



芯和 Channel Expert 通道分析 软件用户手册

产品版本: 2025.02

版权声明

© 2025 Xpeedic 版权所有。该软件和文档包含 Xpeedic 的机密和专有信息。软件和文档按照许可协议提供，仅可根据许可协议的规定使用或复制。未经 Xpeedic 事先书面许可或许可协议明确规定的任何形式或任何方式（包括电子、机械、手动、光学或其他方式）复制、传输或翻译软件和文档的部分内容，Xpeedic 不承担任何责任。

免责声明

Xpeedic 保留修订和改进产品的权利，本用户手册仅描述其出版内容，不反映产品的未来情况。如因不当使用文档而造成任何直接或间接损失，本公司不承担任何责任。

客户支持

若有任何技术问题并要获得芯和公司技术支持，可发送邮件至官方线上技术邮箱 support@xpeedic.com，技术人员将及时给您反馈，也可以电话或者邮件直接联系您所在区域的技术支持人员。

目录

1	简介	4
1.1	软件结构.....	4
1.2	关键技术.....	4
2	软件安装与启动	14
2.1	安装系统要求.....	14
2.2	下载与安装.....	14
2.3	启动软件.....	14
2.4	查看软件版本.....	15
2.5	技术支持.....	15
3	软件总览	15
3.1	UI 布局与界面	15
3.2	通用设置.....	15
3.3	基础操作.....	19
4	器件与功能组件	45
4.1	仿真相关组件.....	45
4.2	SPICE 子电路.....	52
4.3	LUMPED 集总模型	53
4.4	独立源	57
4.5	传输线 TML	65
4.6	IBIS 模型	76
4.7	SYSTEMPI 电源完整性模型	91
4.8	滤波器	92
4.9	受控源模型.....	97
5	结果显示后处理	99
5.1	函数与运算表格.....	99
6	应用场景与案例	103
6.1	S 参数级联仿真	103
6.2	IBIS 通用电路拓扑 TRAN 仿真	109
6.3	DDR 并行链路仿真	116
6.4	用 PDAT 分析工具分析 DDR 最坏情况	174
6.5	MIPI C-PHY 仿真分析	178
6.6	SERDES 眼图仿真分析	188
6.7	COM 通道操作裕度分析.....	222
6.8	用户自定义后处理计算—ICN.....	226
6.9	高级仿真分析.....	230
6.10	BROADSIDE 传输线创建与仿真	256

6.11	CABLE 创建与仿真	264
6.12	行为级滤波器 FILTER 仿真	268
6.13	SERDES PRELYOUT 仿真	270
6.14	XHPC	277
6.15	脚本录制	281
6.16	XTOP 模块仿真	282
6.17	批量仿真网表	288
7	开源软件声明	292

1 简介

本手册提供用户关于 ChannelExpert 软件的功能描述与基本操作说明。

随着云计算、互联网和物联网的快速发展，电子产业在摩尔定律的驱动下，产品的功能越来越强，集成度越来越高，信号的速率越来越快，产品的研发周期也越来越短。由于电子产品不断微小化、精密化、高速化，高速通道设计需要充分考虑从发送端、通孔、传输线到接收端的整个通信链路，并且支持对 buffer 模型（IBIS/AMI）、S 参数、传输线模型和 spice 模型等进行精确仿真。ChannelExpert 能更便捷、更准确地建模和仿真，帮助设计者快速检查关键路径信号完整性指标，如眼图、ISI、浴盆曲线、插入损耗、回波损耗、串扰等。ChannelExpert 内嵌 xspice 求解器，可对包含 s 参数、spice 子电路、ibis 模型、受控源等器件的电路进行时域、频域、直流求解；内嵌卷积与统计求解器支持统计眼图仿真，支持 NRZ 和 PAM-4 调制模式、均衡和预加重技术；内嵌 2D FEM 传输线求解器，可计算 RLGC 参数、损耗、特征阻抗、延时等。灵活参数化与层次化原理图可提供更高效的高级分析功能，可以对仿真项目进行参数扫描分析，OPT，Tuning，DoE，Yield。

1.1 软件结构

以下是 ChannelExpert 的功能结构框图



1.2 关键技术

1.2.1 XSPICE 仿真引擎技术

高速信号完整性仿真面临多个挑战，主要源于高速信号在传输过程中受到的物理限制和复杂的电磁效应。当前芯片设计和制造商主要依赖于国外的 EDA 软件和服务，而本研发项目旨在解决这一难题。通过提升 IBIS 建模与仿真、高速通道复杂物理效应建模与仿真、瞬态仿真引擎优化等关键技术，形成自主的 EDA 仿真平台能力，实现高精度、高效率 and 准确评估的目标，从而满足现代高速电子设计的需求。关键技术突破主要分以下几个部分：



1.2.1.1 S 参数 IFFT 和快速卷积技术

在高速通道信号完整性分析中，通过电磁求解器提取 S 参数具有重要意义。高速信号在传输过程中会受到各种高频效应的影响，如反射、传输线损耗、串扰、频率依赖性衰减和分布参数效应等，这些效应在高频下尤为显著，通过 S 参数能够精确描述这些复杂的高频行为。EM 求解器能够模拟出通道中的所有电磁效应，包括寄生效应、电容、电感、导纳和阻抗等，提取 S 参数（散射参数）可以将这些复杂效应精确地转换为一个统一的参数集，使得后续的仿真更加准确。此外，通过场路联合仿真，工程师可以结合电磁效应和电路特性，实现更全面的系统级分析，优化设计以确保信号完整性和系统性能。这种方法不仅提高了仿真效率和准确性，还便于标准化和模块化处理。

在电路仿真中，S 参数建模和仿真的关键技术在于使用 IFFT（逆快速傅里叶变换）和快速卷积技术。通过这些技术，可以有效地将频域的 S 参数转换为时域响应，显著降低计算复杂度，加快仿真速度。这种方法能够处理复杂的脉冲响应，并在较短时间内生成所需的时域数据，从而提高整体仿真效率和结果的准确性。具体的分析过程如下：

$$B(\omega) = S(\omega)A(\omega)$$

其中， $A(\omega)$ 和 $B(\omega)$ 分别代表端口的入射和反射电压波， $S(\omega)$ 是包含散射参数传递函数的矩阵，描述了每个端口间的传输和反射特性。

在时域内，S 参数 IFFT 转换到时域后，公式为：

$$b(t) = s(t) * a(t)$$

其中，* 表示卷积运算，定义为：

$$s(t) * a(t) = \int_{-\infty}^{\infty} s(t)a(t - \tau) d\tau$$

当时间变量离散化时，当前时间变量 t 表示为时间步长 Δt 的倍数 $M\Delta t$ ，卷积变为：

$$\begin{aligned} b_d(M) &= s_d(M) * a_d(M) = s_{d0}a_d(M) + h_d(M) \\ &= s_d(0)a_d(M)\Delta t + \sum_{k=1}^M s_d(k)a_d(M-k)\Delta t \end{aligned}$$

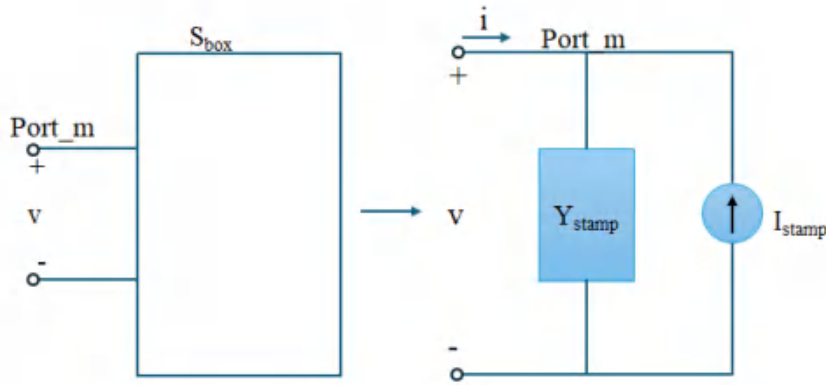
入射波和反射波可以通过端口电压和电流表示为：

$$\begin{aligned} a(t) &= \frac{1}{2}[v(t) + Z_0 i(t)] \\ b(t) &= \frac{1}{2}[v(t) - Z_0 i(t)] \end{aligned}$$

代入上述 S 参数离散化公式，我们可以得到，

$$[1 + s_{d0}]Z_0 i(t) = [1 - s_{d0}]v(t) - 2h(t)$$

因此，我们可以得到 S 参数 IFFT 转换到时域后每个 port 在电路方程中的求解形式，



S 参数的端口等效电路模型

$$Y_{stamp} = Z_0^{-1}[1 + s_{d0}]^{-1}[1 - s_{d0}]$$

$$i_{stamp} = 2Z_0^{-1}[1 + s_{d0}]^{-1}h(t)$$

其中， $h(t)$ 的计算可以通过快速卷积获得，从而加快仿真速度。实际应用当中还需要引入窗函数抑制频谱泄露，并考虑无源性、时延和因果性等校正。

1.2.1.2 传输线 Line Model and Fast Convolution

除了 S 参数，传输线模型在高速信号通道模型中也扮演重要角色。在需要进行时域仿真（如信号完整性分析和时域反射测量）时，传输线模型更为直接和有效。对于长距离传输线，传输线模型能够更好地捕捉传输线效应，如反射和延迟。使用 EM 求解器进行传输线模型提取时，通常采用传输线矩阵方法（TLM）或分布参数提取技术，得到传输线的分布参数（R、L、G、C）。在一些复杂的高速电路设计中，可能需要结合使用 S 参数和传输线模型。例如，对于高速信号通道，可以使用 S 参数来描述互连和连接器的高频行为，同时使用传输线模型来描述 PCB 走线和电缆的时域特性。为了提高传输线模型的仿真精度和效率，通过对传输线的多导体模型进行细致建模，考虑频率依赖性和非线性效应，利用电磁理论和传输线方程，建立精确的 Line model 模型如下：

$$i_1(\omega) = Y_c(\omega)v_1(\omega) - 2i_{b1}(\omega)$$

$$i_2(\omega) = Y_c(\omega)v_2(\omega) - 2i_{f2}(\omega)$$

其中, $Y_c(\omega)$ 为特征阻抗矩阵, $i_{b1}(\omega)$ 为后向电流, $i_{f2}(\omega)$ 为前向电流,

$$i_{b1}(\omega) = W(\omega)[i_2(\omega) + i_{f2}(\omega)]$$

$$i_{f2}(\omega) = W(\omega)[i_1(\omega) + i_{b1}(\omega)]$$

其中, $W(\omega)$ 为传播函数矩阵, 对于 $Y_c(\omega)$ 和 $W(\omega)$, 我们可以通过 vector fitting 得到如下拟合形式:

$$H(s) = H_\infty + \sum_{m=1}^M \frac{A_m}{1 + s/\omega_{cm}}$$

相对应的时域表达式如下:

$$H(t) \simeq H_\infty + \sum_{m=1}^M H_m(t) = H_\infty + \sum_{m=1}^M A_m \omega_{cm} e^{-\omega_{cm} t}$$

假设输入 $x(t)$ 是分段线性连续的, 那么响应 $y(t)$ 应该如下:

