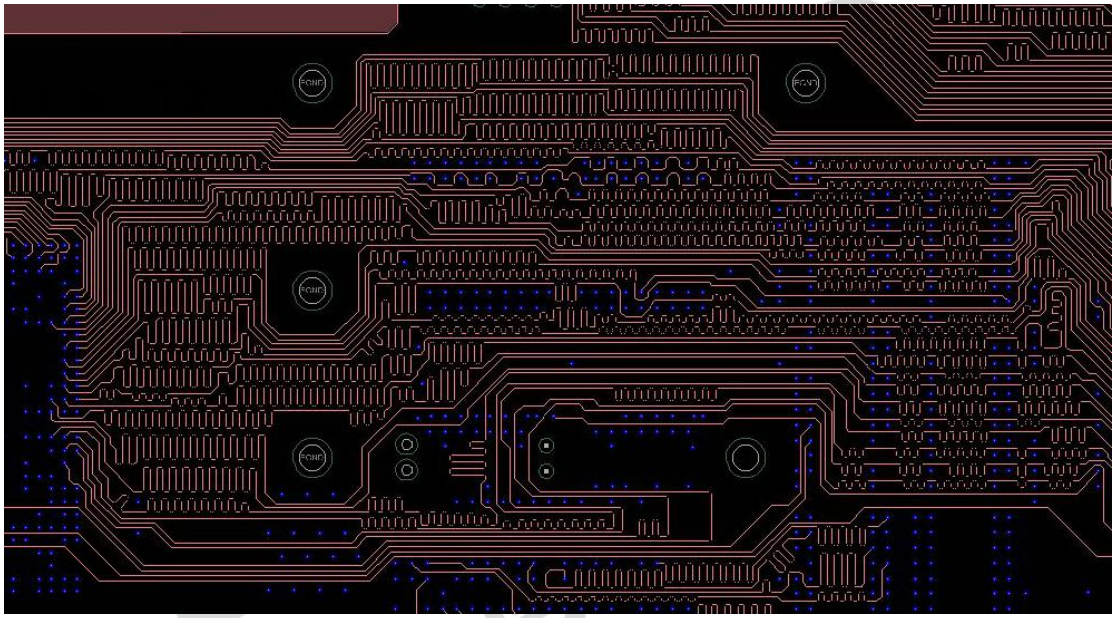
【高速先生原创|十大误区系列】PCB 设计十大误区-绕不完的等长（共 4 篇）

作者：吴均　　一博科技高速先生团队队长

1、关于等长

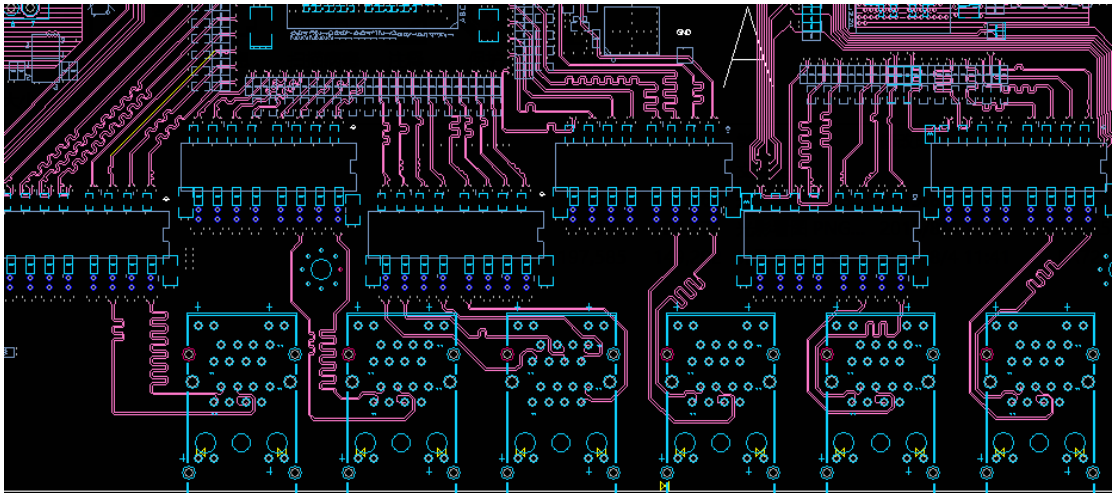
第一次听到“绕等长工程师”这个称号的时候，我和我的小伙伴们都惊呆了。每次在研讨会提起这个名词，很多人也都是会心一笑。



不知道从什么时候起，绕等长成了一种时尚，也成了 PCB 设计工程师心中挥不去的痛。需要等长设计的总线越来越多，等长的规则越来越严格。5mil 已经不能满足大家的目标了，精益求精的工程师们开始挑战 1mil，0.5mil……还听过 100%等长，没有误差的要求。

为什么我们这么喜欢等长？打开 PCB 设计文件，如果没有看到精心设计的等长线，大家心中第一反应应该是鄙视，居然连等长都没做。也有过在赛格买主板或者显卡的经验，拿起板子先看看电容的设计，然后再看看绕线，如果没有绕线或者绕线设计不美观，直接就 Pass 换另一个牌子。或许在我们的心中，等长做的好，是优秀 PCB 设计的一个体现。

做过一个非正规的统计（不过一博每年上万款 PCB 设计，我们的采样基本上也可以算做大数据了），稍微复杂一点的高速板子，绕等长要占据总设计时间的 20%~30%。如果等长规则更严格，或者流程控制不好，做了等长之后再反复修改，这个时间还会更多。



2、那些年，我们一起绕过的等长

培训的时候，我们经常会玩一个游戏，游戏的名字叫做：那些年，我们一起绕过的等长.....

说到等长要求，先说说什么是“裕量”哈。“裕量”是设计时保留的安全间距，百度百科的解释更简单：多出来一部分，就称之为裕量。到底要多多少，什么才是安全，那就见仁见智了。每个人的安全感不同，对“裕量”的定义就不一样。但是在时序设计的时候，有一个现象比较普遍，那就是裕量层层放大，比如产品经理可能要求等长范围是±100mil，项目经理可能就会更严格到±50mil，然后到具体的实施工程师，可能就变成±5mil 了。碰到一些“安全感”不足的工程师，那就恨不得是完全等长，没有偏差。

