

SerDes 知识详解

一、SerDes 的作用

1.1 并行总线接口

在 SerDes 流行之前,芯片之间的互联通过系统同步或者源同步的并行接口传输数据,图

1.1 演示了系统和源同步并行接口。

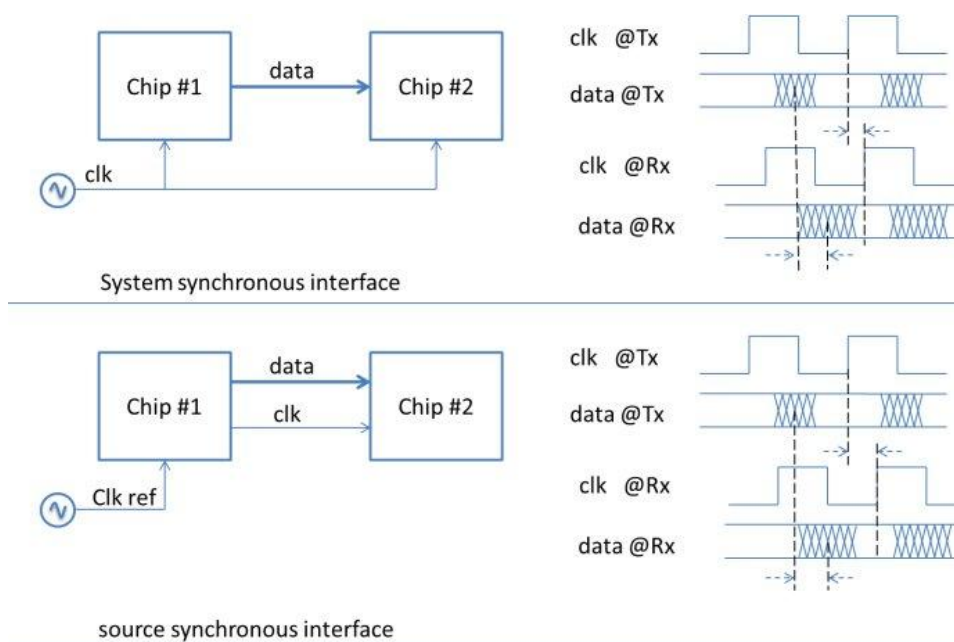


Figure 1.1 traditional data transfer interface

随着接口频率的提高,在系统同步接口方式中,有几个因素限制了有效数据窗口宽度的继续增加。

- 时钟到达两个芯片的传播延时不相等(clock skew)
- 并行数据各个 bit 的传播延时不相等(data skew)
- 时钟的传播延时和数据的传播延时不一致(skew between data and clock)

虽然可以通过在目的芯片(chip #2)内用 PLL 补偿时钟延时差(clock skew),但是 PVT 变化时,时钟延时的变化量和数据延时的变化量是不一样的。这又进一步恶化了数据窗口。

源同步接口方式中,发送侧 Tx 把时钟伴随数据一起发送出去,限制了 clock skew 对有效数据窗口的危害。通常在发送侧芯片内部,源同步接口把时钟信号和数据信号作一样的处理,

也就是让它和数据信号经过相同的路径，保持相同的延时。这样 PVT 变化时，时钟和数据会朝着同一个方向增大或者减小相同的量,对 skew 最有利。

我们来做一些合理的典型假设，假设一个 32bit 数据的并行总线，

- a)发送端的数据 skew = 50 ps ---很高的要求
- b)pcb 走线引入的 skew = 50ps ---很高的要求
- c)时钟的周期抖动 jitter = +/-50 ps ---很高的要求
- d)接收端触发器采样窗口 = 250 ps ---Xilinx V7 高端器件的 IO 触发器

可以大致估计出并行接口的最高时钟 = $1/(50+50+100+250) = 2.2\text{GHz}$ (DDR)或者 1.1GHz (SDR)。

利用源同步接口，数据的有效窗口可以提高很多。通常频率都在 1GHz 以下。在实际应用中可以见到如 SPI4.2 接口的时钟可以高达 DDR 700MHz x 16bits 位宽。DDR Memory 接口也算一种源同步接口，如 DDR3 在 FPGA 中可以做到大约 800MHz 的时钟。

要提高接口的传输带宽有两种方式，一种是提高时钟频率，一种是加大数据位宽。那么是不是可以无限制的增加数据的位宽呢？这就要牵涉到另外一个非常重要的问题-----同步开关噪声(SSN)。

这里不讨论 SSN 的原理，直接给出 SSN 的公式： $SSN = L * N * di/dt$ 。

L 是芯片封装电感，N 是数据宽度，di/dt 是电流变化的斜率。

随着频率的提高，数据位宽的增加，SSN 成为提高传输带宽的主要瓶颈。图 1.2 是一个 DDR3 串扰的例子。图中低电平的理论值在 0V,由于 SSN 的影响，低电平表现为震荡，震荡噪声的最大值达 610mV，因此噪声余量只有 $1.5\text{V}/2 - 610\text{mV} = 140\text{mV}$ 。

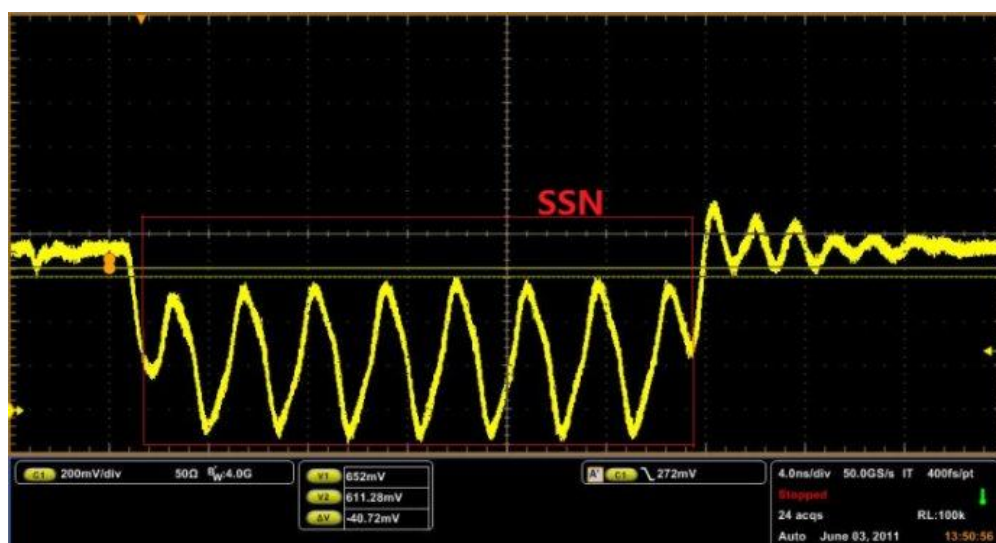


Figure 1.2 DDR3 串扰演示

因此也不可能靠无限的提高数据位宽来继续增加带宽。一种解决 SSN 的办法是使用差分信号替代单端信号，使用差分信号可以很好的解决 SSN 问题，代价是使用更多的芯片引脚。使用差分信号仍然解决不了数据 skew 的问题，很大位宽的差分信号再加上严格的时序限制，给并行接口带来了很大的挑战。

1.2 SerDes 接口

源同步接口的时钟频率已经遇到瓶颈，由于信道的非理想(channel)特性，再继续提高频率，信号会被严重损伤，就需要采用均衡和数据时钟相位检测等技术。这也就是 SerDes 所采用的技术。SerDes(Serializer-Deserializer)是串行器和解串器的简称。串行器(Serializer)也称为 SerDes 发送端(Tx)，(Deserializer)也称为接收端 Rx。Figure1.3 是一个 N 对 SerDes 收发通道的互连演示,一般 N 小于 4。

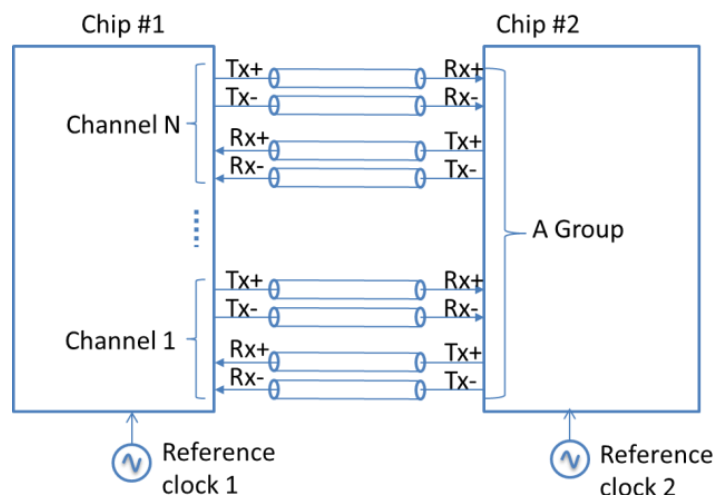


Figure 1.3 SerDes Inter-Connect Example

可以看到，SerDes 不传送时钟信号，这也是 SerDes 最特别的地方，SerDes 在接收端集成了 CDR(Clock Data Recovery)电路，利用 CDR 从数据的边沿信息中抽取时钟，并找到最优的采样位置。

SerDes 采用差分方式传送数据。一般会有多个通道的数据放在一个 group 中以共享 PLL 资源，每个通道仍然是相互独立工作的。

SerDes 需要参考时钟(Reference Clock)，一般也是差分的形式以降低噪声。接收端 Rx 和发送端 Tx 的参考时钟可以允许几百个 ppm 的频差(plesio-synchronous system)，也可以是

同频的时钟，但是对相位差没有要求。

作个简单的比较，一个 SerDes 通道(channel)使用 4 个引脚(Tx+/-,Rx+/-)，目前的 FPGA 可以做到高达 28Gbps。而一个 16bits 的 DDR3-1600 的线速率为 $1.6\text{Gbps} \times 16 = 25\text{Gbps}$ ，却需要 50 个引脚。此对比可以看出 SerDes 在传输带宽上的优势。

