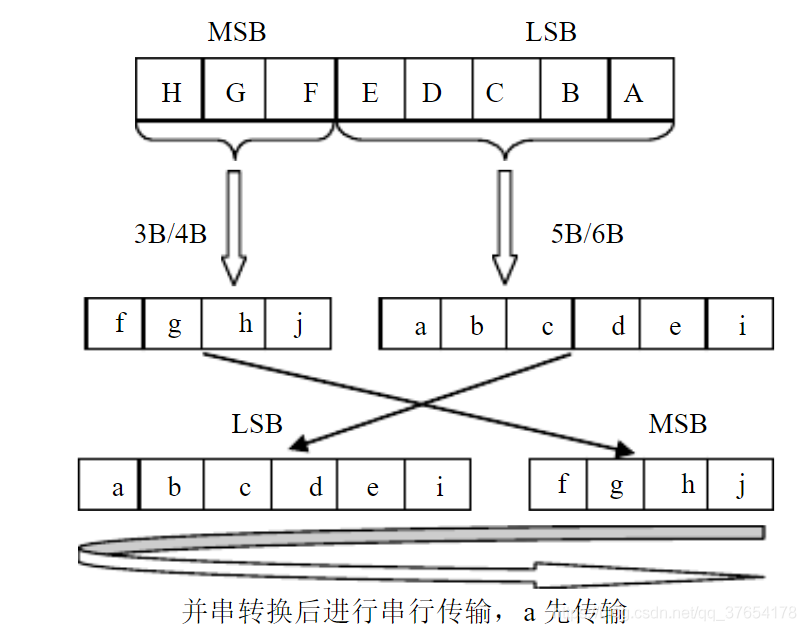
1. aurora\_8b10b编码，为什么？原理？

* 为什么需要8b10B编码？
  + 简单的说是为了直流均衡，在高速串行总线中，通常采用交流耦合的方式，即在发送端（Tx）串接电容，根据电容“隔直流，通交流”的特性，或者理想电容的阻抗公式： xxxxx 信号的频率越高，电容阻抗越低，当数据位流中出现多个连续的0或者1时，可以认为这段时间内信号是直流，电容的损耗变大，导致信号的幅度降低，直流信号被滤除，到最后无法识别出是1还是0，
  + 8B/10B编码以字节为单位，将数据映射成10位宽度的数据（具体映射方式可查表），使得编码后的二进制数据流中1和0的数量基本保持一致，同时确保字节同步易于实现。
* 原理
  + 8B/10B编码是1983年IBM公司的AI Widmer和PeterFranaszek提出的输出传输编码标准[,](_self)目前广泛应用在高速串行总线中。
  + 假设原始8bit数据从高到低采用HGFEDCBA表示，8B/10B编码将8位数据分成高三位HGF和低5位EDCBA两个子组，然后经过5B/6B编码，将低5位EDCBA映射成abcdei;高三位经过3B/4B映射成fghj,最后合成abcdeifghj发送，发送时由于是小端模式，a先发送，其对应关系如下图所示：

通常，认为会将低5位EDCBA按其十进制数值记为x，将高3位按其十进制数值记为y，将原始8bit数据记为D.x.y。例如8bit数“101 10101”，即十进制数181，按照上述划分原则x=10101（21），y=101（5），所示这个数被表示为D.21.5。此外在8B/10B编码中，还需用到12种控制字符，用来标识传输数据的开始和结束，传输空闲等状态，按照上述规则，将控制字符记为K.x.y。

这个时候问题来了，8位原始数据对应256个码，加上12种控制字符，而编码后的10位数据有1024个码，肯定有很多是用不到的，故需选择其中一部分来表示8bit数据，所选的码字0和1的数量应尽可能相等。