所以，后面的讨论里面，我们不会太多纠结在等长到底是 10mil 还是±5mil，我们集中精力来看看哪些等长是没有必要的，哪些等长反而破坏了系统的时序设计要求。

游戏开始，大家直接回复“高速先生”，列举下自己做过的，或者认可的等长设计要求，格式如下：

DDR3-1600，要求同组数据线与 DQS 等长范围是±5mil，地址\控制\命令信号与 CLK 等长±25mil ……

游戏规则，同种类的等长要求，第一个回答的 3 分，第二个回答的 2 分，第三个 1 分，所以想拿分数的抓紧时间哈。

并且只要回答出等长要求的即可拿分，不管回答是对是错，我们的游戏目的是搜集大家做过的等长设计要求，合理还是不合理，我们后面的文章来讨论。

时序设计这个话题会持续比较长时间，这篇文章先收集大家的观点，然后针对大家的观点来思考后续文章的构架，先谢谢大家的配合。

3、关于等长与等时

绕线系列的第一篇文章发完之后，就开始准备美国研讨会，然后就是长达一个月的出差。终于有时间继续这个话题了，先来看看之前大家的回复，我隐去了回复者的名字，只保留了答案：

游戏开始，大家直接回复“高速先生”，列举下自己做过的，或者认可的等长设计要求，之前的部分答复如下：

项目中的，eMMC data、clk、cmd要求等长控制在±25mil以内；fpdlink lvds 3.125G差分数据线等长控制在±10mil以内

千兆网口4对差分等长要求+-100mil

rgmii 同组信号控制在＋－25mil；千兆网口，四对信号控制在+-25mil;lvds信号，控制在＋-100mil;local bus控制在+-100mil。这些现在看起来貌似不都正确，话说当年我都绕过

pcie差分对对内等长误差2mil对间5mil

…… ……

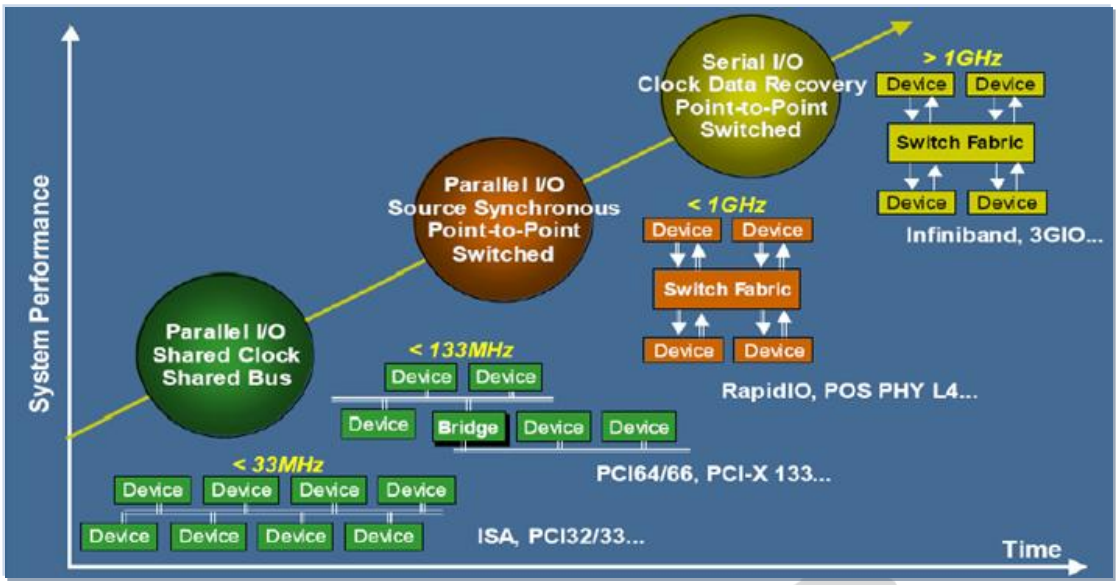
之前也提过，现在流行重要的事情说三遍：

- 等长从来都不是目的，系统要求的是等时……
- 除了差分对内的等时是为了相位之外，绝大多数的等时都是为了时序！
- 为了时序而绕线，就一定要搞通时序关系，看懂时序图

每次看到时序图的时候，都会眼前一黑有没有？

时序是大家非常头痛，也觉得非常复杂的话题，所以高速先生小陈在之前的时序话题中试图用两对恋人的恋爱关系来解释时序问题，绕口令式的比喻不知道有多少人真正看懂了？给我的感觉是 80、90 后们好像很快领悟了小陈的意思，而 70 后们普遍表示更晕了有没有。

想把时序问题简单讲清楚，是一个巨大的挑战，高速先生的精神就是迎难而上，前仆后继。我的目标是不给大家看复杂的时序图，也不引用什么比喻联想，让大家简单理解时序。

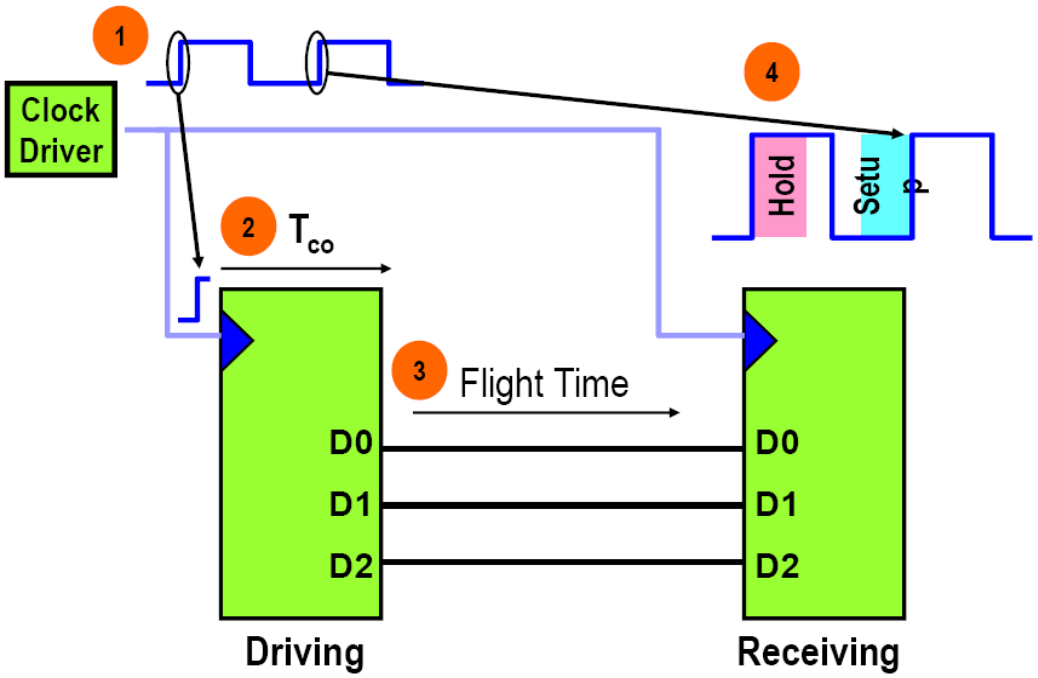


借用一张很好的图，一下子看懂主要的三种时序系统（这里忽略应用较少的内同步时钟系统）

并行总线包括了早期的共同时钟和现在流行的源同步时钟，然后就是串行总线。区分三种系统也很简单，后续文章分别与大家一一道来。

4、共同时钟时序

共同时钟的并行总线，十几年前的技术，跟不上高速设计的需求，但是现在还有一些应用，比如常见的 Local bus 基本是共同时钟总线。还有 CPCI 总线，PCIX 总线，早期的 SDRAM 等。判断是否共同时钟总线的主要特征是：外部时钟分配器（或者 FPGA）分别送出时钟线到发送与接收芯片。如下图所示，能找到外部同步时钟的，一定是共同时钟总线。