相比源同步接口，SerDes 的主要特点包括：

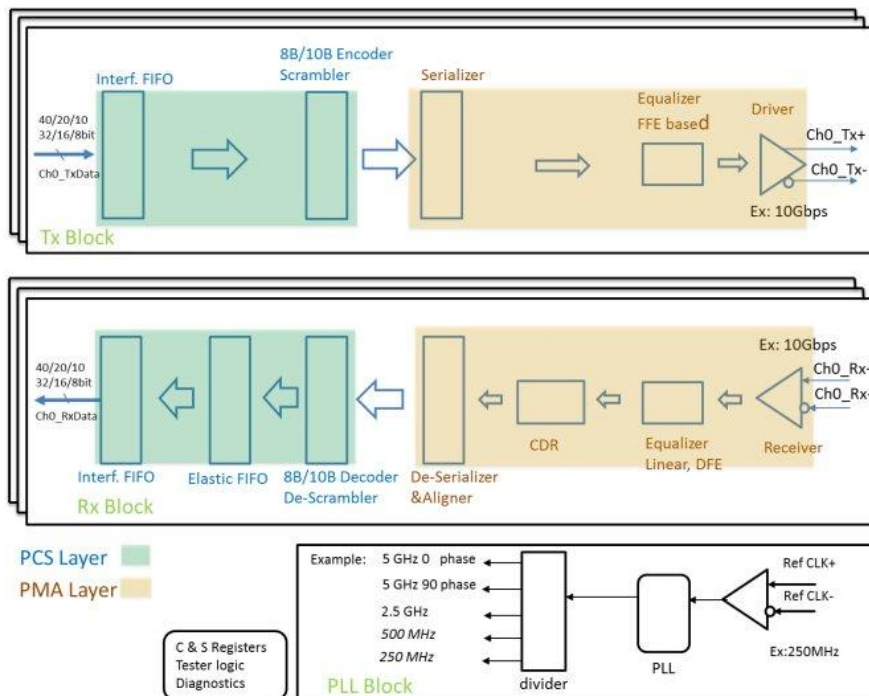
- SerDes 在数据线中时钟内嵌，不需要传送时钟信号。
- SerDes 通过加重/均衡技术可以实现高速长距离传输，如背板。
- SerDes 使用了较少的芯片引脚

1.3 中间类型

也存在一些介于 SerDes 和并行接口之间的接口类型，相对源同步接口而言，这些中间类型的接口也使用串行器(Serializer)解串器(Deserializer)，同时也传送用于同步的时钟信号。这类接口如视频显示接口 7:1 LVDS 等。

二、SerDes 结构(architecture)

SerDes 的主要构成可以分为三部分，PLL 模块，发送模块 Tx，接收模块 Rx。为了方便维护和测试，还会包括控制和状态寄存器，环回测试，PRBS 测试等功能。见图 2.1。



图中蓝色背景子模块为 PCS 层，是标准的可综合 CMOS 数字逻辑，可以硬逻辑实现，也可以使用 FPGA 软逻辑实现，相对比较容易理解。褐色背景的子模块是 PMA 层，是数模混合 CML/CMOS 电路，是理解 SerDes 区别于并行接口的关键，也是本文要讨论的内容。

发送方向(Tx)信号的流向：FPGA 软逻辑(fabric)送过来的并行信号，通过接口 FIFO(Interface FIFO)，送给 8B/10B 编码器(8B/10B encoder)或扰码器(scrambler)，以避免数据含有过长连零或者连 1。之后送给串行器(Serializer)进行 并→串 转换。串行数据经过均衡器(equalizer)调理，有驱动器(driver)发送出去。

接收方向(Rx)信号的流向，外部串行信号由线性均衡器(Linear Equalizer)或 DFE (Decision Feedback Equalizer 判决反馈均衡)结构均衡器调理，去除一部分确定性抖动(Deterministic jitter)。CDR 从数据中恢复出采样时钟，经解串器变为对齐的并行信号。8B/10B 解码器(8B/10B decoder)或解扰器(de-scrambler)完成解码或者解扰。如果是异步时钟系统(plesio-synchronous system)，在用户 FIFO 之前还应该弹性 FIFO 来补偿频差。

补充：均衡器

在通信系统的基带或中频部分插入的，能够减少码间干扰，起到补偿作用的滤波器。分为频域均衡器和时域均衡器。

频域均衡器

频域均衡器利用可调滤波器的频率特性来弥补实际信道的幅频特性和群延时特性，使包括均衡器在内的整个系统的总频率特性满足无码间干扰传输条件。

时域均衡器

时域均衡器是直接从时间响应角度考虑，使包括均衡器在内的整个传输系统的冲激响应满足无码间干扰条件。频域均衡满足奈奎斯特整形定理的要求，仅在判决点满足无码间干扰的条件相对宽松一些。所以，在数字通信中一般时域均衡器使用较多。

时域均衡器可以分两大类：线性均衡器和非线性均衡器。如果接收机中判决的结果经过反馈用于均衡器的参数调整，则为非线性均衡器；反之，则为线性均衡器。在线性均衡器中，最常用的均衡器结构是线性横向均衡器，它由若干个抽头延迟线组成，延时时间间隔等于码元间隔。非线性均衡器的种类较多，包括判决反馈均衡器(DFE)、最大似然(ML)符号检测器和最大似然序列估计等。

PLL 负责产生 SerDes 各个模块所需要的时钟信号,并管理这些时钟之间的相位关系。以图中线速率 10Gbps 为例,参考时钟频率 250MHz。Serializer/Deserializer 至少需要 5GHz 0 相位时钟和 5GHz 90 度相位时钟,1GHz(10bit 并行)/1.25GHz(8bit 并行)时钟等。

一个 SerDes 通常还要具调试能力。例如伪随机码流产生和比对，各种环回测试，控制状态寄存器以及访问接口，LOS 检测，眼图测试等。

2.1 串行器解串器(Serializer/Deserializer)

串行器 Serializer 把并行信号转化为串行信号。Deserializer 把串行信号转化为并行信号。一般地, 并行信号为 8/10bit 或者 16/20bit 宽度, 串行信号为 1bit 宽度(也可以分阶段串行化, 如 8bit->4bit->2bit->equalizer->1bit 以降低 equalizer 的工作频率)。采用扰码(scrambled)的协议如 SDH/SONET, SMPTE SDI 使用 8/16bit 的并行宽度, 采用 8B/10B 编码的协议如 PCIeExpress,GbE 使用 10bits/20bits 宽度。

一个 4:1 的串行器如图 xxx 所示。8:1 或 16:1 的串行器采用类似的实现。实现时, 为了降低均衡器的工作频率, 串行器会先把并行数据变为 2bits, 送给均衡器 equalizer 滤波, 最后一步再作 2:1 串行化, 本文后面部分都按 1bit 串行信号解释。

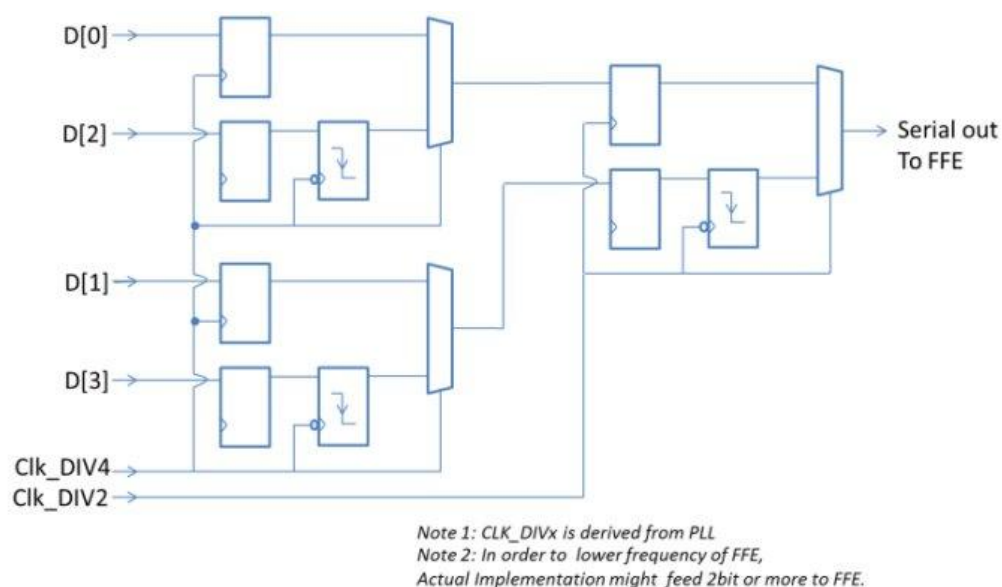


Figure 2.2 4:1 Serializer Demo

一个 1:4 的解串器如图 2.3 所示, 8:1 或 16:1 的解串器采用类似的实现。实现时, 为了降低均衡器(DFE based Equalizer)的工作频率, DFE 工作在 DDR 模式下, 解串器的输入是 2bit 或者更宽, 本文后面部分都按 1bit 串行信号解释。

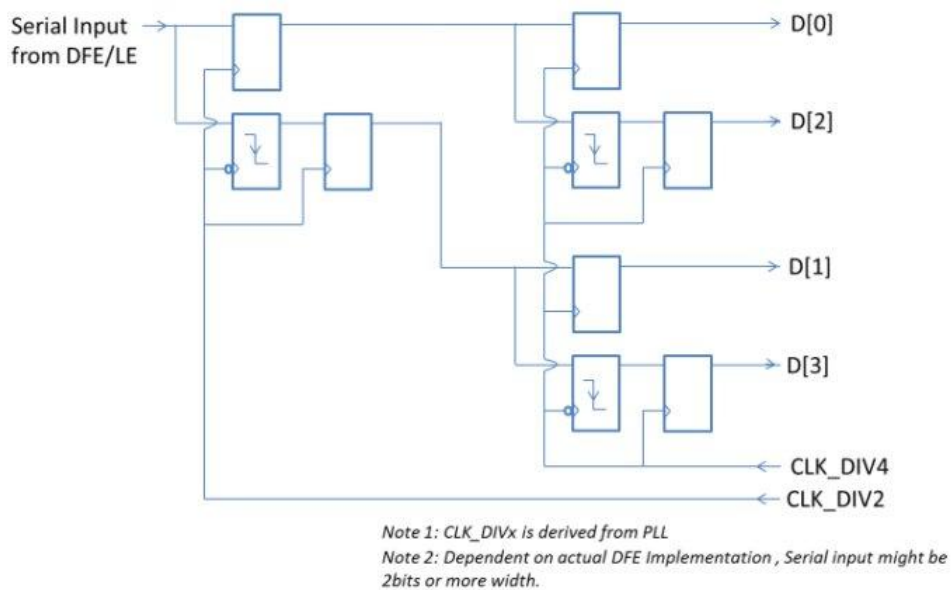


Figure 2.3 1:4 De-Serializer Demo

Serializer/Deserializer 的实现采用双沿(DDR)的工作方式，利用面积换速度的策略，降低了电路中高频电路的比例，从而降低了电路的噪声。

接收方向除了 Deserializer 之外，一般带有还有对齐功能逻辑(Aligner)。相对 SerDes 发送端，SerDes 接收端起始工作的时刻是任意的，接收器正确接收的第一个 bit 可能是发送并行数据的任意 bit 位置。因此需要对齐逻辑来判断从什么 bit 位置开始，以组成正确的并行数据。对齐逻辑通过在串行数据流中搜索特征码字(Alignment Code)来决定串并转换的起始位置。比如 8B/10B 编码的协议通常用 **K28.5**(正码 10' b1110000011, 负码 10' b0001111100)来作为对齐字。图 2.4 为一个对齐逻辑的演示。通过滑窗，逐 bit 比对，以找到对齐码(Align-Code)的位置，经过多次在相同的位置找到对齐码之后，状态机锁定位置并选择相应的位置输出对齐数据。

