Summary

Matching firmware versions are:

- 080514_SP1_ISE92_chain0.svf
- 080514_SP_ISE92_chain1_af51.svf – AF is built with an obsolete Asynchronous FIFO core v5.1 and features 2.X clock cycles minimal latency, where X is write-to-read clock phase difference and “clock cycle” is a period of doubled RF.
- 080514_SP_ISE92_chain1_fg52.svf – AF is built with an active FIFO Generator core v4.2 and features 6.X clock cycles minimal latency, where X is a write-to-read clock phase difference and “clock cycle” is a period of doubled RF.

Eliminated Registers

Table 1: Register Address Field Format for Non Privileged Data Access

<table>
<thead>
<tr>
<th>RA, hex</th>
<th>Register Label</th>
<th>Description</th>
<th>Valid CA / Valid MA</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>SP</td>
<td>DD</td>
</tr>
<tr>
<td>Data Register Group</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x7A</td>
<td>DAT_FTR</td>
<td>Fake Track Data</td>
<td>Eliminated</td>
<td></td>
</tr>
</tbody>
</table>

Added Registers

Table 2: Register Address Field Format for Non Privileged Data Access

<table>
<thead>
<tr>
<th>RA, hex</th>
<th>Register Label</th>
<th>Description</th>
<th>Valid CA / Valid MA</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>SP</td>
<td>DD</td>
</tr>
<tr>
<td>Control / Status Register Group</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x27</td>
<td>CSR_LQE</td>
<td>LCT Quality Enable</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>0x2F</td>
<td>CSR_BCD</td>
<td>BC0 Delay</td>
<td>-</td>
<td>MA</td>
</tr>
</tbody>
</table>

Action Register Group update

ACT_LCR – Link Counter Reset

FA:

Writing Logic ONE to specified bit(s) of this write-only register results in sending a reset pulse to selected counter(s) described under the CSR_LNK, CSR_LEC, DAT_VPC, and CSR_OSY headings.

Table 3: ACT_LCR Data Format for FRONT_FPGA

<table>
<thead>
<tr>
<th>D15</th>
<th>D14</th>
<th>D13</th>
<th>D12</th>
<th>D11</th>
<th>D10</th>
<th>D9</th>
<th>D8</th>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
<th>TRN</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFUR</td>
<td>BCLR</td>
<td>BCER</td>
<td>BXMR</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>VPR</td>
<td>TER</td>
<td>SLR</td>
<td>CER</td>
<td>EWR</td>
</tr>
</tbody>
</table>

Here:

- X – Don’t care bit;
- AFUR – “Alignment FIFO Underflow” error counter Reset;
- BCLR – “BC0 arrived Late” error counter Reset;
- BCER – “BC0 arrived Early” error counter Reset;
- BXMR – “BX0 and BXN[0] Mismatch” error counter Reset;
- EWR – TLK2501 Error Word Counter Reset in the CSR_LEC register (RXDV == HIGH, RXER == HIGH);
- CER – TLK2501 Carrier Extend Counter Reset in the CSR_LEC register (RXDV == LOW, RXER == HIGH);
- SLR – FINISAR optical receiver Signal Loss Counter Reset in the CSR_LEC register (RXSD goes LOW);
- TER – PRBS Test Error Counter Reset in the CSR_LNK register;
- VPR – Valid Pattern Counter Reset in the DAT_VPC register.

SP:

Writing Logic ONE to specified bit(s) of this write-only register results in sending a reset pulse to selected counter(s) described under DAT_VPC heading.

Table 4: ACT_LCR Data Format for SP_FPGA

<table>
<thead>
<tr>
<th>D15</th>
<th>D14</th>
<th>D13</th>
<th>D12</th>
<th>D11</th>
<th>D10</th>
<th>D9</th>
<th>D8</th>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
<th>TRN</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WR</td>
</tr>
</tbody>
</table>

Here:
- X – Don’t care bit;
- VPR – Valid Pattern Counter Reset in the DAT_VPC register;
- TCR – Track Counter Reset (readable via DAQ only);
- OCR – Orbit Counter Reset (readable via DAQ only).

Control/Status Register Group update

CSR_AF – Alignment FIFO Status
This read-only register shows the Alignment FIFO (AF) word count.

