that when firmware mode is used for LF/RF propagation, the removal time of the LF/RF alarms will be delayed by up to 19 ms. During this time, all data packets will be lost.
- Also note that in the case where multiple 10GE channels are being mapped into GSUP43 C7.3 ODU2 payloads *and* a mix of channels supporting both the legacy UPI values and G7041 standard values

is required, LF and RF ordered set propagation must be done with firmware and will be subject to the caveats above.

Consequential actions are also possible in the corresponding transmit direction:

- Upon loss of synchronization or loss of signal in the receive direction, the transmit direction can be automatically (that is, without firmware intervention) overwritten with LF ordered sets.
- With firmware intervention, the transmit data stream can be overwritten with LF or RF ordered sets, or with an idle stream.

### 5.12.3.13.3 Ethernet Client: Bit Transparent Mode

In bit-transparent mode, both data and timing information pass through the device via the use of a packet maker and packet monitor pair that packetize and de-packetize the datastream.

For bit-transparent data paths, there is no automatic consequential actions in the transmit direction based on faults detected in the receive direction. Rather, most consequential actions are in-band, replacing the data stream with local fault ordered sets whenever fault conditions occur. Receive-to-transmit consequential actions are possible through firmware, however.

#### 5.12.3.13.3.1 Ingress Direction

Figure 122 shows the ingress data path inside the EPMM. Here is expected behavior of the ENET on the signal fault conditions.

In bit transparent mode, the receive SERDES carries 10GE (eventually mapped into OPU2e/OPU1e), 40GE (eventually mapped into OPU3) or 100GE (eventually mapped into OPU4). These data streams are all sent towards a packet maker block that packetizes the serial stream and tracks the recovered clock rate for each link. Many of the same fault conditions can be detected in the receive direction, and all of these conditions can generate interrupts to alert firmware of the faults:

- LOS (via DSI LOS pins)
- Loss of 66-bit synchronization (per SERDES lane)
- Loss of lane alignment (for 40G/100G links only)
- Local Fault (LF) Detection
- Remote Fault (RF) Detection

The consequential actions in the receive downstream direction (towards a packet buffer) are mostly the same as for the payload transparent mode. However, as shown below, there are also clocking considerations:

- Upon LOS, loss of 66-bit synchronization (for 40GE paths), loss of lane alignment (40GE and 100GE), or High BER (40GE), the data stream can be configured to be automatically replaced with LF ordered sets.
- Upon LOS, the recovered SERDES clocks cannot be relied upon (may be outside of the frequency and jitter requirements). As such, during a LOS conditions, the data stream will be replaced with ordered sets using a nominal receive clock. This allows timing information to be carried through the device.
- In addition to overwriting the data stream, upon LOS, loss-of synchronization (40GE), or loss of lane alignment (40GE and 100GE), the Signal Fail indication is indicated as a side-band signal to the packet maker, which passes the fault indication through the device.
- Firmware can also force the receive data stream to be overwritten with LF or RF ordered sets via register access.

### 5.12.3.13.3.2 Egress Direction

The [Figure 124](#) shows the egress data path inside the EPMM. The bit-transparent paths supports 10GE (from OPU2e or POPU1e), 40GE (from OPU3) and 100GE (from OPU4). There are several ways that a fault condition can exist on the egress path:

- Side-band signal that propagates through the device and that is extracted by the packet monitor. This Signal Fail indication is a reflection of the Signal Fail that was detected at the packet maker connected to this packet monitor.
- Packet monitor FIFO underrun for the link in question will also produce a Signal Fail condition.

Fault conditions can affect the transmit data stream in the following ways:

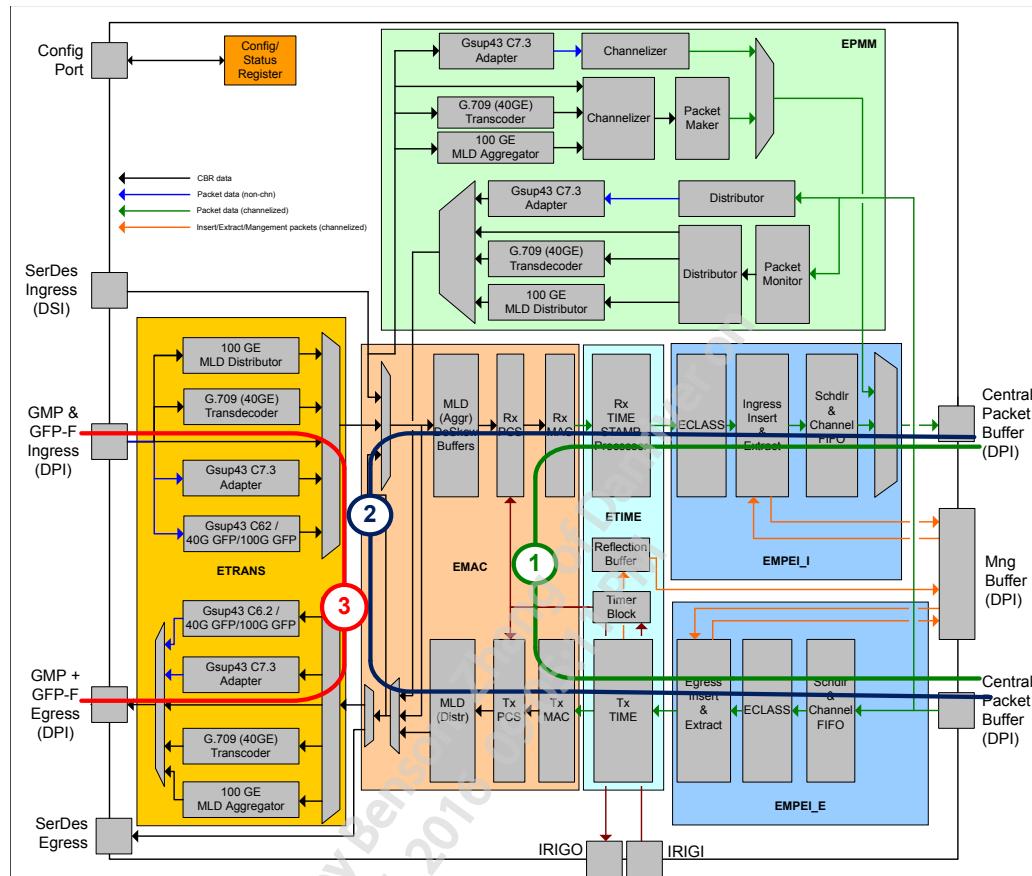
- Upon detection of the Signal Fail condition, the EPMM will overwrite the data stream with Local Fault ordered sets:
  - For 10GE, the 10GE CBR data path can be overwritten with Local Fault ordered sets
  - For 40GE, the G.709 40GE Transdecoder overwrites its outgoing data path with Local Fault ordered sets
  - For 100GE, the G.709 MLD distributor block overwrites its outgoing data path with Local Fault ordered sets
- Firmware can also force each of the outgoing data streams to be overwritten with Local Fault or Remote Fault ordered sets via register access. This can either be done as a consequential action to fault conditions on the corresponding receive link or as a diagnostic feature.

### 5.12.3.14 ENET Loopbacks

Three loopbacks exist in the ENET subsystem:

1. MAC Loopback (XGMII/XLGMII/CGMII): MAC Tx output to MAC Rx input
2. PMA Loopback: Tx PCS output to Rx PCS input
3. ETRANS PMA Loopback: ETRANS ingress output to ETRANS egress input (only functional for CBR datapaths).

Figure 145 ENET Loopbacks



Other loopbacks are possible in the device in adjacent subsystems, notably in the CPB and in the SERDES interface blocks (LIFD/SIFD). The existence of these loopbacks make any additional ENET loopbacks redundant.

All the above loopbacks can be enabled on a per-channel (link) basis.

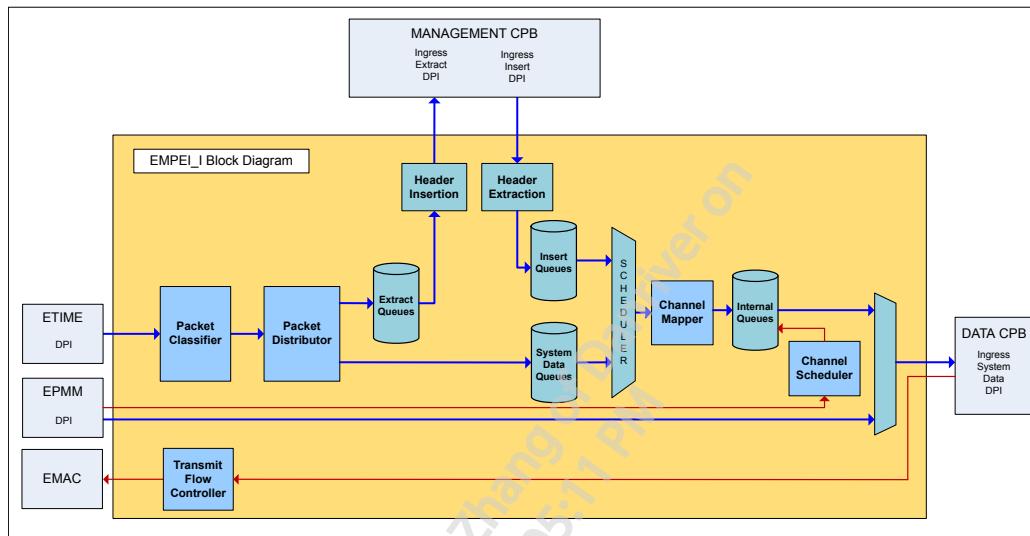
Data squelching in downstream direction can be enabled for all the different loopbacks:

1. For the MAC Loopback, the PCS block will automatically assert 0x00FF (repeating 16-bit pattern) on the PMA lanes for all 10GE paths. For 40GE and 100GE paths, the loopback squelch can optionally be enabled.
2. For the PMA Loopback, each PMA lane can optionally be squelched to 0x00FF (repeating 16-bit pattern) in the downstream direction.
3. For the ETRANS PMA loopback, the data can optionally be squelched to all-zeros towards the EMAC.

## 5.12.4 Ethernet Ingress Management Packet Extract / Insert (EMPEI\_I)

### 5.12.4.1 Block Diagram

Figure 146 Functional Block Diagram of the EMPEI Ingress



### 5.12.4.2 Overview

The Ingress EMPEI receives packets, data and control signals, via a customized DPI interface from the ETIME. Upon the first word of the packet, timestamp information such as Time of Day (TOD), Correction Field Count (CFC) and Page Number, are provided to the Ingress EMPEI over sideband signals. The received packet is provided to the Packet Classifier (ECLASS) for packet identification. The ECLASS module classifies the first 88 bytes of a packet (excluding any VLAN tags if present in the packet), via the general filter. Hence, any packet received, whether VLAN tagged or not, is seen as the same when classified by the General Filter. The VLAN filter, which removes the VLAN tags before reaching the general filter, can also provide information about the packet regarding the VLANs received. With information from the VLAN filter and the general filter, routing decisions can be taken (forward packet to the Data CPB, the Management CPB or to both). Note that the General Filter is comprised of a Hardwired filter and a Configurable Filter, supporting by default numerous standard packet types (PTP, Link OAM, etc), while still providing for configurability.

The packet leaving the Packet Classifier is identical to the packet received by the Packet Classifier (including the timestamp information), while a Classification Tag (General Tag plus VLAN Tag) and a Routing Tag are issued upon the first word of the outgoing packet. The Packet Distributor receives these packets and takes all routing decisions based on the routing tag received upon a packet's word. If the Routing Tag is all zero, or when bit 0 of the tag is asserted, the packet continues on its way towards the Data CPB. When bits[4:1] of the Routing Tag are non 0, the packet is issued to the Management CPB via the Extract block, with the destination channel decoded from these Routing Tag bits. Note that, in the DIGI 120G, only CPU and Management channels exist without any packet prioritization.

The packet received by the Extract Block is stored in an Extract Queue as rate adaption occurs for specific packet sizes, as a PMC Header is added to the received packet. A Header Insert block extracts packets from the Extract Queue and manipulates the packets to include a custom VLAN Tag (TPID =

0x8100, plus encoded VLAN TCI), a 32-bit Classification tag, and for specific packets a 128-bit timestamp. The modified packet is forwarded to the appropriate channel of the MCPB, via the Ingress Extract DPI interface.

A System Data Queue receives packets issued by the Packet Distributor, destined for the DCPB. A Weighted Round Robin (WRR) scheduler, scheduling on a per packet basis, schedules packets between the System Data Queue and an Insert Queue. The scheduled packets are forwarded to the Channel Mapper, discussed later, and then reach the Internal Queue. An Internal Queue allows for a transmission of packets to the DCPB based on the DCPB calendar rather than the calendar of the packets received by the Ingress EMPEI. Ultimately both calendars offer identical throughput, however, the Ingress EMPEI calendar is constant in nature, while the output calendar can be bursty in nature. Note that the WRR Scheduler schedules packets from the System Data Queue becomes priority once the System Data Queue reaches a defined threshold. Also, note that the WRR scheduler is multi-channel, supporting 12 links in a TDM fashion.

The Insert Path receives packets from the MCPB, via the Ingress Insert DPI interface. Received packets are first processed by a Header Extract module, which removes the PMC Header.

The primary function of the EMPEI is to support MAC Terminated packets, however, the EMPEI is the convergence point for Ingress Transparent traffic and MAC Terminated traffic. Hence, Ingress EPMM traffic is fed to the Ingress EMPEI and combined with packet data output from the Internal Queue. This new channelized data stream is issued to the DCPB via the Ingress System Data DPI interface.

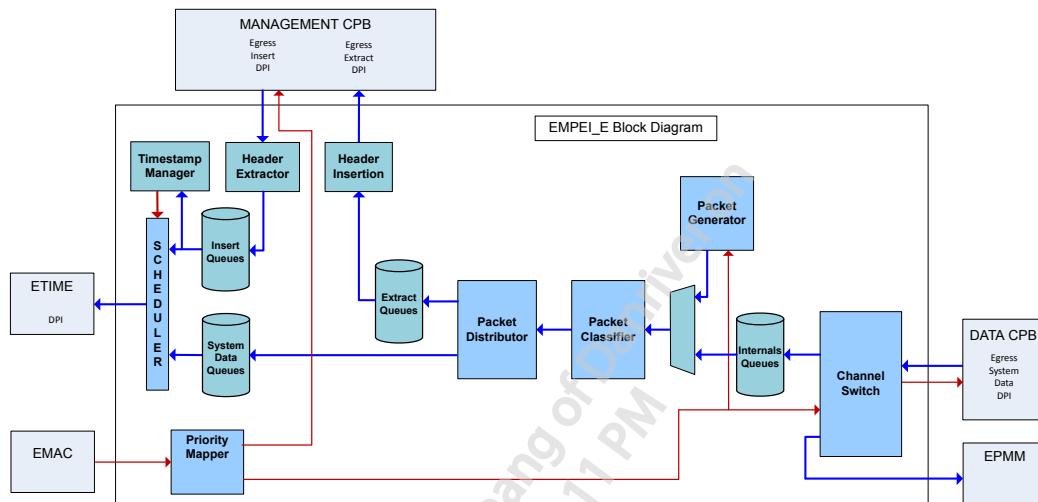
Finally, the Ingress EMPEI also supports Link Pause Flow Control, in conjunction with the DCPB and EMAC. The Transmit Flow Control module monitors the DCPB Zone levels and based on the configured mapping tables manages the EMAC, which ultimately notifies the remote entity with Link Pause messages.

Note that the EMPEI Queues are shallow and are not intended for store and forward purposes. The System Data Queue is sufficiently deep to ensure that ingress line rate is supported, without loss of data. When the System Data Queue reaches a defined threshold, the WRR scheduler is notified and must service the System Data Queue as soon as it terminates scheduling of the any current Insert packet. Based on the worst case insert packet size, the System Data Queue depth was established..

## 5.12.5 Ethernet Egress Management Packet Extract / Insert (EMPEI\_E)

### 5.12.5.1 Block Diagram

Figure 147 Functional Block Diagram of EMPEI Egress



### 5.12.5.2 Overview

The EMPEI Egress receives packets, data and control signals, Data CPB. Received packets can be routed towards the EPMM or towards the Internal Queue (MAC-terminated path), as defined by the configuration on a per link basis. Packets issued to the Internal Queue are buffered until sufficient data is captured, or an EOP is received, before the packet is extracted from the Internal Queue and forwarded towards the Packet Classifier (ECLASS). Sufficient buffering is provided by the Internal Queue to support burst-y reception of packets, from the DCPB (which operates off a 96 timeslot calendar), while supporting constant rate transmission of the packets (internal schedulers operate from a 12 timeslot calendar).

The Packet Classifier receives packets from the Internal Queue and performs packet classification based on the first 88 bytes of the packet. When one VLAN tag is present the classification is performed on the first 92 bytes, minus the 4 bytes of VLAN. When two VLAN tags are present the classification is performed on the first 96 bytes of the packet, minus 8 bytes of VLAN. The packet being classified is untouched, no removal of the VLAN tags, and output with various classification results(tags) from the Packet Classifier; including a Classification Tag (General Tag plus VLAN tag) and a Routing Tag. The output tags are valid upon the SOP of the packet. The Packet Distributor receives the packet with the Classification and Routing Tags and uses the Routing Tag to issue the packet to either the MCPB, via the Extract block, or to the EMAC via the EMPEI\_ETIME DPI interface.

The packet received by the Extract Block is stored in an Extract Queue as rate adaption occurs for specific packet sizes, as a PMC Header is added to the received packet. A Header Insert block extracts packets from the Extract Queue and manipulates the packets to include a custom VLAN Tag (TPID = 0x8100, plus encoded VLAN TCI) as well as a 32-bit Classification tag. The modified packet is forwarded to the appropriate channel of the MCPB, via the Egress Extract DPI interface.

The packet issued from the Packet Distributor towards the EMAC, is stored in the Egress System Data Queue. The System Data Queue is required to facilitate scheduling between system data traffic, issued from the DCPB, and inserted traffic, issued from the MCPB.

The insert path provides the CPU and the Management Port a means to insert control packets. The MCPB connects to the insert path via the Egress Insert DPI. Packets received over this interface are received by a Header Extract module. This module removes the custom VLAN tag from the incoming packet and stores the packet in the Insert Queue. The PMC Header is parsed to extract control information required for packet processing later on in the egress transmit path (i.e. packet requires an FCS to be appended, or the packet needs to be timestamped). The extracted control information is stored in the Insert Queue along with the packet to be inserted.

A Packet Generator provides test capabilities allowing a specific packet to be continually transmitted at a defined rate. The Ethernet header is fully programmable, while the payload is a repetition of the same byte. The Packet Generator can be enabled for any of the 12 links, and can operate while other links of the system are in use.

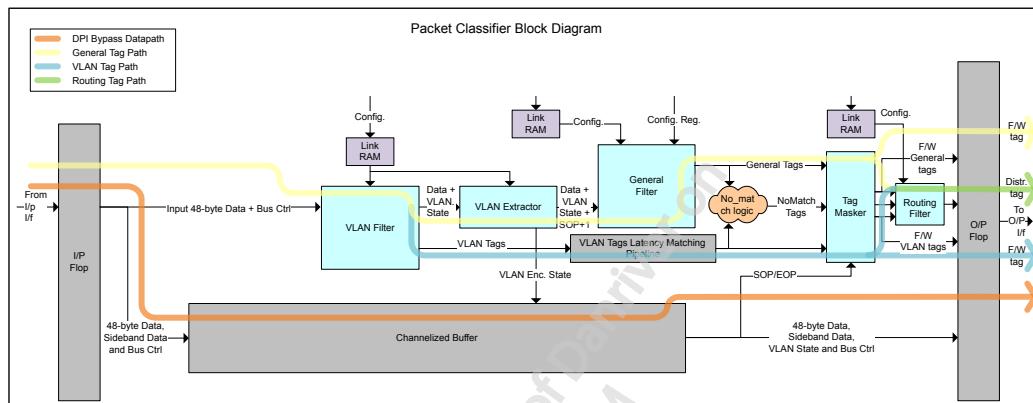
Finally, the Timestamp (TS) Manager is available to regulate the flow of inserted packets that require timestamping, or timestamp reflection by the EMAC. The TS Manager can operate in a handshake mode where all further TS packets are blocked from transmission until a downstream module confirms that the packet has been processed, or an independent timeout mechanism can be used to guarantee a minimum inter packet spacing (spacing between two adjacent packets requiring timestamping).

Downloaded [controlled] by Benson Zhang of Digi International  
Thursday, 21 April, 2016 09:05:11 PM

## 5.12.6 Ethernet Packet Classifier (ECLASS)

### 5.12.6.1 Block Diagram

Figure 148 Functional Block Diagram of ECLASS



### 5.12.6.2 Overview

The ECLASS receives input data and control signals via a customized DPI as well as sideband signals. These input signals are stored for each link and kept ready to be forwarded on the output port. The two first 48-bytes words of each packet for the same link are used to classify and identify the type of packet received with hard-wired and firmware-configurable filters. To achieve this, three stages of filters are available.

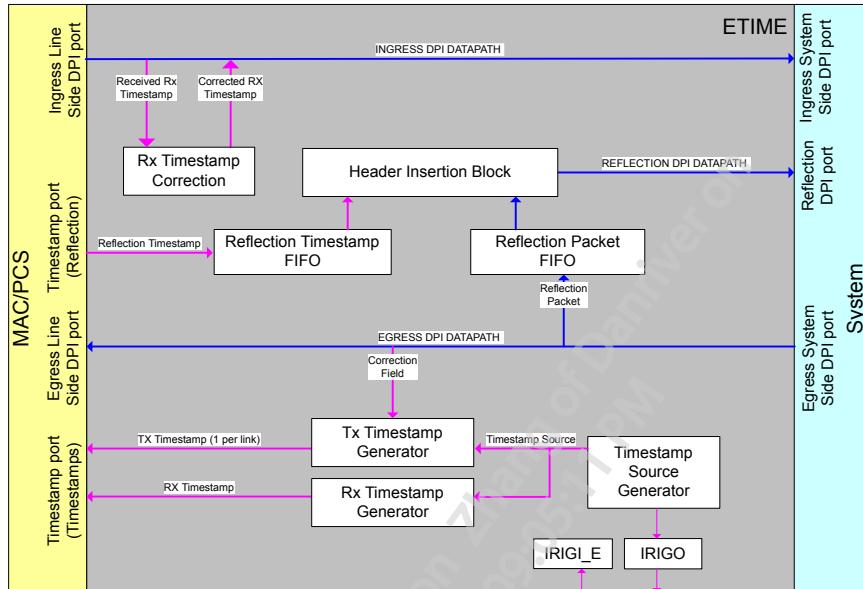
- A first stage is used to identify VLAN fields and produce VLAN tags.
- A second general stage of filtering identifies any wanted type of packet present on the VLAN stripped data. This allows an efficient filter resource usage to identify any type of packet, but also to distinguish packets of the same type based on their VLAN ID or stacked VLAN ID.
- The third stage of filtering combines any of the general and VLAN tags and produces a certain number of routing tags, which are used downstream from the ECLASS to determine the packet destination.

Note that the Routing tag configuration must match the downstream block functionality. Since both general and VLAN tags represent the most detailed information available, they can be sent to firmware as a classification information by a module that follows the ECLASS. This also outputs a VLAN state, identifying if 0, 1 or 2 VLAN tags are present. The DPI input data word and its sideband signals buffered within the ECLASS are output with all the calculated output signals synchronized on the output SOP.

## 5.12.7 Ethernet Timing Shim (ETIME)

### 5.12.7.1 Top-Level ETIME Block Diagram

Figure 149 ETIME TOD Block Diagram



### 5.12.7.2 Overview

The ETIME block is responsible for creating accurate time counters and handling the resulting ingress and egress values so that they can be used by the system in support of PTP functions ([22]). The ETIME's role is thus to generate and maintain time-of-day and correction field counters. There are two time-of-day counters in PTP TOD format: a TOD Free Running Counter (FRC\_ToD) and a TOD that is calculated using a system programmed offset to the FRC\_ToD counter. In the PTP protocol, the TOD format is defined as an 80-bit value with the most significant 48-bits representing integer seconds and the least significant 32-bits representing nanoseconds. There are two correction field counters: a CFC free running nanosecond counter (FRC\_CFC) and a correction field counter (CFC) in nanosecond format that is calculated by adding a system programmable offset to the FRC\_CFC counter. The correction field in the PTP packet is 64-bits wide, of which 48-bits represent the nanosecond value and 16-bits represent the sub-nanosecond value. Since this block is incapable of sub-nanosecond precision, the sub-nanosecond values are not modified in any way and the CFC counters can be limited to 48-bits.

From these time counters, the ETIME produces thirteen sets of running values which are sent to the MAC using the timestamp port. Twelve of these values are sent to the MAC for timestamping egress packets (one for each link). The thirteenth timestamp is used to timestamp ingress packets. The reason that independent timestamps are needed for the egress MAC is that each link may have different latency or require static offset and all egress timestamps may be required simultaneously. On the ingress side, a single timestamp is sent to the MAC timestamping unit and is used for all links. The captured timestamp then accompanies the SOP of the timestamped packet through the ETIME unit and is corrected by the ETIME for link latency and other static per-link offsets. In order to support one and two-step modes in egress, PTP packets with a "reflect enable" flag set are stored by the ETIME. The MAC is sent the packet data with a 2-bit packet identifier, with one bit representing a "reflection enabled"

bit and one bit representing an ID bit that will toggle between packets requiring two-step timestamping on a given link. The MAC then returns all captured timestamps for PTP events (both 1- and 2-step) using the Reflection Timestamp port. The ETIME only keeps the timestamps associated with a 2-step event (identified using the "reflection enabled" bit), this value is associated to the stored PTP packet in the reflection packet buffer and sent to the Management port or CPU through a dedicated DPI bus upon receipt of a returned timestamp. Packets transit from the system-side (EMPEI-side) to/from the MAC-side (MAC/PCS) using dedicated DPI links. The ETIME is designed to interface with a 10G/40G/100G compatible EMPEI block.

The ETIME also contains an IRIG input/output interface used to receive and transmit TOD information between devices or from a timing master as an alternative to setting the time directly using a PTP mechanism.

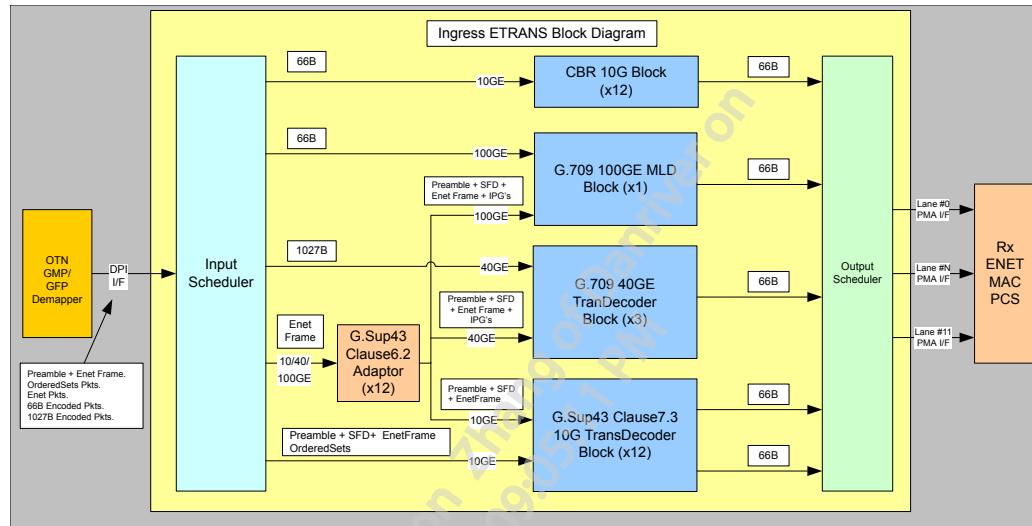
Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:11 PM

## 5.12.8 Ethernet Transcoder (ETRANS)

### 5.12.8.1 Block Diagram

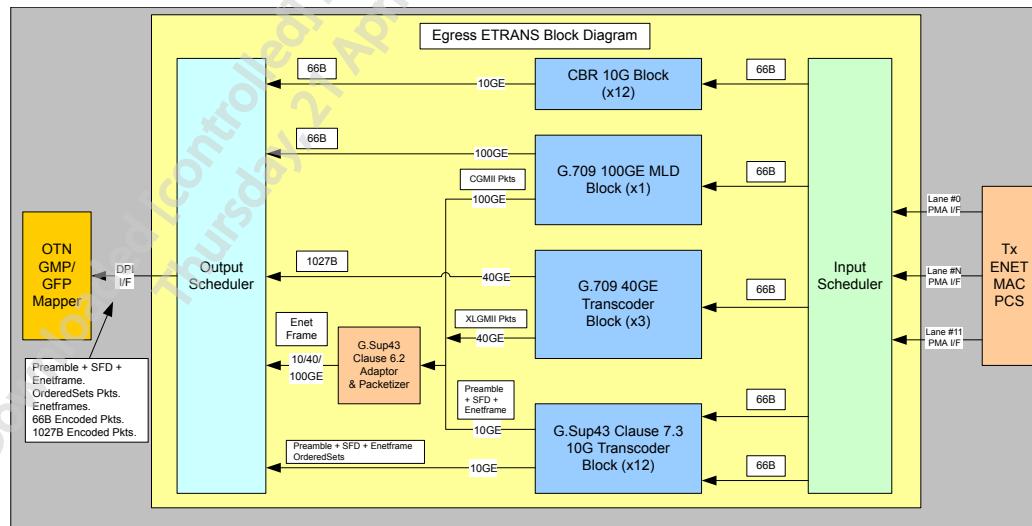
#### 5.12.8.1.1 Block Diagram of Ingress ETRANS

Figure 150 Top Level Block Diagram of Ingress ETRANS



#### 5.12.8.1.2 Block Diagram of Egress ETRANS

Figure 151 Top Level Block Diagram of Egress ETRANS



## 5.12.8.2 Overview

### 5.12.8.2.1 Overview of the Ingress Data Path

The ingress ETRANS block can be configured to operate in any of the following modes:

- 10GE mode: Supports a maximum of twelve 10GE links with an aggregate data rate of up to 120 Gbps
- 40GE mode: Supports a maximum of three 40GE links with an aggregate data rate of up to 120 Gbps
- 2x40GE mode and 4x10GE mode, with an aggregate data rate of up to 120 Gbps
- 1x40GE mode and 8x10GE mode, with an aggregate data rate of up to 120 Gbps
- 1x100GE mode with aggregate data rate of up to 100 Gbps
- Any other valid combination with an aggregate data rate of up to 120 Gbps

In the ingress direction, the ETRANS receives its input data and control signals in TDM fashion from OTN de-mapper through a customized DPI interface. Based on the configuration and operating mode of ETRANS, the ingress input mux routes these input data and control signals to any of the following data paths:

- G.Sup43 C7.3 trans decoder  
The G.Sup43 C7.3 Transdecoder engine processes the 10GE packets that includes preamble and SFD received over the OPU2 of the OTN container as described in Clause 7.3 of [27] and Clause 7.9 of [31].
- G.Sup43 C6.2 / G.7041 C7.1 Adaptor (also called as Ingress Ethernet Adaptor)  
The G.Sup43 C6.2/G.7041 adaptor block (also called as ingress 10/40/100G ethernet adaptor) receives the 10GE/40GE/100GE frames that do not include preamble, SFD and IPGs received over OPU2/3/4 of the OTN container as described in Clause 6.2 of [27] and Clause 7.1 of [31].  
The G.Sup43 C6.2/G.7041 adaptor block performs a pre-processing function that converts the received 10/40/100GE frames into Ethernet packets which includes preamble, SFD and IPGs, and generates the equivalent XLGMII or CGMII control codes for 40GE packets and 100GE packets respectively. Based on the configuration of the input mux, these 10/40/100GE packets are routed to either the G.Sup43 C7.3 10GE Transdecoder engine, the G.709 40GE Transdecoder engine, or the G.709 100GE MLD engine respectively.
- G.709 40GE Transdecoder  
The G.709 40GE Transdecoder engine processes the 40G 1024B/1027B blocks received over OPU3 of OTN container as described in Section 17.7.4.1 and Annexes B, E and F of [28] as well as Clauses 82 and 83 of [24].
- G.709 100GE MLD Block  
The G.709 100GE MLD engine processes the 100G aggregated 64B/66B blocks received over OPU4 of OTN container described in Section 17.7.51 and Annex E of [28].
- G.Sup43 C7.1/C7.2 10G CBR Block  
The CBR 10G Transdecoder engine processes the 10G 64B/66B blocks received over OPU2e/OPU1e of the OTN container as described in Clauses 7.1 and 7.2 of [27] and Clause 17.2 of [28].

Once the data and control signals are routed to the corresponding path, then the Transdecoder engine present in that path processes the data and control signals accordingly, and transmits the corresponding data and control signals to Rx Ethernet Mac through 12 links of the PMA interface as per configuration.

### 5.12.8.2.2 Overview of the Egress Data Path

In the egress path, the ETRANS can be configured to operate in any of the modes described in [Section 5.12.8.2.1](#).

In the egress direction, the ETRANS receives its input data and control signals from Tx Ethernet MAC through 12 lanes of the PMA interface. The egress dsi mux, based on the configuration and operating mode, routes these inputs to any of the data paths mentioned below.

**Note:**

- Since the TxMAC is the source for egress path of ETRANS block, LOS is meaningless. That is, since the data path is coming from the TxMAC there is no LOS (Loss of Signal) in the egress path.

The packets that are processed by below mentioned paths 1 and 2 are routed to GFP mapper block, whereas the packets processed by paths namely 3, 4 and 5 are routed to GMP mapper block, present in MAPOTN subsystem.

1. G.Sup43 C7.3 Transcoder
  - The G.Sup43 C7.3 Transcoder engine after performing necessary transcoding function, transmits the 10GE packet (which includes Ethernet preamble, SFD and Ethernet frame) to OTN GFP mapper subsystem for mapping into OPU2 payload cavity of ODU2/OTU2 frame, as described in Clause 7.3 of [\[27\]](#) and Clause 7.9 of [\[31\]](#).
2. G.Sup43 C6.2 Adaptor (also called as Egress Ethernet Adaptor)
  - The G.Sup43 C6.2 adaptor (also called as 10/40/100G egress ethernet adaptor) block after processing transmits the 10GE/40GE/100GE frames (which does not include preamble, SFD and IPG's) to OTN GFP mapper subsystem for mapping into OPU2 payload cavity of ODU2/OTU2 frame, as described in Clause 6.2 of [\[27\]](#) and Clause 7.1 of [\[31\]](#).
3. G.Sup43 C7.1/7.2 10G CBR Block
  - The G.Sup43 C7.1/7.2 10G CBR engine after processing transmits the 10G 64B/66B transcoded Ethernet packet to OTN GMP mapper subsystem for mapping into OPU2e/OPU1e payload cavity of the OTN frame, as described in Clauses 7.1 and 7.2 of [\[27\]](#) and Clause 17.2 of [\[28\]](#).
4. G.709 40GE Transcoder.
  - The G.709 40GE Transcoder engine after processing transmits the 1024B/1027B transcoded 40GE blocks to OTN GMP mapper subsystem for mapping into OPU3 payload cavity of ODU3/OTU3 frame, as described in Section 17.7.4.1 and Annexes B, E and F of [\[28\]](#) as well as Clauses 82 and 83 of [\[24\]](#).
5. G.709 100GE MLD Block
  - The G.709 100GE MLD engine after processing, transmits the 100G 64B/66B transcoded Ethernet packet to OTN GMP mapper subsystem for mapping into OPU4 payload cavity of ODU4/OTU4 frame, as described in Section 17.7.5.1 and Annex B of [\[28\]](#) as well as Clauses 82 and 83 of [\[24\]](#).

On receiving the pcs stream of input data from Tx Ethernet MAC, the transcoder engines in the corresponding path processes the data and control signals as per configuration. After processing, these data and control signals are transmitted to the OTN GFP or GMP mapper blocks through a customized DPI, which maps these transcoded 10/40/100GE traffic into OPU2/OPU1e/OPU2e/OPU3/OPU4 payload cavity of ODU2,1e,2e/3/4 or OTU2,2e/3/4 frames respectively.

The DPI control signal named "Packet Length" received in egress direction from an upstream block named "TX EMPEI" as a sideband signal is processed and transmitted to the GFP mapper at the SOP through an egress DPI.

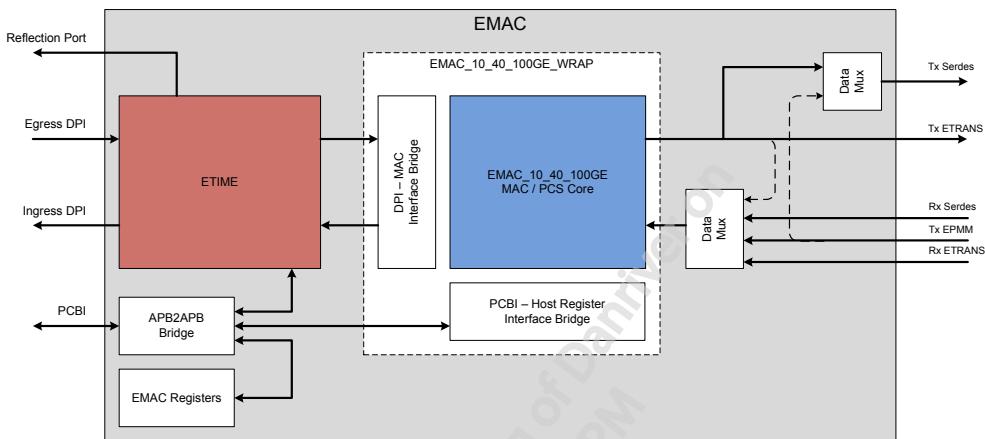
The ETRANS block in both ingress and egress direction supports a PCBI interface to perform read/write access to configuration and status registers.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:11 PM

## 5.12.9 Ethernet PCS/MAC (EMAC)

### 5.12.9.1 Block Diagram

**Figure 152** EMAC120 High-Level Block Diagram



### 5.12.9.2 Channelized MAC/PCS Overview

The channelized 10GE/40GE/100GE MAC/PCS block complies with the applicable clauses of the IEEE Standard 802.3ba-2010.

In the ingress direction, the PMA interfaces of the MAC/PCS Core select, on a per-link basis, the data source. The supported data paths are:

- Receive SERDES: Reception of a 10GE/40GE/100GE stream
  - Transmit to Receive PMA Loopback: Loopback of the egress 10GE/40GE/100GE link into the ingress path
  - Receive Ethernet Transcoder (ETRANS): 10GE/40GE/100GE encapsulation of a de-mapped OTN stream
  - Transmit Ethernet Packet Maker and Monitor (EPMM): Performance monitoring a 10GE/40GE/100GE stream

Once the source is selected for each link, the Rx PCS 66B block lock is achieved on each of the PMA lanes before each of the data streams are individually buffered. Lanes belonging to 40GE and 100GE channels are de-skewed (de-skew does not apply to 10GE channels). Afterwards, the de-skewed lanes are de-scrambled and passed on to the Rx MAC, which recovers the packet delineation, checks for protocol violations and FCS validity. The Rx MAC block passes all received frames downstream, with the exception of link pause frames. Errorred frames can optionally be tagged with a drop "flag". The Rx MAC implements the state machine and timers to support link pause, and asserts indications to inform downstream buffers to halt transmission. The Rx MAC pause frames with a unicast or multicast destination address are processed. Finally, received Ethernet packets from the ingress PMA interfaces are time stamped and forwarded to the ETIME block.

In the egress direction, the Tx MAC accepts Ethernet packet from the ETIME block and transforms the data into an XGMII/XLGMII/CGMII stream. The Tx MAC optionally computes the FCS for outgoing packets. The Tx MAC is also responsible for implementing link pause flow control by generating pause frames based on local triggers. The Tx MAC only supports the multicast address as the destination for the pause frames. Apart from the pause frames, the Tx MAC does not generate any other packets.

Downstream, the Tx PCS scrambles and 66B/64B-encodes the MII data into PMA streams. For 40GE/100GE links, the PMA interfaces of the MAC/PCS block inserts lane alignment markers per the 802.3ba-2010 standard. The PMA interfaces of the MAC/PCS block support mutually exclusive connections to:

- Transmit SERDES: Transmission of a 10GE/40GE/100GE stream
  - Optionally, the transmit EPMM stream can be forwarded to the Tx SERDES interface.
- Transmit Ethernet Transcoder (ETRANS): 10GE/40GE/100GE OTN-mapped stream

A rich set of PMON counters are available within the MAC block. The MAC block provides an independent set of statistic counters for each of the 12 available channels. Events are captured on both receive and transmit, which are in turn conveyed to a channelized statistics counter block. All counters are implemented with sufficient bit precision to capture events per channel at up to a 100Gbps line rate while still maintaining IEEE rollover compliance to 58 minutes (with firmware assistance). The following statistics are supported:

- IEEE 802.3 clause 30: Basic, Mandatory, Recommended packages
- RFC 2665 (MIB), RFC 3635 (updated for SNMP)
- RFC 2819 (RMON)

In an effort to use area efficiently, the application layer interface of the MAC/PCS implements a channelized interface. A different channel may be presented on the interface every clock cycle, in effect providing a TDM, round-robin interleave. Over a period of 12 clock cycles (a rotation), all channels will have had an opportunity to transmit and receive data.

The 12 channels are each statically configured for a specific rate (10GE/40GE/100GE), which in turn defines the number of timeslots that are used for the channel per rotation. A 10GE link will consume 1 timeslot, a 40GE link will consume 4 timeslots and a 100GE link will consume 10 timeslots. A valid signal provides the ability to denote unused cycles (due to interface over speed) or unallocated channels.

#### 5.12.9.2.1 MAC Assignments for 10GE/40GE/100GE Applications

The device supports twelve 10 GE MACs. A group of four MACs support 40 GE and a group of ten MACs support 100 GE. The assignment of MAC instances in the Ethernet Line and Ethernet System to 10GE/40GE/100GE groups is fixed. MAC instances 0 to 3, 4 to 7, and 8 to 11 can be ganged to operate as a 40 GE MACs. Similarly, MAC instances 0 to 9 can be ganged to operate as a 100 GE MAC. For datapaths to or from the Line and System side SerDes the MAC groupings are expected to be associated with 40 GbE or 100 GbE optics. Re-arrangement of the Ethernet to MAC assignments is limited by the physical topology of the line card and will not be necessary. Packets from the Interlaken Interface or demapped from a MO/LO ODUk may have an arbitrary mix of 10 GE / 40 GE / 100 GE signals (in portless server card applications, for example). Provisioning and re-provisioning of the device may lead to a scenario where re-arrangement of signals assigned to MACs is necessary to accommodate the 40 GE / 100 GE grouping. Re-arrangement may also be necessary for applications with a mix of Ethernet to line or system side SerDes and packet traffic adapted from MO/LO ODUk signals or received from the Interlaken interface.

#### 5.12.9.2.2 Runt Packet Drop and Count Behavior

Run packets received by the 10 GbE, 40 GbE, or 100 GbE MACs are flagged and counted as runt packets by the MACs. In the case of the 10 GbE MAC, the MAC performs the drop. In the case of the 40 GbE and 100 GbE MACs, the packets are marked with an internal signal as DPI\_DROP and dropped by the DCPB. Therefore, the DCPB PMON\_CNTx\_VAL counters, which are used for debug only, will not count 10 GbE runt packets when PMON\_CNTx\_PKT\_FILT is configured to select packets marked as

DPI\_DROPed. The PMON\_CNTx\_VAL will include runt packets for 40 GbE and 100 GbE clients when PMON\_CNTx\_PKT\_FILT is configured to select packets marked as DPI\_DROPed.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:11 PM

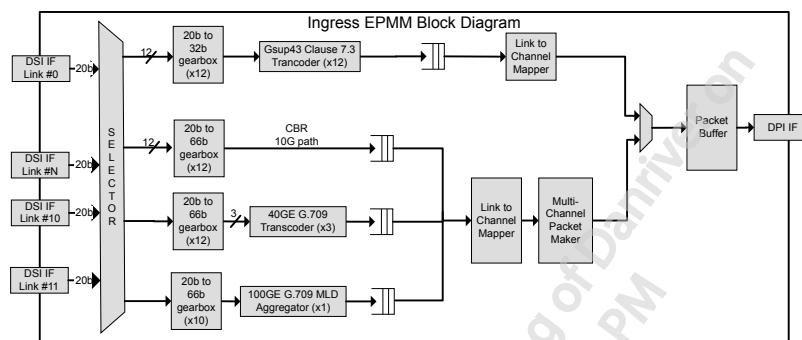
## 5.12.10 Ethernet Packet Maker/Monitor (EPMM)

### 5.12.10.1 Block Diagram

#### 5.12.10.1.1 Ingress EPMM (Packet Maker) Functional Block Diagram

Figure 153 shows the top-level functional block diagram of the ingress EPMM.

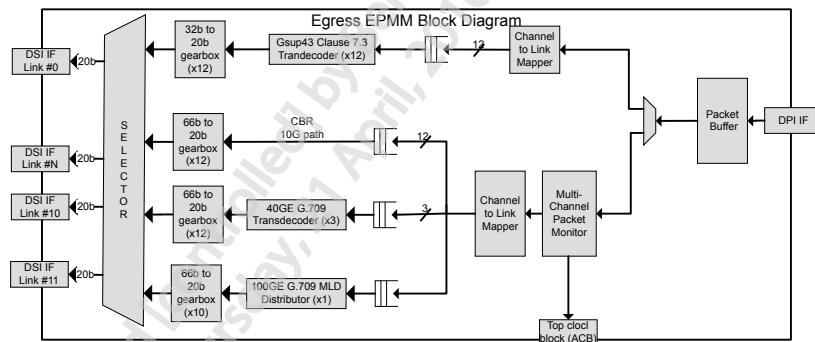
Figure 153 EPMM Ingress Block Diagram



#### 5.12.10.1.2 Egress EPMM (Packet Monitor) Functional Block Diagram

Figure 154 shows the top-level functional block diagram of the egress EPMM.

Figure 154 EPMM Egress Block Diagram



### 5.12.10.2 Overview

#### 5.12.10.2.1 Overview of the Ingress Path

The ingress EPMM supports direct mapping of 64B/66B data stream from a SERDES to the CPB through the data paths described below. Figure 153 shows the ingress data path of the EPMM.

The 10GE payload transparent data path is provided through a Gsup43 clause 7.3 Transcoder engine. A single instance of the Gsup43 Clause 7.3 Transcoder engine supports one 10GE SERDES link. Twelve 10GE Gsup43 C7.3 transcoder engines are instantiated in order to support an aggregate data rate of up to 120 Gbps. In this mode, the 10GE bit stream is converted to a 32-bit wide data bus and passed on the Gsup43 Clause 7.3 Transcoder engine (see Section 7.1 of [31]). In the transcode block there exists a 64B/66B Tx PCS block. This block finds the 64B/66B frame position, implements a 64/66-bit

control code decoder function for the IEEE 802.3 Ethernet MAC layer (see Clause 49 of [23]) at 10GE link speeds, descrambles the incoming data stream with a 1+ X39 + X58 self-synchronizing scrambler, which bypasses the 2-bit sync header, implements a 33:32 rate adaptation function, and removes the IPG from the data stream. The output of the Transcoder engine is channelized to meet the requirements of the DPI.

The 10GE CBR provides transparent data path through the channelized packet maker. A single instance of the CBR engine supports one 10GE SERDES link. Twelve CBR 10GE engines are instantiated in order to support an aggregate data rate of up to 120 Gbps. In this path, the ingress EPMM supports mapping of 10GE traffic in bit transparent mode as defined in ITU-T Gsup43 Clause 7.1 (see Section 7.1 of [27]). In this mode, data from up to 12 links are channelized and passed on to the packet maker. The packet maker receives a continuous stream of data frames from any of the twelve 10GE SERDES links. The packet maker gathers consecutive bytes of data into artificial packets, which are written into the CPB by inserting an SOP and EOP along with necessary control signals.

The G.709 40GE Transcoder bit provides transparent data path through the channelized packet maker. A single instance of the G.709 40GE Transcoder engine supports four 10GE SERDES links. Three G.709 40GE Transcoder engines are instantiated in order to support an aggregate data rate of up to 120 Gbps. Instance #0 is formed by combining 10GE SERDES links 0, 1, 2 and 3. Instance #1 is formed by combining 10GE SERDES links 4, 5, 6 and 7. Instance #2 is formed by combining 10GE SERDES links 8, 9, 10 and 11. In this path, the SERDES data is received on four 20-bit wide links. The 64B/66B Tx PCS block finds 64B/66B frame positions and alignment markers on each of these links. It aligns the 64B/66B frames from four links into a single stream (see Clauses 82 and 83 of [24]). This block converts the 64B/66B frame into 1024/1027-bit frames (see Section 17.7.4.1 and Annexes B, E and F of [28]) before forwarding them to the channelizer. The output of the channelizer is forwarded to the packet maker to convert the 40GE data stream into packets, which are forwarded to the CPB.

The G.709 100GE Transcoder provides a bit transparent data path through the channelized packet maker. A single instance of the G.709 100GE Transcoder engine supports ten 10GE SERDES links, which can support an aggregate data rate of up to 100 Gbps. This single instance is formed by combining the 10GE SERDES links 0 to 9. In this path, the SERDES data is received on ten 20-bit wide links. From these 10 links, 20 virtual lanes are formed by segregating odd and even bits (see Clauses 82 and 83 of [24]). The 64B/66B Tx PCS block finds the 64B/66B frame positions and alignment markers on each of these PMA lanes. It aligns the 64B/66B frames from 20 PMA lanes into a single stream. This stream is sent to the packet maker to convert this 100GE data stream into packets that are forwarded to the CPB.

### 5.12.10.2.2 Overview of the Egress Path

The egress EPMM supports direct mapping of packetized data from the CPB to the SERDES links as 64b/66b data streams using the data paths described below. [Figure 154](#) shows the egress data path of the EPMM.

The 10GE payload transparent data path through Gsup43 Clause 7.3 Transdecoder engine. A single instance of the Gsup43 Clause 7.3 transdecoder path supports one 10GE SERDES link. Twelve instances of the Gsup43 C7.3 Transcoder engines are instantiated to support an aggregate throughput of 120 Gbps. In this mode, the incoming transcoded packets for 10GE are converted from channelized as a 384-bit bus to a 32-bit dedicated data bus on a per link basis. The Gsup43 Clause 7.3 Transdecoder (see Section 7.1 of [31]) logic for each of the 12 links has a Transmit Formatter, which simply takes the signals on the Gsup43 Transdecoder's line interface and, based on the type field, identifies type of information as a data packet or an ordered set. It also identifies characteristics of these signals such as the SOP and EOP positions. This information is passed to the subsequent blocks and the PMON counters. The Gsup43 block optionally controls the IPG between packets and ordered sets. By default, an average of 12 bytes of IPG are inserted between packets and ordered sets. The 64B/66B encoder inside the

transdecode logic implements all the required functionality of the 10GE PCS as specified in the IEEE 802.3 specification (see Clause 49 of [23]). It performs frame delineation on frames, scrambling and a 64B/66B decode function on the scrambled stream. The 64B/66B input stream consists of standard Ethernet (Preamble/SFD, MAC header, data, pad and CRC) frames, ordered sets, and IPGs. The output is pushed out on a 32-bit data bus, which in turn is converted to a 20-bit bus and sent out on the SERDES. It is important to note that the Gsup43 Transdecoder and 32b-to-20b gearbox operate at a nominal 10GE speed.

The 10GE CBR provides a transparent data path through the channelized packet monitor. A single instance of CBR engine supports one 10GE SERDES link. Twelve instances of CBR 10GE engines are instantiated to support an aggregate data rate of up to 120 Gbps. In this path, the egress EPMM also supports 10GE bit transparent traffic in bit transparent mode as defined in ITU-T Gsup43 Clause 7.1 (see Section 7.1 of [27]). In this mode, the artificial packets are read from the CPB and a 64B/66B-encoded data stream is reconstructed by the packet monitor for each of the 12 links. The 66-bit to 20-bit gearbox converts the 66-bit-wide data stream to 20-bits wide and pushes it to the SERDES.

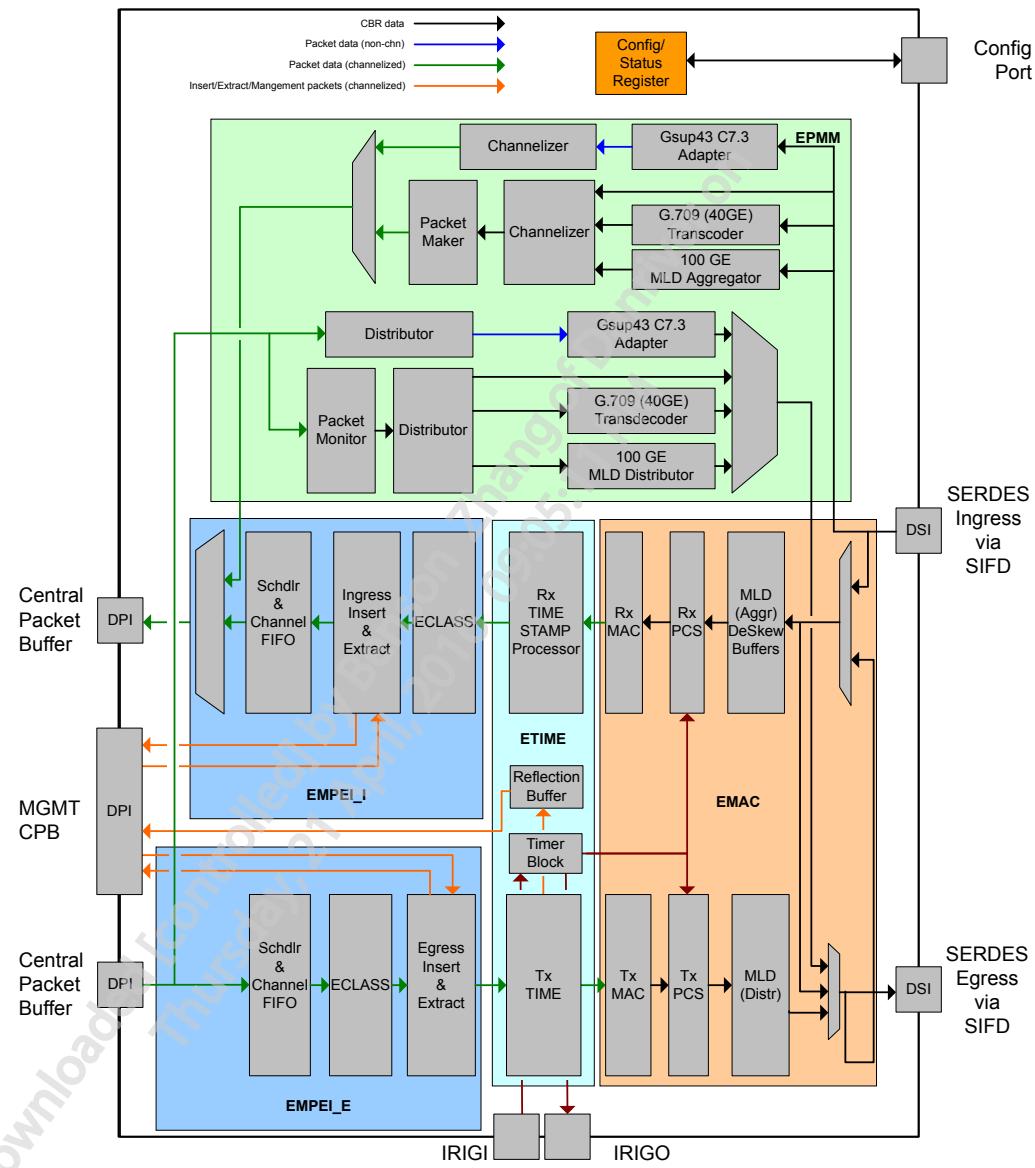
The G.709 40GE Transdecoder provides a bit-transparent data path through the channelized packet monitor. A single instance of the G.709 40GE Transcoder engine supports one 40GE (that is, four 10GE SERDES links). Three instances of the G.709 40GE Transcoder engine are instantiated to support an aggregate data rate of up to 120 Gbps. Instance-0 is formed by combining 10GE SERDES links namely 0, 1, 2 and 3. Instance-1 is formed by combining 10GE SERDES links namely 4, 5, 6 and 7. Instance-2 is formed by combining 10GE SERDES links namely 8, 9, 10 and 11. In this path, the artificial packets are read from the CPB and the packet monitor reconstructs the 1027B encoded data stream. This data stream is passed through 384 to 256 bit converter, and G.709 1027B Transdecoder logic. The G709's Transdecoder logic converts the 1027-bit data into a stream of 64B/66B and the MLD logic inside the transdecode distributes this 64B/66B stream as four streams of 66 bits, as specified in the IEEE P802.3ba/D2.3 specification, along with the insertion of alignment markers. The 66-bit to 20-bit converter converts the data to 20 bits and pushes it to the SERDES.

