User's Guide

Determining Optimal Receive Buffer Delay in JESD204B and JESD204C Receivers

Srinivas Murthy, Ashwin Ravi

ABSTRACT

This user's guide explains the methods to align multiple JESD204 receiver lanes in AFE79xx using Receive Buffer Delay (RBD). In practice, the JESD204 receiver requires buffering of various delays in the received lanes in order to ensure deterministic data throughout the data bus. A key feature in AFE79xx can optimize the buffer pointer through internal lane-to-lane skew and arrival of lane time stamps with respect to the JESD204 clock. The AFE79xx JESD204 receiver block has unique features to read the skew and arrival of lanes with respect to Local Multi Frame Clock (LMFC)/ Local Extended Multiblock Clock (LEMC) clock to help find the right RBD for the system.

The user's guide is structured as follows:

- Understand the need for alignment across lanes.
- Basic review of JESD protocol to help understand functioning of RBD.
- Procedure to find the optimal RBD in AFE79xx with examples using C Application Process Interface (CAPI).
- Potential RBD alarms and alarm recovery plans.

Table of Contents

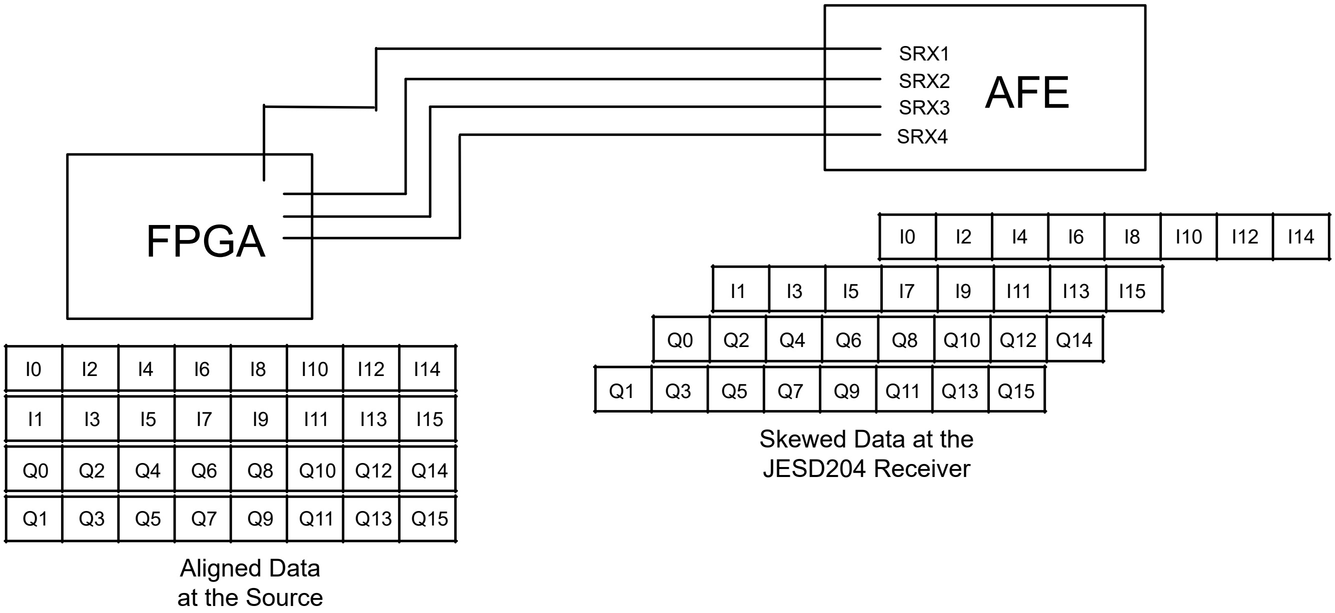
Trademarks

All trademarks are the property of their respective owners.

1 Introduction

The AFE79xx is a family of high-performance, wide bandwidth multi-channel transceivers, which includes four RF sampling transmitter chains. The transmitters support up to 1200-MHz wide bandwidth, which is designed for multi-band 4G and 5G base stations. Such high bandwidths demand up to 198 Gbps serial data transmission. For this reason, a maximum of 8 lanes are used to receive high bandwidth input for the DAC.

Each DAC takes a 16-bit I input and 16-bit Q input stream. For high bandwidth cases, the complex I and Q data reception can be spread among 4 input lanes. Figure 1-1 shows an example where the DAC JESD is configured with an LMFS[1] of 4222. Even with small form factor flip chip ball grid array (BGA) and symmetrical Serdes input and output ball placement, the PCB routing design cannot perfectly match all the SerDes I/O lanes. This causes a skew among lanes, which causes misalignment between the lanes. At the FPGA, data I0, I1, Q0 and Q1 are time aligned. As it traverses through the lanes and arrives at the AFE, the time alignment has been lost. If this is input to the DAC as it is, the spectrum observed at the DAC would be a lot different to what was expected. This example highlights the need to de-skew the receiver lanes in AFE. The role of RBD in JESD layer is exactly this.


SRX1
SRX2
SRX3
SRX4
AFE
FPGA
Skewed Data at the JESD204 Receiver
Aligned Data at the Source

I0	I2	I4	I6	I8	I10	I12	I14
I1	I3	I5	I7	I9	I11	I13	I15
Q0	Q2	Q4	Q6	Q8	Q10	Q12	Q14
Q1	Q3	Q5	Q7	Q9	Q11	Q13	Q15

					I0	I2	I4	I6	I8	I10	I12	I14
		I1	I3	I5	I7	I9	I11	I13	I15			
	Q0	Q2	Q4	Q6	Q8	Q10	Q12	Q14				
Q1	Q3	Q5	Q7	Q9	Q11	Q13	Q15					

Figure 1-1. Example of Skew among Lanes (DAC JESD LMFS = 42220)

Even in cases where a single lane carries data of each DAC, as in DAC JESD LMFS = 24410, it is critical to align the data on multiple lanes. Deterministic latency across power-up cycles necessitates this alignment across lanes. The path latency from FPGA to DAC is a critical metric and RBD helps with maintaining this deterministic latency across transmitter channels and across power-up cycles.

[1] Standard JESD204 Definition for JESD204 Lane and Data Packing Configuration.

L = Number of Lanes

M = Number of Converters

F = Number of Octets per Frame

S = Number of Samples per Frame

2 JESD204C

2.1 Basic Review of JESD204C Protocol

This section provides a very brief review of JESD204C protocol, so as to understand terminologies related to setting the optimal RBD.

JESD204C implements 64b/66b encoding. To each set of eight octets (64 bits), two pilot bits called sync header are inserted. The 2 bits in sync header are invert of each other, which means the sync header can only be 10 or 01. With this unique property of sync header, the JESD receiver identifies and locks on to the 66-bit boundary. These 66 bits are termed as blocks.

The blocks are then built into a multiblock, that consists of 32 blocks, as shown in Figure 2-1 [2]. The sync header SH0 to SH31 follow a pattern as described in the protocol, which helps the JESD receiver lock onto the multiblock boundary.

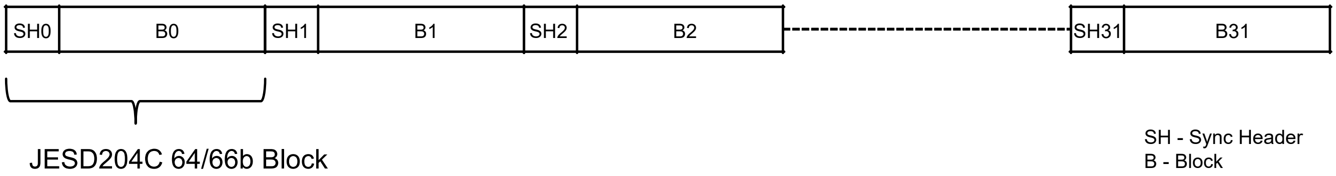


Figure 2-1. JESD204C Multiblock Structure

Further, ‘E’ number of multiblocks are combined into a extended multiblock. The parameter ‘E’ is configurable in Latte -

sysParams.jesdRxK

Typically, for 16-bit data packing (that is, typical F = 1, 2, 4, or 8 cases), E is set to 1. For 12-bit data packing (that is, F = 3 or F = 6 cases) patterns E is set to 3, so that each extended multiblock contains an integer number of samples and integer number of frames. The information stored in sync header, specifically SH22, is used to identify the end of an extended multiblock.

The JESD receiver uses a LEMC to correct for the skew between lanes. The LEMC period is equal to the extended multi-block period. For example,

Lane Rate = 24.33024 Gbps

LEMC clock frequency = 24.33024/66/32/E GHz

For E = 1, LEMC clock frequency can be calculated as 11.52 MHz

To ensure that the processing clock LEMC, between the JESD204 transmitter and JESD204 receiver, are aligned at start-up of the system without drift or wander, a global system reference clock (SYSREF) provides the clock synchronization and alignment. The SYSREF frequency is an integer factor of LEMC frequency. The SYSREF is distributed throughout the JESD204 system in a time aligned, fixed delay manner throughout various temperature cycle and system restart cycle. Since SYSREF is essentially deterministic, the data transfer through the JESD204 link will also be deterministic.