共同时钟总线的时序特点是，时钟的上一个边沿，发送芯片打出数据，然后在下个时钟边沿，接收芯片接收数据。为了简化后面的理解，假定时钟达到驱动端和接收端的时间一致，也就是时钟线等长（这也是最常规的设计思路）。

影响时序的因素有 Tco，Tskew，Tjitter，Tcrosstalk……，看起来很复杂，简单来说，只要满足两个条件，即可达成时序要求

1、 一个时钟周期之内，数据要完成从驱动端发出，到达接收端，并有足够的建立时间
2、 第二个数据来到之前，前一个数据要有足够的保持时间

满足条件 1，就要求 Tclk 能包容数据到达所需的是所有时间，这些时间包括了数据输出延时（Tco），数据飞行时间（Tflighttime），数据保持时间要求（Tsu），以及所有七七八八影响时序的因素（Tcrosstalk，Tjitter……），并且所有这些因素都要取最恶劣情况。

- **TPCB skew+Tclock skew +Tjitter +Tco data +Tflt data +Tsetup<Tcycle**

满足条件 2，就是下个数据最快会在最小数据输出延时（Tco min）加上最小数据飞行时间（Tflighttime min）之后达到，数据必须在下个数据达到之前有足够的保持时间。

- **Tco data +Tflt data +Tclock skew+Tpcb skew>Thold**

真正设计的时候，我们需要从器件手册查找相应的数据来进行时序计算。从理解角度来说，却不用那么复杂。

- $t_{Output\_delay\_max}$ = The CPU_DH[31:0] and CPU_DL[31:0] maximum delay is 2.8 ns (taken from Marvell's MV6446x hardware specification).
- t_{Input_setup} = For all IBM750GX signals, the value is: 1.1ns (taken from the IBM750GX datasheet).

According to the formula above, with a maximum clock skew of 0.3 ns:

$t_{Fly\_time} < t_{cycle} - t_{Output\_delay\_max} - t_{Input\_setup} - t_{Clock\_skew}$ = 5 ns - 2.8 ns - 1.1 ns - 0.3 ns

==> $t_{Fly\_time(data\_bus)} < 0.8$ ns

- $t_{Output\_delay\_max}$ = The CPU_DH[31:0] and CPU_DL[31:0] maximum delay is 2.3 ns (taken from the IBM750GX datasheet).
- t_{Input_setup} = The CPU_DH[31:0] and CPU_DL[31:0] setup is 1.7 ns (taken from Marvell's MV6446x hardware specification).

According to the formula above:

$t_{Fly\_time} < t_{cycle} - t_{Output\_delay\_max} - t_{Input\_setup} - t_{Clock\_skew}$ = 5 ns - 2.3 ns - 1.7 ns - 0.3 ns

==> $t_{Fly_time(data_bus)} < 0.7$ns

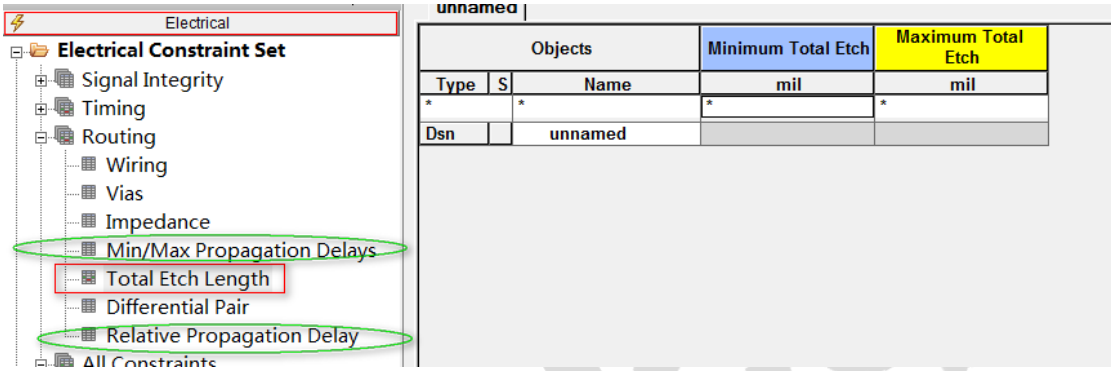
0.3ns<Tfly time<0.7ns

上图是一个实际案例计算后的结果，我们从中只要看懂两个事情：

一、共同时钟总线时序关系随着 TClk 的减小，难度急剧加大。33M、66M 的共同时钟总线，适度关注拓扑结构和端接来保证信号质量就够了，不需要任何绕线。100M 以上的共同时钟总线时序开始变得紧张，133M 以上的系统，建议一定要做时序分析，否则风险很大。

二、共同时钟总线时序是对总长的要求，不是等长，借用 Cadence Allegro 的规则管理器来解释，共同时钟总线最合适的电子规则是 Total Etch Length，而不是我们常用的各种 Propagation Delay。

注：100M 或者 133M 以上的时候，由于时序非常紧张，有可能通过控制外部时钟 Skew（布线或者调整寄存器）的方式来调节时序，这时对以上不等式两边进行调整的过程中会涉及到总长的差异问题。但是也还是用 Total Etch Length 的规则来约束设计，而不是 Propagation Delay



讲了这么多，大家继续晕菜有没有？想把时序讲简单，是不是 Mission Impossible？

还是简单总结一下：

1、共同时钟总线时序关系随着速率增加，时钟周期减小，设计难度增加
2、共同时钟总线时序是对总长的要求，一般情况下可以理解为尽量走短；没有等长要求。
3、如果因为时序调整的原因，需要绕线的时候，尽量保证长线不要绕的更长
4、100M 以上的共同时钟总线，建议进行时序计算，避免风险