The G.709 100GE Transdecoder provides a bit transparent data path through the channelized packet monitor. A single instance of the G.709 100GE Transcoder engine supports one 100GE (that is, ten 10GE SERDES links that can support an aggregate data rate of up to 100 Gbps). In this path, the artificial packets are read from the CPB and the 64B/66B data stream is reconstructed by the packet monitor. The 64B/66B data stream is distributed as 20 PCS lanes, which in turn is converted into 10 links by multiplexing two PCS lanes as specified in IEEE P802.3ba (see Clauses 82 and 83 of [24]), along with the insertion of alignment markers. The 66-bit-to-20-bit converter converts the data to 20 bits and push it to the SERDES.

## 5.13 Ethernet System-side (ENET(SYS)) Subsystem

### 5.13.1 Block Diagram

Figure 155 ENET(SYS) Block Diagram



## 5.13.2 Overview

The system-side instantiation of the ENET subsystem (ENET(SYS)) exists to provide connectivity between the system-side SERDES blocks and the CPB, enabling system-side SERDES blocks to carry Ethernet traffic and performance monitoring of system-side Ethernet links. In the device, this enables applications where OTN on the line-side SERDES is mapped/demapped to/from Ethernet on the system-side SERDES.

Since in the device, OTN mapping and de-mapping functions are only attached to line-side SERDES blocks, the ENET(SYS) does not need to interact directly with the MAPOTN subsystem. For this reason, the ENET(SYS) does not include the ETRANS block. Apart from the absence of the ETRANS block, the ENET(SYS) is identical to the ENET(LINE). They have the same functionality, the same connectivity to the CPB and the management switch (MCPB).

With one exception, the ENET(LINE) and ENET(SYS) operate independently and can be configured differently as warranted by the application. The one inter-dependency is respect to the MCPB. Only one ENET instance can access the MCPB for any one application. This implies that only one instance of the ENET subsystem can extract packets to the CPU or management port, or insert packets from the CPU or management port any any time. The ENET-MCPB selection is performed by a register setting at the MCPB, which is expected to be static for any one application.

## 5.13.3 Ethernet Ingress Management Packet Extract / Insert (EMPEI\_I)

Please see [Section 5.12.4](#) for the EMPEI\_I functional description in the ENET(SYS).

## 5.13.4 Ethernet Egress Management Packet Extract / Insert (EMPEI\_E)

Please see [Section 5.12.5](#) for the EMPEI\_E functional description in the ENET(SYS).

## 5.13.5 Ethernet Packet Classifier (ECLASS)

Please see [Section 5.12.6](#) for the ECLASS functional description in the ENET(SYS).

## 5.13.6 Ethernet Timing Shim (ETIME)

Please see [Section 5.12.7](#) for the ETIME functional description in the ENET(SYS).

## 5.13.7 Ethernet PCS/MAC (EMAC)

Please see [Section 5.12.9](#) for the EMAC functional description in the ENET(SYS).

## 5.13.8 MAC Assignments for 10GE/40GE/100GE Applications

Please see [Section 5.12.9.2.1](#) for MAC assignments in the ENET(SYS)

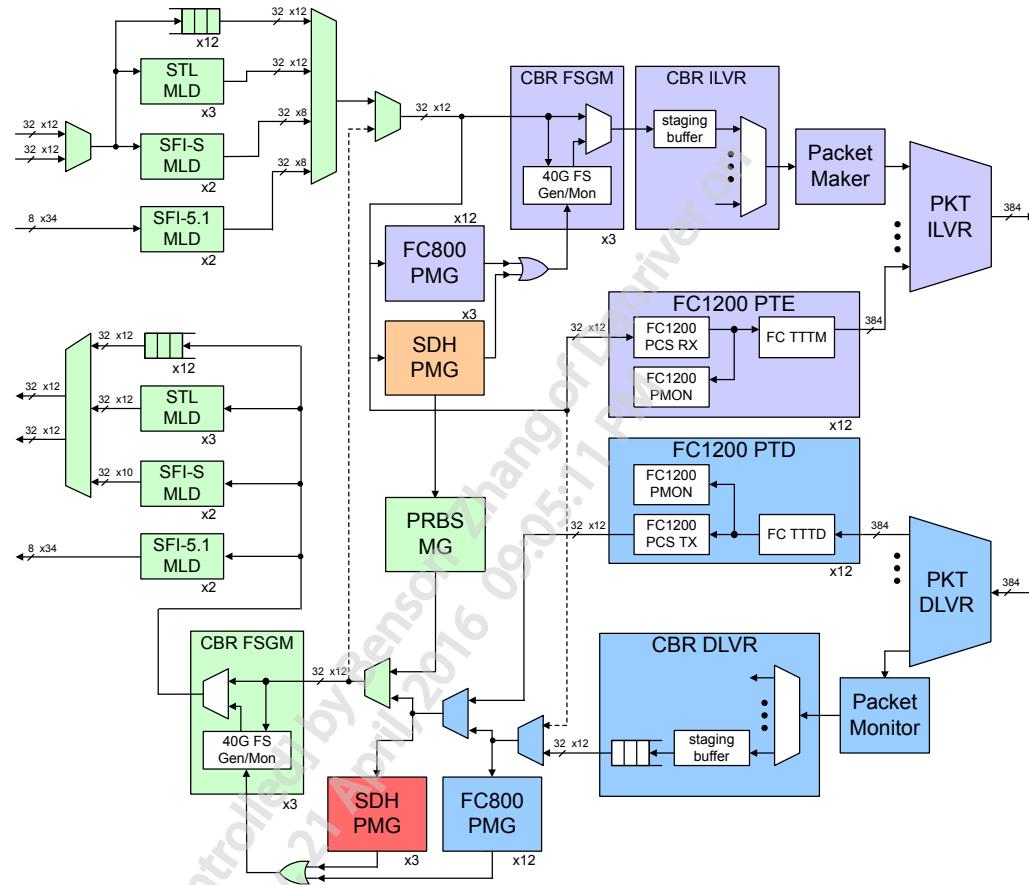
## 5.13.9 Ethernet Packet Maker/Monitor (EPMM)

Please see [Section 5.12.10](#) for the EPMM functional description in the ENET(SYS).

## 5.14 Constant Bit Rate Client (CBRC) Subsystem

### 5.14.1 Block Diagram

Figure 156 CBRC Block Diagram



Note:

1. The PRBS\_MG block is composed of two complementary blocks, PRBS\_RX and PRBS\_TX.

### 5.14.2 Overview

The CBRC (Constant Bit Rate Client) subsystem is used to transition CBR (Constant Bit Rate) client data between individual SERDES streams and an aggregate multi-channel packet interface. As shown in the CBRC block diagram, Figure 156, the CBRC provides a CBR ingress datapath and a CBR egress datapath. The ingress datapath is used to receive single and multi-lane SERDES streams, monitor the performance of each stream and capture the stream's rate as it is packetized for aggregation onto a common packet interface. The egress datapath is used to receive multi-channel packet data, recover the constituent client stream rates as the data is converted into CBR streams, performance monitor each stream and distribute the CBR streams to a mix of single and multi-lane egress interfaces.

### 5.14.2.1 CBR Ingress Datapath

The CBR ingress datapath is shown in the top half of the CBRC block diagram shown in [Figure 156](#). The major ingress functional groupings are the ingress MLD, ingress PMON and errored stream substitution, and the ingress interleaving, encoding and packetization.

The ingress Multi-Lane Distribution (MLD) grouping is used to receive single and multi-lane client streams as they enter the CBRC. Each stream entering the MLD grouping operates in the recovered receive clock domain of its source SERDES. The ingress datapath of the CBRC supports 12 single lane interfaces and 56 multi-lane interfaces. The 56 multi-lane interfaces are composed of 2 x 17 lanes providing two SFI-5.1 interfaces, 2 x 5 lanes providing two SFI-S(n=4+1) interfaces, and 3 x 4 lanes providing three STL256.4 interfaces. Multi-lane streams are processed by various MLDs to deskew and merge the lanes into an aggregate stream. 12 streams are chosen from among the single lane streams and the outputs of the ingress MLD blocks.

The ingress PMON grouping is used to provide performance monitoring for each stream as well as stream substitution for errored streams. The PMON and errored stream substitution for ingress FC1200 streams are combined with the transcode-encode block

The ingress interleaving, encoding and packetization grouping is used to process the datapath along two distinct paths:

- The first path interleaves the client streams onto a multi-channel CBR interface followed by a MPMA to create packets from each CBR stream, encoding each stream's rate in the size of packets created.
- The second path provides transcode encoding for up to 12 Fibre Channel FC-1200 streams required to rate adapt (reduce) ingress FC-1200 streams for transparent transmission within an ODU2e.

The outputs from the MPMA and the FC-1200 transcode encode modules are then merged into a common packet interface for egress from the CBRC.

### 5.14.2.2 CBR Egress Datapath

The CBR egress datapath is shown in the bottom half of the CBRC block diagram shown in [Figure 156](#). The major functional groupings in the egress datapath are the egress de-interleaving (PKT\_DINT, CBR\_DINT), decoding and de-packetization (MPMO, FC1200\_PTD), the egress PMON and sequence generation (FC800\_PMG, SDH\_PMG, FC1200\_PTD, PRBS\_TX), and the egress MLD (STL\_TX, SFIS\_TX, SFI51\_TX).

A packet de-interleaver within the egress de-interleaving, decoding and de-packetization grouping is used to distribute the multi-channel packetized client stream entering the CBRC to either a multi-channel packet monitor or up to 12 FC-1200 transcode decode modules. In the packet monitor path, a multi-channel packet monitor is used to recover client stream rate information as it converts the packet data into CBR streams. This client stream rate information is determined by observing the incoming packet sizes. Transport rate information is received as side channel information along with each channel in this packet stream. The client stream rate information recovered by the packet monitor, along with this transport rate information, is sent to the corresponding ACBs (Adaptive Clocking Blocks). These ACBs are used to recover the clock rate for each client stream. The multi-channel CBR stream, output from the packet monitor, is sent to a CBR de-interleaver where each channel is distributed to a distinct stream interface.

For FC1200, transcode-decoding is required to recover the FC-1200 streams from transcoded ODU2e payloads. The FC1200 client stream is not passed through the packet monitor; the ACBs controlling the clock rate for these streams will be controlled from packet monitors operating on the ODU2e data

within the Core OTN subsystem. The outputs from the CBR de-interleaver and the 12 FC-1200 transcode decode modules are then selectively multiplexed to 12 stream outputs.

The egress PMON and generation grouping is used to provide performance monitoring for each stream, as well as stream substitution for errored streams. For FC1200 the PMON is combined in the transcode-decode block along with errored stream substitution.

The egress MLD grouping is used to encode deskew information and distribute aggregate streams into multiple lanes for egress from the CBRC. Single lane stream formats are also output from the CBRC without modification by the egress MLD grouping. The egress datapath of the CBRC has 12 single lane interfaces and 56 multi-lane interfaces. The 56 multi-lane interfaces are composed of 2 x 17 lanes providing two SFI-5.1 interfaces, 2 x 5 lanes providing two SFI-S( $n=4+1$ ) interfaces, and 3 x 4 lanes providing three STL256.4 interfaces.

In the egress datapath, the CBRC bridges data between the multi-channel packet interface entering the CBRC, the common multi-channel CBR stream clock domain egress from the multi-channel packet monitor and the individual egress stream clock domains for each client.

#### 5.14.2.3 CBR PMON

Performance monitoring is provided transparently by both the ingress and egress PMON and generation groupings. Although no termination is provided for these streams, during various signal fail and performance monitoring alarm events, a replacement alarm indication signal will be generated at a nominal stream rate. FC-1200 64B/66B streams are replaced with local fault sequence ordered sets. The FC-800, CPRI and GDPS 8B/10B streams are replaced with various link fault ordered sets. The OC-768 and OC-192 SONET streams and the STM-256 and STM-64 SDH streams are replaced with PN-11 sequences. For testing purposes, OC-768/STM-256 and OC-192/STM-64 egress streams are able to be generated containing OC-192c/VC-4-64c PRBS payloads. Ingress streams of these types are conversely able to be monitored to detect errors in the overhead or payload.

## 5.14.3 Features

### 5.14.3.1 Interfaces

The following CBR streams are received and transmitted by the CBRC:

- Up to twelve OC-192/STM-64 streams over XFI/SFP+(limiting) interfaces
- Up to twelve FC-1200 streams over XFI/SFP+(limiting) interfaces
- Up to twelve FC-800 streams over XFI/SFP+(limiting) interfaces
- Up to twelve 5G GDPS streams over XFI/SFP+(limiting) interfaces
- Up to twelve 10G GDPS streams over XFI/SFP+(limiting) interfaces
- Up to twelve 4.9G CPRI streams over XFI/SFP+(limiting) interfaces
- Up to twelve 6.1G CPRI streams over XFI/SFP+(limiting) interfaces
- Up to twelve 9.8G CPRI streams over XFI/SFP+(limiting) interfaces
- Up to three OC-768/STM-256 streams over STL256.4 interfaces
- Up to two OC-768/STM-256 streams over SFI-S (n=4+1) interfaces
- Up to two OC-768/STM-256 streams over SFI-5.1 interfaces

With the exception of the streams using the SFI-5.1 interfaces, the interfaces share the same 12 DS1 SERDES interfaces with the interface for connection to the line or system side, configurable in any valid combination. The allowable combinations of these interfaces to be received and transmitted is limited to 120 Gbps of data in each direction. These same SERDES are used by both the LINEOTN and ENET subsystems, placing additional restrictions on available bandwidth. The streams using the SFI-5.1 interfaces share a group of 34 additional line side SERDES. These are not available on the system side. Because SFI-S and SFI-5.1 share internal physical channels, at most, a mixture of two such channels is possible. See [Table 79](#) for details.

### 5.14.3.2 Multi-Lane Distribution (MLD)

On ingress to the CBRC, each CBR stream is sent from the SERDES to the ingress MLD grouping using a DS1 link. Each lane operates in the recovered receive clock domain of its SERDES. Multi-lane stream distributions from the SERDES are processed within the CBRC using receive MLD blocks to deskew and aggregate the lanes. Single-lane streams, although received by the ingress MLD grouping, are processed without the use of MLDs. The receive MLD blocks within the CBRC are the following:

- Three STL256.4 receive MLD blocks for OC-768/STM-256 clients
- Two SFI-S (n=4+1) receive MLD blocks for OC-768/STM-256 clients
- Two SFI-5.1 receive MLD blocks for OC-768/STM-256 clients

Each of these blocks receives multiple lanes of SERDES data, each lane in the receive clock domain of the source SERDES. The mapping of ingress lanes from the CBRC ports to the specific lanes of each MLD is fixed, although mapping flexibility from the source SERDES can be provided by multiplexing outside of the CBRC. In addition to ingress data lanes, the SFI-S and SFI-5.1 MLD blocks receive a DS1 deskew lane. The STL256.4 MLD does not receive a deskew lane, instead using lane alignment markers within the lane data streams to provide the deskew functionality. Each STL256.4 MLD block receives DS1 data from four SERDES. Each SFI-S(n=4+1) MLD block receives DS1 data from five SERDES. Each SFI-5.1 MLD block receives DS1 data from seventeen SERDES. Each receive MLD, deskews, decodes and merges the lanes, keeping the aggregates in one of the recovered clock domains. The output of each receive MLD is also a DS1; the width of this interface is adjusted to the aggregate width of the combined ingress data lanes.

On egress from the CBRC, data is pushed along DSIs from the MLD and multiplexers. Each of these DSIs is synchronous with the transmit clock domain of its target SERDES. Streams to be transmitted in a multi-lane format are divided into lanes by MLD blocks within the CBRC. The transmit MLD blocks within the CBRC are the following:

- Three STL256.4 transmit MLD blocks for OC-768/STM-256 clients
- Two SFI-S (n=4+1) transmit MLD blocks for OC-768/STM-256 clients
- Two SFI-5.1 transmit MLD blocks for OC-768/STM-256 clients

Each of these blocks provides four DSIs all in the transmit clock domain of one of its target SERDES. The mapping of egress lanes from the specific lanes of each MLD to the CBRC ports is fixed, although mapping flexibility to the target SERDES can be provided by multiplexing outside of the CBRC. Depending on the target format, deskew information is added as a separate deskew channel or as in-band lane alignment markers. The STL256.4 transmit MLDs add lane alignment markers as necessary and distribute the data into multiple outgoing DSIs. The SFI-S and SFI-5.1 transmit MLDs create a deskew lane in addition to distributing the egress data stream into multiple outgoing DSIs. Each STL256.4 MLD block transmits DSIs to four SERDES. Each SFI-S(n=4+1) MLD block transmits DSIs to five SERDES. Each SFI-5.1 MLD block transmits DSIs to seventeen SERDES.

### 5.14.3.3 Performance Monitoring, Alarm Indication Stream Generation and PRBS

#### 5.14.3.3.1 SONET/SDH Performance Monitoring

In both the ingress and egress datapaths, each of the CBR streams is performance monitored. Section and line level performance monitoring is provided for up to 12 SONET/SDH OC-192/STM-64 streams and up to 3 OC-768/STM-256 streams in each direction. For the SONET/SDH streams, performance monitoring includes the following:

- Detect optical LOS (ingress streams only).
- Detects OOF, LOF and Signal Fail.
- Counts B1 BIP-8 bit or block errors.
- Counts B2 BIP-8 bit or block errors.
- Detects AIS-L and RDI-L.
- Detects Generic AIS (PN-11) pattern. Note detection of Generic AIS will also result in other defects (LOF) being detected due to invalid SONET/SDH framing.
- Counts REI-L block or bit errors.
- Extracts J0 and monitors the 1/16/64 byte trace message.
- Indicates line dDEG and dEXC. Optionally this can be configured to indicate section dDEG and dEXC, though not both line and section simultaneously. Poisson mode calculation is provided by the CBRC; Bursty mode calculations can be done in software using B1/B2 counts provided by the CBRC.
- Detects section trail trace TIM and TIU.
- Extracts APS and detects APSBF.
- Extracts SSM from S1.
- Multiple descrambling options to enable PMON.

#### 5.14.3.3.2 SONET/SDH Alarm Indication Stream Generation

For SONET/SDH, the replacement signal is a PN-11 stream at either the SONET/SDH OC-192/STM-64 or OC-768/STM-256 rate as required.

For the ingress direction, the PN-11 stream is generated using a user programmable delta-sigma modulator to generate data gapped at a nominal 311.04MHz rate.

For the egress direction, the PN-11 stream is generated using the SERDES-generated transmit clock associated with the stream. Note that in the case of failures in the demapped OTN stream (e.g. SSF) leading to CBR FIFO Underflow, the TX SERDES clock will not necessarily revert to a nominal rate. Software intervention on the clock controller may be required if the replacement stream is to be inserted at the nominal rate.

Though not the standard mode of operation, the replacement signal can also be an MS-AIS signal rather than PN-11.

All signal replacement conditions can be masked by a configurable bit corresponding to the condition. In addition the replacement can be forced to provide the replacement stream by programming a software configurable bit.

In addition to any pattern overwrite, the ingress CBRC can set a Signal Fail (SF) sideband indication in response to the LOS and LOF conditions. The SF signal can then be used downstream by an AMP/BMP mapper to assert the OPUk.CSF bit in the PSI byte.

**Table 74 SONET/SDH Ingress Client Stream Consequential Actions**

Ingress CBR Stream	Fail Condition	Replacement Stream
STS-192/STM-64	Optical Module LOS	PN-11 or MS-AIS
STS-768/STM-256	Optical Module LOS	PN-11 or MS-AIS
STS-192/STM-64	Loss Of Frame	PN-11 or MS-AIS
STS-768/STM-256	Loss Of Frame	PN-11 or MS_AIS

**Table 75 SONET/SDH Egress Client Stream Consequential Actions**

Egress CBR Stream	Fail Condition	Replacement Stream
STS-192/STM-64	CBR FIFO Underflow	PN-11 or MS-AIS
STS-768/STM-256	CBR FIFO Underflow	PN-11 or MS-AIS
STS-192/STM-64	Loss of Frame	PN-11 or MS-AIS
STS-768/STM-256	Loss of Frame	PN-11 or MS-AIS

#### 5.14.3.3.3 SONET/SDH PRBS

The PRBS monitor uses the SDHPMG in combination with the PRBS\_MG to provide monitoring for PRBS errors of a single OC-192/STM-64 framed STS-192c/VC-4-64c PRBS stream or one of the four STS-192c/VC-4-64c PRBS streams framed within each OC\_768/STM-256. The monitor rate is controlled by the RX SERDES recovered clock of the selected stream

The PRBS generator generates a PRBS payload contained in a single STS-192c/VC-4-64c which is framed as independent OC-192/STM-64 or copied four times into a single OC-768/STM256. This datastream can be transmitted in any one of the twelve 32-bit DS1 lanes in the OC-192 mode or in any one of the three 128-bit DSIs in OC-768 mode. The rate is controlled by the TX SERDES recovered clock rate of the selected stream.

#### 5.14.3.3.4 8B/10B and Generic Client Performance Monitoring

##### 5.14.3.3.4.1 Fibre Channel 800

FC800\_PMG can be used to align to Fibre Channel 800 streams. An FC800\_PMG exists in each of the ingress and egress directions. When alignment is achieved, passive monitoring of the Fibre Channel 800 stream is performed. It supports detection of:

- 8B/10B code violations, including disparity errors
- Frame boundaries via SOF ordered set detection
- FCS violations, including FCS checking over scrambled payloads
- Incoming signal transitions

FC800\_PMG also supports real-time collection of the following statistics for Fibre Channel 800 streams:

- Number of 8B/10B code violations
- Number of FCS errors
- Number of received/transmitted words
- Number of received/transmitted frames

FC800\_PMG generates a digital loss-of-link for Fibre Channel 800 streams under some or all of the following conditions:

- Optical loss-of-signal occurred
- Egress FIFO underflow occurred
- Number of 8B/10B code violations exceeds a user-defined threshold
- Duration of lack-of-transition on incoming signal exceeds a user-defined amount of time
- Duration of loss-of-synchronization exceeds a user-defined amount of time

##### 5.14.3.3.4.2 Generic 8B/10B Streams (CPRI / InfiniBand)

FC800\_PMG treats CPRI 4.9G/6.1G/9.8G and InfiniBand (GDPS) 5G/10G as generic 8B/10B-encoded streams. As such, only a limited subset of monitoring is available. Note that performance monitoring of 8B/10B clients is only operative at the configured rate. For generic 8B/10B encoded streams, FC800\_PMG supports detection of:

- 8B/10B code violations, including disparity errors
- Incoming signal transitions

FC800\_PMG supports real-time collection of the following statistics for generic 8B/10B streams:

- Number of 8B/10B code violations
- Number of received/transmitted words

FC800\_PMG can generate a digital loss-of-link for generic 8B/10B streams under some or all of the following conditions:

- Optical loss-of-signal occurred
- Egress FIFO underflow occurred
- Number of 8B/10B code violations exceeds a user-defined threshold
- Duration of lack-of-transition on incoming signal exceeds a user-defined amount of time

### 5.14.3.3.5 8B/10B Error Stream Generation

For FC-800, the replacement signal is a NOS ordered set. For CPRI and InfiniBand (GDPS), the replacement signal are Ethernet local fault ordered sets.

For the Ingress direction, the replacement stream generator can be configured to generate the stream at a nominal rate on the system clock, gapped by a first-order delta-sigma modulator based on a user-programmable rate value.

For the Egress direction, the replacement signal's rate is determined by the transmit SERDES clock. Note that in the case of failures in the demapped OTN stream (e.g. SSF) leading to CBR FIFO Underflow, the TX SERDES clock will not necessarily revert to a nominal rate. Software intervention on the clock controller may be required if the replacement stream is to be inserted at the nominal rate.

All signal replacement conditions can be masked by a configurable bit corresponding to the condition. In addition the replacement can be forced to provide the replacement stream by programming a software-configurable bit.

Beyond the standard fault replacement, any repeating pattern of up to 160 bits can be inserted for 8B/10B clients. As such, Idle characters, Error characters or other patterns can be inserted instead of the standard NOS or Local Fault.

In addition to any pattern overwrite, the ingress CBRC can set a Signal Fail (SF) sideband indication in response to any defects. The SF signal can then be used downstream by an AMP/BMP mapper to assert the OPUk.CSF bit in the PSI byte.

**Table 76 8B/10B Client Stream Consequential Actions**

Client CBR Stream	Signal Fail Condition	Replacement Stream
FC-800	Optical Module LOS (ingress only), CBR FIFO underflow (egress only), Loss of 8B10B Sync, Line Code violations (w/ threshold), Transition detect (w/ threshold)	/K28.5/D21.2/D31.5/D5.2/ (NOS)
5G IB	Optical Module LOS (ingress only), CBR FIFO underflow (egress only), Loss of 8B10B Sync, Line Code violations (w/ threshold), Transition detect (w/ threshold)	/K28.5/D21.5/D0.0/D0.0/K28.5/D2.2/D0.0/D0.0/ (Local Fault, see note)
10G IB	Optical Module LOS (ingress only), CBR FIFO underflow (egress only), Loss of 8B10B Sync, Line Code violations (w/ threshold), Transition detect (w/ threshold)	/K28.5/D21.5/D0.0/D0.0/K28.5/D2.2/D0.0/D0.0/ (Local Fault, see note)
4.9G CPRI	Optical Module LOS (ingress only), CBR FIFO underflow (egress only), Loss of 8B10B Sync, Line Code violations (w/ threshold), Transition detect (w/ threshold)	/K28.5/D21.5/D0.0/D0.0/K28.5/D2.2/D0.0/D0.0/ (Local Fault)
6.1G CPRI	Optical Module LOS (ingress only), CBR FIFO underflow (egress only), Loss of 8B10B Sync, Line Code violations (w/ threshold), Transition detect (w/ threshold)	/K28.5/D21.5/D0.0/D0.0/K28.5/D2.2/D0.0/D0.0/ (Local Fault)
9.8G CPRI	Optical Module LOS (ingress only), CBR FIFO underflow (egress only), Loss of 8B10B Sync, Line Code violations (w/ threshold), Transition detect (w/ threshold)	/K28.5/D21.5/D0.0/D0.0/K28.5/D2.2/D0.0/D0.0/ (Local Fault)

**Note:** The 5G/10G IB replacement stream is not yet formally defined by a standard. The replacement used is an Ethernet local fault signal.

### 5.14.3.3.6 64B/66B Performance Monitoring

#### 5.14.3.3.6.1 FC-1200 Ingress Performance Monitoring

##### Receive PCS

The receive PCS is used to align to the incoming 64B/66B Fibre Channel stream. When alignment is achieved, the PCS will also perform descrambling of the incoming 64B/66B blocks and forward the 64B stream (without the 2-bit sync headers) to the transcode encoder block and the performance monitoring logic. The PCS will also detect and signal the following defects:

- High-BER state of incoming stream
- Loss-of-synchronization
- Loss-of-signal

The PCS will monitor the stream of blocks and count the number of:

- Blocks with invalid block type fields
- Blocks with invalid sync header fields
- Valid data bytes
- Valid ordered set bytes
- PRBS31 bit errors detected (when enabled)

Note the PRBS-31 monitor for FC-1200 clients within the CBRC subsystem will lock to an all-ones pattern. Use the PRBS-31 monitor in the Ethernet Line or Ethernet System subsystems.

##### Ingress Performance Monitoring

The performance monitoring logic receives the aligned and unscrambled 64B block data from the PCS, along with the block and link error indications generated by the PCS. In addition, the performance monitoring logic will be able to detect the following frame errors:

- Invalid frame delimiters
- Frame Check Sequence (FCS) violations

Using these error indicators, the performance monitoring logic will also count the number of:

- Valid frames
- Invalid frames (frames with invalid frame check sequences or invalid frame delimiters)
- Valid bytes in valid frames (including data bytes and ordered set bytes)

The performance monitoring logic will generate a digital loss-of-link output under the following conditions, which can also be individually masked off by the user:

- High-BER state was detected
- Loss-of-synchronization was detected.
- Optical loss-of-signal was detected.

### 5.14.3.3.6.2 FC-1200 Egress Performance Monitoring

#### Transmit PCS

The FC1200\_PTD Transmit PCS block is used to prepare the egress Fibre Channel 1200 stream for transmission. The transmit PCS scrambles the 64B data blocks from the FC\_TTDD transcode decoder block, and prepends the appropriate sync header to the block to create the 64B/66B encoded block. Finally, the transmit PCS implements rate adaptation to convert the stream of 66-bit blocks to a stream of 32-bit words, which are then transmitted.

The PCS will monitor the stream of blocks and count the number of:

- Total data bytes transmitted
- Total ordered sets transmitted
- Number of local fault ordered sets detected
- Number of error control blocks detected

The PCS will also detect and optionally raise an alarm if the local fault signal is detected on the incoming stream.

#### Egress Performance Monitoring

The performance monitoring logic receives 64B block data from the transcode decoder, and is able to detect the following frame errors:

- Invalid frame delimiters
- Frame check sequence (FCS) violations

Using these error indicators, the performance monitoring logic will also count the number of:

- Valid frames
- Invalid frames (frames with invalid frame check sequences or invalid frame delimiters)
- Valid bytes in valid frames (including data bytes and ordered set bytes)

### 5.14.3.3.7 64B/66B Error Stream Generation

#### 5.14.3.3.7.1 FC-1200 Ingress Error Stream Generation

For FC-1200, the replacement signal is a local fault sequence ordered set generated at the nominal FC-1200 rate. The local fault sequence ordered set is generated by gapping the fault sequence stream data at a programmable rate to generate a nominal rate stream.

All signal replacement conditions can be masked by a configurable bit corresponding to the condition. In addition the replacement can be forced to provide the replacement stream by programming a software configurable bit.

For non-transcoded paths (mapping to ODU1f or ODU2f), replacement by a scrambled 66B pattern (typically a Local Fault pattern) is possible in response to Optical LOS. Stream replacement via software-configurable bit is also supported.

Table 77 FC-1200 Ingress Client Stream Consequential Actions

Ingress CBR Stream	Signal Fail Condition	Replacement Characters
FC-1200 (transcoded - towards ODU2e)	Loss of Sync, High-BER, Optical LOS, Block Error Density Exceeded	64B/66B Local Fault Ordered Set

Ingress CBR Stream	Signal Fail Condition	Replacement Characters
FC-1200 (transcoded - towards ODU2e)	Invalid Block Received	Error Control Block
FC-1200 (non-transcoded - ODU1f/2f)	Optical LOS	64B/66B Local Fault Ordered Set

#### 5.14.3.3.7.2 FC-1200 Egress Error Stream Generation

For FC-1200, the replacement signal for a client signal failure is the local fault ordered set sequence. Individual block errors that result from decoding issues result in the replacement of the block with either an Error Control Block, or an Idle Block, as configured by the user.

The replacement signal's rate is determined by the transmit SERDES clock. Note that in the case of failures in the demapped OTN stream (e.g. SSF) leading to CBR FIFO Underflow, the TX SERDES clock will not necessarily revert to a nominal rate. Software intervention on the clock controller may be required if the replacement stream is to be inserted at the nominal rate.

All signal replacement conditions can be masked by a configurable bit corresponding to the condition. In addition the replacement can be forced to provide the replacement stream by programming a software configurable bit.

For non-transcoded paths (de-mapping from ODU1f or ODU2f), replacement by a scrambled 66B pattern (typically a Local Fault pattern) is possible in response to CBR FIFO underflow. Stream replacement via software-configurable bit is also supported.

**Table 78 FC-1200 Egress Client Stream Consequential Actions**

Egress CBR Stream	Error Condition	Replacement Stream
FC-1200 (transdecoded - from ODU2e)	FIFO Underflow due to lack of incoming data (Client failure)	64B/66B Local Fault Ordered Set
FC-1200 (transdecoded - from ODU2e)	'Error Control Block or un-decodable block'	Error Control Block
FC-1200 (non-transdecoded - from ODU1f/2f)	CBR FIFO underflow	64B/66B Local Fault Ordered Set

#### 5.14.3.4 64B/66B (FC1200) Transcode Encoding

12 single lane DSI streams sourced from the SERDES are pushed to the FC-1200 transcode encode blocks, each allowing a 64B/66B FC-1200 stream to be transcoded as defined in G.709 clause 17.8.2. This transcode encoding is required to adapt the line rate of an FC-1200 stream for transport as an ODU2e payload. Each transcode encode block produces a GFP-F Frame containing the aggregated superblock, frame header and FCS.

The CBRC subsystem schedules data from each FC-1200 encode block at an overspeed rate. In conjunction with the CPB, this ensures that the transcoded FC-1200 data is provided to the OTN mapper with latency controlled to ensure that no GFP idle frames will be required; this allows an FC-1200 stream to be BMP mapped into an ODU2e.

Note: Ethernet (10GE) transcoding for transport as an OPU2e payload is provided by the ENET subsystem.

### 5.14.3.5 64B/66B (FC1200) Transcode Decoding

The transcode decoding required to return a stream adapted for transport as an ODU2e payload back into an FC-1200 stream is provided by the CBRC. Transcode decoding is provided for up to 12 simultaneous FC-1200 streams. These decoders provide the transcode decode as defined in G.709, returning the transcoded blocks to the FC-1200 64b/66B PCS stream. Each decode block receives packetized 384 bit data along a DPI sourced from the 12 port packet deinterleaver. The egress interface from each block is a 32 bit DS1 sourcing decoded CBR data. Since the GFP-F within the ODU2e contained no idle frames, the decoded FC-1200 PCS can be considered to have been BMP demapped from an ODU2e.

Note: Ethernet (10GE) transcoding to extract the stream from an OPU2e payload is provided by the ENET subsystem.

### 5.14.3.6 Ingress Client Rate Capture

Within the CBRC, for all clients except FC1200, client rate capture and recovery is performed using the OPSA by the MPMA, multi-channel packet monitor and ACB blocks.

For clients ingress to the CPB, the MPMA is used to packetize each ingress CBR stream. The size of each packet made is an encoding of the arrival rate of each stream's data. Packet size decisions are made at a period of Tb as +/-1 byte size modulations to a nominal size of C. These size encodings will later be used to control the justification of client data as it mapped into ODUk or ODUflex payloads.

In order to be transported within an ODU2e payload, FC-1200 PCS is transcode encoded to reduce the bitrate. This encoding process produces 8800 byte packets which are subsequently GFP-F mapped for transport within an ODU2e without GFP-F idles being introduced into the stream. Since the FC-1200 PCS is essentially BMP mapped into the ODU2e, the ODU2e rate is a fixed ratio of the FC-1200 signalling rate.

### 5.14.3.7 Egress Client Rate Capture

Within the CBRC, for all clients except FC1200, client rate capture and recovery is performed using the OPSA by the MPMA, MPMO, and ACB blocks.

For clients egress from the CPB, the multi-channel packet monitor is generally used to remove the client packet boundaries of each stream's packets, restoring each client to a CBR stream. The packet sizes are used to create a size modulation stream which is sent to the ACB controlling the transmit clock of the target SERDES. As each channel's data egress from the CPB is being sent to a distinct target SERDES, the packet monitor communicates with multiple ACBs, each dedicated to a single channel of the channelized data flow. In addition to the modulation streams created by the packet monitor, a side channel modulation stream accompanying each client is sent to each ACB. These modulation streams are a rate encoding created outside of the CBRC for the OTN transport that originally carried each client. When controlling rate for multi-lane formats, the ACBs controlling the CSU within each SERDES and some cases multiple SERDES are controlled by identical modulation streams. In these cases, the CSU bitrate scaling is adjusted to account for the multi-lane distribution.

For client rate recovery in the egress direction, the ACB is accumulating phase at a nominal rate representing the expected rate of client data. This phase accumulation is used to control a CSU, which generates the SERDES transmit clock. This accumulation of phase is adjusted by the two modulation streams. The first stream represents the variance of the client rate from a nominal client rate relative to a nominal transport. It is used to adjust the amount of phase accumulated in a CPB packet period. This stream is produced by the multi-channel packet monitor as it receives the channelized egress stream. The second stream represents the variance of the transport rate from the nominal transport rate. It is used to adjust the period of the phase accumulation in the ACB, correcting it to represent

the actual period of client data arriving given the captured transport rate. These two modulation streams correct the accumulation of phase in the ACB such that it represents the actual arrival rate of the client after demapping from the transport.

When the encoded FC-1200 payload is demapped from the ODU2e and passed through the CPB to the CBRC for trancode decoding, it is not distributed within packets sized according to rate. Each packet is the 8800 byte GFP-F frame payload that was originally created by the transcode encode process. These packets are therefore not able to be used by a packet monitor to create an ACB modulation stream for recovery of the FC-1200 client rate. However, since the GFP-F payload, and by extension, the transcode decoded FC-1200 PCS are BMP demapped from the ODU2e, a packet monitor between the ODUK switch and the GFP demapper is used to create an ACB modulation stream from the ODU2e stream. The ODU2e stream was packetized to capture rate by a packet monitor on the ingress to the ODUK switch. Therefore, for FC-1200 transcode decoded streams processed by the CBRC, the ACB controlling the transmit clock of the target SERDES and recovering the original FC-1200 client rate is controlled from a packet monitor upstream of the CBRC.

#### 5.14.3.8 Channelization

The organization of the DS1 links is shown in [Table 79](#) below.

The mapping from DPI channel ID to internal DS1 channel is programmable in the CBR and packet de-interleaver with each DS1 channel programmable in the DPI ID range 0 to 11.

**Table 79** CBRC Internal DS1 to SERDES DS1 Channel Mappings

SERDES DS1 Channel	Internal DS1 Channel				
	10G_XFI (CPRI / GDPS / FC800 / OC-192 / STM-64)	10G_XFI (FC1200)	STL256.4 (OC-768/ STM-256)	SFI-S (OC-768 / STM-256)	SFI-5.1 (OC-768 / STM-256)
EXT_10G[0]	0	0			
EXT_10G[1]	1	1			
EXT_10G[2]	2	2			
EXT_10G[3]	3	3			
EXT_10G[4]	4	4			
EXT_10G[5]	5	5			
EXT_10G[6]	6	6			
EXT_10G[7]	7	7			
EXT_10G[8]	8	8			
EXT_10G[9]	9	9			
EXT_10G[10]	10	10			
EXT_10G[11]	11	11			
EXT_2_5G[0:16] (Line only)					0:3
EXT_2_5G[17:33] (Line only)					4:7

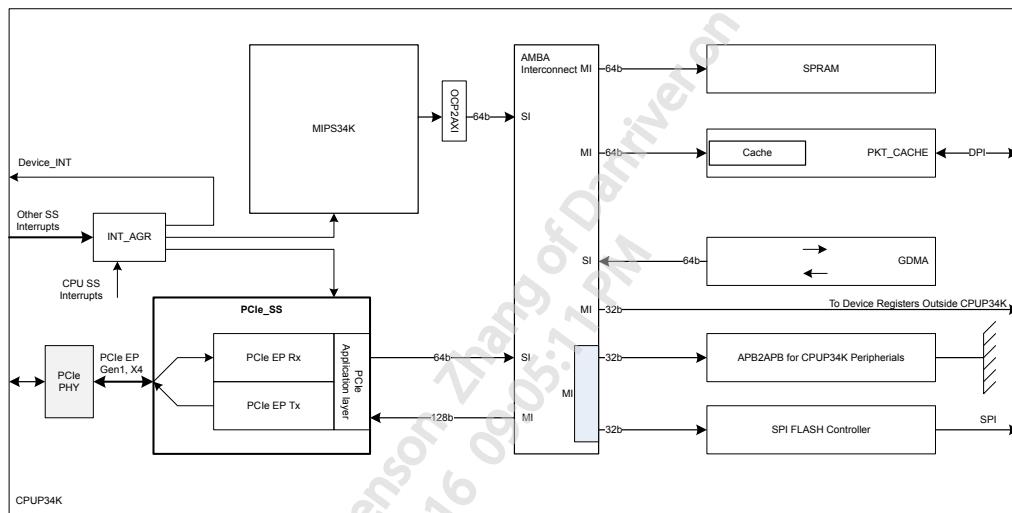
Note: Due to the overlap in physical channel mapping between SFI-S and SFI5.1, there are limitations on the combinations possible. In particular, it is not possible to configure 2 x SFI5.1 with 1 x SFI-S, or 1 x SFI5.1 with 2 x SFI-S even though there are no SERDES conflicts in doing so.

## 5.15 CPU (CPUP34K) Subsystem

### 5.15.1 Block Diagram

The CPU subsystem provides an internal MIPS processor with its local support peripherals and a PCIe interface for the host to access the device subsystems through a bus interconnect. The firmware code executed by the MIPS processor performs various device application functions where the PCIe host software communicates with internal MIPS firmware using interrupts and messages.

Figure 157 CPUP34K Block Diagram



**Note:**

- MI is Master Interface driven by interconnect towards a slave
- SI is Slave Interface driven into interconnect by a master
- Connection arrowhead shown from master to slave

### 5.15.2 Overview

The CPUP34K is a multi-master subsystem with an embedded MIPS34K RISC processor, a PCIe EP (End Point) and generic DMA. The MIPS34Kc processor has 64 KB of instruction cache, 32 KB of data cache, 768 KB of internal scratch RAM and external serial boot flash memory. The MIPS processor's local peripherals include:

- 768 KB SPRAM
- SPI (Serial Peripheral Interface) controller
- Interrupt aggregators
- Watchdog timers
- Software timers
- Packet cache
- A generic DMA engine

The PCIe interface is Gen-1 and supports x1, x2 and x4 link widths at 2.5 Gbps rates per lane.

The CPUP34K executes the device firmware. The device does not support development or execution of user software. The MIPS processor in the CPUP34K boots from an externally-connected SPI flash memory device through a memory-mapped SPI controller. The external SPI flash memory also stores the PCIe initialization code, which is executed by a MIPS boot initialization routine.

External host downloads firmware to SPRAM and then instructs the CPU to begin executing the firmware. Through the PCIe interface, the external host is able to communicate with firmware at run-time by passing messages via a shared internal RAM.

The AMBA interconnect allows both a PCIe master and the internal MIPS processor to drive the SPI master interface. The AMBA interconnect provides dedicated configuration ports to access the CPUP34K subsystem peripheral registers and device subsystem registers. The AMBA network interconnect switch allows multiple masters to simultaneously access multiple slave configuration and status registers.

Access to the device registers including firmware messages, firmware doorbells, interrupts, and DMA messages are big endian.

Downloaded [controlled] by Benson Zhang of Danriver  
Thursday, 21 April, 2016 09:05:11 PM

## 5.15.3 Features

### 5.15.3.1 Hardware Features

The CPUP34K subsystem includes:

- A MIPS34Kc processor with two VPEs
- A PCIe Gen1 interface with configurable x1, x2 and x4 link widths for the external host
- A High-speed AMBA network interconnect for multi-master and multi-slave inter-working
- A memory-mapped SPI to connect to external SPI flash memory for MIPS boot code
- A PCIe system footprint of 32 MB
- Run-time messaging between masters via a shared internal SPRAM
- A software-driven SPI master interface for an external SPI-based GPIO expander
- Highly-flexible interrupt aggregation to an external host
- Hardware and firmware-based software interrupt handling for quick interrupt resolution and servicing
- 768 KB of scratch pad RAM for internal firmware program and data
- A GDMA (General Purpose Master DMA) engine to offload internal firmware from data intensive tasks
- Packet cache and cache management for fast access to packet data by internal firmware running on the MIPS processor
- A dedicated peripheral configuration bus to access device registers

### 5.15.3.2 PCIe Features

The PCIe interface in CPUP34K subsystem:

- Complies with the relevant sub-clauses of the PCI Express Base Specification 1.1 (Gen 1)
- Can be implemented as a PCIe end point
- Supports x1, x2 and x4 links running 2.5 Gbps
- Supports a physical interface running 1 symbol wide
- Supports one virtual channel (VC)
- Provides an AXI master interface for inbound PCIe requests
- Provides an AXI slave interface for outbound PCIe requests
- Observes PCIe ordering rules in both inbound and outbound directions
- Supports a maximum payload size up to 256 bytes
- Supports Advanced Error Reporting (AER)
- Supports Device Power Management D0 and D3 states.
- Supports Link Power Management L0 and L0s states only.
- Supports generation of an MSI/MSI-X/INITx interrupt message

### 5.15.3.3 34K Core

The 34K core implements the MIPS32 Release 2 architecture in a 9-stage pipeline. It also includes support for the MIPS16e™ compressed instruction set for minimizing code space. The core has two VPEs (Virtual Processing Elements) and four TCs (Thread Contexts). The TCs offer lightweight multi-threading, which allows cooperative or independent threads to run concurrently. The TCs may be flexibly assigned by firmware to either VPE. The 34K core also includes a high performance MDU

(Multiply Divide Unit), which is fully pipelined in order to support a single cycle repeat rate for 32x32 MAC instructions.

The instruction and data L1 (Level One) caches are up to 64 KB and 32 KB respectively in size. Each cache is organized as 4-way set associative. Data cache misses are non-blocking and up to four may be outstanding. Two instruction cache misses can be outstanding. Both caches are virtually indexed and physically tagged to allow them to be accessed in the same cycle that the address is translated. The BIU (Bus Interface Unit) implements an OCP (Open Core Protocol) interface with 32-bit addresses and 64-bit data.

#### 5.15.3.4 CPU Packet Cache

The CPU packet cache is a peripheral in CPUP34K that provides a dedicated packet interface and caching of packets in order to provide a low latency path for CPU-based packet processing. It takes packets from the packet interface and caches these packets for firmware to process. Firmware implements protocol-based algorithms and modifies the packet content by doing random read-write accesses into the packet cache. Modified packets are either consumed by firmware or sent back on the packet interface to an external packet buffer. The packet interface carries packet traffic in a channelized form where each channel may carry packets of variable length and type. The packet cache does packet processing on a per channels basis. Firmware implements algorithms for application specific packets generation and process received packets in packet cache based on type. The packet cache uses back-pressure in circumstances where the CPU is overloaded and the packet cache cannot accommodate more packet data.

#### 5.15.3.5 GDMA

The GDMA is a generic DMA engine configured to offload the CPU from frequent and simple bulk data transfers to/from any memory-mapped location. The GDMA has an AXI master interface that connects with an AMBA interconnect slave interface. The GDMA provides an AXI master interface to perform the DMA transfers and APB slave interfaces that control its operation. The GDMA is a local resource to firmware running on the MIPS processor and is not directly accessible to the external host.

#### 5.15.3.6 Memory Mapped SPI Flash Controller

The SPI implements SPI master logic such that the external SPI boot flash memory memory can be connected externally to the device. The SPI flash memory stores the boot initialization code to be executed by the processor at power up. Four SPI devices can be connected externally where one of the SPI slaves must be boot flash memory at the 0th chip select. Another SPI slave can be a GPIO expander that accepts a link status report from the processor or the PCIe. It is a SPI slave that is responsible for using the activity reports for enabling LEDs and blinking them as desired.

The memory mapped SPI Flash controller supports SPI mode 0 in which the SPI data input is sampled on the rising edge of the SPI clock while the SPI data output toggles on the falling edge of SPI clock. The SPI master core implements a direct memory-mapped read function. The SPI master makes the serial SPI flash memory device look like a parallel 32-bit wide ROM connected to the AXI bus. This feature allows the SPI flash memory to be used as an external boot ROM whereby the MIPS boot vector is mapped onto the memory-mapped SPI controller. At power-up the memory mapped SPI Flash controller flash interface inter-operates at 1 MHz with a SPI flash memory device supporting 24-bit (3 bytes) address information.

### 5.15.3.7 Special Considerations for Programming and Accessing of the Flash ROM

In addition to accessing the flash at power-up, the device will access the boot flash memory under the following scenarios:

- A hard reset of the device. This is equivalent to loading the initialization code from flash upon power-up.
- Soft Reset of the MIPS. A soft reset is done whenever firmware needs to be reloaded either directly or when initiated from the 'digi\_handle\_init()' software API call.
- A watchdog timeout occurs. A timeout will occur due to an unexpected error or exception in firmware.
- When firmware requires data synchronization to enforce the order of operations between two tasks, and as a result, the processor SYNC instruction is executed.

Host software or hardware can typically control when events 1 and 2 occur. Events 3 and 4 can occur at any time. Therefore, it is necessary for the SPI flash ROM to remain connected to the device at all times. The device will lockup if any of the events in the description occur and the boot flash ROM is not available on the SPI bus. Therefore, ensure the flash boot device is available and connected to the device during both power-up and during normal operation.

### 5.15.3.8 Firmware Upgrade Without Resetting the Device

The CPUP34K block provides a separate reset control such that the MIPS core within the device can be held at a reset state during a firmware upgrade without the need to reset the entire device. The detailed procedure for upgrading firmware can be found in the DIGI Family Programmer's Manual [\[4\]](#).

## 5.15.4 MIPS34K Processor

The 34K core operates with a 600 MHz core clock. The processor bus interface unit operates at 300 MHz using the direct strobe method. The MIPS processor supports out-of-order response using tags. It also supports six outstanding reads and unlimited outstanding writes. A reference COP2 module is provided to support 64-bit load and store instructions, improving bus efficiency. An external policy manager implements a round-robin policy for each TC to be used by the TSU (Thread Scheduling Unit).

An XYM (External Yield Manager) is implemented to support yields of type:

- Yield 0: terminate thread and make TCs available for subsequent forks
- Yield -1: pause currently running thread, will be resumed by a hardware TSU
- Yield -2: poll event pins

All RAM instances having parity and parity error exceptions are handled by the core internally. MIPS34Kc interrupts are wired to support both interrupt compatibility and VI (Vectored Interrupt) mode. The core powers up in interrupt compatibility mode at reset. During boot initialization, the core switches to VI mode as part of the interrupt handler initialization routine.

The processor is constrained to only generate simple byte enables, that is, only naturally-aligned bytes, halfwords, words, and double-words.

### 5.15.4.1 MIPS Boot

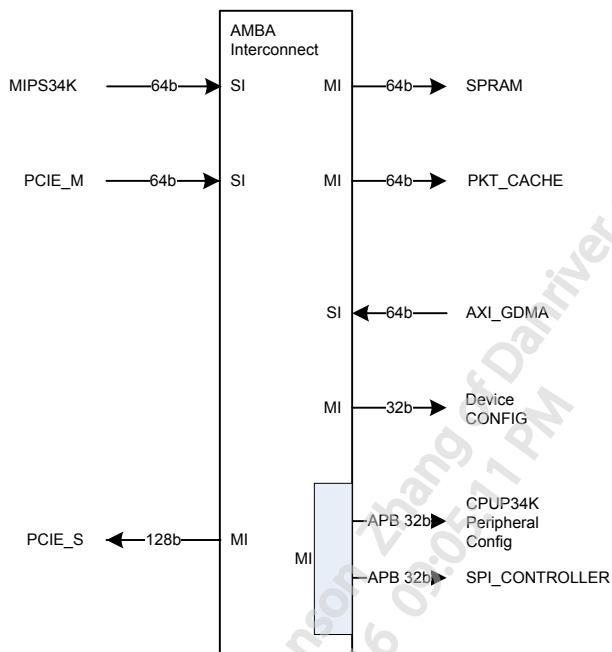
During power-up, the boot sequence is:

1. CPU boot initialization takes place from the memory mapped SPI Flash
2. The CPU initializes the PCIe and then notifies the PCIe host that the CPUP34K is ready
3. The PCIe host writes the firmware application code to the SPRAM
4. The PCIe host notifies the boot-loader in flash memory to execute the firmware application code from the SPRAM

## 5.15.5 AMBA Network Interconnect

### 5.15.5.1 Block Diagram

Figure 158 CPUP34K AMBA Network Interconnect Block Diagram



### 5.15.5.2 Network Interconnect Overview

The AMBA-based CPUP34K network interconnect provides:

- Two 64-bit AXI slave interfaces
- One 32-bit AXI slave interface
- Two 64-bit AXI master interface
- One 128-bit AXI master interface
- One 32-bit AXI master interface
- Two 32-bit APB-3 master interface

### 5.15.5.3 Address Allocation to Each Slave

Table 80 Address View Via AMBA Interconnect

Start	End	Size [MB]	Cumulative Size [MB]	Addressed Slave	PCIe Accessible Range (32MB)
0X2000_0000	0xFFFF_FFFF	-	4000	Unmapped	No
0x1C00_0000	0x1FFF_FFFF	64MB	512	Reserved	No
0x1BF0_0000	0x1BFF_FFFF	1MB	448	Reserved	Yes
0x1BE0_0000	0x1BEF_FFFF	1MB	447	PKT_CACHE	Yes

Start	End	Size [MB]	Cumulative Size [MB]	Addressed Slave	PCIe Accessible Range (32MB)
0x1BD0_0000	0x1BDF_FFFF	1MB	446	SPRAM	Yes
0x1BC0_0000	0x1BCF_FFFF	1MB	445	SPI_CONTROL-LER	Yes
0x1BA0_0000	0x1BBF_FFFF	2MB	444	Reserved	Yes
0x1B80_0000	0x1B9F_FFFF	2MB	442	CPUP34K peripheral configuration	Yes
0x1A00_0000	0x1B7F_FFFF	24MB	440	Device configuration (Minus CPUP34K peripherals)	Yes
0x1800_0000	0x19FF_FFFF	32MB	416	Reserved	No
0x1000_0000	0x17FF_FFFF	128MB	384	PCIe_S	No
0x0000_0000	0x0FFF_FFFF	256MB	256	Reserved	No

The device memory map is designed to have a 32 MB footprint for the PCIe EP in normal application mode. This aliased range provides the ability to access the CPU's firmware boot location within the 32 MB PCIe EP memory footprint.