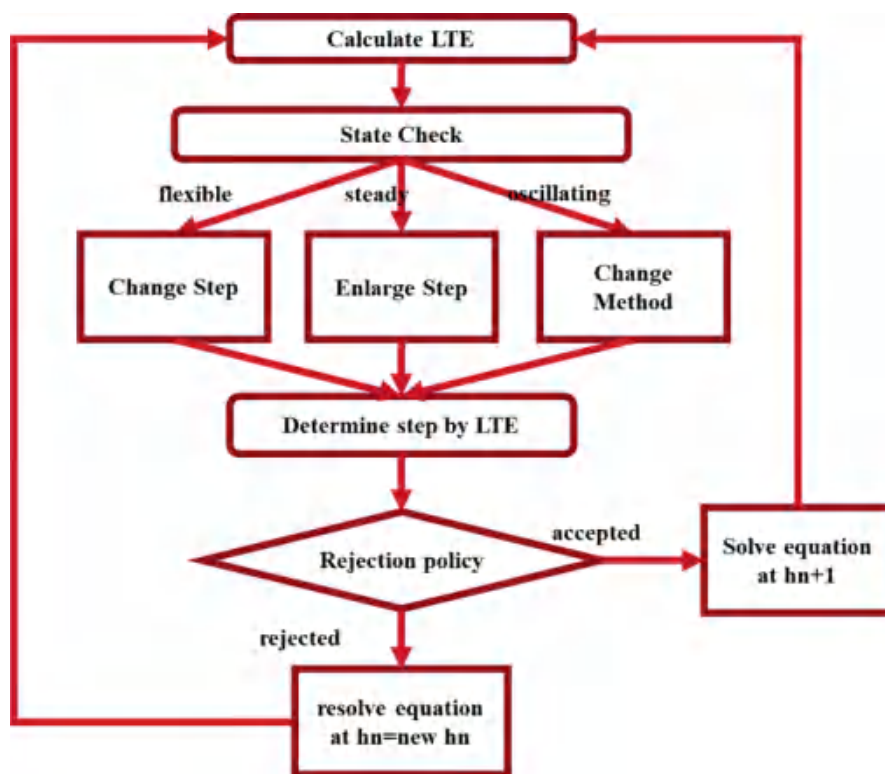
$$y(t_n) = H_0 x(t_n) + \sum_{m=1}^M D_m(T_n) x(t_n) - \sum_{m=1}^M z_m(t_n)$$

$$z_m(t_n) = Dm(T_{n-1})e^{-\omega_m T} - Dm(T_n)x(t_{n-1}) + e^{-\omega_m T} z_m(t_{n-1})$$

以上即为 recursive convolution 的表达式。

1.2.1.3 瞬态仿真引擎优化

除了以上对整个高速链路的建模和仿真加速, 瞬态仿真引擎方面也引入了基于误差的自适应步长控制, 并结合时间点压缩, 振荡检测和抑制等技术, 提升了瞬态仿真引擎的收敛性、精度和求解速度。



时间步计算的流程如图所示：

- 1、根据不同的积分方法计算不同的 LTE 值。
- 2、检测电路的状态。
- 3、根据不同的状态，采取不同的步长策略
- 4、根据 LTE 的值以及不同的策略确定步长。
- 5、根据拒绝策略判断是否拒绝，如果拒绝，则按照新的时间步重新求解电路方程，否则，接受当前时间步且开始下一个时间点的仿真。

提出估算频域器件的 LTE 的新方法。

$$\frac{\partial^2 s(t)}{\partial t^2} = \frac{\sum_{j=0}^N h(j\Delta t) \dot{x}(t_N - j\Delta t)\Delta t - \sum_{j=0}^{N-1} h(j\Delta t) \dot{x}(t_{N-1} - j\Delta t)\Delta t}{\Delta t}$$

$$= \frac{h(N\Delta t)\dot{x}(t_N - N\Delta t)\Delta t + \sum_{j=0}^{N-1} h(j\Delta t) (\dot{x}(t_N - j\Delta t) - \dot{x}(t_{N-1} - j\Delta t))\Delta t}{\Delta t}$$

引入状态检测算法，在不同时刻确定信号的状态。将不同时刻的信号分为三种状态。针对不同的状态采取不同的策略。

变化：信号处于常规变化状态，则通过上述的 LTE 算法控制步长。一些非线性强的点通过上述正则策略进一步控制步长。

稳态：信号的状态已经稳定。如阶跃信号过了激励的时刻。此时可以适当放开步长。

振荡：信号的状态处于非常规的振荡状态。这种振荡往往表明，信号已经计算错误，整体仿真已经不收敛。此时缩小步长或者扩大步长已经无法控制振荡。此时可能需要更换矩阵求解的收敛条件或者积分方法。

状态检测算法：参考通信中的自相关函数的定义。检测不同时刻下信号的相关性。

离散相关性函数： $R_x[t_i] = \sum_{n-k}^n \omega_i * \text{abs}(x_i - x_{i+k})$

1.2.1.4 求解器技术

稀疏矩阵求解器：运用高效稀疏矩阵求解算法，解决电路仿真中大规模线性或非线性方程组，显著提高仿真效率。该求解器广泛适用于各类仿真任务（如瞬态、DC、AC、PSS），共享核心算法模块，可加速多种仿真场景。

线性与非线性求解器：基于 LU 分解、SVD 分解等高效算法构建线性求解器，结合牛顿迭代、同伦法等求解非线性问题。作为核心计算组件，可复用于多种仿真任务，提升复杂电路分析效率，降低计算成本。

并行和分布式仿真技术：基于多核 CPU 和 GPU 的并行计算，以及分布式集群仿真技术，适用于大规模电路和多物理场联合仿真。能够满足百万级节点的大规模电路仿真、高速信号分析以及多物理场仿真任务的需求，可显著降低仿真时间，提高效率。

宏模型与降阶建模技术：通过简化复杂电路（宏模型）和降阶技术（如 Ticer、Pade 近似），减少求解规模，提升大规模电路系统的仿真效率。广泛应用于系统级仿真和功能验证，特别适合 SoC 级别复杂设计和电源网络的快速分析，可复用于多种仿真场景。

1.2.2 统计眼图仿真技术

统计眼图是一种利用统计方法分析信号完整性的工具。相较于传统的瞬态分析，统计眼图能够以较小的计算量获得较低的误码率，同时也可以考虑均衡以及 jitter、串扰的影响。

统计眼图基于线性时不变系统（LTI）的两条基本性质：叠加性和平移性。通过模拟信号在通道中的传输，计算信号的冲击响应，进而得到眼图的统计特性。主要包含以下几个步骤：

计算通道的冲击响应：获取通道的 S 参数或上升/下降边沿响应。

均衡处理：叠加线性均衡 CTLE/FFE、非线性均衡 DFE 改善通道信噪比。

生成统计眼图：通过蒙特卡洛、快速卷积算法生成眼图的分布。

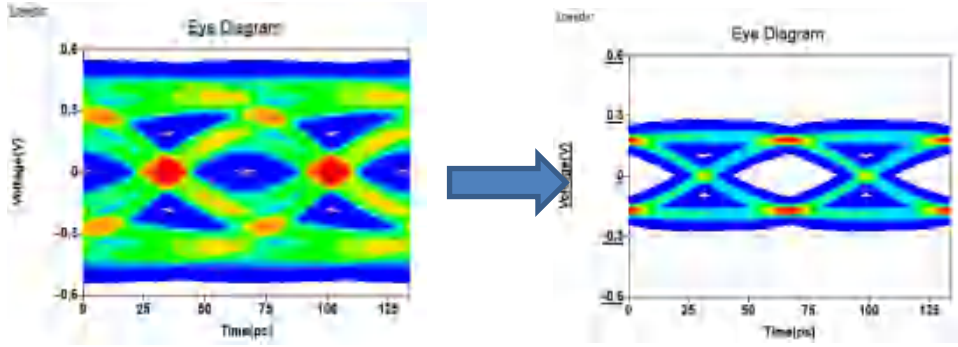
眼图测量：提取眼图的关键参数，如眼高、眼宽、抖动等。

显示眼图：以 2D 或 3D 形式展示眼图，以及浴缸曲线和等高线图

其中 FFE 均衡技术，以一个三抽头为例，计算公式如下所示， C_{-1} 是 pre tap， C_0 、 C_1 是 post tap，可同时调节码前、码后的 ISI，主信号幅度会受影响，但信噪比增大。

$$TX_{FFE}(n) = C_{-1}TX(n+1) + C_0TX(n) + C_1TX(n-1)$$

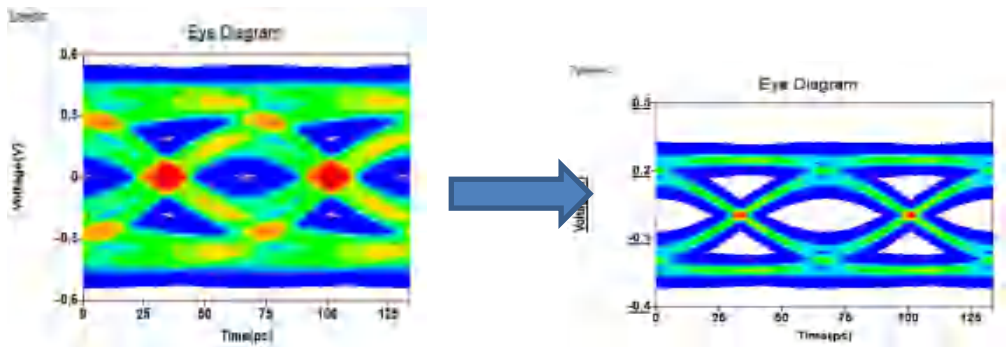
眼图均衡效果如下



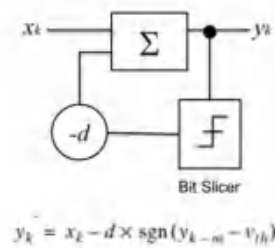
CTLE 均衡，主要是提升高频信号的损耗，减少高频对 ISI 的影响，根据零极点个数以及增益的形式有很多种结构，下面以两极点一零点为例，公式如下， w_{p1} 、 w_{p2} 是两个极点， w_z 是零点， A_{dc} 是 DC 增益。

$$H(s) = \frac{w_{p1} * w_{p2}}{w_z} * \frac{s + A_{dc} * w_z}{(s + w_{p1}) * (s + w_{p2})}$$

眼图均衡效果如下：



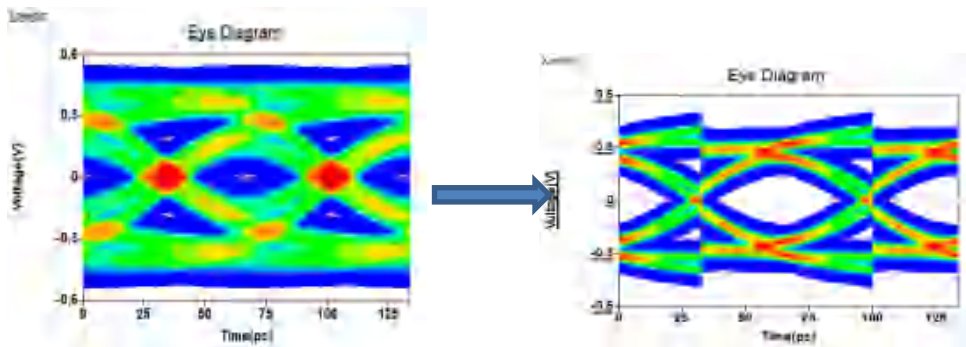
DFE 均衡是一个非线性均衡，和 FFE 比较类似，只有 post tap，且主 UI 信号也不做均衡，其中另外增加了一个数字信号判决器。如下图所示，Bit Slicer 就是判决器， V_{th} 是判决电压， d 是抽头系数，可以有多个。