5、源同步总线时序

上一篇文章不知道大家有没有看晕了，讲时序确实是吃力不讨好哈。看看上一篇文章大家的回复：

@南昌米粉-萝卜妈：最大还是受限于 Tco，一般 2 点几个 ns，速率越高时序越难满足，所以共同时钟就升级为源同步，信号时钟从同一个芯片发出。

@绝对零度：主要因素是时钟的串扰，数据的 Tco 难以减小。解决方法就是使用源同步时钟系统，和差分时钟。典型应用就是 DDR。

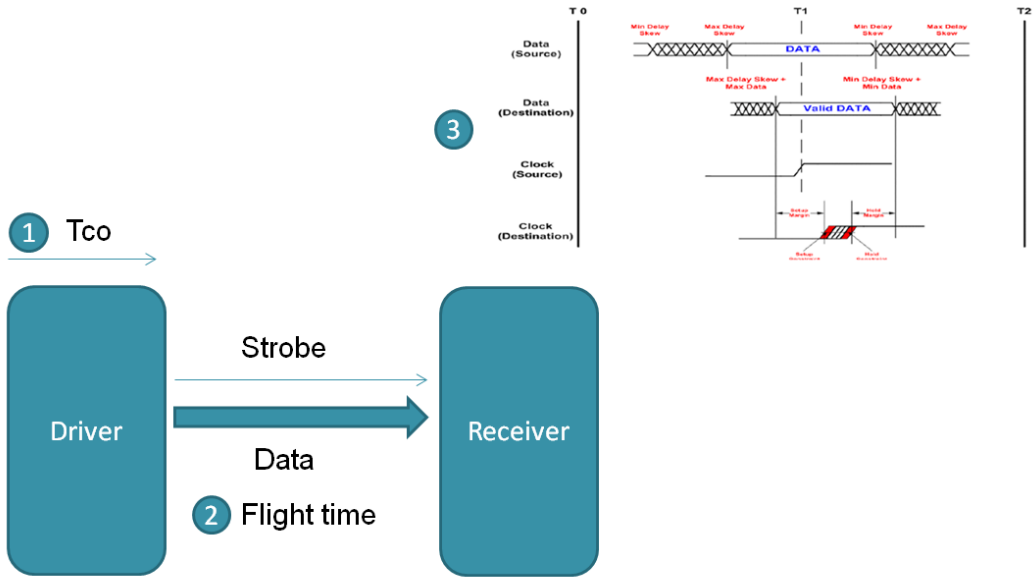
@山水江南：共同时钟总线的数据时长与时钟时长不匹配，还受时序偏差最大的通道影响，如：时钟偏差、数据偏差、Jitter、串扰等。使用源同步时钟，让时钟和每 Bit 数据一起发送，消除时钟和数据的偏差。

@Erick：随着频率的提高，共同时钟的限制因数有如下：时钟到达两个芯片的 clock skew，数据中各个 data 的 skew，以及 clock 和 data 之间的 skew 越来越难控制. 采用源同步时钟，可以有效解决 clock skew，而且 clock 和 data 采用组内差分走相同路径也可以解决 clock 和 data 之间 skew. 但是源同步受限与 clock 频率的提高来提升带宽，这样就要讲讲内嵌时钟的大 serdes 了。

看来大家还是有不少人说到了重点的：

1、 影响共同时钟时序很重要的一个因素是较大的 Tco，当然飞行时间也是一个问题。
2、 由于芯片工艺因素以及适当的保持时间需求，Tco 不能太小，通常都在 3ns 以上，甚至有的 Tco max 达到 5.4ns，这时候如果时钟速率在 133M 以上，一个时钟周期的时间基本被 Tco 吃掉了。
3、 综上因素，共同时钟总线速率很难提升到 200M 以上，其实行业公认共同时钟总线速率在 133M 以上的时候设计难度已经非常大了。

既然 Tco 是影响共同时钟总线速率的重要因素，那么有什么办法可以解决这个问题呢？
工程师的创新力是无穷的，解决办法也非常简单，不再用外部时钟来同步数据了，而是时钟和数据一起往前走。你数据发出有 Tco 延时，我时钟发出照样有 Tco 延时，于是两个 Tco 就抵消了。晕了？我们看图说话，如下图所示：



1、 CLK 触发之后，经过一个相同的 Tco 延时，数据和 Strobe 信号一起发出
2、 数据和 Strobe 信号经过同样的 Flight Time，达到接收端
3、 接收端根据 Strobe 信号来对数据进行采样，需要满足建立保持时间

这样的方式就是源同步时序总线，由于 Strobe 和 Data 一起出发，一起行动，解决了时钟触发的延时（Tco）问题，也一起克服路上遇到的艰难险阻（T Flight Time）。这个方式是不是很像生活中理想的夫妻模式呢？

这种情况下，走线的长短已经不是问题了，反正我们步调一致。对了，关键就是步调一致，反映到走线，就是尽量等长。再次借用 Cadence Allegro 的规则管理器来解释，源同步时钟总线最合适的电子规则就是我们常用的 Propagation Delay 啦。

如果我们把源同步时钟总线比喻成夫妻生活，步调一致，一起克服困难；那么返回头看看共同时钟总线，是不是可以理解为恋爱时候的关系呢，虽然男女之间的目的是一致的，都是到达接收端（结婚），然后有一定的裕量（生活）。但是由于没有达成信任与默契，需要更多的外部条件来协调。然后女生对男生说，不管你有多好，基本条件（房子车子）是要满足的，这就是 Total Etch Length 的要求，你必须满足一个最大最小的范围条件。

6、 DDR3/DDR4 时序关系概述

DDR 是典型的源同步时序，我们就以 DDR3 为例，详细说明下 DDR 设计需要满足的时序关系。

上文说到，源同步时钟的目标就是 Strobe 和 Data 一起到达，然后满足到达之后的建立保持时间关系。按照这个目标，只要 Strobe 和 Data 等长设计，好像 DDR 的速率提升就不是什么问题了。不去说什么能不能跑到 10Gbps 或者更高速率，至少在 DDR3 的 1600Mbps 不会有什么困难。总共一两百皮秒的建立保持时间需求，就算加上 derating 的数据，对于 1.25ns 的 Tck 来说，好像都不会有任何问题。

Speed		DDR3-800		DDR3-1066		DDR3-1333		Units	NOTE
Parameter	Symbol	MIN	MAX	MIN	MAX	MIN	MAX		
Data Timing									
DQS,$\overline{DQS}$ to DQ skew, per group, per access	tDQSQ	-	200	-	150	-	125	ps	13
DQ output hold time from DQS, $\overline{DQS}$	tQH	0.38	-	0.38	-	0.38	-	tCK(avg)	13, g
DQ low-impedance time from CK, $\overline{CK}$	tLZ(DQ)	-800	400	-600	300	-500	250	ps	13,14, f
DQ high-impedance time from CK, $\overline{CK}$	tHZ(DQ)	-	400	-	300	-	250	ps	13,14, f
Data setup time to DQS, $\overline{DQS}$ referenced to $V_{IH}(AC)V_{IL}(AC)$ levels	tDS(base) AC175	75	-	25	-	-	-	ps	d, 17
	tDS(base) AC150	125	-	75	-	30	-	ps	d, 17
Data hold time to DQS, $\overline{DQS}$ referenced to $V_{IH}(DC)V_{IL}(DC)$ levels	tDH(base) DC100	150	-	100	-	65	-	ps	d, 17
DQ and DM Input pulse width for each input	tDIPW	600	-	490	-	400	-	ps	28