表1所示为5B/6B编码映射关系表2：

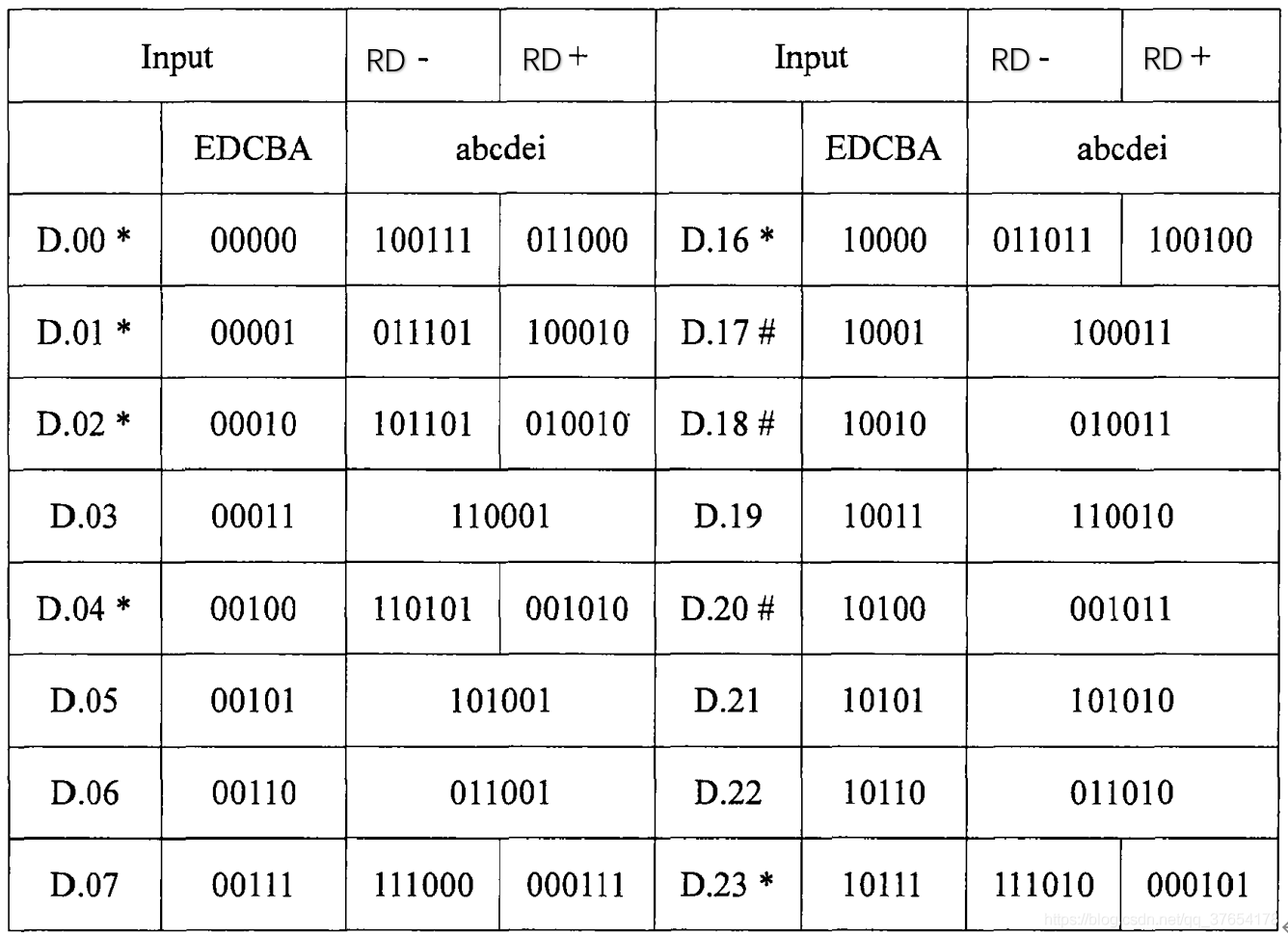


表2所示为3B/4B编码映射关系表：

