再谈 SDRAM 的布线——有关 Mentor WG、DxDesinger、Expedition、CES
2008-06-18 14:57

- 前言的前言

这篇文章我写了很久很久，因为最近很忙很忙。现在我逐渐开始接触开关电源和可靠性设计的东西，好像离原来我定义的 EE 越来越远了。也许以后我要向模电或管理人员发展了……我还是纯朴地希望自己能一直保持做一个不断钻研的 EE 工程师。不说了，做人要厚道，转载请注明来自我是一只鱼同学的 EE 小站，邮件地址 cosine@126.com。

- 前言

最近一个多月都在研究 Mentor WG，已经对 DxDesigner + Expedition 的画板流程有了比较清醒的认识，我对 Mentor WG 评价可以套用对目前国产汽车的评价——配置齐全、做工粗糙。虽然 WG 有很强的功能，但是 BUG 实在是数不胜数，而且有些 BUG 可能导致你的工程彻底报废，所以建议使用时辅以自动备份软件，减小工程崩溃带来的损失。
今天要谈的话题都是基于 WG 的，因为 PADS、Protel / DXP 之类的软件没有这样的功能或功能不完整。不过，也可以使用其他软件进行 PCB 前仿、手动完成线长匹配等工作；工具只是人的技巧的辅助和延伸，要是没有高速 PCB 设计的知识，同样完不成高速数字 PCB 的设计。本文为我是一只鱼同学 EE 小站的原创文章，转载请注明出处；本文对初学者而言，技术难度较高，如果有不明白的地方，可以留言。另外继续废话几句，事实上 SDRAM 对布线的要求是很低的，DDR 才是真正有挑战的东西，可惜我目前没有 DDR 的项目，也没有办法验证我对理论的理解，希望以后有机会和大家分享我的心得。下面正式开始：

- 什么是高速数字 PCB，怎么入手？

高速数字 PCB 简单来说可以理解为关键部分如存储器总线的工作频率高于数十至一百 MHz 的PCB,更严格的定义应该用传输线来描述,当PCB上的信号的传输延迟大于上升时间的1/10时,这个信号的传输路径就应该视为传输线;即应当用与传统低速数字电路不同的方法对待。那么怎么入手？我是学机械出身的，电路原理和模电都是三脚猫知识；我个人认为 High-Speed Digital System Design 是本不错的书。首先看书，弄明白在频率高了以后会出现什么样的现象，有什么东西需要考虑之后，再继续后面的设计。不过我可以做个简单的概括，高频数字电路设计的大部分工作是解决传输线中信号反射问题和延迟问题。BTW，很久以前我还很菜的时候写了一篇文章 http://xianzilu.spaces.live.com/blog/cns!4201FDC93932DDAF!171.entry，这是关于 PCB 后仿的（这个词下面马上解释），大家有兴趣可以看看。

- 高速 PCB 设计的流程

元件布局——〉前仿真——〉布线——〉后仿真——〉出 CAM 文件
其他不多说了，就解释下前仿真和后仿真。
前仿真就是在器件 IBIS 模型、网络拓扑结构和器件分布的基础上做的对 PCB 可实现性仿真。举个例子解释前仿真的作用，如果器件、板子的机械结构都已经定下来了，CPU 和 SDRAM 插座相隔 10000mil，那么在布完这个板子之前，怎么知道这个板子能不能正常工作？关于如何使用 WG 进行前仿真，后面再说。

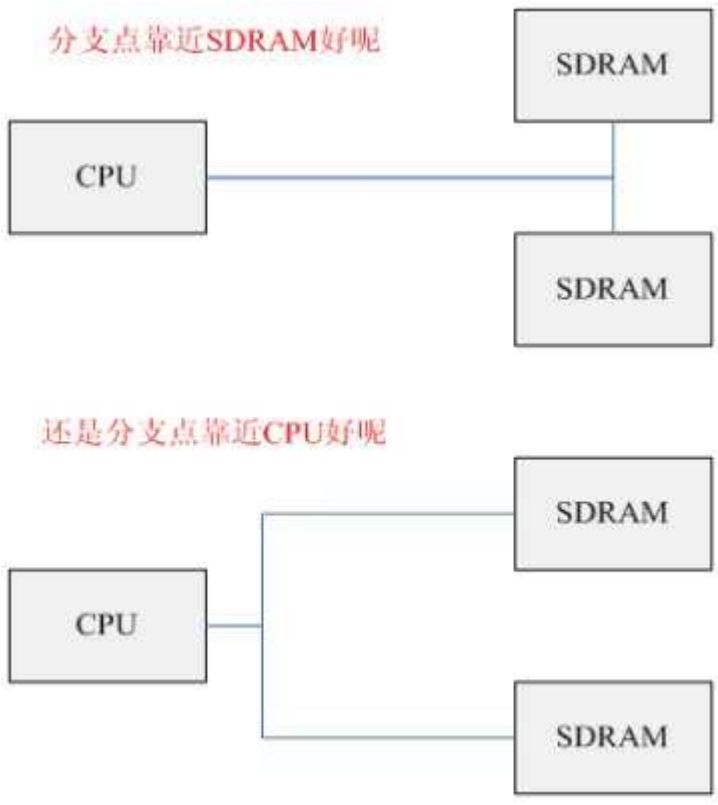
后仿真就是在板子走线已经成型之后，对布线结果进行验证而作的仿真。后仿真会在前仿真基础上加上过孔模型、串扰、电磁兼容性等仿真内容。刚才提到的我的菜菜鸟文章 http://xianzilu.spaces.live.com/blog/cns!4201FDC93932DDAF!171.entry，说的就是后仿真。

- SDRAM 对布线有什么要求？

首先必须明白 SDRAM 是一种什么样的存储器，搞清其接口工作的逻辑时序。SDRAM 是一种同步动态存储器，所有接口信号都是通过时钟同步和采样的。这就对 SDRAM 的布线提出了要求——保证采样的正确性。于是，应用高速数字电路的知识结合某种具体 SDRAM 器件和你的 PCB 进行分析，发现在正常工作频率（如 100MHz）下，在 PCB 走线上的信号传输时间大于其上升时间 1/10。于是，接下来考虑高速数字电路两大问题反射和延迟：反射造成 SDRAM 时钟线信号出现振铃，多次穿越门限造成误触发；数据线和时钟线的传输延迟不相同，造成时钟上升沿采样不到所需要的数据。接下来应用解决方法：时钟线串联电阻做阻抗匹配；布线时控制数据线和时钟线的长度差在一定范围内。当然，我这里说的是一个很简单的演绎过程，还有拓扑结构、最大布线长度等重要问题没有考虑，请大家仔细阅读我是一只鱼同学刚才推荐的课本。提示下，拓扑结构和最大布线长度的选择可以通过前仿真进行验证。

- 进一步的问题，SDRAM 布线用什么拓扑结构好？

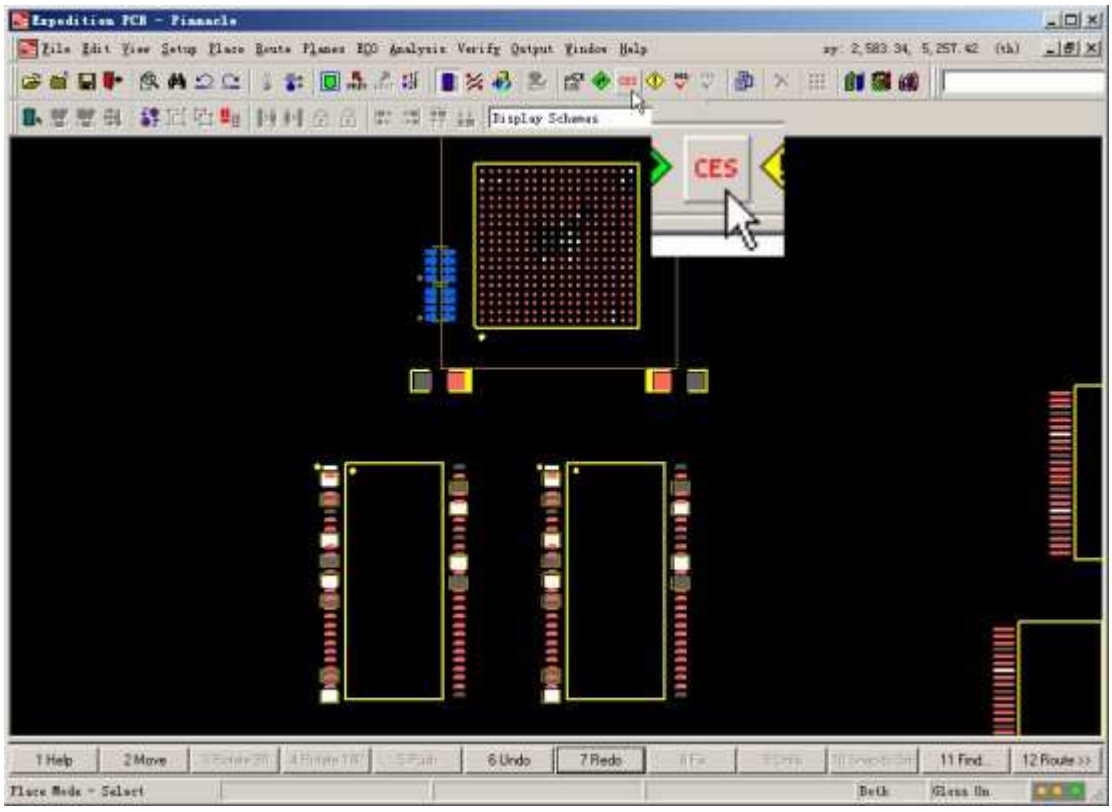
这个问题困扰了我很久很久，终于在学会前仿真后解决了，哈哈。其实大家已经很清楚 SDRAM 要尽量使用 Y 型分支结构（也叫 T 型分支），因为链式结构会产生两个问题：一、两片 SDRAM 的传输延迟不一样，影响 CPU 对数据输出进行采样；二、链式结构的节点处阻抗不连续，是一个反射点，而且反射点和源的距离太大，反射效果明显。但是，如果使用 Y 型分支结构，到底是先分支好呢还是后分支好呢？