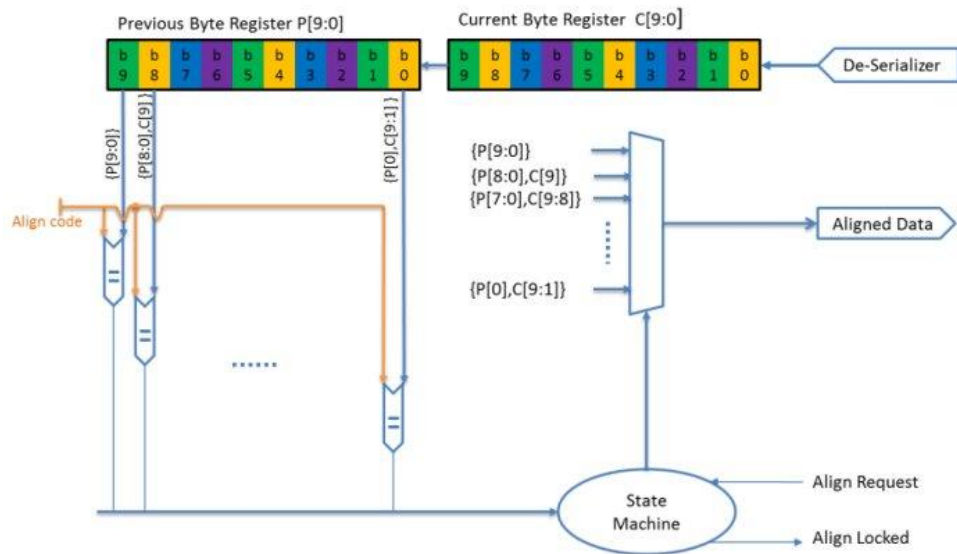


Figure 2.4 Aligner Implementation Demo

2.2 发送端均衡器 (Tx Equalizer)

SerDes 信号从发送芯片到达接收芯片所经过的路径称为信道(channel)，包括芯片封装，pcb 走线，过孔，电缆，连接器等元件。从频域看，信道可以简化为一个低通滤波器(LPF)模型，如果 SerDes 的速率大于信道(channel)的截止频率，就会一定程度上损伤(distort)信号。均衡器的作用就是补偿信道对信号的损伤。

发送端的均衡器采用 FFE(Feed forward equalizers)结构，发送端的 equalizer 也称作加重器(emphasis)。加重(Emphasis)分为去加重(de-emphasis)和预加重(pre-emphasis)。De-emphasis 降低差分信号的摆幅(swing)。Pre-emphasis 增加差分信号的摆幅。FPGA 大部分使用 de-emphasis 的方式，加重越强，信号的平均幅度会越小。

发送侧均衡器设计为一个高通滤波器(HPF),大致为信道频响 $H(f)$ 的反函数 $H^{-1}(f)$ ，FFE 的目标是让到达接收端的信号为一个干净的信号。FFE 的实现方式有很多，一个典型的例子如图 2.5 所示。

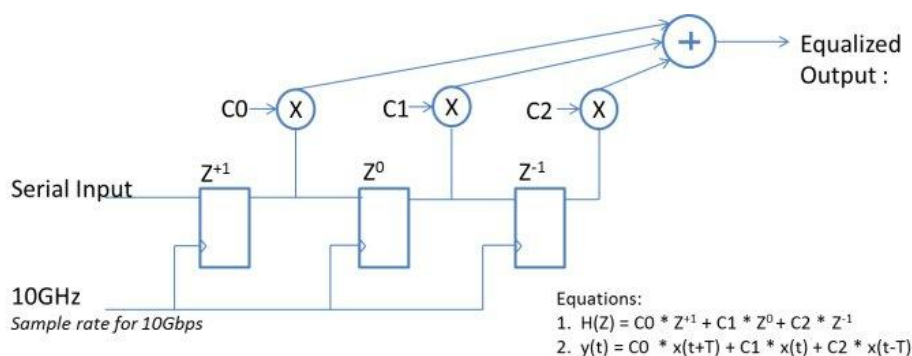


Figure 2.5 baud-spaced 3-Taps FFE

调节滤波器的系数可以改变滤波器的频响,以补偿不同的信道特性,一般可以动态配置。以 10Gbps 线速率为例,图 2.5 为 DFE 频率响应演示。可以看到,对于 $C0=0, C1=1.0, C2=-0.25$ 的配置,5GHz 处高频增益比低频区域高出 4dB,从而补偿信道对高频频谱的衰减。

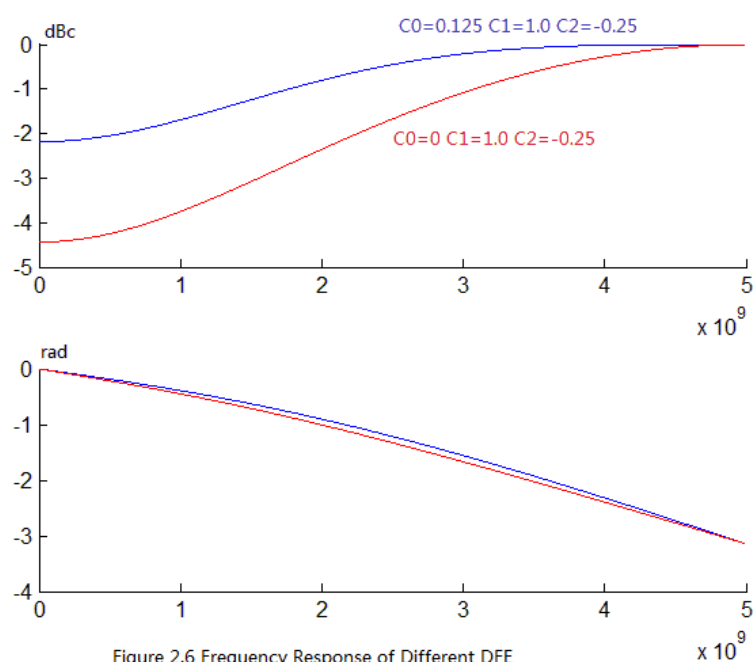


Figure 2.6 Frequency Response of Different DFE

采样时钟的频率限制了这种 FFE 最高只能补偿到 $F_s/2$ (例子中 $F_s/2=5\text{GHz}$)。根据采样定理,串行数据里的信息都包含在 5GHz 以内,从这个角度看也就足够了。如果要补偿 $F_s/2$ 以上的频率,就要求 FFE 高于 F_s 的工作时钟,或者连续时间域滤波器(Continuous Time FFE)。

图 2.7 为 DFE 时域滤波效果的演示,以 10Gbps 线速率为例,一个 UI=0.1 nS=100ps。演示的串行数据码流为二进制[00000000100001111011110000]。

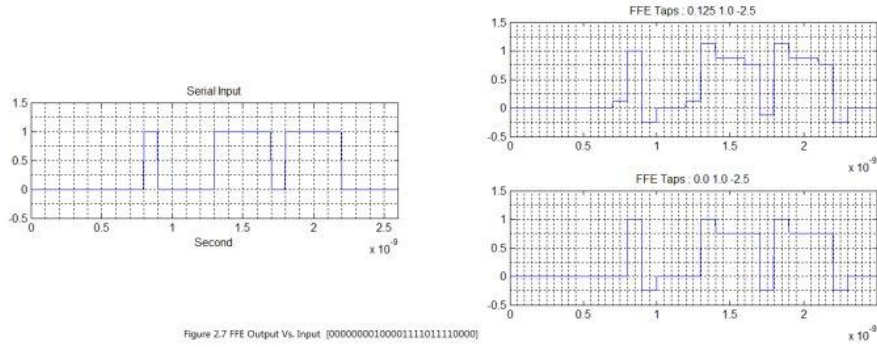


Figure 2.7 FFE Output Vs. Input [00000000100001111011110000]

2.3 接收端均衡器 (Rx Equalizer)

2.3.1 线形均衡器 (Linear Equalizer)

接收端均衡器的目标和发送均衡器是一致的。对于低速(<5Gbps)SerDes，通常采用连续时间域，线性均衡器实现如尖峰放大器(peaking amplifier)，均衡器对高频分量的增益大于对低频分量的增益。图 2.8 为一个线性均衡器的频域特性。通常工厂会对均衡特性封装为数种级别，可以动态设置，以适应不同的信道特性，如 High/Med/Low 等。

