

# 1 JESD204B 确定性时延

JESD204B 规定的两种需要实现确定性时延的 Subclass，主要需要设计两部分的对齐策略。一是 frame 层面的确定性时延，这一部分主要是保证在本地 multiframe 时钟控制下，接收到的数据能同步的传输到接下来的模块中。二是本地 multiframe 时钟层面的确定性时延，这一部分主要保证各个设备间的本地 multiframe 时钟能够同步，从而使整个系统之间能够有同样的收发节拍，能够将各个模块、设备之间的不确定性降到最低。

## 1.1 frame 层面的确定性时延实现

link 之间的确定性时延定义在收端和发端的以 frame 为基础的并行数据层面，即保证在 frame 时钟域。不同 link 间的时延保证需要以 frame 时钟周期为单位，并且能够通过寄存器配置完成具体时延的设置。这样的设置需要在上电或者 link 重连阶段能够有一定的重复性。

为了达成确定性时延，需要满足以下两个条件：

- 对于发端设备，所有 lane 之间的 ILA 序列产生必须同时进行，这也同时保证了数据流同时产生。
- 对于收端设备，每个 lane 收到的数据必须在各个物理通道间设置缓存。收端缓冲需要在一个预先定义好的时刻，同时释放所有 lane 的数据。这个特殊时刻指的是可编程数量 frame 周期 RBD<sup>1</sup>（在本地 multiframe 时钟边界后）。

ILAS 生成和接收端缓冲释放对齐，都是以接收端和发送端的本地 multiframe 时钟为标准。所以，确定性时延的最小不确定性依赖于接收端和发送端之间本地 multiframe 时钟的对齐程度。

为了实现确定性时延的协议要求，系统设计时需要保证以下几点：

- multiframe 长度必须大于 link 间可能存在的最大时延。
- $RDB * T_f$  的值必须大于 link 间可能存在的最大时延。
- RDB 的值，依照 frame 周期，必须介于 1 到 K 之间。

以上三条的目的是保证 RBD 够大，能够使接收端缓冲在释放前，所有 lane 的发送端数据被收到。最终 JESD204B link 间的时延为  $RDB * T_f$ 。JESD204B link 间的确定性时延要求接收端设备能够缓冲收到所有 lane 的 ILA 序列或者用户数据，在接收端缓冲能被释放前。缓冲必须在本地 multiframe 时钟边界过后，RBD 个 frame 周期之后释放。为了能够释放缓冲，需要满足所有 lane 都有“有效数据”存在各自的接收端缓冲中，这里有“有效数据”指：

- 如果 ILA 序列发送到了接收端缓冲中，那么有效数据指的是 ILA 序列的开端。
- 如果 ILA 序列没有发送到接收端缓冲中，那么有效数据指的就是完整 ILA 序列之后出现的样本数据。在这种情况下接收端缓冲将会比第一种情况迟 4 个 multiframe 释放。

link 间的时延可由下式表示，这也是所谓的确定性时延需要保证的确定部分：

$$Delay_{LINK} = \delta T_{LMFC} = TX \text{ delay} + Lane \text{ Delay} + RX \text{ delay}$$

其中， $TX \text{ delay}$  指的是从发送端 ILA 序列产生，并出现在发送端 SerDes 的输出端口上的时间； $Lane \text{ delay}$  指的是穿过外部物理信道的时延； $RX \text{ delay}$  指的是从接收端 SerDes 输入到通入缓冲输出的时延。ILA 序列的开端或者用户数的开端，将会在本本地 multiframe 时钟边加上 RBD 个帧时钟周期时出现在缓冲输出； $\delta T_{LMFC}$  指的是 link 间的全部时延，指的是发送端本地 multiframe 时钟上升沿到接收端本地 multiframe 时钟加上  $T_f * RBD$  上升沿之间的时延。

接收端缓冲的最小大小取决于最早可能到达接收端缓冲输入的数据和下一个接收端缓冲释放条件产生之间的时差。

Subclass 0 类设备并不支持确定性时延。

<sup>1</sup>Rx Buffer Delay

## 2 本地 multiframe 时钟层面的确定性时延

由 frame 层面的确定性时延处理方法可以总结出来，各个设备之间的同步最关键在于同步各自的本地 multiframe 时钟边界，即需要各个设备能够将自己的本地 multiframe 时钟对准到所需要的边界上。Subclass 1 类和 Subclass 2 类设备最大的区别就在于本地 multiframe 时钟的对齐方式上，Subclass 1 类设备需要额外的外部时钟信号 SYSREF 来保证同步，而 Subclass 2 类设备则需要通过自身的 SYNC 信号来保证同步。

在保证本地 multiframe 时钟同步之后，之后对于 frame 同步的处理都是相同的。

### 2.1 Subclass 1 类设备的本地 multiframe 时钟同步

Subclass 1 类设备的本地 multiframe 时钟同步主要是通过对齐外部的 SYSREF 信号。需要处理的是两个问题，一个是 SYSREF 信号到达各个设备的同步问题，一个是设备自己的本地 multiframe 时钟与 SYSREF 信号对齐的时机问题。