[2] Parts of figures were based on JEDEC JESD204C standard, Figure 5 and Figure 50. Copyright JEDEC. Modifications have not been approved by and do not reflect the views of JEDEC.

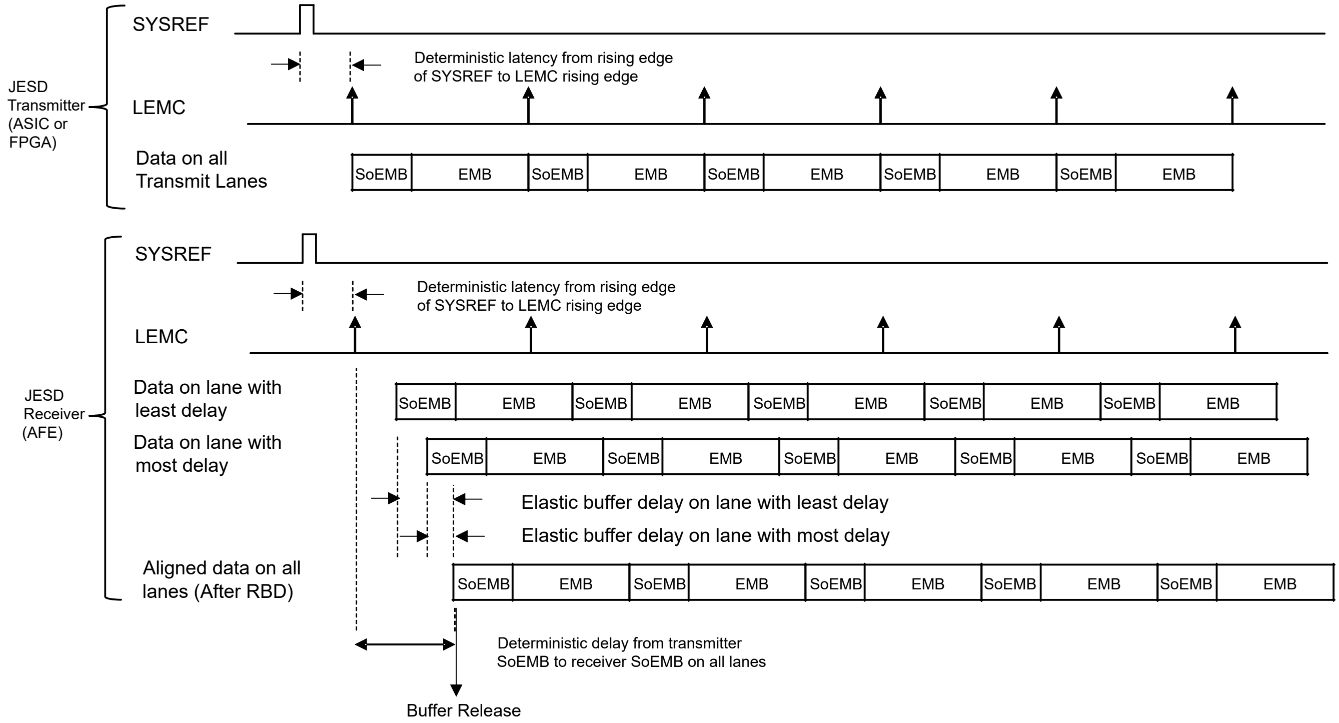

JESD Transmitter (ASIC or FPGA)
SYSREF
Deterministic latency from rising edge of SYSREF to LEMC rising edge
LEMC
Data on all Transmit Lanes
SoEMB
EMB
JESD Receiver (AFE)
SYSREF
Deterministic latency from rising edge of SYSREF to LEMC rising edge
LEMC
Data on lane with least delay
Data on lane with most delay
Elastic buffer delay on lane with least delay
Elastic buffer delay on lane with most delay
Aligned data on all lanes (After RBD)
Deterministic delay from transmitter SoEMB to receiver SoEMB on all lanes
Buffer Release

Figure 2-2. Timing Diagram of JESD204C Depicting RBD Functionality

To compensate for the lane-to-lane skew, the JESD204C receiver has an internal buffer to first absorb the skews amongst all the lanes, and then re-align the lanes at the output of the buffer upon the release of the buffer. This essentially created a zero-skew environment for data processing at the output the JESD204C receiver. This is highlighted in Figure 2-2 [3].

The buffer and the release of the buffer is controlled by RBD, or receive buffer delay. Finding the optimal RBD value in a system that will work across various temperature and restart cycle is essential in the overall system stability.

In the JESD204C transmitter, the Start of Extended Multi Block (SoEMB as shown in Figure 2-2) shall be initiated simultaneously across all lanes at a well-defined moment in time. The 'well-defined moment in time' is a deterministic period of time from the LEMC edge.

In the JESD204C receiver, to align data across lanes, a buffer exists to hold all lane data for release simultaneously at a well-defined moment in time. The 'well-defined moment in time' for RX buffer release is a programmable number of steps after an active LEMC edge. This programmable number of steps is referred to as the Receive Buffer Delay (RBD).

[3] Parts of figures were based on JEDEC JESD204C standard, Figure 5 and Figure 50. Copyright JEDEC. Modifications have not been approved by and do not reflect the views of JEDEC.

2.2 Finding Optimal RBD for JESD204C

This section describes the procedure to be followed for setting the optimal RBD. Figure 2-3 [4] depicts the available features in AFE79xx to readout various latencies, including lane arrival delay and skew among lanes.

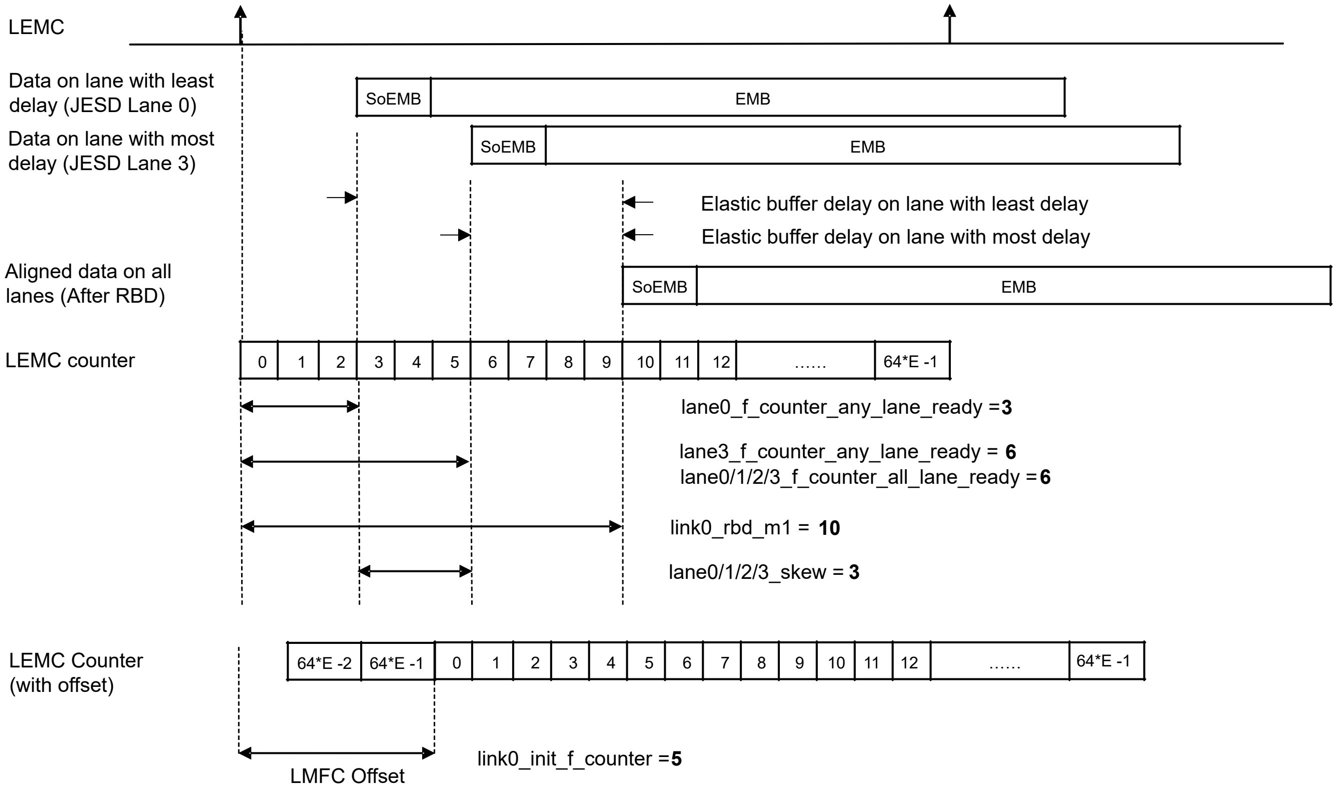

LEMC
Data on lane with least delay (JESD Lane 0)
SoEMB
EMB
Data on lane with most delay (JESD Lane 3)
SoEMB
EMB
Elastic buffer delay on lane with least delay
Elastic buffer delay on lane with most delay
Aligned data on all lanes (After RBD)
SoEMB
EMB
LEMC counter
0
1
2
3
4
5
6
7
8
9
10
11
12
......
64*E -1
lane0_f_counter_any_lane_ready =3
lane3_f_counter_any_lane_ready = 6
lane0/1/2/3_f_counter_all_lane_ready =6
link0_rbd_m1 = 10
lane0/1/2/3_skew = 3
LEMC Counter (with offset)
64*E -2
64*E -1
0
1
2
3
4
5
6
7
8
9
10
11
12
......
64*E -1
link0_init_f_counter =5
LMFC Offset