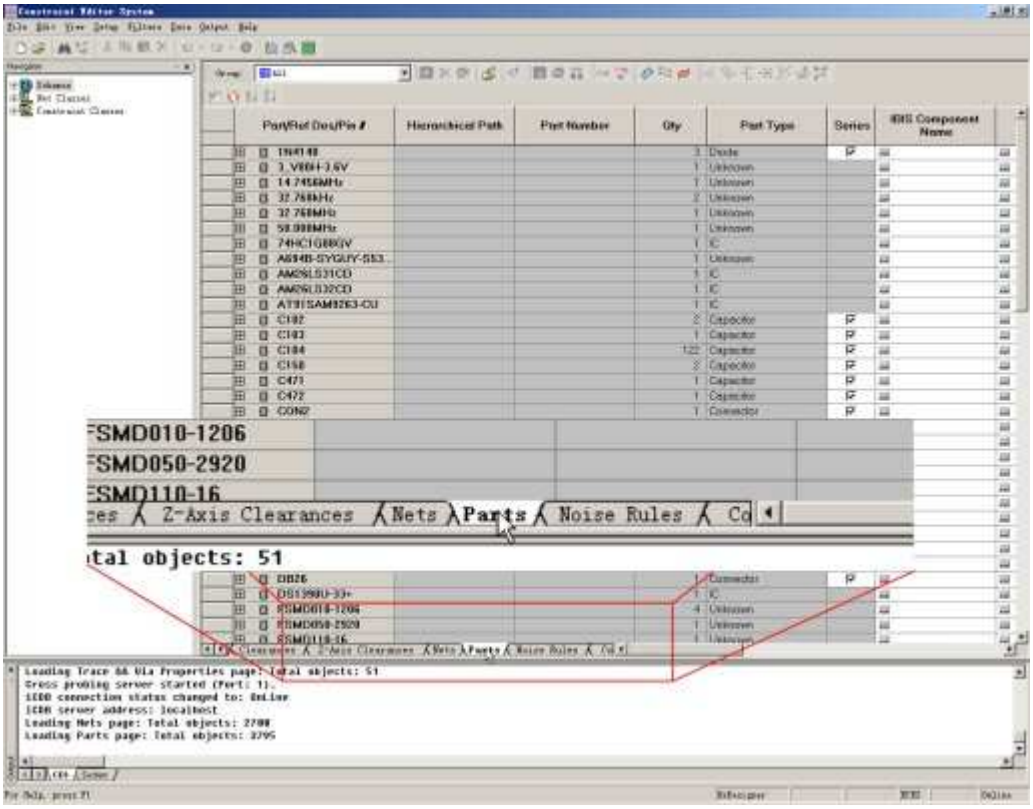
经过前仿真的验证，分支点靠近 CPU 的时候效果稍微好那么一点点。我想这是因为分支点本身是一个阻抗不连续点，也是会发生反射的。如果分支点靠近源端 CPU，反射就会因为传输线的缩短而显得不太明显。我给分枝点靠近 CPU 的这种拓扑结构起个名字，叫短桩 Y 型分支结构。

- 怎样用 WG 进行 PCB 前仿真？

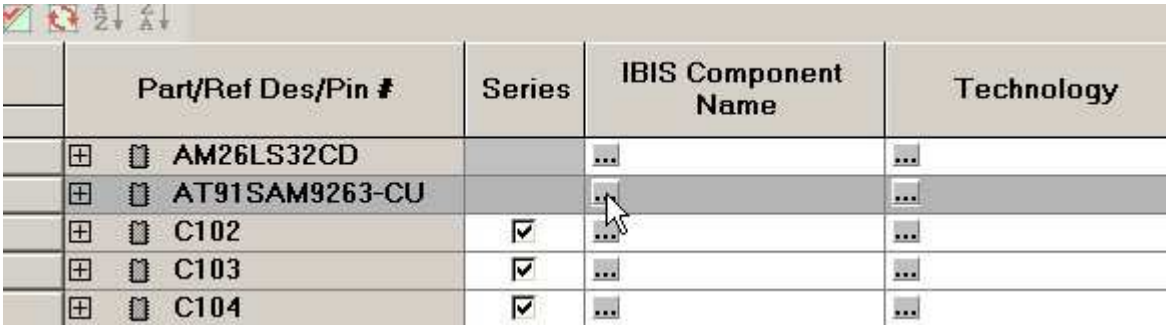
WG 中 PCB 的前仿真的步骤：DxDesigner 画原理图——〉Expedition 布局——〉CES 指定 IBIS 模型——〉CES 指定网络的拓扑结构——〉ICX Pro 前仿真。我们来看图说话。
对于下面这样一个已经完成布局的 PCB，从 Expedition 的工具栏中选择 CES



在 CES 的窗口中见的选项卡中，选择 Parts

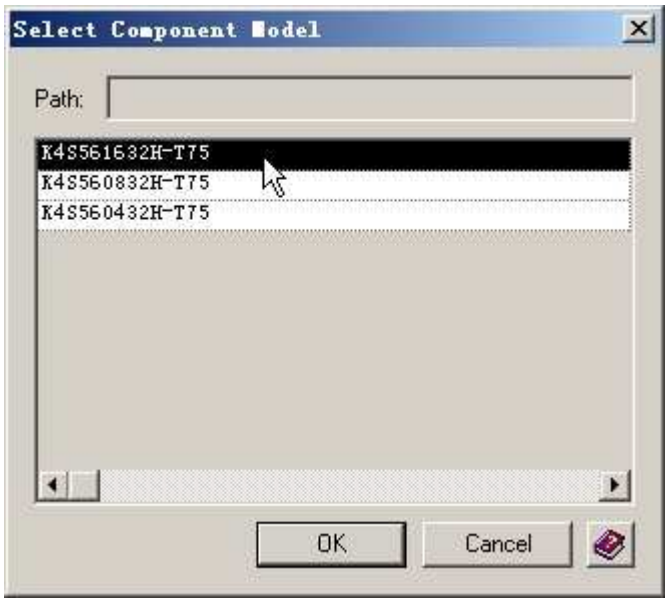


例如对 SDRAM 的走线进行前仿真，就需要制定 CPU、SDRAM 以及其他连接在数据总线上的器件的 IBIS 模型。相应的模型可以在器件的官方网站上下载到。在 Parts 里选中要指定模型的器件。

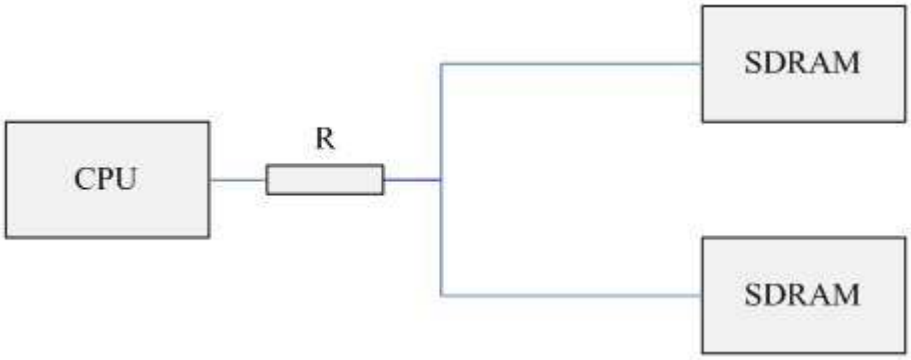


在弹出的窗口中按照步骤 1、2、3（后续图片中的 1、2、3、4 亦表示步骤）选择 IBIS 模型文件

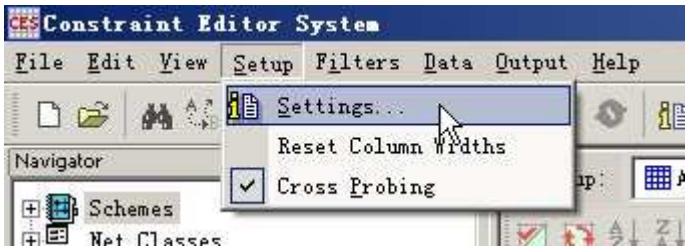
选好后就 OK，重复以上步骤直到把要进行仿真的信号所连接的所有器件的 IBIS 模型都选上。有的 IBIS 文件中含有多个器件的模型，选择你需要的



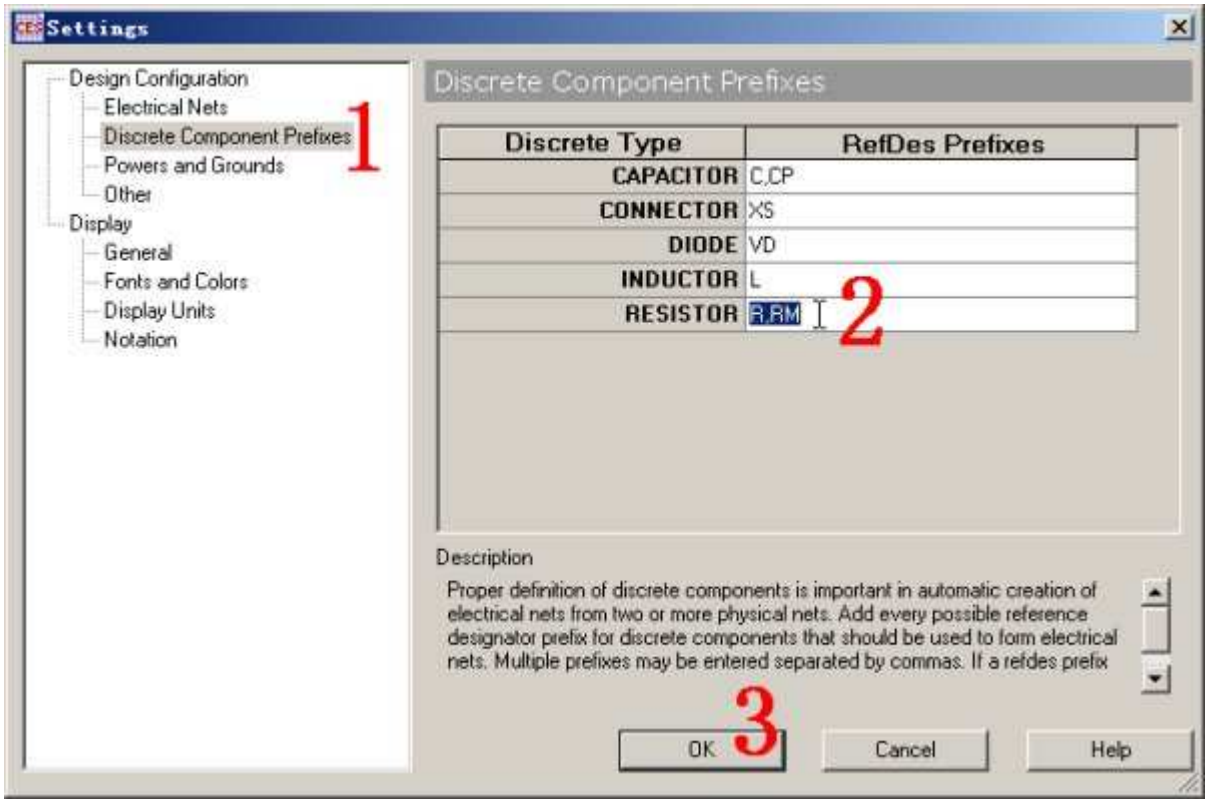
在 SDRAM 信号设计的时候往往会使用电阻对信号进行阻抗匹配，这时候 SDRAM 布线的拓扑结构就会变成下面这样