#### 2.1.1 SYSREF 信号同步

针对第一个问题，协议要求所有设备的本地 multiframe 时钟必须要相同，但是 SYSREF 也是一个时钟信号，在传输和产生过程中，也会产生一定的时延，比如布线布局上存在的时延差问题。所以在 Subclass 1 类设备中通过自己的设备时钟，当 SYSREF 信号到达后，产生指定数量的功能性时延后再对本地 multiframe 时钟进行对齐，以保证各个设备的本地 multiframe 时钟是同步的。而对于时延数量的设置需要根据具体的电路设计来确定。

#### 2.1.2 本地 multiframe 时钟与 SYSREF 对齐时机

针对第二个问题，协议并没有强制设定具体的对齐时机，但给出了一些建议。发送端和接收端设备需要有能力强决定是否校正本地 frame 和 multiframe 时钟到 SYSREF 脉冲。这一功能实现的细节由设计者决定，但是有以下三种可能的选项供选择。

- 每一个 SYSREF 脉冲需要被设备验证，以决定现有的本地 multiframe 时钟相位和 frame 时钟是否需要对齐。
- 一个设备可以通过设备的输入引脚或者控制接口命令来决定是否用下一个收到的 SYSREF 脉冲强制对齐本地 multiframe 时钟和本地 frame 相位。
- 一个设备可以通过设备的输入引脚或者控制接口命令来决定是否忽略所有的 SYSREF 脉冲。

### 2.2 Subclass 2 设备的本地 multiframe 时钟同步

Subclass 2 类设备不同于 Subclass 1 类设备，Subclass 2 类设备的本地 multiframe 时钟对齐要更加的复杂，主要依靠的是接收端和发送端反复的协商完成同步。Subclass 2 类设备由于没有共同的外部时钟源，所以本地 multiframe 时钟的同步完全依赖于 SYNC 接口的信号变化，并且对于发送端和接收端的处理存在主从的区别。需要处理的也是两个问题，一是最初链接建立的本地 multiframe 时钟同步问题，二是重新要求对齐的时机问题。

#### 2.2.1 建立本地 multiframe 时钟同步方法

针对第一个问题，需要明确设备的主从关系，由主设备发起对本地 multiframe 时钟的调整请求。发送端设备作为主设备，接收端设备作为从设备，目标就是通过发送端设备收到的 SYNC 变化边界通过对比发送端设备本地的本地 multiframe 时钟边界判断是否需要调整。由于之前协议的规定，接收端改变 SYNC 信号的时机是来自接收端的本地 multiframe 时钟，这就意味着发送端得到 SYNC 变化实质上是接收端的本地 multiframe 时钟边界。当发送端收到这个信号后就会将这个信号对自己的本地 multiframe 时钟进行比较，从而判断是否符合同步条件，是否需要调整。当发送端设备准确判断出需要调整时，会在接下来传输的 ILA 序列中添加调整参数，这几个参数是 ADJCNT（调整步数）、ADJDIR（调整方向）、PHADJ（调整使能）。这种情况下发送端不会发送具体数据，纯粹是利用 ILA 序列进行本地 multiframe 时钟的同步对齐。而接收端设备在收到 ILA 序列中对于调整的需求时就将对自己的本地 multiframe

时钟进行调整，完成调整后在新的本地 multiframe 时钟上升沿在 SYNC 接口发送一个错误报告。发送端口又会根据这个错误报告的本地 multiframe 时钟边界判断是否需要继续调整，是否在合适的范围呢。如此循环直到同步建立，当 T 发送端认为同步成功后就会发送具体的数据，并置低 PHADJ。

这样的调整机制是建立在两个基础上。

- 一是发送端能够精确的比较 SYNC 变化和本地 multiframe 时钟的差异，于是需要在发送端建立了检测分辨率和检测间隙机制。实际上就是对本地设备时钟的精度有一定的要求，能够准确判断本地 multiframe 时钟和 SYNC 信号变化之间的相位差的差距具体时延数。而本地 multiframe 时钟对齐的精确度也是由这一机制来保证，设备时钟的精确度直接决定了检测的精度。
- 二是接收端能够根据收到的调整数据对本地相位进行调整，于是在接收端建立了调整分辨率和调整时钟机制，调整时钟直接决定了调整的范围和精度。

事实上，光有机制的保证还是不够的，还需要调整算法的支持，以及调整范围、极限的设置。时钟一次性调整的范围是有极限的，最大不会超过 16 个调整时钟的周期。并且调整的方向也是会变化的，如何避免在调整过程中差异发散而非收敛，如何能够使调整尽快收敛进行传输也是需要设计者考虑的问题。

### 2.2.2 重新对齐时机问题

经过一定时间的正常传输后难免产生时钟的偏移问题，累计误差从而导致本地 multiframe 时钟不同步，这时就需要考虑重新对齐的问题。发送端的 SYNC 相位监视窗口会始终去判断是否需要重新对齐，一旦相位变化超出预设的窗口就表明延时丢失，需要重新发起初始化，来对齐本地 multiframe 时钟。重新对齐的时机取决于使用者的配置，发起对齐的方法就是重新进行初始化循环，通过迭代的方法完成本地 multiframe 时钟对齐。