**Table 81 Permanent Address Aliasing**

Address Range Alias	Physical Address Range	Addressed Slave
0x1BC0_0000 to 0x1BCF_FFFF	0x1FC0_0000 to 0x1FCF_FFFF	Memory Mapped SPI Flash Controller

#### 5.15.5.4 Configuration Bus Lock-up

The device configuration bus can become unresponsive in two situations:

- When the configuration bit S16\_LINE\_RESET[A] is set to '1' and a configuration transaction is destined to registers within the S16\_LINE[A] address space, the configuration access will not return and the configuration bus will become locked. Similarly, the S16\_SYS\_RESET[A] and S16\_SYS[A] exhibit the same behavior. To avoid this problem, do not permanently set the S16\_\*\_RESET to '1', but instead toggle it from '1' to '0' to reset the associated S16 SERDES lane and then set GLBL\_RSTB = '1' and CLK\_DEGLITCHER\_EN = '1' within the associated S16 SERDES address space.
- When the configuration bit ENET120\_LINE\_RESET is set to '1' and a configuration transaction is destined to registers within the ENET120\_LINE:EMPEI120\_I:ECLASS or ENET120\_LINE:EMPEI120\_E:ECLASS, the configuration access will not return and the configuration bus will become locked. Similarly, the ENET120\_SYS\_RESET and ENET120\_SYS:EMPEI120\_I:ECLASS and ENET120\_SYS:EMPEI120\_E:ECLASS exhibit the same behavior. To avoid this problem, do not permanently set the ENET120\_\*\_RESET to '1', but instead toggle it from '1' to '0' to reset the associated Ethernet subsystem and then set EMPEI120\_I\_LOWPWR = '0' and EMPEI120\_E\_LOWPWR = '0' within the associated Ethernet subsystem address space.

In both situations, the limitations do not exist immediately after device power-on reset.

## 5.15.6 Interrupt Aggregation

The interrupts from various sources inside and outside the CPU subsystem are aggregated to a few interrupt lines towards the internal MIPS CPU, PCIe host and device pin. The interrupt aggregation is a filtering process where the master is provided with the capability to choose which interrupt is allowed to interrupt itself.

Table 82 CPUP34K's External Interrupt Destination

Interrupt Signal Names	Designated Interrupt Destination
SS_INT_0[9:0]	For internal MIPS CPU usage
SS_INT_1[9:0]	For PCIe MSI/MSIx interrupts
SS_INT_2[9:0]	For device Interrupt pin output.
SS_INT_3[9:0]	For internal MIPS CPU usage

Table 83 CPUP34K's External Interrupt Input Source Map in DIGI 120G

Interrupt Signal Names	DIGI 120G Interrupt Source
SS_INT_0/1/2/3[0]	COREOTN
SS_INT_0/1/2/3[1]	LINEOTN
SS_INT_0/1/2/3[2]	MAPOTN
SS_INT_0/1/2/3[3]	ENET_LINE
SS_INT_0/1/2/3[4]	ENET_SYS
SS_INT_0/1/2/3[5]	CBRC
SS_INT_0/1/2/3[6]	SIFD
SS_INT_0/1/2/3[7]	DCPB
SS_INT_0/1/2/3[8]	MCPB
SS_INT_0/1/2/3[9]	TOPLEVEL    SYSOTN

**Note:**

1. The DCPB and MCPB interrupts can only be directed to the MIPs via SS\_INT\_0. They cannot be directed to the device interrupt pin nor to the PCIe Host.

The MIPS34Kc interrupts are wired to support both interrupt compatibility mode and VI (Vectored Interrupt) mode. The core powers up in interrupt compatibility mode at reset. During boot initialization, the core switches to VI mode as part of the interrupt handler initialization routine. Each instance of the VPE in the MIPS34Kc includes support for six hardware interrupt input pins, a timer interrupt output pin, and a NMI input pin. The timer interrupt output is connected to one of the six pin hardware interrupt inputs. The remaining five hardware interrupt pins are aggregated device and CPU subsystem peripheral interrupts. The NMI inputs are connected to the dual watchdog timer interrupt output. The same aggregated interrupt treatment is provided to 16 external yield system event inputs.

The PCIe subsystem supports 32 physical interrupts input lines and 32 outbound doorbell registers, that is, 64 interrupt sources. The 32 doorbells and 32 interrupt lines are aggregated into 32 unique interrupt signals, which are translated into a PBA (Pending Bit Array). If the MIPS processor writes a 1 to a bit of the Outbound Doorbell Register and the bit is not masked, or any of the interrupt lines is set, this will result in a set-bit in the PBA. The interrupt generation mechanism senses change to the PBA bits to trigger the sending of PCIe interrupts. Supported PCIe interrupts are 32 MSI interrupt and 16 MSI-X vectors. In the case of MSI-x, interrupt[31:16] and interrupt[15:0] alias to the MSI-X vec-

tor[15:0] respectively. If neither MSI not MSI-X is enabled, the interrupt mechanism defaults to the INTx legacy interrupts.

All the CPU subsystem internal interrupts and external interrupts are aggregated also into single device interrupt output pin. The aggregation scheme shown below allows firmware and software to filter interrupts as application demands.

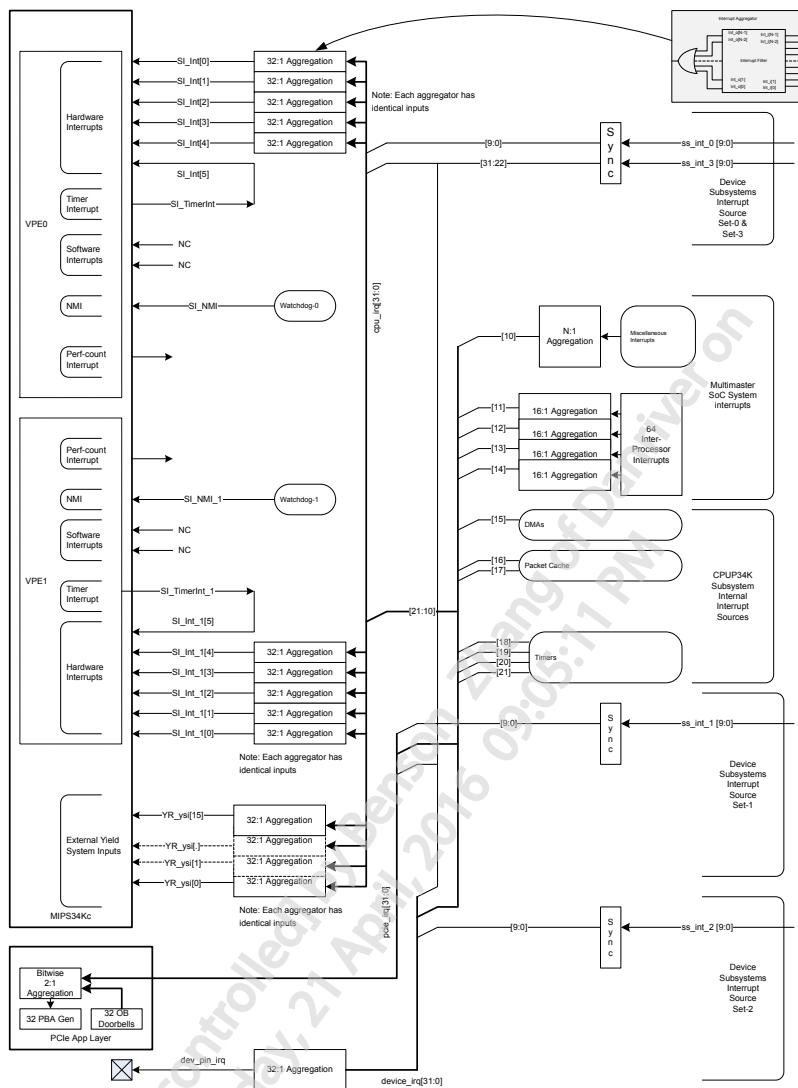
**Table 84 Summary of 32:1 Interrupt Aggregation Towards PCIe Host**

Index	Interrupt Sources	Description
31-22		Reserved
21-18		Reserved
17-16		Reserved
15		Reserved
14-11	IPI_INT	Aggregated 4 IPI interrupts to PCIe
10		Reserved
9-0	SS_INT_1[9:0]	CPUP34K external interrupt inputs dedicated to PCIe

**Table 85 32:1 Interrupt Aggregation Towards Device Interrupt Output Pin**

Index	Interrupt Sources	Description
31-22		Reserved
21-18		Reserved
17-16		Reserved
15		Reserved
14-11	IPI_INT	Aggregated 4 IPI interrupts to PCIe.
10		Reserved
9-0	SS_INT_2[9:0]	CPUP34K external interrupt inputs dedicated to device interrupt output pin.

Figure 159 Interrupt Aggregation



### 5.15.6.1 Known Interrupt Propagation Deficiency

Note that the packet buffer CH\_DROP\_FULL\_I, CH\_DROP\_CLIENT\_I, and CH\_DROP\_OVRSIZE\_I interrupts do not propagate up through the interrupt hierarchy at the subsystem level and above. The per channel interrupts are aggregated and summarized in the per switch port SLV\_DPI\_INT\_SUM [7:0]

within the subsystem. This issue applies to the ODUk switch, data central packet buffer (DCPB), and management central packet buffer (MCPB) instantiations of the packet buffer architecture.

The above interrupts will not propagate to the device INTB pin or PCIe interrupts. Software will not be notified when the interrupt fires.

To address this issue, poll the per packet buffer switch port SLV\_DPI\_INT\_SUM interrupt summary bits to determine when a per channel CH\_DROP\_FULL\_I, CH\_DROP\_CLIENT\_I, and CH\_DROP\_OVRSIZE\_I has occurred for one or more channels associated with the port then read the CH\_DROP\_FULL\_I, CH\_DROP\_CLIENT\_I, and CH\_DROP\_OVRSIZE\_I for every channel to determine which channel caused the SLV\_DPI\_INT\_SUM interrupt. The SLV\_DPI\_INT\_SUM[7:0] interrupts reflect the OR of all interrupts associated with DPI master port x (where x is 0 to 7) for the associated packet buffer. The bit field is located at the following addresses:

- ODUk Switch: 0xF40110
- MCPB: 0x540110
- DCPB: 0x440110

The CH\_DROP\_FULL\_I interrupt will increment the CH\_DROP\_CNT\_VAL counter. Read the CH\_DROP\_CNT\_VAL at regular PMON latch clock intervals to determine if a packet has been dropped during the interval due to a CH\_DROP\_FULL\_I event.

### 5.15.7 Scratch Pad RAM (SPRAM)

The SPRAM is one of the AXI slaves connected to the AMBA network interconnect. The SPRAM size is 768 KB, 64-bit wide, and 96K words deep. The SPRAM is a dedicated scratch pad storage for the internal firmware running on the MIPS processor. The SPRAM is protected by ECC. The internal firmware is responsible for allocating a specific region of SPRAM for exchanging content with the host software.

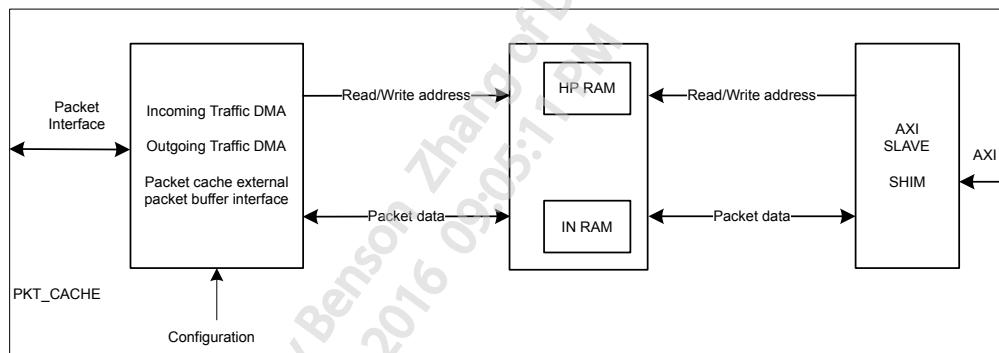
Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:11 PM

## 5.15.8 CPU Packet Cache (PKT\_CACHE)

The PKT\_CACHE is a peripheral in the CPUP34K subsystem that provides a protocol-independent dedicated packet interface that caches packets in order to provide a low latency path for CPU-based packet processing. It receives packets from the packet interface and caches them for firmware to process. Firmware implements protocol-based algorithms and then modifies the packet content by doing random read-write accesses within the packet boundary into the PKT\_CACHE. Modified packets are either consumed by firmware or sent back on the packet interface to an external packet buffer. The packet interface carries packet traffic in a channelized form where each channel may carry packets of variable length and type interleaved in packet mode. The PKT\_CACHE does packet caching in a queue in the order of arrival. A high priority queue is defined to cache the DPI packet data. There is also a packet origination queue that is used by firmware in order to insert packets originated by the PKT\_CACHE into the packet interface.

### 5.15.8.1 Block Diagram

Figure 160 PKT\_CACHE Block Diagram



### 5.15.8.2 Functional Description

#### 5.15.8.2.1 Overview

The content of the cache is DPI packet data where complete packets are cached in a RAM and the relevant portion of the DPI packet data is randomly accessed by the CPU. The CPU may process the packet content right from/in the cache and enables the PKT\_CACHE Outgoing DMA to push/discard the cached packet data out once the processing is over.

#### 5.15.8.2.2 Features

Major features of the PKT\_CACHE are:

- A DPI packet interface with physical characteristics as:
  - 64-bit wide interface.
  - Master and slave interfaces each running at 50 MHz.
- The PKT\_CACHE has an AXI slave interface for CPU access. The CPU side of PKT\_CACHE is memory mapped and is randomly accessible.
- Two packet queues named as:
  - High priority queue.
  - Packet origination queue.

- Two local DMA engines named as Incoming Traffic DMA and Outgoing Traffic DMA to write the incoming packet data into the PKT\_CACHE and read processed data from PKT\_CACHE respectively. Incoming traffic DMA is triggered by incoming traffic on the packet interface while the outgoing traffic DMA is triggered by firmware
- Event driven Interrupts for HW-FW interaction.
- PKT\_CACHE supports growth by up to 16 bytes at the beginning of a packet and up to 16 bytes at the end. It also supports up to 42 byte shrinkage at the start and up to 16 bytes shrinkage at the end.
- PKT\_CACHE does not expect purge command in origination queue.
- PKT\_CACHE can mark the DPI error over the outgoing packet based on per packet firmware command.
- PKT\_CACHE includes hardware FCS computation and insertion for outgoing traffic with per packet firmware control.

Downloaded [controlled] by Benson Zhang of Danriver  
Thursday, 21 April, 2016 09:05:11 PM

### 5.15.9 PCIe Subsystem

The PCIe subsystem integrates a PCIe application layer, a PCIe End Point core with integrated AXI gasket, a PCIe Physical Coding Sublayer (PCS), and a PCIe Physical Media Attachment (PMA) sublayer.

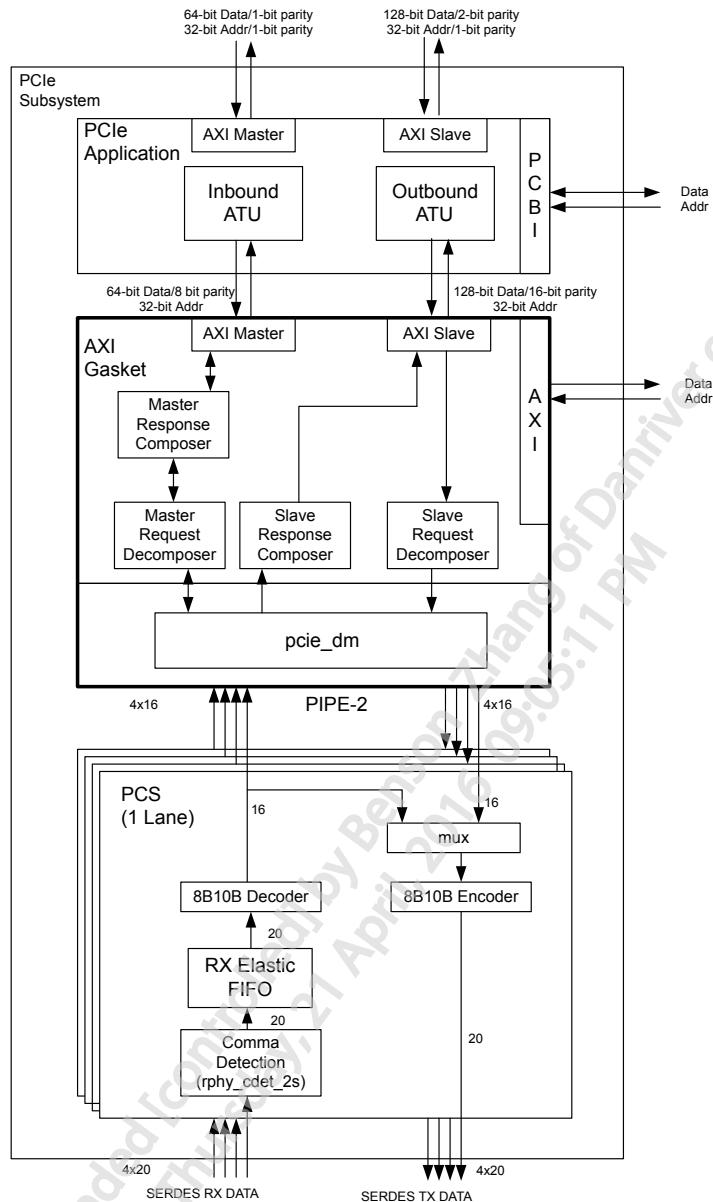
The PCIe application logic is responsible for providing both inbound and outbound AXI address translation. The translation tables are programmed via the application layer configuration interface. The application layer is also responsible for converting interrupt requests received to MSI/MSI-X/INTx requests to the PCIe core. These requests are done via the same data path as normal data so ordering is maintained.

The PCIe core integrates an AXI gasket that operates off of the PCIe core clock. This gasket logic is responsible for converting between the AXI domain and the PCIe TLP interfaces. This logic contains buffers used to retime data from the AXI system clock domain to the PCIe core's clock domain.

The PCIe End Point supports an 4-Lane PIPE-2 interface that is capable of operating up to 2.5 Gbps per lane for an aggregate link rate of 10 Gbps. The PIPE interface defines a standard interface between the PCIe PHY and a Media Access Layer (MAC) and Link Layer ASIC. The End Point can be configured in x1,x2 and x4 modes.

The PCS logic interfaces directly to the PCIe core's PIPE2 interface. It performs 8B/10B encoding/decode on data that is transferred between the PCIe core and the PMA sublayer. This logic also contains an elastic FIFO used to re-time received data from the recovered clock domain to the PCIe core clock domain. The PMA logic provides a direct interface to the PCIe SERDES. It aligns receive data on 10-bit boundaries. Transmitted data is synchronized to the SERDES transmit clock and receive data is synchronized to the SERDES recovered clock.

Figure 161 PCIe Subsystem Block Diagram



### 5.15.9.1 PCIe Inbound to AXI Bridge Overview

The following features are provided by the PCIe to AXI bridge:

- Supports 1 AXI master interface for inbound PCIe requests
- Supports up to 4 in flight AXI transactions using unique AXI IDs
- 32-bit read and write address buses
- 64-bit read and write data buses
- Supports incremental AXI burst type of size 64-bit and length ranging from 1 to 16 data transfers
- Supports a minimum 32-bit wide transfer aligned to 4-byte boundaries

- Supports PCIe read request sizes of 256 bytes
- AXI bursts cannot cross 128-byte page boundary. Supports PCIe maximum request sizes larger than the maximum AXI burst size by splitting PCIe read requests into multiple AXI read requests. The AXI read completions are collated into maximum-size PCIe read completion TLPs.
- Supports PCIe maximum payload sizes greater than the maximum AXI burst size by splitting inbound PCIe writes into multiple AXI writes

### 5.15.9.2 AXI to PCIe Outbound Bridge Overview

The following features are provided by the AXI to PCIe bridge:

- Supports 1 AXI slave interface for outbound PCIe requests
- Supports up to 32 in flight AXI transactions
- PCIe ordering rules are maintained when responding to AXI transfers
- 32-bit read and write address buses
- 128-bit read and write channel data buses
- Supports incremental AXI burst type of size 128-bit and length ranging from 1 to 16 data transfers
- Supports minimum 32-bit wide transfer aligned to four-byte boundaries
- Supports Traffic Class, No\_Snoop, Relaxed\_Order, TLP Digest, and TLP Poison control
- Supports a PCIe maximum read request size of 256 bytes

### 5.15.9.3 PCIe Application Layer Feature Overview

The PCIe application layer sits between the PCIe core and the AMBA network interconnect and has the following features:

- Supports 2 outbound translation window
- Supports 1 inbound translation window associated with PCIe BAR
- One outbound window dedicated to interrupt generation (MSI/MSI-X/INTx)
- Translates outbound 32-bit AXI address to 64-bit PCIe outbound address
- Translates inbound 64-bit PCIe address into (up to) 32-bit internal address
- Translation table and attributes are programmable
- PCIe Traffic Class, No Snoop, Relaxed Ordering, TLP Digest, and TLP Poison attributes are programmable on per window basis
- Supports MSI-X, MSI and INTx PCIe interrupt generation
- Maintains MSI-X Vector Table and MSI-X Pending Bit Array mapped into a PCIe BAR
- Application layer determines appropriate interrupt method and translation based on configuration bits from PCIe core and the MSI-X Vector Table.

### 5.15.9.4 PCIe Behaviour Versus Standards

When the `set_slot_power_limit` message is sent by an upstream port the Captured Slot Power Limit Value (bits 25:18) and the Captured Slot Power Limit Scale (bits 27:26) of the Device Capability Register (config offset 0x74) will not be captured properly IF the message is immediately followed by another TLP. For this to happen, the TLP must arrive at the PCI Express Protocol filter module on the clock cycle after the Set Slot Power Limit message arrives. The behavior is not likely to be observed as system level software or firmware typically will send the message after enumeration is completed but prior to handing over control to the driver software. Therefore, the `set_slot_power_limit` message is unlikely to be immediately followed by another TLP.

If needed, system software or firmware can delay generation of requests to the device after the transmission of the set\_slot\_power\_limit message.

#### 5.15.9.4.1 PCIe LTSSM Unexpected Transition to Recover upon Lane Reversal

The device PCIe supports lane reversal. The PCIe controller will automatically flip the lane order if an application that uses multiple PCIe lanes reverses the lane wiring on the PCB. In lane reversed applications with lane-to-lane skew, the PCIe Link Training Status State Machine (LTSSM) may unexpectedly transition to the recovery state after the first pass through the Configuration state. The de-skew logic is placed such that de-skew is done prior to lane reversal. Upon entering L0, the LTSSM suffers a de-skew error and subsequently transitions to Recovery. Upon exiting the Recovery state, de-skew on the lane reversed configuration is completed and the LTSSM returns to the L0 state for normal operation. This unexpected transition to the Recovery state is benign since the PCIe link will be correctly established during the second pass through configuration state machine.

#### 5.15.9.4.2 I/O Accesses

The PCIe interface does not support I/O mapped accesses. The PCIe standard does allow support for I/O mapped requests for legacy devices. The device does not support I/O mapped access and I/O accesses are not used by the device firmware or the device software driver.

#### 5.15.9.4.3 Hot Swap

Even though the 40nm RAPTR SERDES Cell Set Specification indicates the SERDES supports hot plug, the PCIe interface (which uses the C8 SERDES) does not support hot swap.

#### 5.15.9.4.4 Loopback State

The DIGI device does not support the PCIe loopback state. The PCIe protocol supports a loopback state where a master device can place the link into the loopback state by transmitting TS1 ordered sets with the loopback bit asserted. The feature is used for test and debug and is not used during normal operation.

#### 5.15.9.5 PCIe Read Transaction Behavior

When a PCIe root complex is initiating a read request packet for 4 bytes for a address aligned to 8-byte boundary, the device PCIe subsystem will initiate a 64-bit (8 bytes) read transfer on the AXI Bus. The PCIe subsystem discards the unwanted 4-bytes received from the AXI bus and sends only a 4-byte completion packet to the root complex as expected.

The PCIe subsystem reads one extra 32-byte word from the DIGI register space under the following two scenarios. The read request size is a multiple of 8 bytes and the address is aligned to 4 (not 8) bytes, or, the read request size is 4 bytes and the address is aligned to 8 bytes. If a register exists at the next address with a clear-on-read property, the read of the first address will result in read-on-clear bits within the next address being cleared. Therefore, clear-on-read interrupts are not supported within the DIGI device and other clear-on-read registers need to consider the implications of potentially an adjacent register that is also clear-on-read.

#### 5.15.9.6 Considerations When Addressing Device Registers from a PCIe Root Port

The ENET\_TOP registers within the ENET(LINE) and ENET(SYS) subsystems are allocated an address space of 14 bits (0x0000 to 0x3FFF) relative to the base address of the block within the register map but only 10-bits (0x000 to 0x3FF) are addressable. Accesses to registers within the 0x0000 to 0x3FFF

range are aliased to the addressable portion of the register map. For example, writing register 0x0400 relative ENET\_TOP base address will write register at address 0x000. Similarly, register 0x0000 is accessed at addresses of 0x800, 0xC00, 0x1000, 0x1400, 0x1800, 0x1C00, and so on. The register aliasing is not an issue from a system perspective but does need to be considered when reviewing debug logs of device register accesses since a write to an unused register address within the ENET\_TOP address space may inadvertently reconfigure the block.

Note a read of a 32-bit register will result in the read of the register at the following address since internally, the device implements 64-bit reads. The PRBS31\_ERR\_CNT registers are grouped in 12 consecutive addresses, one address for each channel. The counts clear on read. Therefore, reading the error count for one channel will also clear the error count for the next channel in the address map. The PRBS9\_ERR\_CNT registers are similarly organized. To work around this, perform diagnostic checking using only a single channel at a time. The Ethernet PRBS counters are diagnostic only and the clearing of adjacent channels is not expected to have any system implications.

Downloaded [controlled] by Benson Zhang of Danriver  
Thursday, 21 April, 2016 09:05:11 PM

### 5.15.10 PCIe PHY

The PCIe PHY consists of a Physical Coding Sublayer (PCS), a Physical Media Attachment (PMA) sublayer and the analog SERDES. The PCIe PHY supports the following:

- Compliant with PCI Express Base Specification 1.1 (for Gen 1)
- Supports x1, x2, and x4 links running 2.5 Gbps (Gen 1)
- Supports PIPE specification version 2.0

The PCS logic interfaces directly to the PIPE interface. It performs 8B/10B encoding/decoding on data that is transferred between the PCIe Core and the PMA sublayer. This logic also contains an elastic FIFO used to re-time received data from the recovered clock domain to the PCIe core clock domain.

The PMA logic provides a direct interface to the PCIe SERDES. It aligns receive data on 10-bit boundaries. Transmitted data is synchronized to the SERDES' transmit clock, while receive data is received synchronous to SERDES' recovered clock and then synchronized to the PCIe core clock for further processing.

## 5.15.11 Peripheral Configuration

There is one APB configuration bus and one AXI configuration bus provided by the AMBA interconnect in the CPUP34K. The APB bus is used to access APB peripherals such as watchdog timers, software timers and configuration space of CPUP34K subsystem components. The AXI bus is driven out of CPUP34K and is used to configure the rest of device configuration register space.

### 5.15.11.1 APB Peripherals

#### 5.15.11.1.1 CPUP34K Peripheral Configuration

CPUP34K peripherals such as subsystem configuration register block (Subsystem Top), Interrupt Aggregate, PCIe application layer, PCIe MAC, PCIe PCS/PMA and memory mapped SPI Flash controller block have APB port to provide access to its configuration registers.

#### 5.15.11.1.2 Watchdog Timer

The subsystem integrates two hardware watchdog timers, one for each VPE (Virtual Processing Element) in the MIPS processor operating at the system clock. The watchdog counts from a configurable timeout value in descending order to zero. If the counter reaches zero, a non-maskable interrupt (NMI) is signaled to the MIPS core. Writing to the watchdog load register at any time restarts the counter to the new value. Clearing the watchdog interrupt also causes the watchdog timer to be reloaded with the load register value. If the watchdog timer underflows a second time (for example, for an underflow while the NMI is still unacknowledged) a reset output will become asserted. The watchdog timers are local resources to the MIPS processor and are configured and used only by firmware running on the processor.

#### 5.15.11.1.3 Software Timer

The subsystem has four soft-controllable timers. Each has a 32-bit counter, up to 8 bits of prescaling, and is clocked at the system clock. The combination provides software timer time-out periods of over an hour. The software timer can be enabled through a control register. Once enabled, the software timer starts counting down. When it reaches 0, it generates an interrupt and loads one of two values depending on the timer operating mode. The interrupts are not generated if the timer is disabled. The soft-controllable timers are a local resource to the MIPS processor and are configured and used only by firmware running on the processor.

### 5.15.11.2 AXI Configuration

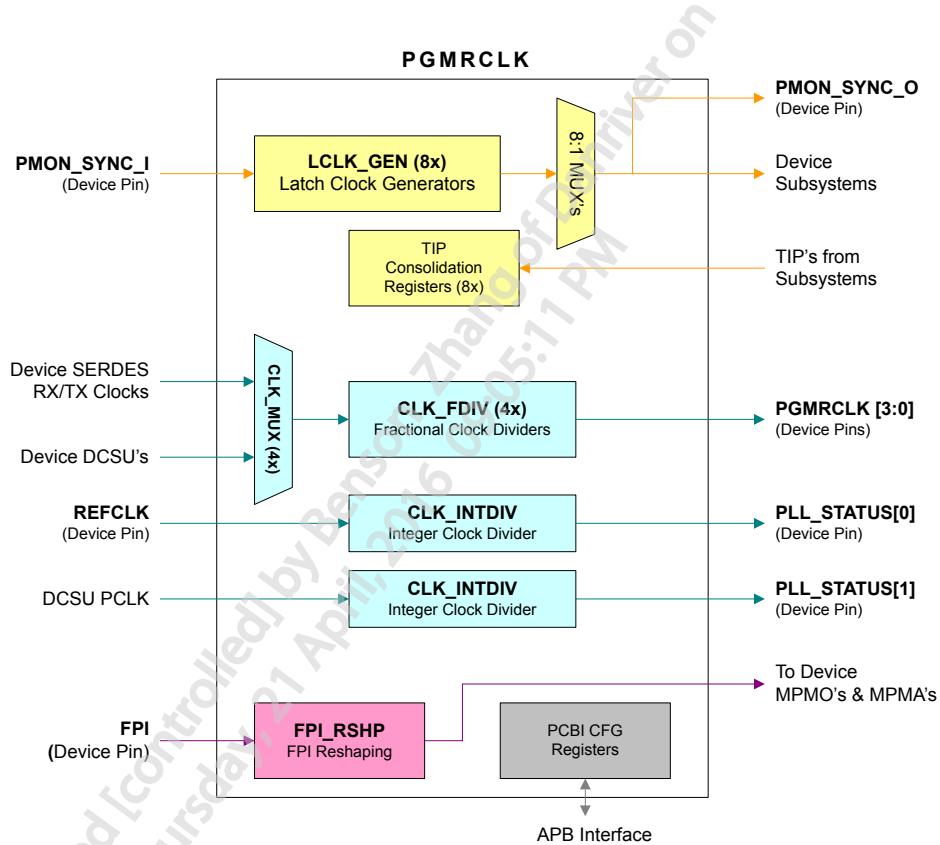
There is no peripheral in the CPUP34K subsystem for the AXI configuration bus. The AXI configuration bus is driven out of the CPUP34K subsystem for device configuration. The master AXI bus used to configure the subsystem outside of the CPU subsystem has a 32-bit data bus that supports only full 32-bit (full word) accesses. The AXI configuration bus supports 6 outstanding reads 12 outstanding writes.

## 5.16 Diagnostic and Test Features

### 5.16.1 PGMRCLK, LCLK, TIP, and FPI Windowing

#### 5.16.1.1 Block Diagram

Figure 162 PGMRCLK Block Diagram



#### 5.16.1.2 Overview

The PGMRCLK (Programmable Recovered Clock) block shown in Figure 162 contains the implementations for the Programmable Recovered Clocks, PMON Latch Clocks, TIP Consolidation, and FPI Reshaping blocks. Selected internal clocks are recovered by this block to the external device pins PGRMCLK[3:0], and generated PMON Latch Clocks are provided by this block to the subsystems within the device. External PMON Latch Clock synchronization is possible through the PMON\_SYNC\_I device input and external PMON latch clock generation is possible through the PMON\_SYNC\_O device output.

The PGMRCLK block provides additional logic to also directly monitor the status of REFCLK and PCLK on the external device pins PLL\_STATUS[1:0]. The block also contains logic to reshape the incoming FPI signal and to monitor the status of the subsystem TIP (Transfer In Progress) signals.

### 5.16.1.3 Programmable Recovered Clocks (PGMRCLK)

The PGMRCLK block provides access to all of the internal reference clocks and recovered/transmit clocks for each of the SERDES device interfaces. Any four of these clocks can be selected to be divided down and accessed on the external PGMRCLK[3:0] device pins:

- SFI-5.1 interface RX/TX Clocks
- Line interface RX/TX Clocks
- System interface RX/TX Clocks
- ENET Management interface RX/TX Clocks
- OTN Overhead interface RX/TX Clocks
- PCIE interface RX/TX Clocks
- Backplane SR interface RX/TX Clocks
- Backplane LR interface RX/TX Clocks
- RCP interface RX/TX Clocks
- SYS1 & SYS2 DCSU Clocks
- OTN1 & OTN2 DCSU Clocks
- ENET1 & ENET2 DCSU Clocks
- PTP DCSU Clocks
- ENET3 DCSU Clock
- Demapped Ethernet RX/TX Clocks
- IRIG Interface Clocks
- FEC Stats Interface Clock
- Interlaken Interface Clocks
- SPI Interface Clock
- Ring Oscillators

A complete list of clocks selectable by each fractional divider is shown in [Table 86](#) below.

Each PGMRCLK[3:0] output has its own fractional divider (CLK\_FDIV) and each clock must be divided down to be less than 125 MHz. This is the maximum frequency supported by the PGMRCLK[3:0] device pads. The division ratio for each fractional divider is register configurable using a 32-bit integer, numerator, and denominator as follows:

$$\text{PGMRCLK[3:0]} = \text{Selected Clock} / (\text{integer} + \text{numerator} / \text{denominator})$$

**Note:** If the selected clock is from one of the device interfaces, the core clock rate is fractionally divided and not the client rate. To see the core clock rates for the various supported clients and device interfaces, please see [Section 5.16.1.3.3](#).

The fractional dividers have the following limitations:

- The numerator must be less than 32 bits (2^31).
- Fractional divides of 4.XXX are not supported; however, integer divides of 4 are supported.
- The numerator must be less than the denominator.
- Fractional divides of 1.XXX are not supported; however, integer divides of 1 are supported.

For applications other than debugging, the PGMRCLK[3:0] outputs can also be used as a timing references that are traceable to the clock source used to derive the timing of the selected PGMRCLK[3:0] source. An external jitter attenuator and PLL with holdover may be used in order to generate a low

jitter clock suitable for use as reference for other devices. For the AC characteristics of the PGMRCLK[3:0] device pins, please see [Section 15.1.3](#).

The numerator and denominator for a desired fractional division can be calculated using Rational Fraction Approximation. The fractional divider configurations to recover a 2.048 MHz and 8 kHz timing reference are provided in [Section 5.16.1.3.4](#) and [Section 5.16.1.3.5](#). The minimum frequency supported by the fractional dividers is constrained by the original frequency of the divided clock and the 32-bit integer portion of the divider. The jitter of the PGMRCLK[3:0] outputs is inversely proportional to the original frequency of the divided clock.

Two separate integer dividers and dedicated outputs are also provided within in the PGMRCLK block in order to provide direct access to the REFCLK and PCLK. This allows for the status of the PCIE interface to be checked on the dedicated device pins PLL\_STATUS[1:0] using an FPGA or clock monitoring device. If the PCIE interface is fully functional, which requires both the REFCLK and PCLK, it can be used to confirm the status of all other DCSU's in the device without any of them being operational. An interrupt can also be generated using software if any of the device DCSU's loses lock during normal device operation. The division ration for each integer divider is fixed to an integer value of 256. PLL\_STATUS[0] is mapped to REFCLK and PLL\_STATUS[1] is mapped to PCLK.

For the AC Characteristics of the PGMRCLK[3:0] device pins, please see [Section 15.1.4](#).

The device clock diagram can be seen in [Section 8.1](#).

### 5.16.1.3.1 Using a Spare S16 Output as a Reference Clock for Optics Modules

The PMC device provides LVDS reference clock outputs, LINE\_REFCLK[11:0]\_P/N and SYS\_REFCLK\_P/N[11:0], for use with optics modules connected to the line or system interfaces, respectively. The device supports use of unused LINE\_TXO\_P/N[11:0] and SYS\_TXO\_P/N[15:0] S16 outputs as low jitter divide by 16 or divide by 64 reference clocks that are frequency locked to the transmit SERDES connecting to an optical module. The following restrictions apply.

- A S16 output can only be used as a reference clock for optics connected to the same line or system side interface. For example, a SYS\_TXO\_P/N[15:0] output cannot be used as a clock source for optics connected to the LINE\_TXO\_P/N[11:0] interface. Similarly, a LINE\_TXO\_P/N[11:0] output cannot be used as a clock source for optics connected to the SYS\_TXO\_P/N[11:0] interface.
- A LINE\_TXO\_P/N[11:0] and SYS\_TXO\_P/N[15:0] or S16 output cannot be used as a clock source for the SFI-5.1 interfaces.
- Use of a S16 output as a reference clock for 10G optics is not supported. Use the LINE\_REFCLK[11:0]\_P/N and SYS\_REFCLK\_P/N[11:0] outputs if a reference clock is required for a 10G optics module (e.g. XFP).
- Do not use lanes 0 and 1 of the system side interface as a clock (SYS\_TXO\_N/P[1:0]). The proximity of the outputs to the device REFCLK input results in spurs in the phase noise when the outputs are used as a clock.
- Do not use adjacent lanes as clocks for the 2 x 40G application. Space the S16 outputs used as a clock as far apart as possible.
- On the line side, the highest index lanes have the best phase noise performance (e.g. LINE\_TXO\_P/N[11] has the best performance). On the system side, the lowest indexed lane has the best phase noise performance (e.g. SYS\_TXO\_P/N[2]) taking into consideration restriction one above.

The above restrictions limit the use cases to 2x40G or 1x100G applications where spare S16 outputs are available. Refer to [\[7\]](#) for the jitter specification of the S16 output when used as a reference clock.

### 5.16.1.3.2 Recovered Clock Selection

Table 86 CLK\_MUX\_SEL[8:0] Values for Recovered Clock Selection

Interface / DCSU	Clock Selected	CLK_MUX_SEL[8:0]
SFI-5.1 Interface	SFI1_TX_CLK[15:0]	0-15
	SFI1_DSC_TX_CLK	16
	SFI1_RX_CLK[15:0]	17-32
	SFI1_DSC_RX_CLK	33
	SFI2_TX_CLK[15:0]	34-49
	SFI2_DSC_TX_CLK	50
	SFI2_RX_CLK[15:0]	51-66
	SFI2_DSC_RX_CLK	67
Line Interface	LINE_TX_CLK[11:0]	68-79
	LINE_RX_CLK[11:0]	80-91
System Interface	SYS_RX_CLK[15:0]	92-107
	SYS_TX_CLK[15:0]	108-123
ENET MGMT Interface	MGMT_RX_CLK	124
	MGMT_TX_CLK	125
PCIE Interface	PCIE_TX_CLK[3:0]	126-129
	PCIE_RX_CLK[3:0]	130-133
Backplane SR Interface	SR_RX_CLK[7:0]	134-141
	SR_TX_CLK[7:0]	142-149
Backplane LR Interface	LR_RX_CLK[39:0]	150-189
	LR_TX_CLK[39:0]	190-229
SYS2 DCSU	PCLK	230
SYS1 DCSU	REF300_CLK	231
SYS1 DCSU	REF600_CLK	232
SYS2 DCSU	REF500_CLK	233
SYS2 DCSU	REF653_CLK	234
ENET2 DCSU	REF515_CLK	235
OTN1 DCSU	REF349_CLK	236
ENET3 DCSU	REF311_CLK	237
Reserved	Reserved	238
Reserved	Reserved	239
OTN2 DCSU	REF355_CLK	240
PTP DCSU	PTP_CLK	241
RCP Interface	RRCP RX	242
	DRCP RX	243
	RRCP TX	244
	DRCP TX	245
OTN Overhead Interface	OH_RX_CLK[2:0]	246-248
	OH_TX_CLK[2:0]	249-251
Demapped Ethernet	DEMAP_OUT_CLK[11:0]	252-263
Reserved	Reserved	252-263
Device Interface	REFCLK	264
ENET1 DCSU	REF518_CLK	265
Reserved	Reserved	266-267

Interface / DCSU	Clock Selected	CLK_MUX_SEL[8:0]
Ring Oscillators	ROMON_OUT_CLK	268-283
	PMON_OUT_CLK	284
IRIG Interface	IRIG_CLKI	285
	IRIG_CLKO	286
SPI Interface	SPI_SCLK	287
SYS2 DCSU	REF50_CLK	288
Reserved	Reserved	289-511
1. There are four separate CLK_MUX_SEL[8:0] values corresponding to each of the PGMRCLK[3:0] device outputs. Use the values in this table to configure each output from the provided selection.		

### 5.16.1.3.3 Core Clock Frequencies

Table 87 SERDES Client Core Clock Frequencies

Standard / Client	Aggregate Rate (Kbps)	# SR Links	Link Bit Rate (Kbps)	Core Width	Core Frequency (MHz)
D8 / T8 / C8 SERDES Client Core Clock Frequencies					
OC-768 / STM-256	39,813.12	16	2,488.32	8	311.04
OTU2	10,709.23	4	2,677.31	8	334.66
OTU1e	11,049.11	4	2,762.28	8	345.28
OTU3	43,018.41	16	2,688.65	8	336.08
OTU3e1	44,570.97	16	2,785.69	8	348.21
OTU3e2	44,583.36	16	2,786.46	8	348.31
SGMII 1.25G	1,250.00	1	1,250.00	10	125.00
PCIE 4 x 2.5G	10,000.00	4	2,500.00	10	250.00
CEI-6G SR	50,000.00	8	6,250.00	20	312.50
CEI-6G SR	25,000.00	8	3,125.00	20	156.25
CEI-6G SR	150,000.00	24	6,250.00	20	312.50
CEI-6G SR	75,000.00	12	6,250.00	20	312.50
CEI-6G LR	150,000.00	24	6,250.00	20	312.50
CEI-6G LR	62,500.00	20	3,125.00	20	156.25
CEI-6G LR	25,000.00	8	3,125.00	20	156.25
CEI-6G LR	125,000.00	20	6,250.00	20	312.50
CEI-6G LR	150,000.00	24	6,250.00	20	312.50
S16 SERDES Client Core Clock Frequencies					
OC-192 / STM-64	9,953.28	1	9,953.28	32	311.04
OC-768 / STM-256	39,813.12	4	9,953.28	32	311.04
CPRI 4.9152G	4,915.20	1	4,915.20	32	153.60
CPRI 6.1440G	6,144.00	1	6,144.00	32	192.00
CPRI 9.8304G	9,830.40	1	9,830.40	32	307.20
FC-800 / 8GFC	8,500.00	1	8,500.00	32	265.63
FC-1200 / 10GFC	10,518.75	1	10,518.75	32	328.71
FC-1200 with FEC	11,320.00	1	11,320.00	32	353.75
OTU2	10,709.23	1	10,709.23	32	334.66
OTU1e	11,049.11	1	11,049.11	32	345.28
OTU2e	11,095.73	1	11,095.73	32	346.74
OTU1f	11,270.09	1	11,270.09	32	352.19
OTU2f	11,317.64	1	11,317.64	32	353.68
OTU3	43,018.41	4	10,754.60	32	336.08

Standard / Client	Aggregate Rate (Kbps)	# SR Links	Link Bit Rate (Kbps)	Core Width	Core Frequency (MHz)
OTU3e1	44,570.97	4	11,142.74	32	348.21
OTU3e2	44,583.36	4	11,145.84	32	348.31
OTU4, OTL4.10/CAUI	111,809.97	10	11,181.00	32	349.41
10GE LAN, Base-R	10,312.50	1	10,312.50	20	515.63
40GE	41,250.00	4	10,312.50	20	515.63
100GE	103,125.00	10	10,312.50	20	515.63
GDPS 5G	5,000.00	1	5,000.00	32	156.25
GDPS 10G	10,000.00	1	10,000.00	32	312.50
CEI-11G SR	165,000.00	16	10,312.50	20	515.63
CEI-11G SR	82,500.00	8	10,312.50	20	515.63
CEI-11G MR	149,500.00	13	11,500.00	20	575.00
CEI-11G MR	150,000.00	12	12,500.00	20	625.00
CEI-6G SR	100,000.00	16	6,250.00	20	312.50
CEI-6G SR	50,000.00	16	3,125.00	20	156.25
CEI-6G SR	150,000.00	24	6,250.00	20	312.50
CEI-6G SR	75,000.00	12	6,250.00	20	312.50

**Note:**

1. D8/T8/C8 SERDES are used for the OTN OH, SFI-5.1, ENET MGMT, RCP, Backplane SR, and Backplane LR Interfaces.
2. S16 SERDES are used for the System and Line Interfaces.
3. Core Frequency is the Aggregate Rate divided by the Number of SERDES Links and by the Core Width. The PGMRCLK[3:0] outputs monitor the Core Frequencies.

#### 5.16.1.3.4 Recovered 2.048MHz Clock

**Table 88** Divider CFG for 2.048 MHz Recovered CLK

Standard / Client	Core Frequency Expressed as Fraction (MHz)	Integer	Numerator	Denominator	PGMRCLK[3:0] Output (MHz)
<b>Line/System Intf. Divider CFG for 2.048 MHz Recovered CLK</b>					
OC-192 / STM-64	7776/25	151	7	8	2.048
OC-768 / STM-256	7776/25	151	7	8	2.048
CPRI 4.9152G	768/5	75	0	1	2.048
CPRI 6.144G	192/1	93	3	4	2.048
CPRI 9.8G	1536/5	150	0	1	2.048
FC-800 / 8GFC	2125/8	129	1,433	2,048	2.048
FC-1200 / 10GFC	42075/128	160	16,495	32,768	2.048
FC-1200 with FEC	1415/4	172	747	1,024	2.048
OTU2	132192/395	163	259	632	2.048
OTU1e	309375/896	168	136,707	229,376	2.048
OTU2e	1753125/5056	169	397,841	1,294,336	2.048
OTU1f	631125/1792	171	444,033	458,752	2.048
OTU2f	3576375/10112	172	1,795,291	2,588,672	2.048
OTU3	99144/295	164	193	1,888	2.048

Standard / Client	Core Frequency Expressed as Fraction (MHz)	Integer	Numerator	Denominator	PGMRCLK[3:0] Output (MHz)
OTU3e1	5259375/15104	170	95,795	3,866,624	2.048
OTU3e2	1889568/5425	170	125	1,736	2.048
OTU4, OTL4.10/CAUI	396576/1135	170	1,105	1,816	2.048
10GE LAN, Base-R	4125/8	251	1,577	2,048	2.048
40GE	4125/8	251	1,577	2,048	2.048
100GE	4125/8	251	1,577	2,048	2.048
GDPS 5G	625/4	76	301	1,024	2.048
GDPS 10G	625/2	152	301	512	2.048
<b>SFI-5.1 Intf. Divider CFG for 2.048 MHz Recovered CLK</b>					
OC-768 / STM-256	7776/25	151	7	8	2.048
OTU3	99144/295	164	193	1,888	2.048
OTU3e1	5259375/15104	170	95,795	3,866,624	2.048
OTU3e2	1889568/5425	170	125	1,736	2.048
<b>Note:</b>					
1. Core Frequencies were calculated in <a href="#">Section 5.16.1.3.3</a> .					

### 5.16.1.3.5 Recovered 8kHz Clock

Table 89 Divider CFG for 8 KHz Recovered CLK

Standard / Client	Core Frequency Expressed as Fraction (MHz)	Integer	Numerator	Denominator	PGMRCLK[3:0] Output (MHz)
<b>Line/System Intf. Divider CFG for 2.048 MHz Recovered CLK</b>					
OC-192 / STM-64	7376/25	38,880	0	1	0.008
OC-768 / STM-256	7776/25	38,880	0	1	0.008
CPRI 4.9152G	768/5	19,200	0	1	0.008
CPRI 6.144G	192/1	24,000	0	1	0.008
CPRI 9.8G	1536/5	38,400	0	1	0.008
FC-800 / 8GFC	2125/8	33,203	1	8	0.008
FC-1200 / 10GFC	42075/128	41,088	111	128	0.008
FC-1200 with FEC	1415/4	44,218	3	4	0.008
OTU2	132192/395	41,832	72	79	0.008
OTU1e	309375/896	43,160	515	896	0.008
OTU2e	1753125/5056	43,342	3,473	5,056	0.008
OTU1f	631125/1792	44,023	1,409	1,792	0.008
OTU2f	3576375/10112	44,209	5,467	10,112	0.008
OTU3	99144/295	42,010	10	59	0.008
OTU3e1	5259375/15104	43,526	5,171	15,104	0.008

Standard / Client	Core Frequency Expressed as Fraction (MHz)	Integer	Numerator	Denominator	PGMRCLK[3:0] Output (MHz)
OTU3e2	1889568/5425	43,538	94	217	0.008
OTU4, OTL4.10/CAUI	396576/1135	43,675	175	227	0.008
10GE LAN, Base-R	4125/8	64,453	1	8	0.008
40GE LAN	4125/8	64,453	1	8	0.008
100GE LAN	4125/8	64,453	1	8	0.008
<b>SFI-5.1 Intf. Divider CFG for 2.048 MHz Recovered CLK</b>					
OC-768 / STM-256	7776/25	38,880	0	1	0.008
OTU3	99144/295	42,010	10	59	0.008
OTU3e1	5259375/15104	43,526	5,171	15,104	0.008
OTU3e2	1889568/5425	43,538	94	217	0.008

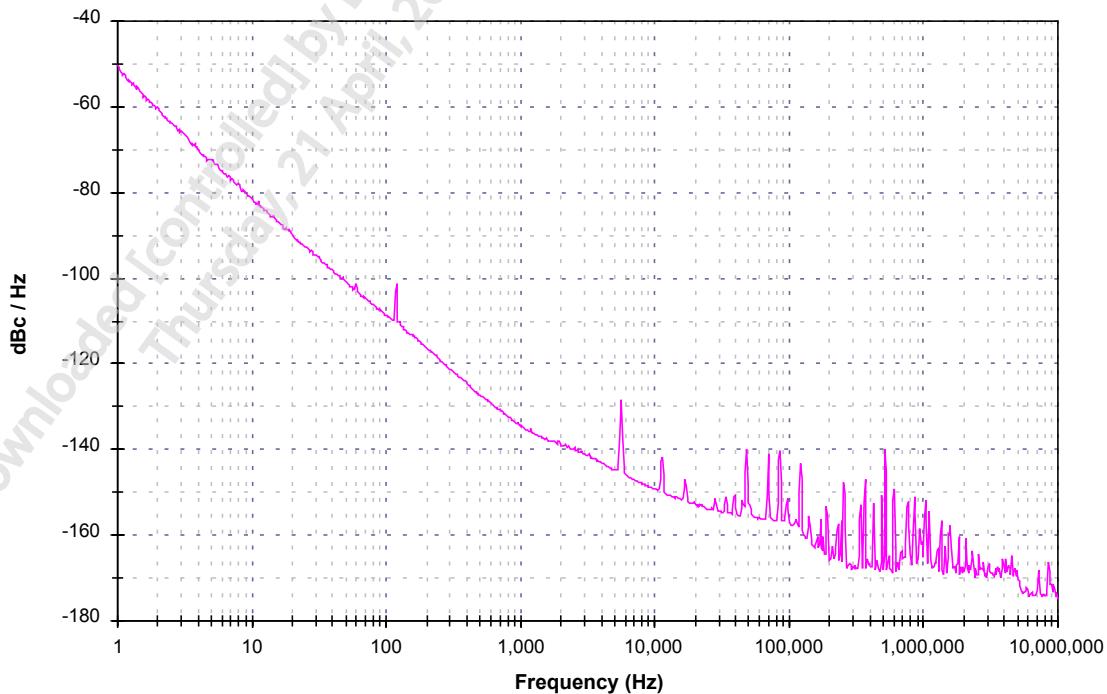
**Note:**

1. Core Frequencies were calculated in [Section 5.16.1.3.3](#).
2. Numerator and denominator were calculated using Rational Fraction Approximation.

### 5.16.1.3.6 Recovered Clock Jitter

The results in this section are not valid for demapped ethernet recovered clocks. For the demapped case, please see [Section 5.16.1.3.7](#).

**Figure 163 Typical Phasenoise Plot For 2.048MHz Recovered Clock on PGMRCLK[3:0] Outputs**



**Note:**

1. If a output frequency other than 2.048MHz is used, this plot should be adjusted by adding  $20 \cdot \log_{10}(2.048\text{MHz}/F_{\text{new}})$ .
2. This phase noise plot is only valid for Integer divides. If fractional division is used, the plot is only valid for frequencies less than 1kHz for the worst case numerator / denominator ratios.
3. The clock source used to form the 2.048MHz output has very little effect on the phasenoise plot.