这个电阻也必须包含到前仿真中去。但是在实际设计中，往往有很多需要匹配的信号，所以一般是使用排阻的。但是默认情况下，CES 是不认识排阻的，这需要设置。选择 Setup 菜单下的 Settings…



在弹出的窗口中选择 Discrete Component Prefixes 选项页，将你所用的排阻前缀输入（如 RM），然后确认

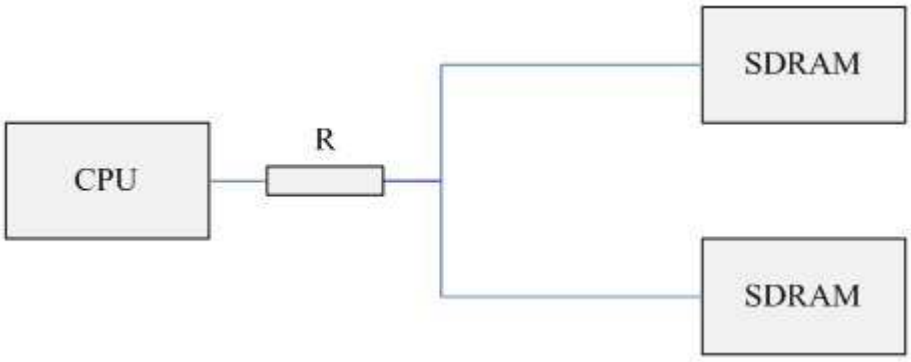


随后你就会发现 CES 把你的排阻认为是串行器件了。顺便提下，如上面这张图所示，CES 会把这些已经定义前缀的器件识别为串行器件。识别为串行器件有好处也有坏处，好处是对于真正用于阻抗匹配等目的的器件，前后的网络会被归为一个网络进行识别（CES 在原来的网络名后面加上“^^^”符号，将两个网络合并）；坏处是很多功能性质的电阻，如运放中设

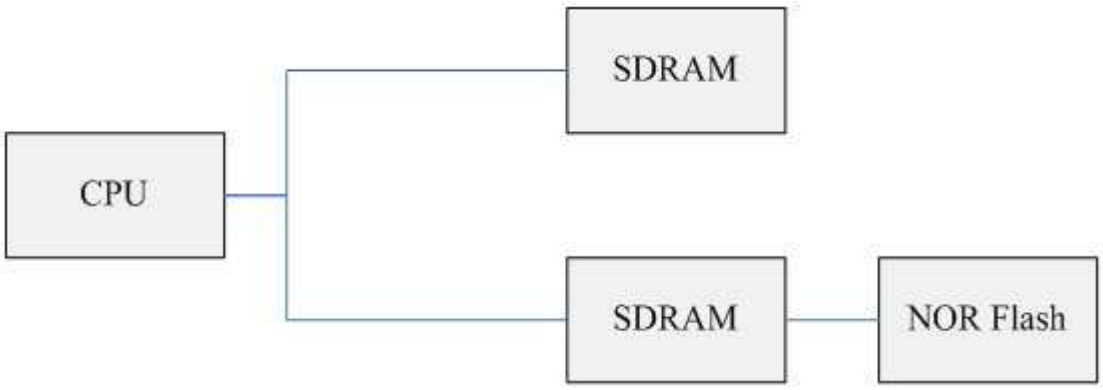
定放大比例的电阻两端的网络也会被归为一个网络识别，这时候就需要把下面这张图中所示的钩号去掉。

接下来，点击 Parts 边上的 Nets 选项卡，选择 SDRAM 的信号

随后配置网络拓扑结构。对于 SDRAM 的控制线来说，它们不连接在 SDRAM 之外的其他上，因此其拓扑结构一般都是之前描述的这种：

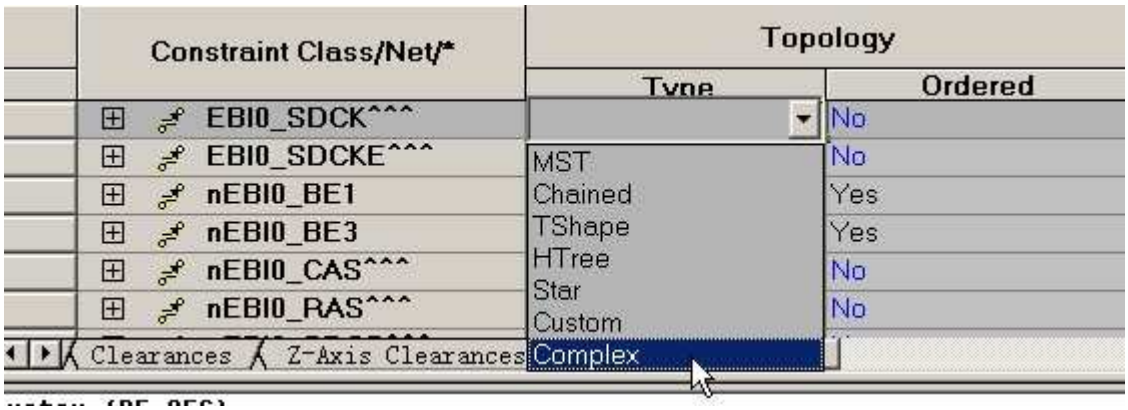


对于一般的地址线而言，往往需要连接除 SDRAM 芯片之外的其他器件，如 NOR Flash，其拓扑结构可能是这样的

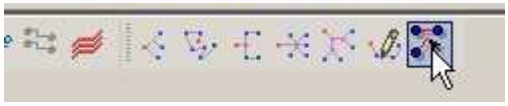


我建议将 NOR Flash 连接在一个 SDRAM 的之后，因为如果再搞短桩 Y 型分支，那么将会有 3 组分支线，布线就很困难了。当然，连接在哪里要根据前仿真的结果来调整，我提供的这种连接对于某些器件应该会有问题。

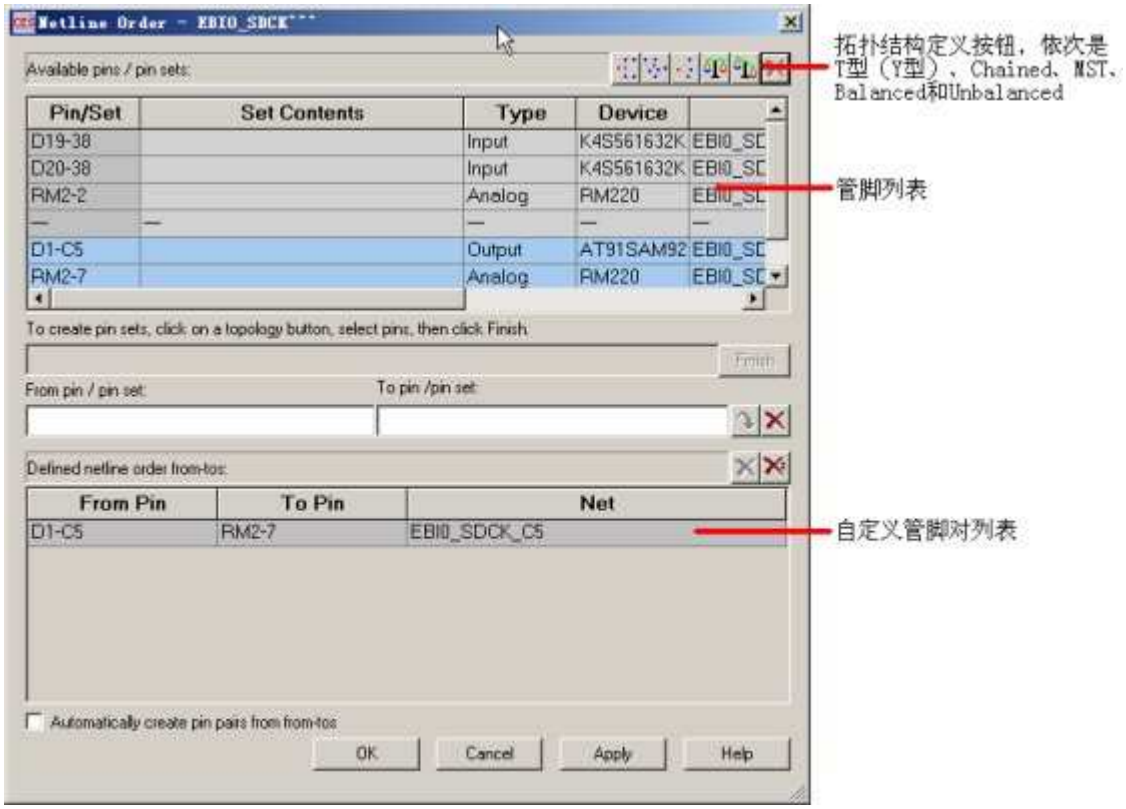
下面要把 SDRAM 的信号定义成上面这 2 种拓扑结构中的一种，在 Nets 列表里找到 Topology 这一列，点击下拉列表。对于 SDRAM 时钟信号 SDCK 这种串联阻抗匹配电阻的拓扑结构而言，选择 Complex。