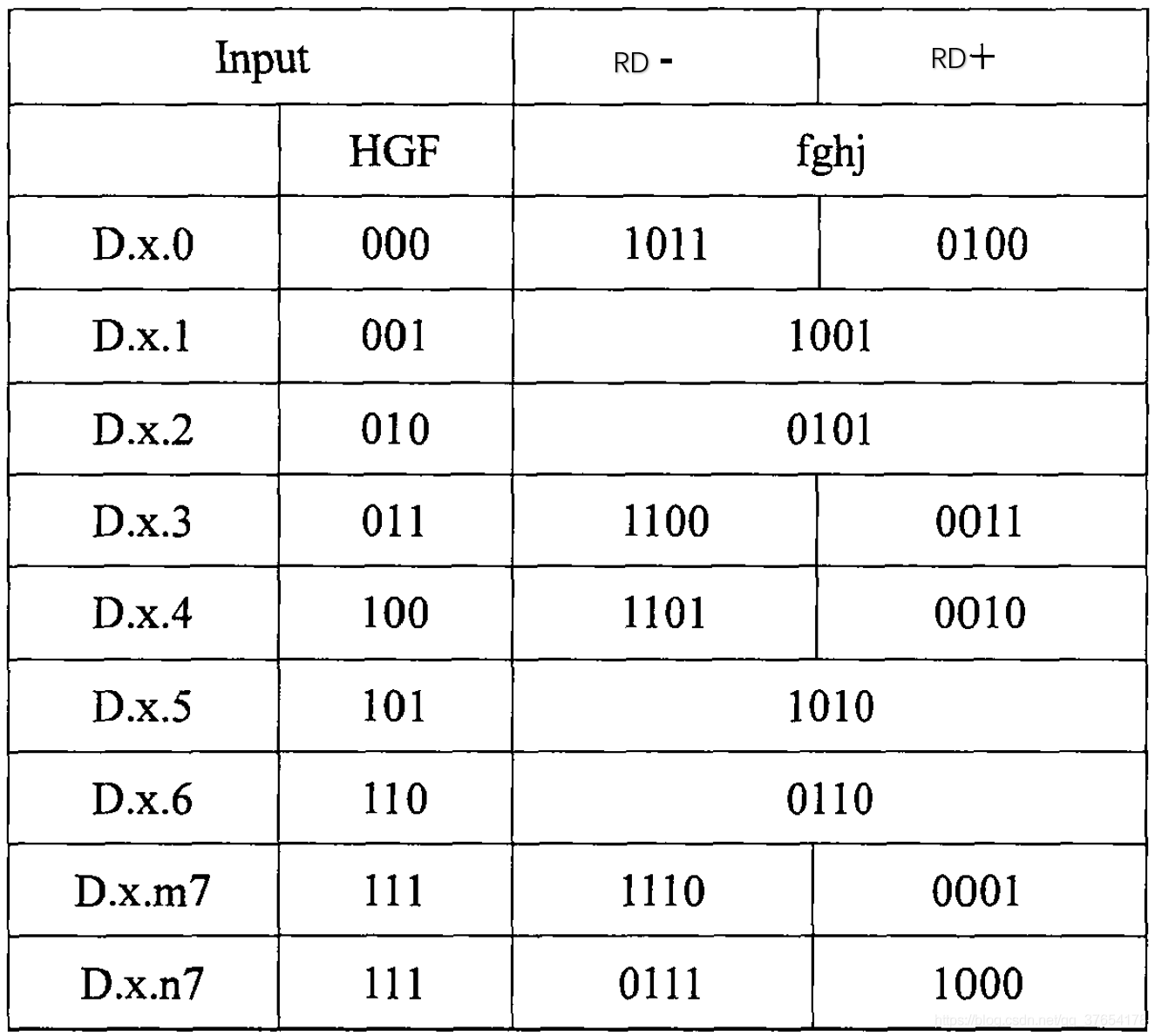
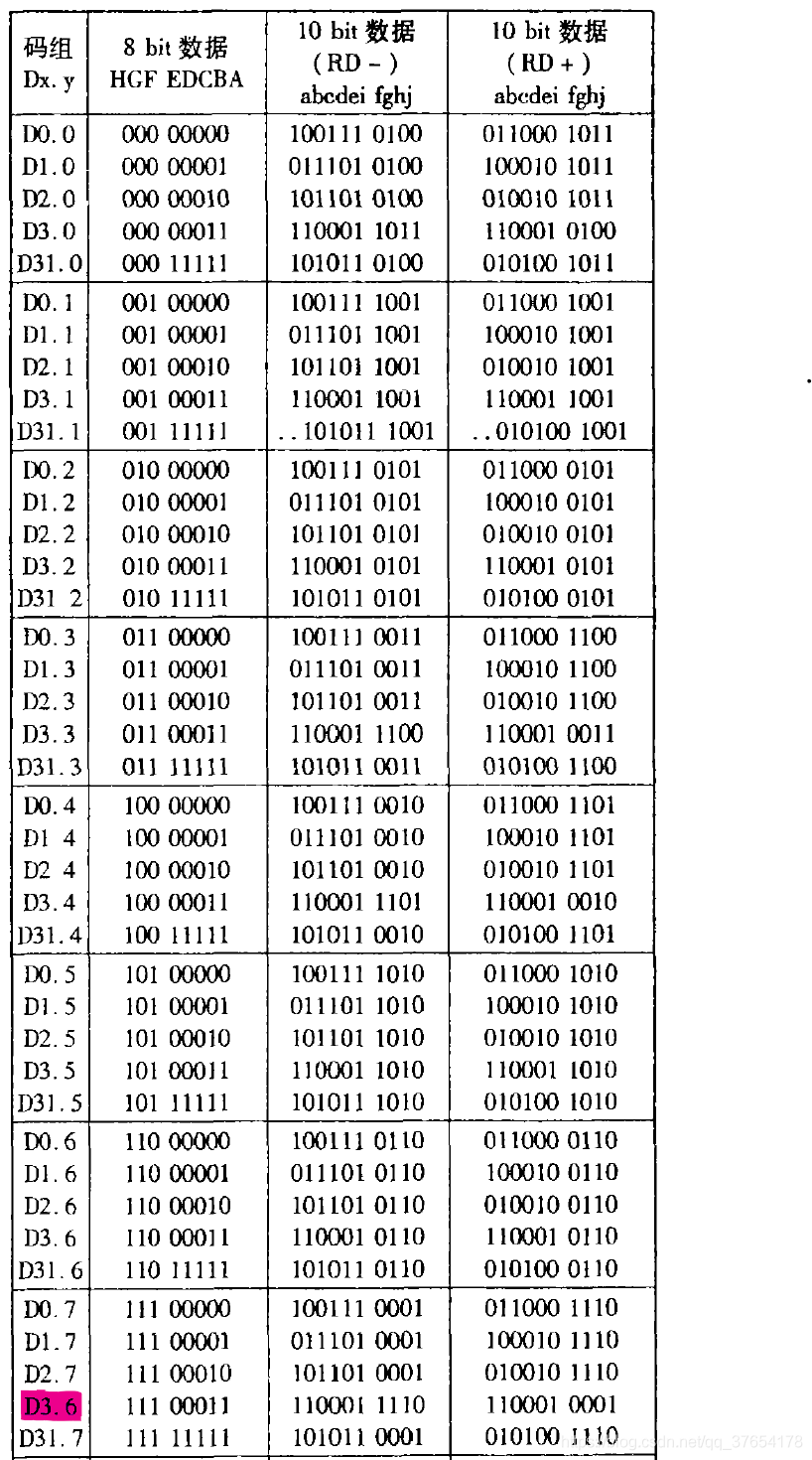