**Table 90** Typical JitterRMS For 2.048MHz Recovered Clock on PGMRCLK[3:0] Outputs

Frequency Band	JitterRMS (ps) Integer Divide (Typical)	JitterRMS (ps) Fractional Divide (Typical)
1e0 - 1e1 Hz	220	220
1e1 - 1e2 Hz	22	22
1e2 - 1e3 Hz	4	4
1e3- 1e4 Hz	1	653
1e4 - 1e5 Hz	1	758
1e5 - 1e6 Hz	2	446
1e6 - 1e7 Hz	2	685
1e1 - 1e7 Hz	221	848

**Note:**

1. The fractional divide jitterRMS is dominated by the numerator/denominator ratio in a non-predictable fashion. The numbers provided in this table use a worst case ratio in attempt to bound the results. Other numerator/denominator ratios could potentially have worse jitterRMS but will most likely have better jitterRMS than the numbers provided above.
2. The jitterRMS for different frequency bands can not be simply added together to form a new frequency band and jitterRMS number. The reason behind this is that the JitterRMS calculation takes the square root of the area underneath of the phase noise plot.
3. The clock source used to form the 2.048MHz output has very little effect on the jitterRMS numbers.

**5.16.1.3.7 Demapped ENET Recovered Clock Jitter**

The pgmrclk[3:0] outputs can optionally be programmed to recover the rate of a demapped 10GE, 40GE, or 100GE client from an ODU carrier using the DEMAP\_OUT\_CLK[11:0] clock signals from [Table 86](#). These signals are provided to the pgmrclk fractional dividers by the ETRANS sub-block in the ENET\_LINE subsystem. These clock signals are also inherently "gappy" which causes poor clock jitter performance on the pgmrclk[3:0] outputs. The PGMRCLK outputs in general require an external jitter attenuator prior to use by other devices since they are digital outputs and the PGMRCLK divider introduces jitter. The demapped Ethernet clocks have significantly greater jitter when compared to a clock derived from the receive or transmit SerDes due to the gapped clock source. Consider the jitter when designing an external JAT to ensure the system requirements are met for any device using the externally jitter attenuated demapped Ethernet clock as a reference clock. [Table 91](#) below shows the jitter performance of 10GE, 40GE, and 100GE demapped ethernet recovered clocks using the pgmrclk[3:0] outputs.

**Table 91** Typical TIE Jitter For 2.048MHz Demapped ENET Recovered Clock on PGMRCLK[3:0] Outputs

Data Rate:	TIE Filter:	TIE Jitter pk-pk (ns):	pgmrclk[3:0] (MHz):	UI:
10GE	1MHZ_LPF	210	2.048	0.431

Data Rate:	TIE Filter:	TIE Jitter pk-pk (ns):	pgmrclk[3:0] (MHz):	UI:
10GE	100kHz_LPF	98	2.048	0.201
10GE	no_LPF	208	2.048	0.4263
40GE	1MHz_LPF	194	2.048	0.3971
40GE	100kHz_LPF	106	2.048	0.2176
40GE	no_LPF	189	2.048	0.3863
100GE	1MHz_LPF	144	2.048	0.2944
100GE	100kHz_LPF	76	2.048	0.1552
100GE	no_LPF	142	2.048	0.2911

#### 5.16.1.4 PMON Latch Clock (LCLK)

The PGMRCLK block also contains eight LCLK generators which can be used by the device subsystems to transfer the contents of holding registers and clear counters for performance monitoring purposes.. Each generator can be individually configured to create a periodic LCLK with a period between 625  $\mu$ s and 2.048 s in 125  $\mu$ s intervals. Manual LCLK generation is also possible in each of the LCLK generators through a register access where an individual bit corresponds to each subsystem.

The LCLK connections to each device subsystem is show in [Table 92](#). Each of these LCLKs can be individually sourced from any one of the eight LCLK generators.

LCLK is owned by firmware and is available to system software via the firmware interface. Refer to the Programmer's Manual [\[4\]](#) for details on usage.

**Table 92** Device Subsystem LCLK Connections

Subsystem	LCLK(s)
CPUK34K	cpuk_lclk [1:0] -[1]: lclk_1s -[0]: lclk_5ms
COREOTN	cotn_lclk [2:0] -[2]: lclk_pmon -[1]: lclk_defects -[0]: lclk_rcp
LINEOTN	ltn_lclk [2:0] -[2]: unused -[1]: otn_otu_lclk -[0] lclk
CBRC	cbrc_lclk
ENET SYS	esys_lclk
ENET LINE	elin_lclk
SYSOTN	sotn_lclk [1:0] -[1]: otnotu_lclk -[0]: lclk
MAPOTN	motn_lclk
SIFD	sifd_lclk
DCPB	dcpb_lclk
MCPB	mcpb_lclk
MGMT FEGE	fege_lclk

The PMON\_SYNC\_I device input pin allows for the LCLK generators to be individually synchronized to a system wide external signal. The generators will resynchronize to PMON\_SYNC\_I each time a rising edge arrives outside of a configurable masking window centered around the expected rising edge. This allows for small movements in PMON\_SYNC\_I to be ignored. In the absence of a synchronization signal on the PMON\_SYNC\_I input, the LCLK generators will act as a flywheel (that is, continue to generate latch clocks off of the last synchronization signal received on PMON\_SYNC\_I). This flywheeling behavior allows for low frequency or non periodic signals to be used as a synchronization source. When PMON\_SYNC\_I is not used for synchronization, one LCLK generator can optionally be used to synchronize the other seven LCLK generators under the assumption that the eight generators are configured to be integer multiples of each other.

The PMON\_SYNC\_O device output pin allows for the device to be the generator (or source) of an external system wide synchronization signal.

For the AC characteristics of the PMON\_SYNC\_O and PMON\_SYNC\_I device pins, please see [Section 15.1.5](#) and [Section 15.1.6](#).

### 5.16.1.5 TIP Consolidation

For each of the LCLKs generated by the PGMRCLK block, there is a corresponding TIP signal from the subsystems which is mapped back to one of the eight TIP consolidation registers. These incoming TIP signals indicate the status of a transfer initiated by the rising edge of an LCLK within one of the subsystems (i.e "is in progress" or "is completed"). All TIP signals are mapped back to the consolidation register that is associated with the LCLK generator that initiated the respective "transfer". For example if an LCLK was generated by LCLK Generator #3, then the TIP signal received back from the subsystem upon transfer completion would be mapped to TIP Consolidation Register #3. Each of the six consolidation registers has a interrupt bit to indicate if all of the TIP signals mapped to it have indicated transfer completed.

Table 93 Device Subsystem TIP Connections

Subsystem	LCLK(s)
CPUK34K	cpuk_tip [1:0]
COREOTN	cotn_tip [2:0]
LINEOTN	lotn_tip [2:0]
CBRC	cbrc_tip
ENET SYS	esys_tip
ENET LINE	elin_tip
SYSOTN	sotn_tip [1:0]
MAPOTN	motn_tip
SIFD	sifd_tip
DCPB	dcpb_tip
MCPB	mcpb_tip
MGMT GE	fege_tip

### 5.16.1.6 FPI Reshaping

The device implements a flywheel counter for the FPI device input. This counter will generate an internal 8 kHz signal provided that the incoming FPI signal occurs with a period that is an integer multiple of 125  $\mu$ s. For example, a 2 kHz frame pulse that occurs every 500  $\mu$ s is supported. The resulting internal 8 kHz FPI signal is used to synchronize OPSA packet header timestamps carried by data transmitted across the backplane fabric or across the ODUk switch.

The size of the windowing mechanism for the device FPI input is configurable from a minimum of -48.23ns / +38.58ns to a maximum of -626.93ns / +617.28ns. Accuracy of the window is  $\pm 3.215$ ns due to sampling error of the device FPI input.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:11 PM

## 5.16.2 Device Loopbacks

This device supports a variety of loopbacks in order to support both system debugging and field diagnostics. However, not all loopbacks that can be physically enabled will make sense at all levels of hierarchy due to possible protocol restrictions. Other configurations or processing may be required for some loopbacks to function correctly with higher level protocols.

The following subsections outline the Internal, Line, and Diagnostic loopbacks support by this device.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:11 PM

### 5.16.2.1 Internal Loopbacks

Internal loopbacks are points in the device where a received or transmitted signal can be looped back towards its corresponding source. [Figure 164](#) below shows the supported internal loopbacks in the device. Detailed descriptions of each internal loopback can be found in [Table 94](#).

**Figure 164** Device Internal Loopbacks

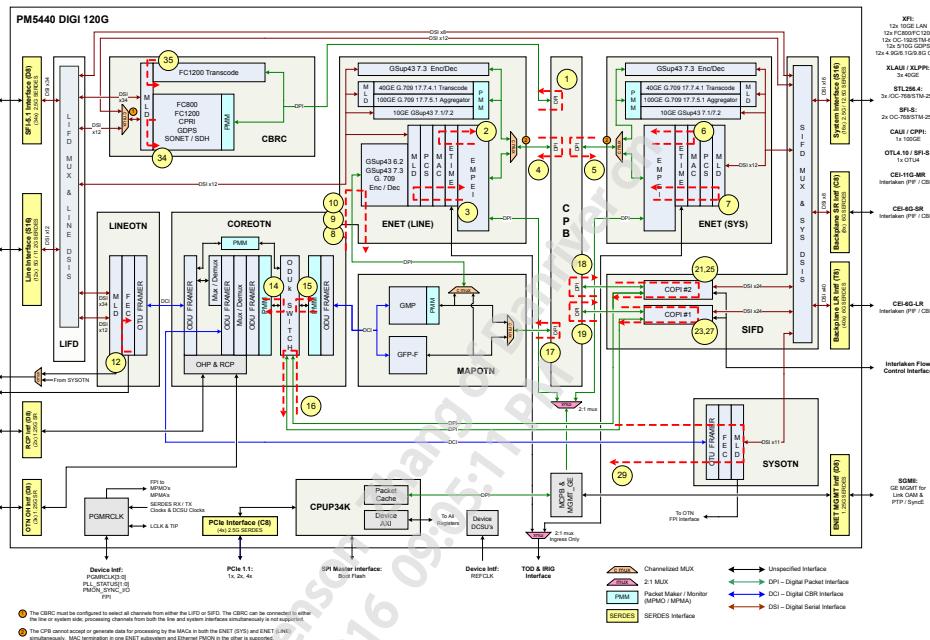


Table 94 Device Internal Loopback Descriptions

Loopback Number	Description
(1)	<b>CPB Loopback to CBRC:</b> CBRC $\leftrightarrow$ CPB $\leftrightarrow$ Loopback
(2)	<b>ENET(LINE) MAC Loopback to CPB:</b> <sup>(A)</sup> CPB $\leftrightarrow$ ENET(Line) $\leftrightarrow$ EMPEI $\leftrightarrow$ ETIME $\leftrightarrow$ MAC $\leftrightarrow$ Loopback
(3)	<b>ENET(LINE) MLD Loopback to CPB:</b> CPB $\leftrightarrow$ ENET(Line) $\leftrightarrow$ EMPEI $\leftrightarrow$ ETIME $\leftrightarrow$ MAC $\leftrightarrow$ PCS $\leftrightarrow$ MLD $\leftrightarrow$ Loopback
(4)	<b>CPB Loopback to ENET(LINE):</b> ENET(Line) $\leftrightarrow$ CPB $\leftrightarrow$ Loopback
(5)	<b>CPB Loopback to ENET(SYS):</b> ENET(SYS) $\leftrightarrow$ CPB $\leftrightarrow$ Loopback
(6)	<b>ENET(SYS) MAC Loopback to CPB:</b> <sup>(A)</sup> CPB $\leftrightarrow$ ENET(Sys) $\leftrightarrow$ EMPEI $\leftrightarrow$ ETIME $\leftrightarrow$ MAC $\leftrightarrow$ Loopback
(7)	<b>ENET(SYS) MLD Loopback to CPB:</b> CPB $\leftrightarrow$ ENET(Sys) $\leftrightarrow$ EMPEI $\leftrightarrow$ ETIME $\leftrightarrow$ MAC $\leftrightarrow$ PCS $\leftrightarrow$ MLD $\leftrightarrow$ Loopback
(8)	<b>ENET(LINE) GSup43 7.3/6.2 Loopback to MAPOTN:</b> <sup>(A)</sup> MAPOTN $\leftrightarrow$ ENET(Line) $\leftrightarrow$ GSup43 7.3/6.2 $\leftrightarrow$ Loopback
(9)	<b>ENET(LINE) G.709 Transcoder Loopback to MAPOTN (40GE):</b> <sup>(A)</sup> MAPOTN $\leftrightarrow$ ENET(Line) $\leftrightarrow$ G.709 Transcoder $\leftrightarrow$ Loopback
(10)	<b>ENET(LINE) GMP Loopback to MAPOTN (10/100GE):</b> <sup>(A)</sup> MAPOTN $\leftrightarrow$ ENET(Line) $\leftrightarrow$ GMP $\leftrightarrow$ Loopback
(11)	<b>Reserved</b>
(12)	<b>MLD Loopback to FEC:</b> <sup>(B)</sup> COREOTN $\leftrightarrow$ LINEOTN $\leftrightarrow$ OTU Framer $\leftrightarrow$ FEC $\leftrightarrow$ Loopback
(14)	<b>COREOTN ODUk Switch Loopback to LINEOTN:</b> LINEOTN $\leftrightarrow$ COREOTN $\leftrightarrow$ ODU Framer $\leftrightarrow$ ODUk SW $\leftrightarrow$ Loopback
(15)	<b>COREOTN ODUk Switch Loopback to MAPOTN:</b> MAPOTN $\leftrightarrow$ ODU Framer $\leftrightarrow$ ODUk SW $\leftrightarrow$ Loopback
(16)	<b>COREOTN ODUk Switch Loopback to SIFD (Port 1&amp;2):</b> SIFD $\leftrightarrow$ COREOTN $\leftrightarrow$ ODUk SW $\leftrightarrow$ Loopback
(17)	<b>CPB Loopback to MAPOTN :</b> MAPOTN $\leftrightarrow$ CPB $\leftrightarrow$ Loopback
(18)	<b>CPB Loopback to SIFD (Port 2):</b> <sup>(A)</sup> SIFD $\leftrightarrow$ CPB $\leftrightarrow$ Loopback
(19)	<b>CPB Loopback to SIFD (Port 1):</b> <sup>(A)</sup> SIFD $\leftrightarrow$ CPB $\leftrightarrow$ Loopback
(20)	<b>Reserved</b>
(21)	<b>SIFD COPI#2 Loopback to CPB at Interlaken Block:</b> <sup>(A,C)</sup> CPB $\leftrightarrow$ SIFD $\leftrightarrow$ COPI#2 $\leftrightarrow$ Interlaken $\leftrightarrow$ Loopback
(22)	<b>Reserved</b>
(23)	<b>SIFD COPI#1 Loopback to CPB at Interlaken Block:</b> <sup>(A,C)</sup> CPB $\leftrightarrow$ SIFD $\leftrightarrow$ COPI#1 $\leftrightarrow$ Interlaken $\leftrightarrow$ Loopback
(24)	<b>Reserved</b>

Loopback Number	Description
(25)	<b>SIFD COPI#2 Loopback to COREOTN at Interlaken Block:</b> <sup>(A,C)</sup> COREOTN ↔ SIFD ↔ COPI#2 ↔ Interlaken ↔ Loopback
(26)	<b>Reserved</b>
(27)	<b>SIFD COPI#1 Loopback to COREOTN at Interlaken Block:</b> <sup>(A,C)</sup> COREOTN ↔ SIFD ↔ COPI#1 ↔ Interlaken ↔ Loopback
(28)	<b>Reserved</b>
(29)	<b>SYSOTN MLD loopback of OTU4 back to ODUk Switch:</b> COREOTN ↔ SYSOTN ↔ OTU Framer ↔ FEC ↔ MLD ↔ Loopback
(30)	<b>Reserved:</b>
(31)	<b>Reserved</b>
(32)	<b>Reserved</b>
(34)	<b>CBRC MLD Diagnostic Loopback:</b> CPB ↔ CBRC ↔ PMM ↔ MLD ↔ Loopback
(35)	<b>CBRC MLD Diagnostic Loopback:</b> CPB ↔ CBRC ↔ FC1200 ↔ MLD ↔ Loopback
<b>Note:</b>	
A. Loopback is for lab diagnostic purposes only.	
B. FEC processing can be optionally bypassed.	
C. COPI can support both COREOTN and CPB traffic simultaneously. If this mode is enabled, the Interlaken loopback will cause both types of traffic to be reflected back to their source (i.e. loopback pairs 21,25 and 23,27 cannot be independently controlled, they are either both enabled or both disabled).	

### 5.16.2.2 Line Loopbacks

Line (or Facility) loopbacks are points in the device where a received signal can be looped back towards its corresponding transmitter. Figure 165 below shows the supported line loopbacks inside the device. Detailed descriptions of each line loopback can be found in Table 95.

Figure 165 Device Line Loopbacks

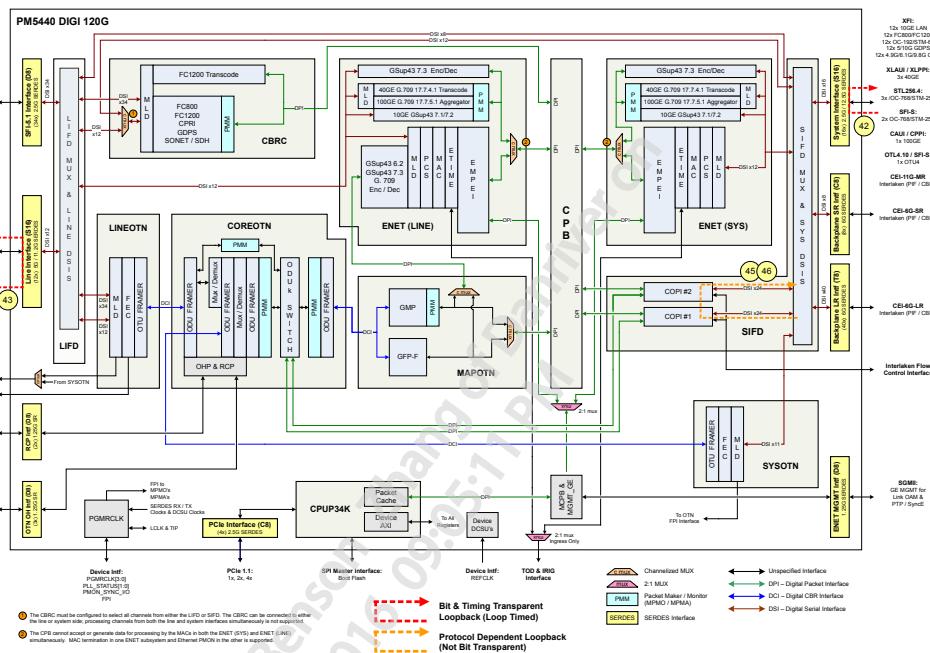


Table 95 Device Line Loopback Descriptions

Loopback Number	Description
(42)	<b>SERDES RX to TX Loopback (System Interface):<sup>([A])</sup></b> SERDES RX ↔ SERDES TX
(43)	<b>SERDES RX to TX Loopback (Line Interface):<sup>([A])</sup></b> SERDES RX ↔ SERDES TX
(45)	<b>SIFD RX to TX Loopback (Interlaken PIF/CBRI):<sup>([B])</sup></b> SIFD ↔ COPI#2 ↔ Loopback
(46)	<b>SIFD RX to TX Loopback (Interlaken PIF/CBRI):<sup>([B])</sup></b> SIFD ↔ COPI#1 ↔ Loopback
<b>Note:</b>	
A. This is an asynchronous loopback, the transmit clock is loop timed to the recovered receive clock.	
B. Loopback is for lab diagnostic purposes only.	

### 5.16.2.3 Diagnostic Loopbacks

Diagnostic loopbacks are the last points in the device where a transmitted signal can be looped back towards its corresponding receiver. [Figure 166](#) below shows the supported diagnostic loopbacks inside the device. Detailed descriptions of each diagnostic loopback can be found in [Table 96](#).

[Figure 166](#) Device Diagnostic Loopbacks

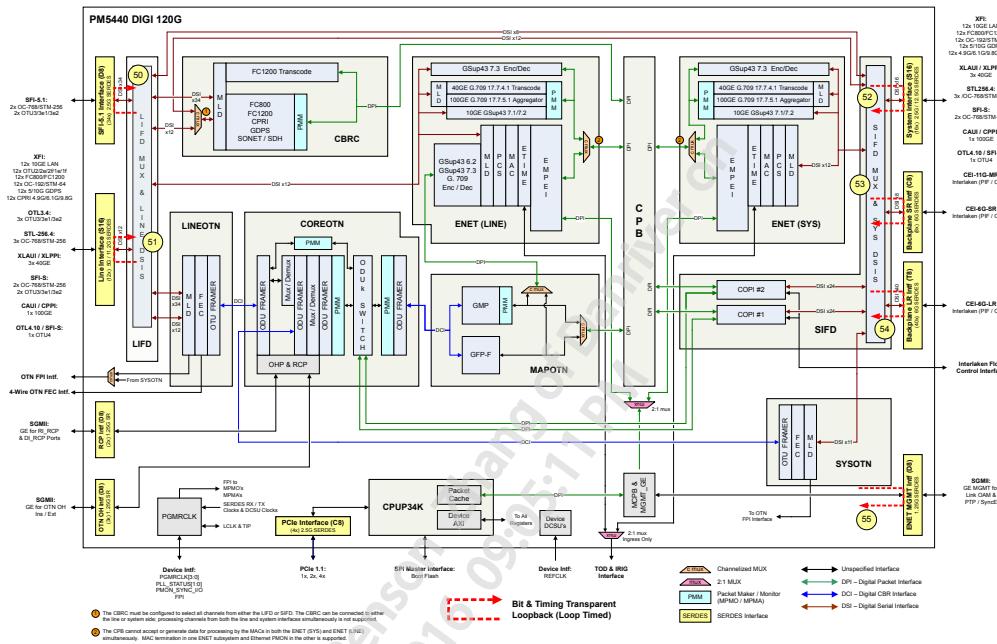


Table 96 Device Diagnostic Loopback Descriptions

Loopback Number	Description
(50)	<b>SFI-5.1 Interface SERDES TX to RX Loopback:<sup>(A)</sup></b> <b>SERDES TX ↔ SERDES RX</b>
(51)	<b>Line Interface SERDES TX to RX Loopback:</b> <b>SERDES TX ↔ SERDES RX</b>
(52)	<b>System Interface SERDES TX to RX Loopback:</b> <b>SERDES TX ↔ SERDES RX</b>
(53)	<b>Backplane Short Range Interface SERDES TX to RX Loopback:<sup>(B)</sup></b> <b>SERDES TX ↔ SERDES RX</b>
(54)	<b>Backplane Long Range Interface SERDES TX to RX Loopback:<sup>(B)</sup></b> <b>SERDES TX ↔ SERDES RX</b>
(55)	<b>Ethernet Management Interface SERDES TX to RX Loopback:<sup>(B)</sup></b> <b>SERDES TX ↔ SERDES RX</b>

**Note:**

A. Data rate changes may cause a one-time disruption of the data stream.

B. Loopback is for lab diagnostic purposes only.

## 5.16.3 Test Pattern Generating and Monitoring

### 5.16.3.1 Unframed Pattern Generation

All of the device SERDES contain built in pattern generators that can create unframed PRBS and user-defined patterns on the outgoing data stream. All patterns can optionally be inverted (all bits negated) and single errors can also be inserted.

Conversely, all of the device SERDES also contain pattern monitors that can check for PRBS and user-defined patterns on the incoming data stream. Each monitor counts the number of mismatches (errors) per word using a register accessible counter.

The supported unframed patterns for each SERDES device interface are summarized in [Table 97](#) below:

**Table 97 Unframed Pattern Generation Supported on SERDES Device Interfaces**

Device Interface	Unframed Patterns Supported
<ul style="list-style-type: none"> <li>Line Interface (S16)</li> <li>System Interface (S16)</li> <li>ENET MGMT Interface (D8)</li> <li>SFI-5.1 Interface (D8)</li> <li>OTN OH Interface (D8)</li> <li>OTN RCP Interface (D8)</li> <li>Backplane LR Interface (T8)</li> </ul>	<ul style="list-style-type: none"> <li>PRBS 7: <math>x^7 + x^6 + 1</math></li> <li>PRBS 9: <math>x^9 + x^5 + 1</math></li> <li>PRBS 23: <math>x^{23} + x^{18} + 1</math></li> <li>PRBS 31: <math>x^{31} + x^{28} + 1</math></li> <li>40-bit user defined patterns</li> </ul>
<ul style="list-style-type: none"> <li>PCIE Interface (C8)</li> <li>Backplane SR Interface (C8)</li> </ul>	<ul style="list-style-type: none"> <li>PRBS 7: <math>x^7 + x^6 + 1</math></li> <li>PRBS 23: <math>x^{23} + x^{18} + 1</math></li> <li>PRBS 31: <math>x^{31} + x^{28} + 1</math></li> <li>40-bit user defined patterns</li> </ul>

### 5.16.3.2 Line and System-side Ethernet

This device supports Test Pattern Generation / Checking for link testing according to IEEE802.3-2008 [23] Clause 49.2.8, Clause 49.2.12 (square wave and PRBS31), and Clause 68-1 (PRBS9). It also implements the optional PMA PRBS9 and PRBS31 generation and checking features for 40GBASE-R and 100GBASE-R as defined in IEEE 802.3ba-2010 [24] Clause 83.5.10. Per lane error counters are provided to count the PRBS31/PRBS9 errors. The device also supports the scrambled idle test pattern generation for 100GBASE-R and 40GBASE-R as defined in IEEE 802.3ba-2010 Clause 82.2.10 and test pattern checking as defined in IEEE 802.3ba-2010 Clause 82.2.17. A Test-pattern error counter is also supported.

Generation of test packets on one channel up to the line rate is supported. Test packets will have constant payload, but programmable DA/SA/length/type and a valid FCS. The channel under test will have a programmable test packet bandwidth utilization percentage, configurable in steps of ~10%. While test packet generation is enabled on one channel, other channels will operate normally.

### 5.16.3.3 SONET and SDH

For testing purposes, OC-192/STM-64 egress streams can be generated containing OC-192c/VC-4-64c PRBS payloads. This PRBS sequence polynomial is  $x^{23} + x^{18} + 1$ . The OC-192/STM-64 stream can be sent on all of the 12 egress SONET/SDH interfaces. PRBS sequence payload generation for 4x STS-192/VC-4-64c framed as OC-768/STM256 is also supported. This stream can be sent out on all of the 3x OC-768/STM-256 interfaces.

One OC-192c/VC-4-64c stream selected from the 12 ingress SONET/SDH OC-192/STM STM-64 streams or one of the 3 OC-768/STM-256 streams can be monitored for PRBS errors in the payload.

#### 5.16.3.4 **Interlaken**

Test pattern generation and checking is supported on the Interlaken Device Interfaces. Two types of pattern generation and checking are supported: a programmable pattern and PRBS31/23/7 patterns.

The programmable pattern generator creates a bit stream of the form Pattern A \* Repetition A, followed by Pattern B \* Repetition B followed by Pattern C \* Repetition C. Pattern A, B, and C are 32-bit patterns. Repetition values are 8-bit values.

The PRBS pattern generator supports the following polynomials:

1. PRBS 31:  $x^{31} + x^{28} + 1$
2. PRBS 23:  $x^{23} + x^{18} + 1$
3. PRBS 7:  $x^7 + x^6 + 1$

Inverted patterns for the above polynomials are also supported.

Statistics counters are provided per lane to count the errors.

#### 5.16.3.5 **OTN**

Individual OPUk PRBS31/PRBS11(inverted and non-inverted) generators and monitors are available for each OTUk stream. Single bit errors can be introduced by the generator. The PRBS monitors provide a counter to perform the pN\_TSE one-second performance monitoring. For the PRBS generation and checking features in the COREOTN, please refer to [Section 5.8.4.13. Mapping of Null client into OPUk](#) is also supported and this mapping is performed in the MAPOTN.

#### 5.16.3.6 **Ethernet Management, OTN RCP and OTN OH Interfaces**

This device supports Test Pattern Generator/Checker for link testing according to Annex 36A, IEEE802.3-2008 [\[23\]](#). The test patterns supported are:

1. High-frequency test pattern
2. Mid-frequency test pattern
3. Low-frequency test pattern

Additionally, inverting PRBS11 pattern and a custom pattern of 10-bits are supported.

## 5.17 Alarms and Consequential Actions

Where not explicitly shown, the response time for consequential actions implemented by device firmware is 19 ms.

### 5.17.1 ENET Consequential Actions Support

This section details the general defect and consequential action mechanism supported by the device for Ethernet clients. The ENET subsystem can be configured to support a number of mapping modes, which differ in their signal failure mechanisms.

The Ethernet mapping modes include:

- 10GE via ITU-T G.Sup43 clause 6.2 (GFP-F)
- 10GE via ITU-T G.Sup43 clause 7.3 (GFP-F)
- 10GE via ITU-T G.Sup43 clause 7.1 (BMP)
- 10GE via ITU-T G.Sup43 clause 7.2 (BMP)
- 40GE via ITU-T G.7041 clause 7.1 (GFP-F)
- 40GE via ITU-T G.709 clause 17.7.4 (GMP)
- 100GE via ITU-T G.7041 clause 7.1 (GFP-F)
- 100GE via ITU-T G.709 clause 17.7.5 (GMP)

Note that the typical firmware response time to the above consequential actions is approximately 100ms.

#### 5.17.1.1 Ethernet as GFP-F (G.Sup43-6.2 / G.7041-7.1)

Similar behavior applies to:

- 10GE via ITU-T G.Sup43 clause 6.2 (GFP-F)
- 40GE via ITU-T G.7041 clause 7.1 (GFP-F)
- 100GE via ITU-T G.7041 clause 7.1 (GFP-F)

In this mapping mode, the Ethernet receive stream is mapped into packets while preserving only the client PDUs (payload-transparent). Note that the GFP mapper will insert GFP idles whenever the CPB is empty.

The following subsections describe the defect set condition mechanisms.

Figure 167 Ethernet as GFP-F Atomic Functions (G.Sup43-6.2 / G.7041-7.1)

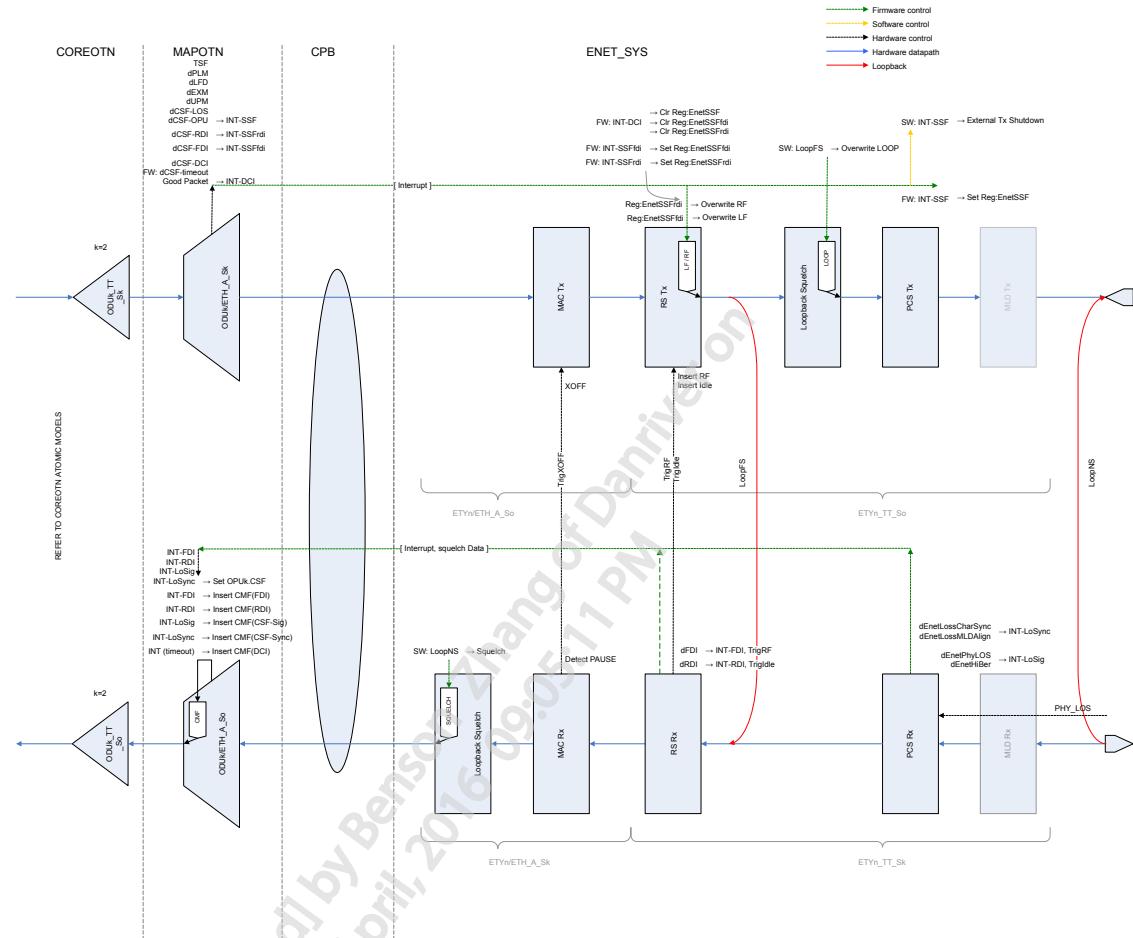
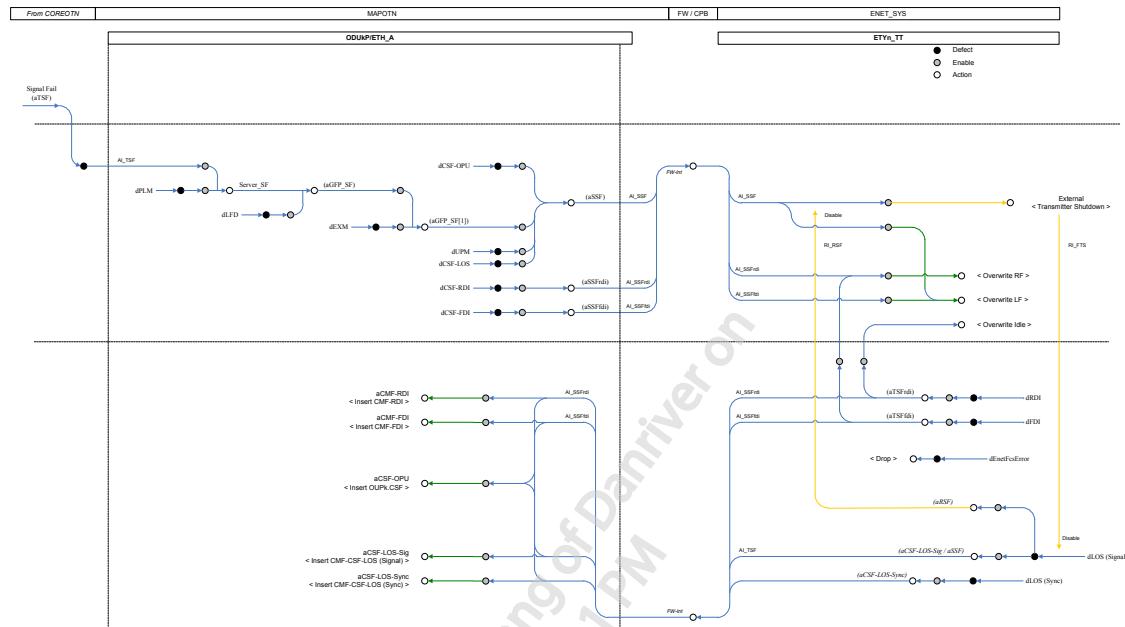


Figure 168 Ethernet as GFP-F Waterfall Diagram (G.SuP43-6.2 / G.709-17.4 / G.7041-7.1)



### 5.17.1.1.1 Client Sink

## Response Due to PHY LOS:

- Upon detecting a loss of signal at the optical module (conveyed through external input pin), the Ethernet receive logic will trigger an interrupt (INT-LoSig).
  - No data will be transmitted from the Ethernet receive logic to the CPB.
  - Firmware will respond to the interrupt and request the insertion of a GFP-CMF (LOS) frame that is mapped into the OPUk.
  - The GFP mapper function can be optionally configured to assert the OPUk.CSF bit in the PSI byte via firmware when the fault occurs.

### **Response Due to PCS Failures:**

- The receive PCS function can detect loss of character (66b) synchronization. Detection of this condition will trigger an interrupt (INT-LoSync).
  - The receive PCS function can detect a high bit-error rate condition. Detection of this condition will trigger an interrupt (INT-LoSync).
  - No data will be transmitted from the Ethernet receive logic to the CPB.
  - Firmware will respond to the interrupt and request the insertion of a GFP-CMF (Loss of Character Sync) frame that is mapped into the OPUk.
  - The GFP mapper function can be optionally configured to assert the OPUk.CSF bit in the PSI byte via firmware when the fault occurs.

### Response Due to Local Fault (LF)/Remote Fault (RF):

- Fault ordered sets(LF/RF) received from the line will be mapped into ordered set GFP frames, preserving the fault condition without consequential action. Note that repeated ordered sets will have 11/12 deleted as required to accommodate internal bandwidth restrictions and this can lead to the alarm not being declared if multiple mapper/demapper nodes are used *or* if the incoming fault density is low at the client interface.

- To avoid the 11/12 deletion of LF/RF ordered sets, firmware can be enabled to generate the propagation of LF and RF upon reception of LF and RF, respectively.
- In the case of LF/RF propagation via FW, any data packets that are accompanying the LF/RF ordered sets will be overwritten, affecting IEEE 802.3 clause 66 unidirectional Ethernet.
- Note that when firmware mode is used for LF/RF propagation, the removal time of the LF/RF alarms will be delayed by up to 19 ms. During this time, all data packets will be lost.
- Also note that in the case where multiple 10GE channels are being mapped into GSup43 C7.3 ODU2 payloads *anda* mix of channels supporting both the legacy UPI values and G7041 standard values is required, LF and RF ordered set propagation must be done with firmware and will be subject to the caveats above.

#### 5.17.1.1.2 Client Source

##### Response Due to OTN Failures:

- The presence of GFP de-mapper defects (dPLM, dLFD, dEXM, dUPM) or TSF from upstream OTN functions will trigger an interrupt.
- Firmware will respond to the interrupt and can optionally configure the Ethernet egress function to overwrite the stream with Local Fault ordered sets. Alternately, firmware can trigger host software to request a LASER transmitter shutdown.
- The above defects can also be read via the RCP port to support customer software/hardware implementing the laser shutdown function.

##### Response Due to GFP CMF-CSF Frames:

- The GFP de-mapper can detect GFP-CMF frames (CSF-LOS) and raise an interrupt to firmware.
- Firmware will respond to the interrupt and can optionally configure the Ethernet egress function to overwrite the stream with Local Fault ordered sets. Alternately, firmware can trigger host software to request a LASER transmitter shutdown.
- The above defects can also be read via the RCP port to support customer software/hardware implementing the laser shutdown function.

##### Response Due to OPUk.CSF:

- The GFP de-mapper can detect the presence of the OPUk.CSF indication and raise an interrupt to firmware.
- Firmware will respond to the interrupt and can optionally configure the Ethernet egress function to overwrite the stream with Local Fault ordered sets. Alternately, firmware can trigger host software to request a LASER transmitter shutdown.
- The above defects can also be read via the RCP port to support customer software/hardware implementing the laser shutdown function.

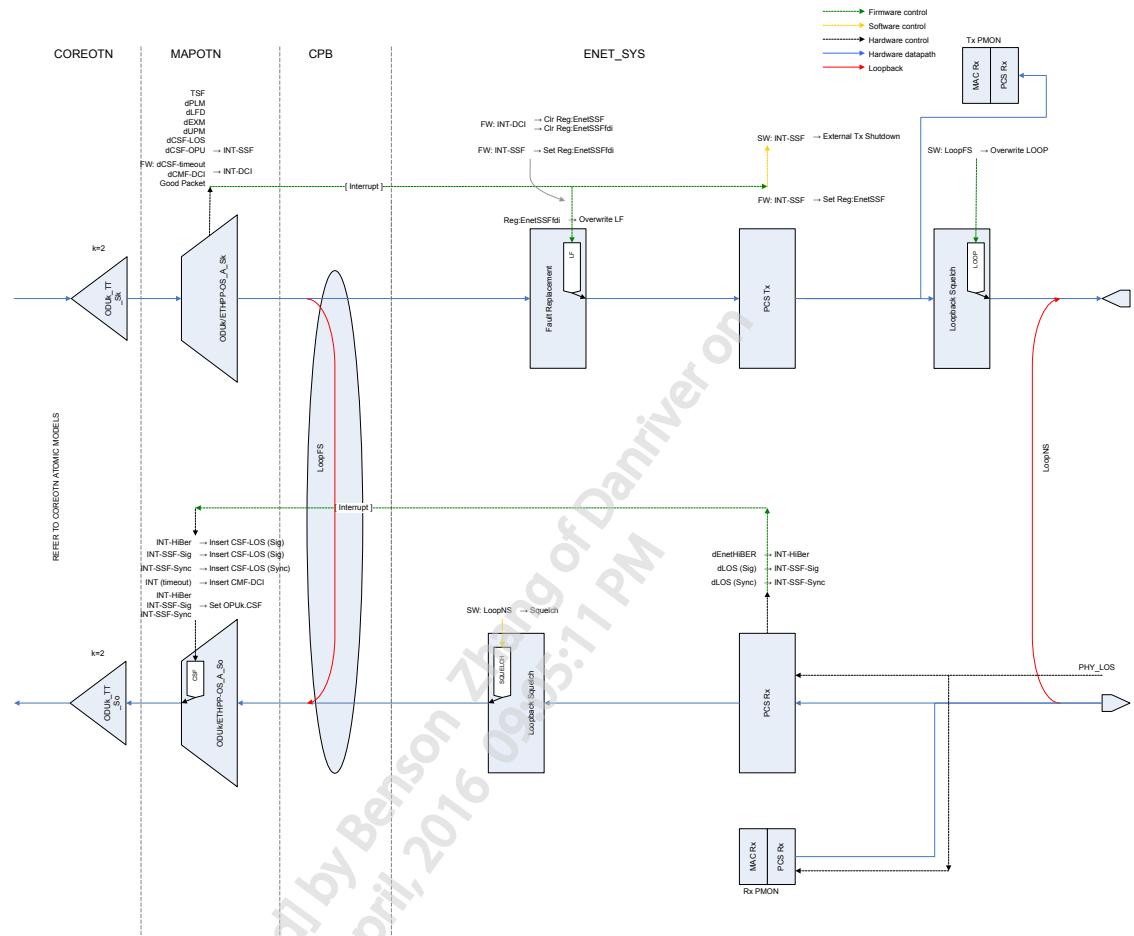
##### Response Due to GFP CMF-FDI/RDI Frames:

- The GFP de-mapper can detect GFP-CMF frames (FDI, RDI) and raise an interrupt to firmware.
- Firmware will respond to the interrupt, decode the type of CMF frame and configure the Ethernet egress function to overwrite the stream with Local Fault or Remote Fault ordered sets.

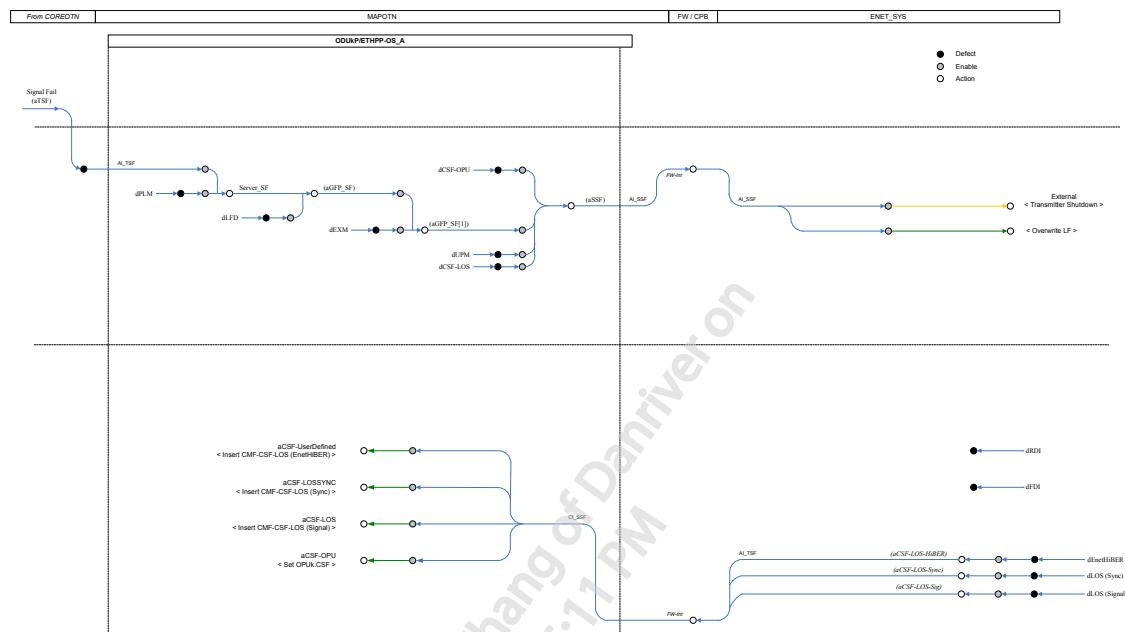
#### 5.17.1.2 Ethernet as PP-OS (G.Sup43-7.3)

In this mapping mode, the Ethernet receive stream is mapped into packets while preserving client PDUs in addition to preamble and ordered sets transparently. Note that the GFP mapper will insert GFP idles whenever the CPB is empty.

Figure 169 Ethernet as PP-OS Atomic Functions(G.Sup43-7.3)



**Figure 170 Ethernet as PP-OS Waterfall Diagram (G.Sup43-7.3 / G.709-17.4.1 / G.7041-7.9 / G.8021-11.5.3)**



The following section describes the defect set condition mechanisms.

### 5.17.1.2.1 Client Sink

#### Response Due to PHY LOS:

- Upon detecting a loss of signal at the optical module (conveyed through external input pin), the Ethernet Rx logic will trigger an interrupt (INT-SSF-LoSig).
- No data will be transmitted from the Ethernet Rx logic to the CPB.
- Firmware will respond to the interrupt and request the insertion of a GFP-CMF (Loss of Signal) frame which is mapped into the OPUk.
- The GFP mapper function can be optionally configured to assert the OPUk.CSF bit in the PSI byte via firmware, when the fault occurs.

#### Response Due to PCS Failures:

- The receive PCS function can detect loss of character (66b) synchronization. Detection of this condition will trigger an interrupt (INT-SSF-LoSync).
- The receive PCS function can detect a high bit-error rate condition. Detection of this condition will trigger an interrupt (INT-SSF-HiBer).
- No data will be transmitted from the Ethernet Rx logic to the CPB.
- Firmware will respond to the loss of character sync interrupt and request the insertion of a GFP-CMF (Loss of Character Sync) frame which is mapped into the OPUk.
- Firmware will respond to the high bit-error rate interrupt and request the insertion of a GFP-CMF (Loss of Signal) frame which is mapped into the OPUk.
- The GFP mapper function can be optionally configured to assert the OPUk.CSF bit in the PSI byte via firmware, when the fault occurs.

#### Response Due to LF/RF:

- Fault ordered sets received from the line will be mapped into ordered set GFP frames, thus preserving the fault condition without consequential action. Note that consecutive ordered sets may be deleted as required to accommodate internal bandwidth restrictions.

#### 5.17.1.2.1.1 Client Source

##### Response Due to OTN Failures:

- The presence of GFP DeMapper defects (dPLM, dLFD, dEXM, dUPM) or TSF from upstream OTN functions will trigger an interrupt.
- Firmware will respond to the interrupt and can optionally configure the Ethernet egress function to overwrite the stream with Local Fault ordered sets. Alternately, firmware can trigger host software to request a LASER transmitter shutdown.
- The above defects can also be read via the RCP port to support customer software/hardware implementing the laser shutdown function.

##### Response Due to GFP CMF-CSF Frames:

- The GFP DeMapper can detect GFP-CMF frames (CSF-LoSignal, CSF-LoSync), and raise an interrupt to firmware.
- Firmware will respond to the interrupt and can optionally configure the Ethernet egress function to overwrite the stream with Local Fault ordered sets. Alternately, firmware can trigger host software to request a LASER transmitter shutdown.
- The above defects can also be read via the RCP port to support customer software/hardware implementing the laser shutdown function.
- The appearance of a valid demapped GFP-F frame can signal to firmware to release the GFP-CMF condition and its associated fault replacement action. The response time for this release will be at most 19 ms. Note that during this time, valid GFP-F frames may be discarded. If frame loss is not desired, it is recommended that the egress fault replacement action be disabled.

##### Response Due to OPUk.CSF:

- The GFP DeMapper can detect the presence of the OPUk.CSF indication and raise an interrupt to firmware.
- Firmware will respond to the interrupt and can optionally configure the Ethernet egress function to overwrite the stream with Local Fault ordered sets. Alternately, firmware can trigger host software to request a LASER transmitter shutdown.
- The above defects can also be read via the RCP port to support customer software/hardware implementing the laser shutdown function.

#### 5.17.1.3 Ethernet as CBR (G.Sup43-7.1/7.2)

Similar behavior applies to:

- 10GE via ITU-T G.Sup43 clause 7.1 (BMP)
- 10GE via ITU-T G.Sup43 clause 7.2 (BMP)
- 40GE via ITU-T G.709 clause 17.7.4 (GMP)
- 100GE via ITU-T G.709 clause 17.7.5 (GMP)

In this mapping mode, an Ethernet receive stream is mapped into a CBR stream with full bit-transparency. The following subsections describe the defect set condition mechanisms.

Figure 171 Ethernet as CBR Atomic Functions (G.Sup43 7.1/7.2)

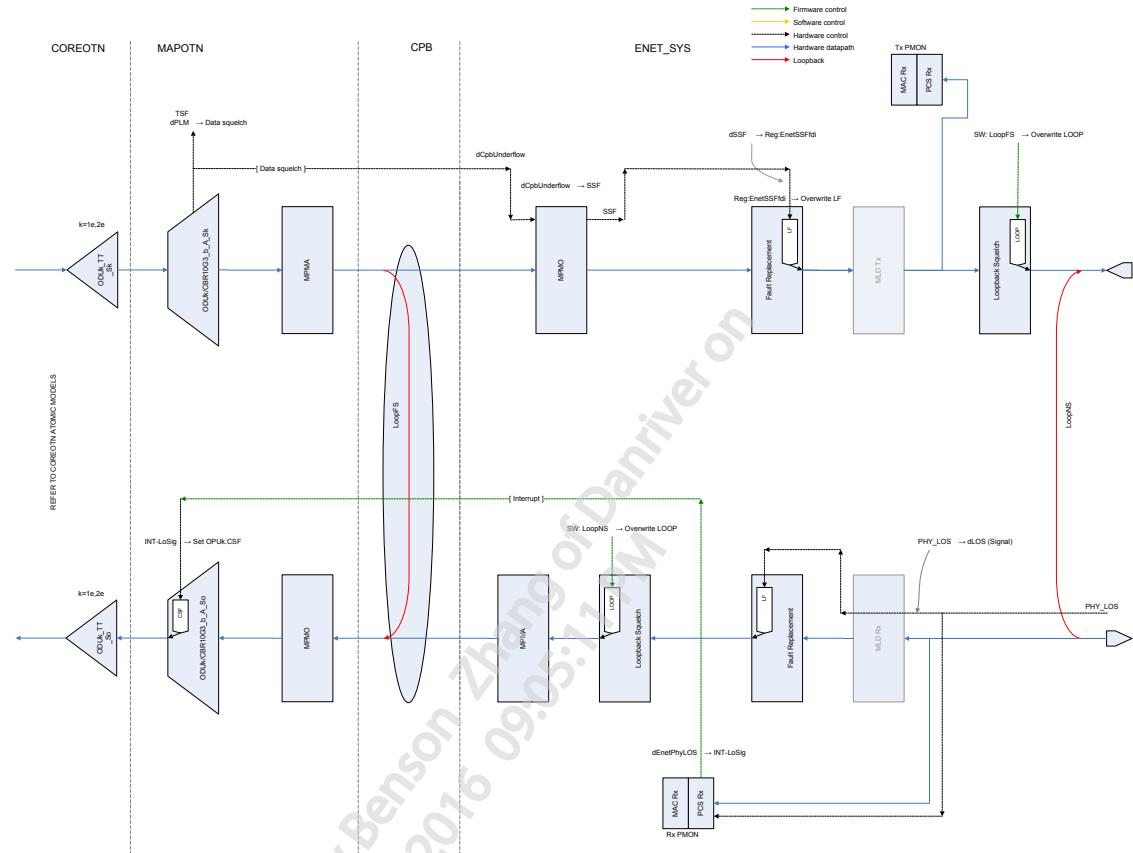
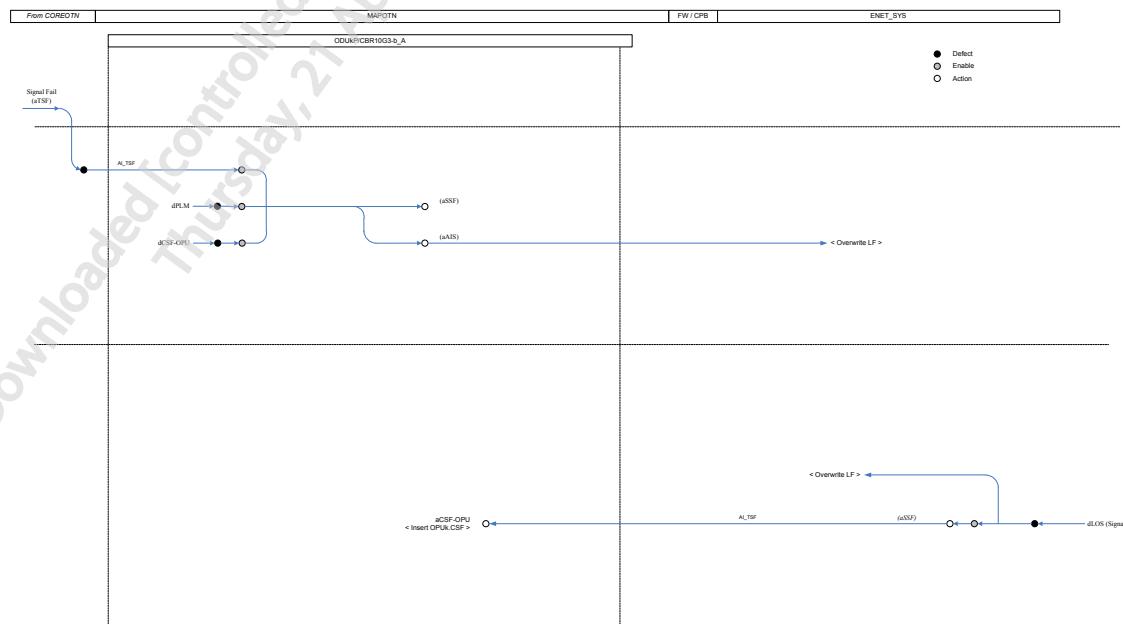


Figure 172 Ethernet as CBR Waterfall Diagram (G.Sup43-7.1 / G.709-17.2.4)



### 5.17.1.3.1 Client Sink

#### Response Due to PHY LOS:

- Upon detecting a loss of signal at the optical module (conveyed through external input pin), the Ethernet receive logic will generate local fault ordered sets at nominal rate, with a Signal Fail (SF) sideband indication.
- The local fault ordered sets will be transmitted from the Ethernet receive logic to the CPB.
- The BMP/GMP mapper function will assert the OPUk.CSF bit in the PSI byte as a result of the SF indication.