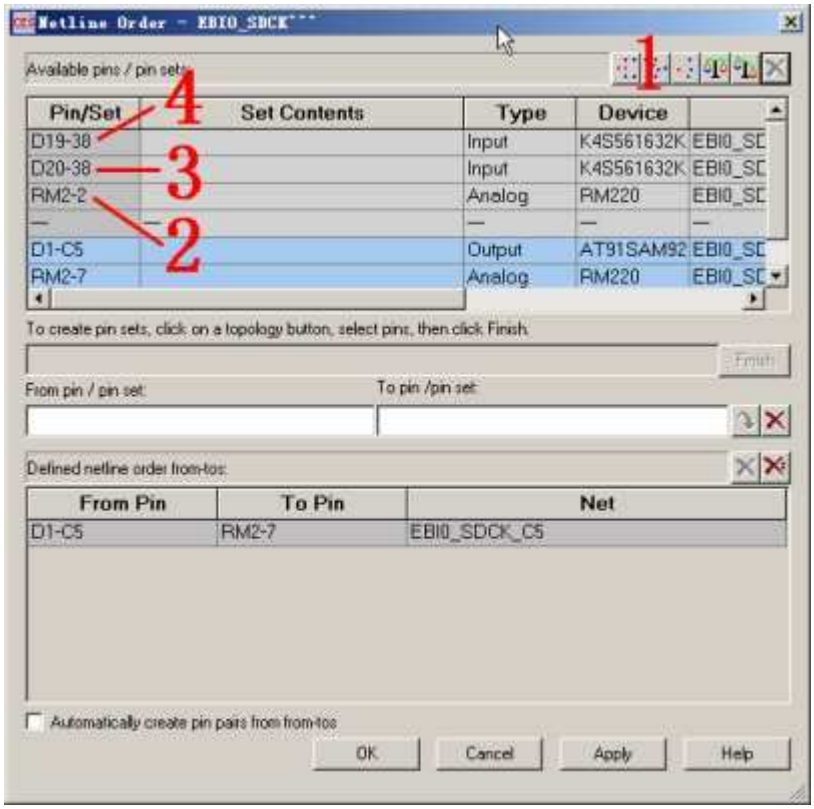
随后点击工具栏上的 Netline order 按钮（这个按钮左边的几个按钮可以定义其他不同的拓扑结构，与上图列表中对应，依次是 MST、Chained、T、Star、HTree、Custom，这些不同拓扑结构的含义可以 Google 下或者参看 Mentor WG CES 的手册）



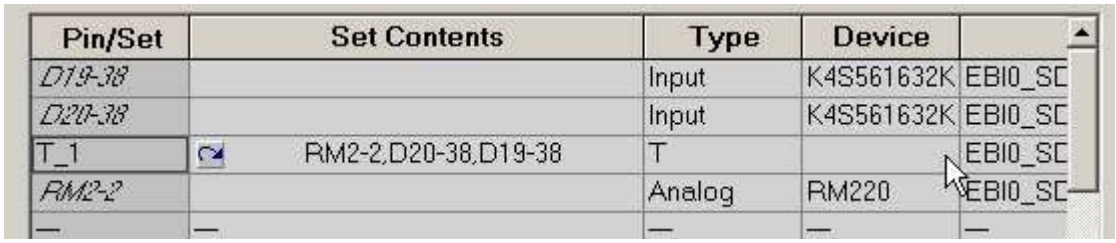
出来这样一个对话框



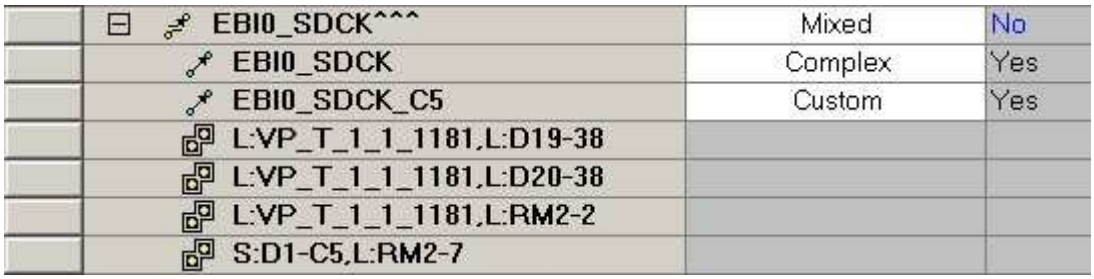
之前已经提到 SDRAM 的地址和控制信号应当是短桩 Y 型分支，如果还有别的器件就连接在 SDRAM 之后。先说明下，因为定义了排阻为串行期间，所以 CES 自动的将 CPU 到排阻的连接识别出来并列在管脚对列表里了；管脚列表里蓝色背景的部分是已经在管脚对列表中存在的管脚。随后就需要定义 Y 型分支，点击 Y 型分支拓扑结构图标，再依次点击信号源管脚和两个负载管脚，如下图所示



在管脚列表中就会出现如下显示

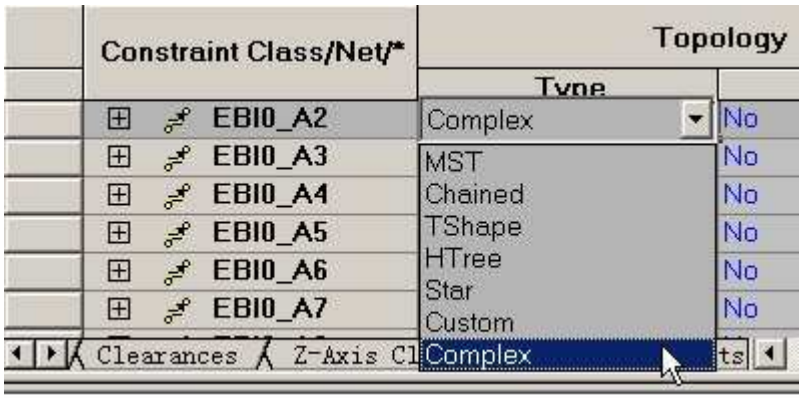


这说明已经定义了一个 Y 型分支。定义之后的 Y 型分支可以删除和修改，具体细节请看 CES 手册。随后点击对话框下面的那个复选框，“Automatically create pin pairs from from-tos”，确定，CES 的 Nets 列表中，刚才定义的 Net 下就出现如下图所示的内容

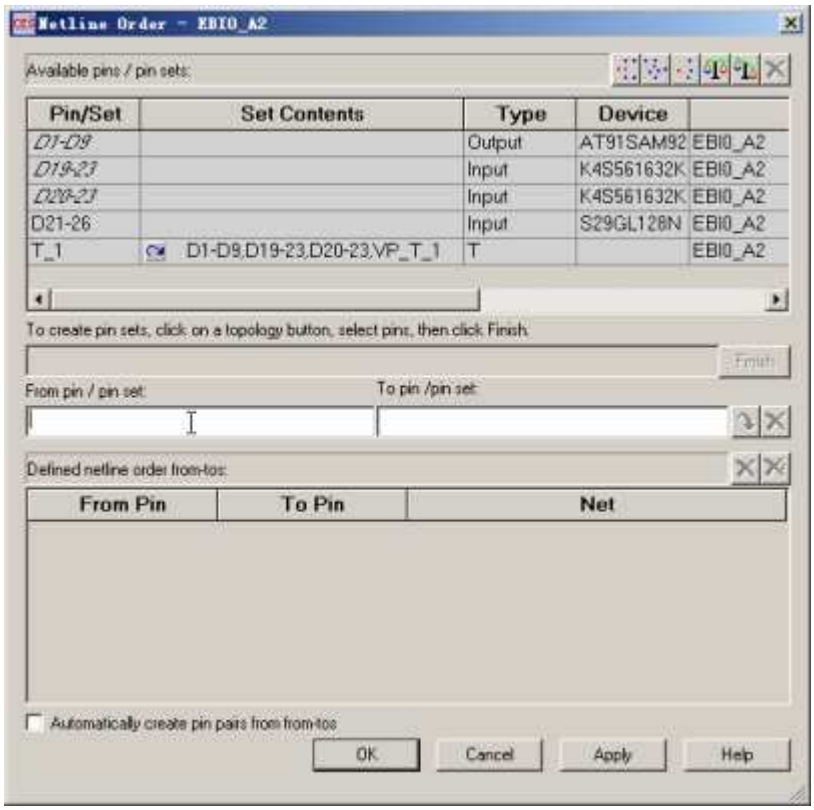


那些 L:VP_T_1_1_1181, L:D19-38 之类的东西就是产生的拓扑结构描述。

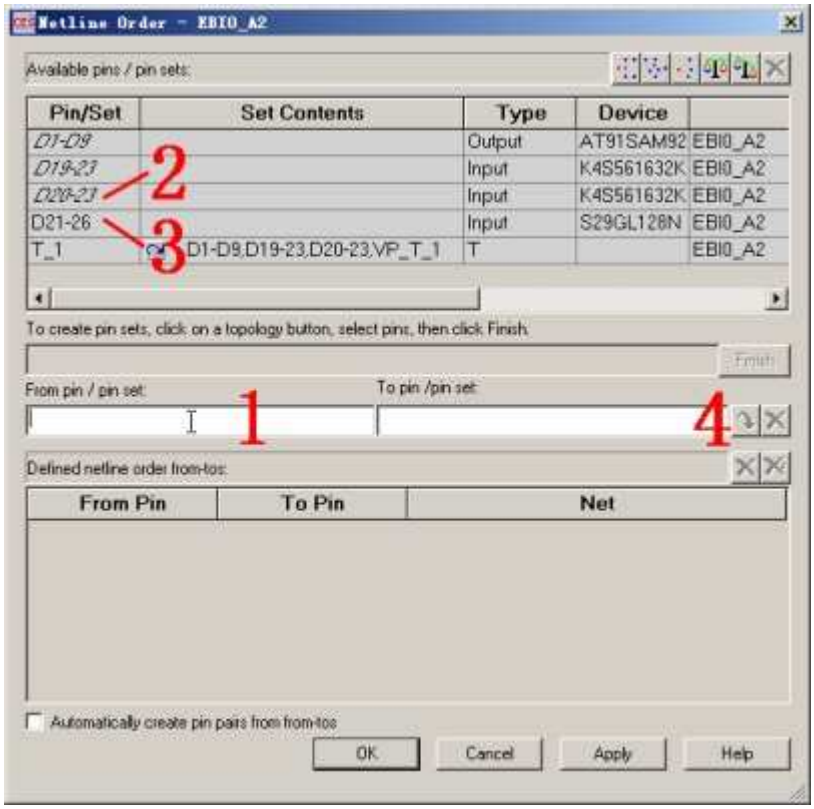
刚才已经提到 SDRAM 的地址线因为还需连接其他器件，拓扑结构的设置还需要有一步添加自定义管脚对。以地址线 A2 举例，同样选择其为 Complex 结构



