Control/Status Register Group

 CSR_AF = 0x31 – Alignment FIFO Status

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

- **R**: [F1|F2|F3|F4|F5] / [M1|M2|M3]

Alignment FIFOs in the FA 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 in 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 FA AF runs at 80 MHz clock, so the AF latency in bunch crossings is twice less than the AFC value.

![Figure 1: Alignment FIFO Finite State Machine](image)

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 to the AFIFO starting from the third received frame or from the second bunch-crossing. The AFIFO resumes reads after the CSR_AFD delay.

<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>RXSD</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>R</td>
</tr>
</tbody>
</table>

**Table 1: CSR_AF Data Format for FA for 081015 and later versions**

Here:

- **AFFF** – Alignment FIFO Full Flag or AFC = 255; AFFF should be 0 on a successful TTC_RSYNC command;
- **AFEQ** – Alignment FIFO Empty Flag; AFEF should be 0 on a successful TTC_RSYNC command;
- **AFER** – Alignment FIFO Error Flag; AFER should be 0 on a successful TTC_RSYNC command;
- **RXSD** – Signal Detect Flag from FINISAR optical receiver; RXSD = 1 if optical fiber is connected to the SP05 input and MPC drives it, otherwise RXSD = 0;
- **AFIFO FSM one-hot states: Write Enable (AF_WREN), Frame 1 (AF_FRM1), Idle (AF_IDLE) and Init (AF_INIT). Power-up default AFIFO FSM state is AF_INIT.**
AFIFO FSM is in AF_WREN state on a successful TTC_RSYNC command. All other states indicate error conditions.

− AFC [7:0] = 0...255 – Alignment FIFO Read Word Count;

### Table 2: CSR_AF Data Format for FA for 080514 and earlier versions

<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>Flags</td>
<td>NU</td>
<td>AFIFO FSM States</td>
<td>Alignment FIFO Word Count</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>
<td></td>
</tr>
<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>R</td>
</tr>
</tbody>
</table>

Here:

− AFFF – Alignment FIFO Full Flag or AFC = 255;
  AFFF should be 0 on a successful TTC_RSYNC command;
− AFEF – Alignment FIFO Empty Flag;
  AFEF should be 0 on a successful TTC_RSYNC command;
− AFER – Alignment FIFO Error Flag;
  AFER should be 0 on a successful TTC_RSYNC command;
− AFIFO FSM one-hot states: Write Enable (AF_WREN), Frame 1 (AF_FRM1), Idle (AF_IDLE) and Init (AF_INIT). Power-up default AFIFO FSM state is AF_INIT.
  AFIFO FSM is in AF_WREN state on a successful TTC_RSYNC command. All other states indicate error conditions.
− AFC [7:0] = 0...255 – Alignment FIFO Read Word Count;

### Table 3: CSR_AF Data Format for FA for 080514 and earlier versions

<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>Flags</td>
<td>NU</td>
<td>AFIFO FSM States</td>
<td>Alignment FIFO Word Count</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>
<td></td>
</tr>
<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>R</td>
</tr>
</tbody>
</table>

Here:

− AFFF – Alignment FIFO Full Flag or AFC = 255;
  AFFF should be 0 on a successful TTC_RSYNC command;
− AFEF – Alignment FIFO Empty Flag;
  AFEF should be 0 on a successful TTC_RSYNC command;
− AFER – Alignment FIFO Error Flag;
  AFER should be 0 on a successful TTC_RSYNC command;
− AFIFO FSM one-hot states: Write Enable (AF_WREN), Frame 1 (AF_FRM1), Idle (AF_IDLE) and Init (AF_INIT). Power-up default AFIFO FSM state is AF_INIT.
  AFIFO FSM is in AF_WREN state on a successful TTC_RSYNC command. All other states indicate error conditions.
− AFC [7:0] = 0...255 – Alignment FIFO Read Word Count;

R: DD / MA

The DDU output link runs in its own clock domain, so there are two Alignment FIFOs in the DD: 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 system clock of 80.1574 MHz and the DDU link reference clock of 80.0000 MHz.
Table 4: CSR_AF Data Format for DD

<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>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>
<td>R</td>
</tr>
<tr>
<td>Flags</td>
<td>Alignment FIFO Read Word Count</td>
<td>Flags</td>
<td>Alignment FIFO Write Word Count</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>
<td></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; corresponds to AFWC = 31 or AFRC = 31;
- AFEF – Alignment FIFO Empty Flag.

R: SP / [M1|M2]

Alignment FIFOs in the SP perform the same function for DT muon data as they do in the FA for the CSC muon data. The difference is that to time-in the barrel data the user has to adjust the CSR_AFD value for each of the DT muons individually, refer to the CSR_AFD register for detail. The SP AF runs at system clock of 40.078 MHz, so the AF latency in bunch crossings equals to the AFC value.