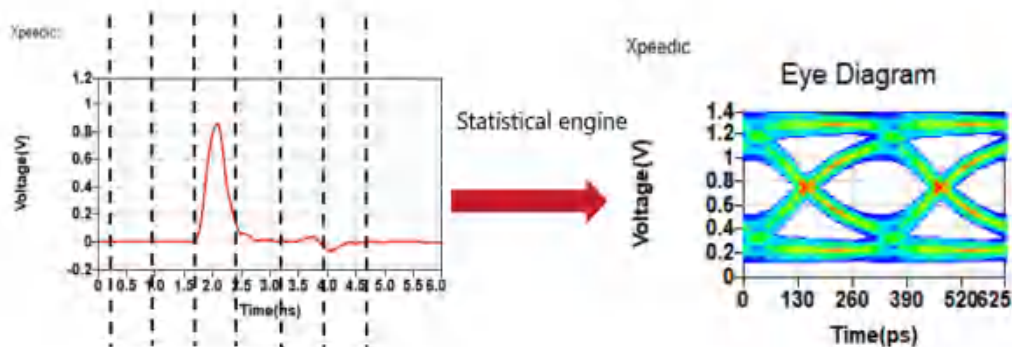
具体计算公式如下：

$$RX_{DFE}(n) = RX(n) - \sum_{i=1}^{N_b} C_i \times \text{sgn}(RX(n - i))$$

眼图均衡效果如下：



眼图分布的计算，如下图，将一个单比特的主 UI 信号去叠加所有的平移反转带来的 ISI 信号，虚线就是以主 UI 信号为中心往外划分的其它 UI，也就是 ISI 信号，这里是 bit‘1’的响应曲线，还有 bit‘0’的响应曲线，相应的 ISI 需要取相反数。

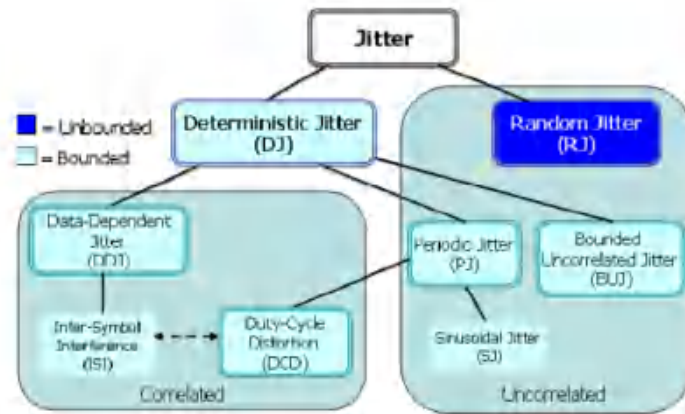


计算公式如下：

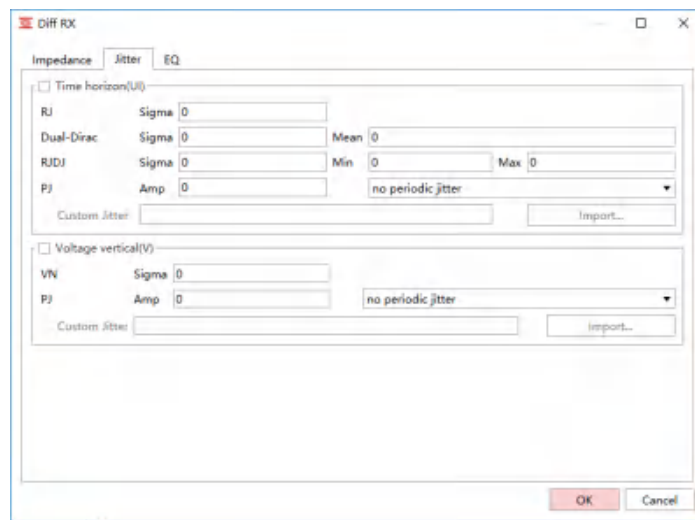
$$S(t) = V_0(t) + \sum_{i \neq 0} V_i(t - i * T) * d_i$$

$V_0(t)$ 是主 UI 信号， $V_i(t - i * T)$ 是 ISI 信号， d_i 有两个值，1 和-1，概率各为 50%，需要统计出所有叠加出的电压的可能性，假设 ISI 有 100 个，那么叠加的电压就有 2^{100} 种，计算量非常大，耗时也是一个天文数字，几乎不可能完成。这里面需要设计算法，用到快速卷积算法，一个一个 ISI 的去卷积，再去统计概率，可以达到 $O(n \log n)$ 的运算速度，大大节省计算时间。

Jitter 的注入，实际系统中不可避免的会引入 jitter，分为两大类，水平的 jitter 就是时间的抖动，以及垂直抖动，就是电压噪声。具体又包括确定性抖动和随机抖动，细分如下图所示，有随机抖动 RJ、确定性抖动 DDJ、PJ、BUJ、DCD 等。



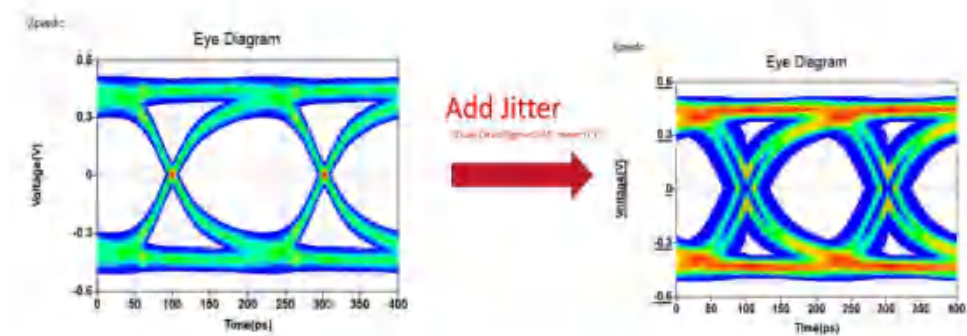
通过概率分布对抖动建模，如下图一些期望和方差等参数进行概率分布计算。



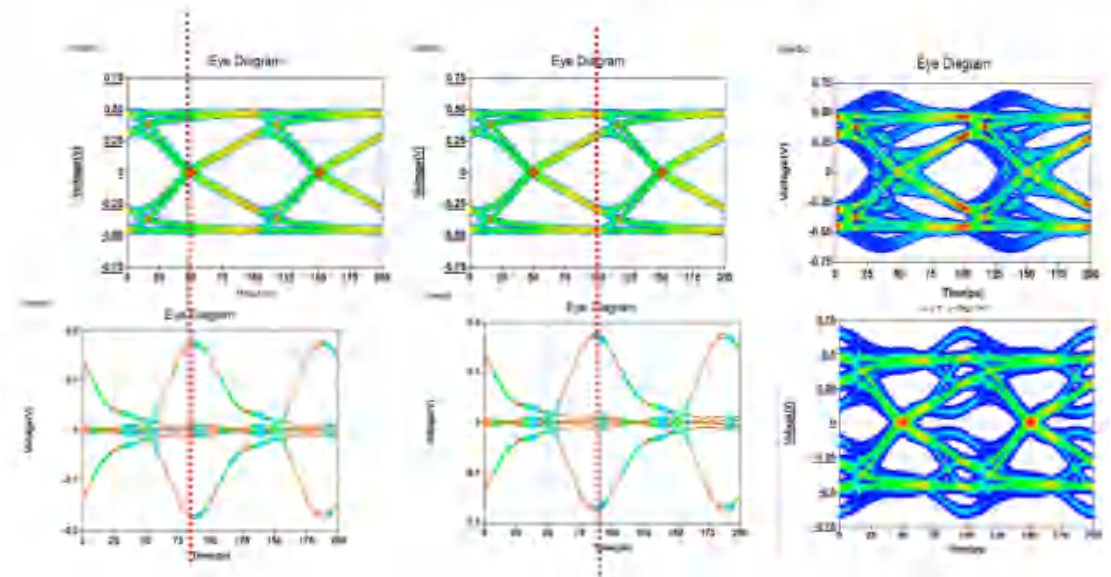
再通过下面随机变量叠加的概率分布计算公式，和眼图概率分布卷积，从而注入 jitter。

$$f_{X+Y}(z) = \int_{-\infty}^{\infty} f_X(x) * f_Y(z-x) dx$$

下图是一个 Dual-Driac 的 jitter 的注入：



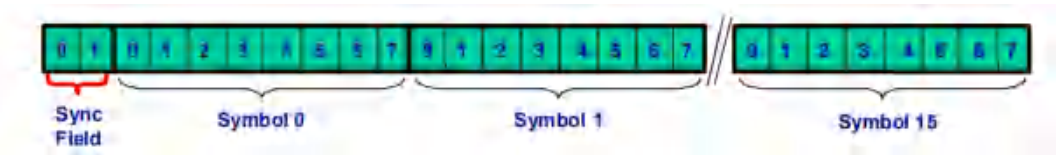
串扰的叠加，和 jitter 类似，它是在垂直方向的叠加，同样采取卷积的方式，和 jitter 的公式类似。串扰和主通路的相位差也影响很大，相位不一样叠加效果也不一样，如下图所示。



很明显，相位差也可能会使眼图变得更差，实际系统中相位可能随机，这就需要仿真去考虑这种情况，算法内部会去反向优化最差的 phase，一般最差的 phase 是串扰眼图的峰值的位置与主信号位置对齐，再基于最差的 phase 重新去均衡仿真，最大限度去拯救眼图的质量。

另外在实际系统中还需要用到编码技术，来改善信号，平衡直流，通用的编码方式有 8b/10b,64b/66b,128b/130b,128b/132b 等，下面以 128b/130b 为例来讲述。

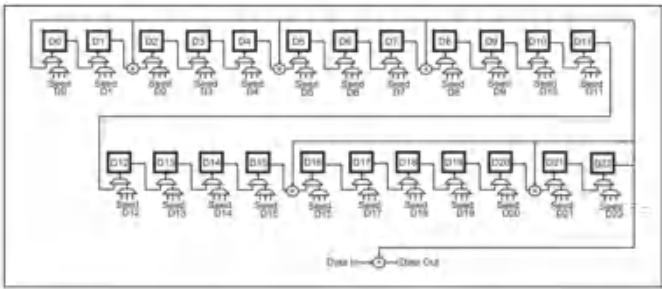
128b/130b 是控制数据集与基于扰码机制的编码方式，编码流程如下图所示：



16 个 8bit 数据，和两个同步位数据位。

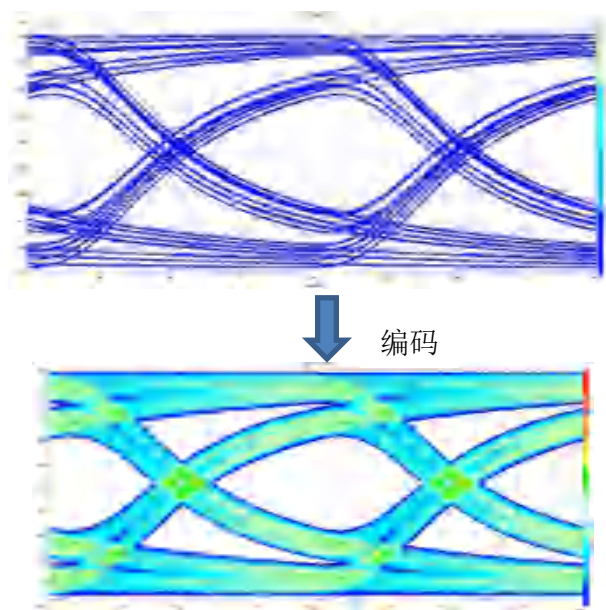
加扰方式，多个 LFSR，每个通道实现一个单独的 LFSR。G(X)是加扰码的本原多项式。