FA:

In the FRONT_FPGA Alignment FIFOs are used to switch from the receiver clock domain to the system clock domain and to compensate for different optical link latencies. Different links may show different AF word counts after link synchronization procedure has been performed. Dispersion of word count values corresponds to the dispersion of link latencies. To time-in all active links and minimize the overall AF latency the user has to adjust the CSR_AFD register value and repeat TTC_RSYNC commands until the minimal AF value amongst all active links becomes equal to 2 or 3. The FRONT_FPGA AF runs at 80 MHz clock, so the AF latency in bunch crossings is twice less the AFC value.
The AFIFO Finite State Machine (FSM) gives further details on the AFIFO status. After receiving a TTC_RSYNC command the AFIFO FSM enters the INIT state. IDLE characters from the MPC push the AFIFO FSM into the IDLE state. When MPC resumes sending data frames, the AFIFO FSM detects normal data characters received by the TLK2501 deserializer and enables writes (af_wen input) to the AFIFO starting from the third received frame or from the second bunch-crossing. AFIFO resumes reads after the CSR_AFD delay. The Alignment FIFO Error flag indicates that the TTC_RSYNC has failed and needs to be repeated with the increased CSR_AFD value.

Table 5: CSR_AF Data Format for FRONT_FPGA

<table>
<thead>
<tr>
<th>D15</th>
<th>D14</th>
<th>D13</th>
<th>D12</th>
<th>D11</th>
<th>D10</th>
<th>D9</th>
<th>D8</th>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
<th>TRN</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFFF</td>
<td>AFEF</td>
<td>AFER</td>
<td>0</td>
<td>WREN</td>
<td>FRM1</td>
<td>IDLE</td>
<td>INIT</td>
<td>AFC7</td>
<td>AFC6</td>
<td>AFC5</td>
<td>AFC4</td>
<td>AFC3</td>
<td>AFC2</td>
<td>AFC1</td>
<td>AFC0</td>
<td>RD</td>
</tr>
</tbody>
</table>

Here:
- AFFF – Alignment FIFO Full Flag or AFC = 255;
- AFEF – Alignment FIFO Empty Flag;
- AFER – Alignment FIFO Error Flag;
- NU – Not Used;
- AFIFO FSM one-hot states: Write Enable (AF_WREN), Frame 1 (AF_FRM1), Idle (AF_IDLE) and Init (AF_INIT); Power-up default state is AF_INIT = 1;
- AFC [7:0] = 0...255 – Alignment FIFO Read Word Count;

DD:
In the DDU_FPGA there are two Alignment FIFOs: one at the TLK2501 output and the other at the TLK2501 input. The Alignment FIFOs perform function of the elastic output and input buffers to compensate for differences between the doubled RF clock of 80.1574 MHz and the DDU link reference clock of 80.0000 MHz.

Table 6: CSR_AF Data Format for DDU_FPGA

<table>
<thead>
<tr>
<th>D15</th>
<th>D14</th>
<th>D13</th>
<th>D12</th>
<th>D11</th>
<th>D10</th>
<th>D9</th>
<th>D8</th>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFFF</td>
<td>AFEF</td>
<td>0</td>
<td>AFR4</td>
<td>AFR3</td>
<td>AFR2</td>
<td>AFR1</td>
<td>AFR0</td>
<td>AFFF</td>
<td>AFEF</td>
<td>0</td>
<td>AFW4</td>
<td>AFW3</td>
<td>AFW2</td>
<td>AFW1</td>
<td>AFW0</td>
</tr>
</tbody>
</table>

Here:
- AFR [4:0] = 0...31 – Input Alignment FIFO Read Word Count;
- AFW [4:0] = 0...31 – Output Alignment FIFO Write Word Count;
- AFFF – Alignment FIFO Full Flag: AFWC = 31 or AFRC = 31;
- AFEF – Alignment FIFO Empty Flag.
SP:
In the SP_FPGA Alignment FIFO performs the same function for barrel muon data as it does in
the FRONT_FPGA for the EMU muon data. The difference is that to time-in the barrel data the
user has to adjust only the CSR_AFD value, no TTC_RSYNC is required. The SP_FPGA AF
runs at 40 MHz clock, so the AF latency in bunch crossings equals to the AFC value.