ΔtDS, ΔtDH Derating in [ps] AC/DC based[1]																	
		DQS,DQS Differential Slew Rate															
		4.0 V/ns		3.0 V/ns		2.0 V/ns		1.8 V/ns		1.6 V/ns		1.4V/ns		1.2V/ns		1.0V/ns	
		ΔtDS	ΔtDH	ΔtDS	ΔtDH	ΔtDS	ΔtDH	ΔtDS	ΔtDH	ΔtDS	ΔtDH	ΔtDS	ΔtDH	ΔtDS	ΔtDH	ΔtDS	ΔtDH
DQ Slew rate V/ns	2.0	88	50	88	50	88	50	-	-	-	-	-	-	-	-	-	-
	1.5	59	34	59	34	59	34	67	42	-	-	-	-	-	-	-	-
	1.0	0	0	0	0	0	0	8	8	16	16	-	-	-	-	-	-
	0.9	-	-	-2	-4	-2	-4	6	4	14	12	22	20	-	-	-	-
	0.8	-	-	-	-	-6	-10	2	-2	10	6	18	14	26	24	-	-
	0.7	-	-	-	-	-	-	-3	-8	5	0	13	8	21	18	29	34
	0.6	-	-	-	-	-	-	-	-	-1	-10	7	-2	15	8	23	24
	0.5	-	-	-	-	-	-	-	-	-	-	-11	-16	-2	-6	6	10
	0.4	-	-	-	-	-	-	-	-	-	-	-	-	-30	-26	-22	-10

但是下面这张表格告诉我们，源同步时序系统速率提升之后，除了 Tck 变小会让时序裕量变小之外，还有更多其他因素会影响时序。

- **WRITEs at DDR3-1066**

Element	*Skew Component*	*Setup*	*Hold*	*Units*	*Comments*
Clock	Data/Strobe chip PLL jitter	35	35	ps	
	DRAM tJITper	45	45	ps	*derate back out what DRAM tests for*
	Clock skew	**0**	**0**	ps	
Transmitter	**Memory Controller Skew**	**238**	**238**	**ps**	Assume similar to DRAM, used DRAM's
Interconnect	*Controller uses uncoupled package model, some increase can be expected pending Controller model used; probably in the 15ps to 30ps				
	DQ Crosstalk and ISI*	25	25	ps	1 victim (1010...), 4 aggressors (PRBS); Different termnetion scheme for 800 other than that used for 1066 and 1333 should reduce xtlk to less than 25ps
	DQS Crosstalk and ISI*	5	5	ps	1 shielded victim (1010...), 2 aggressors (PRBS)
	Vref: Reduction	10	10	ps	+/- 30 mV included in DRAM skew; additional = (+/- 10 mV) / (1 V/ns)
	Reff Mismatch	0	0	ps	+/-6% accounted for by DRAM spec
	Path Matching (Board)	10	10	ps	Within byte lane: 165 ps/in * 0.1 in; Impedance mismatch within DQS to DQ
	Path Matching (Module)	10	10	ps	Module routing skew (30% reduction with leveling)
	Input Capacitance Matching	5	5	ps	strobe & data shift differently
	ODT Skew (1%)	5	5	ps	Estimated
	Total Interconnect	**70**	**70**	**ps**	
Receiver	DRAM Skew	**165**	**165**	**ps**	tDS, tDH from DRAM spec.derated for faster slew rate and to Vref
Total Loss	Total Skew	**472**	**472**	**ps**	Trans. + rec. + interconnect skew
Max Eye	**Time Allowed**	**469**	**469**	**ps**	
Budget 6L	**Timing Margin**	**-3**	**-3**	**ps**	6 layer board (stripline), 45-ohms, 0.135mm trace to trace spacing
6 to 4 layer	DQ Crosstalk and ISI	7	7	ps	increase using microstrip vs strip line
	DQS Crosstalk and ISI	17	17	ps	increase using microstrip vs strip line
Budget 4L	**Timing Margin**	**-27**	**-27**	**ps**	4 layer board (micronstrip) 45-ohms, 0.135mm trace to trace spacing

码间干扰（ISI），数据线之间的串扰（Crosstalk），还有同步开关噪声（SSN）等，都会吃掉大量的时序裕量。

所以在设计中，我们要遵循下面的分组等长原则（DDR3 的 DQS 和 CLK 不需要用绕线等长的方式来控制时序，而是通过芯片内部的读写平衡功能，我们在 DDR3 系列文章中已经多次提出来了）

Technology	DDR2	DDR3
Delay Matching Requirement（等长控制要求）		
Match ADDR/CMD/CTRL to CLK tightly	YES	YES
Match DQ to DQS in same lane tightly	YES	YES
Match DQS to CLK loosely	YES	Not Required

	DDR3	DDR4
Source-Synchronous Signal	同组DQ/ECC与对应DQS等长保持在100mil内	同组DQ/ECC与对应DQS等长保持在50mil内
	DQS 差分线对内等长保持在5mil内	DQS 差分线对内等长保持在5mil内
	同组DQ/ECC等长保持在30mil内	同组DQ/ECC等长保持在10mil内
	同组信号严格要求同层走线	同组信号严格要求同层走线
Clock Signal	CLK差分线对内等长保持在2mil内	CLK差分线对内等长保持在2mil内
	同一DIMM的CLK之间等长保持在20mil内	同一DIMM的CLK之间等长保持在5mil内
Source Clocked Signal (MA[15:00], BA[2:0], RAS_N, CAS_N, WE_N, MA_PAR)/BG[1:0],ACT_N,C[2]	同一个DLL组,CMD要求等长保持在500mil内 所有CMD信号要求与对应时钟保持在2inch内	同一个DLL组,CMD要求等长保持在25mil内 所有CMD信号要求与对应时钟保持在2inch内
Source Clocked Signal CS_N[9:0], ODT[5:0], CKE[5:0]	同一个DLL组,UDIMM CTL要求与时钟等长保持在100mil内，for RDIMM保持在240mil内	同一个DLL组,CTL要求与时钟等长保持在25mil内，for RDIMM保持在100mil内

7、题外话

说了“围殴”话题的时候，大家各诉己见。
观点可以百花齐放，话题可以争议性质！
诚征各路英雄参与，微信群同步展开讨论！
所以第一个争议性话题来了：等长越严格，时序裕量越大，系统越稳定！