Figure 2-3. Visual Description of Registers Related to RBD in JESD204C

Internal to AFE79xx, there exists a LEMC counter which operates on a clock with frequency of LaneRate/33. The counter is periodic with the LEMC, whose frequency is LaneRate/(66*32*E). The LEMC counter counts 64 times faster than the internal LEMC frequency, and therefore, the LEMC counter will count from 0 to 64*E-1, as shown in Figure 2-3. There is also an option to add an offset to this counter by use of *link0/1_init_f_counter*. In Latte, this can be done by setting the system parameter *jesdRxInitLmfcCounter*. For example, as per Figure 2-3,

sysParams.jesdRxInitLmfcCounter = 5

can be added to the Latte bringup. For more details, see Section 5.2.

Lane arrival information can be read back from *lane0/1/2/3_f_counter_any_lane_ready*. *lane0/1/2/3_f_counter_any_lane_ready* is the value of LEMC counter when the start of EMB arrives at the corresponding lane. Figure 2-3 shows such an example for JESD lane 0 and JESD lane 3. The start of EMB arrives at JESD lane 0 when the LEMC counter is 3, this information can be read back from *lane0_f_counter_any_lane_ready*. Similarly, *lane3_f_counter_any_lane_ready* would read as 6.

The instant last lane arrives is critical information for setting the RBD. This information can be read back from *lane0/1/2/3_f_counter_all_lane_ready*. *lane0/1/2/3_f_counter_all_lane_ready* would be equal to the *lane_f_counter_any_lane_ready* of slowest lane. In the case shown in Figure 2-3, the last lane to arrive is JESD lane 3. So, *lane0/1/2/3_f_counter_all_lane_ready* would read the same as *lane3_f_counter_any_lane_ready*, which is 6.

lane0/1/2/3_skew equals the difference between the LEMC counter values for earliest arrival lane and latest arrival lane. In Figure 2-3, it is difference between *lane3_f_counter_any_lane_ready* and *lane0_f_counter_any_lane_ready*. So *lane0/1/2/3_skew* would read 3.

[4] Parts of figures were based on JEDEC JESD204C standard, Figure 5 and Figure 50. Copyright JEDEC. Modifications have not been approved by and do not reflect the views of JEDEC.

With such visibility to lane arrival information in AFE79xx, it becomes easy to estimate the optimal RBD value. The recommended calculation for RBD is as follows.

RBD = (lane0/1/2/3_f_counter_all_lane_ready + 4)%(64*E)

The range of valid values the RBD can take is described by below condition

lane0/1/2/3_f_counter_all_lane_ready < RBD

and

RBD < min(lane0/1/2/3_f_counter_any_lane_ready) + BufferDepth - 10

Default value of buffer depth is 32. Buffer depth is configurable – value of (*link0/1_buffer_depth + 1*) determines the buffer depth. The register *link0_buffer_depth* can take values from 0 to 31, meaning buffer depth can vary from 1 to 32.

link0_rbd_m1 is written to with value of the value of RBD. Register map and procedure to be followed in system bringup is described in detail in Section 4. Below examples describe the basic RBD calculation.

From the latest arrival lane, a margin of 4 is accounted for to include temperature and process variations. Once the margin is added, modulus operation is performed with 64*E. This is because the LEMC counter counts from 0 to a maximum of 64*E-1. The buffer releases the aligned data on all lanes once the LEMC counter reaches the RBD value. In the example shown in Figure 2-3, the RBD is set to 10, which is 4 LEMC counter value ahead of latest arrival lane. Few more examples are shown below. The value of skew is assumed to be 3 in the examples.

Note

Below calculations are just examples specific to 204C. See Section 4 for detailed procedure to set RBD appropriately.

Example 1

E = 1

lane0/1/2/3_f_counter_all_lane_ready = 62

RBD = (62+4)%64 = 2

With a skew of 3, lane0/1/2/3_f_counter_any_lane_ready is expected to read values in range [59,62]

Example 2

E = 2

lane0/1/2/3_f_counter_all_lane_ready = 62

RBD = (62 + 4)%(64*3) = 66

With a skew of 3, lane0/1/2/3_f_counter_any_lane_ready is expected to read values in range [59,62]

Example 3

E = 3

lane0/1/2/3_f_counter_all_lane_ready = 189

RBD = (189+4)%(64*3) = 1

With a skew of 3, lane0/1/2/3_f_counter_any_lane_ready is expected to read values in range [186,189]

3 JESD204B

3.1 Basic Review of JESD204B Protocol

This section provides a very brief review of JESD204B protocol, so as to understand terminologies related to setting the optimal RBD.

JESD204B implements 8b/10b encoding, where 8 bits of data are encoded into 10 bits of data to ensure DC balanced signal. A frame in JESD204B consists of ‘F’ number of octets. In Latte, F is specified as part of *LMFSHd* parameter definition for DAC as follows.

sysParams.LMFSHdTx

Further, ‘K’ number of frames are combined into a multiframe. The parameter ‘K’ is configurable in Latte - *sysParams.jesdRxK*. Typically, a value of 32 or 16 is used for K. In each multiframe, there will be F*K octets.

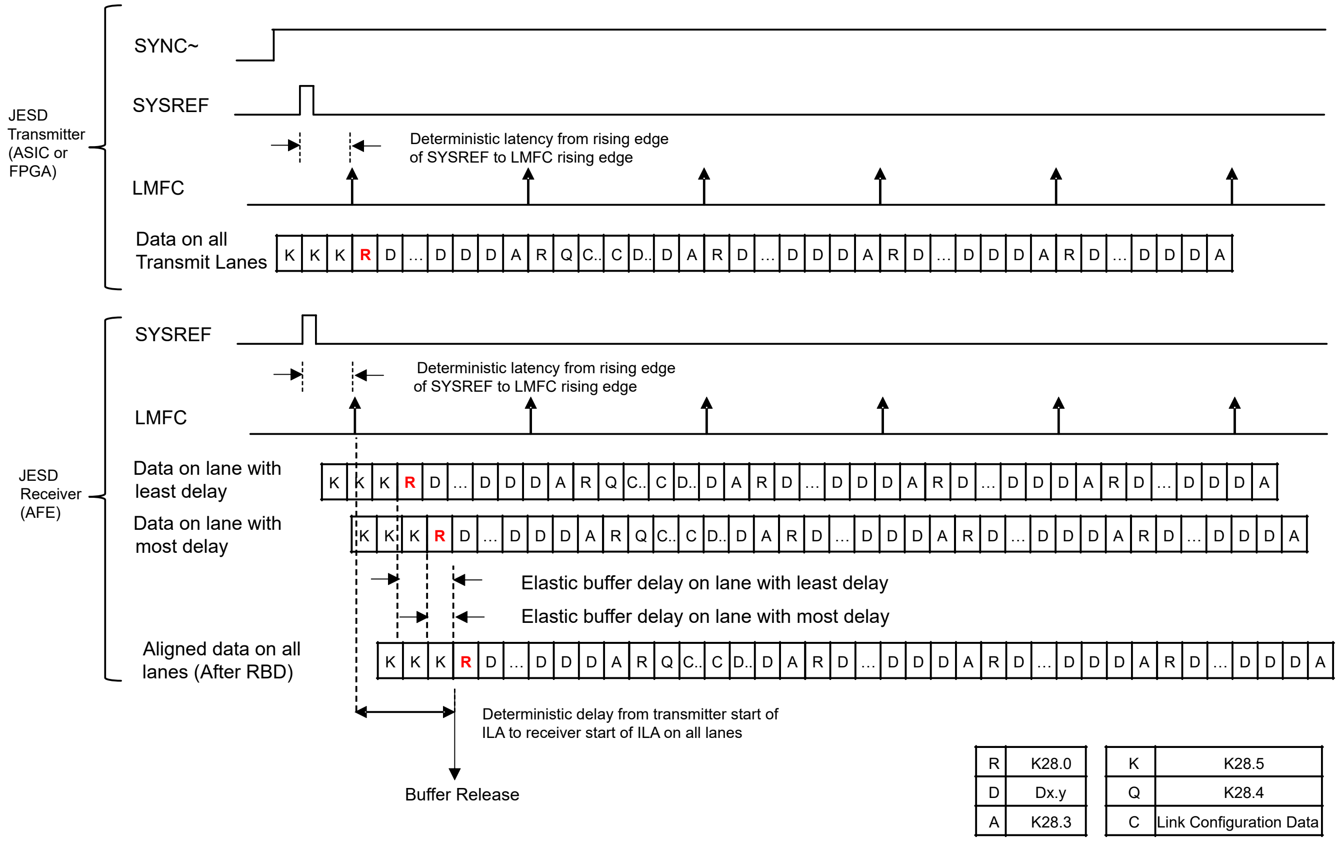
The JESD receiver uses a LMFC to correct for the skew between lanes. The LMFC period is equal to the multiframe period. For example,

Lane Rate = 9830.4 Gbps

LMFC clock frequency = 9830.4/10/F/K GHz

For F = 4 and K = 32, LMFC clock frequency can be calculated as 7.68 MHz.