Table 7: CSR_AF Data Format for SP_FPGA

| D15 | D14 | D13 | D12 | D11 | D10 | D9  | D8  | D7  | D6  | D5  | D4  | D3  | D2  | D1  | D0  | TRN |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| AFFF| AFEF| 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | AFFC3| AFFC2| AFFC1| AFFC0| RD  |

Flags

Alignment FIFO Word Count

Here:

- AFC[3:0] = 0...15 – Alignment FIFO Read Word Count;
- AFFF – Alignment FIFO Full Flag or AFC = 15;
- AFEF – Alignment FIFO Empty Flag.

CSR_BCD – BC0 Delay

DD:
This read/write register sets additional delay for TTC_BC0 command to compensate for L1A
latency, so the bunch counter running at TTC timing could be synchronized to bunch counters
running at link’s timing.

Table 8: CSR_BCD Data Format for DD

| D15 | D14 | D13 | D12 | D11 | D10 | D9  | D8  | D7  | D6  | D5  | D4  | D3  | D2  | D1  | D0  | TRN |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| X   | X   | X   | X   | X   | X   | BCD9| BCD8| BCD7| BCD6| BCD5| BCD4| BCD3| BCD2| BCD1| BCD0| RD  |
| 0   | 0   | 0   | 0   | 0   | 0   | BCD9| BCD8| BCD7| BCD6| BCD5| BCD4| BCD3| BCD2| BCD1| BCD0| RD  |

BC0 Delay

Here:

- BCD[9:0] = 0 (default) … 1023 = BC0 delay for the L1A_BXN counter to
  compensate for L1A latency.

CSR_LQE – LCT Quality Enable

FA:
This read/write register enables LCTs with qualities 15 to 1 to be passed to the SP core for track-
finding. If a certain quality is disabled then the VP bit for an LCT with such quality gets reset,
when forwarded to the SP_FPGA. Original LCTs will nevertheless show up in the ME readout
block, if requested by L1A. There is one register per each FRONT_FPGA, 5 registers in total.

Table 9: CSR_LQE Data Format for F1|F2|F3|F4|F5

| D15 | D14 | D13 | D12 | D11 | D10 | D9  | D8  | D7  | D6  | D5  | D4  | D3  | D2  | D1  | D0  | TRN |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Q15E| Q14E| Q13E| Q12E| Q11E| Q10E| Q9E | Q8E | Q7E | Q6E | Q5E | Q4E | Q3E | Q2E | Q1E | Q0E | WR  |
| Q15E| Q14E| Q13E| Q12E| Q11E| Q10E| Q9E | Q8E | Q7E | Q6E | Q5E | Q4E | Q3E | Q2E | Q1E | Q0E | RD  |

LCT Quality Enables

Here:

- Q15E … Q0E = 0 / 1 (default) = disable / enable (default) LCT with quality 15 … 0.
**CSR_REQ – L1 Request Configuration**

**VM:**

This read/write register allows setting the delay for L1REQ, being sent by the SP to the backplane. The delay in implemented with the Synchronous FFO, so besides the delay setting, the register allows to read the actual REQ FIFO word count.

**Table 10: CSR_REQ Data Format**

<table>
<thead>
<tr>
<th>D15</th>
<th>D14</th>
<th>D13</th>
<th>D12</th>
<th>D11</th>
<th>D10</th>
<th>D9</th>
<th>D8</th>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>RFC6</td>
<td>RFC5</td>
<td>RFC4</td>
<td>RFC3</td>
<td>RFC2</td>
<td>RFC1</td>
<td>RFC0</td>
<td>X</td>
<td>LRD6</td>
<td>LRD5</td>
<td>LRD4</td>
<td>LRD3</td>
<td>LRD2</td>
<td>LRD1</td>
<td>LRD0</td>
</tr>
</tbody>
</table>

**Request FIFO Word Count**

**L1 Request Delay [6:0]**

Here:
- **X** – don’t care bit for writes and zero for reads;
- **LRD [6:0] = 0 (default)...127** – Additional delay for L1Request signal being sent to the Backplane;
- **RFC [6:0] = 0 (default)...127** – Actual Request FIFO Word Count.