这句话应该有很多硬件工程师是同意的，所以我们也经常能看到类似的规则：

- DDR3 同组的 DQ 和 DQS 需要+/-1mil 等长
- DDR3 同组的 CLK 和 Add/Ctrl/Cmd 需要+/-10mil 等长
- DDR3 的 CLK 和 DQS 需要+/-100mil 等长
- PCIE3.0，差分对内需要+/- 0mil 等长，没有误差

……

类似的规则还可以衍生为：

- 线间距越大约好，减少串扰
- 走线需要两面严格参考完整地平面

……

类似的规则，用现在最流行的话来说，应该就是“理都懂”，但是臣妾做不到呀！

这里忍不住要吐槽一下：随着 PCB 设计这个工种慢慢从硬件设计中独立出来，大部分公司里面 PCB 设计工程师和硬件原理工程师不是同一个人。所以有些 EE 会想当然或者很理直气壮把很严格的规则提交给上下游，反正需要绕线的不是我……更不用说外包的 PCB 设计了，更是占住了甲方的位置，你们给我往死里绕线。

当然，更多的硬件原理工程师并不是这么想的，只是当这个项目落在自己头上的时候，战战兢兢，如履薄冰，想把一切事情做到完美。

这篇文章，就是为后面这种工程师写的。继续往下看之前，先分享一篇摘自 SI-List 的讨论。我最喜欢里面的一句话：

http://www.edadoc.com/cn/jswz/show_780.html

"We are engineers. We are supposed to think."

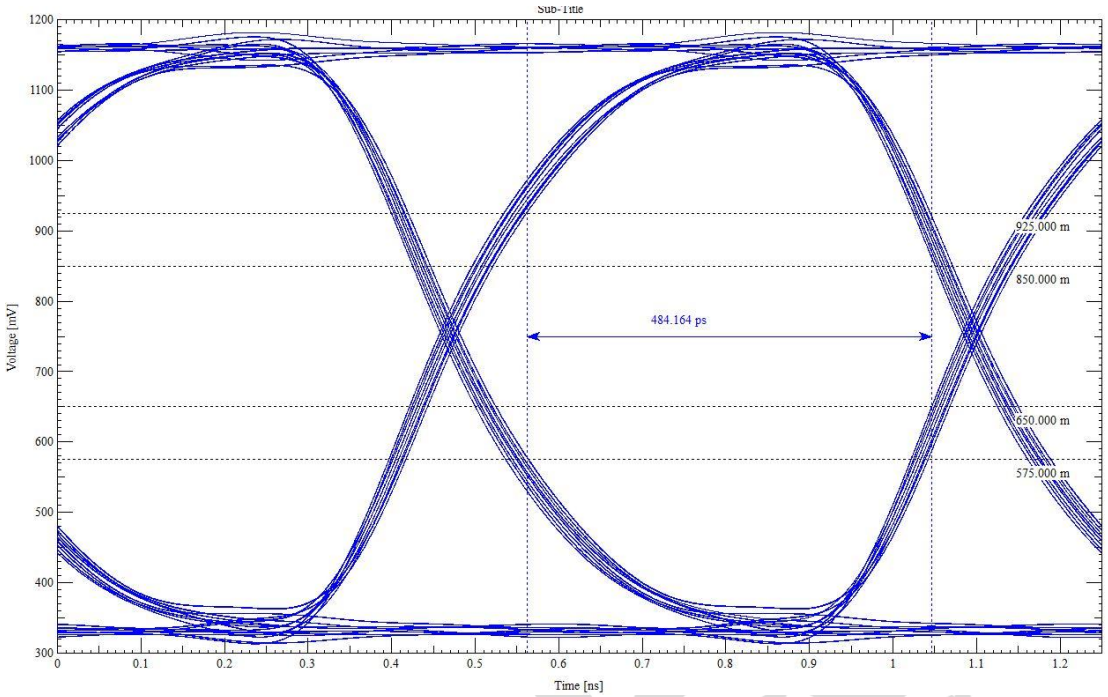
也喜欢文章里面说的，搞清楚什么是“Design Guide”，什么是“Spec”

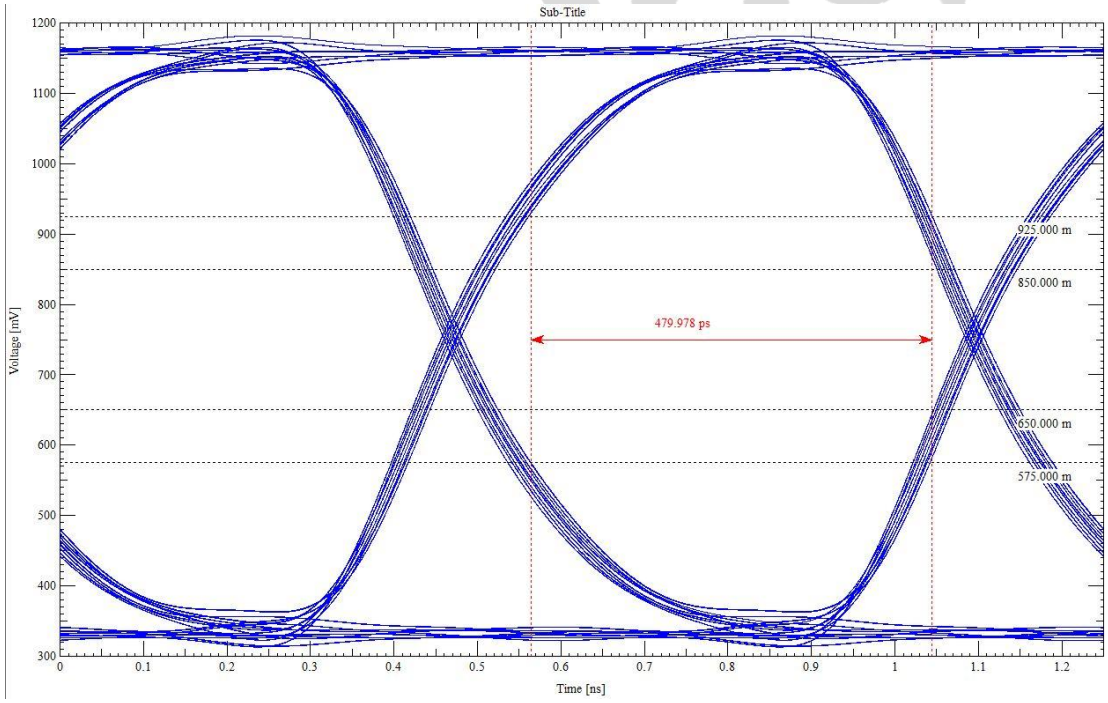
8、等长与等时

之前的文章有提过，所有的等长都不是目的，设计的目的是等时。所以我们首先要知道，+/-10mil 对应的时间是多少？貌似+/-1mil 等长比+/-10mil 严格了 10 倍，我的时序裕量会不会好很多？