To ensure that the processing clock LMFC, between the JESD204 transmitter and JESD204 receiver, are aligned at start-up of the system without drift or wander, a global system reference clock (SYSREF) provides the clock synchronization and alignment. The SYSREF frequency is an integer factor of LMFC frequency. The SYSREF is distributed throughout the JESD204 system in a time aligned, fixed delay manner throughout various temperature cycle and system restart cycle. Since SYSREF is essentially deterministic, the data transfer through the JESD204 link will also be deterministic.


JESD Transmitter (ASIC or FPGA)
SYNC~
SYSREF
Deterministic latency from rising edge of SYSREF to LMFC rising edge
LMFC
Data on all Transmit Lanes
K K K R D ... D D D A R Q C.. C D.. D A R D ... D D D A R D ... D D D A R D ... D D D A
JESD Receiver (AFE)
SYSREF
Deterministic latency from rising edge of SYSREF to LMFC rising edge
LMFC
Data on lane with least delay
K K K R D ... D D D A R Q C.. C D.. D A R D ... D D D A R D ... D D D A R D ... D D D A
Data on lane with most delay
K K K R D ... D D D A R Q C.. C D.. D A R D ... D D D A R D ... D D D A R D ... D D D A
Elastic buffer delay on lane with least delay
Elastic buffer delay on lane with most delay
Aligned data on all lanes (After RBD)
K K K R D ... D D D A R Q C.. C D.. D A R D ... D D D A R D ... D D D A R D ... D D D A
Deterministic delay from transmitter start of ILA to receiver start of ILA on all lanes
Buffer Release

R	K28.0	K	K28.5
D	Dx.y	Q	K28.4
A	K28.3	C	Link Configuration Data

Figure 3-1. Timing Diagram of JESD204B Depicting RBD Functionality

To compensate for the lane-to-lane skew, the JESD204B receiver has an internal buffer to first absorb the skews amongst all the lanes, and then re-align the lanes at the output of the buffer upon the release of the buffer. This essentially created a zero-skew environment for data processing at the output the JESD204B receiver. This is highlighted in Figure 3-1.[5]

The buffer and the release of the buffer is controlled by RBD, or receive buffer delay. Finding the optimal RBD value in a system that will work across various temperature and restart cycle is essential in the overall system stability.

There exists a SYNC signal from the JESD receiver to JESD transmitter, which is used to request synchronization characters such as K28.5 characters and indicate signal detection of the synchronization signal. Once the receiver detects consecutive K28.5 characters, the receiver de-asserts the SYNC~ signal. Once the transmitter detects a toggle in the SYNC pin, it moves to the Initial Lane Alignment (ILA) phase.

In the JESD204B transmitter, the start of ILA (shown in Figure 3-1) shall be initiated simultaneously across all lanes at a well-defined moment in time. The transition from /K/ character to /R/ character marks the start of ILA (highlighted in red in Figure 3-1). The ‘well-defined moment in time’ is a deterministic period of time from the LMFC edge.

In the JESD204B receiver, to align data across lanes, a buffer exists to hold all lane data for release simultaneously at a well-defined moment in time. The ‘well-defined moment in time’ for RX buffer release is a programmable number of steps after an active LMFC edge. This programmable number of steps is referred to as the RX Buffer Delay (RBD).

3.2 Finding Optimal RBD for JESD204B

This section describes the procedure to be followed for setting the optimal RBD. Figure 3-2 [6] depicts the available features in AFE79xx to readout various latencies, including lane arrival delay and skew among lanes.

[5] Parts of figures were based on JEDEC JESD204C standard, Figure 5 and Figure 50. Copyright JEDEC. Modifications have not been approved by and do not reflect the views of JEDEC.

[6] Parts of figures were based on JEDEC JESD204C standard, Figure 5 and Figure 50. Copyright JEDEC. Modifications have not been approved by and do not reflect the views of JEDEC.

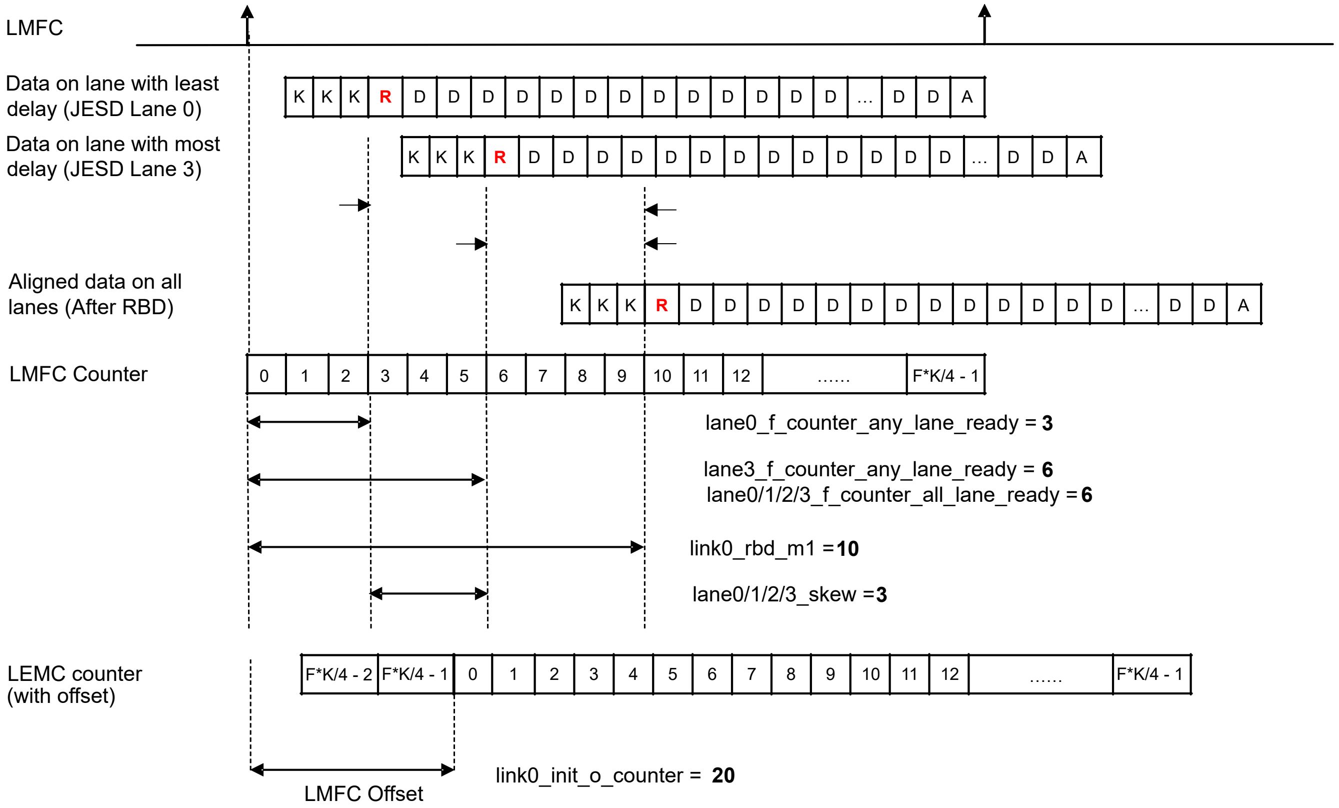

LMFC
Data on lane with least delay (JESD Lane 0)
K K K R D D D D D D D D D D D D D ... D D A
Data on lane with most delay (JESD Lane 3)
K K K R D D D D D D D D D D D D D ... D D A
Aligned data on all lanes (After RBD)
K K K R D D D D D D D D D D D D D ... D D A
LMFC Counter
0 1 2 3 4 5 6 7 8 9 10 11 12 ...... F*K/4 - 1
lane0_f_counter_any_lane_ready = 3
lane3_f_counter_any_lane_ready = 6
lane0/1/2/3_f_counter_all_lane_ready = 6
link0_rbd_m1 = 10
lane0/1/2/3_skew = 3
LEMC counter (with offset)
F*K/4 - 2 F*K/4 - 1 0 1 2 3 4 5 6 7 8 9 10 11 12 ...... F*K/4 - 1
LMFC Offset
link0_init_o_counter = 20

Figure 3-2. Visual Description of Registers Related to RBD in JESD204B

Internal to AFE79xx, there exists a LMFC counter which operates on a clock with frequency of LaneRate/40. The counter is periodic with the LMFC, whose frequency is LaneRate/(10*F*K). This means, the LMFC counter will count from 0 to F*K/4-1, as shown in Figure 3-2. There is also an option to add an offset to this counter by use of *link0/1_init_o_counter*. In Latte, this can be done by setting the system parameter *jesdRxInitLmfcCounter*. For example, as per Figure 3-2,

sysParams.jesdRxInitLmfcCounter = 5

can be added to the Latte bringup. See Section 5.3 for more details.