Lane	Seed Value
0	1DBFBCh
1	0607BBh
2	1EC760h
3	18C0DBh
4	010F12h
5	19CFC9h
6	0277CEh
7	1BB807h



$$G(X) = X^{23} + X^{21} + X^{16} + X^8 + X^5 + X^2 + 1$$

眼图的仿真效果如下，编码器前后对比。



2 软件安装与启动

2.1 安装系统要求

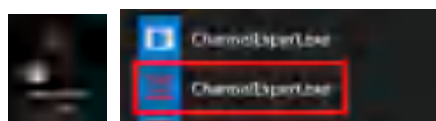
ChannelExpert 软件在 64 位 Windows 和 Linux 计算机系统平台上均可以稳定运行，目前支持 Windows 7 SP1、Windows 8.1 KB2999226 、Windows 10 和 Linux RedHat 6 系统及更高版本系统。

2.2 下载与安装

ChannelExpert 软件安装包可在芯和公司官方网站下载：<https://www.xpeedic.com>，或者联系芯和技术人员获取软件安装包。软件安装许可证书必须获得芯和公司官方授权否则无法正常安装，在软件安装过程中有任何问题可联系技术支持人员。

2.3 启动软件

ChannelExpert 软件安装成功后，在电脑桌面双击软件快捷图标即可启动 ChannelExpert 软件，或者在开始界面的 Xpeedic 安装文件下打开 ChannelExpert.exe 文件也可以启动 ChannelExpert 软件。



2.4 查看软件版本

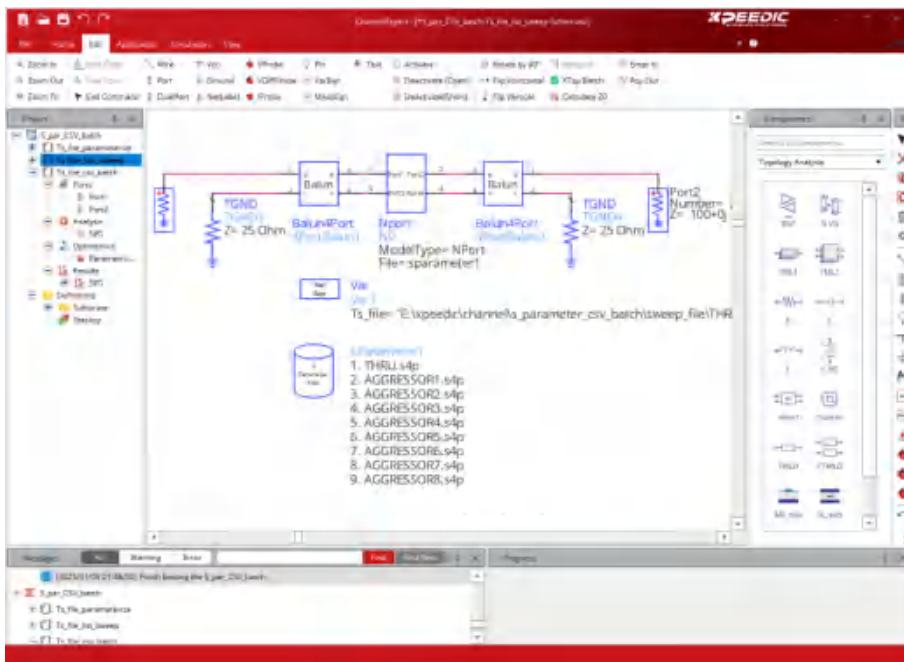
打开 ChannelExpert 软件点击右上角的问号可显示当前软件版本编号。

2.5 技术支持

若有任何技术问题并要获得芯和公司技术支持，可发送邮件至官方线上技术邮箱 support@xpeedic.com，技术人员将及时给您反馈，也可以电话或者邮件直接联系您所在区域的技术支持人员。

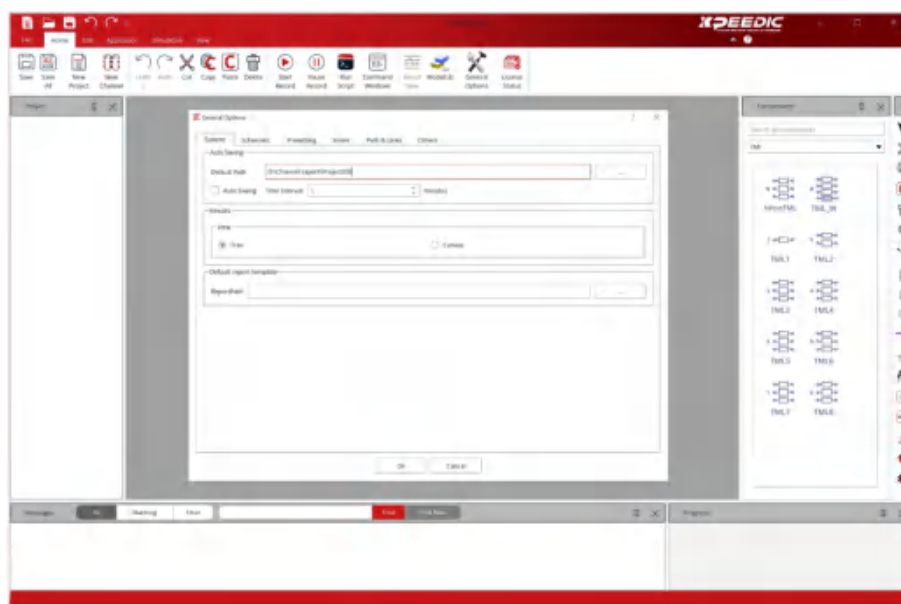
3 软件总览

3.1 UI 布局与界面

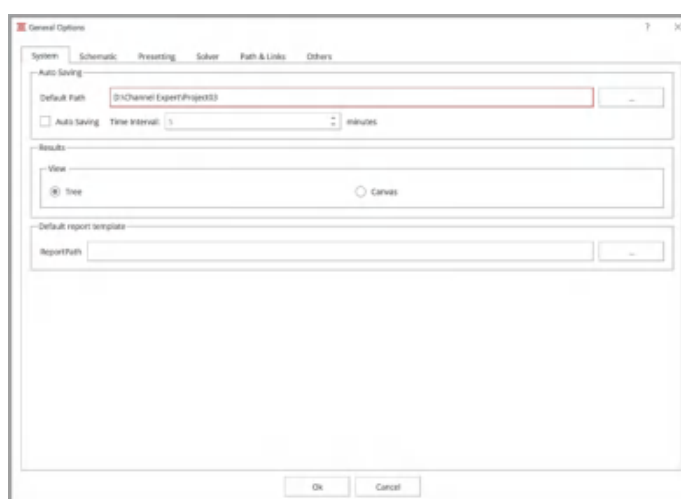


3.2 通用设置

主界面菜单点击 General Options 界选项，弹出各系统配置选项：



3.2.1 System 窗口



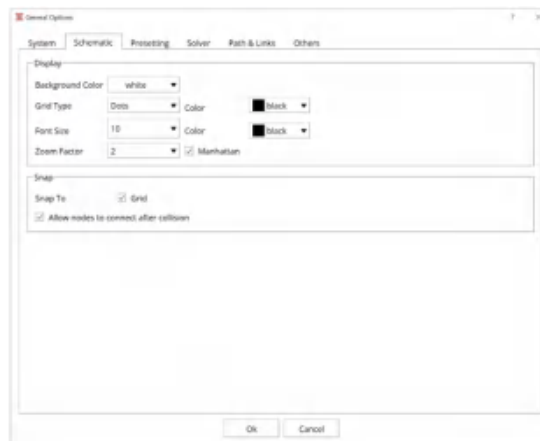
Default Path: 创建工程的默认路径，可通过“...”处进行更改；

Auto Saving: 勾选后触发自动保存功能，自动保存时间间隔在 Time Interval 内设置；默认为 5 分钟；

View: 仿真结果的两种挂载形式，Tree 为在主界面树节点下挂载仿真结果，Canvas 为在后处理 ResultView 界面挂载仿真结果；

Default report template: 工程模板导出路径，可通过“...”处进行更改。

3.2.2 Schematic 窗口



Display 信息页: Back Ground Color: 原理图背景色, 默认为白色;

Grid Type: 原理图网格参照方式, 默认为点, 颜色为黑色;

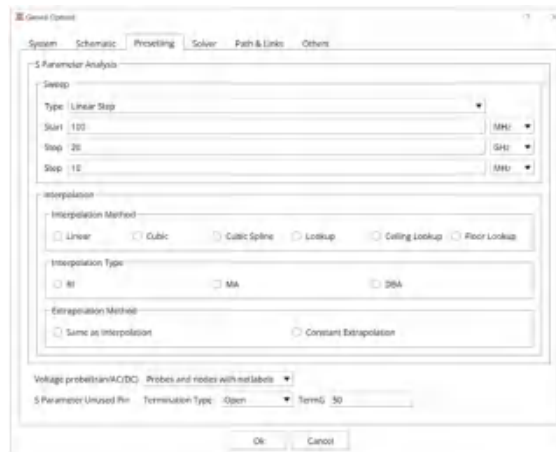
Font Size: 原理图内文本的字体大小, 默认大小为 10, 字体颜色为黑色;

Zoom Factor: 原理图内使用滚动轮放大缩小原理图的速度; Manhattan 选项为原理图内连接线的的方式, 勾选后只能为直线, 不能为斜线;

Snap To: 勾选 Grid 选项后, 连接线的移动, 只能半格半格移动;

Allow nodes to connect after collision: 勾选后当器件的 pin 脚相触碰时, 会自动连接。

3.2.3 Presetting 窗口



S Paramrter Analysis:

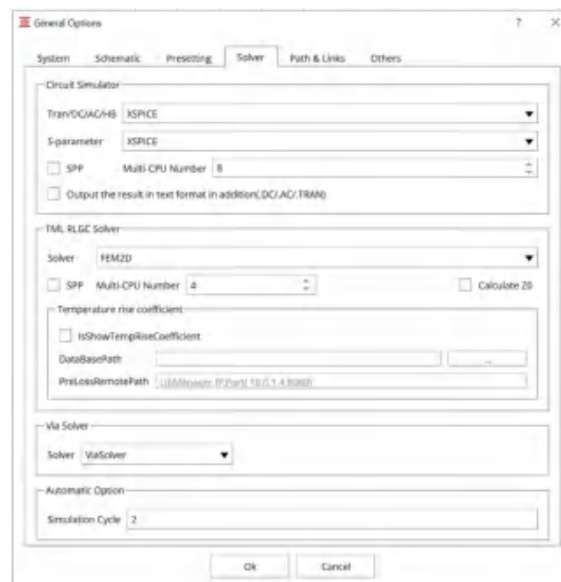
Sweep: 设置 S 参数扫描的选项, 默认 Type 为 Linear Step, 默认 start 频率为 100MHz, Stop 频率为 20GHz, Step 为 10MHz;