**SP:**

Historically, this read/write register allows choosing the source of a L1 request signal to be sent by the SP to the backplane. Normally, the L1 Request is an OR of tracks with none-zero Modes, found by the SP core logic. The user can also chose the L1 Request to be generated on an OR of Valid Pattern bit occurrences for enabled CSC muons and non-zero quality occurrences for enabled DT muons.

Beginning with firmware version 080421 single triggers from enabled ME VP or/and MB quality bits generate pseudo tracks being passed to the MS with Mode = 11 = 0xB, and the CORE bit also controls the SP_CORE output to the MS.

**Table 11: CSR_REQ Data Format**

<table>
<thead>
<tr>
<th>D15</th>
<th>D14</th>
<th>D13</th>
<th>D12</th>
<th>D11</th>
<th>D10</th>
<th>D9</th>
<th>D8</th>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORE</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>MB1D</td>
<td>MB1A</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>ME4</td>
<td>ABC</td>
<td>ME3</td>
<td>ABC</td>
<td>ME2</td>
</tr>
</tbody>
</table>

**L1 Request Enable**

Here:
- **X** – don’t care bit for writes and zero for reads;
- **ME1ABC = 0 (default) /1** – Disable (default) / Enable VP-bits of ME1A, ME1B or ME1C muons (from F1) to be the source of L1 Request and pseudo tracks to be passed to the MS;
- **ME1DEF = 0 (default) /1** – Disable (default) / Enable VP-bits of ME1D, ME1E or ME1F muons (from F2) to be the source of L1 Request and pseudo tracks to be passed to the MS;
- **ME2ABC = 0 (default) /1** – Disable (default) / Enable VP-bits of ME2A, ME2B or ME2C muons (from F3) to be the source of L1 Request and pseudo tracks to be passed to the MS;
- **ME3ABC = 0 (default) /1** – Disable (default) / Enable VP-bits of ME3A, ME3B or ME3C muons (from F4) to be the source of L1 Request and pseudo tracks to be passed to the MS;
− ME4ABC = 0 (default) / 1 – Disable (default) / Enable VP-bits of ME4A, ME4B or ME4C muons (from F5) to be the source of L1 Request and pseudo tracks to be passed to the MS;
− MB1A = 0 (default) / 1 – Disable (default) / Enable non-zero quality of MB1A muons to be the source of L1 Request and pseudo tracks to be passed to the MS;
− MB1D = 0 (default) / 1 – Disable (default) / Enable non-zero quality of MB1D muons to be the source of L1 Request and pseudo tracks to be passed to the MS;
− CORE = 0 / 1 (default) – Disable / Enable (default) none-zero Mode output of the SP core to be the source of L1 Request and SP core tracks to be passed to the MS.

**CSR_SCC – SP Core Configuration**

**SP:**

This read/write register keeps the SP core configuration options. The register is protected against accidental accesses: in order to get VME access to this register the SP should be set to the VME fast control mode.

Please note that the register format has slightly changed: DTE bit has moved from D8 to D7 to free up space for Pre-Trigger control setting. Also, the BXA control has increased by 1 BX. Core Version bits are read-only bits. **PRE and BXA fields of the register may be loaded only with specified values, core behavior for other values is unspecified.**

### Table 12: CSR_SCC Data Format for SP_FPGA (SP Core Version 0)

<table>
<thead>
<tr>
<th>D15</th>
<th>D14</th>
<th>D13</th>
<th>D12</th>
<th>D11</th>
<th>D10</th>
<th>D9</th>
<th>D8</th>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
<th>TRN</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>DTE</td>
<td>X</td>
<td>X</td>
<td>Q4EN</td>
<td>Q3EN</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>BXE</td>
<td>WR</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>DTE</td>
<td>0</td>
<td>0</td>
<td>Q4EN</td>
<td>Q3EN</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>BXE</td>
<td>RD</td>
</tr>
</tbody>
</table>

Not Used Core Version Mode Control Bunch Crossing Analyzer

Here:
− X – Don’t care bit for writes and zero for reads;
− BXE = 0 (default) / 1 – disable (default) / enable Bunch Crossing Analyzer;
− DTE = 0 (default) / 1 – disable (default) / enable Drift Tube data inputs to SP Core;
− Q3EN = 0 (default) / 1 – disable (default) / enable processing stubs with Quality = 3;
− Q4EN = 0 (default) / 1 – disable (default) / enable processing stubs with Quality = 4.