大家应该还记得高速先生王锐写的那篇文章：《信号是怎样传输的？》，没看过的可以翻出来再看看。文章的最后，给出了一个案例：同一种阻值的走线，微带线时延是 145.9ps/in，带状线时延为 173.6ps/in。

我们需要记住的值是每英寸 145~170ps，对于带状线来说，每 ps 延时对应的走线长度是 6mil 左右。所以，+/-10mil 等长和+/-1mil 等长，在时间上的差异不超过 3ps。再回头看看我们的系统，有多少是真正需要 3ps 裕量的？





上图是同一个波形，不同的两个人测量的时序窗口大小，有 4 个多 ps 的误差。测量的波形也是一样，大家都有这样的经验。

高速先生有时候为了说服一些客户不要纠结在等长是 1mil 还是 5mil，或者劝告说完全等长是没有意义的，我们也是拼了，下面是我们用过的一些比喻：

- 蚊子再小也是肉，你要吃蚊子我忍了，但是蚊子腿就不要去吃了吧
- 中华民族艰苦节约的良好美德，但是现在的社会，就不用纠结于要不要省 1 毛钱了吧
- ……

如果你能理解以上比喻，那么应该可以理解：

大部分的系统，不管是 DDR3 的数据分组还是 PCIE3 或者 10G Base KR 的差分对内，等长做到 5mil 已经足够了。速率更低的系统，还可以放宽更多。

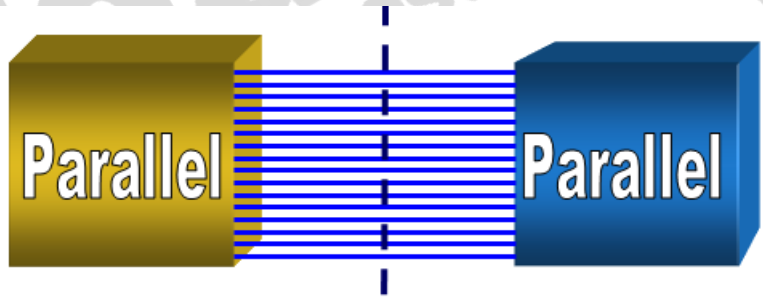
这篇文章的篇幅已经比较长，所以主要讨论等长不需要过度考虑这一个问题点。另外一个相关的话题是：严格等长会不会带来负面影响，会不会成为系统失效的根源。这个话题我们稍后单独讨论，大家也可以就此提出自己的意见和建议。

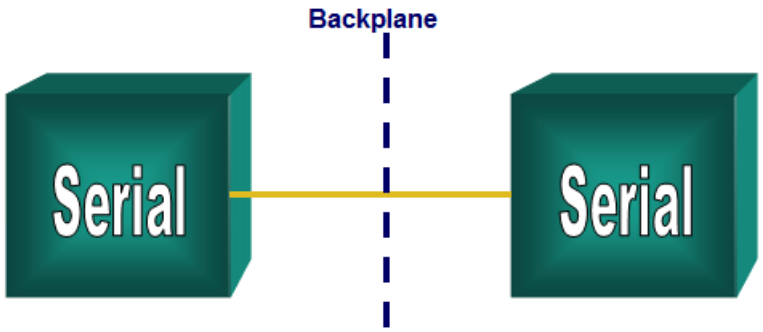
9、串行总线来了

绕线话题从开篇到结尾，花了好几个月哈，老是出差，没有时间静下来写东西。不过或许出差也只是借口，而是因为时序绕线这个话题实在是有点难写好吧。不管怎么说，挖下的坑是一定要埋上的，今天就是绕不完的等长的最后一篇，串行总线来了。

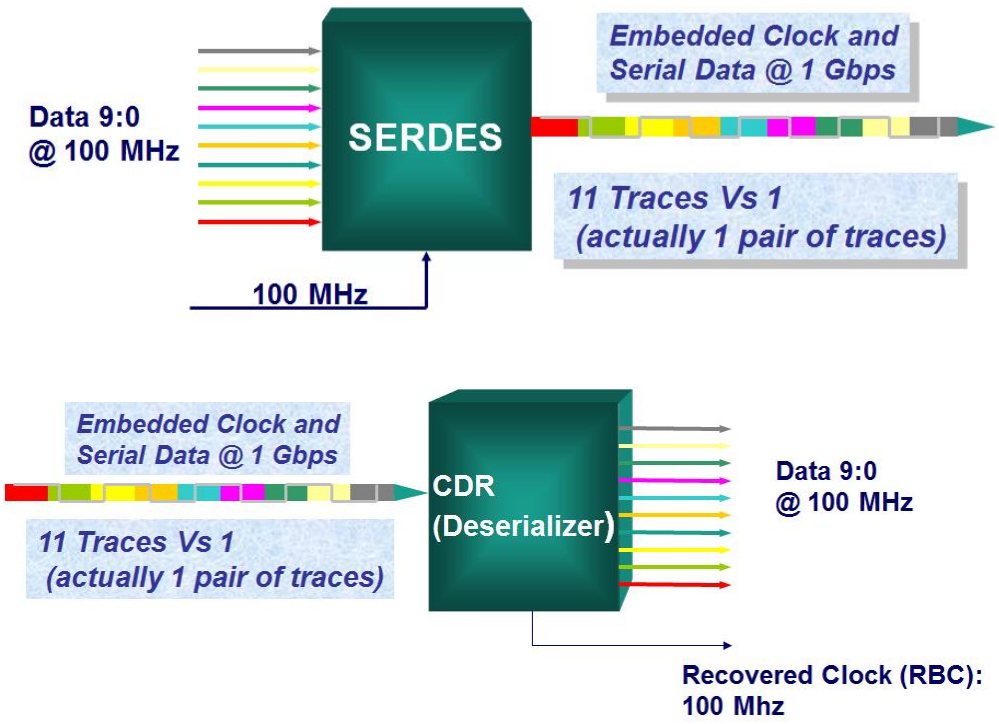
上一篇文章发出来之后，不少网友回复说，DDR3 的同组数据并不需要做到 5mil 等长这么严格呀。看到这样的回复，高速先生们都是热泪盈眶：“同志，见到你真好……”。说实话，写这个系列文章还是有点私心的，希望以后不会再收到客户提出的+/-1mil，+/-0.5mil 等长这样的要求，我们已经是很满足了。+/-5mil 或者+/-10mil，这已经不是个事了，咬咬牙，加点班，这个等长我们就忍了。

到了串行总线，貌似速率更高了，大家对等长的要求也更严格了。那么串行总线到底是什么鬼？





高速串行总线的技术核心是 SerDes 技术，也叫 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。下面一张图，轻松看懂 SerDes 的工作原理。