Lane arrival information can be read back from *lane0/1/2/3_f_counter_any_lane_ready*. *lane0/1/2/3_f_counter_any_lane_ready* is the value of LMFC counter when the transition from K28.5 to ILA occurs at the corresponding lane. Figure 3-2 shows such an example for JESD lane 0 and JESD lane 3. At JESD lane 0, ILA phase starts when the LMFC counter is 3, this information can be read back from lane0_f_counter_any_lane_ready. Similarly, *lane3_f_counter_any_lane_ready* would read as 6.

The instant last lane arrives is critical information for setting the RBD. This information can be read back from *lane0/1/2/3_f_counter_all_lane_ready*. *lane0/1/2/3_f_counter_all_lane_ready* would be equal to the *lane_f_counter_any_lane_ready* of slowest lane. In the case shown in Figure 3-2, the last lane to arrive is JESD lane 3. So, *lane0/1/2/3_f_counter_all_lane_ready* would read the same as *lane3_f_counter_any_lane_ready*, which is 6.

lane0/1/2/3_skew equals the difference between the LMFC counter values for earliest arrival lane and latest arrival lane. In Figure 3-2, it is difference between *lane3_f_counter_any_lane_ready* and *lane0_f_counter_any_lane_ready*. So *lane0/1/2/3_skew* would read 3.

With such visibility to lane arrival information in AFE79xx, it becomes easy to estimate the optimal RBD value. The recommended calculation for RBD is as follows:

RBD = (lane0/1/2/3_f_counter_all_lane_ready + 4) % (F*K/4)

The range of valid values the RBD can take is described by below condition:

lane0/1/2/3_f_counter_all_lane_ready < RBD

and

RBD < min(lane0/1/2/3_f_counter_any_lane_ready) + BufferDepth - 10

Default value of buffer depth is 32. Buffer depth is configurable – value of (*link0/1_buffer_depth* + *1*) determines the buffer depth. The register *link0_buffer_depth* can take values from 0 to 31, meaning buffer depth can vary from 1 to 32.

link0_rbd_m1 is written to with value of the RBD. Register map and procedure to be followed in system bringup is described in detail in Section 4. Below examples describe the basic RBD calculation.

From the latest arrival lane, a margin of 4 is accounted for to include temperature and process variations. Once the margin is added, modulus operation is performed with F*K/4. This is because the LMFC counter counts from 0 to a maximum of F*K/4-1. The buffer releases the aligned data on all lanes once the LMFC counter reaches the RBD value. In the example shown in Figure 3-2, the RBD is set to 10, which is 4 LMFC counter value ahead of latest arrival lane. Few more examples are shown below. The value of skew is assumed to be 3 in the examples.

Note

Below calculations are just examples specific to 204B. See Section 4 for detailed procedure to set RBD appropriately.

Example 1

F = 4, K = 32

lane0/1/2/3_f_counter_all_lane_ready = 30

RBD = (30 + 4) % 32 = 2

With a skew of 3, lane0/1/2/3_f_counter_any_lane_ready is expected to read values in range [27,30]

Example 2

F = 6, K = 32

lane0/1/2/3_f_counter_all_lane_ready = 30 (1)

RBD = (30 + 4) % 48 = 34 (2)

With a skew of 3, lane0/1/2/3_f_counter_any_lane_ready is expected to read values in range [27,30] (3)

Example 3

F = 6, K = 32

lane0/1/2/3_f_counter_all_lane_ready = 47

RBD = (47 + 4) % 48 = 3

With a skew of 3, lane0/1/2/3_f_counter_any_lane_ready is expected to read values in range [44,47]

4 Setting RBD in AFE79xx

4.1 Register Map

The addresses of all registers referred to in this user's guide are described in Table 4-1. There exists two pages of registers for DAC JESD block. The page select to choose appropriate page is ‘dac_jesd’ (0x16[3:2]).

Table 4-1. Register Map

Register Name	Read/Write	Address
dac_jesd	R/W	0x16[3:2]
lane0_f_counter_any_lane_ready	R	0x89[7:0], 0x88[7:0]
lane1_f_counter_any_lane_ready	R	0x8B[7:0], 0x8A[7:0]
lane2_f_counter_any_lane_ready	R	0x8D[7:0], 0x8C[7:0]
lane3_f_counter_any_lane_ready	R	0x8F[7:0], 0x8E[7:0]
lane0_f_counter_all_lane_ready	R	0x91[7:0], 0x90[7:0]
lane1_f_counter_all_lane_ready	R	0x93[7:0], 0x92[7:0]
lane2_f_counter_all_lane_ready	R	0x95[7:0], 0x94[7:0]
lane3_f_counter_all_lane_ready	R	0x97[7:0], 0x96[7:0]
lane0_skew	R	0x13C[4:0]
lane1_skew	R	0x13D[4:0]
lane2_skew	R	0x13E[4:0]
lane3_skew	R	0x13F[4:0]
link0_rbd_m1	R/W	0x69[7:0], 0x68[7:0]
link1_rbd_m1	R/W	0x6B[7:0], 0x6A[7:0]
link0_init_f_counter	R/W	0x71[7:0]
link1_init_f_counter	R/W	0x73[7:0]
link0_init_o_counter	R/W	0x70[7:0]
link1_init_o_counter	R/W	0x72[7:0]
alarms	R	0x11F[8:0], 0x11E[8:0], 0x11D[8:0], 0x11C[8:0], 0x11B[8:0], 0x11A[8:0], 0x119[8:0], 0x118[8:0],
link0_sysref_cnt_on_release_opportunity	R	0x98[3:0]
link1_sysref_cnt_on_release_opportunity	R	0x98[7:4]
link0_buffer_depth	R/W	0x6E[3:0]
link1_buffer_depth	R/W	0x6F[3:0]

4.2 Setting RBD in Configuration Sequence

This section describes the overall procedure to be followed while setting RBD. See Section 4.3, for the detailed procedure with CAPI calls that can be used.

1. Firstly, the configuration that was generated from Latte is run on the PCB, for which optimal RBD needs to be found.
2. Ensure that the CS state is achieved, this ensures that the start of EMB boundary is locked onto for JESD204C and ILA boundary is locked onto for JESD204B. Only after this, the *fcounter* registers would contain the intended values. Ignore failures related to BUF state at this stage.
3. Read back the *lane0/1/2/3_f_counter_all_lane_ready* registers and calculate the optimal RBD based on Section 2.2 for JESD204C and Section 3.2 for JESD204B. In case there are multiple AFE ICs on the same board, ensure to read the *lane0/1/2/3_f_counter_all_lane_ready* registers on all AFEs. The highest out of all AFEs is then used for RBD calculation. If the values are significantly different, different RBD values might have to be used for each location of AFE on the PCB. See Section 4.3 for detailed procedure to find and set the optimal RBD value.
4. Optimal value found from previous step is set and *adcDacSync* is executed. Ensure that the SYSREF is running when *adcDacSync* is executed. The command *adcDacSync* resynchronizes the JESD link. Running this command applies the RBD value that was set in Step 3.

This step is the second iteration in identifying the right RBD value.

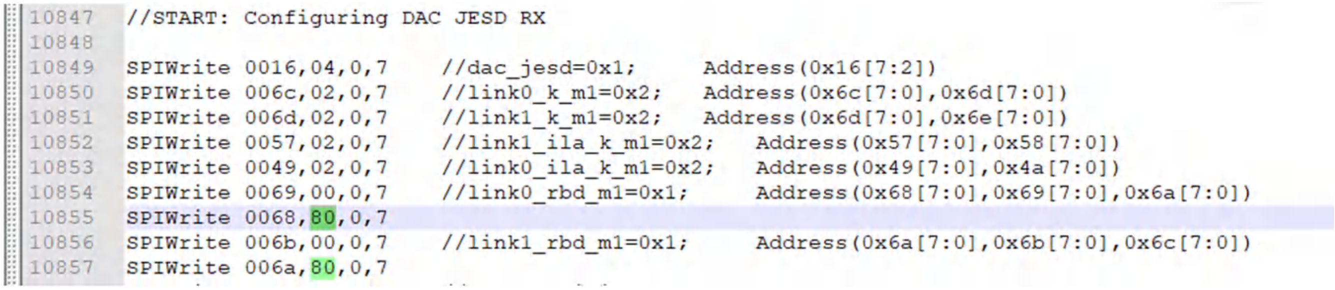
link0/1_sysref_cnt_on_release_opportunity is read back.

For 204C, *(RBD_Step3+ link0/1_sysref_cnt_on_release_opportunity)%(64*E)* is chosen as the optimal RBD.