### Table 13: CSR_SCC Data Format for SP_FPGA (SP/MA/CSR_SID dated Mar 9, 2008 or +5 bx release)

<table>
<thead>
<tr>
<th>D15</th>
<th>D14</th>
<th>D13</th>
<th>D12</th>
<th>D11</th>
<th>D10</th>
<th>D9</th>
<th>D8</th>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
<th>TRN</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>PRE1</td>
<td>PRE0</td>
<td>DTE</td>
<td>X</td>
<td>Q4EN</td>
<td>Q3EN</td>
<td>X</td>
<td>X</td>
<td>BXA1</td>
<td>BXA0</td>
<td>WR</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>PRE1</td>
<td>PRE0</td>
<td>DTE</td>
<td>0</td>
<td>Q4EN</td>
<td>Q3EN</td>
<td>0</td>
<td>0</td>
<td>BXA1</td>
<td>BXA0</td>
<td>RD</td>
</tr>
</tbody>
</table>

Not Used Core Version Pre-Trigger Mode Control Bunch Crossing Analyzer

Here:
− X – Don’t care bit for writes and zero for reads;
− PRE [1:0] = 1, 2 (default), 3 – Pre-Trigger control;
− DTE = 0 (default) / 1 – disable (default) / enable Drift Tube data inputs to SP Core;
− Q3EN = 0 (default) / 1 – disable (default) / enable processing stubs with Quality = 3;
− Q4EN = 0 (default) / 1 – disable (default) / enable processing stubs with Quality = 4;
− BXA [1:0] = 0, 1, 2 (default) – Bunch Crossing Analyzer history control.
Table 14: CSR_SCC Data Format for SP_FPGA (SP/MA/CSR_SID dated Apr 2, 2008 or +4 bx release)

<table>
<thead>
<tr>
<th>D15</th>
<th>D14</th>
<th>D13</th>
<th>D12</th>
<th>D11</th>
<th>D10</th>
<th>D9</th>
<th>D8</th>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
<th>TRN</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>PRE1</td>
<td>PRE0</td>
<td>DTE</td>
<td>X</td>
<td>X</td>
<td>Q2EN</td>
<td>X</td>
<td>X</td>
<td>BXA1</td>
<td>BXA0</td>
<td>WR</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>PRE1</td>
<td>PRE0</td>
<td>DTE</td>
<td>0</td>
<td>Q2EN</td>
<td>X</td>
<td>X</td>
<td>BXA1</td>
<td>BXA0</td>
<td>RD</td>
<td></td>
</tr>
</tbody>
</table>

Here:
- **X** – Don’t care bit for writes and zero for reads;
- **PRE [1:0] = [1, 2]** (default), 3 – Pre-Trigger control;
- **DTE = 0** (default) / 1 – disable (default) / enable Drift Tube data inputs to SP Core;
- **Q1EN = 0** (default) / 1 – disable (default) / enable processing stubs with Quality = 1;
- **Q2EN = 0** (default) / 1 – disable (default) / enable processing stubs with Quality = 2;
- **BXA [1:0] = [0, 1, 2]** (default) – Bunch Crossing Analyzer history control.

CSR_OSY – Out-of-Synch Control / Status

**VM:**

In the VME_FPGA the CSR_OSY register displays status of seven input and one output OSY lines. Besides, it carries eight mask bits, so each input or/and output can be either disabled (mask bit = 0) or enabled (mask bit = 1 => default):

\[
\text{OSY0} = (\text{OSY1} \cdot \text{OSM1} + \text{OSY2} \cdot \text{OSM2} + \text{OSY3} \cdot \text{OSM3} + \text{OSY4} \cdot \text{OSM4} + \text{OSY5} \cdot \text{OSM5} + \text{OSY6} \cdot \text{OSM6} + \text{OSY7} \cdot \text{OSM7}) \cdot \text{OSM0}
\]

Indexes 0…7 stand for chip numbers. By default all masks are set to enable state.