Interpolation: 为 S 参数计算的插值方法, 包括 Linear、Cubic 等 6 种方法; 插值类型包括 RI、MA、DBA 三种; 外推方法包括 Same as Interpolation 和 Constant Extrapolation 两种;

Voltage Probe (tran/AC/DC): 为使用 Probe 时, 是对 Netable 自己生效还是对 Probe 和 Netable 两种方式都生效;

S Parameter Unused Pin Termination Type: 为 S 参数加载后 Pin 脚的默认方式, 包括 Open、Match、Short 三种方式; TermG 为勾选 Match 状态时, 连接的电阻阻值大小。

3.2.4 Solver 窗口



Circuit Simulation:

Tran/DC/AC/HB: 进行 Tran/DC/AC/HB 仿真时使用的求解器, 可选择 XSPICE、HSPICE 等;

S-Parameter: 进行 S 参数仿真时使用的求解器, 可选择 XSPICE、HSPICE 等;

SPP: 勾选该选项后, 可以启用 XSPICE 多核运行, 核数在 Multi-CPU Number 处设置, 如果不勾选, CPU 核数默认为 1;

Output the result in text format in addition(.DC/.AC/.TRAN): 勾选该选项后, 结果在原基础上, 还会以文本格式生成新的结果;

TML RLGC Solver:

Solver: 在进行传输线类型仿真时, 选择 Solver, 有两种选择, X2D 或 BEM2D;

SPP: 勾选该选项后, 可以启用 X2D 多核运行, 核数在 Multi-CPU Number 处设置, 如果不勾选, CPU 核数默认为 1;

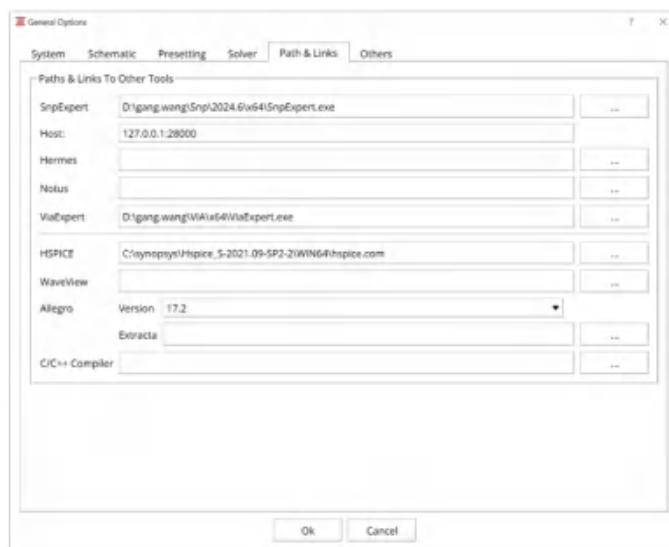
Calculate Z0: 是否自动计算 TML 的阻抗值;

Temperature rise coefficient:

IsShowTemRiseCoefficient: 勾选后选择加入温升系数, DataBasePath 为用户自己导入温升损耗系数的文件;

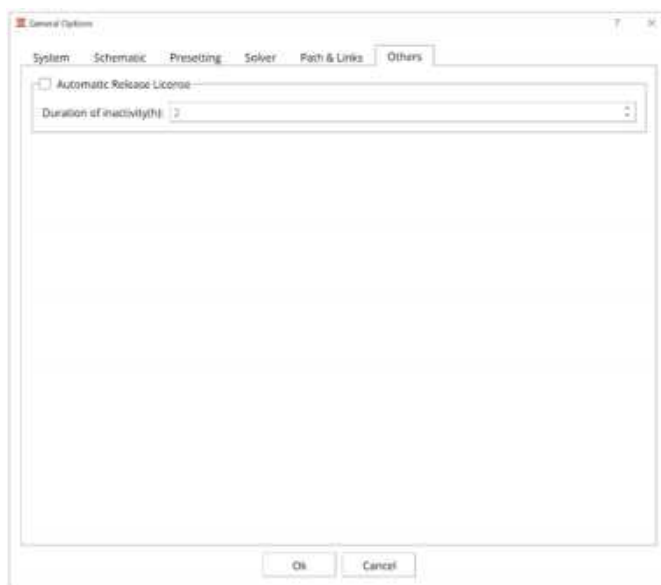
Simulation Cycle: 设置拓扑提取仿真长度, 默认为 2。

3.2.5 Path&Links 窗口



连接各个软件的运行程序，可在 Channel 中直接启动相关软件，如 SnpExpert、Host、HSPICE、WaveView 等。

3.2.6 Others 窗口



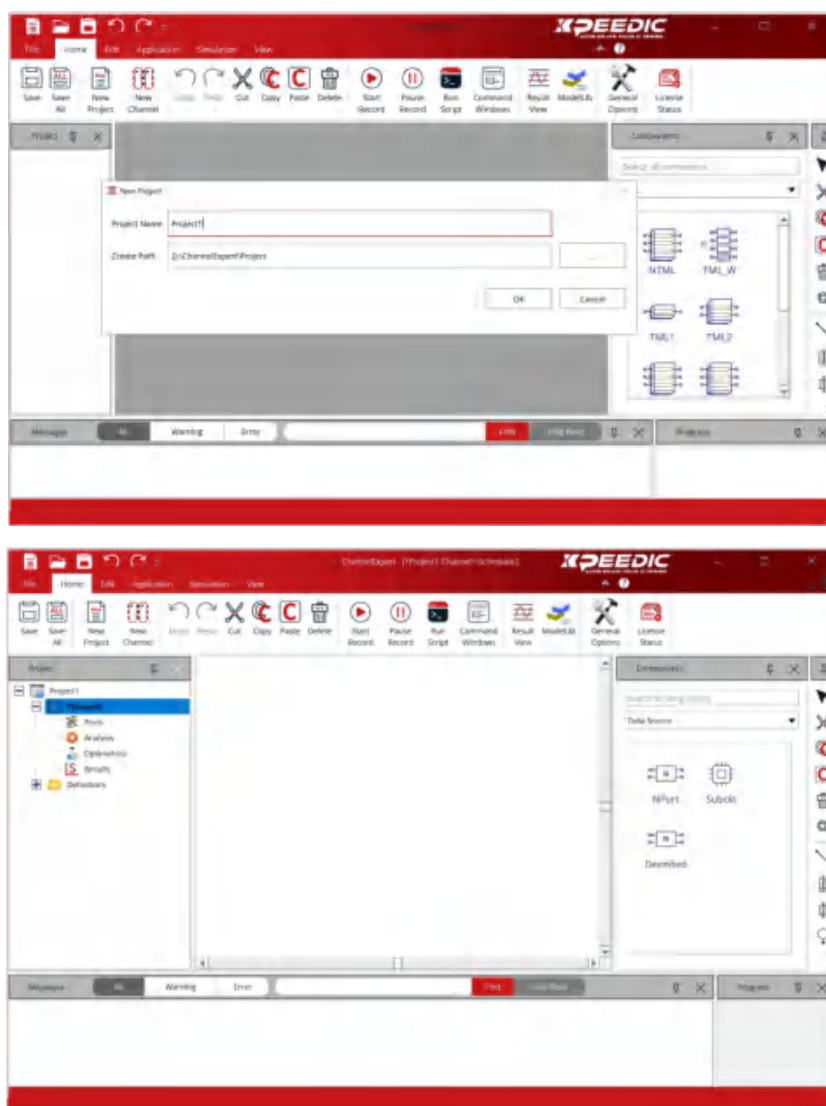
Automatic Release License:勾选后触发自动释放 license 功能；

Duration of inactivity: 时间设置，默认为 2 个小时。

3.3 基础操作

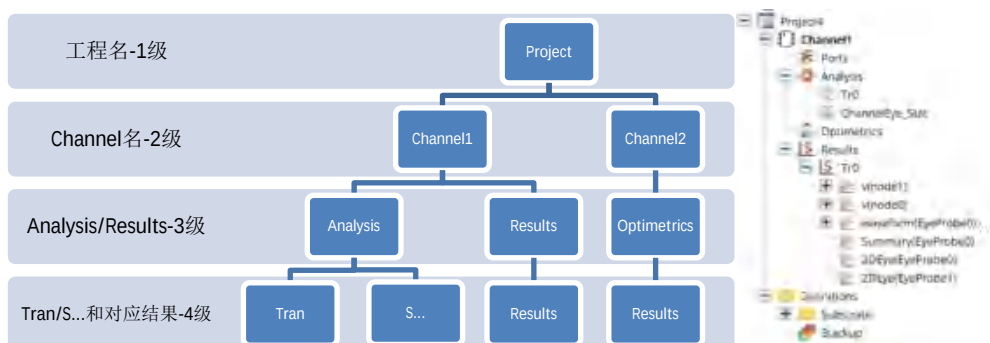
3.3.1 新建工程

点击左上角 New Project 或 New Channel 新建工程或新建原理图界面：

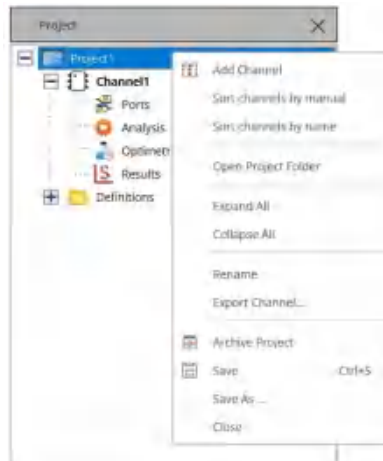


工程下有四级结构，同一个工程下可以有多个 Channel，同一个 channel 下根据原理图，可以进行多个类型的仿真，也可以将多个结果挂在在工程树下。

工程树通用设置：



3.3.1.1 右键 project 名称



Add channel: 增加新的 channel 工程

Sort channels by manual: 用户自己选择 channel 在 Project 下的位置;

Sort channels by name: 对 channel 按照首字母名字进行排序

Open Project Folder: 打开该 project 工程所在的文件夹

Expand All: 展开该工程下所有的 channel

Collapse All: 折叠该工程下所有的 channel

Rename: 重命名该工程名

Export Channel: 将 channel 的结果导出为报告, 在报告中展示原理图及结果等信息;

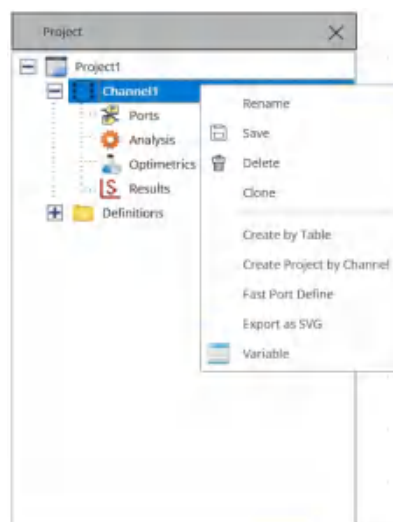
Archive Project: 打包整个工程, 将 Project 压缩为 Project.channelz 文件

Save: 保存工程

Save as: 另存为工程

Close: 关闭工程。

3.3.1.2 右键 Channel 名称



Rename: 重命名该 channel;