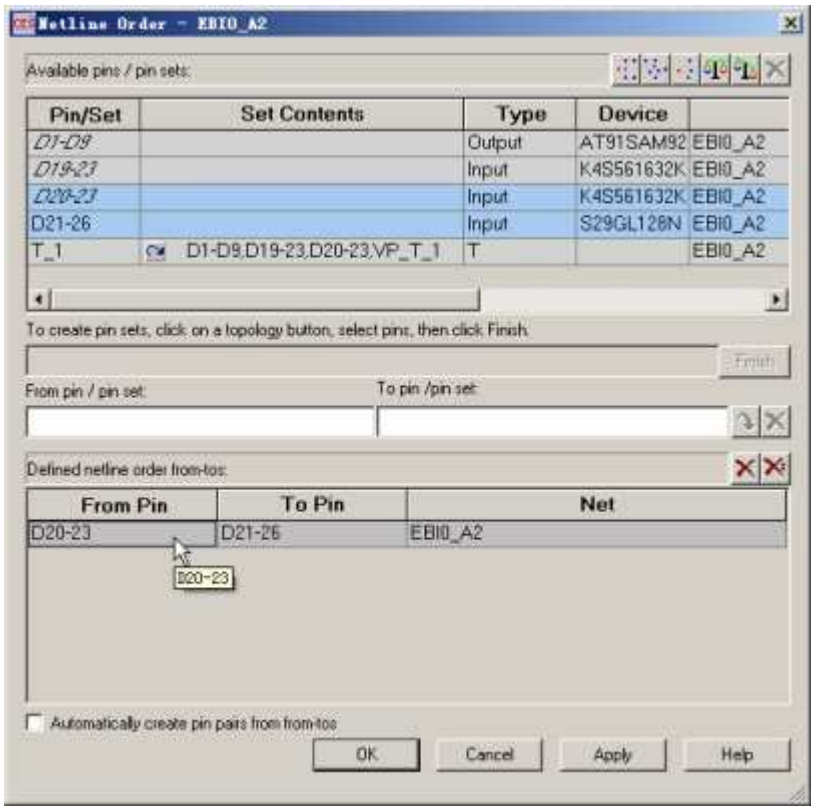
然后选择工具栏上的 Netline order，在出现的对话框中将 CPU 和 SDRAM 之间的连接配置为 Y 型分支，如下图所示



随后配置和 NOR Flash 芯片的连接，先点击“From pin / pin set”下面的文本框，然后依次点击 SDRAM 和 NOR Flash 的管脚，再点击右边的下箭头，如下图



于是一个自定义的 Pin Pair 就出现了，同时管脚列表中对应的管脚背景色也会变蓝，显示这个管脚被指派过了。

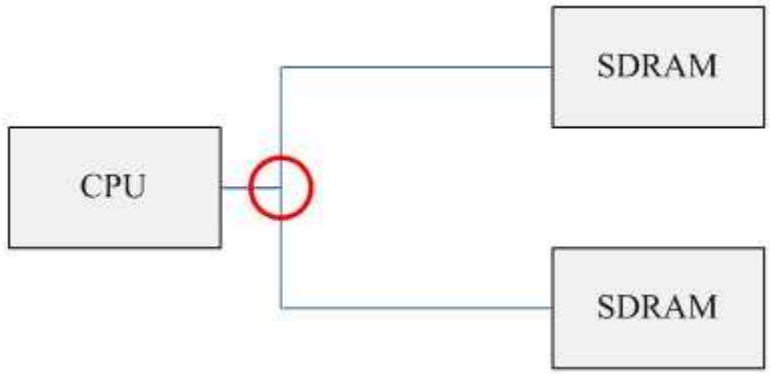


需要注意的是，NOR Flash 连接在那个 SDRAM 器件的管脚上是没有要求的，但是为了走线方便，还是建议连接在理 NOR Flash 相应管脚较近的 SDRAM 器件上。

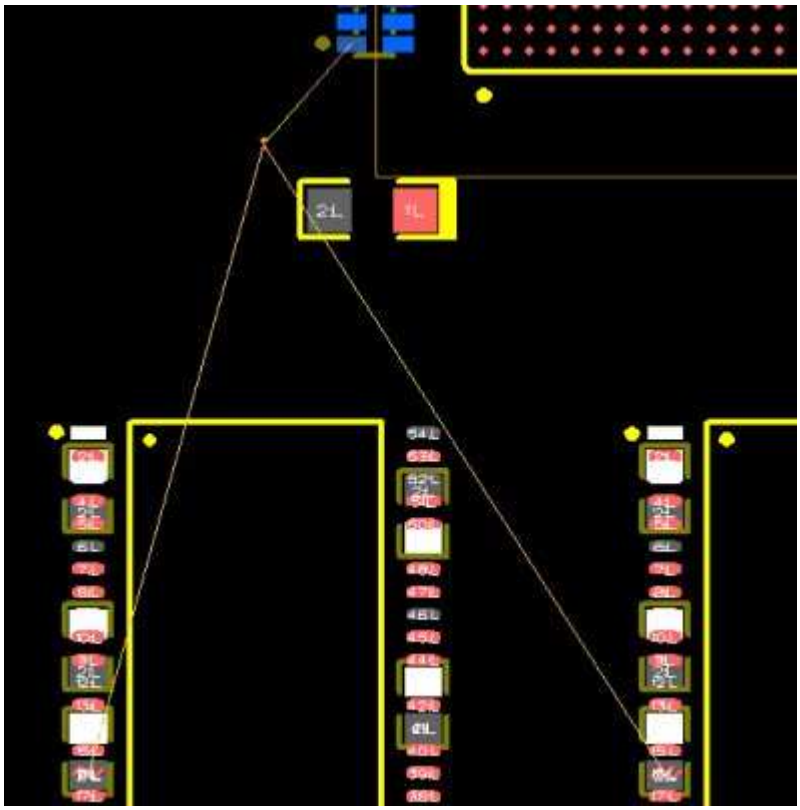
接下来同样选中“Automatically create pin pairs from from-tos”复选框，确定，A2的拓扑结构就配置好了。

	EBI0_A2	Complex	No
	EBI0_A2	Complex	Yes
	L:D20-23,L:D21-26		
	L:VP_T_1_1_1139,L:D19-23		
	L:VP_T_1_1_1139,L:D20-23		
	L:VP_T_1_1_1139,S:D1-D9		

这里还需要提一个概念——Virtual Pin（虚拟管脚），这是 WG 为了方便对拓扑结构的管理而设定的一种虚拟的控制点。还是用图来说明，对于 SDRAM 的连接拓扑结构

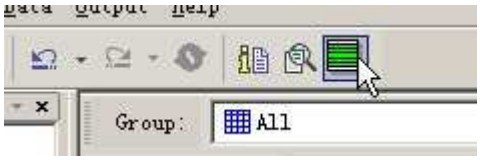


WG 把图中红圈标示的那个分支点分立出来，当作一个可以控制的元素 Virtual Pin，这个元素可以移动、定位；一个 Y 型分支的拓扑结构就拆分成了各个元件到这个 Virtual Pin 连接的结构。刚才我们看到的“VP_T_1_1_1181”就是 Virtual Pin，它在 PCB 设计的时候是这样显示的：

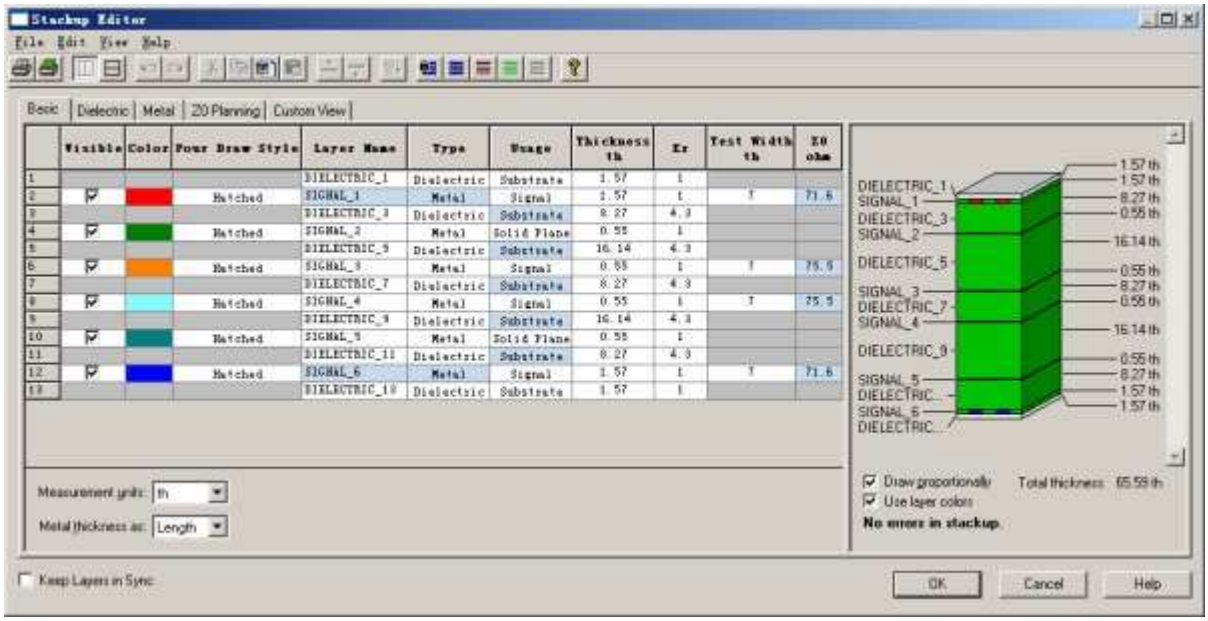


Protel / DXP、PADS 是没有这种功能的，PADS 只能通过 Matchlength Pin Pair 来实现类似的功能，但对于 SDRAM 不太适用。不仅 WG，强大 Alergo 也有这个功能。说了这么多，Virtual Pin 的作用其实就一句话，用来做 Y 型分支的两个分支等长——因为没有 Virtual Pin，去哪里设置等长规则，怎么检验等长的情况？