For 204B, *(RBD_Step3+ link0/1_sysref_cnt_on_release_opportunity)%(F*K/4)* is chosen as the optimal RBD.

RBD_Step3 is the RBD calculated from Step 3.

5. The configuration is then modified with the optimal RBD as shown in Figure 4-1. In this case the RBD has been set to a value of 0x80.
6. The modified configuration sequence can now be frozen as the final configuration with optimal RBD settings.



```
//START: Configuring DAC JESD RX

SPIWrite 0016,04,0,7    //dac_jesd=0x1;     Address(0x16[7:2])
SPIWrite 006c,02,0,7    //link0_k_m1=0x2;   Address(0x6c[7:0],0x6d[7:0])
SPIWrite 006d,02,0,7    //link1_k_m1=0x2;   Address(0x6d[7:0],0x6e[7:0])
SPIWrite 0057,02,0,7    //link1_ila_k_m1=0x2;   Address(0x57[7:0],0x58[7:0])
SPIWrite 0049,02,0,7    //link0_ila_k_m1=0x2;   Address(0x49[7:0],0x4a[7:0])
SPIWrite 0069,00,0,7    //link0_rbd_m1=0x1;     Address(0x68[7:0],0x69[7:0],0x6a[7:0])
SPIWrite 0068,80,0,7
SPIWrite 006b,00,0,7    //link1_rbd_m1=0x1;     Address(0x6a[7:0],0x6b[7:0],0x6c[7:0])
SPIWrite 006a,80,0,7
```

Figure 4-1. Changing RBD in Configuration Sequence

To summarize, initially the configuration is run once to identify the fcounter read backs and calculate the optimal RBD. Once optimal RBD is found, it is updated in the configuration sequence and used.

4.3 Finding Optimal RBD using CAPI

The previous discussions explained the process of setting optimal RBD from a top level view. This section explains the changes to procedure in case of various cases with the help of CAPIs.

In Section 4.3.1 through Section 4.3.4, below terminology is used.

Top JESD – This refers to the first 4 lanes in JESD. The page select for this is 0x16[2]. 0x16[2] is termed as the first page.

Bottom JESD – This refers to the last 4 lanes in JESD. The page select for this is 0x16[3]. 0x16[3] is termed as the second page.

All examples in this section use 204C with E = 1 and skew value of 3. For other values of E or for JESD204B, the modulus operation has to be performed with the right numbers as described in examples from Section 2.2 and Section 3.2. The text in italics correspond to CAPI calls.

4.3.1 Use Case with 1 JESD Links

In this case, lanes in top JESD and bottom JESD are part of a single JESD link. In this case, based on *lane0_f_counter_all_lane_ready* in the first page, link0_rbd_m1 is set in the first page.

Example:

1. Read Page 0x16[2] : lane0_f_counter_all_lane_ready = 16

   getAllLaneReady(afeInst,0,&laneAllReady)

   laneAllReady = 16
2. Obtain valid range of RBD.

   getGoodRbdRange(afeInst,0,&rbdMin,&rbdMax)

   rbdMin = 17

   rbdMax = 35 (Based on skew = 3, rbdMax = min (allLaneReady) + BufferDepth -10 = 16 - 3 + 32 - 10).
3. In this example, optimal value is chosen as rbdMin + 3 = 20. In cases with multiple AFEs on a PCB, valid ranges can be looked at for all AFEs and optimal RBD can be chosen.

4. The optimal RBD is set and adcDacSync is run.

   setManualRbd(afeInst,0,20)

   adcDacSync(afeInst,1)
5. Read back link0_sysref_cnt_on_release_opportunity from the first page.

   getSysrefCntOnReleaseOpp(afeInst,0,&readVal)

   If readVal=1, optimal RBD is chosen as (20+1)%64 = **21**
6. In the config, value of 21 is set for the link0_rbd_m1 in the first page. In Latte, *sysParams.jesdRxRbd* is set to [21,21,21,21]

4.3.2 Use Case with 2 JESD Links

When there are two links, top JESD uses one link and bottom JESD uses another link. For the first link, the registers in page 0x16[2] are used and second link uses page 0x16[3].

1. For the first link, based on the *lane0_f_counter_all_lane_ready* in the first page, *link0_rbd_m1* is set in the first page.
2. Similarly, for the second link, based on the *lane0_f_counter_all_lane_ready* in the second page, *link0_rbd_m1* is set in the second page.

Example:

1. **Link 1**Read Page 0x16[2] : lane0_f_counter_all_lane_ready = 16 for link 1

   getAllLaneReady(afeInst,0,&laneAllReady)

   laneAllReady = 16 for link 1

   Link 2 Read Page 0x16[3] : lane0_f_counter_all_lane_ready = 36 for link 2

   getAllLaneReady(afeInst,2,&laneAllReady)

   laneAllReady = 36 for link 2
2. Obtain valid range of RBD.

   Link 1 *getGoodRbdRange(afeInst,0,&rbdMin,&rbdMax)*

   rbdMin = 17

   rbdMax = 35 (Based on skew = 3, rbdMax = min(allLaneReady) + BufferDepth-10 = 16 - 3 + 32 - 10)

   Link 2 *getGoodRbdRange(afeInst,2,&rbdMin,&rbdMax)*

   rbdMin = 37

   rbdMax = 55 (Based on skew = 3, rbdMax = min(allLaneReady) + BufferDepth - 10 = 36 - 3 + 32 - 10)
3. In this example, optimal value is chosen as rbdMin + 3 = 20 for link1 and 40 for link 2. In cases with multiple AFEs on a PCB, valid ranges can be looked at for all AFEs and optimal RBD can be chosen.
4. The optimal RBD is set and adcDacSync is run.

   Link 1

   setManualRbd(afeInst,0,20)

   Link 2

   setManualRbd(afeInst,2,40)

   adcDacSync(afeInst,1)

5. Read back link0_sysref_cnt_on_release_opportunity from the first page for link 1 and second page for link 2.

   Link 1 *getSysrefCntOnReleaseOpp(afeInst,0,&readVal)*

   If readVal=1, optimal RBD is chosen as (20+1)%64 = **21**

   Link 2 *getSysrefCntOnReleaseOpp(afeInst,2,&readVal)*

   If readVal=1, optimal RBD is chosen as (40+1)%64 = **41**

6. In the config, value of 21 is set for the link0_rbd_m1 in the first page and value for 41 is set for the link0_rbd_m1 in the second page.

   In Latte, *sysParams.jesdRxRbd* is set to [21,21,41,41]

4.3.3 Use Case with 3 JESD Links

In use-cases where there are 3 JESD links, there are two possibilities.

1. Top JESD has two links, bottom JESD has one link.
2. Top JESD has one link, bottom JESD has one link.

We discuss only the first case here, but the same procedure holds for the second case if the pages are switched around. For the CAPIs, for first case, the linkNo argument would be 0, 1, 2 for links 1, 2 and 3 respectively. For the second case, the linkNo argument would be 0, 2, 3 for links 1, 2 and 3 respectively.

1. For the first link, based on the *lane0_f_counter_all_lane_ready* in the first page, *link0_rbd_m1* is set in the first page.
2. For the second link, based on the *lane2_f_counter_all_lane_ready* in the first page, *link1_rbd_m1* is set in the first page.
3. For the third link, based on the *lane0_f_counter_all_lane_ready* in the second page, *link0_rbd_m1* is set in the second page.

Example:

1. **Link 1**

   Read Page 0x16[2] : lane0_f_counter_all_lane_ready = 16 for link 1

   getAllLaneReady(afeInst,0,&laneAllReady)

   laneAllReady = 16 for link 1

   Link 2

   Read Page 0x16[2] : lane2_f_counter_all_lane_ready = 26 for link 2

   getAllLaneReady(afeInst,1,&laneAllReady)

   laneAllReady = 26 for link 2

   Link 3

   Read Page 0x16[3] : lane0_f_counter_all_lane_ready = 36 for link 3

   getAllLaneReady(afeInst,2,&laneAllReady)

   laneAllReady = 36 for link 3

2. Obtain valid range of RBD.

   Link 1

   getGoodRbdRange(afeInst,0,&rbdMin,&rbdMax)

rbdMin = 17

rbdMax = 35 (Based on skew=3, rbdMax = min(allLaneReady) + BufferDepth - 10 = 16 - 3 + 32 - 10)

Link 2

getGoodRbdRange(afeInst,1,&rbdMin,&rbdMax)

rbdMin = 27

rbdMax =45 (Based on skew=3, rbdMax = min(allLaneReady) + BufferDepth - 10 = 26 - 3 + 32 - 10)

Link 3

getGoodRbdRange(afeInst,2,&rbdMin,&rbdMax)

rbdMin = 37

rbdMax = 55 (Based on skew=3, rbdMax = min(allLaneReady) + BufferDepth - 10 = 36 - 3 + 32 - 10)

3. In this example, optimal value is chosen as rbdMin+3 = 20 for link1, 30 for link 2 and 40 for link 3. In cases with multiple AFEs on a PCB, valid ranges can be looked at for all AFEs and optimal RBD can be chosen.
4. The optimal RBD is set and adcDacSync is run.