Save: 保存 Channel

Delete: 删除该 channel

Clone: 克隆该工程

Create by Table: 为工程内的引脚增加 Port

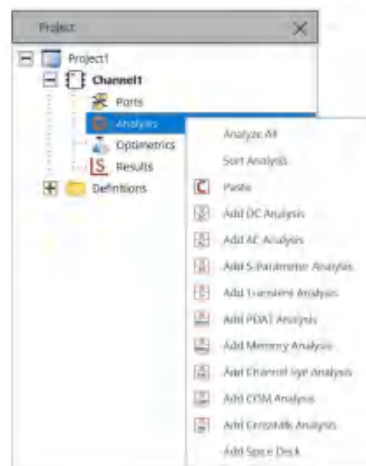
Create Project by Channel: 创建新的 Project 和 Channel;

Fast Port Define: 批量设置为引脚增加 port

Export as SVG: 导出为 SVG 格式的文件

Variable: 使用各扫描功能

3.3.1.3 右键 Analysis



Analysis All: 对树节点下所有的工程进行仿真

Sort Analysis: 有多个仿真时, 对仿真节点按名称进行首字母排序

Paste: 粘贴已经复制的仿真节点

Add DC Analysis: 增加直流仿真

Add AC Analysis: 增加交流仿真

Add S-Parameter Analysis: 增加 S 参数仿真

Add Transient Analysis: 增加瞬态仿真

Add PDAT Analysis: 增加 PDAT 仿真

Add Memory Analysis: 增加模拟存储器仿真

Add Channel Eye Analysis: 增加眼图仿真

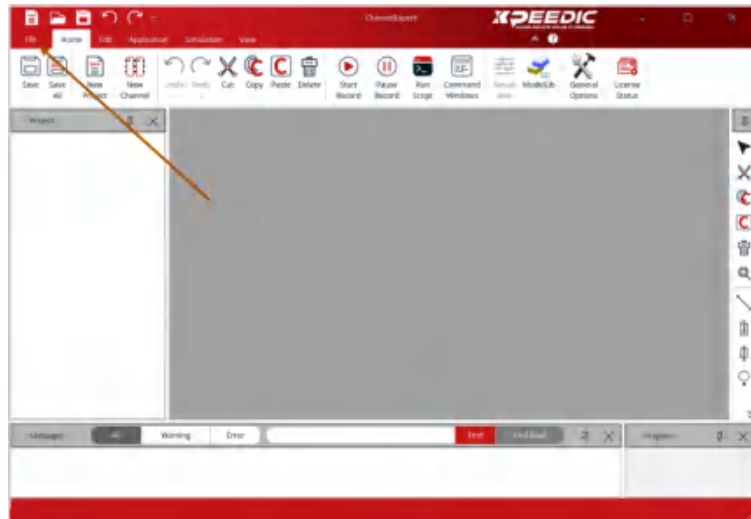
Add COM Analysis: 增加 COM 仿真

Add Crosstalk Analysis: 增加 Crosstalk 仿真

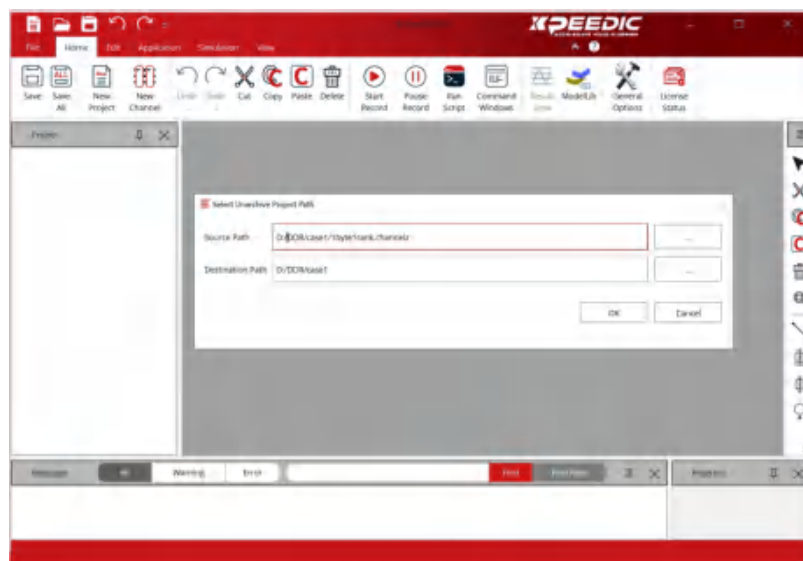
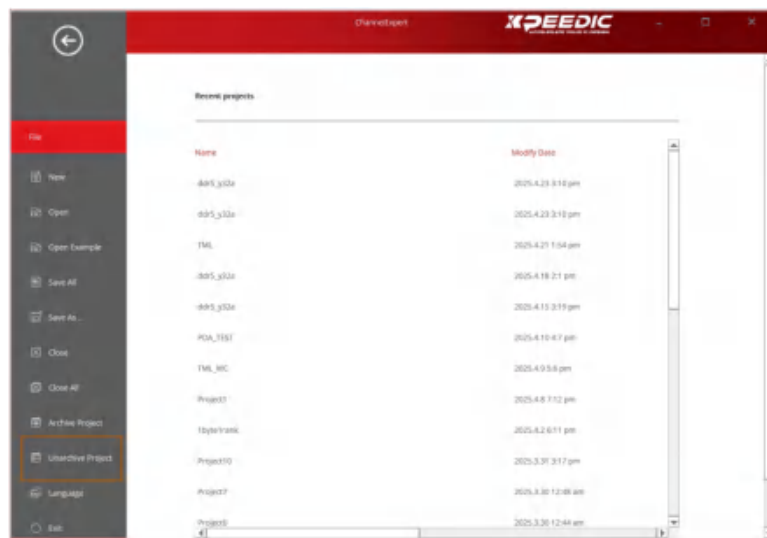
Add SpiceDeck: 增加网表仿真

3.3.2 工程解档归档与 example

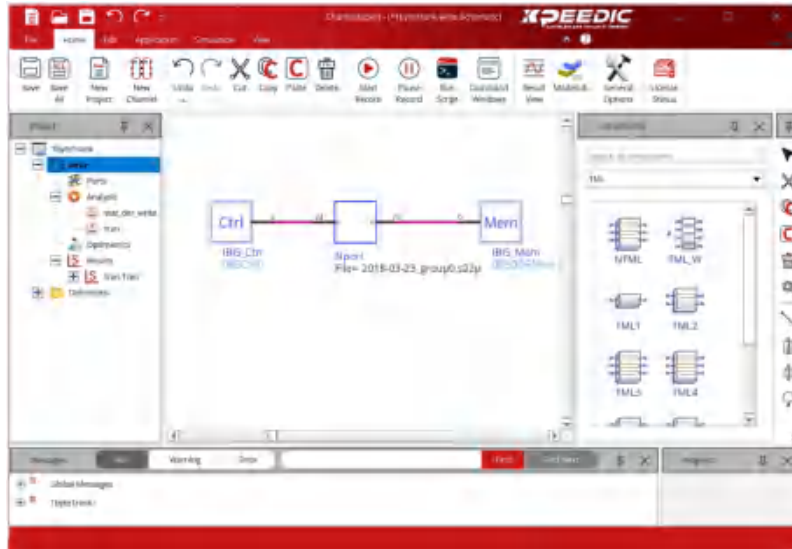
Archive Project: 压缩工程, 将 Channel 工程压缩为 Channelz 工程; 点击左上角 File, 弹出新的界面:



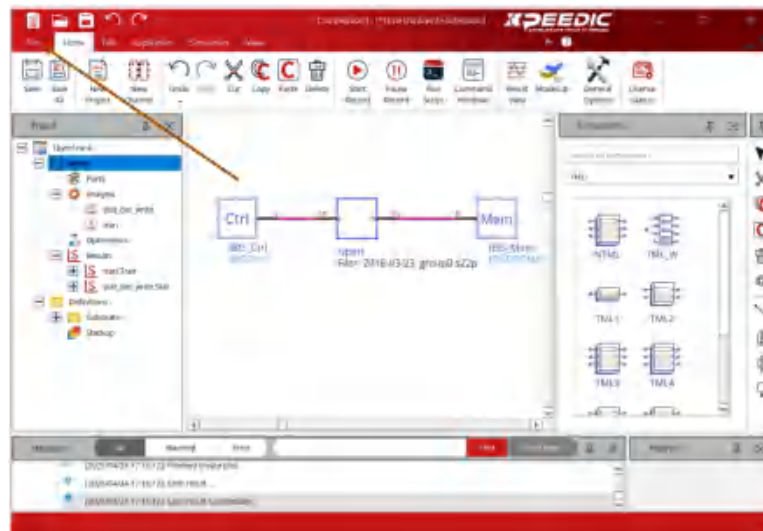
点击 Unarchive Project，弹出选择解压文件的路径，选择路径，点击 OK，



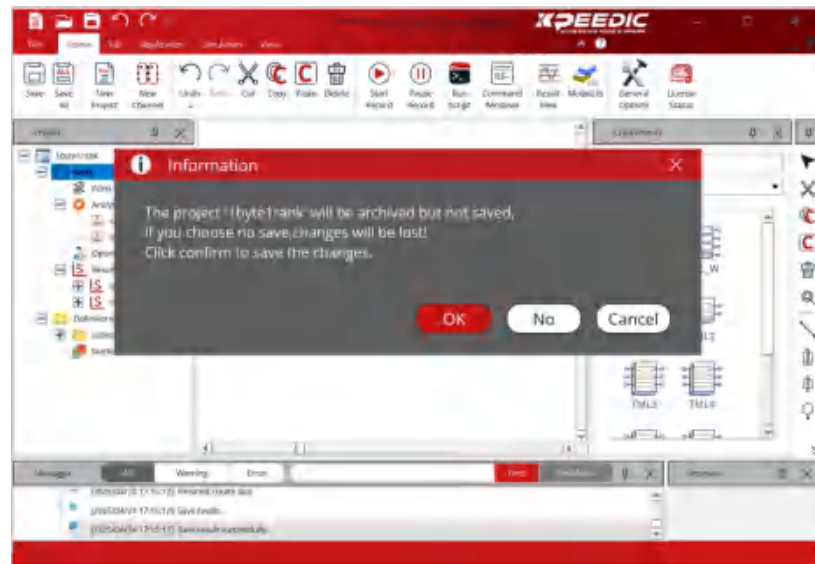
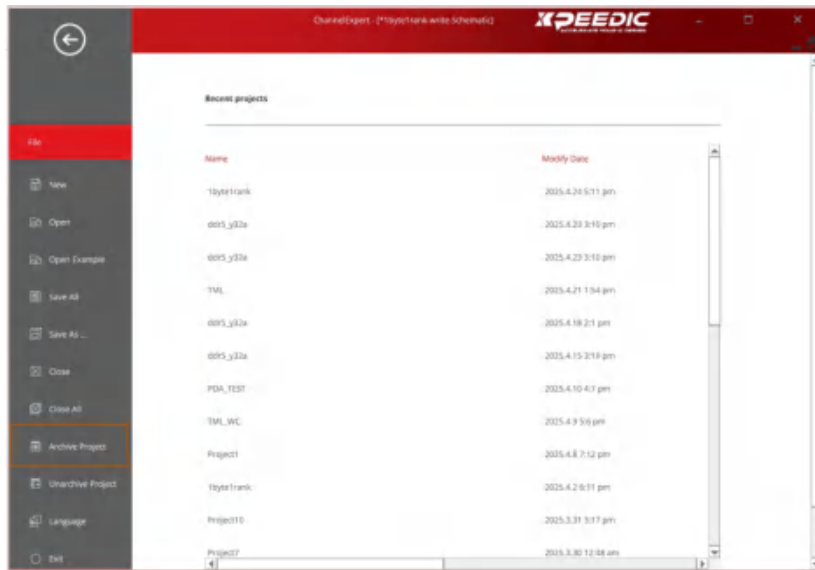
文件解压成功：