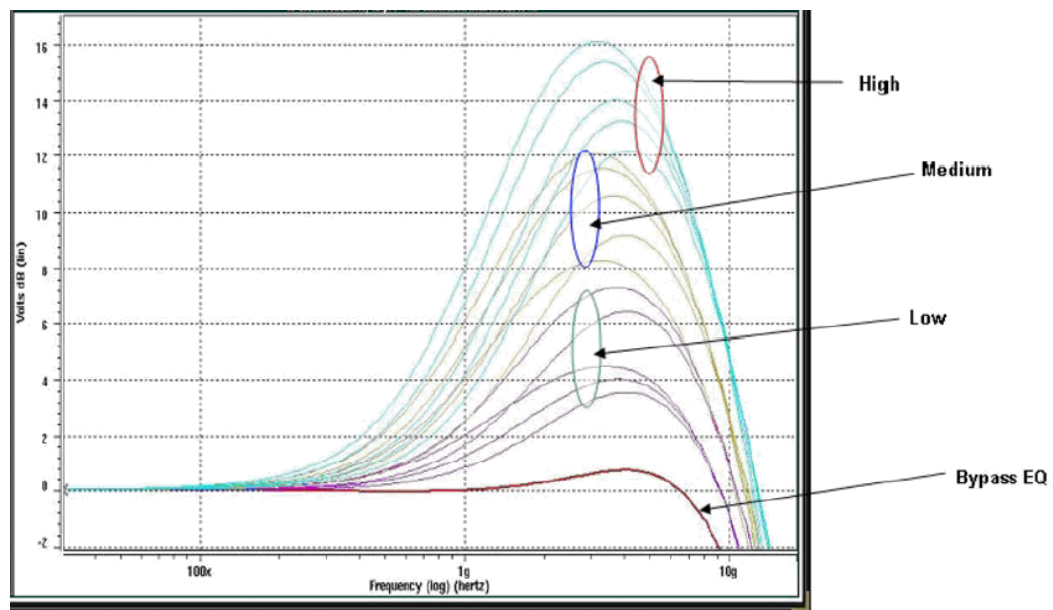


Figure 2.8 Frequency Response of A peaking Amplifier based Rx Equalizer

2.3.2 DFE 均衡器 (Decision Feedback Equalizer)

对于高速(>5Gbps)SerDes，由于信号的抖动(如 ISI 相关的确定性抖动)可能会超过或接近一个符号间隔(UI, Unit Interval)，单单使用线性均衡器不再适用。线性均衡器对噪声和信号一

起放大, 并没有改善 SNR 或者说 BER。对于高速 SerDes, 采用一种称作 DFE (Decision Feedback Equalizer 判决反馈均衡器) 的非线性均衡器。DFE 通过跟踪过去多个 UI 的数据(history bits)来预测当前 bit 的采样门限。DFE 只对信号放大, 不对噪声放大, 可以有效改善 SNR。

补充:

Unit Interval 单位时间间隔: 通常在通信信号的抖动测试中用来表示抖动幅度的单位。表示一个等步信号的两个相邻的有效瞬时之间的标称时间差

图 2.9 演示了一个典型的 5 阶 DFE。接收的串行数据由比较器(slicer)来判决 0 或者 1, 然后数据流由一个滤波器来预测码间干扰(ISI), 再从输入的原始信号中减掉码间干扰(ISI), 从而的到一个干净的信号。为了让 DFE 均衡器的电路工作在电路线形范围内, 串行信号先经过 VGA 自动控制进入 DFE 的信号幅度。

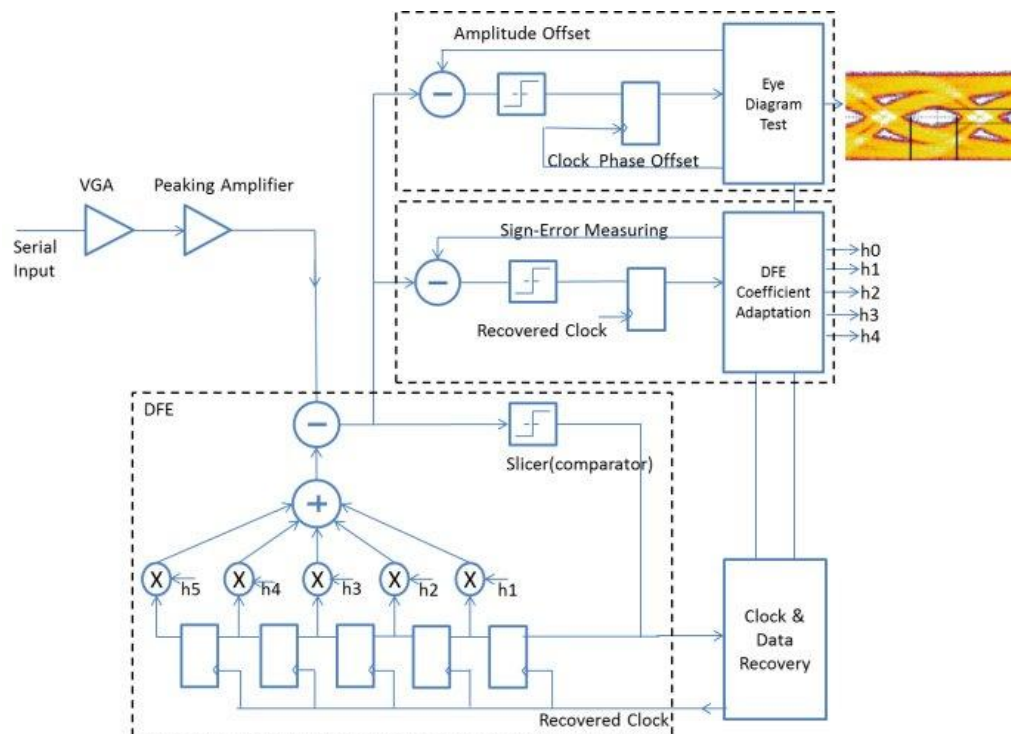


Figure 2.9 SerDes DFE Equalizer Structure, with Linear Equalizer & Eye-Test

为了理解 DFE 的工作原理, 先来看一个 10Gbps 背板的脉冲响应, 这个背板模型是 matlab 给出的一个基于实测的模型, 具有典型特性。

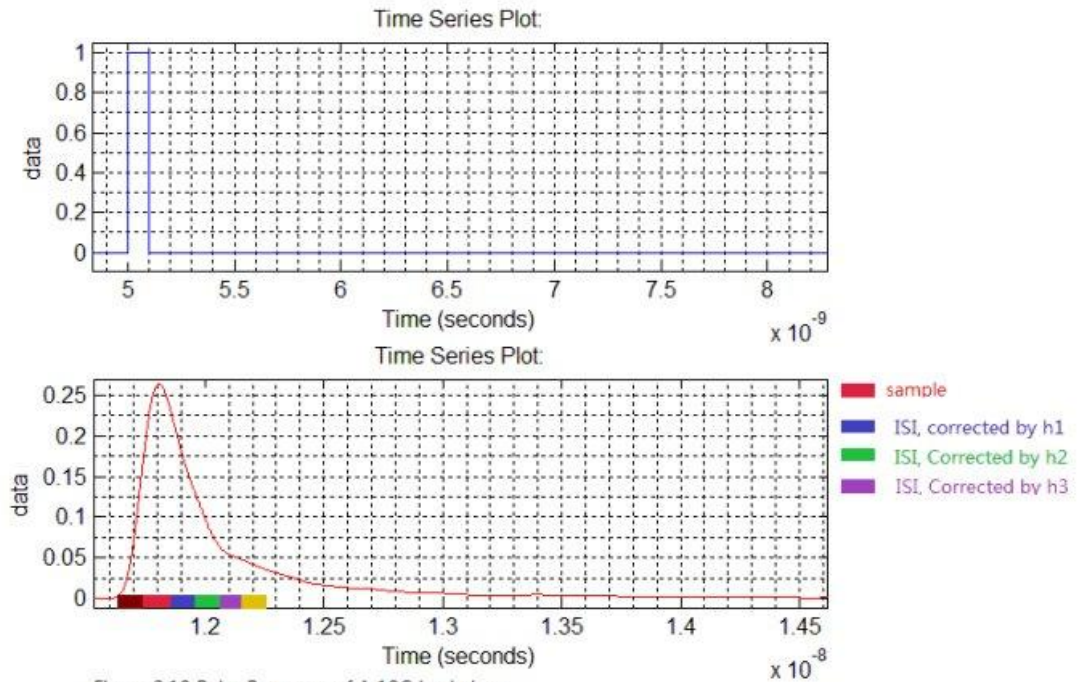


Figure 2.10 Pulse Response of A 10G backplane

图 2.10 中,一横格代表一个 UI 的时间。可以看出,一个 UI($0.1\text{ns} = 1/10\text{GHz}$)的脉冲信号,通过背板后,泄漏到前后多个相邻的 UI 里面,从而对其他 UI 的数据产生干扰。采样点后面的干扰叫做 **post-cursor** 干扰,采样点前面的叫做 **pre-cursor** 干扰。DFE 的第一个系数 h_1 (此例中 0.175)矫正第一个 **post-cursor**, 第二个系数 h_2 (此例中 0.075)矫正第二个 **post-cursor**。DFE 的阶数越多,能够校正的 **post-cursor** 也越多。

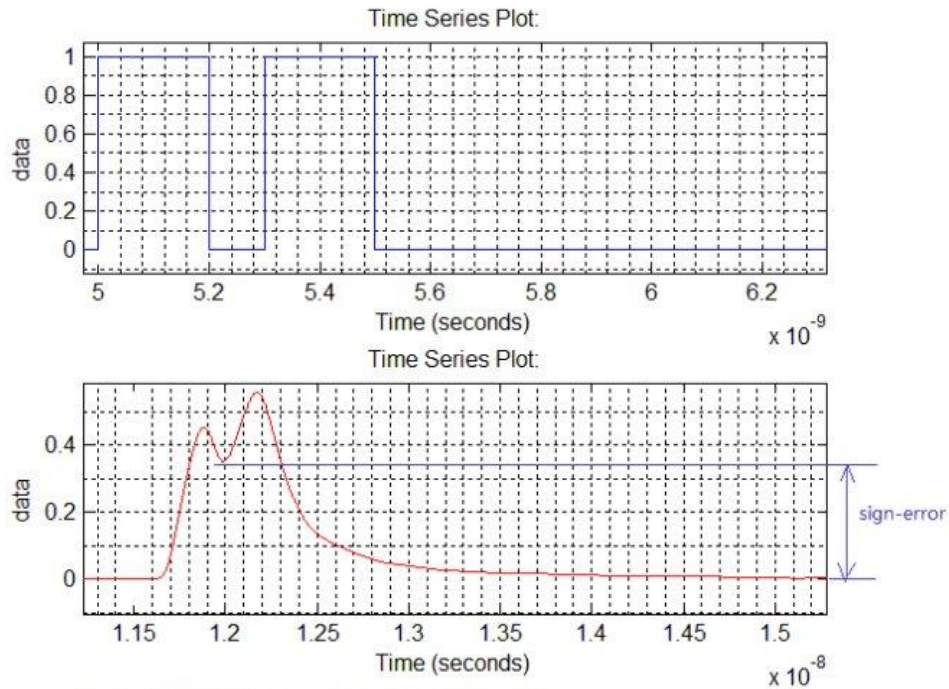


Figure 2.11 ISI of bitstream '11011' for a 10G backplane

用上述的背板传输一个 11011 的码流，由于 post-cursor 和 pre-cursor 的泄漏，如果没有均衡，将会导致'0'不能识别，见图 2.11。假定有一个 2 阶的 DFE，那么'0'bit 处的幅度应该减去第一个'1'bit 的 h_2 ，第二个'1'bit 的 h_1 ，得到 $0.35 - 0.075 - 0.175 = 0.1$ ，足够被识别为 0。