Link 1

setManualRbd(afeInst,0,20)

Link 2

setManualRbd(afeInst,1,30)

Link 3

setManualRbd(afeInst,2,40)

adcDacSync(afeInst,1)

5. Read back link0_sysref_cnt_on_release_opportunity from the first page for link 1, *link1_sysref_cnt_on_release_opportunity* from the first page for link 2 and *link0_sysref_cnt_on_release_opportunity* from second page for link 3.

Link 1

getSysrefCntOnReleaseOpp(afeInst,0,&readVal)

If readVal = 1, optimal RBD is chosen as (20+1)%64 = **21**

Link 2

getSysrefCntOnReleaseOpp(afeInst,1,&readVal)

If readVal=1, optimal RBD is chosen as (30+1)%64 = **31**

Link 3

getSysrefCntOnReleaseOpp(afeInst,2,&readVal)

If readVal=1, optimal RBD is chosen as (40+1)%64 = **41**

6. In the config, value of 21 is set for the link0_rbd_m1 in the first page, value of 31 is set for the link1_rbd_m1 in the first page and value for 41 is set for the link0_rbd_m1 in the second page.

   In Latte, *sysParams.jesdRxRbd* is set to [21,31,41,41]

4.3.4 Use Case with 4 JESD Links

In the case of 4 DAC JESD links, there exist 2 links in top JESD and 2 other links in bottom JESD. In such a case, below procedure is used to set the RBD.

1. For the first link, based on the *lane0_f_counter_all_lane_ready* in the first page, *link0_rbd_m1* is set in the first page.
2. For the second link, based on the *lane2_f_counter_all_lane_ready* in the first page, *link1_rbd_m1* is set in the first page.
3. For the third link, based on the *lane0_f_counter_all_lane_ready* in the second page, *link0_rbd_m1* is set in the second page.
4. For the fourth link, based on the *lane2_f_counter_all_lane_ready* in the second page, *link1_rbd_m1* is set in the second page.

Example:

1. **Link 1**

   Read Page 0x16[2] : lane0_f_counter_all_lane_ready = 16 for link 1

   getAllLaneReady(afeInst,0,&laneAllReady)

   laneAllReady = 16 for link 1

   Link 2

   Read Page 0x16[2] : lane2_f_counter_all_lane_ready = 26 for link 2

   getAllLaneReady(afeInst,1,&laneAllReady)

   laneAllReady = 26 for link 2

   Link 3

   Read Page 0x16[3] : lane0_f_counter_all_lane_ready = 36 for link 3

   getAllLaneReady(afeInst,2,&laneAllReady)

   laneAllReady = 36 for link 3

   Link 4

   Read Page 0x16[3] : lane2_f_counter_all_lane_ready = 46 for link 4

   getAllLaneReady(afeInst,3,&laneAllReady)

   laneAllReady = 46 for link 4

2. Obtain valid range of RBD.

   Link 1

   getGoodRbdRange(afeInst,0,&rbdMin,&rbdMax)

   rbdMin = 17

rbdMax = 35 (Based on skew = 3, rbdMax = min(allLaneReady) + BufferDepth - 10 = 16 - 3 + 32 - 10)

Link 2

getGoodRbdRange(afeInst,1,&rbdMin,&rbdMax)

rbdMin = 27

rbdMax =45 (Based on skew = 3, rbdMax = min(allLaneReady) + BufferDepth - 10 = 26 - 3 + 32 - 10)

Link 3

getGoodRbdRange(afeInst,2,&rbdMin,&rbdMax)

rbdMin = 37

rbdMax =55 (Based on skew = 3, rbdMax = min(allLaneReady) + BufferDepth - 10 = 36 - 3 + 32 - 10)

Link 4

getGoodRbdRange(afeInst,3,&rbdMin,&rbdMax)

rbdMin = 47

rbdMax = 1 (Based on skew = 3, rbdMax = min(allLaneReady) + BufferDepth - 10 = 46 - 3 + 32 - 10)

3. In this example, optimal value is chosen as rbdMin+3 = 20 for link1, 30 for link 2, 40 for link 3 and 50 for link 4. In cases with multiple AFEs on a PCB, valid ranges can be looked at for all AFEs and optimal RBD can be chosen.
4. The optimal RBD is set and adcDacSync is run.

   Link 1

   setManualRbd(afeInst,0,20)

   Link 2

   setManualRbd(afeInst,1,30)

   Link 3

   setManualRbd(afeInst,2,40)

   Link 4

   setManualRbd(afeInst,3,50)

   adcDacSync(afeInst,1)

5. Read back link0_sysref_cnt_on_release_opportunity from the first page for link 1, link1_sysref_cnt_on_release_opportunity from the first page for link 2, link0_sysref_cnt_on_release_opportunity from second page for link 3 and link1_sysref_cnt_on_release_opportunity from second page for link 4.

   Link 1

   getSysrefCntOnReleaseOpp(afeInst,0,&readVal)

   If readVal=1, optimal RBD is chosen as (20+1)%64 = **21**

Link 2

getSysrefCntOnReleaseOpp(afeInst,1,&readVal)

If readVal=1, optimal RBD is chosen as (30+1)%64 = **31**

Link 3

getSysrefCntOnReleaseOpp(afeInst,2,&readVal)

If readVal=1, optimal RBD is chosen as (40+1)%64 = **41**

Link 4

getSysrefCntOnReleaseOpp(afeInst,3,&readVal)

If readVal=1, optimal RBD is chosen as (50+1)%64 = **51**

6. In the config, value of 21 is set for the link0_rbd_m1 in the first page, value of 31 is set for the link1_rbd_m1 in the first page, value for 41 is set for the link0_rbd_m1 in the second page and value for 51 is set for the link1_rbd_m1 in the second page. In Latte, *sysParams.jesdRxRbd* is set to [21,31,41,51]

5 Fixing Potential Alarms Related to RBD

5.1 RBD Alarm

The basic alarm for RBD overflow error is identified by read of alarms register. The mapping is as follows.

If 0x11C[4] = 1, JESD Lane 0/4 has RBD overflow error.

If 0x11D[4] = 1, JESD Lane 1/5 has RBD overflow error.

If 0x11E[4] = 1, JESD Lane 2/6 has RBD overflow error.

If 0x11F[4] = 1, JESD Lane 3/7 has RBD overflow error.

Depending on the page selected, the lane that has RBD overflow error can be identified. 0x16[2] provides information for lanes 0-3 and 0x16[3] for lanes 4-7.

Various possibilities in which RBD overflow alarm can happen is described as follows.

1. If the error shows up consistently across multiple configuration runs, it would mean that an invalid value of RBD is set. In this case, fcounter can be read back and optimal RBD can be set as described in Section 3 and Section 4.
2. If errors are intermittent, it could be either
   a. SYSREF not being latched deterministically across runs by AFE or FPGA/ASIC
   b. The FPGA/ASIC does not have a deterministic delay between SYSREF rising edge to start of EMB in 204C or start of ILA phase in 204B.
3. If running resync sequence (*adcDacSync* in CAFE) helps to solve the RBD error consistently, when it initially failed during the initial configuration run, it is possible that the SerDes did not adapt during the configuration sequence. Ensure that there exists data on the SRX lanes of AFE during the configuration.

5.2 SoEMB Close to LEMC Edge

It is possible that when the SoEMB is close to LEMC edge, some of *lane0/1/2/3_f_counter_any_lane_ready* read a very small value close to 0, and some lanes read values close to maximum value of fcounter (63 when E=1). The buffer index is essentially a circular indexing as the buffer has to absorb both advancements and delays in time. Such a case is shown in Figure 5-1 [7].

Initially, the *lane0/1/2/3_f_counter_any_lane_ready* registers read from the range of (62,63,0,1,2). Specifically, *lane0_f_counter_any_lane_ready* reads 62 and *lane3_f_counter_any_lane_ready* reads 2. In such cases, it is recommended to shift the LEMC counter by writing to *link0_init_f_counter*, so that all fcounter values are close by values.

On rising edge of LEMC, the counter starts from value of *link0_init_f_counter*. In other words, the LEMC counter is shifted by *link0_init_f_counter*.

In Figure 5-1, we see that *link0_init_f_counter* is set to 7. With this, *lane0_f_counter_any_lane_ready* reads ((62+7)%64) = 5 and *lane3_f_counter_any_lane_ready* reads 2+7 = 9. RBD is set as 13 which has a margin of 4 from the last arriving lane. Note that the RBD value is also with reference to the shifted LEMC counter.

This can be done in Latte by setting the following parameter:

sysParams.jesdRxInitLmfcCounter = 7

[7] Parts of figures were based on JEDEC JESD204C standard, Figure 5 and Figure 50. Copyright JEDEC. Modifications have not been approved by and do not reflect the views of JEDEC.