### 5.17.1.3.2 Client Source

#### Response Due to OTN Failures:

- The presence of BMP/GMP de-mapper defects (dPLM) or TSF from upstream OTN functions will trigger a data squelch condition.
- No data will be passed from the de-mapper to the Ethernet egress function.
- The Ethernet egress function will detect an underflow condition and interpret this as a Server Signal Fail (SSF).
- The SSF condition triggers the generation of local fault ordered sets at nominal rate.

## 5.17.2 COREOTN Consequential Actions Support

Please see [Section 5.8.4.7](#).

## 5.17.3 LINEOTN Consequential Actions Support

Please see [Section 5.6.3.5](#).

## 5.17.4 SYSOTN Consequential Actions Support

Please see [Section 5.7.3.4](#).

## 5.17.5 MAPOTN Consequential Actions Support

For consequential actions relating to the GFP mapper, please refer to the relevant portions of [Section 5.17.1.1](#). For consequential actions relating to the GMP/BMP mapper, please refer to the relevant portions of [Section 5.17.1.3](#).

## 5.17.6 CBRC Consequential Actions Support

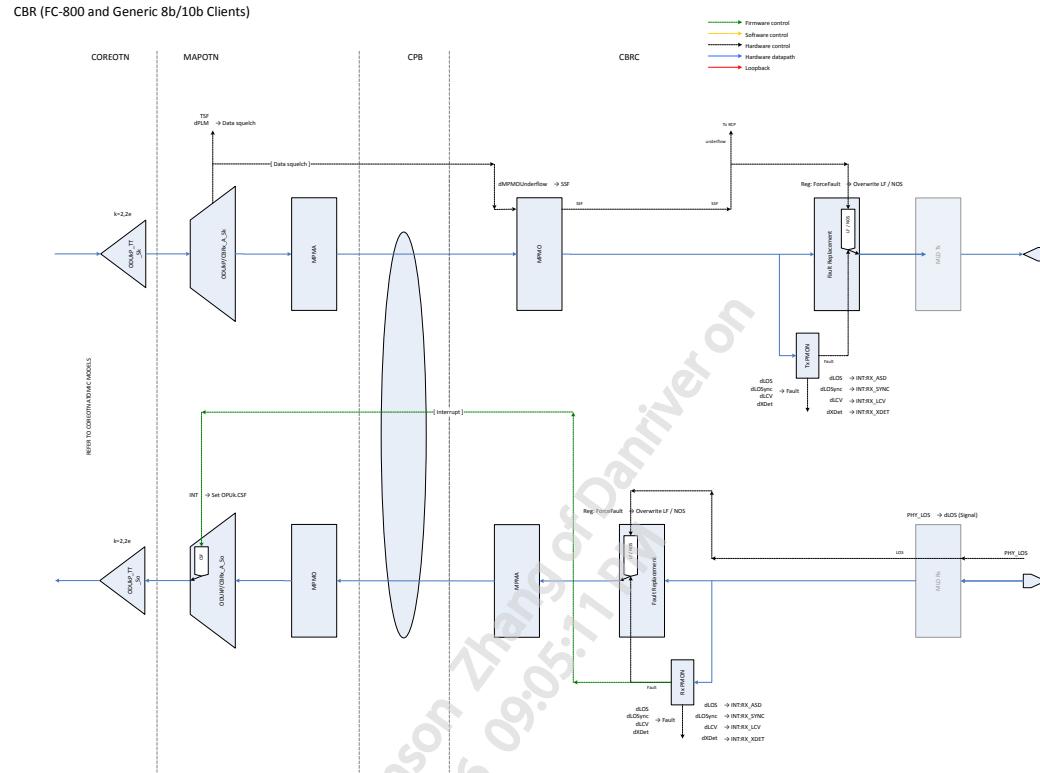
This section details general defect and consequential action support for the following CBR clients:

- FC-800 and Generic 8b/10b
- SONET/SDH
- FC-1200

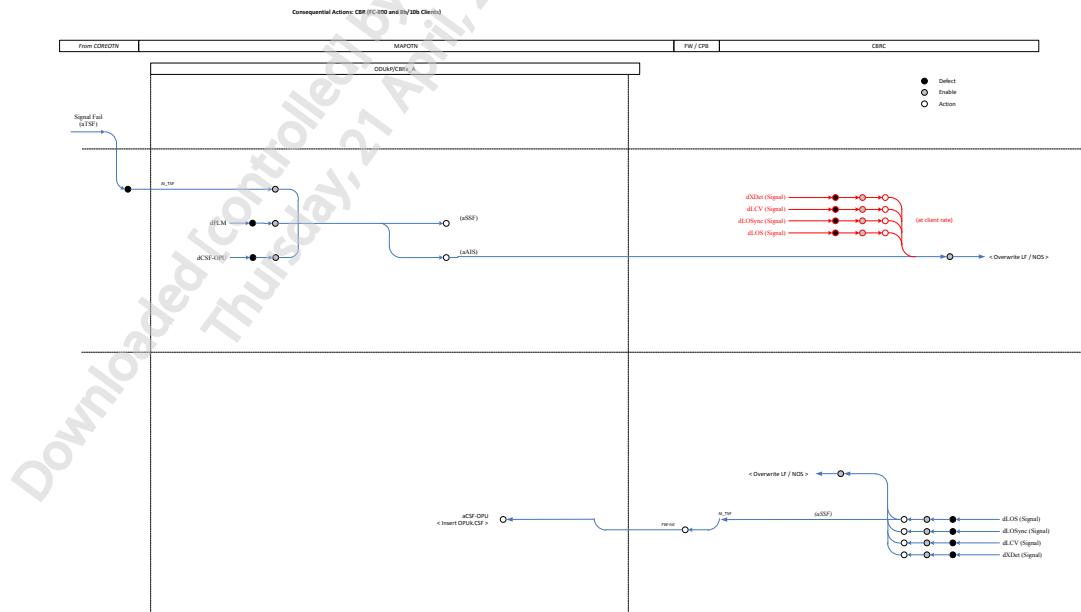
### 5.17.6.1 FC-800 and Generic 8b/10b Clients

See [Section 5.14.3.3.5](#) for an overview of the generation of the 8B/10B error stream.

**Figure 173 FC-800 and Generic 8b/10b Clients**



**Figure 174 Consequential Actions: FC-800 and 8b/10b Clients**



### 5.17.6.2 SONET/SDH Clients

See [Section 5.14.3.3.2](#) for an overview of the generation of the SONET/SDH alarm indication stream.

**Figure 175 SONET/SDH Clients**

CBR (SONET/SDH Clients)

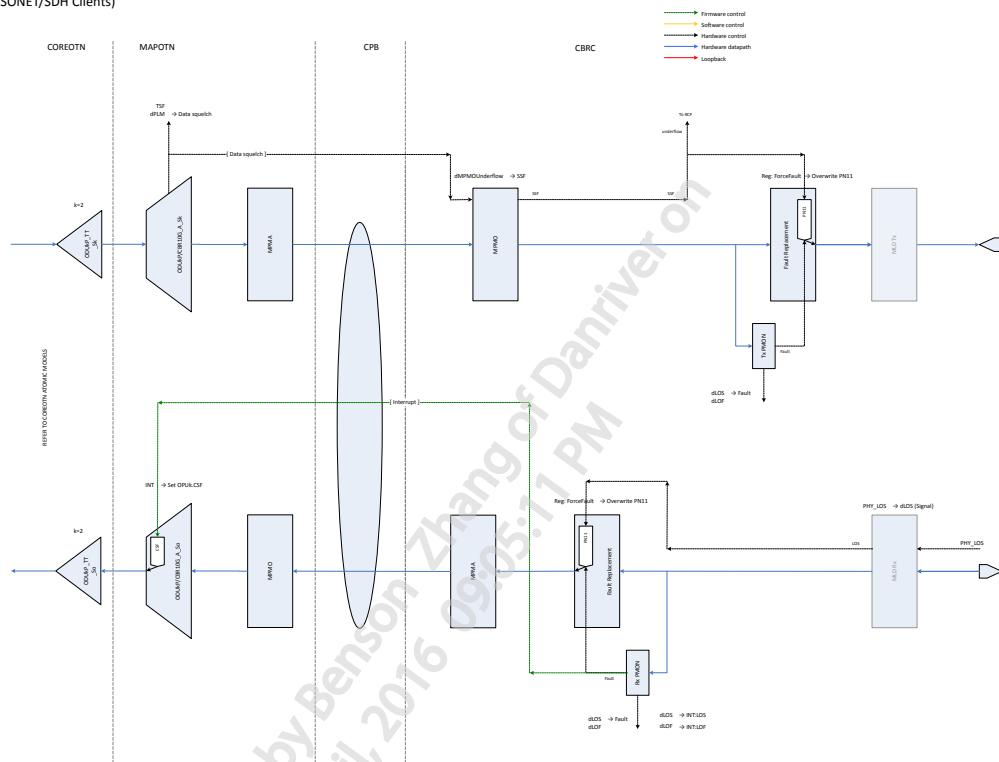
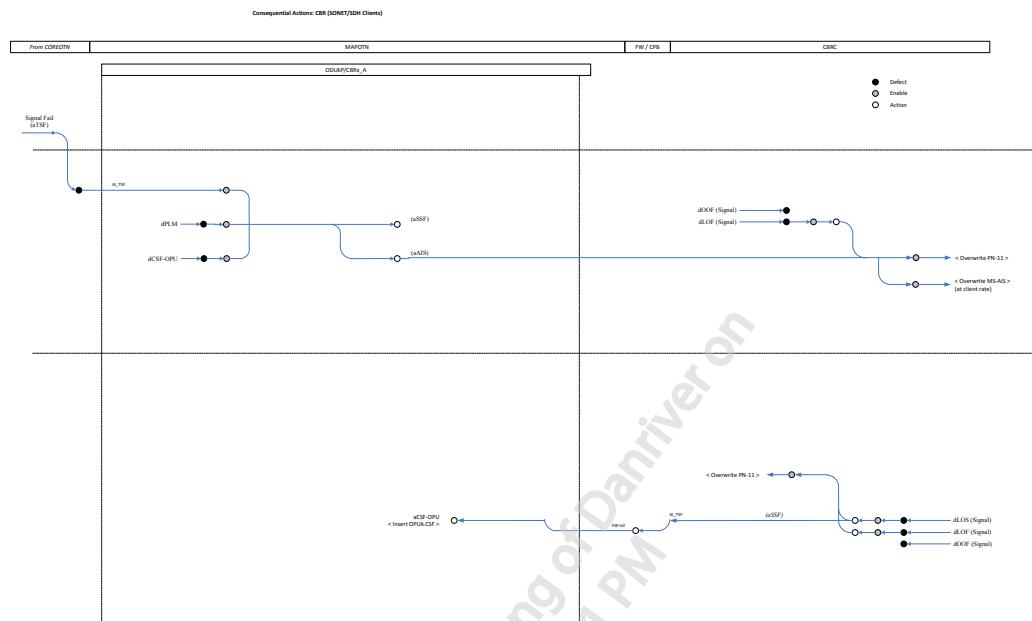


Figure 176 Consequential Actions: SONET/SDH Clients

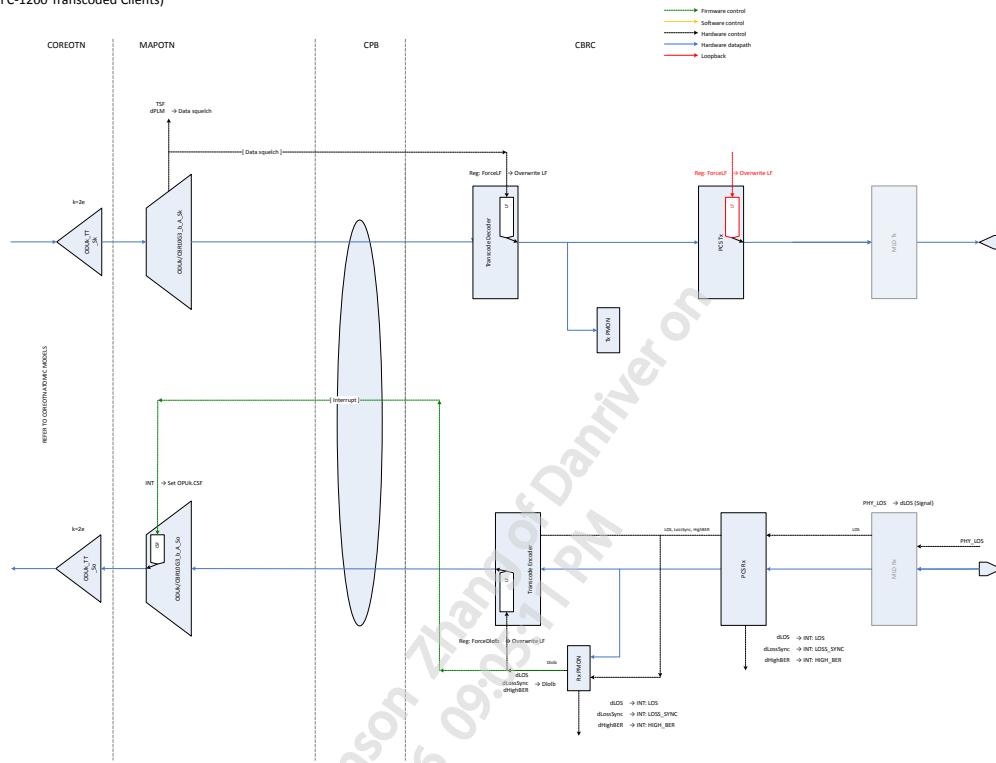


### 5.17.6.3 FC-1200 Clients

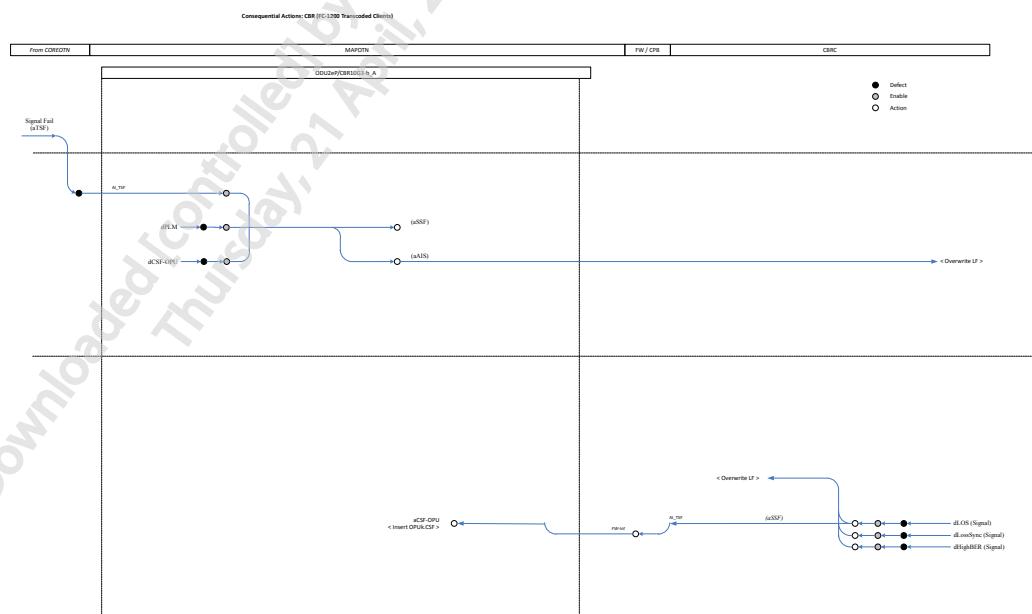
See [Section 5.14.3.3.7](#) for an overview of the FC-1200 ingress and egress stream generation.

Figure 177 FC-1200 Clients

## CBR (FC-1200 Transcoded Clients)



**Figure 178 Consequential Actions: FC-1200 Clients**



## 5.17.7 CBRI Consequential Actions Support And Protection Switching

Protection Switching can be categorized as Facility/Network Protection and Equipment Protection. Examples of Facility/Network Protection are Bridge and Selector changes in response events such as fibre cuts, APS commands, or node failures in the network. Examples of Equipment Protection are Bridge and Selector changes in response to linecard or fabric card failures. One important hallmark of Facility/Network Protection is the parts of the local network element is error-free and can be successfully controlled by system software. Consequently, orderly shut-down and bring-up strategies such as Break-Before-Make can be implemented. Methodical step-by-step strategies are not always possible in the event of Equipment Failures as certain device may become unresponsive to software commands.

At the sink Network Element, Facility/Network protection involves selecting between a Working ODUk and a Protect ODUk. To avoid single point of failure, it is desirable to home the Working and Protect ODUk onto disparate ingress linecards. Selection between the two ODUk streams is usually implemented by the switch fabric. It is the responsibility of the egress linecard to establish the latency of the new active stream. As packet switches often have significant buffering, switching immediately between streams will likely result in the egress card getting Interlaken packets that are a mingle of the original stream and new active stream. A simple way to avoid mixed traffic, and to signal to the egress linecard that a new active stream has been selected is to do a Break-Before-Make. The fabric will first de-select the original stream, wait for all the data from it to have drained from the system, and finally select the new active stream. FIFO underruns at the egress linecards can be used to trigger re-initialization sequences.

The underrun of an ODUk stream at the ODUk Switch is treated as the SSF condition. If the ODUk is configured to enter the COREOTN towards the OTN line, the downstream framer is responsible for insertion of the maintenance signal (ODUk-AIS) at nominal rate. If the ODUk stream is configured to be demapped towards a client interface, the demapper will squelch the resulting stream entering the CPB. The egress CBR client logic then detects the underflow condition and generates the appropriate client replacement signal at nominal rate. Egress packet client logic will insert a replacement signal such as Local Fault.

Equipment Protection is typically implemented using duplicated linecards and fabric cards. The duplicated linecards will normally be connected to Working and Protect ODUk streams. A failure in a linecard can be treated much like a Facility/Network protection. Break-Before-Make implemented by the switch fabric can effect Equipment Protection with simple signaling to the Egress linecard. In the case of a failure in a fabric card, the selector resides in the egress linecard. A Break-Before-Make can still be implemented. The selector will first de-select the original fabric, wait for all the data from it to have drained from the system, and finally select the new active fabric. FIFO underruns at the egress linecard can again be used to trigger re-initialization sequences, resulting in the replacement signal at nominal rate as described earlier.

Break-Before-Make is a simple scheme to pace and signal a Protection Switch event. The DIGI device will naturally recover even if Selector updates are done asynchronously, in an unmanaged manner. In order for the egress linecard to re-initialize the latency of the new active ODUk stream. System software will have to issue a re-initialization command.

## 5.18 Device Latency

Device latency information consists of measured results for end to end datapaths, per datapath segment estimated latencies based on simulations and analysis that has been correlated to measurements, and latencies through the packet buffers (ODUk Switch and CPB).

### 5.18.1 End to End Datapath Latency Examples

Datapath latencies were measured after initial provisioning and settling of timing control loops. Absolute maximum and minimum latencies are expected to be higher or lower prior to settling of the control loop and FIFO depths. The primary contributor to the additional latency is the ODUk Switch and CPB due to their store and forward architecture, and in the case of the ODUk Switch, centering mechanisms used as part of the ODUk timing recovery loop. Refer to [Section 5.18.3](#).

Table 98 End to End Datapath Round Trip Latency Examples

Datapath	Typ (μs)	Notes
10 GbE line <> MAC Terminated <> CPB <> GFP-F <> ODUkSW <> OTU2 line	14.3	2
FC-1200 line <> G.709 Transcode <> CPB <> GMP to ODU2e <> ODUkSW <> OTU2e line	30.4	
OTU2 line <> ODU2 <> ODUk Switch	7.4	
OTU2 line <> ODU2 <> ODUk Switch <> ILKN	7.3	
OTU4 line <> ODU4 <> ODUk Switch	6.6	
OTU4 line <> OTL4.10 <> ODU4 <> ODU2 <> ODUk Switch	10.8	
OTU4 line <> OTL4.10 <> ODU4 <> ODU2 <> ODUk Switch <> ILKN	10.9	
OTU4 line <> OTL4.10 <> ODU4 <> ODU2 <> ODU0 <> ODUk Switch	15.5	
STM-64 line <> CPB <> BMP to ODU2 <> ODUk Switch <> ODU2 <> OTU2 line	31.7	
STM-64 line <> CPB <> BMP to ODU2 <> ODUk Switch <> ILKN	24.0	
10 GbE line <> MAC terminated <> CPB <> GFP-F <> ODUk Switch	7.2	2
10 GbE line <> MAC terminated <> CPB <> GFP-F <> ODUk Switch <> ILKN	7.5	2
10 GbE line <> MAC terminated <> CPB <> GFP-F <> ODUk Switch <> ILKN	23.4	3

**Note:**

1. Software version dev 2.04. Datapaths were manually re-centered where applicable to reflect changes expected in version 3.00 and later of the driver.
2. 64-byte packets.
3. 9600-byte packets.
4. No FEC. Refer to [Table 99](#) for FEC latencies.
5. Run to run latency variation of approximately 1 μs was observed during measurements.

### 5.18.2 Subsystem Latency

Subsystem latencies can be composed to determine end-to-end datapath latency.

Refer to [Table 99](#) for FEC encoder and decoder latencies.

Table 99 FEC Latency

FEC	Rate	Encoder Latency	Decoder Latency
I.4	10G	6 $\mu$ s	61 $\mu$ s
I.4	40G	3 $\mu$ s	15.2 $\mu$ s
I.7	10G	5.76 $\mu$ s	122 $\mu$ s
I.7	40G	1.45 $\mu$ s	33.3 $\mu$ s
RSFEC	10G	8.5 ns	6.76 $\mu$ s
RSFEC	40G	8.5 ns	2.12 $\mu$ s
RSFEC	100G	8.5 ns	0.6 $\mu$ s
SWIZZLE	40G	30 ns	34 $\mu$ s
SWIZZLE reduced latency mode	40G	30 ns	23.36 $\mu$ s
SWIZZLE	100G	104 ns	39 $\mu$ s
SWIZZLE reduced latency mode	100G	104 ns	34.8 $\mu$ s

Refer to [Table 100](#) for datapath segment latencies excluding the CPB and ODUk Switch. As per the end-to-end datapath examples, datapath segment latencies reflect nominal values after initial provisioning and settling of timing control loops. Low and high nominal values are stated when the latency variation exceeds the resolution of the estimate and is statistically relevant.

Table 100 Datapath Segment Latency

Datapath Segment	Min ( $\mu$ s)	Max ( $\mu$ s)	Typ ( $\mu$ s)
<b>CBRC</b>			
CBRC FC1200 Encode	1.7	2.1	1.9
CBRC FC1200 Decode	2.0	2.4	2.2
CBRC 10G CBR Client Encode			0.2
CBRC 10G CBR Client Decode			0.5
<b>CBRI</b>			
CBRI Tx			0.1
CBRI Rx			0.1
<b>ENET(LINE) and ENET(SYS)</b>			
ENET_LINE/SYS Line to CPB 10/40/100 GbE PCS/MAC Rx	—	—	1.3
<b>COREOTN Mux / DeMux</b>			
Mux ODU3 to ODU4	0.4	3.2	1.8
Mux ODU2 to ODU4	0.5	3.2	1.9
Mux ODU1 to ODU4	0.7	3.4	2.1
Mux ODU0 to ODU4	1.0	3.7	2.3
Mux ODU2 to ODU3	0.5	3.2	1.9
Mux ODU1 to ODU3	0.7	3.4	2.1
Mux ODU0 to ODU3	1.0	3.7	2.3
Mux ODU1 to ODU2	0.8	3.5	2.1
Mux ODU0 to ODU2	1.0	3.8	2.4
Mux ODU0 to ODU1	1.2	4.0	2.6
DeMux (any ODUk)	1.9	2.7	2.3

Datapath Segment	Min (μs)	Max (μs)	Typ (μs)
<b>MAPOTN</b>			
AMP/BMP/GMP Map	0.1	3.0	1.8
AMP/BMP/GMP DeMap	0.1	0.5	0.3
GFP-F Map	0.1	2.7	1.9
GFP-F DeMap	0.1	1.3	0.3
<b>LINEOTN</b>			
OTL 4.10 Rx	0.2	0.2	0.2
OTL 4.10 Tx	0.2	0.2	0.2
OTL 3.4 Rx	0.2	0.2	0.2
OTL 3.4 Tx	0.2	0.2	0.2
LINEOTN OTU2 Rx	0.3	0.6	0.4
LINEOTN OTU2 Tx	0.4	0.4	0.4
LINEOTN OTU3 Rx	0.2	0.6	0.4
LINEOTN OTU3 Tx	0.4	0.4	0.4
LINEOTN OTU4 Rx	0.2	0.6	0.4
LINEON OTU4 Tx	0.4	0.4	0.4
<b>Note:</b>			
1. Minimum and maximum estimates are provided when relevant and are theoretical.			

### 5.18.3 ODUk Switch and CPB Latency

The ODUk Switch and CPB are implemented using a store and forward architecture. Latency through the blocks is a function of the packet size, the nominal operating level of the queues (near-empty or near-full), the number of blocks allocated to the queue, latency jitter associated with the number of blocks allocated for internal flow control and buffering specific to the ODUk Switch and CPB usecases.

#### 5.18.3.1 ODUk Switch Latency

In the ingress direction (PMC device to external fabric via Interlaken), the external device is not expected to flow control the PMC device. The typical latency through the ODUk Switch can then be approximated as:

Typical latency =  $B_{nom} \times 8 / (\text{ODUk rate})$

Where:

- $B_{nom}$  = nominal ODUk packet size in bytes
- ODUk rate = rate of the ODUk in bits per second

In the egress direction, for the purpose of latency estimates, all packet delay variation buffering is assumed to be located within the PMC device. Thresholds are nominally configured as follows:

- FIFO centering (for PDV): 50%
- High threshold (XOFF): 87.5%
- Low threshold (XON): 12.5%

Nominal latency through the ODUk Switch occurs at a 50% fill level. Typical latency will vary based on the 87.5% and 12.5% flow control threshold settings with excursions to the maximum and minimum FIFO depth for a worst case flow control scenario. Refer to the ODUk Switch Buffer Size Calculator [12]

to compute the required size of the halt buffer. [Table 101](#) provides example latency values for typical configurations.

**Table 101 ODUk Switch Egress Latency From Interlaken**

Parameter	Units	Value	ODU0	ODU1	ODU2	ODU3	ODU4
PDV Assumed	μs	20	20	20	20	20	20
Calculated Queue Size <sup>1</sup>	blocks		8	15	58	233	604
Size per block	B	768	768	768	768	768	768
Size of queue	B		6144	11520	44544	178944	463872
Queue center position	%	50%	50%	50%	50%	50%	50%
Queue center position	B		3072	5760	22272	89472	231936
ODUk rate	Gbps		1.244	2.499	10.037	40.319	104.794
Latency to center	μs		19.753	18.441	17.751	17.753	17.706
Fence (high)	%	87.50%					
Fence (low)	%	12.50%					
Typical variation	μs		29.63	27.66	26.63	26.63	26.56

**Note:**

1. 12x12.5 Gbps Lane Interlaken, In-band Flow Control.

### 5.18.3.2 CPB Latency

Datapaths that traverse the CPB can be categorized as either datapaths to and from the Interlaken interface or internal datapaths between subsystems. Datapaths between subsystems are used when mapping or demapping clients to OTN. Latency calculations differ depending on the datapath.

For packet traffic to or from the Interlaken interface to the GFP-F mapper or demapper, the latency is a function the packet rate, the size of the packets, and the size of the queues. Additional latency will be incurred by the COPI communal FIFO (approximately 1 μs) and due to XON / XOFF flow control buffering to the mate device. Refer to [Table 102](#) for the intrinsic latency of the CPB to and from MAPOTN GFP-F blocks excluding the COPI communal FIFO and any additional latency due to flow control XON / XOFF buffering. Refer to the CPB Buffer Size Calculator [\[13\]](#) to estimate CPB buffering requirements for flow control.

**Table 102 Packet Interface (Interlaken) To / From MAPOTN GFP Mapper / Demapper CPB**

Client Rate	Packet Size	Mapping <sup>3</sup>				Demapping			
		Queue Size	Center	Transit <sup>1</sup>	Variation <sup>2</sup>	Queue Size	Center	Transit <sup>1</sup>	Variation <sup>2</sup>
[Gbps]	[B]	[B]	[%]	[μs]	[μs]	[B]	[%]	[μs]	[μs]
1.3	64	24576	0	0.4	157.3	24576	0	0.4	157.3
2.5	64	24576	0	0.2	78.6	24576	0	0.2	78.6
10	64	24576	0	0.05	19.7	24576	0	0.05	19.7
40	64	24576	0	0.01	4.9	24576	0	0.01	4.9
100	64	24576	0	0.01	2.0	24576	0	0.01	2.0
1.3	9600	24576	0	61.4	157.3	24576	0	61.4	157.3
2.5	9600	24576	0	30.7	78.6	24576	0	30.7	78.6
10	9600	24576	0	7.7	19.7	24576	0	7.7	19.7
40	9600	24576	0	1.9	7.7	24576	0	1.9	4.9
100	9600	24576	0	0.8	1.9	24576	0	0.8	2.0

Client Rate	Packet Size	Mapping <sup>3</sup>					Demapping						
		Queue Size	Center	Transit <sup>1</sup>	Variation <sup>2</sup>	Queue Size	Center	Transit <sup>1</sup>	Variation <sup>2</sup>				
[Gbps]	[B]	[B]	[%]	[μs]	[μs]	[B]	[%]	[μs]	[μs]				
<b>Note:</b>													
1. Transit time including store and forward delay and average fill level 2. Transit time when full 3. Excludes additional latency for XON/XOFF flow control and approximately 1us of delay / delay variation through the COPI communal FIFO													

### 5.18.3.2.1 Intrinsic Latencies

Latencies for internal datapaths through the CPB differ for the mapping and demapping direction and for different client types. Several parameters impact the latency including the packet size and the centering mechanism.

For MAC-terminated paths, the queues operate near empty. Latency through the CPB is a function of the queue size, client rate and packet size. Refer to [Table 103](#) for example latencies.

**Table 103 MAC Mapping / Demapping Paths (MAC to TGFP-F or RGFP-F to MAC)**

Client Rate	Packet Size	Mapping					Demapping				
		Queue Size	Center	Transit <sup>1</sup>	Variation <sup>2</sup>	Queue Size	Center	Transit <sup>1</sup>	Variation <sup>2</sup>		
[Gbps]	[B]	[B]	[%]	[μs]	[μs]	[B]	[%]	[μs]	[μs]		
10	64	24576	0	0.051	19.7	24576	0	0.067	25.805		
40	64	24576	0	0.013	4.9	24576	0	0.017	6.451		
100	64	24576	0	0.005	2.0	24576	0	0.007	2.580		
10	9600	24576	0	7.696	19.7	24576	0	7.696	19.702		
40	9600	24576	0	1.924	4.9	24576	0	1.924	4.925		
100	9600	24576	0	0.770	2.0	24576	0	0.770	1.970		

**Note:**

- Transit time including store and forward delay and average fill level
- Transit time when full

For GMP/BMP-mapped Ethernet from the ENET(LINE) or ENET(SYS) subsystems, the CPB queues are nominally centered in both the mapping and de-mapping direction. The packetized clients are nominally 240-bytes. Latency is a function of the nominal queue size and the client rate. Refer to [Table 104](#) for latency examples.

**Table 104 BMP/GMP Mapped Ethernet Latencies (Ethernet subsystems to or from CPB)**

Client Rate	Mapping					Demapping				
	Queue Size	Center	Transit <sup>1</sup>	Variation <sup>2</sup>	Queue Size	Center	Transit <sup>1</sup>	Variation <sup>2</sup>		
[Gbps]	[B]	[B]	[%]	[μs]	[μs]	[B]	[%]	[μs]		
10	24576	50	10.0	19.7	24576	50	10.0	19.7		
40	24576	50	2.5	4.9	24576	50	2.5	4.9		
100	24576	50	1.0	2.0	24576	50	1.0	2.0		

**Note:**

- Transit time including store and forward delay and average fill level
- Transit time when full

For GFP-mapped transcoded FC-1200 or GMP/BMP mapped CBR clients from the CBRC subsystem, the CPB queues are nominally centered in the demapping direction and nominally 25% full in the mapping direction. The packetized clients are nominally 48-bytes for FC-1200 and 240 bytes for other clients. Latency is a function of the nominal queue size and the client rate. Refer to [Table 105](#) for latency examples.

**Table 105 GFP-F Mapped FC-1200 and BMP/GMP Mapped CBR Latencies (CBRC Subsystem To or From CPB)**

Client Rate	Mapping				Demapping			
	Queue Size	Center	Transit <sup>1</sup>	Variation <sup>2</sup>	Queue Size	Center	Transit <sup>1</sup>	Variation <sup>2</sup>
[Gbps]	[B]	[B]	[%]	[μs]	[μs]	[B]	[%]	[μs]
10	24576	25	5.0	19.8	24576	50	9.9	19.8
40	24576	25	5.1	19.8	24576	50	10.1	19.8
100	24576	25	1.3	4.9	24576	50	2.5	4.9

**Note:**

1. Transit time including store and forward delay and average fill level
2. Transit time when full

## 5.18.4 Interlaken Flow Control Latencies

The following tables contain the Interlaken flow control latency parameters using inband and out-of-band flow control. Parameters are defined as per the Interlaken Protocol Specification Section 5.8 and [Table 106](#). The numbers assume worst case flow control scheduling.

**Table 106 Interlaken Flow Control Timing Parameter Definitions**

Acronym	Definition
T <sub>SCL</sub>	Status Change Latency
T <sub>STL</sub>	Status Transmit Latency
T <sub>TCL</sub>	Transmitter Control Latency
T <sub>TPL</sub>	Transmitter Pipeline Latency
T <sub>RPL</sub>	Receiver Pipeline Latency

**Table 107 Interlaken In-band Channel Level Flow Control Latency**

Configuration	LMAX-xoff (B)	Ttcl+Ttpl (ns)	Tscl + Tstl + Trpl (ns)	Total (ns)
6x10.3125	7700	1077	2657	3734
8x10.3125	8500	892	2454	3346
12x10.3125	10400	727	2259	2986
14x10.3125	11200	671	2200	2871
15x10.3125	11700	655	2176	2831
16x10.3125	12100	635	2156	2790
14x3.125	9300	1840	3541	5380
16x3.125	10000	1731	3392	5123
24x3.125	12400	1431	3069	4500
10x6.25	8700	1205	2811	4015
12x6.25	9500	1096	2688	3785
14x6.25	10200	1009	2591	3600

Configuration	LMAX-xoff (B)	Ttcl+Ttpl (ns)	Tscl + Tstl + Trpl (ns)	Total (ns)
16x6.25	11000	952	2518	3470
20x6.25	12500	865	2416	3281
24x6.25	14000	808	2354	3161
7x11.5	8300	892	2451	3343
8x11.5	8800	828	2380	3208
11x11.5	10200	698	2230	2928
12x11.5	10700	671	2201	2872
6x12.5	8000	923	2492	3415
8x12.5	9000	779	2328	3107
10x12.5	10000	692	2222	2915
12x12.5	11000	635	2158	2792

Table 108 Interlaken Out-of-band Channel Level Flow Control Latency

Configuration	LMAX-xoff (B)	Ttcl+Ttpl (ns)	Tscl + Tstl + Trpl (ns)	Total (ns)
6x10.3125	8300	1161	2182	3343
8x10.3125	10200	1070	2413	3482
12x10.3125	13100	916	2392	3308
14x10.3125	14500	869	2380	3249
15x10.3125	15300	856	2378	3234
16x10.3125	16000	839	2376	3215
14x3.125	8000	1582	2552	4134
16x3.125	8500	1471	2527	3998
24x3.125	10500	1212	2492	3704
10x6.25	8700	1205	2285	3489
12x6.25	9500	1096	2261	3358
14x6.25	11000	1088	2512	3600
16x6.25	12000	1038	2501	3539
20x6.25	13900	962	2485	3448
24x6.25	15700	906	2481	3386
7x11.5	10000	1075	2408	3483
8x11.5	10800	1016	2399	3414
11x11.5	13200	903	2381	3284
12x11.5	14000	878	2377	3255
6x12.5	9000	1038	2146	3185
8x12.5	11300	978	2388	3366
10x12.5	13100	907	2375	3281
12x12.5	14800	854	2365	3219

Table 109 Interlaken In-band Link Level Flow Control Latency

Configuration	LMAX-xoff (B)	Ttcl+Ttpl (ns)	Tscl + Tstl + Trpl (ns)	Total (ns)
10x12.5	10000	692	2222	2915
12x12.5	11000	635	2158	2792

Tscl and Tstl specify the maximum time it takes for the status of a PMC device packet buffer to be transmitted out the Interlaken flow control interface as XOFF or XON. Trpl specifies the depth of the receiver pipeline from the serial interface of the PMC device to the CPB or ODUK Switch.

Ttcl and Ttpl specify the maximum number of data bytes that will be transmitted by the device after an incoming FIFO status is received from the mate device. The number of bytes, including the 64/67 encoding and control words of the Interlaken Interface, that need to be received by the mate device when the device receives an XOFF can be computed by multiplying the sum of Ttcl and Ttpl by the aggregate rate of the Interlaken Interface. The number of bytes that the mate device would need to buffer (LMAX-xoff) is the number of bytes received without the Interlaken protocol overhead. The numbers are calculated as  $(T_{TCL} + T_{TPL}) * (64 / 67 \text{ for Interlaken line coding}) * (99.8\% \text{ for meta frame efficiency}) * (97\% \text{ for burst control words}) * (\text{raw Interface rate} = \# \text{ of lanes} * \text{bit rate per lane})$ .

The burst control overhead assumes 256B packets. The buffering that needs to be budgeted in the device for a buffer to not overflow when an XOFF is issued to the mate device is the amount of buffering required for  $T_{RPL} + T_{SCL} + T_{STL}$  of the device plus LMAX-xoff of the mate device ( buffering needed for  $T_{TCL} + T_{TPL}$  of the mate device). Similarly, the amount buffering required in the device to ensure a buffer does not run dry before XON takes affect is the amount of buffering required for  $T_{RPL} + T_{SCL} + T_{STL}$  of the device plus LMAX-xon of the mate device (buffering needed for  $T_{RPL} + T_{SCL} + T_{STL}$  of the mate device). Additional buffering requirements are application dependent.

## 5.18.5 PCIe Latencies

See the following table for a summary of PCIe 4x lane latency data.

Table 110 Summary of PCIe 4x Lane Latencies

	Single Write (ns)	Single Read (ns)	Burst Write (ns)	Burst Read (ns)
<b>SPRAM</b>	437	747	532	903
<b>DCPB</b>	577	828	1740	1955
<b>MCPB</b>	577	828	1740	1955
<b>SIFD</b>	567	868	2015	2270
<b>OTN1 DCSU</b>	868	1059	5106	5291
<b>ENET SYS</b>	792	1043	5180	5395
<b>ENET LINE</b>	802	1078	5180	5375
<b>MAPOTN</b>	562	808	1135	1325
<b>COREOTN</b>	552	828	1390	1635
<b>MFEC_GLUE</b>	582	833	1820	2035
<b>SFI_RX</b>	637	883	2010	2215

**Note:**

1. A single write/read consists of 4 bytes whereas a burst write/read consists of 64 bytes.
2. The results in this table are based on simulated 4x lane PCIe transactions.

## 5.18.6 OTN Frame Pulse Interface Latencies

Table 111 shows the TX/RX propagation delay and uncertainty for:

- The start of frame (first FAS bit) at the LINE\_TXO\_P/N[11:0] serial output to a frame pulse on the OTU\_TX\_MFPO[11:0] output.
- The start of frame (first FAS bit) at the LINE\_RXI\_P/N[11:0] serial input to a frame pulse on the OTU\_RX\_MFPO[11:0] output.

**Table 111 OTN Frame Pulse Interface Latencies and Uncertainties**

<b>Scenario</b>	<b>Delay from LINE_TXO_P/N[11:0] to OTU_TX_MFPO[11:0]</b>		<b>Delay from LINE_RXI_P/N[11:0] to OTU_RX_MFPO[11:0]</b>	
	<b>Time (ns)</b>	<b>Serial Clock Cycles</b>	<b>Time (ns)</b>	<b>Serial Clock Cycle</b>
XFI, OTU2	$-32.36 \pm 4.62$	$-346.5 \pm 49.5$	$50.61 \pm 5.14$	$542 \pm 55$
XFI, OTU1e	$-31.36 \pm 4.48$	$-346.5 \pm 49.5$	$49.05 \pm 4.98$	$542 \pm 55$
XFI, OTU2e	$-31.23 \pm 4.47$	$-346.5 \pm 49.5$	$48.85 \pm 4.96$	$542 \pm 55$
OTL3.4, OTU3	$-32.22 \pm 4.61$	$-1386 \pm 198$	$47.42 \pm 5.12$	$2040 \pm 220$
OTL4.10, OTU4	$-30.99 \pm 4.44$	$-3465 \pm 495.5$	$57.06 \pm 4.93$	$6380 \pm 550$

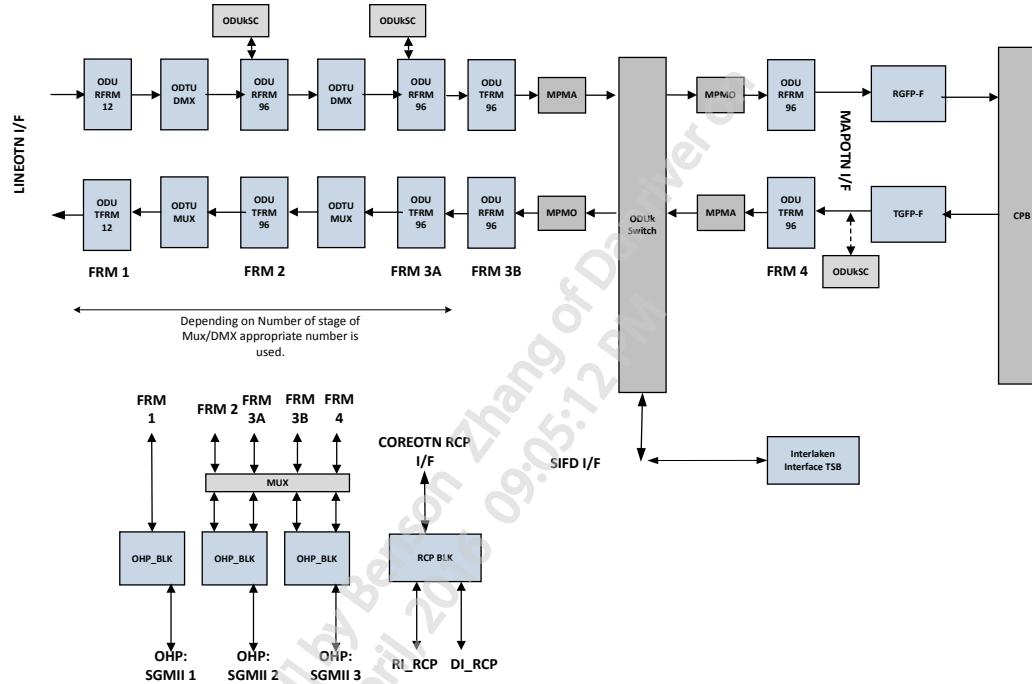
Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:12 PM

## 5.19 Hitless ODUflex Resizing (G.HAO – ITU-T G.7044)

### 5.19.1 Overview

The device supports Hitless ODUflex (GFP) Resizing (G.HAO) as recommended by ITU-T G.7044 [44]. The blocks involved in the G.HAO protocol are shown in Figure 179:

Figure 179 Hitless ODUflex Resizing Block Diagram



The datapaths supported for Hitless resizing are shown in Table 112.

Table 112 Datapath Support

Node	Datapaths
Source/Sink Node	<ol style="list-style-type: none"> <li>MAPOTN (ODUflex.GFP) &lt;=&gt; COREOTN (ODUflex &lt;=&gt; ODUK &lt;=&gt; LINEOTN (ODUK &lt;&gt; OTUK))</li> <li>MAPOTN (ODUflex.GFP) &lt;=&gt; COREOTN (ODUflex) &lt;=&gt; SIFD (ODUflex &lt;=&gt; ILKN)</li> </ol>
Intermediate Node	<ol style="list-style-type: none"> <li>LINEOTN (OTUk &lt;=&gt; ODUK) &lt;=&gt; COREOTN (ODUK &lt;=&gt; ODUflex &lt;=&gt; ODUj) &lt;=&gt; LINEOTN (ODUj &lt;&gt; OTUj)</li> <li>LINEOTN (OTUk &lt;=&gt; ODUk) &lt;=&gt; COREOTN (ODUK &lt;=&gt; ODUflex &lt;=&gt; ODUj) &lt;=&gt; SIFD (ODUj &lt;=&gt; ILKN)</li> </ol>

Note: The datapaths where ODTU Muxing is involved in COREOTN, both 1 stage and 2 stage muxing are supported, but the ODUflex(GFP) is always the Low-Order.

The GFP traffic can either be provided to the MAPOTN by the Interlaken interface (PIF) or by the Ethernet subsystems (ENET(LINE) or ENET(SYS)). As the bandwidth grows or shrinks with ODUflex.GFP, the GFP bandwidth can be utilized appropriately.

## 5.19.2 Features

The device's G.HAO protocol implementation complies with ITU-T G.7044 for protocol definition and ITU-T G.798 for atomic function realization. The features supported by the implementation include:

- Support for ramp up and down of ODUflex GFP into HO ODU4, ODU3/ODU3e2, ODU2. Only in GMP MUXed High Orders.
  - Ramp up from flex. $[>=1]$  to flex. $[<= \text{Max}]$  hitlessly into HO ODUK.
  - Ramp down from flex. $[<= \text{Max}]$  to flex. $[>=1]$  hitlessly into HO ODUK.
  - Full support for Link Connection Resize (LCR) protocol as recommended by ITU-T G.7044.
  - Full support for Bandwidth Resize (BWR) protocol as recommended by ITU-T G.7044.
  - Support for performing reconfiguration (provisioning/deprovisioning) on another channel while running hitless ODUflex resizing.
  - Overhead insertion of G.HAO control plane overhead (RCOH) in both the HO ODUK and ODUflex, including CRC generation.
  - Overhead extraction and interrupt mechanisms of G.HAO control plane overhead (RCOH) in both the HO ODUK and ODUflex, including CRC checking.
  - Maintenance of 1  $\mu$ s latency variation during the ramp up or down of the ODUflex within the system as per G.7044.
  - Support for hitless overhead insertion via the device's OHP and RCP ports during the ramp.
  - Support for multiple ramps as an intermediate node datapath. Includes doing one LCR acknowledgement at a time, but multiple BWR simultaneously.
  - Support for only a single ramp as a source or sink node datapath.
  - Consequential action handling as per G.7044 during the ramp or before the ramp.
  - Ability to use or reduce the ODUflex (GFP) bandwidth as resizing is occurring.
  - Ability to automatically terminate the ramp after target rate is achieved (that is, no upstream BWR IND de-assertion required).

The following subsections highlight the details of the features mentioned above.

### 5.19.2.1 High-Order Resizing Overhead Support

The High Order Resizing OPUk Overhead (RCOH) defined per G.7044 is shown in [Figure 180](#).

Figure 180 HO ODUkflex RCOH

As noted in G.7044, these overheads belong to row 1-3 of column 15 of the OPUk overhead. These are originally part of the CnD field in the GMP process. They are converted to Resizing Overhead (RCOH) when the RP bit is activated by software.

All these overheads can be accessed via the ODTU\_DMX within the device and also can be inserted via the ODTU\_MUX.

Register Access within the MUX is as follows:

- Enable MUX into G.HAO mode: Essentially this stops insertion of Cnd fields and inserts the RCOH from user configured registers for the active tribslot overhead (TSOH) in row 1-3 of column 15 of the OPUk Overhead.
  - Resize Overhead Register (RCOH) per TSOH: This contains the overhead that Software wants to insert in row 1-2 of column 15. Row 3's CRC-5 and CRC-3 is automatically calculated by the Hardware and is inserted.
  - NORM command Trigger: Immediately before the Knife-edge, but after the [ADD, ACK] is sent via the RCOH overhead, the hardware has an automatic method to send [NORM, ACK] per TS and then automatically do LCR resize in the next Resizing Multi-frame boundary [RMF].

Register Access within the DEMUX is as follows:

- Enable DEMUX into G.HAO mode: Similar to the Mux, this bit will stop extraction of Cnd and extract it as RCOH (into a separate register) per TSOH in row 1-3 of column 15 of the OPUk Overhead.
  - Resize Overhead Extraction Register (RCOH) per TSOH: This contains the extracted RCOH after a CRC check is performed. This register has an interrupt associated with it which indicates to software whenever the RCOH is changed from the last stored value.
  - NORM Command Trigger: Similar to the MUX this register is used to put the DEMUX into listening to NORM mode after the [ADD, ACK] is received. Upon receiving NORM the DEMUX will automatically do LCR resize in the next RMF boundary.

### 5.19.2.2 Low-Order Resizing Overhead Support

The Low-Order ODUflex Resizing OPUk Overhead (RCOH) defined per G.7044 is shown in [Figure 181](#).

**Figure 181 LO ODUflex RCOH**

Row	Column	1	2	3	4	5	6	7	8
1	RCOH1	1	BWR_IND	RES	RES		RES		
2	RCOH2	2	BWR_IND	NCS	RES		RES		
3	RCOH3	3	CRC-3				RES		
4	PSI								
QPUflex OH					QPUflex RCOH				

As noted in G.7044, these overheads belong to row 1-3 of column 15 of the OPUflex overhead. These are reserved originally in G.709, but they get converted to Resizing Overhead (RCOH) when OPUflex is in G-HAO mode of operation.

In the device, this overhead can be inserted via the TGFP in the MAPOTN. For extraction, it can be extracted in the RGFP. The BWR\_IND extraction can also be extracted from the ODU\_RFRM stages to automatically propagate that bit to ramp generators within the device (ODUk Scheduler and MPMO blocks) to start the ramp.

Register Access within the TGFP is as follows:

- Enable TGFP into G.HAO mode: Enables the RCOH register to be inserted within Column 15 and row 1-3 of the OPUflex.
  - Resize Overhead Insertion Register (RCOH): This contains RCOH information written via software that needs to be inserted by Hardware. NCS, BWR\_IND and all the reserved fields can be inserted via this register.

Register Access within the RGFP is as follows:

- Enable RGFP into G.HAO mode: Enables the extraction process of the RCOH register from the OP-Uflex overhead.
- Resize Overhead Extraction Register (RCOH): This contains the extracted RCOH after CRC check is performed. This register has an interrupt associated with it which indicates to software whenever the RCOH is changed from the last stored value.

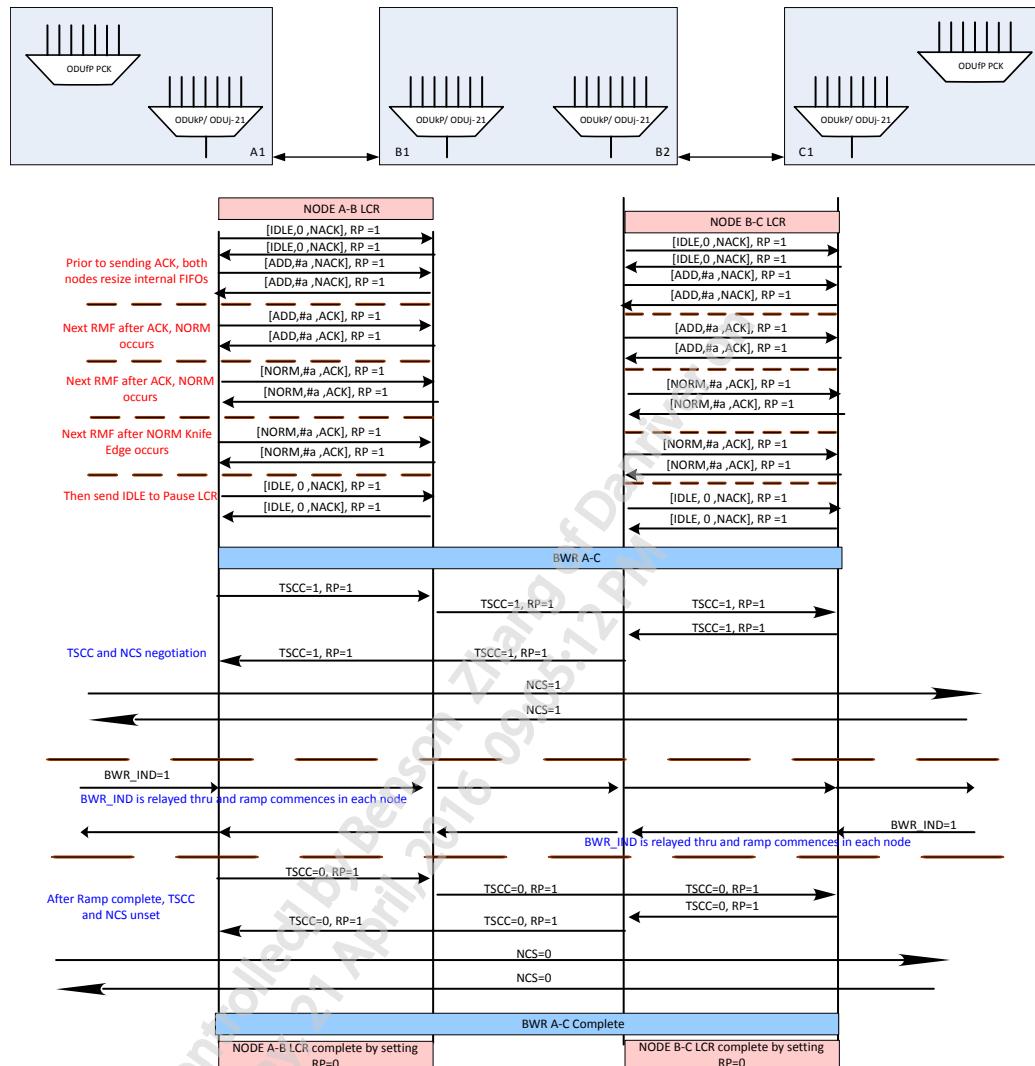
BWR\_IND detection within the ODU\_RFRM:

- There are no configurations required to enable this detection, it is always running to extract BWR\_IND as specified by G.7044 (i.e after the acceptance process). Once extracted it sends the BWR\_IND bit per channel to the ramp generation blocks (MPMO or ODUK Scheduler). If they are set to Ramp mode (i.e listening to BWR\_IND mode) then they will start the ramp process upon receiving the asserted bit. There is an interrupt from the ramp generator blocks to indicate Ramp start or ramp done.
- The applicable ODU\_RFRM's are stage 2, 3A, 3B and 4. The use cases will be discussed in the datapath description sections below.