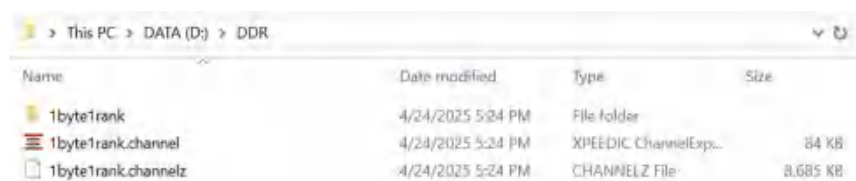
Unarchive Project: 解压工程，将 Channelz 工程解压为 Channel 工程。点击菜单蓝 File 选项：

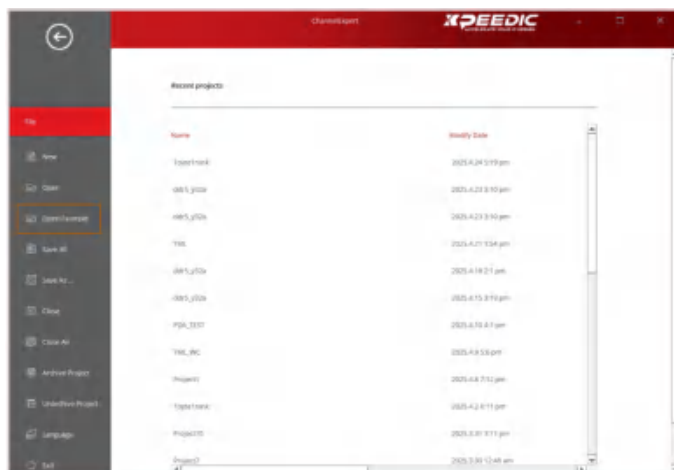


在弹出的界面中，选择 Archive Project，若工程被修改，会弹出是否保存提示框，点击 OK，弹出需要压缩的文件夹路径，点击 select 后开始压缩：

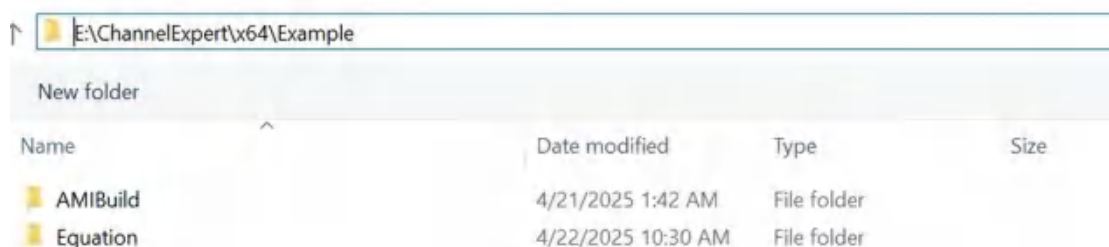


压缩成功后，在文件夹中产生.channelz 文件：





在左侧选项中点击 Open Examale，会打开软件内自带不同的仿真类型的工程，可点击 OK 选项打开查看：



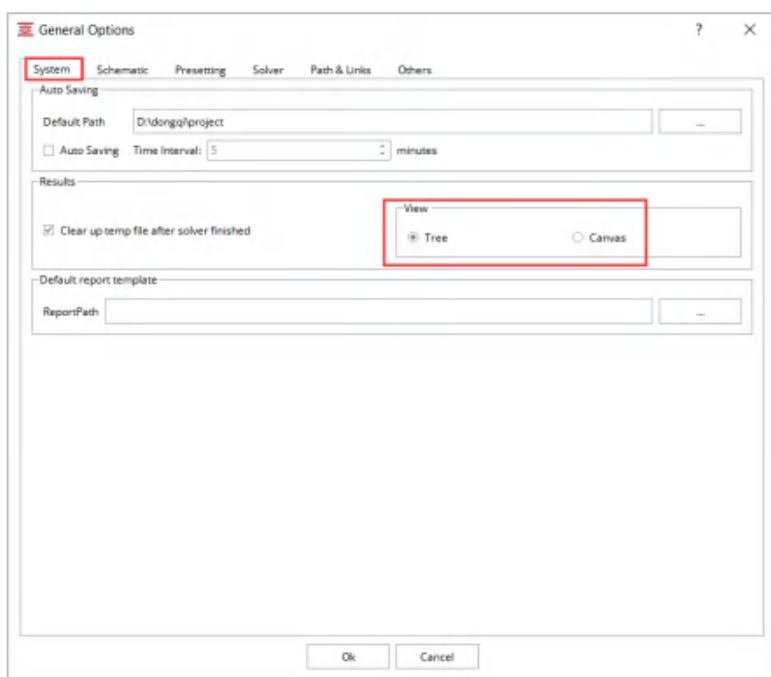
3.3.3 软件常用快捷键

类型	操作	快捷键
通用	New（新建立工程）	Ctrl+N
	Open(打开工程)	Ctrl+O
	Save（保存工程）	Ctrl+S
	Undo	Ctrl+Z
	Redo	Ctrl+Y
	End Command	Esc
	Copy	Ctrl+C
	Cut	Ctrl+X
	Paste	Ctrl+V
	Delete	Del
原理图	Zoom Fit	Ctrl+D
	Insert wire	Ctrl+W
	Rotate 90	Ctrl+R

	Flip Horizontal	Shift+H
	Flip Vertical	Shift+V
	Enter In Symbol	Shift+E
	Pop Out Symbol	B
	按住 Control 键拖动器件时，所选部分与未选部分断开连接关系	Ctrl
	改变 90°转折方式(交互式走线)	Space bar
	原理图视图上下移动	Ctrl+滚轮
	原理图视图左右移动	Alt+滚轮

3.3.4 结果显示与查看

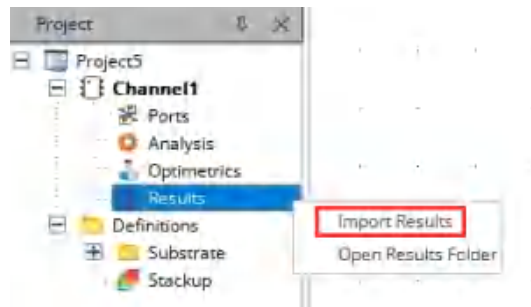
对结果后处理，产品支持两种模式。点击主 Home UI 界面上方的 General Options，进入 System 界面。View 下分 Tree 模式仿真完直接在左侧树节点挂结果曲线和 Canvas 模式仿真完不挂节点，通过 result view 窗口按需观测波形。