Table 5: CSR_AF Data Format for SP for 090701 and later versions

<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>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>AFC3</td>
<td>AFC2</td>
<td>AFC1</td>
<td>AFC0</td>
</tr>
<tr>
<td>Flags</td>
<td>Not Used</td>
<td>Alignment FIFO Word Count</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>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Table 6: CSR_AF Data Format for SP for 090601 and earlier versions

<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>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>AFC3</td>
<td>AFC2</td>
<td>AFC1</td>
<td>AFC0</td>
</tr>
<tr>
<td>Flags</td>
<td>Not Used</td>
<td>Alignment FIFO Word Count</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>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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_OSY = 0x2B – Out-of-Synch Control / Status

RW: VM / MA

In the VM 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):

\[
\text{OSY0} = ( \text{OR} (\text{OSYN AND OSMn}) ) \text{ AND OSM0, } n = 1,\ldots,7
\]

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

\[\text{RW: [F1|F2|F3|F4|F5] / [M1|M2|M3]}\]
\[\text{W: [F1|F2|F3|F4|F5|FA] / MA}\]

In the FA, each MPC-to-SP05 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 comes 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 comes 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 BX read 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 FAs in total.
In the SP, each DT-to-SP05 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 comes 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 comes 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 BXN0 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 BX read 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 FAs in total.

<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>TRN</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>W</td>
<td></td>
<td></td>
<td></td>
<td></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>R</td>
</tr>
<tr>
<td>Enable</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>R</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;

**CSR_LQE = 0x27 – LCT Quality Enable**

This read/write register enables ME LCTs with qualities 15 to 0 to be passed to the SP05 core for track-finding. If a certain quality is disabled then the VP bit for an ME LCT with such quality gets reset, when forwarded to the SP. VP bit for the readout data path is not reset, so all original LCTs will nevertheless show up in the ME readout block, if they fit in the readout window.
Table 10: CSR_LQE Data Format for FA for 080514 and later versions

<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>Q15E</td>
<td>Q14E</td>
<td>Q13E</td>
<td>Q12E</td>
<td>Q11E</td>
<td>Q10E</td>
<td>Q9E</td>
<td>Q8E</td>
<td>Q7E</td>
<td>Q6E</td>
<td>Q5E</td>
<td>Q4E</td>
<td>Q3E</td>
<td>Q2E</td>
<td>Q1E</td>
<td>Q0E</td>
<td>W</td>
</tr>
<tr>
<td>Q15E</td>
<td>Q14E</td>
<td>Q13E</td>
<td>Q12E</td>
<td>Q11E</td>
<td>Q10E</td>
<td>Q9E</td>
<td>Q8E</td>
<td>Q7E</td>
<td>Q6E</td>
<td>Q5E</td>
<td>Q4E</td>
<td>Q3E</td>
<td>Q2E</td>
<td>Q1E</td>
<td>Q0E</td>
<td>R</td>
</tr>
</tbody>
</table>

LCT Quality Enable

Here:

- Q15E … Q0E = 0 / 1 (default) - disable / enable (default) ME LCT with quality 15 … 0.

RW: SP / MA:

This read/write register enables MB stubs with qualities 7 to 1 to be passed to the SP05 core for track-finding. If a certain quality is disabled then it gets reset to 0, when forwarded to the SP core. Original quality is retained for the readout path and shows up in the MB readout block, if MB stub fits in the readout window.

Table 11: CSR_LQE Data Format for SP for 090701 and later versions

<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>Q7DE</td>
<td>Q6DE</td>
<td>Q5DE</td>
<td>Q4DE</td>
<td>Q3DE</td>
<td>Q2DE</td>
<td>Q1DE</td>
<td>X</td>
<td>Q7AE</td>
<td>Q6AE</td>
<td>Q5AE</td>
<td>Q4AE</td>
<td>Q3AE</td>
<td>Q2AE</td>
<td>Q1AE</td>
<td>X</td>
<td>W</td>
</tr>
<tr>
<td>Q7DE</td>
<td>Q6DE</td>
<td>Q5DE</td>
<td>Q4DE</td>
<td>Q3DE</td>
<td>Q2DE</td>
<td>Q1DE</td>
<td>0</td>
<td>Q7AE</td>
<td>Q6AE</td>
<td>Q5AE</td>
<td>Q4AE</td>
<td>Q3AE</td>
<td>Q2AE</td>
<td>Q1AE</td>
<td>0</td>
<td>R</td>
</tr>
</tbody>
</table>

MB1D Quality Enable | NU | MB1A Quality Enable | NU

Here:

- Q7DE … Q1DE = 0 / 1 (default) - disable / enable (default) MB1D stub with quality 7 … 1;
- Q7AE … Q1AE = 0 / 1 (default) - disable / enable (default) MB1A stub with quality 7 … 1.