### 5.19.2.3 Hitless ODUflex Increase

G.HAO increase is handled by first doing LCR to resize the link to the final value and then doing BWR to grow the rate into the link gradually and then finally coming back to LCR to terminate the entire G.HAO protocol. Prior to running the increase protocol all the Nodes in the system should know that a G.HAO protocol is going to be run and should have 'expected' TS growth programmed in their NMS and they should be setup to run the protocol. See [Figure 182](#) for the bandwidth increase sequence diagram.

Figure 182 G.HAO Bandwidth Increase Protocol



Note: In the diagram,  $[x, y, z] = [\text{CTRL}, \text{TPID}, \text{TSGS}]$  (for example  $[\text{NORM}, \#a, \text{NACK}]$ ).

The LCR protocol, assuming single stage muxing for this example, executes as follows:

1. NMS sets up Node A, B and C into G.HAO mode of operation. In turn, each of the nodes' software places the ODTU MUX/DMX within the device into G.HAO mode of operation, where now the Cnd JC columns of the HO are replaced with RCOH overhead, similarly the JC bytes of the LO are RCOH as well.
2. In the HO OPUk, the node's software commences G.HAO by inserting  $[\text{IDLE}, 0, \text{NACK}]$  with RP=1 on the tribslots of the flex and to be added tribslots.
3. LCR is a Node to Node protocol, so Node A to B can negotiate independently of Node B to C. Node A or B can begin the negotiation by Software sending  $[\text{ADD}, \text{TPID}, \text{NACK}]$  on the newly to be added tribslots. Each of the counter-part nodes receive this message and check if this is the new tribslots assigned by NMS and then sends back an acknowledgement as  $[\text{ADD}, \text{TPID}, \text{ACK}]$ . But prior to

sending this acknowledgement, Node A and B's Software should prepare its' FIFOs and configurations to accept the NORM command at any time after this acknowledgement.

4. After resizing its FIFOs and standby tribslot configurations, the ODTU\_MUX sends [ADD, TPID, ACK] and waits to receive the same command from the ODTU\_DMX. Then in the next RMF, the ODTU\_MUX sends [NORM, TPID, ACK], and immediately in the next RMF after, the MUX switches to the new tribslot configuration. The ODTU\_DMX, at the counterpart Node receives the [NORM,TPID,ACK] and will switch over to DEMUX to the new TS configuration automatically in the next RMF.
5. The resize is complete between Node A and B. Node B and C can perform steps 1-4 in parallel/before/after. After completing resize, each nodes' software sends out [IDLE, 0, NACK] on all the TS of the HO where the flex lives in. This ends this portion of LCR.

The BWR protocol executes as follows:

1. After LCR is complete in each node, each node enters BWR by relaying TSCC. BWR is a directional protocol (Node A to C and Node C to A). Each direction can happen independently.
2. In Node A-C direction: Software sets TSCC in ODTU\_MUX after placing the MPMO into GMP special mode with Node A. Node B receives TSCC and enters its MPMO into GMP special mode and then relays TSCC to Node C. Similar action happens independently from Node C to A.
3. Once Node A has TSCC set on both the TX and seen on the RX direction, software sets NCS bit in the OPUflex at the TGFP of Node A. Software then waits to receive NCS bit enabled in the RGFP, which will be initiated by Node C.
4. After receiving the NCS, Node A and C can begin the ramp independently.
5. Node A commences the ramp, by firmware setting the BWR\_IND bit in TGFP and then initiating the ramp (after 125us) using the ODUk Scheduler in the MAPOTN Subsystem.
6. Node B receives the BWR\_IND bit in RX framer stage 3B and sends the bit to the MPMO beside RFRM (Stage 3B), which will begin the ramp after 125  $\mu$ s as per G.7044. Note that the ODUk Scheduler in step 5 and the MPMO in step 6, ramp the rate as per G.7044, at 512000 kbps $^2$ . Node C follows the same in receiving the same BWR\_IND bit in-stream of the ODUflex.
7. Similarly from Node C to A, steps 5 and 6 occur independently.
8. Note that each of the Nodes already has the target rate programmed (within ODUk SC and MPMO). Once the target rate is reached, each of the nodes will end the ramp with an interrupt to Software (this is regardless of the state of BWR\_IND signal).
9. Firmware issues ramp done. NMS will see ramp done and then complete the end of the ramp phase using software, initially at Node A, then takes the Node out of TSCC (i.e. GMP special mode) and signals downstream TSCC=0. Each node then upon receiving TSCC=0, exits GMP special mode and forwards TSCC=0.
10. Once all nodes are out of GMP special mode and TSCC=0. The NCS=0 bit is relayed across from Node A to Node C and then Node C to A independently.
11. After Node A has both TSCC=0 and NCS=0 in RX and TX direction, then it exits BWR.

LCR exit executes as follows:

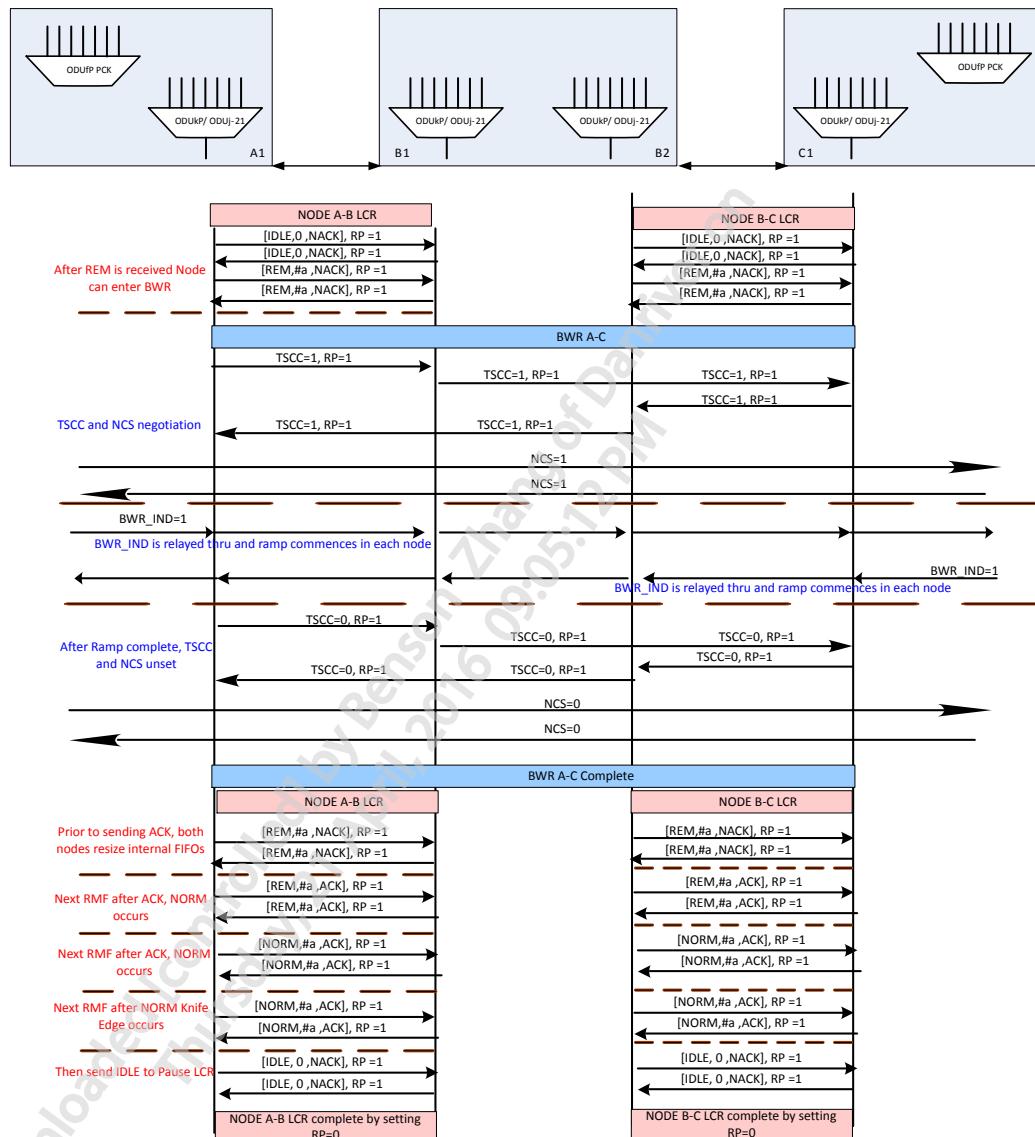
- To wrap up G.HAO, Node A sends RP = 0 to Node B and Node B sends it back to A. Again since LCR is node to node, Node B to C can unset RP independently.

#### 5.19.2.4 Hitless ODUflex Decrease

G.HAO decrease is handled by first doing LCR to start the protocol and then immediately entering BWR to shrink the bandwidth to the final value and then coming back to LCR to hitlessly resize the HO ODUk container prior to finishing the protocol. Prior to running the decrease protocol all the nodes in the

system should know that a G.HAO protocol is going to be run, should have 'expected' TS shrink programmed in their NMS, and they should be setup to run the protocol. See [Figure 183](#) for the bandwidth decrease sequence diagram.

[Figure 183 G.HAO Bandwidth Decrease Protocol](#)



Note: In the diagram,  $[x,y,z] = [\text{CTRL}, \text{TPID}, \text{TSGS}]$  (for example  $[\text{NORM}, \#a, \text{NACK}]$ ).

The LCR protocol, assuming single stage muxing for this example, executes as follows:

1. NMS sets up Node A, B and C into G.HAO mode of operation. In turn each of the nodes' software places the ODTU MUX/DMX within the device into G.HAO mode of operation, where now the CnD JC columns of the HO are replaced with RCOH overhead, similarly the JC bytes of the LO are RCOH as well.
2. In the HO OPUK the nodes' software commences G.HAO by inserting  $[\text{IDLE}, 0, \text{NACK}]$  with RP=1 on the tribslots of the flex and to be removed tribslots.

3. LCR is a Node to Node protocol, so Node A to B can negotiate independently of Node B to C. Node A or B can begin the negotiation by software sending [REM, TPID, NACK] on the to be removed tribslots. Both nodes will receive this remove request and if both nodes are OK with the TS to be removed, then LCR is paused to begin BWR.

The BWR protocol, executes as follows:

1. Each node enters BWR by relaying TSCC. BWR is a directional protocol (Node A to C and Node C to A). Each direction can happen independently.
2. In Node A-C direction: Software sets TSCC in ODTU\_MUX after placing the MPMO into GMP special mode with Node A. Node B receives TSCC in ODTU\_DMX and enters its MPMO into GMP special mode and then relays TSCC to Node C. Similar action happens independently from Node C to A.
3. Once Node A has TSCC set on both the TX and seen on the RX direction, software sets NCS bit in the OPUflex at the TGFP of Node A. Software then waits to receive NCS bit enabled in the RGFP, which will be initiated by Node C.
4. After receiving the NCS bit, Node A and C can begin the ramp down independently.
5. Node A commences the ramp, by firmware setting the BWR\_IND bit in TGFP and then initiating the ramp (after 125us) using the ODUk Scheduler in the MAPOTN subsystem.
6. Node B receives the BWR\_IND bit in RX framer stage 3B and sends the bit to the MPMO beside RFRM (stage 3B), which will begin the ramp after 125us as per G.7044. Note that the ODUk SC in step 5 and the MPMO in step 6, ramp the rate down as per G.7044, at 512000 Kbps<sup>2</sup>. Node C follows the same in receiving the same BWR\_IND bit in-stream of the ODUflex.
7. Similarly from Node C to A, steps 5 and 6 occur independently.
8. Note that each of the Nodes already has the target rate programmed (within ODUk SC and MPMO). Once the target rate is reached, each of the nodes will end the ramp with an interrupt to Software (this is regardless of the state of BWR\_IND signal).
9. Firmware issues ramp done. NMS will see ramp done and then complete the end of the ramp phase using software, initially at Node A, then takes the Node out of TSCC (i.e. GMP special mode) and signals downstream TSCC=0. Each node then upon receiving TSCC=0, exits GMP special mode and forwards TSCC=0.
10. Once all nodes are out of GMP special mode and TSCC=0. The NCS=0 bit is relayed across from Node A to Node C and then Node C to A independently.
11. After Node A has both TSCC=0 and NCS=0, then it exits BWR.

LCR to do knife edge and resize the ODUK container executes as follows:

1. The RP bit is set to 1 but TSCC =0 and NCS in the ODUflex is set to 0 at this stage. The HO ODUK from the MUX in Node A, B and C are still sending [REM, TPID, NACK] on all the removed TS.
2. The nodes will then in the ODTU\_MUX direction send [REM, TPID, ACK], prior to which they will setup all the standby pages and FIFOs for shrinking.
3. After receiving [REM, TPID, ACK] in the ODTU\_DMX, Node A will signal [NORM, TPID, ACK] to Node B and vice-versa. In the next RMF, after NORM, the tribslots to be removed are gone in the ODTU container.
4. At the end software sends [IDLE, 0, NACK] on all TSs from the ODTU\_MUX and is received in the ODTU\_DMX of all nodes appropriately. At which point software unsets the RP bit to 0 from Node to Node to terminate the protocol.

### 5.19.2.5 Datapaths

The various datapaths to conduct G.HAO increase and decrease of the ODUflex are listed below. Most of the protocol for increase and decrease are conducted via Software APIs except for some timing critical changes that are required during the ramp and to setup the ramp, which are done in firmware.

Note in the following datapaths, nodes A, B and C are referenced from [Section 5.19.2.3](#) and [Section 5.19.2.4](#).

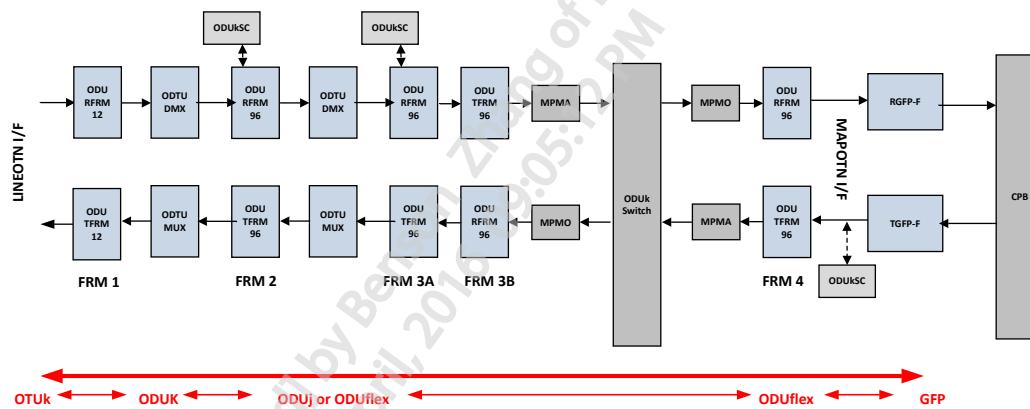
#### 5.19.2.5.1 Source/Sink Node Datapath 1

This datapath involves the following path:

MAPOTN(ODUflex.GFP) <= => COREOTN(ODUflex <= => ODUk) <= => LINEOTN (ODUk <= => OTUK).

This datapath represents Node A or C where both the ODUkP/ODUj-21 and ODUkP/PCK adaptation exists within the same device (that is, not through a fabric). [Figure 184](#) represents this datapath through the device's subsystems.

**Figure 184 LINE\_CORE\_MAP Datapath**



In [Figure 184](#), the flex can either land on ODTU MUX/DMX between framer stage 1-2 (1 stage muxing case) or between framer stage 2-3A (2 stage muxing case).

In this datapath, all the LCR steps outlined in the increase ([Section 5.19.2.3](#)) and decrease ([Section 5.19.2.4](#)) are done using the ODTU\_MUX and ODTU\_DMX.

In BWR, setting of the TSCC is done in the ODTU\_MUX and receiving is done in the ODTU\_DMX. The NCS bit is set in the TGPF and received in the RGFP. GMP special mode is realized by setting the MPMO (close to stage 3B and 4) from OPSA mode to a hold-over mode.

The ramp itself is triggered at the ODUk Scheduler at MAPOTN after setting TGFP to insert the BWR\_IND bit. In the egress direction (towards LINEOTN), RFRM stage 3B picks up the BWR\_IND bit inserted and sends it back to the MPMO (close to stage 3B), which will commence the ramp according to the ramp rate but with tracking the ODUk Switch FIFO fill level. In the ingress direction (from LINEOTN), RFRM stage 4 picks up the BWR\_IND bit inserted and sends it back to the MPMO (close to stage 4), which will commence the ramp according to the ramp rate but with tracking the ODUk Switch FIFO fill level.

Once the ramp is complete, both the MPMO and ODUk Scheduler will issue a 'ramp\_done' interrupt that will allow software to wrap up the G.HAO protocol by finishing LCR using the ODTU MUX and DMX.

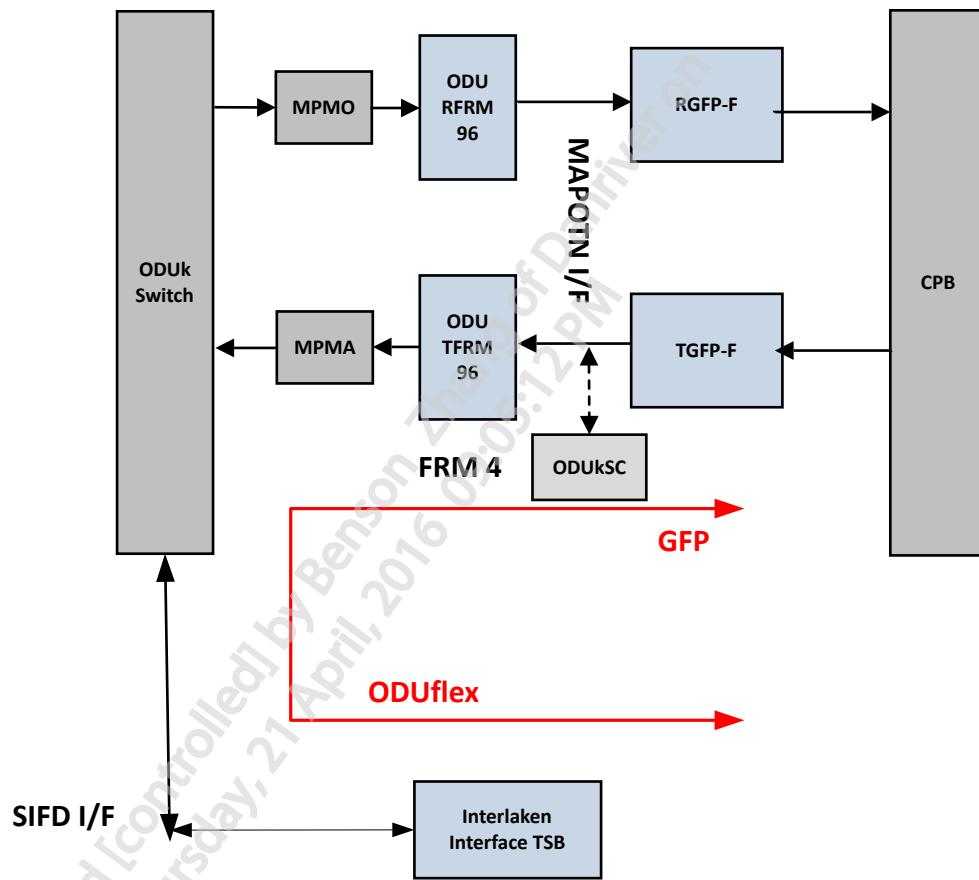
### 5.19.2.5.1.1 Source/Sink Node Datapath 2

This datapath involves the following path:

- MAPOTN(ODUflex.GFP) <= > COREOTN (ODUflex) <= > SIFD (ODUflex <-> ILKN)

This datapath represents Node A or C where the ODUkP/PCK adaptation is represented and the ODUkP/ODUj-21 adaptation is in another DIGI device. Essentially only ODUflex.GFP exists within this device. [Figure 185](#) represents this datapath through the device subsystems:

[Figure 185](#) MAP\_CORE\_SIFD Datapath



The LCR steps for increase and decrease through the Interlaken path are not conducted with this device instead are conducted in node-level software. This device only is aware when G.HAO is entered and when BWR begins. But prior to ramp beginning, the customer must prepare their fabric to expect a ramp (that is, change any rate shapers, FIFOs, schedulers). This is usually done before relaying the NCS bit from this device.

In BWR, the NCS bit is set in the TGFP and received in the RGFP. GMP special mode is realized by setting the MPMO (close to stage 4) from OPSA mode to a hold-over mode.

The ramp itself is triggered at the ODUK Scheduler at MAPOTN after setting TGFP to insert the BWR\_IND bit. In the egress direction (towards the ILKN), no ramp towards ILKN is required; the ODUK SC ramp is sufficient. In the ingress direction (from the ILKN), RFRM stage 4 picks up the BWR\_IND bit inserted

and sends it back to the MPMO (close to stage 4), which will commence the ramp according to the ramp rate, but with tracking of the ODUK Switch FIFO fill level.

Once the ramp is complete, both the MPMO and ODUK Scheduler will issue a 'ramp\_done' interrupt that will allow software to wrap up the G.HAO protocol.

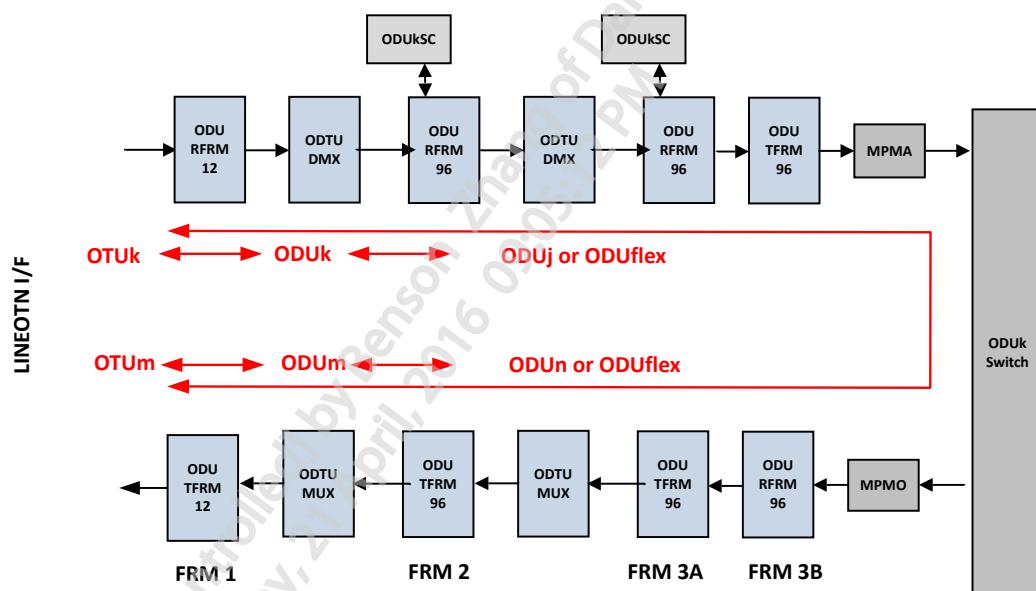
### Intermediate Node Datapath 1

This datapath involves the following path:

LINEOTN (OTUk <= > ODUk) <= > COREOTN(ODUK <= > ODUflex <= > ODUj) <= > LINEOTN (ODUj <= > OTUj)

This datapath represents Node B, where both B1 and B2 of ODUkP/ODUj-21 adaptation exists within one device. In this datapath, the highest ODUk order supported is ODU3/3e2, since two ODU4s cannot be active in the LINEOTN. [Figure 186](#) represents this datapath through the device subsystems.

Figure 186 LINE\_CORE\_LINE Datapath



The LCR of each B1 and B2 adaptation can be conducted independently in the datapath through different channels within the ODTU\_MUX and DMX. Also, there can be a mixture of one stage or two stage muxing to realize this datapath.

In BWR, setting of the TSCC is done in the ODTU\_MUX for each channel and receiving is done in the ODTU\_DMX for the counterpart channel. GMP special mode is realized by setting the MPMO (close to stage 3B) from OPSA mode to a hold-over mode for each of the specific channels after the TSCC is received.

The ramp itself is triggered by receiving the BWR\_IND bit in the ODUflex at stage 3B RFRM and relaying it back to MPMO (beside stage 3B). Once the MPMO receives BWR\_IND separately for each channel, it will commence the ramp at the ramp rate towards the ODTU\_MUX, with also monitoring the ODUK Switch FIFO fill level.

Once the ramp is complete, the MPMO will issue a 'ramp\_done' interrupt which will allow software to wrap up the G.HAO protocol by finishing LCR using the ODTU\_MUX and DMX.

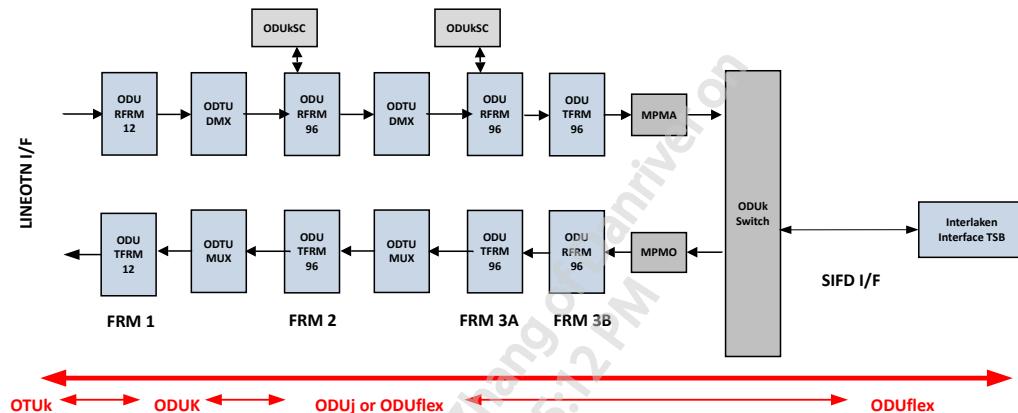
### Intermediate Node Datapath 2

This datapath involves the following path:

- LINEOTN(OTUK <=> ODUK) <=> COREOTN(ODUK <=> ODUflex) <=> SIFD(ODUflex <=> ILKN)

This datapath represents either Node A, B or C, where the ODUkP/ODUj-21 adaptation function is directly connected to the fabric rather than directly to the source/sink of the ODUkP/PCK adaptation function. [Figure 187](#) represents the datapath through the device subsystems:

[Figure 187](#) LINE\_CORE\_SIFD Datapath



The LCR steps for increase/decrease are done through the ODTU\_MUX and ODTU\_DMX, and from an ILKN interface perspective, the fabric has to resize as well appropriately in LCR steps. The potential opportunity to resize the fabric is after the LCR knife edge of the ODTU\_DMX and ODTU\_MUX.

In BWR, setting of TSCC is done in the ODTU\_MUX and receiving is done in the ODTU\_DMX. GMP special mode is realized by setting the MPMO (beside stage 3B) into hold-over.

The ramp in the egress direction (towards the LINEOTN), is triggered by receiving the BWR\_IND bit in the ODUflex at stage 3B RFRM and relaying it back to MPMO (beside stage 3B). In the ingress direction (from the LINEOTN), there is no need for the device to generate the ramp if communicating via the ILKN (the far-end DIGI will ramp appropriately). Once the MPMO receives BWR\_IND, it will commence the ramp at the ramp rate, with also monitoring the ODUk Switch FIFO fill level.

Once the ramp is complete, the MPMO will issue a 'ramp\_done' interrupt which will allow software to complete the G.HAO protocol by finishing LCR using the ODTU\_MUX and DMX.

#### 5.19.2.6 Consequential Action Handling During G.HAO

Per clause 7.1 of Amd1 (02/2012) G.7044, dTIM related consequential actions can be disabled (via the TCMActDis function) in the RFRM stage 2 or 3A or 3B or 4, depending on which RFRM the ODUflex is processed. This should be done via a software API before the protocol begins.

Per clause 8 of Amd1 (02/2012) G.7044, there is maintenance signal insertion capability in the device of the ODUflex in BWR during the ramp during failure state. The ODUk Schedule in the COREOTN begins the ramp after receiving the BWR\_IND signal from RFRM Stage 2 or 3A (depending on which stage MUX is used). If a signal fail occurs from the HO, the signal fail will indicate to the ODUk Scheduler to drive the ramp instead of using the incoming rate of the ODUflex from the DMX. The RFRM stage 2 or 3A will use the same signal fail indicator from the server to insert maintenance signal (ODUk-AIS) on the ramping rate from the ODUk Scheduler.

If the failure occurs at the ODUflex level itself during BWR\_IND phase, then the rate out of the DMX is used to insert maintenance signal (ODUk-AIS).

In the case of fabric failure during the ramp part of BWR protocol, the egress card will hold onto the current ramp rate and insert maintenance signal at this held rate until the fabric recovers. Since this is a rare scenario, this is not expected to happen during G.HAO.

If the server fails (HO ODUk) prior to the ramp or prior to BWR\_IND signal starting, the G.HAO protocol can be terminated by the NMS and restarted when it is appropriate to do so.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:12 PM

## 5.20 Enhanced PMON Modes

The ENET subsystem's enhanced PMONs allows use of the EMAC receive PMONs when none are available on the datapath. Use the ENET EMAC receive PMONs instead of the EMAC/EPMM transmit PMONs because they offer a more complete collection of statistics. Ingress PMONs are fully supported, while egress PMONs are partially supported.

The datapaths supported through Enhanced PMONs are presented below. Unless explicitly defined, the different mapping modes (GFP, GMP, BMP, ...) and rates (10G, 40G, 100G) are available for all datapaths and PMON paths presented. Paths that naturally include an EMAC Rx are not presented.

For more information about the enhanced PMONs, see [Section 5.12.3.9](#).

### Egress Enhanced PMONs

The egress enhanced PMONs are as follows:

**Datapath:** SERDES Rx -> OTN -> Demap -> DCPB (broadcast) -> ENET\_LINE -> SERDES Tx

**PMON:** SERDES Rx -> OTN -> Demap -> DCPB (broadcast) -> ENET\_SYS (Tx to Rx)

**Datapath:** SERDES Rx -> OTN -> Demap (broadcast) -> DCPB -> ENET\_SYS -> SERDES Tx

**PMON:** SERDES Rx -> OTN -> Demap (broadcast) -> ETRANS -> EMAC Rx

**Datapath<sup>1</sup>:** SERDES Rx -> PIF -> DCPB (broadcast) -> ENET\_LINE -> SERDES Tx

**PMON:** SERDES Rx -> PIF -> DCPB (broadcast) -> ENET\_SYS (Tx to Rx)

Note:

1. The SERDES Rx -> PIF -> DCPB (broadcast) -> ENET\_LINE -> SERDES Tx datapath is not supported when configured in 10GE GSUP43 7.3 mode.

**Datapath:** SERDES Rx -> PIF -> DCPB (broadcast) -> ENET\_SYS -> SERDES Tx

**PMON:** SERDES Rx -> PIF -> DCPB (broadcast) -> ENET\_LINE (Tx to Rx)

### Ingress Enhanced PMONs

The ingress enhanced PMONs are as follows:

**Datapath:** SERDES Rx -> OTN -> GFPDemap(PPOS) (broadcast) -> DCPB -> PIF(PPOS) -> SERDES Tx

**PMON:** SERDES Rx -> OTN -> GFPDemap(PPOS) (broadcast) -> ETRANS(10GE GSUP43 7.3) -> EMAC Rx

**Datapath:** SERDES Rx -> ENET\_LINE(EPMM) -> DCPB -> \* (any datapath)

**PMON:** SERDES Rx -> ENET\_LINE(EMAC) Rx

**Datapath:** SERDES Rx -> ENET\_SYS(EPMM) -> DCPB -> \* (any datapath)

**PMON:** SERDES Rx -> ENET\_SYS(EMAC) Rx

## 5.21

## ODUk Switch and CPB Queue Sizing

The CPB and ODUk Switch share a common store-forward packet switch architecture. The CPB has a total of six DPI connections providing connectivity to the following subsystems:

- CBRC
- ENET(LINE)
- MAPOTN
- SIFD (PORT 1)
- SIFD (PORT 2)
- ENET(SYS)

The ODUk Switch has a total of four DPI connections providing connectivity to the following subsystems:

- COREOTN
- SIFD (PORT 1)
- SIFD (PORT 2)
- MAPOTN

Both of the CPB and ODUk Switch containan array (or pool) of memory "pages" that are assigned to a "queue" structure associated with each DPI interface ingress/egress pair. Queues are created by composing a "link list" of the memory "blocks" contained within the individual pages. Note that the size of a queue can be dynamically changed when required.

Queue: Pages >> Blocks

Maximum CPB buffering capacity:

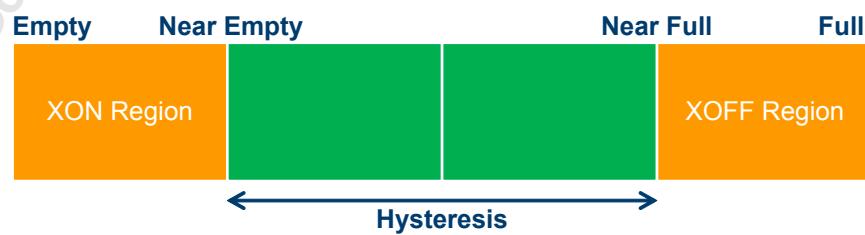
- Total of 32 pages with 16 blocks per page
- Block size is 12 kB
- Page size is 16 blocks \* 12 kB = 192 kB
- Total page memory is 32 pages \* 192 kB = 6.144 MB

Maximum ODUk Switch buffering capacity:

- Total of 22 pages with 64 blocks per page
- Block size is 0.75 kB
- Page size is 64 blocks \* 0.75 kB = 48 kB
- Total page memory is 22 pages \* 48 kB = 1.056 MB

The queue size of the CPB and ODUk Switch (or buffer size) is a function of the flow control latency, client rate, the interlaken interface, and buffering required for internal purposes. The configured queue size must be large enough to ensure that the queue does run dry or overflow after taking these variables into account.

Figure 188 Queue XON, XOFF, and Hysteresis Regions



The configured queue size can be broken up into an XON, XOFF, and Hysteresis region as seen in [Figure 188](#). When the queue fill level crosses the "near empty" empty threshold, XON is sent to the far end to start the flow of data. The queue must have enough headroom to ensure that it does not run dry while waiting for the far end to start sending data. When the queue crosses the "near full" threshold, XOFF is sent to the far end to stop sending data. The queue must have enough headroom to ensure that it does not overflow while waiting for the far end to stop sending data.

For the Interlaken flow control interface,  $T_{tcl}$  and  $T_{tpl}$  are the latency values of the device that sources data whereas  $T_{scl}$ ,  $T_{stl}$ , and  $T_{rpl}$  are the latency values of the device that sinks data. The Interlaken flow control latencies can be found in [Section 5.18.4](#).

$$\text{XOFF / XON delay} = ((\text{near end}) + (\text{far end})) * \text{data\_rate} = ((T_{scl} + T_{stl} + T_{rpl}) + (T_{tcl} + T_{tpl})) * \text{data\_rate}$$

$$\text{Queue Size} = (\text{XON delay} + \text{XOFF delay}) + (\text{headroom})$$

For further information on the CPB and ODUk Switch see [Section 5.10](#) and [Section 5.8.5](#) respectively.

Note that in the following subsections ILKN refers to the Interlaken block located within the two COPI instances in the SIFD subsystem ([Section 5.11.4.5](#)).

## 5.21.1 ODUk Switch Queue Size Calculation

Use the CPB Queue Size Calculation Spreadsheet [\[12\]](#). Values to enter are in yellow and the results of interest are in green. Note that overlapping internal buffering requirements are in blue.

### 5.21.1.1 MAPOTN or COREOTN to/from ILKN

**Figure 189 ODUk Switch Queue Size MAPOTN or COREOTN to/from ILKN**

Egress Direction (ILKN to MAPOTN or COTN)

F	Trpl+Tscl+Tstl (time to detect threshold crossing and send XOFF)	PDV*	2 blocks per ODU0	Trpl+Tscl+Tstl (time to detect threshold crossing and send XON)	E
LMAX-Xoff					LMAX-Xon

\*For external buffering where DIGI buffering is not sufficient, use maximum available DIGI buffering for partial PDV compensation

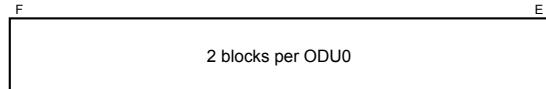
Ingress Direction (MAPTON or COTN to ILKN)

F	E
HOL Blocking (when mixing ODUk / PKT traffic)	2 blocks per ODU0

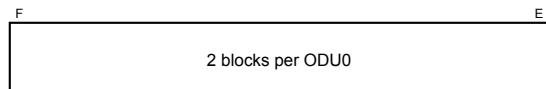
### 5.21.1.2 MAPOTN to/from COREOTN

Figure 190 ODUk Switch Queue Size MAPTON to/from COREOTN

MAPOTN to COREOTN



COREOTN to MAPOTN



## 5.21.2 CPB Queue Size Calculation

Use the ODUk Switch Queue Size Calculation Spreadsheet [13]. Values to enter are in yellow and the results of interest are in green. Note that overlapping internal buffering requirements are in blue.

The spreadsheet calculates queue size estimates for the following datapath segments:

- ILKN  $\iff$  CPB
- ENET  $\iff$  CPB
- MAPOTN  $\iff$  CPB
- CBRC  $\iff$  CPB

To calculate buffer requirement for end-to-end datapaths, calculate the segment buffering in each direction and use the worst case internal buffering for segments that overlap. Note that the Internal estimates are in blue.

### 5.21.2.1 CPB to/from ILKN

Figure 191 CPB Queue Size CPB to/from ILKN

Egress Direction (ILKN to CPB)

F	LMAX-Xoff	Trpl+Tscl+Tstl (time to detect threshold crossing and send XOFF)	DIGI to DIGI: PKT MODE = 1 block + fabric ingress allocation DIGI to DIGI: SEG MODE = 2 blocks DIGI to Other: 2 blocks + rate shaping at mate device	Trpl+Tscl+Tstl (time to detect threshold crossing and send XON)	E LMAX-Xon	
+Extra MTU		+Calendar dribble				

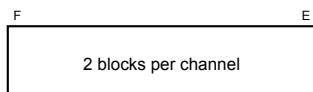
Ingress Direction (CPB to ILKN)

F	HOL Blocking (when mixing ODUk / PKT traffic) ODUk packet for every ODUk channel @ 508B per packet sent at the ILKN rate	Pkt Mode Rev A: # of cal entries +1 (to max of 20, and min of 3) Pkt Mode Rev B: number of calendar entries + 1 to Max of 20 Segment Mode: 2 blocks per channel	E
---	-----------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------	---

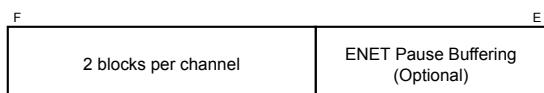
### 5.21.2.2 ENET to/from CPB

Figure 192 CPB Queue Size ENET to/from CPB

Egress Direction (CPB to ENET)



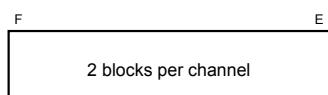
Ingress Direction (ENET to CPB)



### 5.21.2.3 CPB to/from MAPOTN

Figure 193 CPB Queue Size CPB to/from MAPOTN

Egress Direction (CPB to MAPOTN)



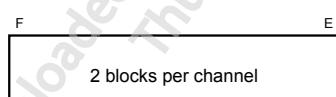
Ingress Direction (MAPTON to CPB)



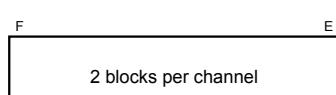
### 5.21.2.4 CBRC to/from CPB

Figure 194 CPB Queue Size CBRC to/from CPB

Egress Direction (CPB to CBRC)

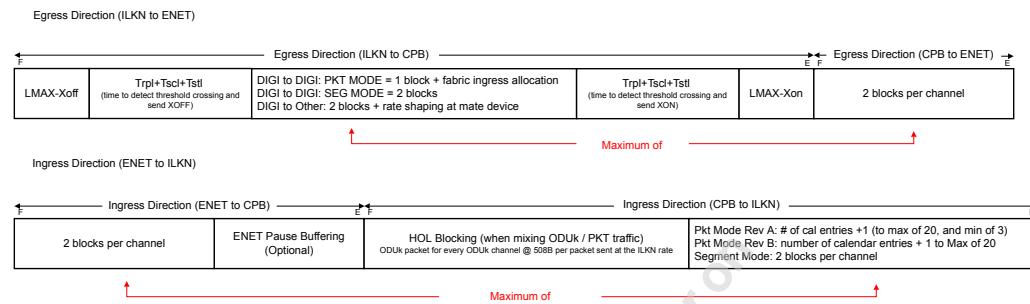


Ingress Direction (CBRC to CPB)



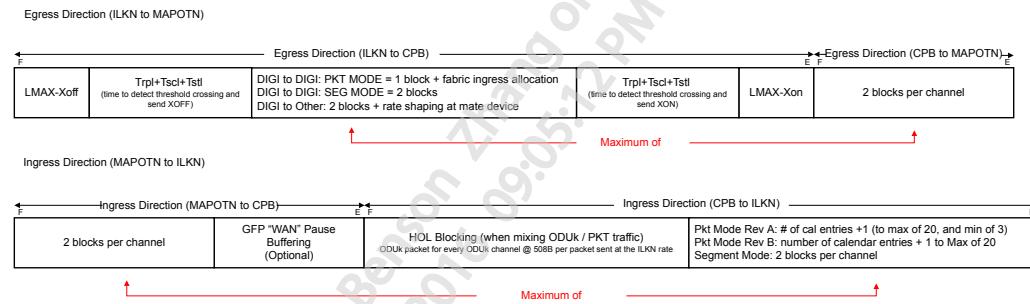
### 5.21.2.5 ENET to/from CPB to/from IKLN

Figure 195 CPB Queue Size ENET to/from CPB to/from IKLN



### 5.21.2.6 MAPOTN to/fom CPB to/from IKLN

Figure 196 CPB Queue Size MAPOTN to/from CPB to/from IKLN



## 6

# Non-Compliances

This section discusses standards compliance issues in the hardware and possible approaches to achieve compliance in a system.

## 6.1 OTN Non-Compliances

### 6.1.1 dCSF Is Not Declared After a Persistency Check

#### Description

G.798 clause 6.2.10 requires dCSF to be declared if the CSF bit in the OPUk PSI overhead (bit 1 of the PSI[2] byte) is "1" for X consecutive 256-frame multiframe where "X=3". The dCSF defect reported by the device is not debounced as per the G.798 requirement.

The dCSF defect reported by the device is not debounced as per the G.798 requirement. The dCSF\_I / dCSF\_V interrupt indication / interrupt status bits are not debounced.

The dCSF defect reported to the DI\_RCP ring control port is not debounced.

#### Workaround

Device firmware will provide a debounced dCSF\_I and dCSF\_V interrupt indication and interrupt status bits for system software.

Designs using the dCSF indication from the DI\_RCP need to perform the persistency check in the FPGA connected to the DI\_RCP ring control port.

### 6.1.2 10G I.7 FEC Bit Errors When Unrelated 10G Channel Defects are Present

#### Description

The device 10G I.7 FEC asserts a FEC\_FAIL status signal to suppress counting of corrected bits as per the PMC interpretation of G.798 clause 12.3.1.5: During AI\_TST-P, dAIS, dLOF and dLOM, no corrected bits shall be counted.

Per channel FEC\_STATUS signals are broadcast to all 10G FEC decoder disable control inputs within the same 4x10G slice. The decoder disable is used to implement bypass mode where the FEC latency is preserved without correcting errors. The decoder disable is a register configurable mode.

For a mix of 10G I.7 FEC and 10G GFEC channels within the same 4x10G FEC slice, the FEC\_FAIL associated with a given 10G channel will drive the decoder disable input of both the I.7 and GFEC 10G decoders that belong to that channel. In this scenario, 10G I.7 channels may be corrupted when the FEC\_FAIL signals intended for active 10G GFEC links are asserted.

Examples of the coupling of errors from one 10G I.7 FEC link to another are:

- For a 4x10G FEC slice configured to support I.7 FEC, when FEC\_FAIL is asserted for a I.7 10G link and a bit error occurs on the link, bit errors may occur on one of the other I.7 10G links processed by the same 4x10G slice.
- Loss of signal on one OTU2 link (or any alarm noted in G.798 clause 12.3.1.5 above) will result in a bit error on an unrelated OTU2 using I.7 FEC where the OTU2 links are processed by the same 4x10G FEC slice.

For a 4x10G FEC slice configured for I.7, a 10G OTUk with I.7 FEC can report invalid bit errors when a different I.7 10G OTUk link processed by the same 4x10G FEC slice reports FEC\_FAIL as a result of a

defect (e.g. on loss of signal) or a different 10G I.7 FEC or GFEC within the same 4x10G FEC slice is bypassed using the decoder disable.

The workaround requires defect contributions to FEC\_FAIL to be disabled and the decoder disable is not longer supported. As a result, the I.7 FEC will not support the same functionality as other FECs supported on the device which do support FEC\_FAIL and decoder disable modes.

### Workaround

The defects that contribute to the FEC\_FAIL status of a 4x10G FEC slice are enabled using registers. For 10G I.7 FEC or mixes of 10G I.7 FEC and GFEC in a 4x10G FEC slice, all FEC\_FAIL outputs for that 40G slice must be disabled by disabling the defect contributors. In addition, the decoder disable mode is not supported and the associated DECODER\_DISABLE register bits must be set to zero.

Defect correlations for the corrected error counts of 10G I.7 FEC and 10G GFEC from the same 4x10G FEC slice must be performed in software. Software must monitor the defects from the OPSM23 block and ignore corrected error counts for seconds where a defect is present.

## 6.1.3 Adding a TCM or PM-NIM to A Signal in a Failure Condition will Result in SF Deasserting for One Frame

### Description

A one frame glitch of aAIS / aTSF will occur when a TCM or PM-NIM is added to a ODUk signal that is already in a server signal fail state due to other alarms. This will result in downstream layers reporting that CI\_SSF and dAIS as not occurring for one frame. The glitch will not occur when adding TCM or PM-NIMs to a high order or mid order ODUk when one or two stages of multiplexing is used.

Downstream layers will initially assert the CI\_SSF and dAIS interrupts when the ODUk signal fails. When a TCM or PMNIM layer is added with the ODUk in the SSF state, the downstream CI\_SSF and dAIS interrupts will fire again if not previously cleared due to a one frame transition out of the aAIS / aTSF state.

### Workaround

Check the interrupt status to confirm the state of CI\_SSF and dAIS interrupts when they fire.

## 6.1.4 Early or Late Assertion of dDEG during dIAE for the OTUk\_TT\_Sk Function

### Description

The device dDEG declaration logic for the OTUk\_TT\_Sk function does not suppress the near-end errored block count (NE\_EBC) correctly during dIAE for the following scenarios:

- Scenario 1: if dDEG = 0 and pIAE = 1, and NE\_EBC > DEGTHR, the monitored degraded defect 1-second errored block count (DEGM) will increment and it is possible for a dDEG to be declared earlier than expected unless dIAE occurs during the errored second where DEGM = DEGTHR.
- Scenario 2: If dDEG = 1 and pIAE = 1, and dDEG was activated at the start of the current interval, and NE\_EBC > DEGTHR, the dDEG will be set to 0, but the DEGM count at the end of the period will be 1 rather than 0. It will take only DEGM – 1 intervals for dDEG to be declared again.
- Scenario 3: If dDEG = 1 and pIAE = 1 and NE\_EBC > DEGTHR, the DEGM count will be reset to 0 and the OTUk will stay in the dDEG state for longer than expected.

The dDEG defect can be declared earlier or later than expected when dIAE is present. Only the OTUk dDEG implemented in the line OTN subsystem is impacted. ODU and TCM layer dDEG are computed correctly during dIAE in the core OTN subsystem.

## Workaround

The workaround requires setting the dDEG interval to one second and performing the dIAE defect correlation in system software.

1. Set the dDEG interval to one (DEGM = 1). Implement the persistence check in software to create dDEG after M intervals (dDEG\_SW).
2. Disable the propagation of SD to COREOTN only when dDEG\_SW is deasserted. This will ensure the correct client signal indication (CSI) value in the OTN packet fabric header for OTN packets sent over the Interlaken interface.
3. Use the hardware dDEG interrupt to determine in software when NEBC > DEGTHR. DEGM is configured to 1. The hardware dDEG interrupt will fire for any interval where NEBC > DEGTHR.
4. Implement a software counter to declare dDEG\_SW when dDEG is asserted for DEGM intervals, or, clear dDEG\_SW when dDEG is not asserted for DEGM intervals.
5. When dDEG\_SW = 1, set the SD output enable to zero and set the DEGTHR to zero for the associated channel. For every one second interval, read the dDEG and pIAE interrupts to maintain an interval DEGM count. When "M" number of good intervals are found, set dDEG\_SW = 0. Go to step 2.

Implement the same workaround in hardware (e.g. FPGA) for dDEG reported to the defect indication ring control port.

### 6.1.5 Insertion and Extraction Points for APS Bytes are Not Compliant to the 12/2012 Version of G.798

#### Description

The pre-published 12/2012 issue of G.798 added specifications for the source and sink location for the ODUk server layer trail APS/PCC bytes (denoted by MFAS bits 6, 7, and 8 values of 1, 1, and 1) to be at the OTUk/ODUk and ODUk/ODUj adaptation layer functions. The device sources and sinks all APS/PCC bytes at the ODUk layer. The bytes are inserted prior to the OTUk/ODUk and ODUk/ODUj adaptation source functions. The bytes are extracted after the OTUk/ODUk and ODUk/ODUj adaptation sink functions.

For the sink functions, consequential actions at the ODUk server layer trail (e.g. OTUk or a high order ODUk) will overwrite the APS/PCC bytes prior to extraction at the ODUk layer. Similarly, an AIS / LCK / OCI maintenance signal forced by software at the ODUk server layer trail will overwrite the APS/PCC prior to extraction.

For the source functions, there are no consequential actions that can overwrite the APS bytes after they are inserted at the ODUk layer. Insertion at the ODUk layer is not an issue although non-compliant with the pre-published 12/2012 G.798 standard. For the source functions, APS bytes are inserted after ODUk-LCK maintenance signal insertion.

This impacts both register access to the APS bytes and overhead port / RI\_RCP access to the APS bytes.

#### Workaround

For the extraction of APS/PCC bytes, the following defects occur due to an incorrectly provisioned ODUk or when system software has connected the wrong ODUk source to the device:

- ODUk\_TT: OTUk-SM dTIM
- ODUkP\_TT: ODUk\_PM dTIM
- ODUk[i]/[j]: dPLM and dMSIM

The received APS/PCC bytes are not expected to be correct during the above defects since the link is either incorrectly provisioned or connected to the wrong source. Disable the aAIS consequential action for applications that do require APS/PCC bytes to be extracted in the presence of dTIM, dPLM or dMSIM defects.

For insertion of LCK/OCI/AIS maintenance signals in the sink, source the maintenance signal from a framer downstream of the APS extraction point.

### **6.1.6 Fault Isolation and pFCS Handling for Transcoded and GFP Mapped FC-1200**

#### **Description**

For demapping of transcoded and GFP-F mapped FC-1200 into ODU2e as per G.709 clause 17.8.2, the pFCS from the received GFP-F frame cannot be monitored without affecting the FC-1200 CBR stream. Furthermore, monitoring of pFCS cannot be disabled in the MAPOTN as this prevents striping of the pFCS towards the CBRC subsystem.

#### **Workaround**

Force the RGFPF to detect a GENERIC defect. An updated software driver will be provided by PMC with the configuration to enable a GENERIC\_FILTER that will mask pFCS error correction and pFCS error propagation towards the CBRC subsystem. Note, this workaround means that pFCS interrupt and PMON capability is lost. However, BIP and FC block error PMON can be used to monitor the FC-1200 stream.

### **6.1.7 Update to Hardware DDEG Functionality**

The device's hardware dDEG counters are updated using an internally generated one second LCLK trigger. This same internally generated one second LCLK trigger is used to update the other OTN PMON counters of the device when automatic PMON update mode is enabled. Therefore, when hardware dDEG counter updates are enabled, manual triggering via the manual trigger API is not available for the remaining OTN PMON counters.

#### **System Implications**

Automatic triggering of the other PMON counters is not affected by enabling the hardware dDEG functionality. Using the hardware PMON update function with a multiple of one second trigger that is synchronous to the device's 155 Mhz reference clock is also supported. Triggering using an asynchronous external hardware PMON update signal is supported if the internal LCLK generator is bypassed.

#### **Solution**

If manual updates of the OTN counters via the SW API are required the HW dDEG must be disabled and FW dDEG can be used. Note that stage 3B is not available using FW dDEG. For stage 3B dDEG can be calculated off chip using the RCP ports. The RCP ports can output the SM/PM/TCM BIP8 errors to be processed externally for the dDEG alarm.

## 6.2 SONET/SDH Non-Compliances

### 6.2.1 Invalid RRMP LOS\_V Bit

#### Description

In some cases, the RRMP LOS\_V register bit is not able to indicate the status of the incoming stream. The behavior implemented in the RRMP is based on the GR-253-CORE standard prior to Issue 4, where valid SONET/SDH framing is required in order to declare LOS, and does not comply with the current versions of the SDH or SONET standards, where no SONET/SDH framing is necessary for LOS.