可见，DFE 计算历史 bits 的 post-cursor 干扰，在当前 bit 中把干扰减去，从而得到干净的信号。由于 DFE 只能能够校正 post-cursor ISI，所以 DFE 前面一般会带有 LE。只要 DFE 的系数接近信道(channel)的脉冲响应，就可以得到比较理想的结果。但是信道是一个时变的媒介，比如温度电压工艺的慢变化等因素会改变信道 channel 的特性。因此 DFE 的系数需要自适应算法，自动扑获和跟随信道的变化。DFE 系数自适应算法非常学术，每个厂商的算法都是保密的，不对外公布。对于 NRZ 码，典型的算法准则是基于 sign-error 驱动的算法。Sign-error 是均衡后信号的幅度和期望值的误差，算法以 sign-error 均方差最小为优化目标，逐次优化 $h_1/h_2/h_3...$ 。因为 sign-error 和采样位置是耦合在一起相互影响，因此也可以 sign-error 和眼图宽度两个准则为目标进行 DFE 系数的预测。也因此，采用 DFE 结构的 SerDes 通常都会带有内嵌眼图测试电路，如图 2.9 所示。眼图测试电路通过垂直方向上平移信号的幅度，水平方向上平移采样位置，计算每一个平移位置上的误码率 BER，从而得到每一个偏移位置与误码率关系的“眼图”，见图 2.12。

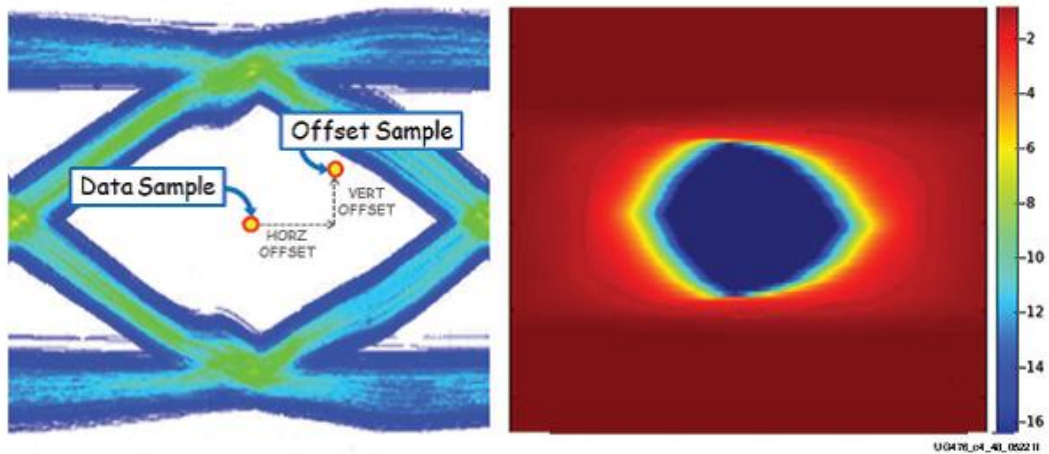


Figure 2.12 SerDes Embedded Eye-Diagram Test Function

2.4 时钟数据恢复(CDR)

CDR 的目标是找到最佳的采样时刻,这需要数据有丰富的跳变。CDR 有一个指标叫做最长连 0 或连 1 长度容忍(Max Run Length 或者 Consecutive Identical Digits)能力。如果数据长时间没有跳变,CDR 就无法得到精确的训练,CDR 采样时刻就会漂移,可能采到比真实数据更多的 1 或者 0。而且当数据重新恢复跳变的时,有可能出现错误的采样。比如有的 CDR 采用 PLL 实现,如果数据长时间停止跳变,PLL 的输出频率就会漂移。实际上, SerDes 上传输的数据要么利用加扰,要么利用编码的方法来保证 Max Run Length 在一定的范围内。

- 8B/10B 编码的方法可以保证 Max Run Length 不超过 5 个 UI。
- 64B/66B 编码的方法可以保证 Max Run Length 不超过 66 个 UI
- SONET/SDH 加扰得方法可以保证 Max Run Length 不超过 80 个 UI($BER < 10^{-12}$)

在点到点的连接中,大部分 SerDes 协议采用连续模式(continuous-mode),线路上数据流是持续而没有中断的。在点到多点的连接中,往往采用突发模式(burst-mode)如 PON。很显然 Burst-Mode 对 SerDes 锁定时间有苛刻的要求。

Continuous-Mode 的协议如 SONET/SDH 则要求容忍较长的连 0, 而且对 CDR 的抖动传输性能也有严格的要求(因为 loop timing)。

如果收(Rx)发(Tx)是异步模式(asynchronous mode),或者频谱扩展(SSC)应用中,则要求 CDR 有较宽的相位跟踪范围以跟踪 Rx/Tx 频率差。

根据应用场景的不同需求,CDR 的实现也有非常多种架构。FPGA SerDes 常常采用的基于数字 PLL 的 CDR, 和基于相位插值器的 CDR。这两种 CDR 在环路中采用数字滤波器,相对

模拟 charge pump 加模拟滤波器的结构更节省面积。

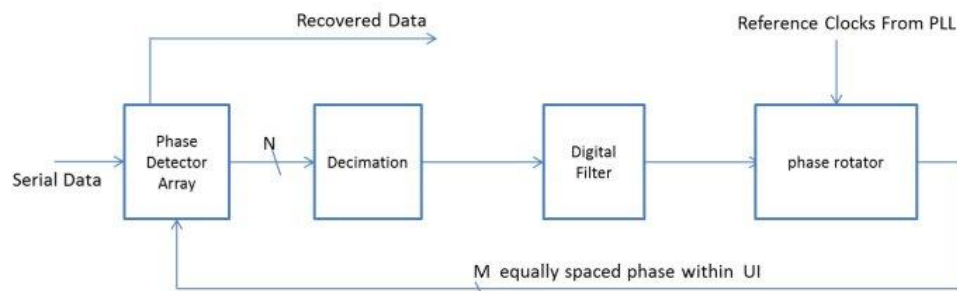


Figure 2.13 Phase Rotators based CDR

图 2.13 是基于相位插值器的 CDR。鉴相器阵列对输入的串行数据与 M 个等相位间隔的时钟在多个 UI 的跨度上进行相位比较, 得到多个 UI 跨度上的相位误差信号。相位误差信号的频率很高, 宽度也很宽, 经过抽取器降速并平滑后, 送给数字滤波器。数字滤波器的性能会影响环路的带宽, 稳定性, 反应速度等。经数字滤波器平滑后的误差信号送给相位插值器 (phase rotators) 修正时钟相位。最终环路锁定时, 理论上相位误差为零, 90 度偏移的时钟作为恢复时钟采样串行输入。

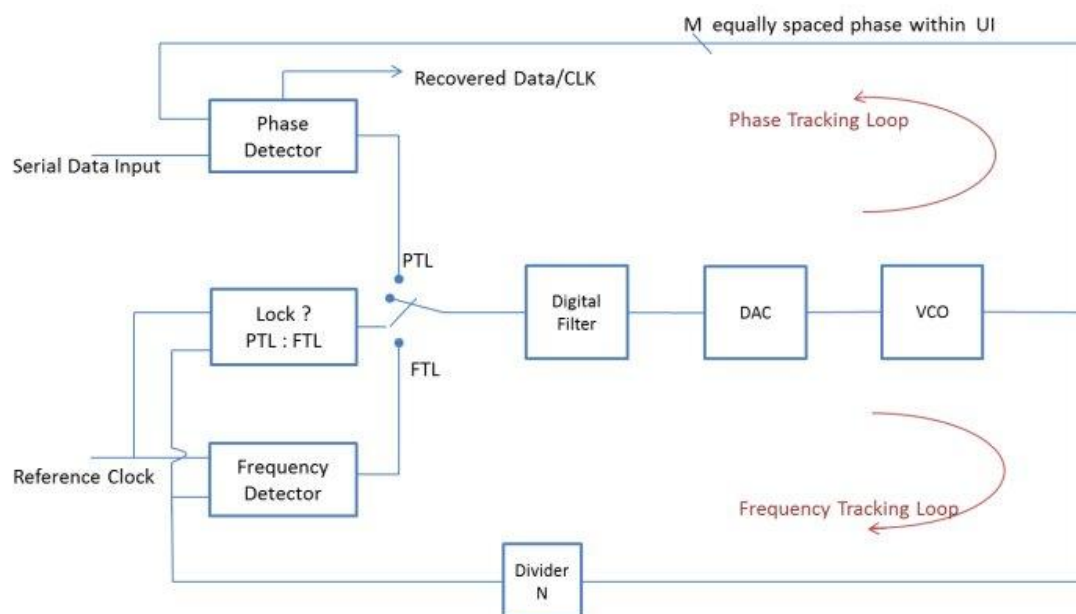


Figure 2.14 Digital PLL based CDR

图 2.14 是基于 DPLL 的 CDR, 分为两个环路, 对数据锁相的环路(phase tracking loop)和图 2.13 的 CDR 工作原理类似。鉴相器阵列对输入的串行数据与 M 个等相位间隔的时钟进行相位比较(也可能是在多个 UI 的跨度上), 得到相位误差信号。相位误差信号送给数字滤波器。