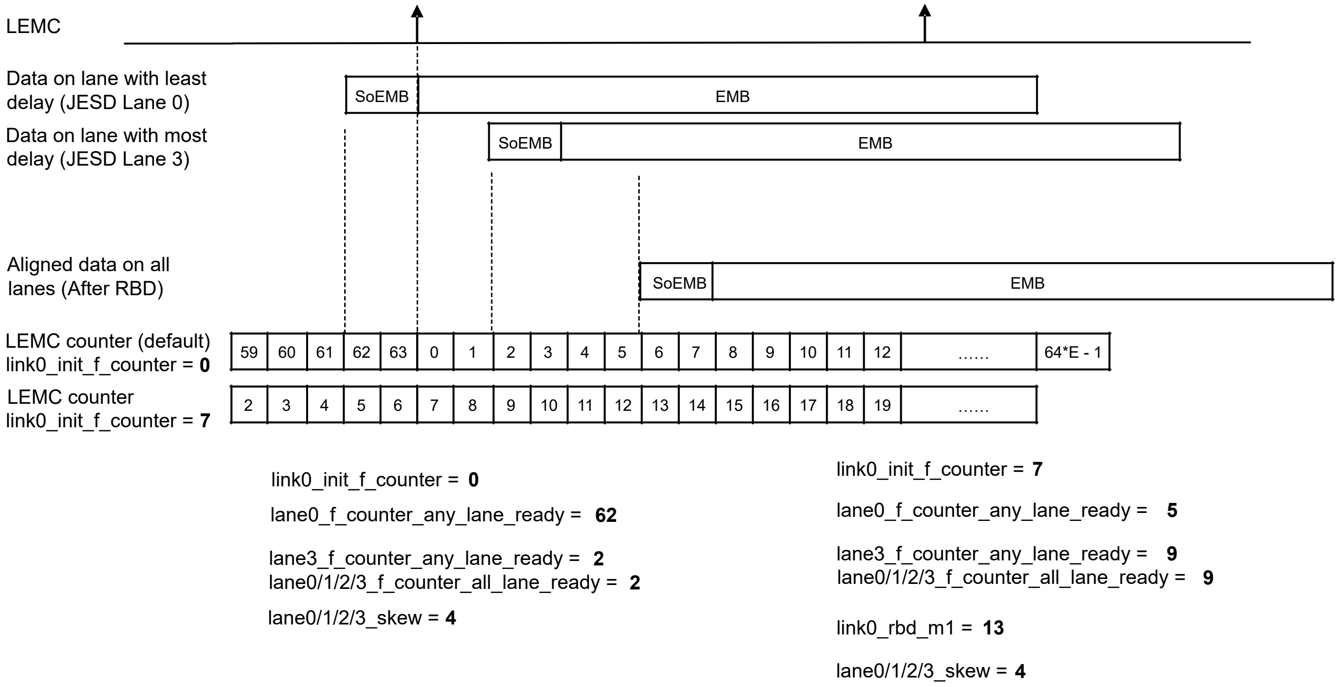

LEMC
Data on lane with least delay (JESD Lane 0)
SoEMB
EMB
Data on lane with most delay (JESD Lane 3)
SoEMB
EMB
Aligned data on all lanes (After RBD)
SoEMB
EMB
LEMC counter (default) link0_init_f_counter = 0
59 60 61 62 63 0 1 2 3 4 5 6 7 8 9 10 11 12 ...... 64*E - 1
LEMC counter link0_init_f_counter = 7
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 ......
link0_init_f_counter = 0
lane0_f_counter_any_lane_ready = 62
lane3_f_counter_any_lane_ready = 2
lane0/1/2/3_f_counter_all_lane_ready = 2
lane0/1/2/3_skew = 4
link0_init_f_counter = 7
lane0_f_counter_any_lane_ready = 5
lane3_f_counter_any_lane_ready = 9
lane0/1/2/3_f_counter_all_lane_ready = 9
link0_rbd_m1 = 13
lane0/1/2/3_skew = 4

Figure 5-1. JESD204C LEMC Edge Close to SoEMB

5.3 Start of ILA Close to LMFC Edge

It is possible that when the start of ILA is close to LMFC edge, some of lane0/1/2/3_f_counter_any_lane_ready read a very small value close to 0, and some lanes read values close to maximum value of fcounter (31 when K=32 and F=4). The buffer index is essentially a circular indexing as the buffer has to absorb both advancements and delays in time. Such a case is shown in Figure 5-2 [8].

Initially, the *lane0/1/2/3_f_counter_any_lane_ready* registers read from the range of (30,31,0,1,2). Specifically, *lane0_f_counter_any_lane_ready* reads 30 and *lane3_f_counter_any_lane_ready* reads 2. In such cases, it is recommended to shift the LMFC counter by writing to *link0_init_f_counter*, so that all fcounter values are close by values.

On rising edge of LMFC, the counter starts from value of *link0_init_o_counter/4*. In other words, the LMFC counter is shifted by *link0_init_o_counter/4* × *F*. In JESD204B, the *link0_init_o_counter* can only be programmed to values that are multiples of 4.

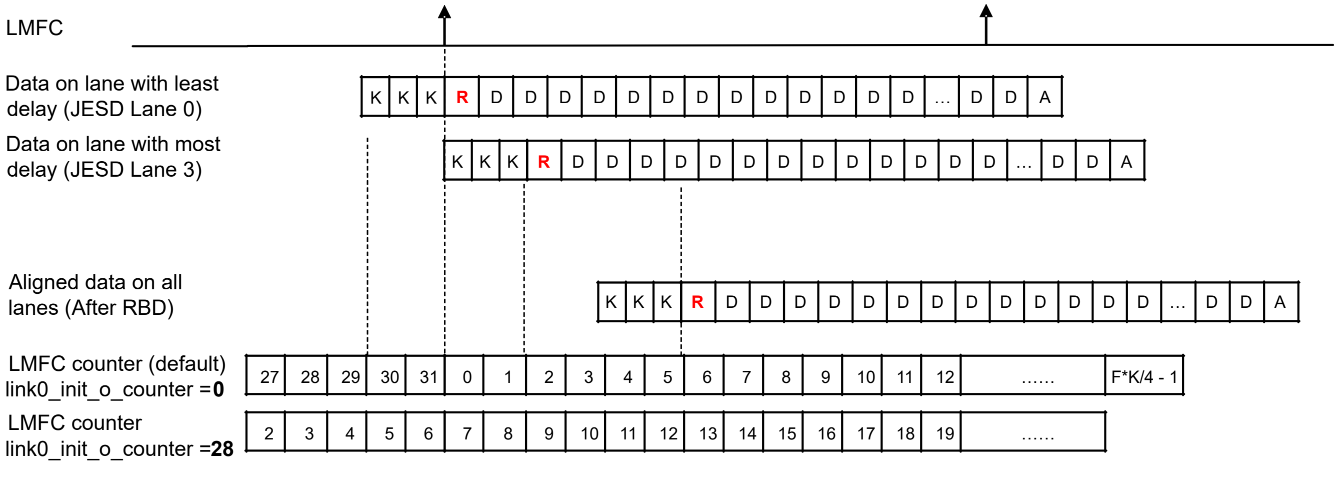
In Figure 5-2, *link0_init_o_counter* is set to 28 (In this case F is assumed to be 1). With this, *lane0_f_counter_any_lane_ready* reads ((30+28/4)%32) = 5 and *lane3_f_counter_any_lane_ready* reads 2+28/4 = 9. RBD is set as 13 which has a margin of 4 from the last arriving lane.

Note that the RBD value is also with reference to the shifted LMFC counter.

This can be done in Latte by setting the following paramter:

sysParams.jesdRxInitLmfcCounter = 7

[8] Parts of figures were based on JEDEC JESD204C standard, Figure 5 and Figure 50. Copyright JEDEC. Modifications have not been approved by and do not reflect the views of JEDEC.


LMFC
Data on lane with least delay (JESD Lane 0)
K K K R D D D D D D D D D D D D D ... D D A
Data on lane with most delay (JESD Lane 3)
K K K R D D D D D D D D D D D D D ... D D A
Aligned data on all lanes (After RBD)
K K K R D D D D D D D D D D D D D ... D D A
LMFC counter (default) link0_init_o_counter =0
27 28 29 30 31 0 1 2 3 4 5 6 7 8 9 10 11 12 ...... F*K/4 - 1
LMFC counter link0_init_o_counter =28
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 ......

link0_init_o_counter =**0**

lane0_f_counter_any_lane_ready =**30**

lane3_f_counter_any_lane_ready = **2**
lane0/1/2/3_f_counter_all_lane_ready = **2**

lane0/1/2/3_skew = **4**

link0_init_o_counter =**28**

lane0_f_counter_any_lane_ready = **5**

lane3_f_counter_any_lane_ready = **9**
lane0/1/2/3_f_counter_all_lane_ready = **9**

link0_rbd_m1 =**13**

lane0/1/2/3_skew = **4**

Figure 5-2. JESD204B LMFC Edge Close to Start of ILA

6 References

1. JEDEC Standard: JESD204C.1
2. AFE79xx Technical Reference Manual
3. Texas Instruments, JESD204B Overview, high speed data converter training.
4. Texas Instruments, System Design Considerations when Upgrading fromJESD204B to JESD204C, application note.

IMPORTANT NOTICE AND DISCLAIMER

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265