#### Workaround

The FC-800 RX\_ASD\_V/I (Analog Signal Detect) bits may be used in place of SONET/SDH LOS detection. This interrupt will allow software and hardware to react directly to changes in the LOS/SD status without any framing requirements on deassertion. To enable this workaround, a limited set of FC-800 monitoring can be enabled in parallel with the SONET/SDH PMON:

- Disable the ::cbrc::cbrc\_sdh\_pmg(0)::sdh\_pmg(X)::top::LOS\_PN11\_INSERT bit to stop the RRMP LOS\_V bit from propagating to the CBR\_FSGM fault;
- Enable the ::cbrc::cbrc\_fc\_ilvr::cbrc\_fc\_ilvr::FC800PMG\_CH\_EN bit to enable the FC800\_PMG functionality;
- Optionally enable the ::cbrc::cbrc\_fc\_ilvr::fc800\_pmg(X)::fc\_cbr::ASD\_DLOLB\_EN bit to enable the RX\_ASD\_V bit to propagate to the CBR\_FSGM fault input for automatic stream replacement.

If a non-framed replacement signal were received, the RRMP\_U would continue to report the LOS defect, but RX\_ASD\_V would be deasserted. Note that other conditions may, by default software configuration, force signal replacement on the ingress. Specifically SONET/SDH LOF may be configured to cause replacement, though this too may be disabled.

If the SERDES transition density checker is used as the sole source of LOS detection, the S16 LOS interrupt bits may be used instead of RX\_ASD\_V to determine the Loss-of-Signal status of the incoming stream. For SYS\_S16, the interrupt bit is CHANNEL\_PCBI:CHANNEL\_PCBI\_LANE\_N\_STAT\_INT:LOS\_V and for LINE\_S16, the interrupt bit is CHANNEL\_PCBI:CHANNEL\_PCBI\_LANE\_N\_STAT\_INT:LOS\_V. The FSGM can react directly to this indication for stream replacement if desired.

## 6.3 Ethernet Non-Compliances

### 6.3.1 High BER Status Bits of the 10GE MAC/PCS Blocks

The High BER status bits of the 10GE MAC/PCS blocks clear every 125us in the presence of the High BER alarm and therefore do not report a constant alarm state 10GE MAC terminated modes.

The affected bits are:

- HI\_BER\_V in Register 0x610038/0x710038 EMAC Ingress PCS Interrupt Value (3/4): EMAC\_IN-GRESS\_PCS\_INT\_VAL\_3\_4\_
- HI\_BER\_I in Register 0x610048 /0x710048 EMAC Ingress PCS Interrupt Status (3/4): EMAC\_IN-GRESS\_PCS\_INT\_STAT\_3\_4\_

10GE MAC terminated mapping to ODU2 via GSUP43 C6.2 uses the High BER alarm to propagate consequential actions therefor those propagated alarms will also be inconsistent. The datapath affected is 10GE<->ODU2(Gsup43 C6.2)<->ODUk\_SW.

10GE MAC terminated demapping from ODU2 via GSUP43 C7.1 or C7.2 should not enable interrupts from the High BER status of the 10GE MAC as they will interrupt continuously while the High BER alarm is detected. The affected datapath is OTU2 <>ODUK\_SW<>MAPOTN<>ETRANS <>10GE MAC.

### Workaround

For 10GE MAC terminated mapping to ODU2 via GSUP43 C6.2 DIGI, SW/FW will be updated in 3.05 to use the GSUP43\_C73 Block's High BER alarm indications in MAC terminated modes for propagating consequential actions.

The EPMM(ENET\_LINE and SYS) bits are:

- HI\_BER\_V in Register 0x648018/0x748018 + 0x0400\*A(A=0:11) RX Status: RX\_STAT
- HI\_BER\_IC in Register 0x648014/0x748014 + 0x0400\*A(A=0:11) RX INTERRUPT INDICATION: RX\_INT\_INDICATION

10GE MAC terminated demapping from ODU2 via GSUP43 C7.1 or C7.2 should poll the High BER interrupt (I) bits to determine the status of the High BER alarm rather than using the status (V) bits.

## 7 Operations

### 7.1 SFI-5.1 SERDES Mapping

To facilitate layout, the mapping of the SFI-5.1 SERDES pins to the dual SERDES instances is not straightforward. The following table provides the mapping of each SFI-5.1 SERDES pin (p/n pair) to the SERDES instance (denoted in register description addressing as A=0:8) and to the side of the instance (denoted in register description fields with \_A/\_B post pend or register addresses with N=0:1), as well as to the base address for configuration via PCBI. The SERDES side (A/B) indicates which lane of the SERDES pair is mapped to the device pin.

Table 113 SERDES Mapping

Device Pin Name (n=1:2)	SERDES Pair Instance (A=0:8)	SERDES Lane (A/B)/ (N=0:1)	SFI1 PCBI Base Address	SFI2 PCBI Base Address
SFI <sub>n</sub> _RXDATA[0]	1	A/0	0x00070C00	0x00074C00
SFI <sub>n</sub> _RXDATA[1]	3	A/0	0x00071400	0x00075400
SFI <sub>n</sub> _RXDATA[2]	5	B/1	0x00071C00	0x00075C00
SFI <sub>n</sub> _RXDATA[3]	8	B/1	0x00072800	0x00076800
SFI <sub>n</sub> _RXDATA[4]	2	B/1	0x00071000	0x00075000
SFI <sub>n</sub> _RXDATA[5]	4	B/1	0x00071800	0x00075800
SFI <sub>n</sub> _RXDATA[6]	6	A/0	0x00072000	0x00076000
SFI <sub>n</sub> _RXDATA[7]	7	B/1	0x00072400	0x00076400
SFI <sub>n</sub> _RXDATA[8]	2	A/0	0x00071000	0x00075000
SFI <sub>n</sub> _RXDATA[9]	3	B/1	0x00071400	0x00075400
SFI <sub>n</sub> _RXDATA[10]	5	A/0	0x00071C00	0x00075C00
SFI <sub>n</sub> _RXDATA[11]	7	A/0	0x00072400	0x00076400
SFI <sub>n</sub> _RXDATA[12]	1	B/1	0x00070C00	0x00074C00
SFI <sub>n</sub> _RXDATA[13]	4	A/0	0x00071800	0x00075800
SFI <sub>n</sub> _RXDATA[14]	6	B/1	0x00072000	0x00076000
SFI <sub>n</sub> _RXDATA[15]	8	A/0	0x00072800	0x00076800
SFI <sub>n</sub> _RXDSC	0	A/0	0x00070800	0x00074800
SFI <sub>n</sub> _TXDATA[0]	5	B/1	0x00079C00	0x0007DC00
SFI <sub>n</sub> _TXDATA[1]	4	A/0	0x00079800	0x0007D800
SFI <sub>n</sub> _TXDATA[2]	2	B/1	0x00079000	0x0007D000
SFI <sub>n</sub> _TXDATA[3]	1	A/0	0x00078C00	0x0007CC00
SFI <sub>n</sub> _TXDATA[4]	7	B/1	0x0007A400	0x0007E400
SFI <sub>n</sub> _TXDATA[5]	5	A/0	0x00079C00	0x0007DC00
SFI <sub>n</sub> _TXDATA[6]	3	B/1	0x00079400	0x0007D400
SFI <sub>n</sub> _TXDATA[7]	0	B/1	0x00078800	0x0007C800
SFI <sub>n</sub> _TXDATA[8]	6	B/1	0x0007A000	0x0007E000
SFI <sub>n</sub> _TXDATA[9]	6	A/0	0x0007A000	0x0007E000
SFI <sub>n</sub> _TXDATA[10]	2	A/0	0x00079000	0x0007D000
SFI <sub>n</sub> _TXDATA[11]	1	B/1	0x00078C00	0x0007CC00
SFI <sub>n</sub> _TXDATA[12]	7	A/0	0x0007A400	0x0007E400

Device Pin Name (n=1:2)	SERDES Pair Instance (A=0:8)	SERDES Lane (A/B) / (N=0:1)	SFI1 PCBI Base Address	SFI2 PCBI Base Address
SFI <sub>n</sub> _TXDATA[13]	4	B/1	0x00079800	0x0007D800
SFI <sub>n</sub> _TXDATA[14]	3	A/0	0x00079400	0x0007D400
SFI <sub>n</sub> _TXDATA[15]	0	A/0	0x00078800	0x0007C800
SFI <sub>n</sub> _TDXSC	8	B/1	0x0007A800	0x0007E800
SFI <sub>n</sub> _TDXCK	8	A/0	0x0007A800	0x0007E800

## 7.2 OH SERDES Mapping

The OH, DI\_RCP, RI\_RCP and ENET\_MGMT ports share a set of three dual SERDES. The following table provides the mapping of these SERDES and the PCBI register base addresses.

Table 114 OH SERDES Mapping

Pin Name	Overhead SERDES Instance	SERDES Side (A/B)	Base Address
OH_RXI[2]	1	A	0x00090400
OH_RXI[1]	0	B	0x00090000
OH_RXI[0]	0	A	0x00090000
DI_RCP_RXI	2	B	0x00090800
RI_RCP_RXI	2	A	0x00090800
MGMT_RXI	1	B	0x00090400
OH_TXO[2]	1	A	0x00091400
OH_TXO[1]	0	B	0x00091000
OH_TXO[0]	0	A	0x00091000
DI_RCP_TXO	2	B	0x00091800
RI_RCP_TXO	2	A	0x00091800
MGMT_TXO	1	B	0x00091400

## 7.3 LR SERDES Mapping

The Long Range SERDES are implemented as a dual RX/TX MABC. As a result it is necessary to map the two instances to device pins. The following table provides the mapping of the Long Range SERDES pin (p/n pair) to the SERDES instance (denoted in register description addressing as A=0:19) and side of the instance (denoted in register description fields with \_A/\_B post pend or register addresses with N=0:1), as well as the base address for configuration via PCBI. The SERDES side (A/B) indicates which lane of the SERDES pair is mapped to the device pin.

Table 115 LR SERDES Mapping

Pin Name (N=0:19)	LR SERDES Instance (A=0:19)	SERDES Side (A/B) / (N=0:1)	Base Address
LR_RXI [2*N]	N	B/1	0x00040000 +0x1000*N
LR_TXO [2*N]	N	B/1	0x00040000 +0x1000*N
LR_RXI [2*N+1]	N	A/0	0x00040000 +0x1000*N
LR_TXO [2*N+1]	N	A/0	0x00040000 +0x1000*N

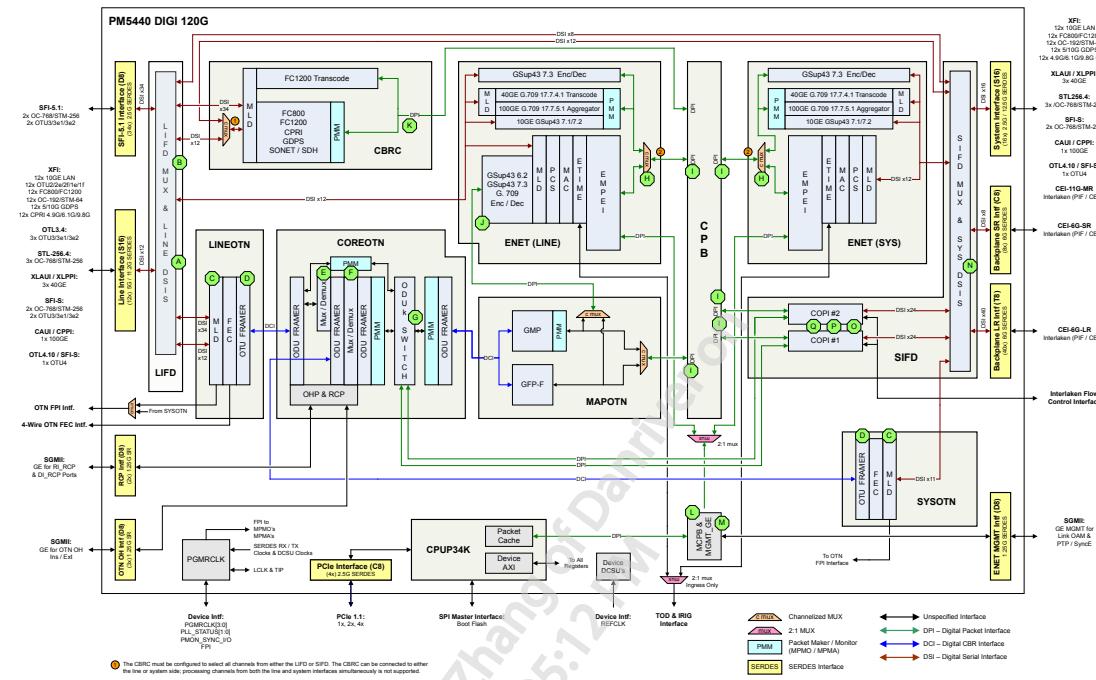
The SERDES side A/0 mapping to odd LR\_RXI/LR\_TXO pins and B/0 mapping to even LR\_RXI/LR\_TXO pins is intentional.

## 7.4 Channel Remapping

The device offers considerable flexibility in configuring the datapath, requiring several conversion points between the external SERDES lane numbering and the channel ID used across the various internal buses (DSI, DCI and DPI). The following diagram identifies the major channel remapping functions within the device.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:12 PM

Figure 197 Channel Remapping



For each channel remapping point identified above, a brief description is included, along with the block responsible for the configuration (marked in square brackets).

- A: Mapping from line-side 10G SERDES link ID to internal DS1 lane ID. [LIFD.LINE\_DSIS]

- B: Ability to swap SFI-5.1 interfaces 1 & 2, producing DS1 lane ID. [LIFD.LIFD\_MUX]
- C: Mapping from DS1 lane ID to internal lane ID. Hardcoded. [LINEOTN.OTN\_MLD]
- D: Mapping from internal lane ID to DCI channel ID. Hardcoded. [LINEOTN.OTU\_FRM]
- E: Mapping from High Order DCI channel to Mid/Low order DCI channel is based channel numbers managed by software as a pool of IDs. Please see the Programmer's Manual ([4]). In cases of non-terminating payload (passthrough), the output DCI channel is equal to the input DCI channel. [COREOTN.MUX]
- F: Mapping from High Order DCI channel to Mid/Low order DCI channel is based channel numbers managed by software as a pool of IDs. Please see the Programmer's Manual ([4]). In cases of non-terminating payload (passthrough), the output DCI channel is equal to the input DCI channel. [COREOTN.MUX]
- G: Mapping from ODUkSW queue ID to egress DPI channel ID. AppLib abstracts the allocation of the DPI channels through channel handles. [COREOTN.ODUkSW.DCS]
- H: Mapping from DS1 lane to DPI channel. This mapping is hardcoded. [ENET.EMPEI, ENET.EPMM]
- I: Mapping from CPB queue ID to egress DPI channel ID. Device AppLib abstracts the allocation of the DPI channels through channel handles. [DCPB.DCS]
- J: Mapping from DPI channel to internal lane ID for ETRANS. [ENET\_LINE.ETRANS]
- K: Mapping from DS1 lane to DPI channel. [CBRC.FC\_ILVR:CBR\_INT]
- L: Mapping from MCPB queue ID to egress DPI channel ID. This mapping is hardcoded. [MCPB.DCS]
- M: Mapping from DPI channel to VLAN header. [MGMT\_GE]
- N: Mapping of system-side client 10G SERDES lane to DS1 channel ID. [SIFD.SYS\_DSIS]
- O: Arbitrary re-ordering of Interlaken lanes. [SIFD.COPI]
- P: Mapping of Interlaken Burst Control Word channel to DPI channel. [SIFD.COPI]
- Q: Mapping of inband 2-byte custom header to DPI channel (optional mode for CBRI). [SIFD.COPI]

Note: In all instances of the MPMA, the input DCI channel ID becomes the same channel ID on DPI. Likewise, for all instances of the MPMO, the input DPI channel ID becomes the same channel ID on DCI.

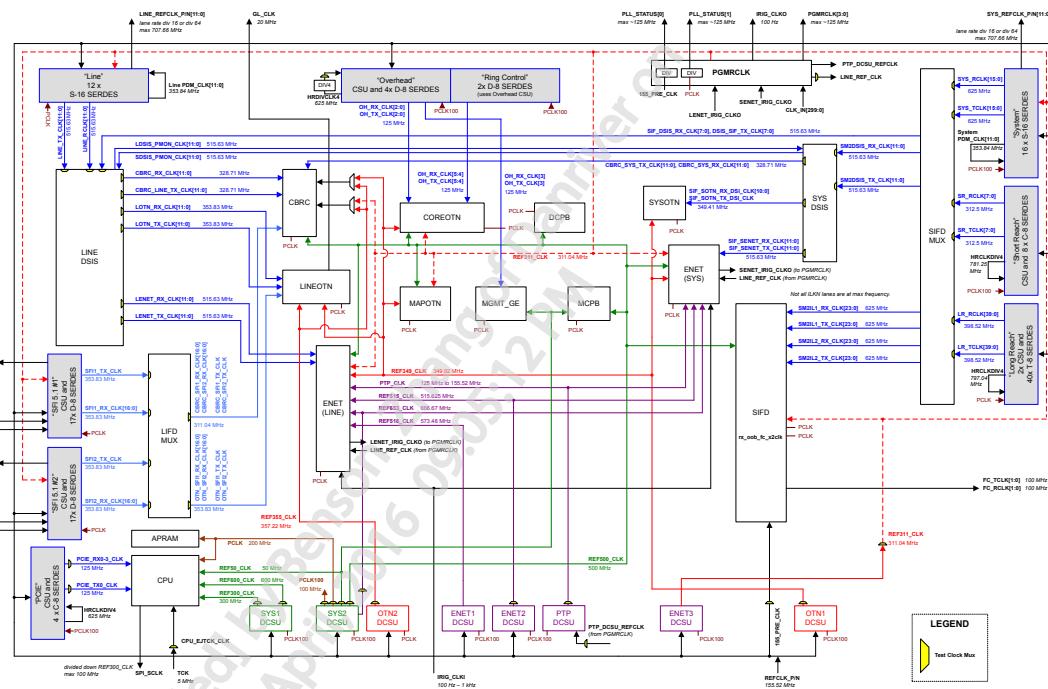
## 8 Functional Timing

### 8.1 Device Clock Diagram

See [Figure 198](#).

For a list of PGMRCLK sources, see [Table 86](#).

**Figure 198** PM5440 Top Level Device Clock Diagram



### 8.2 Interlaken Flow Control Interface

See the "*Interlaken Protocol Definition v1.2*"<sup>[25]</sup> and the "*Interlaken Interoperability Recommendations*"<sup>[26]</sup> for details about the Interlaken flow control interface.

The pin description for this interface can be seen in [Section 4.2.7](#).

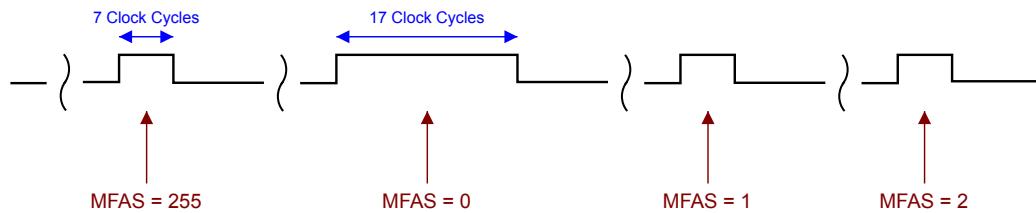
### 8.3 OTN Frame Pulse Interface

The OTU\_RX\_MFPO[11:0] and OTU\_TX\_MFPO[11:0] device outputs mark the multi-frame and frame boundaries observed by the Rx and Tx framers in the LINEOTN subsystem. These outputs are asynchronous and a pulse width of 17 clock cycles denotes a multi-frame boundary where MFAS = 0. A pulse width of 7 clock cycles denotes all other frame boundaries (that is, MFAS = 1 - 255). Each OTU\_RX\_MFPO[x] corresponds to a LINEOTN lane.

Note that when the OTN framer for a given link is in an OOF state, the behavior of OTU\_RX\_MFPO and OTU\_TX\_MFPO is undefined.

The functional timing for this interface is shown in [Figure 199](#) below.

Figure 199 OTN Frame Pulse Interface Functional Timing



The clock cycles in Figure 199 above correspond to the core clock frequencies for each OTN rate listed in Table 116 below.

Table 116 Core Clock Frequencies for OTN Clients

OTN Rate	Core Clock Frequency (MHz)
OTU2	334.66
OTU1e	345.28
OTU2e	346.74
OTU1f	352.19
OTU2f	353.68
OTU3	336.08
OTU3e1	348.21
OTU3e2	348.31
OTU4	349.41

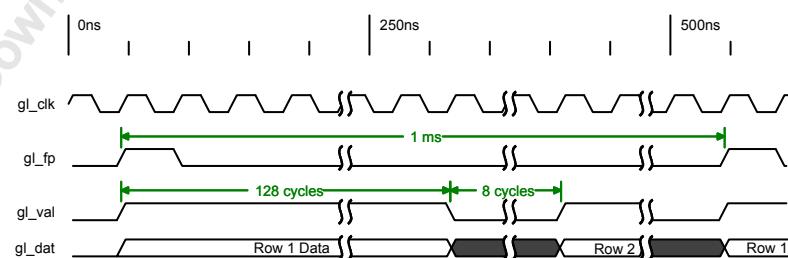
The pin description for this interface can be seen in Section 4.2.10.

## 8.4 4-Wire OTN FEC Statistics Interface

Figure 200 FEC Statistics Logic Table (FSL-T)

FEC Statistics Interface																
Row	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	LLID=0	PHY_LN	ALM	E0 (Corrected Zeroes)		E1 (Corrected Ones)										CRC-8
2	LLID=1	PHY_LN	ALM	E0 (Corrected Zeroes)		E1 (Corrected Ones)										CRC-8
...	LLID=N-2	PHY_LN	ALM	E0 (Corrected Zeroes)		E1 (Corrected Ones)										CRC-8
N	LLID=N-1	PHY_LN	ALM	E0 (Corrected Zeroes)		E1 (Corrected Ones)										CRC-8

Figure 201 FSL-T Timing



**Table 117** FSL-T Description

Legend	Description
LL ID	OTL3.4/OTL4.10 Logical Lane ID or SFI-5.1/SFIS Bit Position. Values are [19:0] for OTU4. For 3xOTU3, LL ID is [11:0], calculated from 4*(OTU3_Index) + ([3:0]). For 12xOTU2, LL ID is just OTU2_Index. LL ID is [47:0] for OTU3 or OTU4 when an OTUk frame is divided into a maximum of 16 bits rather than 16 bytes when the SFI instead of OTL is used
PHY_LN	Index of the physical (10Gbps SERDES) lane that the virtual lane is transmitted over. Values are [11:0] for OTL and [1:0] for SFI.
ALM	Alarm. The ALM bit indicates defects (dAIS, dLOF) in the MLD layer where dLOF can be configured to report dLOF, dOOF, or dLOF ORed with dOOF. Note that dLOM is not included in the defect contributors. All other bits are reserved.
E0	32-bit count of corrected zeroes.
E1	32-bit count of corrected ones.
Uncorr	32-bit count of uncorrectable block errors. Where uncorrectable blocks cannot be specifically associated with one LLID, the uncorrectable count for the entire period is used in each component LLID.
CRC-8	8-bit CRC over the 128-bit row.

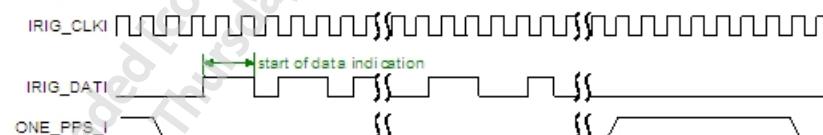
The pin description for this interface can be seen in [Section 4.2.11](#).

## 8.5 TOD and IRIG Interface

The pin description for this interface can be seen in [Section 4.2.14](#).

### 8.5.1 TOD Interface

Each of the ingress and egress TOD interfaces present a 1PPS signal, a serial TOD data stream, and a clock that enables sampling of the TOD data stream. This protocol of this interface is shown in [Figure 202](#) and [Figure 203](#).

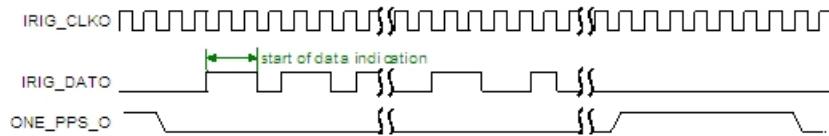
**Figure 202** TOD Input Interface Functional Timing

The ONE\_PPS\_I signal is expected to toggle at a frequency of 1 Hz and its 0 to 1 transition is expected to align with the moment when the integer seconds field of the timing master's TOD changes. When ONE\_PPS\_I transitions to logic 1, a TSTMPR block dedicated to the timing interface functions will capture the local free-running counter value and TOD value from the local TSTMP\_CNTR block.

The rising edge of ONE\_PPS\_I indicates the transition from one integer second to the next. The TOD that corresponds to that rising edge is given in the following IRIG\_DATI data stream. Note that after the first ONE\_PPS\_I falling edge, IRIG\_DATI is logic 0. Then, IRIG\_DATI is set high for two IRIG\_CLKI cycles. After these two clock cycles, the next integer seconds field of the time stamp value is given, starting with the most-significant bit. For PTP, the time stamp value gives the integer seconds field and is 48 bits long. The data on IRIG\_DATI gives the time that was true at the previous ONE\_PPS\_I

event. The most significant bit of this time stamp value is given first. Extra bits after the 48 valid IRIG\_DATI bits give no information. IRIG\_CLKI is normally 100 Hz, but higher clock rates can easily be accommodated.

Figure 203 TOD Output Interface Functional Timing



The IRIG\_DATO output gives the TOD that was true at the previous rising edge of ONE\_PPS\_O. ONE\_PPS\_O will toggle at a nominal frequency of 1 Hz, based on the local clock and transitions from 0 to 1 at the moment when the integer seconds field of the local TOD changes. ONE\_PPS\_O is an asynchronous input signal and its capture resolution is nominally 8 ns (one clock cycle of the internal 125 MHz clock).

The IRIG\_DATO format is identical to that of IRIG\_DATI of the IRIGI component. After ONE\_PPS\_O transitions from 1 to 0, the IRIG\_DATO output will be logic 0. Then, IRIG\_DATO will be logic 1 for two consecutive IRIG\_CLKO intervals. After this, two-period pulse on IRIG\_DATO, the next 48 bits give the integer seconds field of the local PTP time stamp counter that was true at the previous ONE\_PPS\_O rising edge. The most-significant bit of the time stamp value is output first.

The output clock, IRIG\_CLKO, is a ~100 Hz clock derived from the local PTP\_CLK. IRIG\_CLKO is generated from the local PTP\_CLK.

## 8.5.2 IRIG Interface

Please see the "IRIG Standard 200-04" [49].

## 8.6 SPI Master Interface

Figure 204 SPI Master Connected to Multiple Slaves

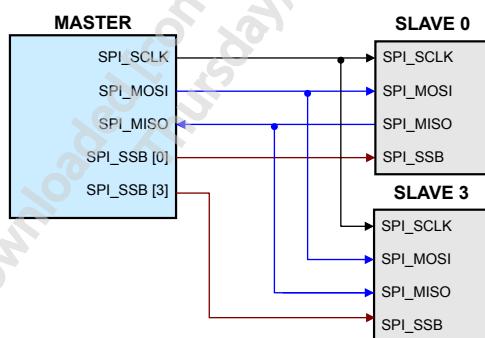
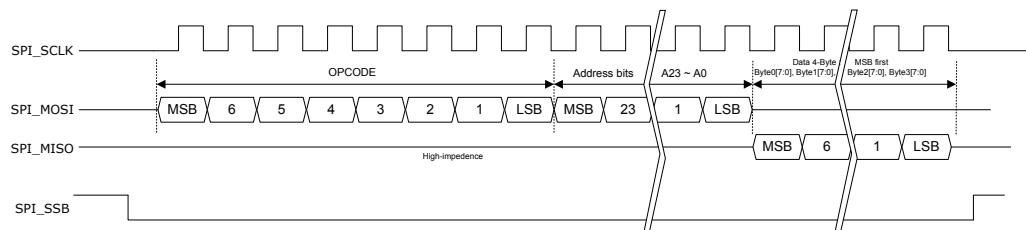


Figure 205 SPI Interface 24-bit Address Functional Timing



In a typical application as shown in [Figure 204](#), the SPI master is connected to multiple slave devices using the SPI\_SSB signal. This signal acts as a "chip select" in order to select the desired slave device. The functional timing for the SPI master interacting with a single slave device, is shown in [Figure 205](#) for a slave device using 24-bit addresses.

The pin description for this interface can be seen in [Section 4.2.15](#).

## 8.7 JTAG Interface

Please see the "*IEEE Standard for Test Access Port and Boundary-Scan Architecture*"[\[20\]](#) and the "*IEEE Standard for Boundary-Scan Testing of Advanced Digital Networks*"[\[21\]](#).

The pin description for this interface can be seen in [Section 4.2.16](#).

## 8.8 Power and Ground Interface

The pin description for this interface can be seen in [Section 4.2.18](#).

Details on power supply sequencing can be seen in [Section 9.2](#).

### 8.8.1 AVS Startup Sequence

Use of the device requires Adaptive Voltage Scaling (AVS). The device will assert the AVS\_ENB output to indicate the correct voltage to apply (0.875V or 0.925V). The AVS\_RSTB input controls the device logic that generates the AVS\_ENB output.

Functional timing is as follows:

1. Power up device at the AVS\_HIGH (0.925) or AVS\_LOW (0.875V) nominal voltage, with the device AVS\_RSTB and RSTB inputs asserted low. Power up the 1.8V supply. The 0.925/0.875 and 1.8V supplies can be powered up in any order. Use AVS\_ENB to select the device AVS voltage. The device can remain in this state indefinitely.
2. De-assert AVS\_RSTB. AVS\_ENB will be valid in under 1 uS.
3. The device AVS power supplies will adjust to the AVS voltage as per the AVS\_ENB output value in step 2.
4. The device can remain in this state indefinitely.
5. Wait for the voltage rails to settle to the new voltage.
6. De-assert RSTB.

## 8.9 Ethernet Timing Modes

### 8.9.1 Loop Timing

Loop timing is where each transmit data stream is timed from its partner receive data stream. Each LINE and SYSTEM SERDES provides this support through the use of a receive rate monitoring block (PMM) and a transmit rate control block (ACB).

### 8.9.2 Through Timing

Through Timing is where a transparent transmit data stream is timed with its original clock. The original clock rate is determined through the MPMA/MPMO which adjusts the output transmit clock rate via the ACB in the LINE or SYSTEM SERDES.

### 8.9.3 External Timing

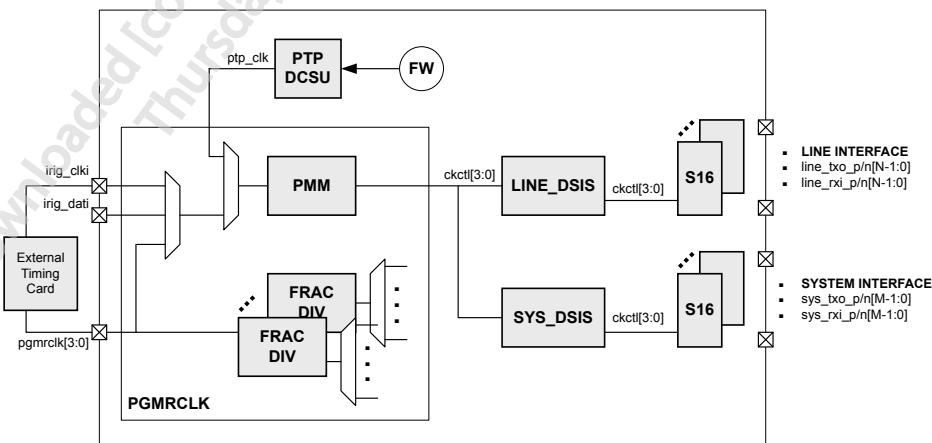
External timing is where the transmit data stream is timed from an external source like a BITS clock or a local crystal oscillator. In this case, the external source drives the REFCLK\_P/N device input which in turn is used to time all the LINE and SYSTEM SERDES.

### 8.9.4 Line Timing

Line Timing is where a transmit data stream is timed with the clock from any active receive data stream in the system or the PTP clock. An extra rate monitoring block (PMM) is used to measure the selected receive data stream or PTP clock. The output rate modulation control signal (ckctl) is then used by the ACB of any/all of the LINE and SYSTEM SERDES to control the transmit rate.

The most common use model of Line Timing is shown in [Figure 206](#) below. The selected internal receive reference clock is fractionally divided down by FRAC\_DIV, and provided as an output on any of the PGMRCLK[3:0] device pins to an external timing card that can provide holdover support. The timing reference is then fed back to the device on either the IRIG\_CLKI or IRIG\_DATI inputs. The PMM monitors the rate of this incoming timing reference and produces the rate modulation control signals used to adjust the transmit rate of any/all of the LINE and SYSTEM SERDES.

**Figure 206** Ethernet Line Timing Application



Alternatively the timing reference could be the PTP clock. In this case, the PTP DSCU is configured to generate a nominal 125 MHz clock and firmware monitors PTP messages and adjusts the PTP DCSU rate as required. The PMM will monitor the PTP clock and produces the rate modulation control signals used to adjust the transmit rate of any/all of the LINE and SYSTEM SERDES.

Line Timing mode can be used to support Synchronous Ethernet (SyncE). However, by itself, the DIGI 120G cannot guarantee all the clock characteristics necessary for SyncE. In order to ensure compliance to SyncE (G.813 and G.8262), the external timing card would be responsible for meeting standards requirements concerning wander generation, holdover performance, and clock switchover transient performance.

## 8.10 PMON Latch Time

Performance monitor latching is performed either by the LCLK (see [Section 5.8.12.3](#) and [Section 5.16.1.4](#)) or triggered by a configuration register in the individual functional blocks. The PMON information will be available in the results register less than 1ms after the trigger.

The appropriate "busy" or "transfer in progress (TIP)" indication should be confirmed to be clear before accessing the counter values. For LCLK transfers, the consolidated TIP can be used as described in [Section 5.16.1.5](#). The configuration register controls for each of the functional blocks are detailed in the Device Register Description Document [\[2\]](#).

## 9 Power

### 9.1 Normal Operating Voltages

Table 118 Normal Operating Voltages

Supply Voltages	Pins	Operating Range <sup>1</sup>			Tolerance
		Min (V)	Typ (V)	Max (V)	
0.9 V Core SLOW (see Note 2)	VDD	0.902	0.925	0.948	$\pm 2.5\%$
0.9 V Core FAST (see Note 2)	VDD	0.853	0.875	0.897	$\pm 2.5\%$
0.9 V Analog SLOW (see Notes 2 and 5)	AVD_*_0V9, AVD_*_0V9_*, AVD_*_*_0V9_*, AVD_0V9_REF, AVD_DC-SU_*	0.907	0.925	0.944	$\pm 2\%$
0.9 V Analog FAST (see Notes 2 and 5)	AVD_*_0V9, AVD_*_0V9_*, AVD_*_*_0V9_*, AVD_0V9_REF, AVD_DC-SU_*	0.858	0.875	0.893	$\pm 2\%$
1.2 V Analog	AVD_*_1V2, VREF_1V2_LINE, VREF_1V2_SYS	1.176	1.20	1.224	$\pm 2\%$
1.8 V Digital	VDDO_1V8, VDDO_1V8_REF	1.764	1.80	1.836	$\pm 2\%$
1.0 V Analog T8 Overdrive (see Note 3)	AVD_LR_TX*, AVD_LR_0V9_CSU*, AVD_LR_0V9_VCO*	0.98	1.00	1.02	$\pm 2\%$

#### Note:

1. Power supply, A.C. Characteristics and timing are characterized across these operating ranges, unless otherwise stated.
2. Adaptive voltage scaling (AVS) is required for the 0.9 V Core and 0.9 V Analog supply rails. SLOW refers to a device at the slow process corner, indicated by AVS\_ENB =1. FAST refers to a device at the fast process corner, indicated by AVS\_ENB=0.
3. The 1.0 V Analog T8 Overdrive is only needed if the backplane channel has marginal performance since this will boost the transmit swing by ~10%. Otherwise it can be connected to the 0.9V Analog supply.
4. The 0.9 V VDD analog supplies will collectively draw a large current (~6A) if powered-up before the digital VDD 0.9V supply. See [Section 9.2.1](#) for details.
5.  $\leq 5\text{mV}$  ripple on the DCSU AVD pins in the 3 kHz to 1 MHz range based on simulation data from 3 kHz to 20 kHz.

Please refer to the device Hardware Design Guide for power supply filtering, power supply voltage sensing, and regulator design recommendations. Refer to the device BSDL file for an unabbreviated list of power pin names.

## 9.2 Power Supply Sequencing Requirements

It is required that I/O supplies are applied before or at the same time as any input voltage to avoid damage to the device.

### 9.2.1 AVD Current is Large when VDD Is Powered Off

The 0.9V AVD analog supplies will collectively draw a large current (~6A) if powered-up before the 0.9V digital VDD supply. The large draw is due to the analog being enabled when the digital VDD supply is not powered. There are no reliability implications for the large power draw and bring up of the device supplies can be sequenced in any order. However, if the large current is problematic for a given board design, then bring up the digital VDD 0.9V supplies prior to the analog AVD supplies.

Use the following power-up sequence if avoiding the large current draw is desirable.

1. Power up the digital 0.9V VDD supply.
2. Assert RSTB = 0.
3. Wait 100 ns.
4. Power up the analog 0.9V AVD supply.

### 9.2.2 PCIe Receiver Impedance Dependency on Power Supply Bring Sequencing

The input termination of PCIe receivers in the device will temporarily be in a low impedance during power-up if the VDDO\_1V8 supply is enabled prior to the analog AVD 0.9V supply or if the analog AVD supply is brought up before the digital VDD 0.9V supply. To avoid this issue, it is recommended to bring up Digital supply VDD\_0V9 supply first, followed by AVD\_0V9 supply and then VDDO\_1V8 supply. The input termination of PCIe receivers are in a high impedance state after all supplies are brought up and before the device processor has initialized the PCIe end point.

Enumeration of the device by a root port may fail if the low impedance state of the link is detected prior to configuration of the end point by the boot code. Typically, a system will not start enumeration until all devices are powered and out of reset due to the time required to configure and initialize system resources. If this is not the case, the order in which the supplies are powered up will ensure the PCIe receivers are in the high impedance state prior to initialization of the PCIe endpoint by firmware. The required sequence is to power up the digital VDD supply before the analog AVD supply and power up the VDDO\_1V8 supply last.

## 9.3 Power Supply Ramp Time

The ramp times of all external power supplies must be slower than 100  $\mu$ s.

## 9.4 Power Requirements

Table 119 estimates the power requirements for a representative 100G application that uses many of the device's features. The power numbers are estimates based on device tests running typical traffic under realistic operating conditions.

**Table 119 Power Requirements - 100G Hybrid Application**

Conditions	Parameter	Typical	Thermal	Units
OTU4 with GFEC	Total Power	24.22	45.37	W
Demux into 10xODU2	Total IDD (0.9V Digital)	20.14	42.23	A
Split into two groups	Total IDD (0.9V Analog)	5.15	7.28	A
Group 1: 5xODU2	Total IDD (1.2 V)	0.11	0.12	A
Demux into 40xODU0	Total IDD (1.8 V)	0.39	0.46	A
ODUK Switch	Static IDD (0.9V Digital)	7.64	28.28	A
Merge with group 2 at COPI (Interlaken)	Static IDD (0.9V Analog)	0.23	1.39	A
SYS I/F (16 x S16 SERDES)				
Group 2: 5xODU2				
ODUK Switch				
Demap into 10GE				
MAC termination				
DCPB				
Merge with group 1 at COPI (Interlaken)				
SYS I/F (16 x S16 SERDES)				
3 x OTN Overhead ports				
No SYSOTN				
No ENET_TRANSCODER				
No MCPB				
No CBRC				
No ENET Sys				
Backplane LR I/F analog supplies grounded				
Backplane SR I/F analog supplies grounded				
SFI-5.1 Interface analog supplies grounded				

**Note:**

1. See the [Section 9.5](#) and [Section 9.1](#) tables for a definition of the Typical and Thermal operating conditions.
2. For the thermal condition, the 0.9 V Core FAST (0.897 V) and 0.9 V Analog FAST (0.893 V) supply voltages are used.
3. Power numbers are calculated by summing the powers of the various supplies. Power = sum(Vi x IDDi) where Vi is the voltage of supply "i", and IDDi is the current of supply "i".

**Table 120** estimates the incremental power requirements for various features. The first subsection lists the incremental power for all the subsystems of the device. These blocks and their respective utilizations are listed in the conditions section of the table. Assume dynamic power scales according to traffic utilization. 120G is 100% utilization unless otherwise noted.

Unless specifically noted (Note 3), the incremental leakage is fixed and is not proportional to traffic utilization. The second subsection lists the incremental power for the different SERDES types at various rates. The third subsection lists the analog leakage power that can be saved if all lanes of an interface are unused and the supplies are grounded.

To scale the VDD current with temperature, first estimate the dynamic and static currents using the tables above and below. Then scale the static current according to [Figure 207](#). Finally add the scaled static current to the dynamic current to obtain the total.

Due to the highly flexible nature of the device, power dissipation is both application and traffic dependent. The power numbers are estimates based on device tests running typical traffic under realistic operating conditions. Please contact PMC-Sierra's Applications department to obtain additional power estimation utilities.

Table 120 Incremental Power Requirements

Conditions	Parameter	Typical	Thermal	Units
<b>Baseline</b> All Modes start with this baseline power	Thermal	Total	Power (W)	18.01
		Total	VDD Current (A)	18.97
		Total	AVD_0V9 Current (A)	0.00
		Total	AVD_1V2 Current (A)	0.00
		Total	VDDO Current (A)	0.02
		Total	VDD Leakage Current (A)	18.97
		Total	AVD_0V9 Leakage Current (A)	0.00
	Typical	Total	Power (W)	5.62
		Total	VDD Current (A)	6.05
		Total	AVD_0V9 Current (A)	0.00
		Total	AVD_1V2 Current (A)	0.00
		Total	VDDO Current (A)	0.02
		Total	VDD Leakage Current (A)	6.05
		Total	AVD_0V9 Leakage Current (A)	0.00
<b>LINEOTN</b> Per 100G of Line OTN Scaled according to the actual amount of traffic. For example multiply by 12/10 for 120G of Line OTN	Thermal	Block	Power (W)	1.63
		Block	VDD Current (A)	1.72
		Block	VDD Leakage Current (A)	0.19
	Typical	Block	Power (W)	1.28
		Block	VDD Current (A)	1.38
		Block	VDD Leakage Current (A)	0.07
	Thermal	Block	Power (W)	0.64
		Block	VDD Current (A)	0.67
		Block	VDD Leakage Current (A)	0.07
	Typical	Block	Power (W)	0.54
		Block	VDD Current (A)	0.58
		Block	VDD Leakage Current (A)	0.02
<b>LINEOTN_GFEC</b> Per 120G of Traffic. Scaled according to the actual amount of traffic. For example, multiply by 10/12 for 100G of traffic.	Thermal	Block	Power (W)	5.88
		Block	VDD Current (A)	6.20
		Block	VDD Leakage Current (A)	2.82
	Typical	Block	Power (W)	3.64
		Block	VDD Current (A)	3.94
		Block	VDD Leakage Current (A)	0.63
	Thermal	Block	Power (W)	3.09
		Block	VDD Current (A)	3.26
		Block	VDD Leakage Current (A)	1.52
	Typical	Block	Power (W)	1.94
		Block	VDD Current (A)	2.10
		Block	VDD Leakage Current (A)	0.35

Conditions	Parameter	Typical	Thermal	Units
<b>LINEOTN_I7</b> Per 120G of Traffic. Scaled according to the actual amount of traffic. For example, multiply by 10/12 for 100G of traffic.	Thermal	Block	Power (W)	3.08
		Block	VDD Current (A)	3.25
		Block	VDD Leakage Current (A)	2.72
	Typical	Block	Power (W)	1.08
		Block	VDD Current (A)	1.17
		Block	VDD Leakage Current (A)	0.61
	Block	VDD Leakage Current (A)	3.15	
	<b>SYSOTN_GFEC</b> Block power if in use.	Thermal	Block	Power (W)
			Block	VDD Current (A)
			Block	VDD Leakage Current (A)
		Typical	Block	Power (W)
			Block	VDD Current (A)
			Block	VDD Leakage Current (A)
<b>SYSOTN</b> Block power if in use.	Thermal	Block	Power (W)	0.34
		Block	VDD Current (A)	0.36
		Block	VDD Leakage Current (A)	0.10
	Typical	Block	Power (W)	0.25
		Block	VDD Current (A)	0.27
		Block	VDD Leakage Current (A)	0.04
	Thermal	Block	Power (W)	1.85
		Block	VDD Current (A)	1.95
		Block	VDD Leakage Current (A)	0.78
	Typical	Block	Power (W)	1.06
		Block	VDD Current (A)	1.15
		Block	VDD Leakage Current (A)	0.18
<b>COREOTN</b> Per 100G of Line OTN. Scaled according to the actual amount of traffic. For example multiply by 12/10 for 120G of traffic.	Thermal	Block	Power (W)	2.81
		Block	VDD Current (A)	2.97
		Block	VDD Leakage Current (A)	1.36
	Typical	Block	Power (W)	1.64
		Block	VDD Current (A)	1.77
		Block	VDD Leakage Current (A)	0.31
	<b>COREOTN_FMF</b> Per 100G of 2 level OTN muxing. Multiply by 0.125 for 100G of OTN framing only (no multiplexing). Multiply by 0.5 for 100G of 1 level OTN muxing. Scale according to the amount of traffic. For example multiply by 12/10 for 120G of traffic.	Thermal	Block	Power (W)
			Block	VDD Current (A)
			Block	VDD Leakage Current (A)
	Typical	Block	Power (W)	1.64
		Block	VDD Current (A)	1.77
		Block	VDD Leakage Current (A)	0.31
<b>COREOTN_FO</b> Per 100G of FO connected to LINEOTN or SYSOTN plus MAPOTN. Multiply by 0.5 for 100G of traffic from LINEOTN or SYSOTN only. Scale according to the amount of traffic. For example multiply by 12/10 for 120G of traffic.	Thermal	Block	Power (W)	1.04
		Block	VDD Current (A)	1.09
		Block	VDD Leakage Current (A)	0.26
	Typical	Block	Power (W)	0.71
		Block	VDD Current (A)	0.77
		Block	VDD Leakage Current (A)	0.08
	<b>COREOTN_SW</b> Per 120G of traffic. Scale according to the amount of traffic. For example multiply by 10/12 for 100G of traffic.	Thermal	Block	Power (W)
			Block	VDD Current (A)
			Block	VDD Leakage Current (A)
	Typical	Block	Power (W)	1.25
		Block	VDD Current (A)	1.35
		Block	VDD Leakage Current (A)	0.13

Conditions	Parameter	Typical	Thermal	Units
<b>MAP OTN</b> Per 100G of AMP/BMP/GMP and per 100G of GFP. Multiply by 0.5 for 100G of GFP only traffic. Multiply by 0.5 for 100G of AMP/BMP/GMP only traffic. Scale according to the amount of traffic. For example, multiply by 12/10 for 120G of traffic.	Thermal	Block	Power (W)	1.48
		Block	VDD Current (A)	1.56
		Block	VDD Leakage Current (A)	0.44
	Typical	Block	Power (W)	1.01
		Block	VDD Current (A)	1.09
		Block	VDD Leakage Current (A)	0.12
	Thermal	Block	Power (W)	0.82
		Block	VDD Current (A)	0.87
		Block	VDD Leakage Current (A)	0.14
	Typical	Block	Power (W)	0.61
		Block	VDD Current (A)	0.66
		Block	VDD Leakage Current (A)	0.04
<b>ENET_TRANSCODER</b> Per 100G of traffic. Scale according to the amount of traffic. For example multiply by 12/10 for 120G of traffic.	Thermal	Block	Power (W)	2.34
		Block	VDD Current (A)	2.47
		Block	VDD Leakage Current (A)	0.25
	Typical	Block	Power (W)	1.83
		Block	VDD Current (A)	1.98
		Block	VDD Leakage Current (A)	0.06
	Thermal	Block	Power (W)	2.34
		Block	VDD Current (A)	2.47
		Block	VDD Leakage Current (A)	0.25
	Typical	Block	Power (W)	1.83
		Block	VDD Current (A)	1.98
		Block	VDD Leakage Current (A)	0.06
<b>ENET_SYS</b> Per 100G of traffic. Scale according to the amount of traffic. For example multiply by 12/10 for 120G of traffic.	Thermal	Block	Power (W)	2.34
		Block	VDD Current (A)	2.47
		Block	VDD Leakage Current (A)	0.25
	Typical	Block	Power (W)	1.83
		Block	VDD Current (A)	1.98
		Block	VDD Leakage Current (A)	0.06
	Thermal	Block	Power (W)	4.50
		Block	VDD Current (A)	4.74
		Block	VDD Leakage Current (A)	1.81
	Typical	Block	Power (W)	2.70
		Block	VDD Current (A)	2.92
		Block	VDD Leakage Current (A)	0.46
<b>MCPB</b> Block power if in use.	Thermal	Block	Power (W)	1.14
		Block	VDD Current (A)	1.20
		Block	VDD Leakage Current (A)	0.29
	Typical	Block	Power (W)	0.80
		Block	VDD Current (A)	0.87
		Block	VDD Leakage Current (A)	0.07
	Thermal	Block	Power (W)	1.33
		Block	VDD Current (A)	1.41
		Block	VDD Leakage Current (A)	0.13
	Typical	Block	Power (W)	1.04
		Block	VDD Current (A)	1.12
		Block	VDD Leakage Current (A)	0.03

Conditions	Parameter	Typical	Thermal	Units
<b>SIFD</b> Block power if Interlaken is used.	Thermal	Block	Power (W)	0.24
		Block	VDD Current (A)	0.25
		Block	VDD Leakage Current (A)	0.00
	Typical	Block	Power (W)	0.19
		Block	VDD Current (A)	0.21
		Block	VDD Leakage Current (A)	0.00
<b>SIFD_COPI</b> Power for both Interlaken interfaces carrying 100G of traffic. Multiply by 0.5 for only 1 Interlaken interface. Scale according to the amount of user traffic. For example. multiply by 12/10 for 120G of traffic.	Thermal	Block	Power (W)	1.13
		Block	VDD Current (A)	1.19
		Block	VDD Leakage Current (A)	0.10
	Typical	Block	Power (W)	0.89
		Block	VDD Current (A)	0.96
		Block	VDD Leakage Current (A)	0.02
<b>CPU</b> Block power always in use.	Thermal	Block	Power (W)	1.18
		Block	VDD Current (A)	1.24
		Block	VDD Leakage Current (A)	0.60
	Typical	Block	Power (W)	0.69
		Block	VDD Current (A)	0.75
		Block	VDD Leakage Current (A)	0.19
<b>TOP</b> Block power always in use.	Thermal	Block	Power (W)	0.96
		Block	VDD Current (A)	1.01
		Block	VDD Leakage Current (A)	0.00
	Typical	Block	Power (W)	0.77
		Block	VDD Current (A)	0.83
		Block	VDD Leakage Current (A)	0.00
<b>SERDES_SYS_S16</b> All 16 Serdes active @ 10G. Scale by N/16 for N active serdes. Power scales linearly with rate.	Thermal	Block	Power (W)	5.04
		Block	VDD Current (A)	1.73
		Block	AVD_0V9 Current (A)	3.58
		Block	AVD_1V2 Current (A)	0.00
		Block	VDDO Current (A)	0.00
		Block	VDD Leakage Current (A)	0.08
		Block	AVD_0V9 Leakage Current (A)	0.32
	Typical	Block	Power (W)	3.81
		Block	VDD Current (A)	1.38
		Block	AVD_0V9 Current (A)	2.74
		Block	AVD_1V2 Current (A)	0.00
		Block	VDDO Current (A)	0.00
		Block	VDD Leakage Current (A)	0.02
		Block	AVD_0V9 Leakage Current (A)	0.12

Conditions	Parameter	Typical	Thermal	Units
<b>SERDES_SYS_C8</b> All Serdes active @ 6.25G. Scale by N/8 for N active serdes.	Thermal	Block	Power (W)	1.95
		Block	VDD Current (A)	0.22
		Block	AVD_0V9 Current (A)	1.84
		Block	AVD_1V2 Current (A)	0.00
		Block	VDDO Current (A)	0.00
		Block	VDD Leakage Current (A)	0.01
		Block	AVD_0V9 Leakage Current (A)	0.05
	Typical	Block	Power (W)	1.30
		Block	VDD Current (A)	0.17
		Block	AVD_0V9 Current (A)	1.24
		Block	AVD_1V2 Current (A)	0.00
		Block	VDDO Current (A)	0.00
		Block	VDD Leakage Current (A)	0.00
		Block	AVD_0V9 Leakage Current (A)	0.02
<b>SERDES_SYS_TITAN</b> All Serdes active @ 6.25G. Scale by N/40 for N active serdes.	Thermal	Block	Power (W)	10.94
		Block	VDD Current (A)	5.14
		Block	AVD_0V9 Current (A)	6.40
		Block	AVD_1V2 Current (A)	0.00
		Block	VDDO Current (A)	0.00
		Block	VDD Leakage Current (A)	0.09
		Block	AVD_0V9 Leakage Current (A)	0.31
	Typical	Block	Power (W)	8.82
		Block	VDD Current (A)	4.31
		Block	AVD_0V9 Current (A)	5.23
		Block	AVD_1V2 Current (A)	0.00
		Block	VDDO Current (A)	0.00
		Block	VDD Leakage Current (A)	0.03
		Block	AVD_0V9 Leakage Current (A)	0.07

Conditions	Parameter	Typical	Thermal	Units
<b>SERDES_LINE_S16</b> All 12 Active @ 10G. Scale by N/12 for N active serdes. Power scales linearly with rate.	Thermal	Block	Power (W)	3.75
		Block	VDD Current (A)	1.27
		Block	AVD_0V9 Current (A)	2.68
		Block	AVD_1V2 Current (A)	0.00
		Block	VDDO Current (A)	0.00
		Block	VDD Leakage Current (A)	0.06
		Block	AVD_0V9 Leakage Current (A)	0.24
	Typical	Block	Power (W)	2.84
		Block	VDD Current (A)	1.02
		Block	AVD_0V9 Current (A)	2.05
		Block	AVD_1V2 Current (A)	0.00
		Block	VDDO Current (A)	0.00
		Block	VDD Leakage Current (A)	0.01
		Block	AVD_0V9 Leakage Current (A)	0.09
<b>SERDES_SFI51</b> Power if both SFI Interfaces active. Multiply by 0.5 if only one SFI Interfaces active.	Thermal	Block	Power (W)	3.49
		Block	VDD Current (A)	0.67
		Block	AVD_0V9 Current (A)	1.98
		Block	AVD_1V2 Current (A)	0.80
		Block	VDDO Current (A)	0.00
		Block	VDD Leakage Current (A)	0.01
		Block	AVD_0V9 Leakage Current (A)	0.11
	Typical	Block	Power (W)	2.83
		Block	VDD Current (A)	0.55
		Block	AVD_0V9 Current (A)	1.62
		Block	AVD_1V2 Current (A)	0.68
		Block	VDDO Current (A)	0.00
		Block	VDD Leakage Current (A)	0.00
		Block	AVD_0V9 Leakage Current (A)	0.04

Conditions	Parameter	Typical	Thermal	Units
<b>SERDES_OH</b> All OH ports active.	Thermal	Block Power (W)	0.57	
		Block VDD Current (A)	0.11	
		Block AVD_0V9 Current (A)	0.33	
		Block AVD_1V2 Current (A)	0.12	
		Block VDDO Current (A)	0.00	
		Block VDD Leakage Current (A)	0.00	
		Block AVD_0V9 Leakage Current (A)	0.02	
	Typical	Block Power (W)	0.47	
		Block VDD Current (A)	0.09	
		Block AVD_0V9 Current (A)	0.28	
		Block AVD_1V2 Current (A)	0.11	
		Block VDDO Current (A)	0.00	
		Block VDD Leakage Current (A)	0.00	
		Block AVD_0V9 Leakage Current (A)	0.01	
<b>SERDES_PCIE</b> Block power always in use.	Thermal	Block Power (W)	0.66	
		Block VDD Current (A)	0.16	
		Block AVD_0V9 Current (A)	0.53	
		Block AVD_1V2 Current (A)	0.00	
		Block VDDO Current (A)	0.00	
		Block VDD Leakage Current (A)	0.00	
		Block AVD_0V9 Leakage Current (A)	0.03	
	Typical	Block Power (W)	0.40	
		Block VDD Current (A)	0.13	
		Block AVD_0V9 Current (A)	0.30	
		Block AVD_1V2 Current (A)	0.00	
		Block VDDO Current (A)	0.00	
		Block VDD Leakage Current (A)	0.00	
		Block AVD_0V9 Leakage Current (A)	0.01	

Conditions	Parameter	Typical	Thermal	Units
<b>DCSU</b> Block power always in use.	Thermal	Block	Power (W)	0.13
		Block	VDD Current (A)	0.00
		Block	AVD_0V9 Current (A)	0.14
		Block	AVD_1V2 Current (A)	0.00
		Block	VDDO Current (A)	0.00
		Block	VDD Leakage Current (A)	0.00
		Block	AVD_0V9 Leakage Current (A)	0.00
	Typical	Block	Power (W)	0.10
		Block	VDD Current (A)	0.00
		Block	AVD_0V9 Current (A)	0.11
		Block	AVD_1V2 Current (A)	0.00
		Block	VDDO Current (A)	0.00
		Block	VDD Leakage Current (A)	0.00
		Block	AVD_0V9 Leakage Current (A)	0.00
<b>Per LVDS TX</b> Per LINE_REFCLK[11:0] and SYS_REFCLK[11:0] pin, max 12.	Thermal	Block	Power (W)	0.03
		Block	VDD Current (A)	0.00
		Block	AVD_0V9 Current (A)	0.00
		Block	AVD_1V2 Current (A)	0.00
		Block	VDDO Current (A)	0.02
		Block	VDD Leakage Current (A)	0.00
		Block	AVD_0V9 Leakage Current (A)	0.00
	Typical	Block	Power (W)	0.03
		Block	VDD Current (A)	0.00
		Block	AVD_0V9 Current (A)	0.00
		Block	AVD_1V2 Current (A)	0.00
		Block	VDDO Current (A)	0.02
		Block	VDD Leakage Current (A)	0.00
		Block	AVD_0V9 Leakage Current (A)	0.00
<b>Unused SERDES Supply Grounding</b>				
Line Interface	Static IDD (0.9V Analog)	-88	-251	mA
System Interface	Static IDD (0.9V Analog)	-118	-335	mA
Backplane LR Interface	Static IDD (0.9V Analog)	-73	-332	mA
Backplane SR Interface	Static IDD (0.9V Analog)	-24	-53	mA

Conditions	Parameter	Typical	Thermal	Units
SFI-5.1 Interface	Static IDD (0.9V Analog)	-35	-120	mA
	Static IDD (1.2V Analog)	-140	-209	mA
OTN OH / RCP / ENET MGMT Interface	Static IDD (0.9V Analog)	-6	-21	mA
	Static IDD (1.2V Analog)	-23	-35	mA

**Note:**

1. See the [Section 9.5](#) and [Section 9.1](#) tables for a definition of the Typical and Thermal operating conditions.
2. For the thermal condition, the 0.9 V Core (0.945 V) and 0.9 V Analog FAST (0.948 V) supply voltages are used.
3. Power numbers are calculated by summing the powers of the various supplies. Power = sum(V<sub>i</sub> x I<sub>DDi</sub>) where V<sub>i</sub> is the voltage of supply "i", and I<sub>DDi</sub> is the current of supply "i".
4. Incremental leakage is proportional to traffic utilization.
5. All DCPB/MCPB/ODUKSW memory pages are active and in a higher leakage state.
6. For a calculation tool to estimate the total device power and current requirements based upon a selection of the above blocks please contact PMC-Sierra Applications: [apps@pmcs.com](mailto:apps@pmcs.com).