数字滤波器的性能会影响环路的带宽，稳定性，反应速度等。经数字滤波器平滑后的误差信号送给 VCO 修正时钟相位。最终环路锁定时，理论上相位误差为零，90 度偏移的时钟作为恢复时钟采样串行输入。

基于 DPLL 的 CDR 多了一个频率跟踪环路(Frequency Tracking Loop)。这是为了减小 CDR 的锁定时间，减少对环路滤波器的设计约束。只有当频率跟踪环路 锁定后，才会切换到数据相位跟踪环路。相位跟踪环路失锁时，再自动切换到频率跟踪环路。N 倍参考时钟 (Reference Clock)频率 和线路速率接近相等，因此两个环路的 VCO 稳态控制电压是接近相等的。借助 频率跟踪环路，减小了 相位跟踪环路 的捕获时间。

相位跟踪环路锁定时，频率跟踪环路不会影响相位环路。因此 SerDes 接收侧对参考时钟的抖动没有很高的要求。

基于相位插值器的 CDR 的参考时钟可以是收发公用的 PLL，也可以是每个通道独立的 PLL。这种结构的参考时钟抖动会直接影响恢复时钟的抖动以及接收误码率。

2.4.1 鉴相器(PD)

鉴相器用来比较相位误差，相位误差以 UP 或者 DN 的信号表示, UP/DN 持续的时间正比于相位误差。一个 bang-bang 结构鉴相器的例子如图 2.15。例子中只用了四个相位的恢复时钟作为例子。

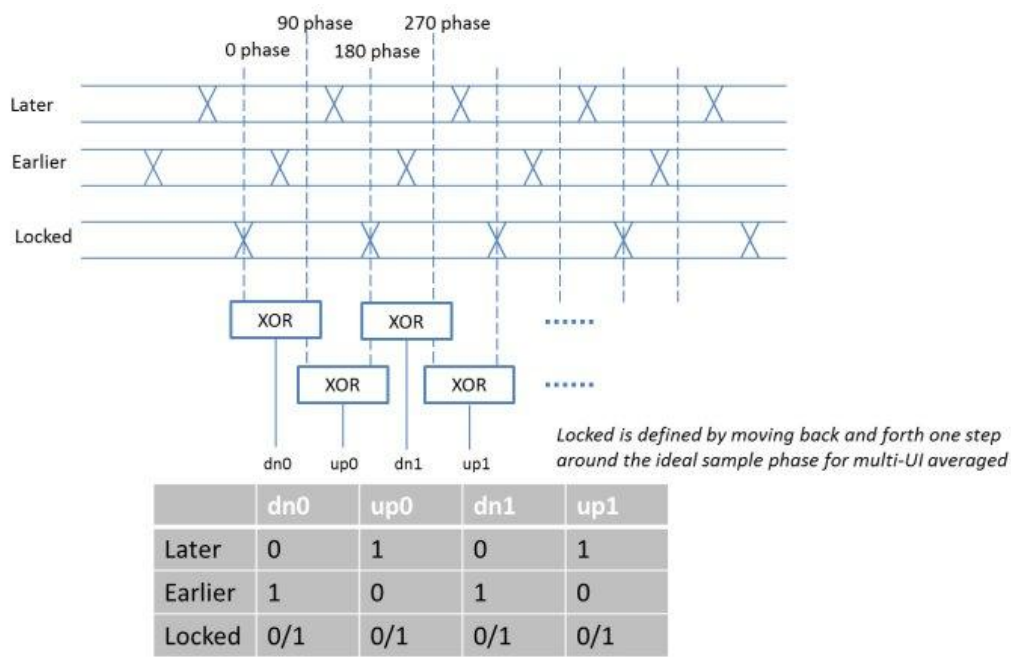


Figure 2.15 Bang-Bang Phase Detector

2.4.2 抽取器和滤波器

抽取器是为了让滤波器在较低的频率下工作。抽取的步长，平滑的方法都会影响环路性能。数字滤波器有比例分支(Proportion)和积分分支(Integral)构成，分别跟踪相位误差和频率误差。另外数字滤波器的处理延时也不能太大，如果处理延时过大，就会导致环路不能跟踪相位和频率的快速变化，导致误码。

CDR 的结构不限于以上两种，还有其他很多变种。基本上都是一个锁相环路。环路的跟随性能,稳定性(STABILITY)，带宽(bandwidth)/增益(gain)性能分析是一个非常学术的问题，用小信号线形模型分析,有非常多的书籍和资料解释了环路的量化性能。CDR 环路有一些的特点总结如下：

2.4.3 环路带宽

1.频率低于环路带宽的相位抖动会透过 CDR 转移到恢复时钟上。换句话说，频率低于环路带宽的抖动可以被 CDR 跟踪，不会引起误码。高频的抖动分量根据抖动幅度的大小，可能会引起误码。

2.环路带宽越大，锁定时间越短，恢复时钟的抖动也越大。反之则锁定时间越长，恢复时钟的抖动也越小。作为 CDR，我们希望环路带宽大一点，这样可以有更大的抖动容忍能力，但是对于 loop timing 的应用如 SONET/SDH 对恢复时钟的抖动有限制，又不能太大。

3. 开关电源的开关频率一般小于环路带宽，可以被 CDR 跟踪。但是，一方面开关电源耦合到 VCO(Digital to Multi-Phase Convertor)上的噪声不能被环路跟踪，低成本 Ring VCO 尤其对电源噪声敏感。另一方面开关电源的谐波可能超出环路带宽。

一些协议提供了 CDR 增益模板，如 SDH/SONET。兼容这些协议需要计算输入和输出的抖动预算。

2.5 公用锁相环(PLL)

SerDes 需要一个工作在数据波特率上的内部时钟，或者 $1/2$ 数据波特率的内部时钟，工作在 DDR 模式。片外提供给 SerDes 的参考时钟频率远远低于数据波特率，PLL 用来倍频产生内部高频时钟。FPGA 的 SerDes PLL 一般有 8x,16x,10x,20x,40x 模式，以支持常用的 SerDes 接口协议。比如 PCIeExpress 工作在 5Gbps，在 40x 模式下需要提供 125MHz 的片外参考时钟，20x 模式下需要提供 250MHz 的片外参考时钟。

一个三阶 PLL 电路如图 2.17，输入信号的相位和 VCO 反馈信号的相位由鉴相器比较，

相位误差有 charge pump 转化为电压或电流信号，经过 Loop Filter 平滑后产生控制电压，修正 VCO 的相位，最终使相位误差趋于零。

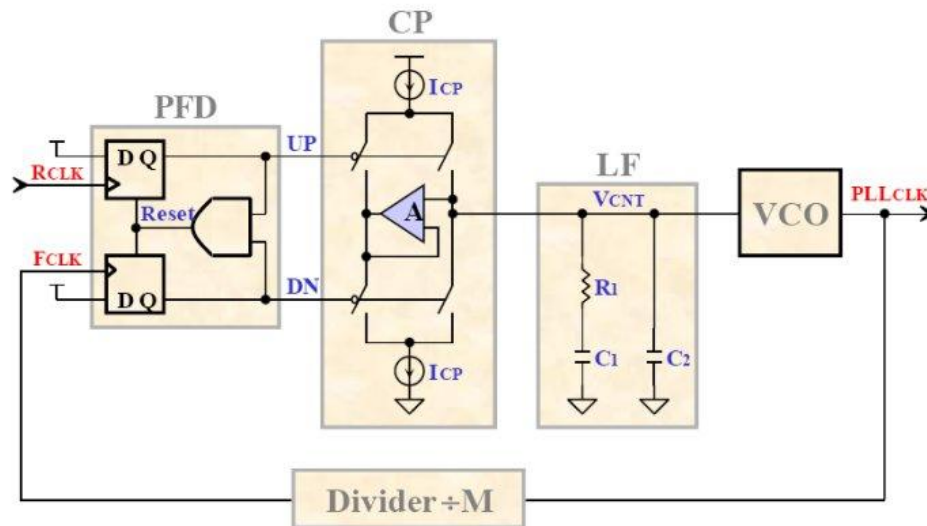


Figure 2.17 A 3-order Type II PLL

PLL 的工作过程分为入锁过程和跟踪过程。在入锁过程，环路的模型可以用一个非线性微分方程表示，可以评估捕获时间，捕获带宽等指标。入锁后，在小信号范围内，PLL 的模型是一个常系数线性方程，可以在拉普拉斯变换域研究 PLL 的带宽，增益，稳定性等性能，图 2.18 是小信号数学模型。

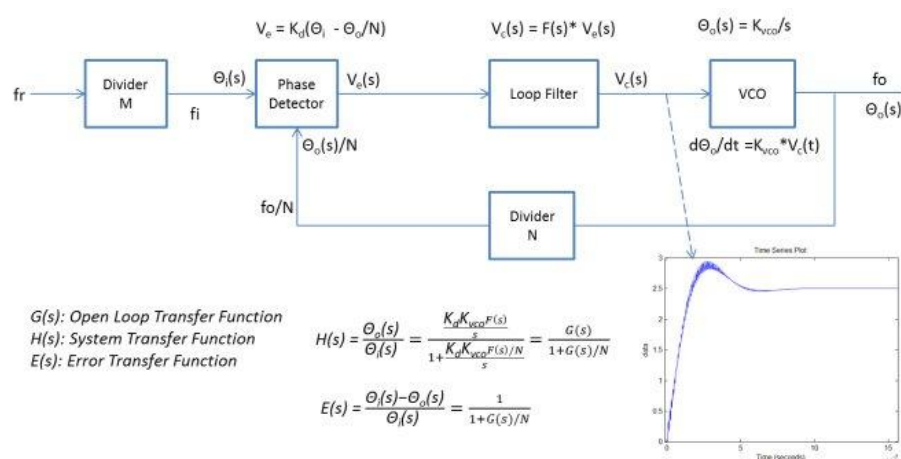


Figure 2.18 PLL small signal Model by Laplace transform

PLL 以传输函数极点(分母的根)个数命名环路的阶数。VCO 对相位有积分作用(K_{vco}/s)，因此不带滤波器的环路称为一阶环。带一阶滤波器的环路称为二阶环。一阶环和二阶环是无条