Table 15: CSR_OSY Data Format for VME_FPGA

<table>
<thead>
<tr>
<th>D15</th>
<th>D14</th>
<th>D13</th>
<th>D12</th>
<th>D11</th>
<th>D10</th>
<th>D9</th>
<th>D8</th>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSM7</td>
<td>OSM6</td>
<td>OSM5</td>
<td>OSM4</td>
<td>OSM3</td>
<td>OSM2</td>
<td>OSM1</td>
<td>OSM0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>WR</td>
<td></td>
</tr>
<tr>
<td>OSM7</td>
<td>OSM6</td>
<td>OSM5</td>
<td>OSM4</td>
<td>OSM3</td>
<td>OSM2</td>
<td>OSM1</td>
<td>OSM0</td>
<td>OSY7</td>
<td>OSY6</td>
<td>OSY5</td>
<td>OSY4</td>
<td>OSY3</td>
<td>OSY2</td>
<td>OSY1</td>
<td>OSY0</td>
<td>RD</td>
</tr>
</tbody>
</table>

Here:
- **X** – Don’t care bit;
- **OSM [7:0]** – Out-of-Synch Chip mask for SP, DD, F5…F1 and VM chips;
- **OSY [7:0]** – Out-of-Synch status for SP, DD, F5…F1 and VM chips.

**FA:**

In the FRONT_FPGA, each MPC-to-SP link has a Bunch Crossing Counter (BXN) associated with it. On power-up and/or on FC_RSYNC command the BXN presets to a 0xDEC=3564 value. On every BC0 mark the BXN gets from the link after FC_RSYNC command it resets to 0 and then starts counting up to 3563. Normally, next BC0 arrives exactly when the BXN is at 3563 and resets it back to 0. If BC0 arrives later (longer orbit) or does not come at all, the BXN reaches its max value of 0xDEC=3564 and stays at it, until next BC0 is received. If BC0 arrives earlier (shorter orbit) then the BXN resets to 0 earlier.

In between FC_RSYNC commands the following error conditions are detected and reported:
- If link BX0 does not match BXN [0] for BX with VP=1, then the “BX0 and BXN[0] Mismatch” (BXM[3:0]) error counter increments by 1;
- If link BC0 comes later, then the “BC0 arrived Later” (BCL[3:0]) error counter increments by 1;
− If link BC0 comes earlier, then the “BC0 arrived Earlier” (BCE[3:0]) error counter increments by 1;
− If the link Alignment FIFO gets empty, then the “AF Underflow” (AFU[2:0]) error counter increments by 1 for each frame read out from the empty Alignment FIFO;

Any counter stops incrementing, when it reaches its maximum value of 15 or 7;

Only the “AF Underflow” error can be reported to the FMM as an “Out-of-Sync” condition, since it is the only “fatal” error, which requires a TTC_RSYNC.

All counters are reset on power-up, with SOFT_RESET, FC_RSYNC and, individually, with ACT_LCR command.

There is a separate CSR_OSY register for each link, 15 registers in 5 FRONT_FPGAs in total.

Table 16: CSR_OSY Data Format for FRONT_FPGA

<table>
<thead>
<tr>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
<th>D11</th>
<th>D10</th>
<th>D9</th>
<th>D8</th>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFEN</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>WR</td>
<td>WR</td>
</tr>
<tr>
<td>AFEN</td>
<td>AFU2</td>
<td>AFU1</td>
<td>AFU0</td>
<td>BCL3</td>
<td>BCL2</td>
<td>BCL1</td>
<td>BCL0</td>
<td>BCE3</td>
<td>BCE2</td>
<td>BCE1</td>
<td>BCE0</td>
<td>BXM3</td>
<td>BXM2</td>
<td>BXM1</td>
<td>BXM0</td>
<td>RD</td>
</tr>
<tr>
<td>Enable</td>
<td>AF Underflow Counter</td>
<td>BC0 arrived Later counter</td>
<td>BC0 arrived Earlier counter</td>
<td>BX0/BXN[0] Mismatch counter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Here:
− X – Don’t care bit;
− AFEN – Enable reporting Alignment FIFO Underflow error to FMM
− AFU [2:0] – “Alignment FIFO Underflow” error counter;
− BCL [3:0] – “BC0 arrived Later” error counter;
− BCE [3:0] – “BC0 arrived Earlier” error counter;
− BXM [3:0] – “BX0 and BXN0 Mismatch” error counter;