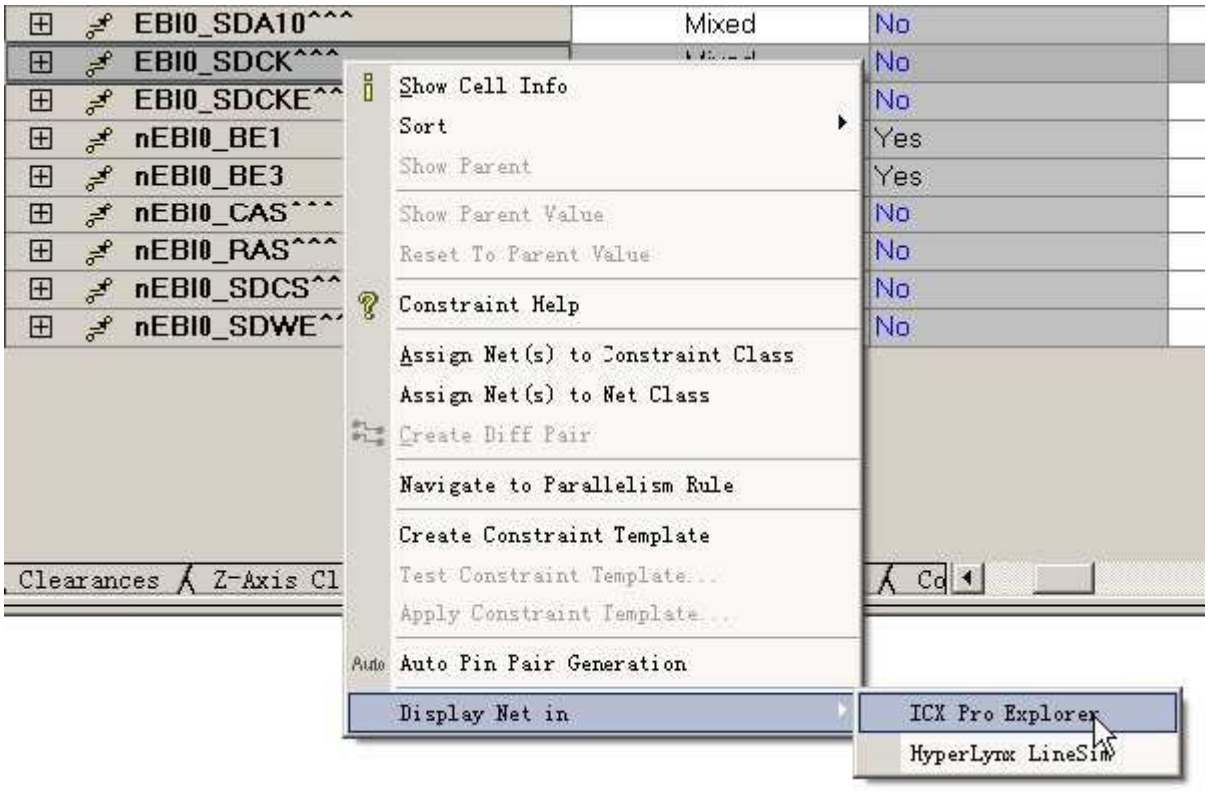
接下来设定 PCB 层叠，不指定 PCB 的层叠，前仿真中的阻抗参数的计算是根本没有意义的。选择 CES 工具栏上的 Stackup Editor



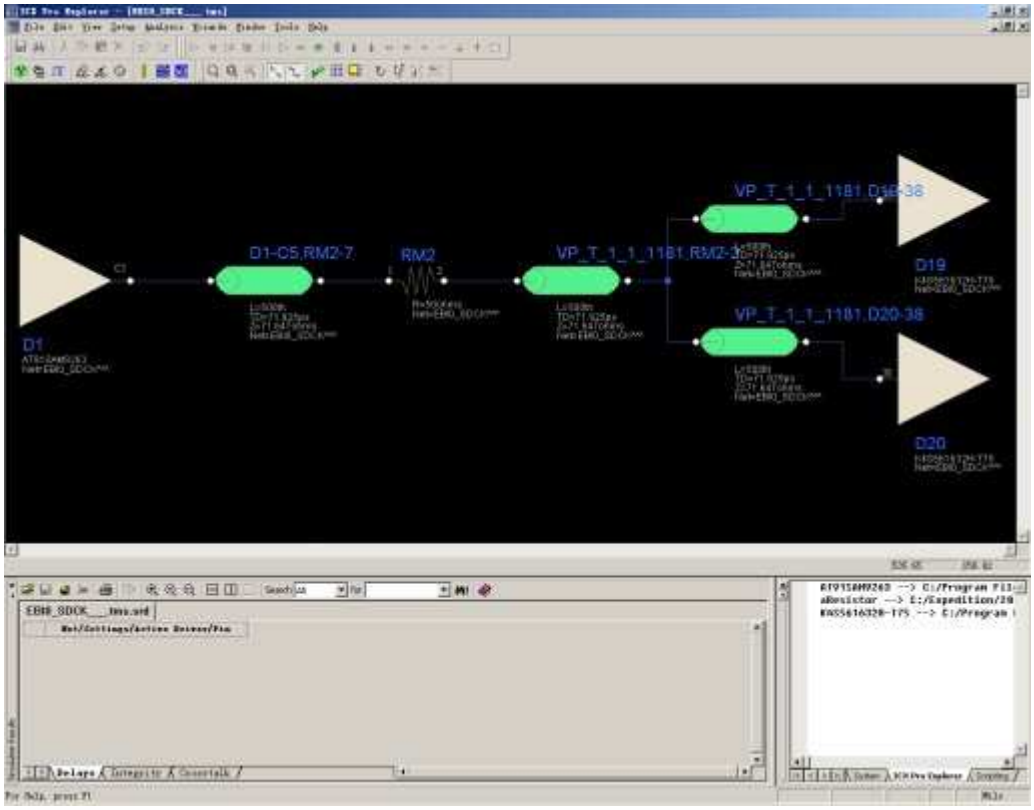
出现下面的窗口，这个东西比较傻瓜式了，我就不做保姆式指导了。需要指定的东西有，PCB 每一层的铜厚，相临两层之间的绝缘层厚度和介电常数，参考面（电源层、地层）是哪几层等。PCB 的厚度和介电常数等参数需要向 PCB 厂家索取，参考面的位置可以 Google 下，看看大多数人用的层叠方法。



有了以上的工作，前仿真就可以进行了，右键选择需要仿真的网络，例如 SDCK，在弹出菜单中选择“Display Net in”——〉“ICX Pro Explorer”

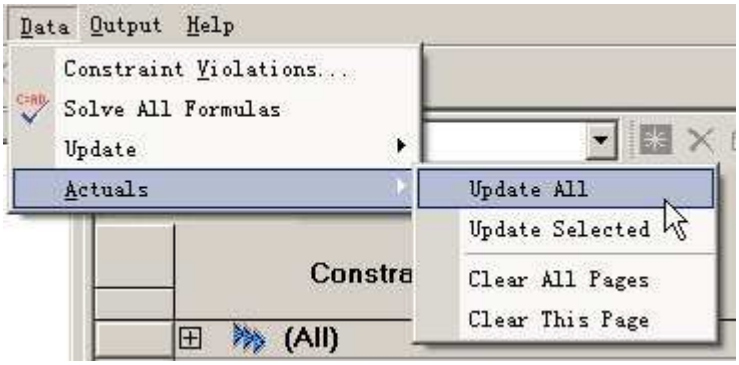


出现以下窗口



这就是我们进行前仿的软件了。可以看到，在中间的黑色背景区域是器件管脚、传输路径以及其他器件（如阻抗匹配电阻）的连接拓扑结构——这也就是我们刚才设定的那些东西。图中绿色那些短棒就是传输路径，双击它可以修改长度；这个长度应由 PCB 布局决定。在 CES

中选择菜单“Data”——〉“Actuals”——〉“Upadate All”命令将 PCB 的数据导入 CES 之后

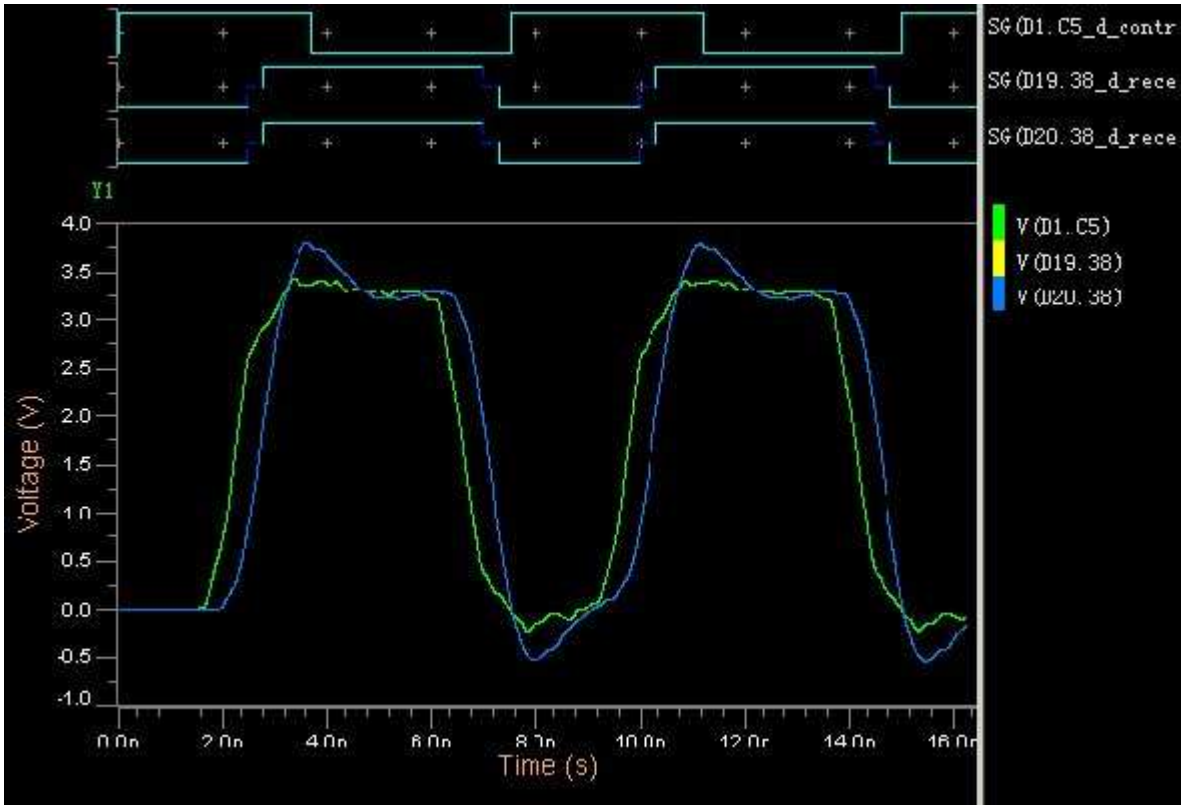


在 Nets 的列表里可以看到 Manhattan Length 这一项的数值出现了

Constraint Class/Net/*	Length or TOF D	
	Manhattan (	Min Length (
EBI0_SDCK^^^	2,818.06	
EBI0_SDCK	2,579.01	
EBI0_SDCK_C5	239.06	
L:VP_T_1_1_1181,L:D19-38	380	359.14
L:VP_T_1_1_1181,L:D20-38	463	442.02
L:VP_T_1_1_1181,L:RM2-2	1,736.01	1,311.51
S:D1-C5,L:RM2-7	239.06	191.41