篇幅关系，也是高速先生的风格，我们不去解释 SerDes 的工作原理细节，从上图，我们只需要看懂：

- 时钟是内嵌在差分对里面的
- 到了接收端，时钟信号被重新恢复

10、 高速串行总线（SerDes）的等长要求

从上文可以很容易得到结论：高速串行总线技术采用时钟和数据恢复技术，从而解决了限制数据传输速率的信号时钟偏移问题，减少布线冲突、降低开关噪声、更低的功耗和封装成本等。所以差分对与差分对之间基本没有等长要求；时钟是依赖串行解串的技术进行传输与恢复。

高速串行总线设计的难点从传统的时序问题，变成自身的 Jitter，误码，损耗衰减等问题，关注的重点是差分对本身的信号质量，以及尽量避免受外界干扰影响。

差分对自身的问题，包括：

- 对内等长带来的相位问题以及差模共模模态转换
- 差分对间的串扰问题评估及优化
- 导体损耗，介质损耗等高频损耗问题

这些话题里面，差分相位及模态转换是绕线和走线拐角关注的问题点，会在这个时序系列里面进行探讨。其他问题则会在以后专门的高速串行总线系列来进行讨论。

注：说到高速串行总线基本没有差分对与差分对之间等长要求，需要注意的是差分线不完全等于高速串行总线；换句话说，高速串行总线基本都是差分形式，但是不是所有的差分线都是高速串行总线。

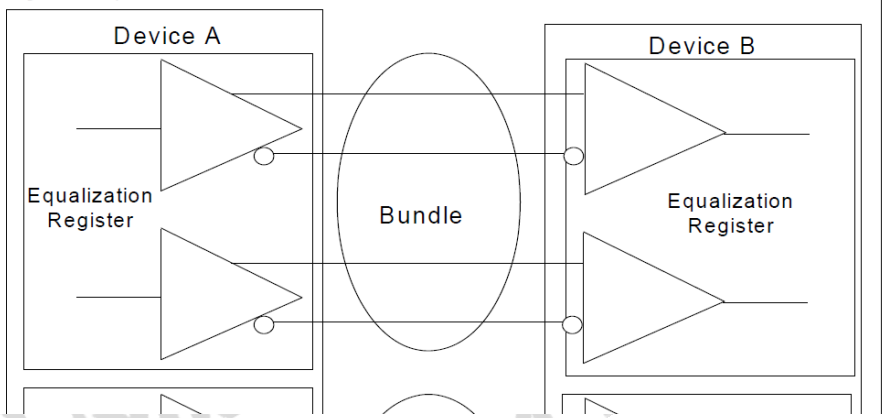
如果差分线不是高速串行模式，而是并行总线，等长设计按照之前导论的并行总线原则来执行（比如传说中还没有推出的差分版本 DDR4）

既然说高速串行总线基本没有差分对与差分对之间等长要求，那么就还是有特例：

Intel 的 PDG 里面，对同一 Bundle 内部的 PCIE3 信号和 QPI 信号提出了等长要求。不过这是一个很宽松的要求，正常的布局布线设计，基本是不用考虑绕线这个事情的。

Table 3-6. Length Matching within a Routing Bundle (Sheet 1 of 4)

Guideline	Routing bundles are differential pairs that share a common exqualiztation register. Differential pair 0 and 1 are a bundle, differential pair 2 and 3 are a bundle, and so on. For example on PCIe PE1B_TX_DN/DP[0] and PE1B_TX_DN/DP[1] or PE1B_TX_DN/DP[2] and PE1B_TX_DN/ DP[3] are two examples. On Intel QPI, QPI0_DTX_DN/DP[00] and QPI0_DTX_DN/DP[01] or Intel QPI, QPI0_DTX_DN/ DP[02] and QPI0_DTX_DN/DP[03] are also two examples. Length match pairs in a bundle to within 0.5 inch for PCIe3 and 1 inch for Intel QPI.



回顾一下之前文章的要点，温故而知新嘛，现在不都是流行重要的事情多说几遍吗？：

1、共同时钟总线的时序要求，不是等长，而是满足一个范围，更多的时候，需要注意不要走线太长。（这时候男女还没结婚，步调不一致，等长没有意义，大家看的是外部条件，需要满足一个基本要求）
2、源同步时钟总线的时序要求，主要是分组等长。但是等长只是满足了静态偏移，做到几个 ps 已经是足够好了（+/-10mil 左右）。影响更大的是动态偏移，也就是 SSN，ISI，Crosstalk 等，不要过度强调等长（+/-1mil），而忽略了其他更重要的设计要求（这时候男女已经结婚了，步调一致最重要，只要夫妻齐心，Tco，飞行时间那都不是事。但是外部的风雨还是会影响感情，担心来自于电源噪声、串扰的影响）
3、高速串行总线，时钟内嵌，差分传输，更关注信号自身的品质，外部的干扰已经很难影响到时序了，需要关注差分线自身的设计质量（男女经过磨合，达到了灵魂伴侣的层次，只要两人同心，一切外部的事那都不是事了，所以关注的重点变成两人是否同心 - 差分）

问题来了

集思广益，搜集下哪些属于差分线设计，却需要较严格差分对间等长的特例？

高速先生欢迎您和我们一起进行交流，关注微信名（高速先生），直接将答案通过会话回复，参与互动答题即有机会获得奖品，回复关键词“奖品”查看更多。

【关于一博】

一博科技专注于高速 PCB 设计、PCB 制板、焊接加工、物料供应等服务。作为全球最大的高速 PCB 设计公司，我司在中国、美国、日本设立研发机构，全球研发工程师 500 余人。超大规模的高速 PCB 设计团队，引领技术前沿，贴近客户需求。

一博旗下 PCB 板厂成立于 2009 年，位于广东四会（广州北 50KM），采用来自日本、德国的一流加工设备，TPS 精益生产管理以及品质管控体系的引入，致力为广大客户提供高品质、高多层的制板服务。

一博旗下 PCBA 总厂位于深圳，并在上海设立分厂，现有 12 条 SMT 产线，配备全新进口富士 XPF、NXT3、全自动锡膏印刷机、十温区回流炉等高端设备，并配有波峰焊、AOI、XRAY、BGA 返修台等配套设备，专注研发打样、中小批量的 SMT 贴片、组装等服务。

【关于高速先生】

高速先生由深圳市一博科技有限公司 R&D 技术研究部创办，用浅显易懂的方式讲述高速设计，成立至今保持每周发布两篇原创技术文章，已和大家分享了百余篇呕心沥血之作，深受业内专业人士欢迎，是中国高速电路第一自媒体品牌。

如何关注

1、搜索微信号“高速先生”
2、扫描右侧二维码，开始学习

扫一扫，即可关注