件的稳定系统。然而高阶环路有更多的极点和零点可以独立的调整带款，增益，稳定性，捕获带，捕捉时间等性能。

PLL 的频域传输函数特性主要有环路滤波器 $F(s)|_{s=j\omega}$ 决定，一个通用的 PLL 频域传输曲线如图 2.19 所示。有两个重要特征，环路带款和 jitter peaking。过大的 peaking 会放大 jitter，大的阻尼系数(damping factor)可以限制 peaking，但是会增加环路的如锁时间，影响滚降的速度和固有频率(natural frequency)。

➤ 当环路锁定后，固定相位差：

Kdc 为环路的直流开环增益， $\Delta\omega$ 为 VCO 中心频率和受控频率的差。对于 charge pump + passive filter 结构的 PLL 相位误差为零。

➤ 当环路锁定后，只有固定相位差，两个输入信号频率相等。

$$f_r/M = f_o/N$$

对于输入端的噪声，环路是一个低通滤波器，可以抑制高于环路截止频率的噪声或干扰。作为 SerDes 的 PLL，希望带宽的小一些，以抑制参考时钟上的干扰和噪声。

对于 VCO 噪声，环路是一个高通滤波器的作用。只有低于环路截止频率的 VCO 噪声得到了抑制。过量的 VCO 高频噪声会恶化时钟的抖动。低速 SerDes(<5Gbps)的 VCO 出于成本考虑采用 Ring 结构的 VCO，噪声大且对电源敏感。高速 SerDes 的 VCO 采用噪声小较小的 LC 结构 VCO

三、抖动和信号集成(Jitter, SI)

抖动是指信号的跳边时刻偏离其理想(ideal)或者预定(expected)时刻的现象。噪声，非理想的信道，非理想的电路都是产生抖动的原因。

3.1 时钟的抖动(clock jitter)

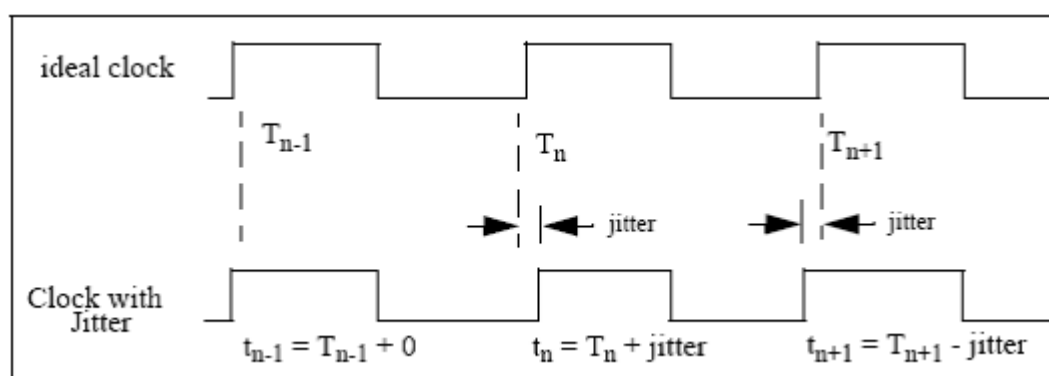


Figure 3.1 Clock Jitter

对于时钟信号，根据应用场景的不同，对抖动的定义也不一样。比如数字逻辑计算时序余量的时候，关心的是周期抖动。而时钟设计人员更喜欢相位抖动，因为可以利用频谱评估相位抖动，并可以用频谱来评估具体的干扰对总相位抖动的贡献。

参考图 3.1,介绍一下几种抖动的定义。

- 相位抖动(phase jitter)

$J_{\text{phase}}(n) = t_n - n \cdot T$ 。理想时钟的每个周期 T 都是相等的，没有抖动。真实时钟的跳边沿相对于理想时钟的偏离称作相位抖动。

- 周期抖动(period jitter)

$J_{\text{period}}(n) = (t_n - t_{n-1}) - T$ 。周期抖动是实际时钟的周期相对于理想周期的偏离(deviation)。显然 $J_{\text{period}}(n) = J_{\text{phase}}(n) - J_{\text{phase}}(n-1)$ 。

- l Cycle-to-Cycle jitter

$J_{\text{cycle}}(n) = (t_n - t_{n-1}) - (t_{n-1} - t_{n-2})$ 。前后相邻的两个周期的偏差是 Cycle-Cycle 抖动。显然 $J_{\text{cycle}}(n) = J_{\text{period}}(n) - J_{\text{period}}(n-1)$ 。

假设相位抖动的最大值为 $\pm J_p$ ，而且抖动的频率 $f_{\text{jitter}} = 0.5f_{\text{clock}} = 0.5/T$ ，也就是，

t_{n-2} 时刻的相位抖动为最大值 $+J_p$ ， t_{n-1} 时刻的相位抖动为最小值 $-J_p$

t_n 时刻的相位抖动为最大值 $+J_p$ ， t_{n+1} 时刻的相位抖动为最小值 $-J_p$

那么，周期抖动最大值 $J_{\text{period}} = \pm 2 \cdot J_p$

那么，Cycle-Cycle 抖动最大值 $J_{\text{cycle}} = \pm 4 \cdot J_p$

3.2. 数据的抖动(data jitter)

在高速 SerDes 领域每个人都在说抖动，因为抖动直接和误码率(BER)相关。

SerDes 发送端的一个重要要求是抖动(jitter generation)----针对特定的码型(pattern)，速率和负载情况下，发送端所生成的抖动。

信号经过信道(channel)到达接收端时，又会进一步放大抖动，不同的码型(pattern)包含的频率成分也不一样，信道对不同频率成分的传输延时也不一样(非线性相位)，产生和数据 pattern 相关的确定性抖动。阻抗不连续产生的反射，相邻信号的串扰和噪声都会引起数据抖动。

SerDes 接收端的一个重要指标是抖动容忍能力(Jitter Tolerance)----针对特定的码型和

误码率要求($BER < 10^{-12}$), SerDes 接收端能够容忍的抖动大小。对抖动评估时,会使用眼图(eye-diagram),浴缸曲线(bath curve),抖动分布柱状图(PDF),抖动频谱(jitter spectrum)等图形手段。

有一点需要说明,在谈论高速 SerDes 的数据抖动时(T_j, R_j, D_j etc.),是不包括低频率抖动的。这是因为低频率的抖动被认为是一种 wander,可以被 CDR 跟踪,不会引起误码。在用示波器(SDA)测量数据抖动的时候,可以设置示波器内嵌的 CDR 环路带宽,示波器测量的抖动数据已经滤掉了低频抖动。

根据抖动产生的原因和概率密度函数,常常将抖动分为几类。对抖动进行分类的意义在于某些类型的抖动可以被校正,而其他类型的不能被校正。经典的,总抖动 T_j (Total Jitter)被分类为确定性抖动 D_j (deterministic jitter)和随机抖动 R_j (random jitter)。抖动以 UI 或者 ps 为单位,可以是均方根值,或者峰峰值。

3.2.1 D_j

D_j 被进一步细分:

DCD(Duty cycle distortion)

占空比失真抖动。差分信号的正端负端的偏置电压不一致,或者上升沿和下降沿时间不一致会导致占空比失真。因为 DCD 和数据 pattern 相关,是可以被校正的抖动。

DDJ(Data dependent jitter)

数据码型相关的抖动,也称 码间干扰 ISI(intersymbol interference)。DDJ 是由于不理想的信道导致。是可以被均衡器校正的抖动。

Pj(Periodic jitter)

周期性抖动。 P_j 由电路上周期性干扰源导致。比如开关电源的开关频率,时钟信号的串扰等。虽然电源的开关频率一般在 CDR 的跟踪范围内,但是低次谐波成分可能会落在环路带宽外,或者 jitter peaking 区域,更重要的是电源谐波对 CDR 内 VCO 的干扰是不能被抑制和跟踪的,所以对于基于 Ring VCO 的 CDR 一定要尽可能的使用 LDO 供电。 P_j 不能被均衡器校正。

BUJ(Bounded uncorrelated jitter)

BUJ 由非时钟的干扰源引起。如果干扰源 aggressor 和 victim 是异步的,抖动的概率分布为有界的高斯分布,此时也称作 CBGJ(Correlated Bounded Gaussian Jitter)。BUJ/CBGJ 不能被校正。

3.2.2 Rj

Rj 有半导体本身的噪声引起，一个重要特征就是 Rj 的概率密度函数是高斯分布的，没有边界，且和数据 pattern 无关。只有在一定误码率约束下，才可以被认为是有限的。

3.2.3 Tj

数学意义上，抖动的概率分布函数可以近视为高斯分布和双底拉克分布的卷积。

对高斯分布有贡献的抖动为：

n Rj 为高斯分布

n 大量 Pj 叠加的效果也为高斯分布

n 部分 BUJ 也为高斯分布

对双底拉克分布有贡献的抖动为：

n DCD 被近视为双底拉克概率分布

高斯分布和双底拉克分布的卷积：

$$TJ(\tau, W, \sigma) = \frac{1}{2\sqrt{2\pi}} \cdot \frac{1}{\sigma} \cdot \left[e^{-\frac{\left(\tau - \frac{W}{2}\right)^2}{2\sigma^2}} + e^{-\frac{\left(\tau + \frac{W}{2}\right)^2}{2\sigma^2}} \right]$$

其中，W 被认为是确定性抖动的峰峰值，σ 是高斯分布的均方差。见图 3.2，可以看到，随着确定性抖动 W 的增加，概率密度分布曲线的顶部出现了双峰。一般来讲顶部曲线反映了确定性抖动的大小程度。