这个数值的含义是布局完成之后目前 PCB 上这一网络所有飞线（用直线连接管脚的线）长度之和。一般来说把这个数值乘以 1~2 之间的一个数，就是最后的布线长度。你可以把这些数值输入到 ICX Pro Explorer 中去。对于含有拓扑结构的网络，应当把拓扑结构的每一段都输入到相应的传输线上去。你可以在 PCB 上试着移动 Y 型分支点即 Virtual Pin 的位置，然后更新 Manhattan Length，看看是先分支的拓扑结构前仿的效果好，还是后分支的。

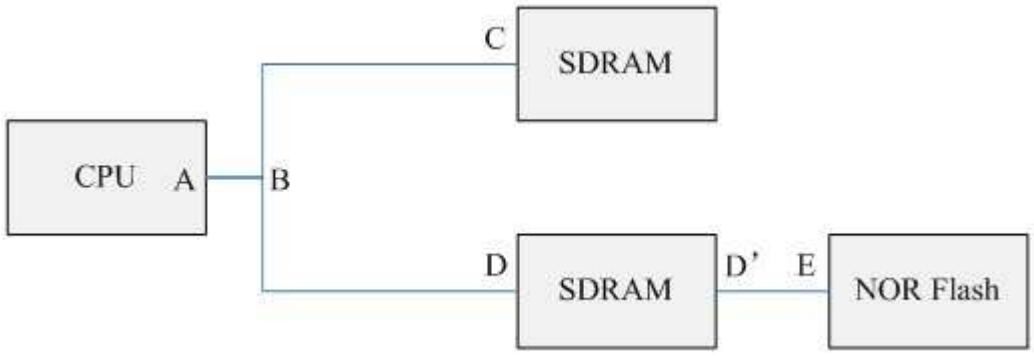
ICX Pro Explorer 也是一个比较傻瓜式的软件，用一用很快就会了，我就不介绍了，给大家看个仿真结果



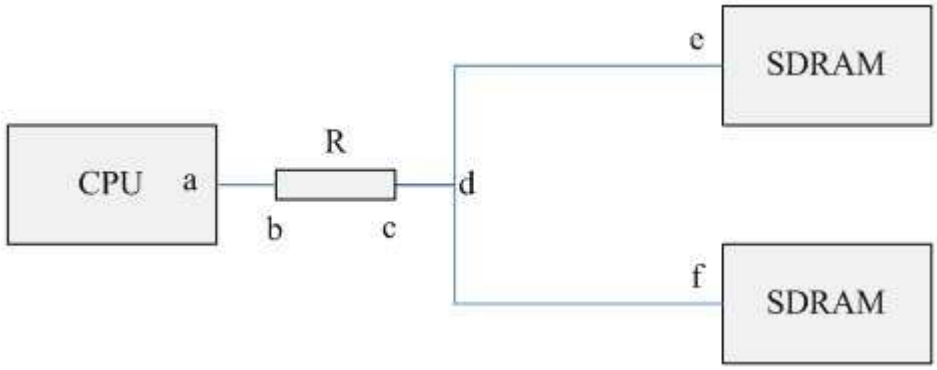
这是 AT91SAM9263 和 K4S561632K-UC75 组成的存储器系统，阻抗匹配电阻为 22 欧，SDRAM 时钟在 133MHz 下的仿真波形。

- WG 中怎么设置 SDRAM 的线长匹配规则

这是一个非常麻烦的部分，我研究了很久，转载请注明来自 EE 小站。在介绍规则设置之前，先介绍下 SDRAM 线长匹配的原则。
一般来说，SDRAM 的地址线是这样的拓扑结构



SDRAM 的控制线是这样的拓扑结构



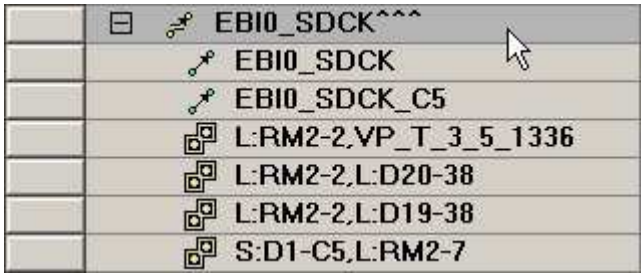
不同地址线之间，需要保证单路分支的长度匹配，即 AB+BC 或 AB+BD 相等；同时，需要保证同一地址线的两个分支相等，即 BC=BD。不同控制线之间，需要保证单路分支的长度匹配，即 ab+cd+de 或 ab+cd+df 相等；同时，需要保证同一控制线的两个分支相等，即 de=df。在控制线和地址线之间，同样需要保证单路分的支长度匹配，即 AB+BC=ab+cd+de 或 AB+BC=ab+cd+df 或 AB+BD=ab+cd+de 或 AB+BD=ab+cd+df。

而软件统计的结果，地址线的长度为 AB+BC+BD+D’E；控制线的长度为 ab+cd+de+df。统计的内容都不一样，怎么做匹配呢？有 2 种方法，使用公式、使用 Pin pair。使用公式的好处是无论有没有阻抗匹配电阻，都可以很清晰的看到线长匹配的误差，这对手工布线非常有帮助，但是过程极其繁琐。使用 Pin Pair 的好处是很简单，不用输入很长的公式，但是对于有阻抗匹配电阻的信号，不能看到匹配误差，需要自己计算。

先说公式方法

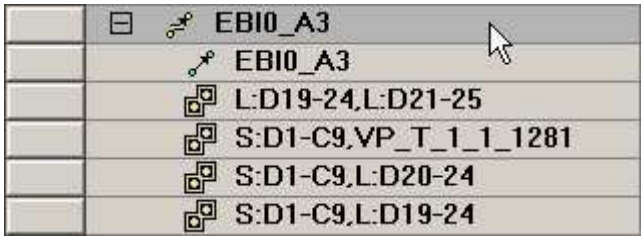
记住，软件统计的是所有线段长度之和；我们需要控制的仅仅是所有地址线的 AB+BC、所有控制线的 ab+cd+de 这些长度都相等（BC=BD、de=df 这两个条件在设定 Y 型分支拓扑结构之后，布线器会自动地做到，于是将上面的那些表达式简化成等价的 AB+BC 和 ab+cd+de 这两个）。因为 SDRAM 的时钟是最重要的信号，选择时钟线的长度作为参考，时钟线的单路分支长度为 ab[时钟]+cd[时钟]+de[时钟]（Blog 没有办法表示下标，用[]来修饰）。则对于其他信号，如地址线，理论上约束的内容应该是 AB[地址]+BC[地址] = ab[时钟]+cd[时钟]+de[时钟]。软件统计的是所有线段长度之和，对于某一地址线，其长度应该是 AB[地址]+BC[地址]+BD[地址]+D’E[地址]。因此，公式约束的内容就变为 AB[地址]+BC[地址] = ab[时钟]+cd[时钟]+de[时钟]+ BD[地址]+D’E[地址]。

很复杂，来看个例子，对于一个含有匹配电阻的时钟信号，EBIO_SDCK，其拓扑结构如下所示



其中 VP_T_3_5_1336 为 Virtual Pin。它的单路分支长度按 WG CES 的语法，应写成 \D1\-\C5\@\RM2\-\7\ + \RM2\-\2\@\VP_T_3_5_1336\-\VP_T_3_5_1336\ + \RM2\-\2\@\D19\-\38\，当然最后一项是\RM2\-\2\@\D20\-\38\也可以。

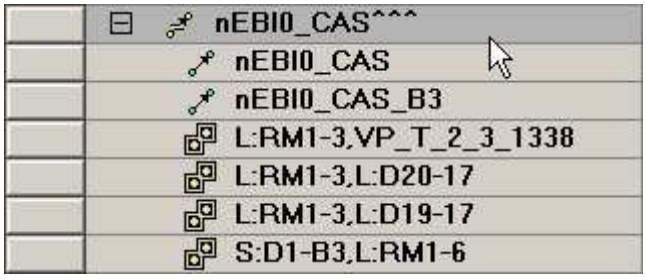
确定了时钟的长度，接下来用它来约束其他地址线、控制线的长度。以 EBI0_A3 为例，其拓扑结构如下所示



由于软件统计 EBI0_A3 的长度为上面 4 个 Pin Pair 线段长度之和，因此在 EBI0_A3 单段分支的基础上（这段长度和 EBI0_SDCK 的单段分支长度相等），需要加上 \D19\-\24\@\D21\-\25\ + \D1\-\C9\@\D20\-\24\这段长度，当然第二项是 \D1\-\C9\@\D19\-\24\也可以。所以限制 EBI0_A3 的长度为，\D1\-\C5\@\RM2\-\7\ + \RM2\-\2\@\VP_T_3_5_1336\-\VP_T_3_5_1336\ + \RM2\-\2\@\D19\-\38\ + \D19\-\24\@\D21\- \25\ + \D1\-\C9\@\D20\-\24\ +/-200th，当然倒数第二项是 \D1\-\C9\@\D19\-\24\也可以，最后的+/-200th 是控制的误差长度。然后把这个公式填到 EBI0_A3 后面的 Formula 中去。

Constraint Class/Net/*	Formulas	
	Formula	*Violation*
EBI0_A3	\D1\-\C5\@\RM2\-\7\...	
EBI0_A3	\D1\-\C5\@\RM2\-\7\...	
L:D19-24,L:D21-25		
S:D1-C9,VP_T_1_1_1281		
S:D1-C9,L:D20-24		
S:D1-C9,L:D19-24		