表3所示为8B/10B编码映射关系表[3](_self)：其映射原理将在下文中阐述



1. DDR效率太低怎么调整才能提高效率？
2. Serdes误码了怎么办？
3. 高速通道时钟偏移怎么办？
4. 建立时间保持时间是什么？（建立时间：时钟上升沿之前数据必须稳定的最短时间。保持时间：时钟上升沿之后数据必须稳定的最短时间）
5. 建立时间满足，保持时间不满足是什么情况？

* setup violation是由于前级组合的延迟过大引起的，因此要用set\_max\_delay来限制，  
  hold violation是由于前级组合延迟过小引起的，因此不能让他太小，要用set\_min\_delay来限制。
* 如果Tco延时太短导致上一级寄存器锁存的数据侵占了下一级寄存器正在锁存数据的保持时间,那么下一级寄存器就无法有效的锁存数据,系统时序也就无法达到要求。
* 上面说的约束限制可能不对。

1. 64个if else综合出来底层是什么 LUT
2. 1bit信号，不同时钟域，不知道具体时钟频率时，怎么解决跨时钟域问题

参考：分析思路：主要是分为两种情况：

一是：快时钟域采慢时钟域

二是：慢时钟采快时钟域：这个要注意让快时钟域的数据保持的时间要够长

异步处理主要是靠打拍来解决。或者是使用异步FIFO

1. 三段式状态机，状态转移时如何保证状态不乱跳（在case上面增加一个状态赋值为IDLE）
2. 同步fifo现场编写

* 主要是考FIFO的设计思路，需要两套指针系统，分别为读地址指针，和写地址指针。

1. 异步fifo原理，怎么写？
2. cordic原理及精度