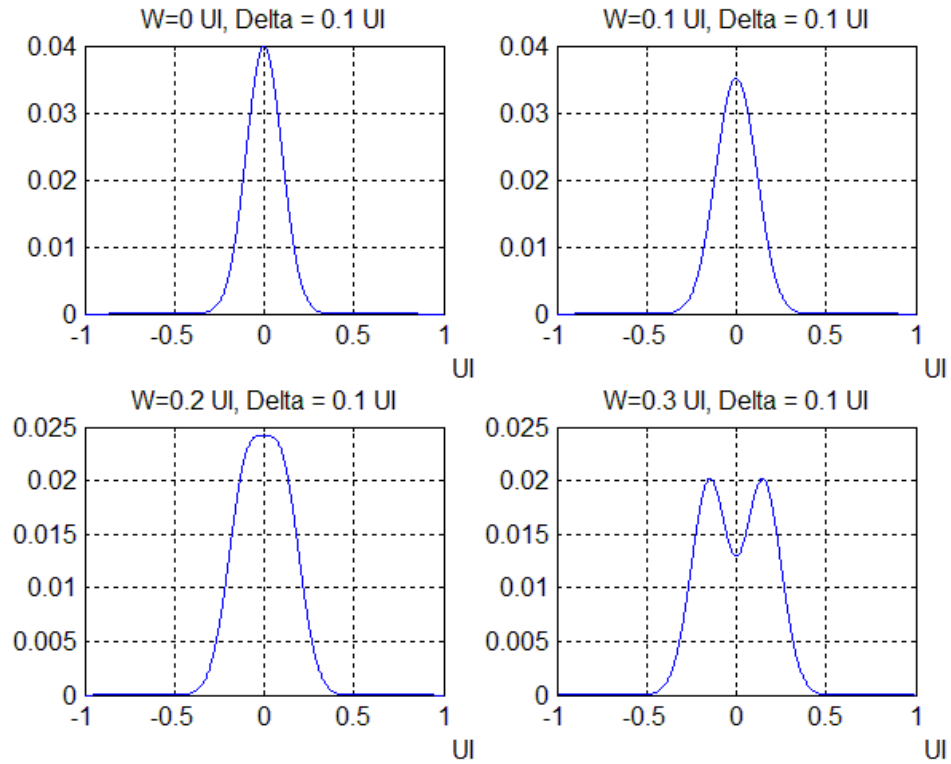


Figure 3.2 PDF of T_j with different D_j and R_j

把一个 UI 内的两个跳变沿处(0 UI 处和 1UI 处)的概率分布函数放在一张图中, 就是抖动的浴缸曲线(*bathtub curve*)。因为对数的宽动态范围, Y 坐标以对数显示。图 3.3 为确定性抖动 $W=0.05UI$, 高斯抖动方差 $0.05UI$ 的浴缸曲线。

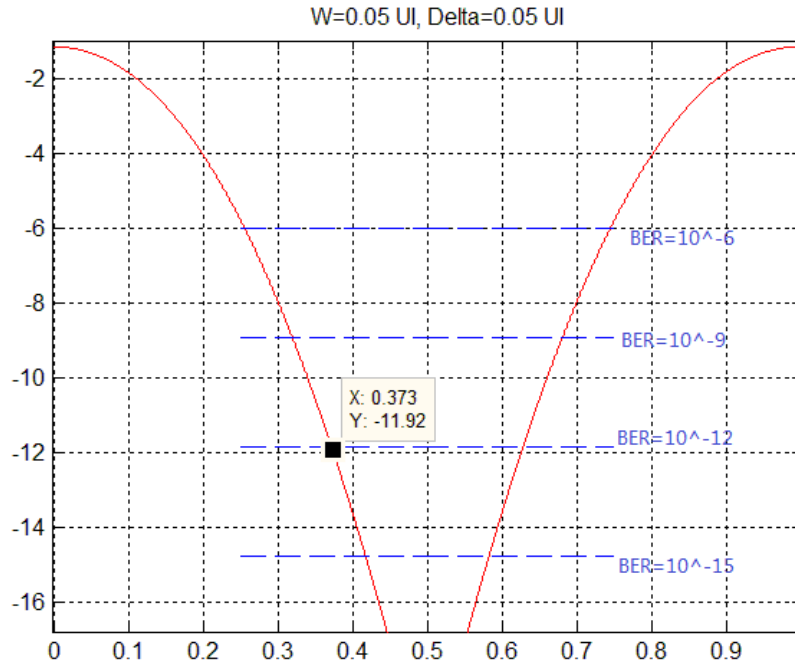


Figure 3.3 Bathtub Curve of Tj with 0.05 Dj peak and 0.05 Rj RMS

浴缸曲线还会标出对应的误码率 BER 坐标,比如图中 $BER=10^{-12}$ 的峰峰值抖动为 $T_j(p-p)=0.373 \times 2 = 0.746 \text{ UI}$ 。曲线下面的面积占总面积的比率就是误码率。比如图中,

$$T_j @ 10^{-12} = \frac{\int_{0.373 \text{ UI}}^{0.627 \text{ UI}} \text{Jitter}(t)}{\int_0^1 \text{Jitter}(t)}$$

浴缸曲线的顶部主要为确定性抖动 Dj 的贡献,越靠近底部,高斯抖动的贡献越大,并以高斯曲线的斜率衰减,也因此常利用高斯分布的特性进行估算。下表为高斯分布

$$BER = CDF = \frac{\int_{-\infty}^{Q \cdot \delta} \text{Jitter}(t)}{\int_{-\infty}^{+\infty} \text{Jitter}(t)}$$

和均方差关系。

BER	Q	BER	Q
10^{-3}	3.09	10^{-10}	6.36
10^{-4}	3.72	10^{-11}	6.71
10^{-5}	4.27	10^{-12}	7.03
10^{-6}	4.75	10^{-13}	7.35
10^{-7}	5.20	10^{-14}	7.65
10^{-8}	5.61	10^{-15}	7.94
10^{-9}	6.00	10^{-16}	8.22

在规定的 BER 内，利用该表可以快速估算均方差值和峰峰值之间的关系。比如高斯抖动的均方根 $0.05UI$ ，误码率要求为 10^{-12} BER，查表可知 $Q=7$ ，那么高斯抖动的峰峰值要求是 $0.05UI \times 7 \times 2 = 0.7UI$ 。

如前所述， $W=0.05UI, R_j=0.05UI$ 计算出的总抖动 $T_j=0.746UI$ ；

利用高斯特性估算的高斯抖动为 $0.7UI$ 。

如果按 $T_j = R_j(0.7UI) + D_j(0.05UI)$ 计算得到 $0.75U$ ，基本是一致的，差异是因为画图程序有量化误差。

四、信号集成(SI)及仿真

4.1 信道 channel

SerDes 信道关注的频率范围是 $0Hz$ 到奈奎斯特频率，也就是 2 倍的信号基频。信号的基频是信号线速率的一半，也就是说信号的奈奎斯特频率就是线速率。信道对信号的损伤包括插入损失(insertion loss), 反射 (reflection), 串扰(crosstalk)等。这些损伤可以用 S-parameter 信道模型来表达。S-parameter 可以有矢量网络分析仪测量(Vector Network Analyzer)得到。信道不是一个纯阻性网络，还包括容性和感性。这样对不同频率里成分的时延也不一样，从而产生和 data pattern 相关的抖动。

信道上的每一个不连续阻抗点都会产生反射，根据反位置的不同，反射信号会以不同的相位叠加在原始信号上，增加或者减小信号的幅度。

SerDes 信号为差分形式，对共模干扰有较强的抑制。如果在 \pm 端上的干扰有差异，就会引入串扰。通常外部 PCB 可以保证 SerDes 数据和干扰源保持足够的距离，但是芯片内部由于考虑到经济性，很难保证 SerDes 信号和干扰源足够的隔离距离，尤其是一个通道自己的发送信号干扰自己的接收信号。

4.2 芯片封装 Package

封装 package 也是信道的一部分。芯片外部的信道可以通过 VNA 测量的到，而封装的 S-parameter 通常有芯片制造商提供，仿真时可以把两者级联起来。封装 package 由于距离较短，insertion loss 通常不是主要问题，主要考虑的是阻抗匹配问题。

4.3 SI 仿真

信号集成性(SI)仿真可以通过把 SerDes 发送端 SPICE 模型, 封装和信道的 S-parameter 模型, 接收端 SPICE 模型级联起来搭建仿真平台, 利用仿真工具对不同的激励在不同的测试条件作电路仿真。通过测量 SerDes 接收端的眼图来评估是否满足设计需求。也可以通过实测接收端眼图, 检验是否满足接收端的眼图模板, 或者协议规定的眼图模板。图 4.1 为一个实测的 3.125Gbps 信号的眼图及模板, 同时也包含了浴缸曲线和统计图。

Figure 4.1 Rx-end Eye-diagram of A 3.125Gbps SerDes

对于高速 SerDes(>5Gbps), 这种传统的电路仿真方法已经不能满足设计的需求。首先, 过量的码间干扰 ISI 导致接收端眼图完全闭合, 但是通过芯片内的 DFE 均衡后, 眼图可能是很好的。其次, 电路仿真(SPICE)的速度非常慢, 即使是有办法把 DFE 均衡加入仿真, 由于 DFE 仿真需要足够长时间的 bits 来训练, 此时, 电路仿真的仿真时间是不可接受的。

对于高速 SerDes 的仿真需要借助统计分析 (statistical analysis) 的方法。统计分析的方法把发送端-信道-接收端的连接视为线性系统, 计算系统脉冲响应 $h(t)$, 加入噪声源来模拟抖动, 然后用激励对脉冲响应进行卷积, 得到接收端的信号, 这种方法可以把厂家私有的 FFE, DFE 自适应算法加入仿真。

统计分析 (statistical analysis) 方法不能仿真电路的非线性和时变特性, 所以高速 SerDes 往往要两者结合起来仿真 SI。更多关于统计分析 (statistical analysis) 方法可以参考。

五、 结尾

有人讲过, 现在的汽车是如此复杂, 以至于虽然每一个零件都有人懂, 但是合起来作为整车却没人能够全懂。近些年来, FPGA 变的越来越复杂, 对工程师的要求也越来越高。要成为一个合格的 FPGA 应用工程师, 不仅要擅长数字电路的设计, 还要懂得高速 SerDes, 信号集成 SI, DSP 算法, 多核 CPU, 嵌入式操作系统等。每一项技术背后都是一个专业领域, 一个人不会在每一个领域内都是专家, 只要比别人多学习一点, 关键时候就会突出你的价值。本篇主要介绍了 SerDes 的基本结构和用好 SerDes 需要掌握一些的知识, 希望对你的工作有所帮助