对于有阻抗匹配电阻的信号，如 nEBI0_CAS



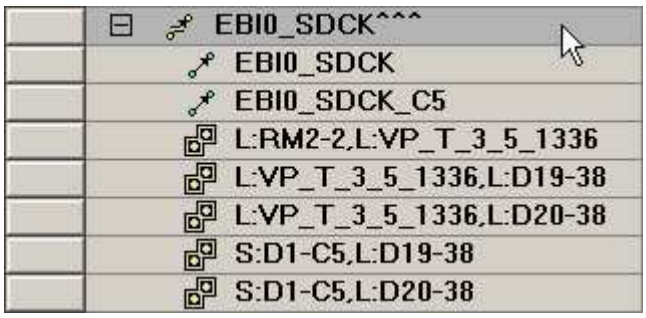
由于软件统计 nEBI0_CAS 的长度为上面 4 个 Pin Pair 线段长度之和，因此在 nEBI0_CAS^^^ 单段分支的基础上（这段长度和 EBI0_SDCK 的单段分支长度相等），需要加上 \RM1\-\3\@\D20\-\17\或\RM1\-\3\@\D19\-\17\的长度。所以限制 nEBI0_CAS 的长度为，\D1\-\C5\@\RM2\-\7\ + \RM2\-\2\@\VP_T_3_5_1336\-\VP_T_3_5_1336\ + \RM2\-\2\@\D19\-\38\ + \RM1\-\3\@\D19\-\17\ +/-200th，当然倒数第二项是 \RM1\-\3\@\D20\-\17\也可以，最后的+/-200th 是控制的误差长度。

按照上面的步骤把所有 SDRAM 信号线都操作一遍，是不是会崩溃？

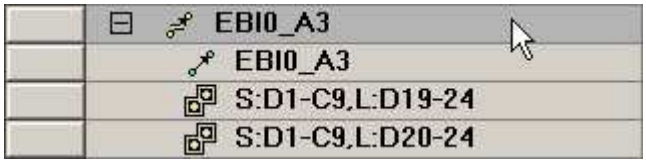
下面介绍用 Pin Pair 的方法

给地址线、控制线建立这样的 Pin Pair，例子如下：

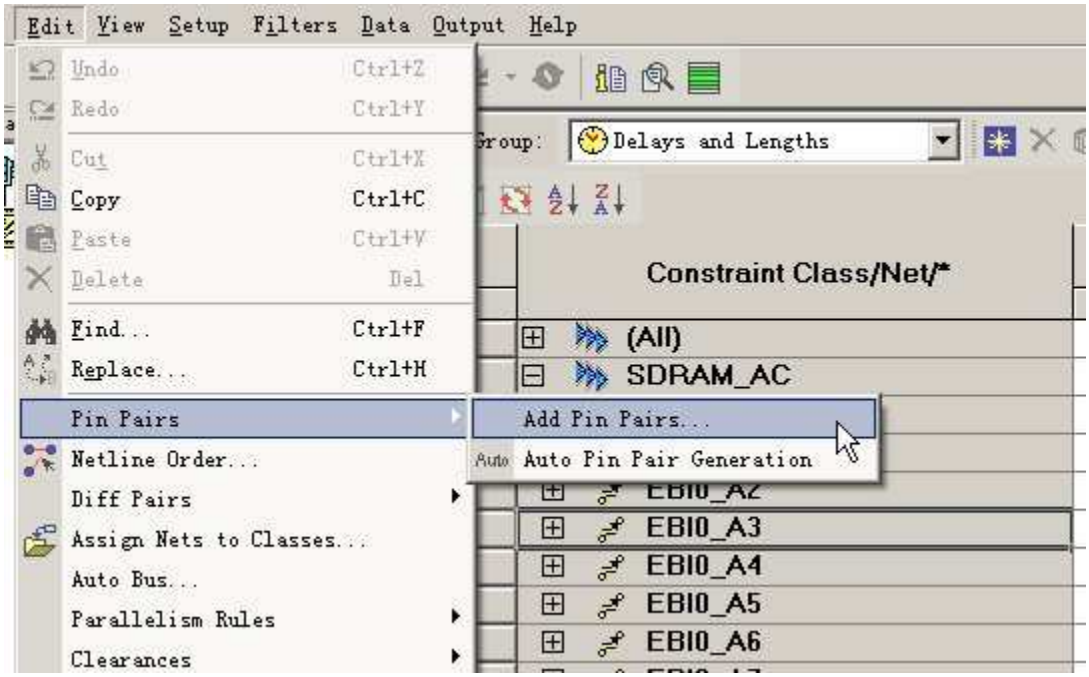
EBI0_SDCK^^^信号

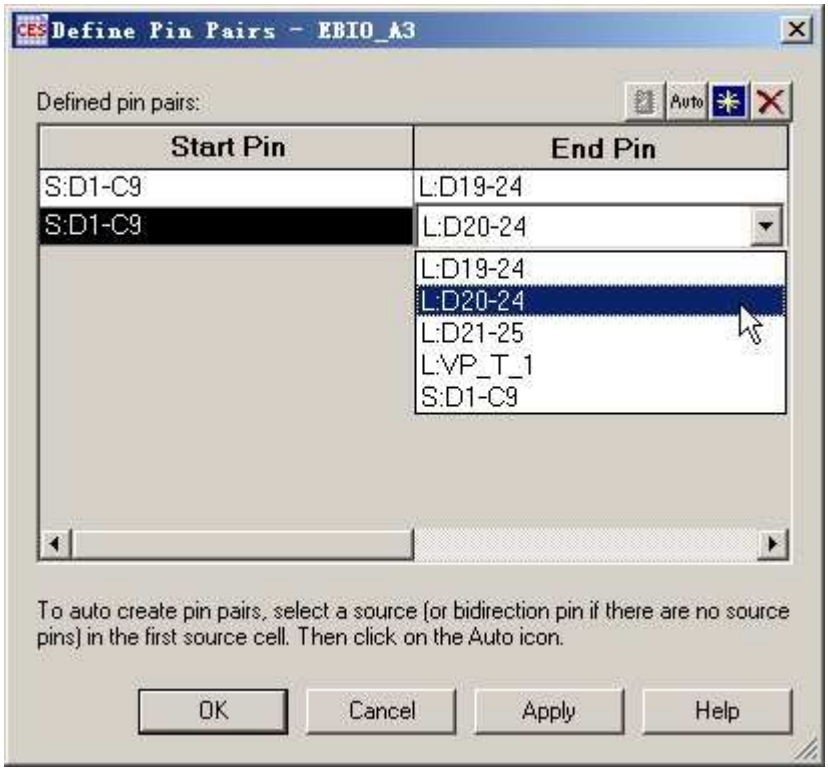


EBI0_A3 信号



这和之前的 Pin Pair 不同，这些 Pin Pair 是手工生成的，也就是说，在制定这些信号的拓扑结构的时候，不选择“Automatically create pin pairs from from-tos”复选框。然后，选中这一信号，通过菜单 Edit>Pin Pair>Add Pin Pairs 来手工添加，如下面两张图片所示。





这样，只需要关心信号最开始是从哪个芯片的哪个管脚出来的，最后到哪个芯片的哪个管脚里去，有多少个单路分支，添加多少个 Pin Pair，中间的阻抗匹配电阻、Virtual Pin 全部可以忽略；最后再在 Pin Pair 的 Match 列用同一名字约束就可以了，如下所示

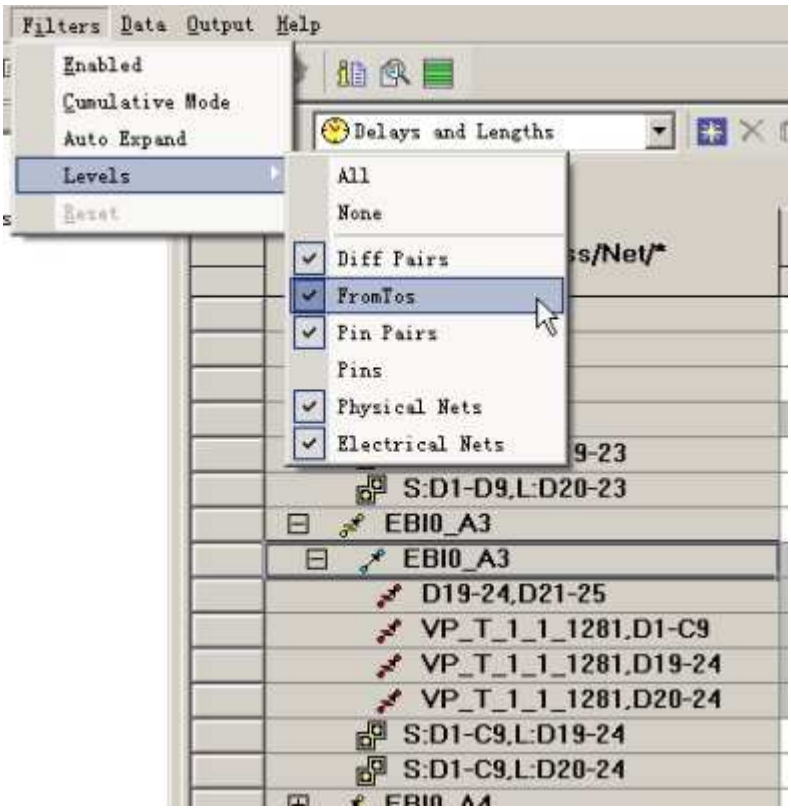
Constraint Class/Net/*	Match	Tol(th)|(ns)
EBI0_A2		
EBI0_A2		
S:D1-D9,L:D19-23	SDRAM_AC	200
S:D1-D9,L:D20-23	SDRAM_AC	200
EBI0_A3		
EBI0_A3		
S:D1-C9,L:D19-24	SDRAM_AC	200
S:D1-C9,L:D20-24	SDRAM_AC	200

控制信号也一样

Constraint Class/Net/*	Match	Tol(th)|(ns)
EBI0_SDCK^^^		
EBI0_SDCK		
EBI0_SDCK_C5		
L:RM2-2,L:VP_T_3_5_1336		
L:VP_T_3_5_1336,L:D19-38		
L:VP_T_3_5_1336,L:D20-38		
S:D1-C5,L:D19-38	SDRAM_AC	200
S:D1-C5,L:D20-38	SDRAM_AC	200

需要说明的是，上面这张图头三个 Pin Pair 是用来方便手工布线的。因为 Pin Pair 如果穿越了器件（对于信号路径穿越阻抗匹配电阻这种情况而言），Pin Pair 的长度在 CES 里是显示不出来的——至少目前我还没有找到什么办法可以让它显示出来——这对手工布线来说非常不方便；但是自动布线却没有任何问题，Expedition 可以正常的完成 Tune（自动长度调整）操作。

也许你有些疑惑，既然 Pin Pair 这么定义了，怎样才能看到自定义拓扑结构呢？把菜单中 Filters>Levels>FromTos 选项钩起来，就可以看见了，如下图所示。需要说明的是，Pin Pair 仅仅是一种虚拟的连接关系，拓扑结构是用 FromTos 这种物理连接关系确定的。



转载请注明来自 EE 小站，以方便后人查询。至于其他的布局、布线这些简单的东西就没有什么好说的了。就写到这里。