The following table lists the maximum current requirements of the various 0.9V and 1.2V analog supply filters. This table is intended for PCB trace and component sizing only. Please refer to the Hardware Design Guide [\[3\]](#) for a description of the analog filters.

Power numbers calculated using this table will not match [Table 119](#) and [Table 120](#) .

**Table 121 Analog Filter Current Requirements**

Filter Name (from HWDG)	Ball Name	Current	Units
AVD_DCSU_Q1	AVD_DCSU_ENET1, AVD_DCSU_ENET2, AVD_DCSU_PTP	50	mA
AVD_DCSU_Q2	AVD_DCSU_ENET3, AVD_DCSU_OTN1, AVD_DCSU_OTN3	50	mA
AVD_DCSU_Q3	AVD_DCSU_SYS1, AVD_DCSU_SYS2	50	mA
AVD_LINE_0V9_VCO	AVD_LINE_0V9_VCO	450	mA
AVD_LINE_0V9_PI	AVD_LINE_0V9_PI	1050	mA
AVD_LINE_0V9	AVD_LINE_0V9_CSU[11:0], AVD_LINE_0V9	2400	mA
AVD_SYS_0V9_VCO	AVD_SYS_0V9_VCO	600	mA
AVD_SYS_0V9_PI	AVD_SYS_0V9_PI	1400	mA
AVD_SYS_0V9	AVD_SYS_0V9_CSU[11:0], AVD_SYS_0V9	3200	mA
AVD_0V9_REF	AVD_0V9_REF	10	mA
VREF_1V2	VREF_1V2_LINE, VREF_1V2_SYS	10	mA
AVD_SFI_0V9_Q	AVD_DCSU_OTN2, AVD_SFI1_TX_0V9_VCO, AVD_SFI1_RX_0V9_VCO, AVD_SFI1_RX_0V9_CPGEN, AVD_SFI2_TX_0V9_VCO, AVD_SFI2_RX_0V9_VCO, AVD_SFI2_RX_0V9_CPGEN	800	mA
AVD_SFI_0V9	AVD_SFI1_0V9, AVD_SFI2_0V9	2250	mA
AVD_SFI_1V2	AVD_SFI1_1V2, AVD_SFI2_1V2	1200	mA

Filter Name (from HWDG)	Ball Name	Current	Units
AVD_OH_PCIE_0V9_Q	AVD_OH_0V9_VCO, AVD_OH_0V9_CPGEN, AVD_PCIE_0V9_VCO, AVD_PCIE_0V9_CPGEN	400	mA
AVD_OH_PCIE_0V9	AVD_OH_0V9, AVD_PCIE_0V9	900	mA
AVD_OH_1V2	AVD_OH_1V2	200	mA
AVD_LR_0V9_VCO	AVD_LR_0V9_VCO0, AVD_LR_0V9_VCO1	100	mA
AVD_LR_0V9_CP0	AVD_LR_0V9_CPGENO	600	mA
AVD_LR_0V9_CP1	AVD_LR_0V9_CPGEN1	600	mA
AVD_LR_0V9	AVD_LR_0V9_0, AVD_LR_0V9_1	6400	mA
AVD_LR_TX_CSU	AVD_LR_TX0, AVD_LR_0V9_CSU0, AVD_LR_TX1, AVD_LR_0V9_CSU1	1350	mA
AVD_SR_0V9_Q	AVD_SR_0V9_VCO, AVD_SR_0V9_CPGEN	450	mA
AVD_SR_0V9	AVD_SR_0V9_CSU, AVD_SR_0V9	1100	mA

## 9.5 Conditions for Power Requirements

Table 122 Conditions for Power Requirements

	Typical Conditions	Thermal Conditions
Process	Nominal	Nominal + expected process variation <sup>1</sup>
Voltage	Nominal Vdd	Maximum Operating Vdd
Temperature	T <sub>j</sub> =75 °C	T <sub>j</sub> =105 °C

**Note:**

1. The power number for Nominal + expected process variation of process variation is recommended for thermal calculations as it is the highest power dissipation of all parts in almost all applications.

## 10 Thermal Information

Table 123 Device Thermal Information

Thermal Information	
Maximum long-term operating junction temperature ( $T_j$ ) to ensure adequate long-term life	105°C
Maximum junction temperature ( $T_j$ ) for short-term excursions with guaranteed continued functional performance <sup>2</sup>	125°C
Minimum ambient temperature ( $T_A$ )	-40°C
Device Compact Model <sup>3</sup>	
<p>Ambient SA Heat Sink CS Case JC Junction JB Board</p> <p>Device Compact Model</p>	
Junction-to-Case Thermal Resistance, $\Theta_{JC}$	0.02°C/W
Junction-to-Board Thermal Resistance, $\Theta_{JB}$	2.82°C/W
Heat Sink Requirements	
$\Theta_{SA} + \Theta_{CS}$ <sup>4</sup>	The sum of $\Theta_{SA} + \Theta_{CS}$ must be less than or equal to: [(105 - $T_A$ ) / $P_D$ ] - $\Theta_{JC}$ °C/W where: $T_A$ is the ambient temperature at the heat sink location $P_D$ is the operating power dissipated in the package <sup>5</sup>
	$\Theta_{SA}$ and $\Theta_{CS}$ are required for long-term operation

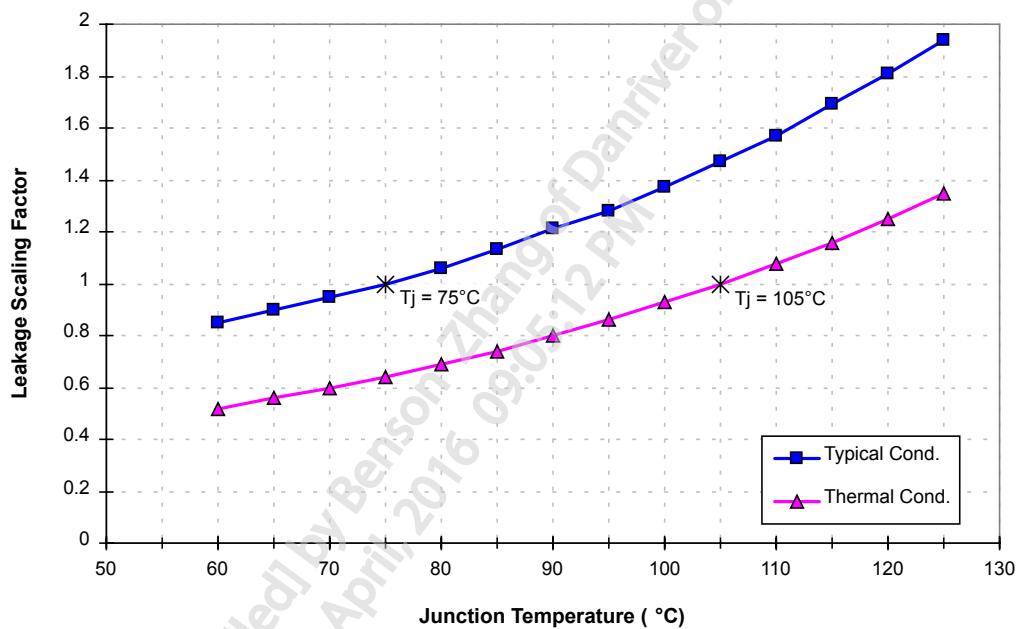
**Note:**

1. The minimum ambient temperature requirement for Outside Office Equipment approximates the minimum ambient temperature requirement for Industrial Equipment.
2. Short-term is used as defined in Telcordia Technologies Generic Requirements GR-63-Core; for more information about the GR-63-CORE standard, please see [18].
3.  $\Theta_{JC}$ , the junction-to-case thermal resistance, is a measured nominal value plus two sigma.  $\Theta_{JB}$ , the junction-to-board thermal resistance, is obtained by simulating conditions described in JEDEC Standard JESD 51-8; for more information about the JESD51-8 standard, please see [19].

4.  $\Theta_{SA}$  is the thermal resistance of the heat sink to ambient.  $\Theta_{CS}$  is the thermal resistance of the heat sink attached material. The maximum  $\Theta_{SA}$  required for the airspeed at the location of the device in the system with all components in place. Power depends upon the operating mode. To obtain power information, see the column under thermal in the tables in [Section 9.4](#).
5. For more information about attaching heat sinks to organic flip chip packages (including maximum pressure and torque), please see [\[9\]](#).

## 10.1 Leakage versus Temperature Curve

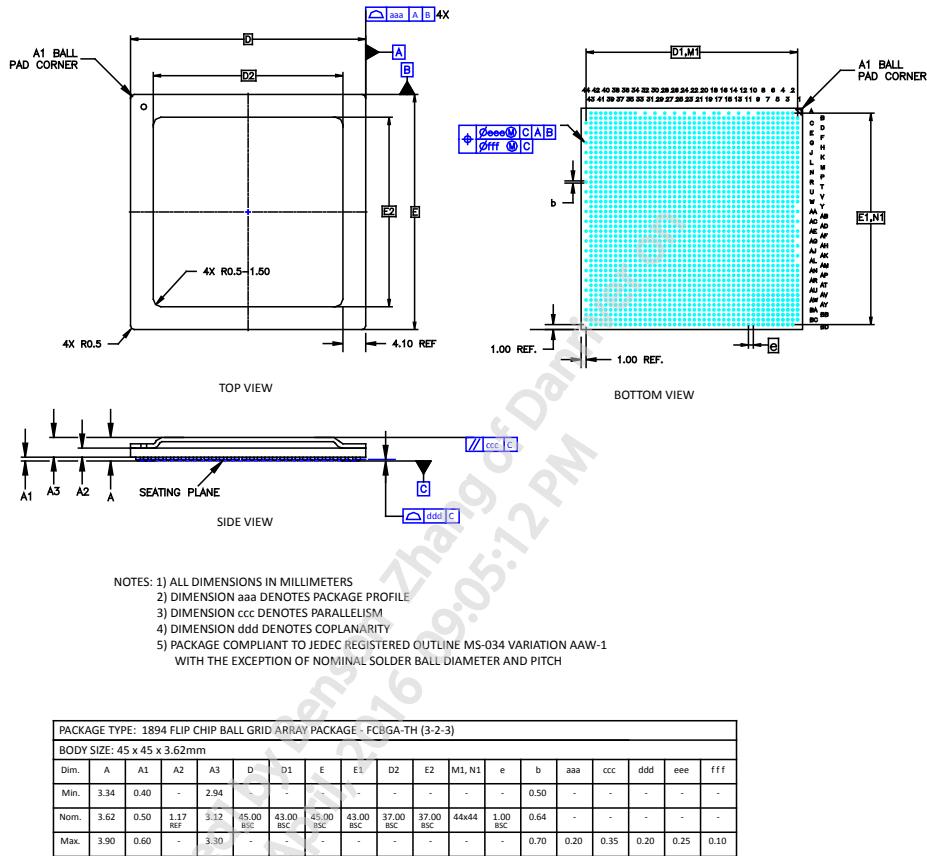
Figure 207 DIGI 120G Typical Leakage Scaling Factor VS. Temperature



Typical and Thermal conditions in [Figure 207](#) are specified in [Section 9.5](#).

## 11 Package Specification

Figure 208 Device Package Specifications



## 12

## Absolute Maximum Ratings

Table 124 Device Absolute Maximum Ratings

Parameter	Min/Max Value
Storage Temperature	0°C to +85°C
Shipping Temperature	-40°C to +125°C
Reflow Temperature	This is a function of the package type. For more information please see <a href="#">[8]</a> .
Absolute Maximum Junction Temperature (T <sub>J</sub> )	+125°C
0.925V Supply Voltage (x_AVDL, x_CAVDL)	-0.3V to 1.2V
1.2V Supply Voltage (AVD_*_1V2)	-0.3V to 1.5V
1.8V Supply Voltage (VDDO_1V8)	-0.3V to 2.2V
Input Current	±20mA

## Note:

1. Absolute Maximum Ratings are not intended for normal device operation. Absolute Maximums are values beyond which permanent damage to the device is likely to occur. Sustained operation of a device at conditions in excess of the recommended operating conditions (given in [Section 9](#) and [Section 10](#)), even though within Absolute Maximum Ratings, may also result in permanent device damage or loss of reliability. Functionality is not guaranteed at conditions exceeding recommended operating conditions. The absolute maximum rating is meant to cover transient voltage conditions only, it is not meant for DC voltage levels.

## 13 DC Characteristics

All specifications in this section are based on  $\pm 2\%$  VDDO\_1V8.

### 13.1 LVCMOS DC Characteristics

Table 125 LVCMOS Interface Drive Strengths

Interface	Pin	Drive Current
Device Interface	PGMRCLK[3:0]	15 mA
	PLL_STATUS[1:0]	15 mA
	INTB	7 mA
	PMON_SYNC_O	7 mA
OTN Frame Pulse Interface	OTU_TX_MFPO[11:0]	15 mA
	OTU_RX_MFPO[11:0]	15 mA
4-Wire OTN FEC Statistics Intf.	GL_CLK	7 mA
	GL_VAL	7 mA
	GL_FP	7 mA
	GL_DAT	7 mA
TOD & IRIG Interface	IRIG_CLKO	7 mA
	IRIG_DATAO	7 mA
	ONE_PPS_O	7 mA
Interlaken Flow Control Intf.	FC_RCLK[1:0]	15 mA
	FC_RSYNC[1:0]	15 mA
	FC_RDATA[1:0]	15 mA
SPI Interface	SPI_SCLK	18 mA
	SPI_MOSI	18 mA
	SPI_SSB[3:0]	7 mA
JTAG Interface	TDO	18 mA

Table 126 DC Characteristics for 1.8V Drive LVCMOS

Symbol	Parameter	Min	Typical	Max	Units	Notes
VIH (DC)	DC Input HIGH Voltage	0.66*VDDO_1V8		VDDO_1V8+0.3	V	1
VIL (DC)	DC Input LOW Voltage	-0.3		0.35*VDDO_1V8	V	1
VOH (DC)	DC Output High Voltage	VDDO_1V8-0.45V			V	
VOL (DC)	DC Output LOW Voltage			0.45	V	
V <sub>HYST</sub> (DC)	Schmitt Trigger Hysteresis Voltage	0.1*VDDO_1V8			V	
RPU	Pull-up Resistance		18		k $\Omega$	
RPD	Pull-down Resistance		21		k $\Omega$	
<b>Note:</b>						
1. Specification for both with and without Schmitt Trigger.						

## 13.2 LVDS DC Characteristics

Table 127 LVDS Pin List

Pin	Type
LINE_REFCLK_P/N[11:0]	Output
SYS_REFCLK_P/N[11:0]	Output
REFCLK_P/N	Input
SFI1_TXCKSRC	Input
SFI2_TXCKSRC	
SFI1_TXREFCLK	
SFI2_TXREFCLK	

Figure 209 LVDS Transmit DC Characteristics

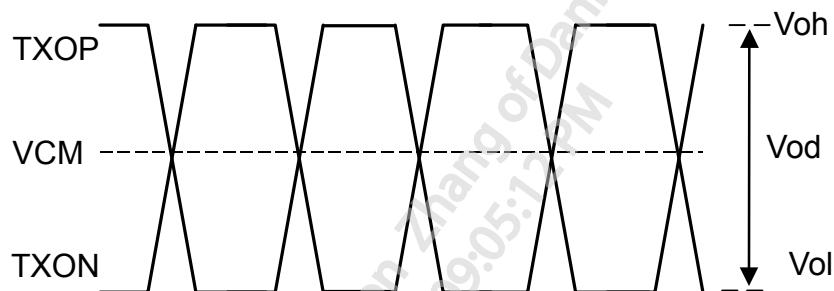


Table 128 TRANSMIT DC Characteristics for 1.8V LVDS

Symbol	Parameter	Min	Typical	Max	Units	Notes
VOD	Output Differential Voltage	230		450	mV	1, 5
Ro	Single Ended Output Impedance	40		140	Ohms	2

Note:

1. 100 Ohms differential load.
2. Measured by setting the common mode of the transmitter at both 1.0V and 1.4V.
3. The above table is derived from simulation results.
4. All outputs are expected to be AC coupled.
5. Ensure the trace length is minimized between the LINE\_REFCLK and SYS\_REFCLK outputs and the mate device. Check the 230 mV output voltage is sufficient to meet the minimum input voltage of the mate device. If it is necessary to meet the 250 mV specification, use a resistive divider to connect the device VREF\_1V2\_LINE and VREF\_1V2\_SYSTEM power balls to 0.75 \* VDDO\_1V8. The current drawn by the VREF power balls is insignificant (<10  $\mu$ A) and use of a signal trace for routing is sufficient.

Table 129 RECEIVE DC Characteristics for 1.8V LVDS

Symbol	Parameter	Min	Typical	Max	Units	Notes
Vid	Input Peak Differential Voltage	120		600	mVpd	2
Rin	Receiver Differential Input Impedance	75		125	Ohm	1
Via, Vib	Input Voltage	0		2.4	V	

**Note:**

1. The above table is derived from simulation results.
2. Long PCB traces may not meet the minimum input voltage specification. Ensure PCB traces are short enough such that the minimum input voltage specification is met. Perform spice simulations to confirm if necessary.

## 14

## High Speed I/O Characteristics

Documentation of the High Speed I/O Characteristics for the following electrical interfaces are available separately from PMC-Sierra's "40nm RAPTR SERDES Cell Set Specifications"[\[7\]](#) here:

- CAUI
- CEI-6G-LR
- CEI-6G-SR
- CEI-11G-MR
- CXP
- OTL3.4
- OTL4.10
- PCIe1
- SFI-S
- SFI-5.1
- SFP+ (limiting)
- SGMII
- STL256.4
- XFI
- XLAUI
- XLPPI / CPPI

### 14.1

### Multi-lane Interface Skew

The tables below provide the maximum or minimum skew between any two SERDES lanes on each specific high speed interface.

Table 130 Maximum Transmit Skew and Receive Skew Tolerance Between SERDES Lanes

Interface	Electrical Interface	Max Rx Skew	Max Tx Skew
Line Interface	OTL3.4	500 ns	29ns
	SFI-S	84 UI	500 ps
	XLAUI	160 ns	29 ns
	CAUI	160 ns	29 ns
	XLPI	145 ns	43 ns
	CPPI	145 ns	43 ns
	OTL4.10	500 ns	29ns
SFI-5.1 Interface	SFI-5.1	20 UI	2.6 UI
System Interface	OTL4.10	500 ns	29ns
	XLAUI	160 ns	29 ns
	CAUI	160 ns	29 ns
	XLPI	145 ns	43 ns
	CPPI	145 ns	43 ns
	CEI-11G-MR (Interlaken)	214 UI	67 UI
Backplane LR Intf.	CEI-6G-LR (Interlaken)	107 UI	67 UI
Backplane SR Intf.	CEI-6G-SR (Interlaken)	107 UI	67 UI
PCIe Interface	PCIe	15 ns	500ps + 2 UI

Table 131 STL256.4 Transmit and Receive Skew Tolerance

Interface	Electrical Interface	Max Rx Skew	Min Tx Skew <sup>1</sup>
Line Interface	STL256.4	180 ns	255UI
System Interface	STL256.4	180 ns	255UI

## Note:

1. Minimum skew is provided as required by ITU G.707 Annex I to ensure sufficient clock content in applications where the stream is re-formed by simple bit multiplex.

## 14.2 SERDES LOS/DLOS Support

This device supports two types of LOS indicators for the Line and System side client interfaces: an LOS indicated externally from a device pin (e.g. by an optical module), and a DLOS (Digital LOS) indicated internally by the device SERDES. A DLOS indication is triggered by the SERDES whenever the signal strength falls below a threshold for a set amount of time. Values for both the threshold and time amount are provided in [Table 132](#). The following sections discuss the LOS/DLOS support for the Line and System side in further detail.

### 14.2.1 Line Side LOS/DLOS Support

On the Line side of the device, LOS indicators can be mapped to the CBRC and LINEOTN subsystems via the LIFD\_MUX. Additionally, LOS and DLOS indicators can be mapped to the CBRC, ENET\_LINE and LINEOTN subsystems via the LIFD LINE\_DSIS. For both cases, the LOS/DLOS indicators can be independently enabled/disabled and optionally inverted. This LOS/DLOS muxing is identical to the data/clock muxing shown in [Figure 13](#).

Using the LIFD\_MUX, the LOS (or more specifically Received Status Signal (RXS)) associated with each SFI interface is directed to either the CBRC orLINEOTN subsystem. Note that DLOS is not provided for either of the two SFI-5.1 interfaces.

Using the LIFD LINE\_DSIS, the combined LINE\_LOS[11:0] and Line Interface SERDES DLOS[11:0] are fed to a cross point mux that allows each of the twelve combined LOS indicators to be directed to any of the CBRC,ENET\_LINE and LINEOTN subsystems lanes' LOS. Since the cross point mux in the LIFD LINE\_DSIS allows any combined LOS to be mapped to any subsystem lane LOS, a combined LOS can be broadcast to any number of internal subsystem lanes' LOS. When LOS is used exclusive of DLOS, the full flexibility for each lane is available. However, for each lane, it is possible to use LINE\_LOS and DLOS simultaneously for the same link with significantly reduced flexibility on the mapping of a LOS input lane to SERDES lane. See [Section 4.5.1.2](#) for details.

#### 14.2.2 System Side LOS/DLOS used for Line Side Support

The first eight System Interface SERDES can be repurposed to connect to the CBRC, ENET\_LINE, or LINEOTN subsystems to support a 3x40G Multi-Service Line Card where one port can optionally support either 40G or 100G. The Line Interface SERDES are used for the pluggable 40G/100G port and the eight System Interface SERDES are used to support up to two additional 40G ports. In this manner, the device supports either a single 100G port or three 40G ports. For this application, the SYS\_LOS[7:0] or internal System Interface SERDES DLOS[7:0] indicators can be mapped to the CBRC, ENET\_LINE, and LINEOTN subsystems via the SIFD\_MUX and the LIFD LINE\_DSIS. Each SYS\_LOS[7:0] and System Interface SERDES DLOS[7:0] can be independently enabled/disabled and optionally inverted. This LOS/DLOS muxing is identical to the data/clock muxing shown in [Figure 13](#).

Using the LIFD LINE\_DSIS, the combined SYS\_LOS[7:0] and System Interface SERDES DLOS[7:0] are fed to a cross point mux that allows each of the eight combined LOS indicators to be directed to any of the CBRC,ENET\_LINE, and LINEOTN subsystems lanes' LOS. Since the cross point mux allows any combined LOS to be mapped to any subsystem lane LOS, a combined LOS can be broadcast to any number of internal subsystem lanes' LOS. When LOS is used exclusive of DLOS, the full flexibility for each lane is available. However, for each lane, it is possible to use LINE\_LOS and DLOS simultaneously for the same link with significantly reduced flexibility on the mapping of a LOS input lane to SERDES lane. See [Section 4.5.1.3](#) for details.

#### 14.2.3 System Side LOS/DLOS Support

On the system side of the device, client interface support is only provided on the first twelve System Interface SERDES and as a result there are only twelve LOS/DLOS indicators. Similar to the Line side support, SYS\_LOS[11:0] or internal System Interface SERDES DLOS[11:0] indicators can be mapped to the CBRC, ENET\_SYS, and SYSOTN subsystems via the SIFD SYS\_DSIS. Each LOS/DLOS can be independently enabled/disabled and optionally inverted. This LOS/DLOS muxing is identical to the data/clock muxing shown in [Figure 119](#).

Using the SIFD SYS\_DSIS, the combined SYS\_LOS[11:0] and System Interface SERDES DLOS[11:0] are fed to a cross point mux that allows each of the twelve combined LOS indicators to be directed to any of the CBRC, ENET\_SYS and SYSOTN subsystems lanes' LOS. Since the cross point mux allows any combined LOS to be mapped to any subsystem lane LOS, a combined LOS can be broadcast to any number of internal subsystem lanes' LOS. When LOS is used exclusive of DLOS, the full flexibility for each lane is available. However, for each lane, it is possible to use SYS\_LOS and DLOS simultaneously for the same link with significantly reduced flexibility on the mapping of a LOS input lane to SERDES lane. See [Section 4.5.2.1](#) for details.

For system side backplane applications, internal SERDES DLOS indications for each SERDES lane used in the given configuration can be independently enabled and optionally inverted prior to reaching the internal subsystem. No cross point mux is provided. Allowable backplane configuration are paths shown in [Section 5.11.5.2.5](#) that do not go through the SYS\_DSIS. For these paths, the internal SERDES DLOS indications have the same mux support as the data lanes. For example, when Interlaken 2 uses the Long Reach Backplane SERDES[23:0], the internal SERDES DLOS[23:0] of the Long Reach Backplane SERDES can be independently enabled and optionally inverted before feeding to the COPI #2 internal lanes[23:0] and no reordering of the 24 lanes is provided.

## 14.3

### SERDES DLOS Characteristics

The Digital Loss of Signal (DLOS) is a function of the SERDES to indicate to the system the presence or absence of a signal being received on the serial receive SERDES input. DLOS characteristics are loosely represented in terms of receive input signal threshold to detect the DLOS and the duration which the signal must be below (or above) this threshold to declare (or clear) a DLOS condition.

[Table 132](#) lists the minimum/maximum threshold possible for each SERDES along with the minimum/maximum hold times.

**Table 132 SERDES DLOS Characteristics**

Symbol	Parameter	Min	Max	Units
<b>Backplane SR Interface (C8 SERDES link-down to link-up)</b>				
Threshold	When the SERDES drops below the Threshold for any time greater than the Hold Time, DLOS is asserted.	Vin (mVppd) * (1.5/16)	Vin (mVppd) * (2 .5/16)	mVppd
Hold Time	Amount of time required for a change in the Threshold must be held before asserting/de-asserting DLOS.	20	250	ns
<b>Backplane SR Interface (C8 SERDES link-up to link-down)</b>				
Threshold	When the SERDES drops below the Threshold for any time greater than the Hold Time, DLOS is asserted.	Vin (mVppd) * (1.5/16)	Vin (mVppd) * (2 .5/16)	mVppd
Hold Time	Amount of time required for a change in the Threshold must be held before asserting/de-asserting DLOS.	60	350	ns
<b>Backplane LR Interface (T8 SERDES link-down to link-up)</b>				
Threshold	When the SERDES drops below the Threshold for any time greater than the Hold Time, DLOS is asserted.	Vin (mVppd) * (3 .5/32)	Vin (mVppd) * (4 .5/32)	mVppd
Hold Time	Amount of time required for a change in the Threshold must be held before asserting/de-asserting DLOS.	60	3250	ns
<b>Backplane LR Interface (T8 SERDES link-up to link-down)</b>				
Threshold	When the SERDES drops below the Threshold for any time greater than the Hold Time, DLOS is asserted.	Vin (mVppd) * (3 .5/32)	Vin (mVppd) * (4 .5/32)	mVppd

Symbol	Parameter	Min	Max	Units
Hold Time	Amount of time required for a change in the Threshold must be held before asserting/de-asserting DLOS.	1270	6350	ns
<b>Line and System Interfaces (S16 SERDES)</b>				
Threshold	When the SERDES drops below the Threshold for any time greater than the Hold Time, DLOS is asserted.	Vin (mVppd) * (13.5 / 32)	Vin (mVppd) * (14.5 / 32)	mVppd
Hold Time	Amount of time required for a change in the Threshold must be held before asserting/de-asserting DLOS. Applicable to both cases of link-up/down and link-down/up.	310	490	ns

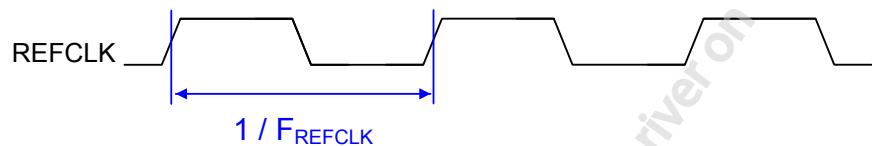
## 15 AC Timing Characteristics

### 15.1 Device Interface

The pin description for this interface can be seen in [Section 4.2.1](#).

#### 15.1.1 REFCLK

**Figure 210** REFCLK AC Timing (Input)



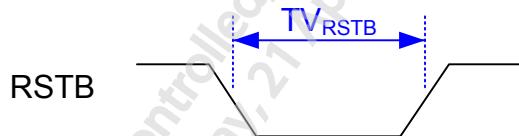
**Table 133** REFCLK AC Timing Parameters (Input)

Symbol	Description	Min	Typ	Max	Units	Note
$F_{REFCLK}$	REFCLK Frequency		155.52		MHz	1
<b>Note:</b>						
1. REFCLK Frequency Tolerance is +/- 20 ppm.						

Please refer to [\[7\]](#) for the REFCLK jitter requirements.

#### 15.1.2 RSTB

**Figure 211** RSTB AC Timing (Input)



**Table 134** RSTB AC Timing Parameters (Input)

Symbol	Description	Min	Typ	Max	Units	Note
$TV_{RSTB}$	RSTB Pulse Width	1000			$\mu s$	
$TV_{RSTB}$	RSTB Pulse Width	100			$\mu s$	1
<b>Note:</b>						
1. Minimum RSTB pulse width after the reference clock input reaches normal operation.						

### 15.1.3 PGMRCLK [3:0]

Figure 212 PGMRCLK[3:0] AC Timing (Output)

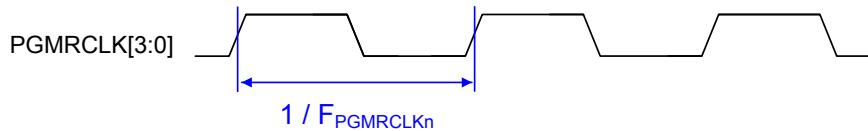


Table 135 PGMRCLK[3:0] AC Timing Parameters (Output)

Symbol	Description	Min	Typ	Max	Units
$F_{\text{PGMRCLK}n}$	PGMRCLK[n] Frequency			125	MHz
	PGMRCLK[n] Clock Duty Cycle	45		55	%

## Note:

1. Minimum PGMRCLK[3:0] frequency depends on the frequency of the selected clock. It can be calculated by dividing the selected clocks frequency by  $2^{20}$ . See [Section 5.16.1.3](#).
2. Outputs are simulated to 30 pF.

### 15.1.4 PLL\_STATUS [1:0]

Figure 213 PLL\_STATUS[1:0] AC Timing (Output)

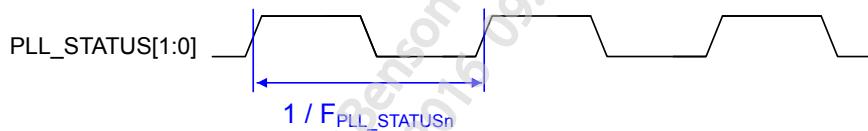


Table 136 PLL\_STATUS[1:0] AC Timing Parameters (Output)

Symbol	Description	Min	Typ	Max	Units
$F_{\text{PLL\_STATUS}0}$	PLL_STATUS[0] Frequency	607.5			kHz
$F_{\text{PLL\_STATUS}1}$	PLL_STATUS[1] Frequency	781.25			kHz
	PLL_STATUS[1:0] Clock Duty Cycle	45		55	%

## Note:

1. PLL\_STATUS[0] is REFCLK divided by 256, PLL\_STATUS[1] is PCLK divided by 256. See [Section 5.16.1.3](#).
2. Outputs are simulated to 30 pF.

### 15.1.5 PMON\_SYNC\_I

PMON\_SYNC\_I is considered an asynchronous signal, pulse width requirements are specified below in [Figure 214](#) and [Table 137](#).

Figure 214 PMON\_SYNC\_I AC Timing (Input)

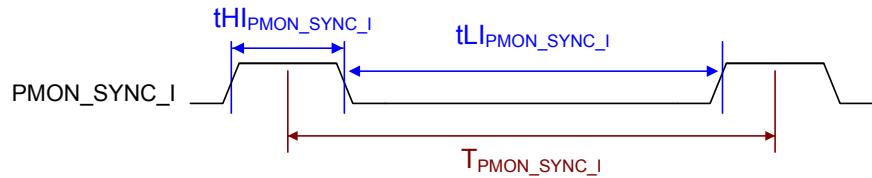


Table 137 PMON\_SYNC\_I AC Timing Parameters (Input)

Symbol	Description	Min	Typ	Max	Units
$T_{PMON\_SYNC\_I}$	PMON_SYNC_I Period		1		s
$t_{Hi}_{PMON\_SYNC\_I}$	PMON_SYNC_I High Pulse Width	12.86			ns
$t_{Li}_{PMON\_SYNC\_I}$	PMON_SYNC_I Low Pulse Width	12.86			ns

### 15.1.6 PMON\_SYNC\_O

PMON\_SYNC\_O is considered an asynchronous signal, pulse width requirements are specified below in [Figure 215](#) and [Table 138](#).

Figure 215 PMON\_SYNC\_O AC Timing (Output)

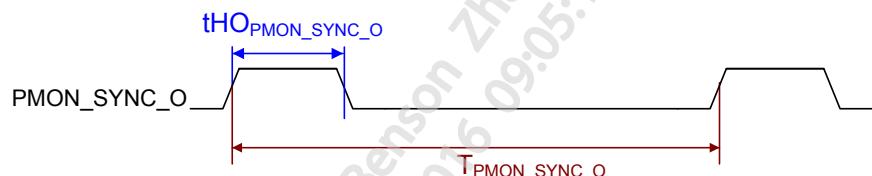


Table 138 PMON\_SYNC\_O AC Timing Parameters (Output)

Symbol	Description	Min	Typ	Max	Units
$T_{PMON\_SYNC\_O}$	PMON_SYNC_O Period	20.84 $\mu$ s		2.048s	
$t_{HO}_{PMON\_SYNC\_O}$	PMON_SYNC_O High Pulse Width	44	48.23		ns

**Note:**

1. The period of the PMON\_SYNC\_O output is configurable. See [Section 5.16.1](#).
2. Outputs are simulated to 30 pF
3. In bypass mode, the period is equal to that of the incoming PMON\_SYNC\_I pulse.
4. In bypass mode, the minimum pulse width is 9ns.

### 15.1.7 FPI

Figure 216 FPI AC Timing (Input)

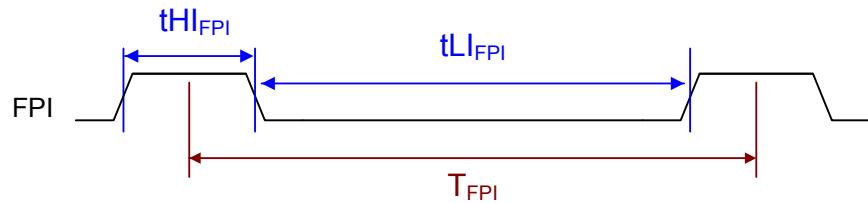


Table 139 FPI AC Timing Parameters (Input)

Symbol	Description	Min	Typ	Max	Units
$T_{FPI}$	Frame Pulse Input Period		125		$\mu s$
$tHI_{FPI}$	Frame Pulse Input High Pulse Width	12.86			ns
$tLI_{FPI}$	Frame Pulse Input Low Pulse Width	12.86			ns

## 15.2 Line Interface

### 15.2.1 LINE\_REFCLK\_P/N[11:0] AC Timing Characteristics

AC timing specifications for the LINE\_REFCLK\_P/N[11:0] outputs are defined in [Table 140](#).

Table 140 LINE\_REFCLK\_P/N AC Timing Parameters

Symbol	Description	Min	Typ	Max	Units
LN_SYSREF_CFP Tjorms	LINE_REFCLK output Jitter for CFP Modules	-	-	10	ps RMS (10 kHz to 10 MHz)
LN_SYSREF_XFP Tjorms	LINE_REFCLK output Jitter for XFP Modules <sup>1</sup>	-	-	10	ps RMS 
	LINE_REFCLK[n] Clock Duty Cycle	40	-	60	%

**Note:**

1. Does not meet the XFI optional synchronous CMU clock specification.

## 15.3 System Interface

### 15.3.1 SYS\_REFCLK\_P/N[11:0] AC Timing Characteristics

AC timing specifications for the SYS\_REFCLK\_P/N[11:0] outputs are defined in [Table 141](#).

**Table 141 SYS\_REFCLK\_P/N AC Timing Parameters**

<b>Symbol</b>	<b>Description</b>	<b>Min</b>	<b>Typ</b>	<b>Max</b>	<b>Units</b>
LN_SYSREF_CFP Tjorms	SYS_REFCLK output Jitter for CFP Modules	-	-	10	ps RMS (10 kHz to 10 MHz)
LN_SYSREF_XFP Tjorms	SYS_REFCLK output Jitter for XFP Modules <sup>1</sup>	-	-	10	ps RMS (<100 MHz)
	SYS_REFCLK[n] Clock Duty Cycle	40	-	60	%

**Note:**

1. Does not meet the XFI optional synchronous CMU clock specification.

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:12 PM

## 15.4 Interlaken Flow Control Interface

Figure 217 Interlaken Mode OOB Flow Control Interface AC Timing

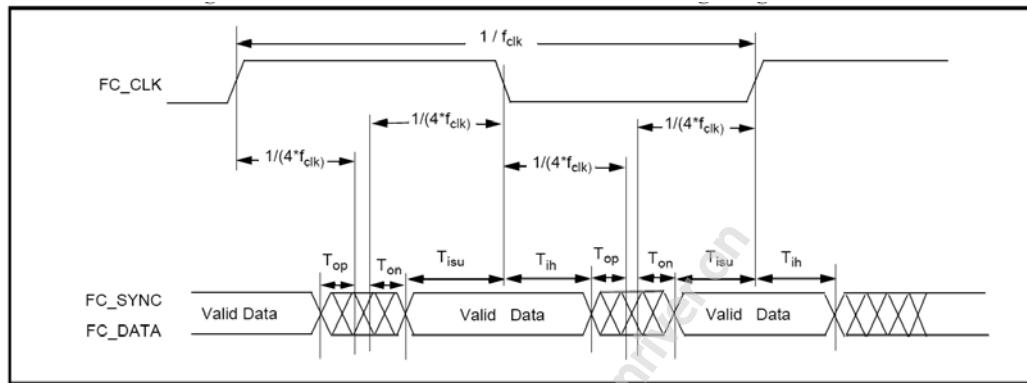


Table 142 Interlaken Mode OOB Flow Control Interface AC Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units
$f_{clk}$	FC_CLK Frequency		100		MHz
	FC_CLK Clock Duty Cycle	45		55	%
$T_{isu}$	FC_TDATA/FC_TSYNC Setup Time for the FC_TCLK Edge	0.75			ns
$T_{ih}$	FC_TDATA/FC_TSYNC Hold Time for the FC_TCLK Edge	0.75			ns
$T_{on}$	Next FC_RDATA/FC_RSYNC Invalid Time for the Previous Quadrature Point of the Next FC_RCLK Edge			0.75	ns
$T_{op}$	Previous FC_RDATA/FC_RSYNC Invalid Time for the Quadrature Point of the Previous FC_RCLK Edge			0.75	ns

**Note:**

1. FC\_RCLK[1:0], FC\_RDATA[1:0] and FC\_RSYNC[1:0] have an output load of 30pF.

The pin description for this interface can be seen in [Section 4.2.7](#).

## 15.5 OTN Frame Pulse Interface

Table 143 OTN Frame Pulse Interface AC Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units
Output Load	Load on OTU_RX_MFPO[11:0] and OTU_TX_MFPO[11:0]			30	pF
Output Transition Time	Output transition time on OTU_RX_MFPO[11:0] and OTU_TX_MFPO[11:0]	2.7		2.7	ns

For OTN Frame Pulse Interface functional timing, please see [Section 8.3](#).

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:12 PM

## 15.6 4-Wire OTN FEC Statistics Interface

Figure 218 4-Wire OTN FEC Statistics AC Timing

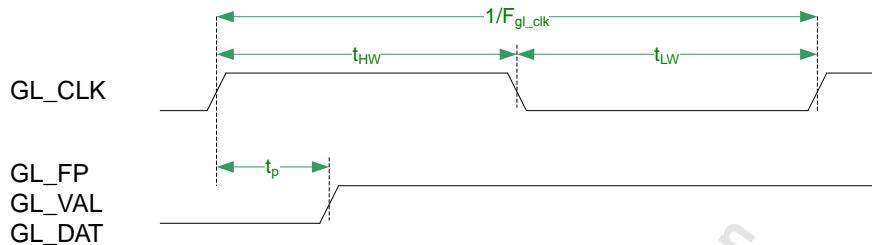


Table 144 4-Wire OTN FEC Statistics AC Timing Parameters

Symbol	Description	Min	Max	Units
$F_{gl\_clk}$	GL_CLK Frequency		20	MHz
$t_{HW}$	GL_CLK High Pulse Width	20		ns
$t_{LW}$	GL_CLK Low Pulse Width	20		ns
$t_p$	GL_FP, GL_VAL, GL_DAT propagation time from rising edge of GL_CLK	1	12	ns

The pin description for this interface can be seen in [Section 4.2.11](#).

## 15.7 TOD and IRIG Interface

The pin description for this interface can be seen in [Section 4.2.14](#).

### 15.7.1 TOD Interface

**Table 145** TOD Interface AC Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units
TOD_Fos	IRIG_CLKO		100		Hz
TOD_DCos	IRIG_CLKO Duty Cycle	45		55	%
TOD_Tcq	IRIG_CLKO Falling Edge to IRIG_DATO Valid	-2		2	ns
TOD_PPS_Fos	ONE_PPS_O Output Frequency		1		Hz
TOD_PPS_opw	ONE_PPS_O Output Pulse Width		1		ms
TOD_Fis	IRIG_CLKI	100		1000	Hz
TOD_DCs	IRIG_CLKI Duty Cycle	45		55	%
TOD_Ts	IRIG_DATI Setup Time for the Rising Edge of IRIG_CLKI	2			ns
TOD_Th	IRIG_DATI Hold Time for the Rising Edge of IRIG_CLKI	2			ns
TOD_PPS_Fis	ONE_PPS_I Input Frequency		1		Hz
TOD_PPS_ipw	ONE_PPS_I Input Pulse Width	32			ns

**Note:**

1. IRIG\_CLKO and IRIG\_DATO have an output load of 20pF.

### 15.7.2 IRIG-B B00x Interface

**Table 146** IRIG-B B00x Interface AC Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units
IRIG_Piw0	IRIG_DATI High Pulse Width for Logic 0	1.9	2	2.1	ms
IRIG_Piw1	IRIG_DATI High Pulse Width for Logic 1	4.9	5	5.1	ms
IRIG_Piwp	IRIG_DATI High Pulse Width for Position Identifiers	7.9	8	8.1	ms
IRIG_Pow0	IRIG_DATO High Pulse Width for Logic 0		2		ms
IRIG_Pow1	IRIG_DATO High Pulse Width for Logic 1		5		ms
IRIG_Powp	IRIG_DATO High Pulse Width for Position Identifiers		8		ms

**Note:**

1. IRIG\_Piw0, IRIG\_Piw1, and IRIG\_Piwp are based on PTP\_CLK = 125 MHz and the default values of the following configuration bits: PWM\_NOM\_PERIOD, PWM\_NOM\_ZERO\_WIDTH, PWM\_NOM\_ONE\_WIDTH, PWM\_NOM\_PX\_WIDTH, PWM\_MIN\_DIFF, and PWM\_MAX\_DIFF.
2. Typical values for IRIG\_Pow0, IRIG\_Pow1, and IRIG\_Powp are based on PTP\_CLK = 125 MHz and the default values of the following configuration bits: PWM\_NOM\_PERIOD, PWM\_NOM\_ZERO\_WIDTH, PWM\_NOM\_ONE\_WIDTH, and PWM\_NOM\_PX\_WIDTH.

## 15.8 SPI Master Interface

Figure 219 SPI Master Interface AC Timing

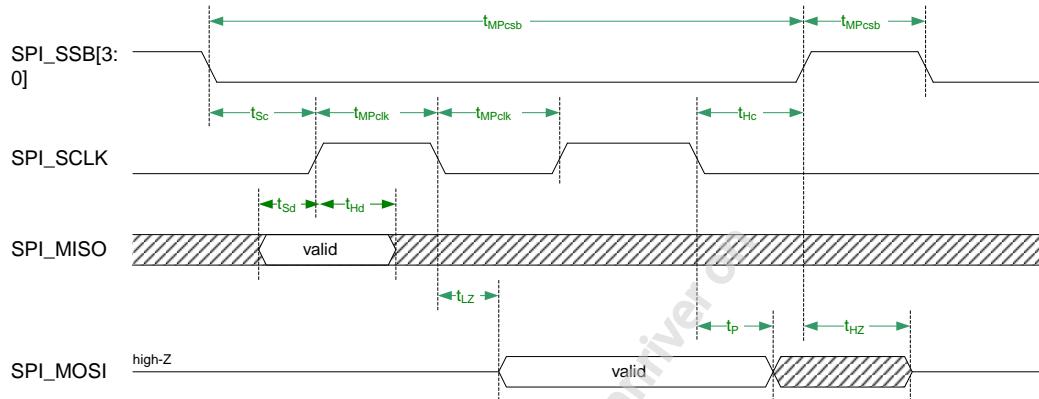


Table 147 SPI Master Interface AC Timing Parameters

Symbol	Description	Min	Typ	Max	Units
$T_{SCLK}$	SPI_SCLK Period <sup>1</sup>	50	1000		ns
$t_{MPcsb}$	SPI_SSB High/Low Pulse Width	$T_{SCLK}/2-5$			ns
$t_{MPclk}$	SPI_SCLK High/Low Pulse Width	$T_{SCLK} * 0.45$			ns
$t_{Sc}$	SPI_SSB Setup Time to the Rising Edge of SPI_SCLK	$T_{SCLK}/2-5$			ns
$t_{Hc}$	SPI_SSB Hold Time from the Falling Edge of SPI_SCLK	$T_{SCLK}/2-5$			ns
$t_{Sd}$	SPI_MISO Setup Time to the Rising Edge of SPI_SCLK	9			ns
$t_{Hd}$	SPI_MISO Hold Time from the Rising Edge of SPI_SCLK	1			ns
$t_P$	SPI_MOSI Propagation Time from the Falling Edge of SPI_SCLK	-5		5	ns
$t_{LZ}$	SPI_MOSI Turn-on Time from the Falling Edge of SPI_SCLK	-5		5	ns
$t_{HZ}$	SPI_MOSI Turn-off Time from the Rising Edge of SPI_SCLK	-5		5	ns

**Note:**

1.  $T_{SCLK}$  is 1000 ns at startup.
2. SPI\_MOSI and SPI\_SCLK have an output load of 60pF.
3. SPI\_SSB[3:0] have an output load of 30pF.

The pin description for this interface can be seen in [Section 4.2.15](#).

## 15.9 JTAG Interface

Figure 220 JTAG AC Timing

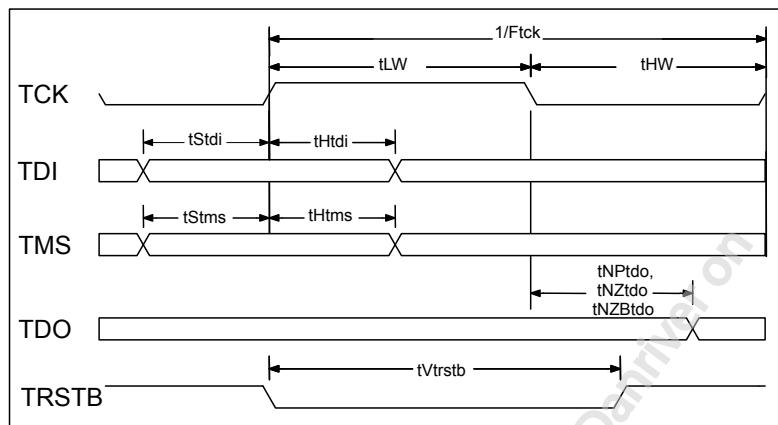


Table 148 JTAG Port Interface AC Timing Parameters

Symbol	Description	Min	Max	Units
$F_{TCK}$	TCK Frequency	-	5	MHz
$t_{LW}$	TCK Low Pulse Width	100	-	ns
$t_{HW}$	TCK High Pulse Width	100	-	ns
$t_{S_{TMS}}$	TMS Setup Time to TCK	35	-	ns
$t_{H_{TMS}}$	TMS Hold Time to TCK	35	-	ns
$t_{S_{TDI}}$	TDI Setup Time to TCK	35	-	ns
$t_{H_{TDI}}$	TDI Hold Time to TCK	35	-	ns
$t_{NP_{TDO}}$	TCK Falling to TDO Valid	2	50	ns
$t_{NZ_{TDO}}$	TCK Falling to TDO High-Impedance	2	50	ns
$t_{NZB_{TDO}}$	TCK Falling to TDO Driven	2	50	ns
$t_{V_{TRSTB}}$	TRSTB Pulse Width	200	-	ns

**Note:**

1. TDO has an output load of 30pF.

The pin description for this interface can be seen in [Section 4.2.16](#).

**16      End of Document**

Downloaded [controlled] by Benson Zhang of Danriver on  
Thursday, 21 April, 2016 09:05:12 PM