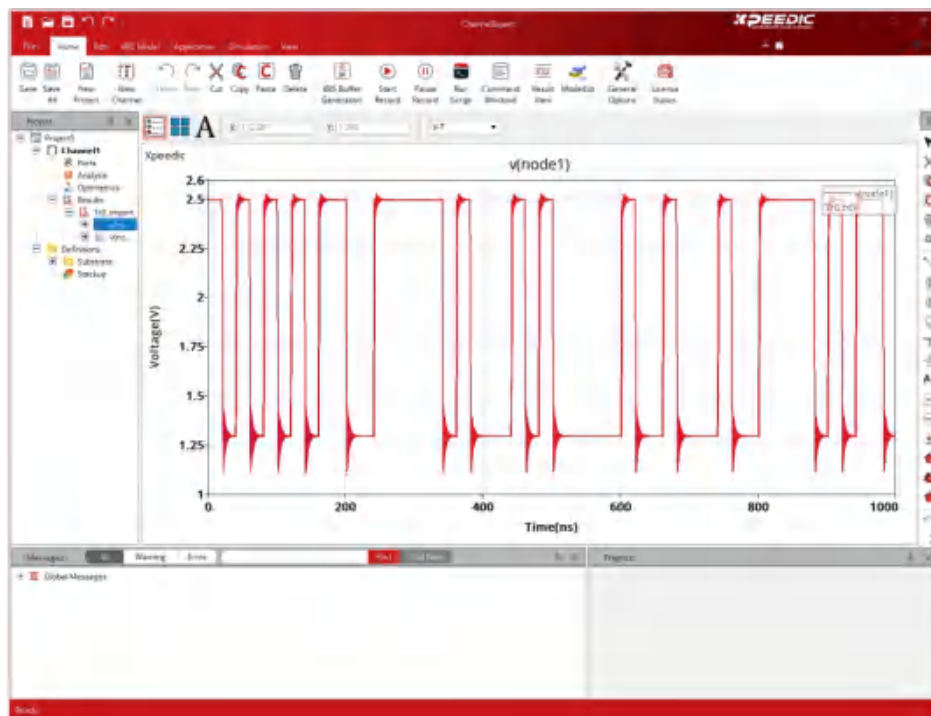
下面通过 Tree 模式介绍波形和眼图的后处理操作。

3.3.4.1 波形导入

右击工程树 Results，弹出 Import Results，支持导入.tr、.raw、.ac、.sw、.txt、.csv 等数据，可以绘制成曲线。

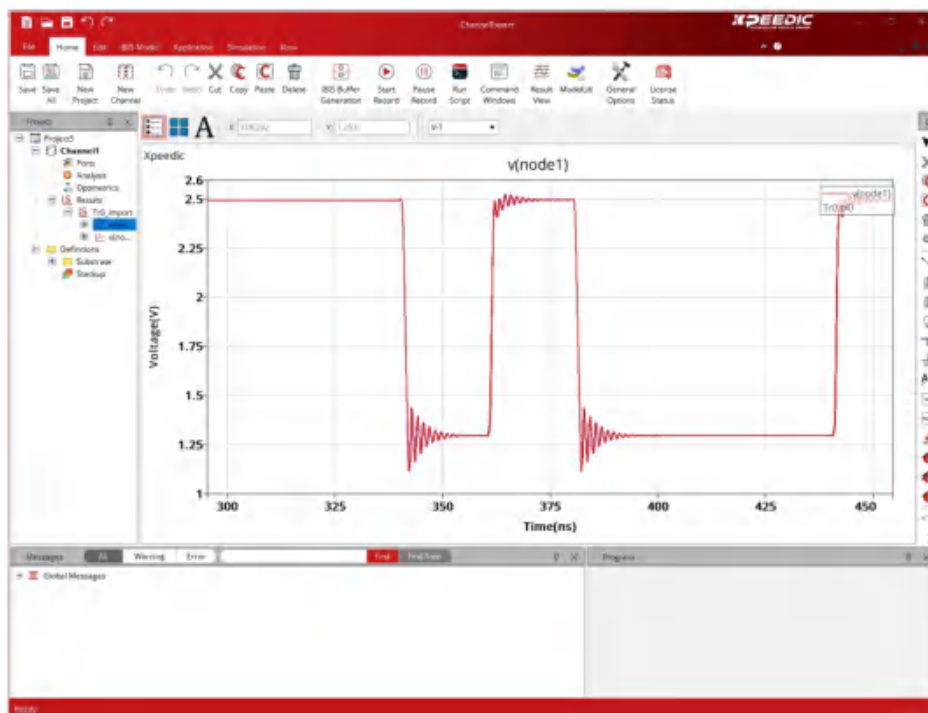
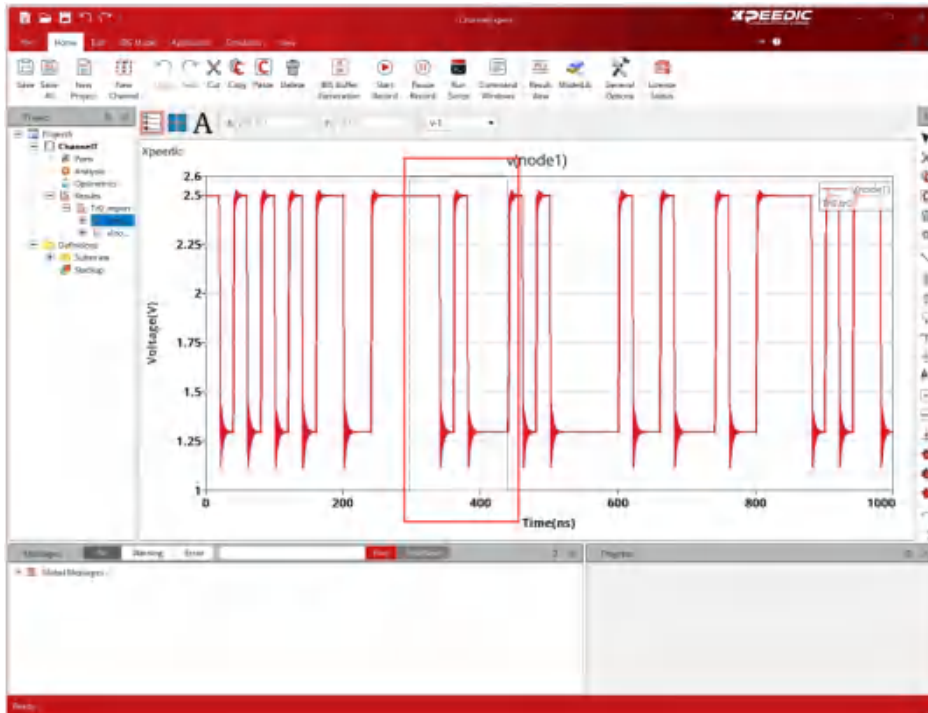


导入瞬态波形：

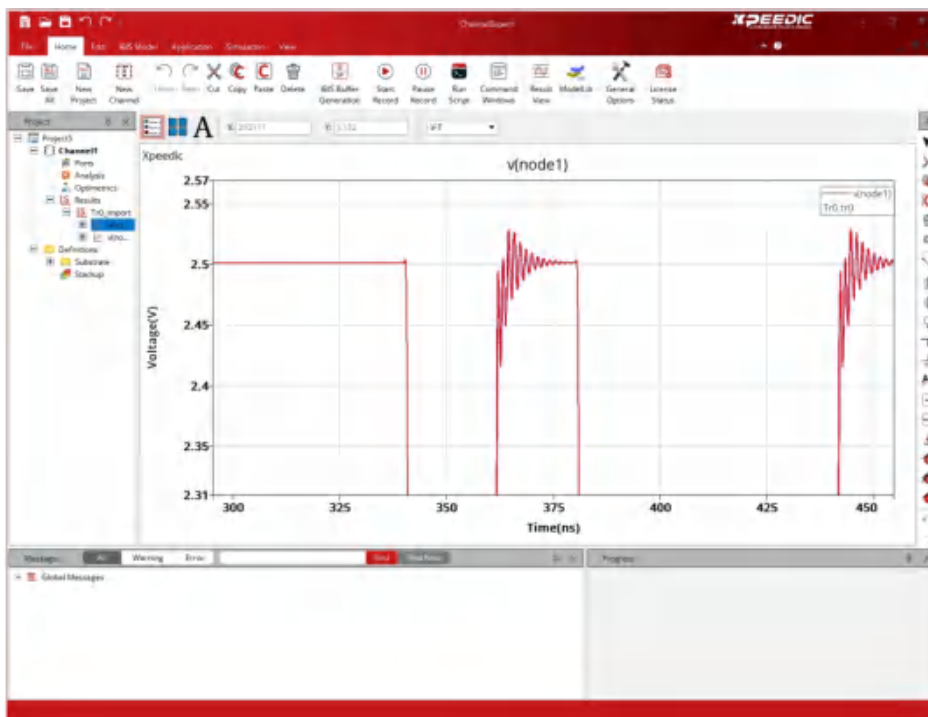
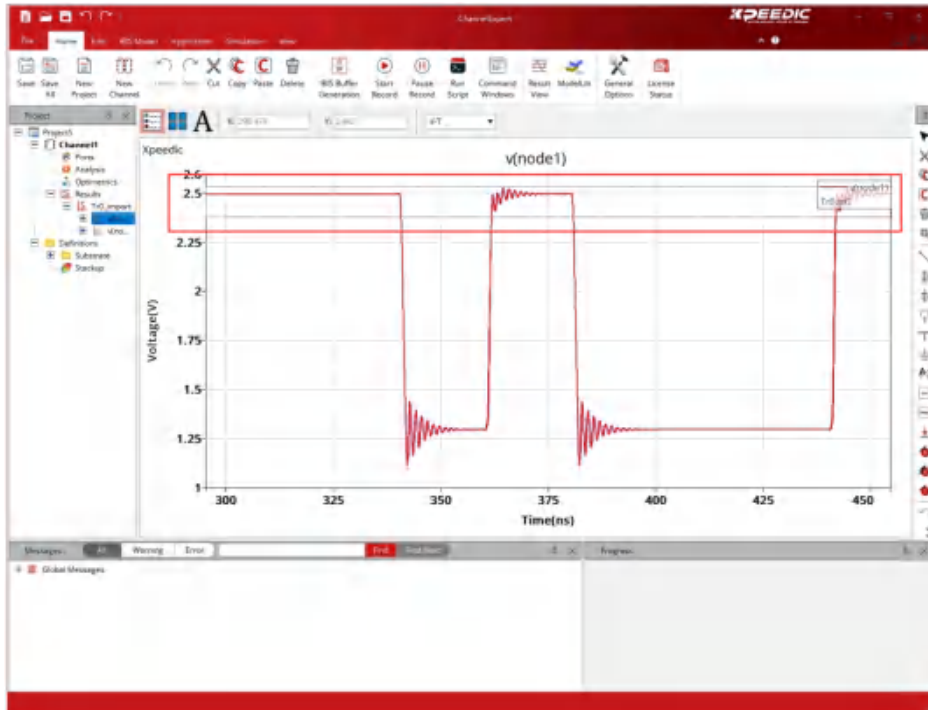


3.3.4.2 波形局部放大缩小

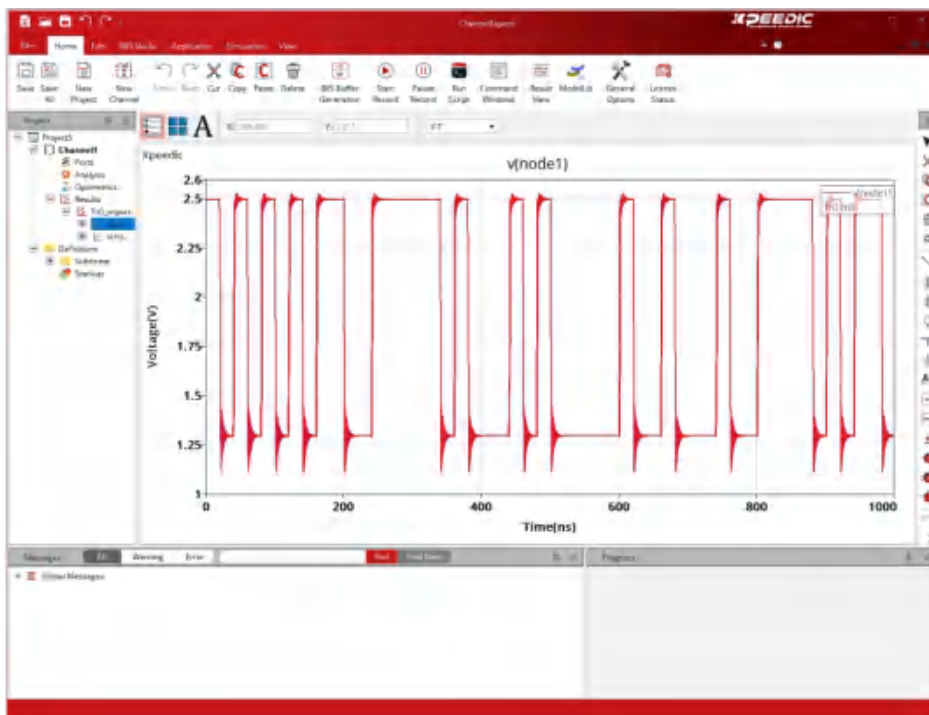
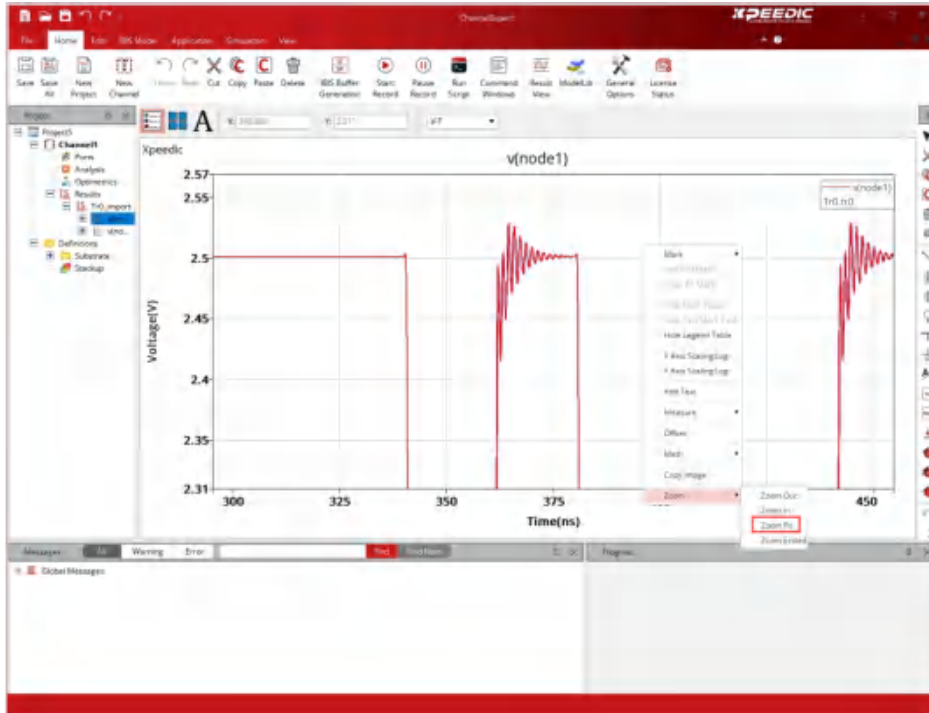
放大横坐标区间，鼠标放在横坐标下方空白，在开始位置按下鼠标，选中的区域有虚框，到需要停止的时间位置放开，按横坐标区域显示完成。



放大纵坐标区间，鼠标放在纵坐标左侧空白，在开始位置按下鼠标，选中的区域有虚框，到需要停止的电压位置放开，按纵坐标区域显示完成。

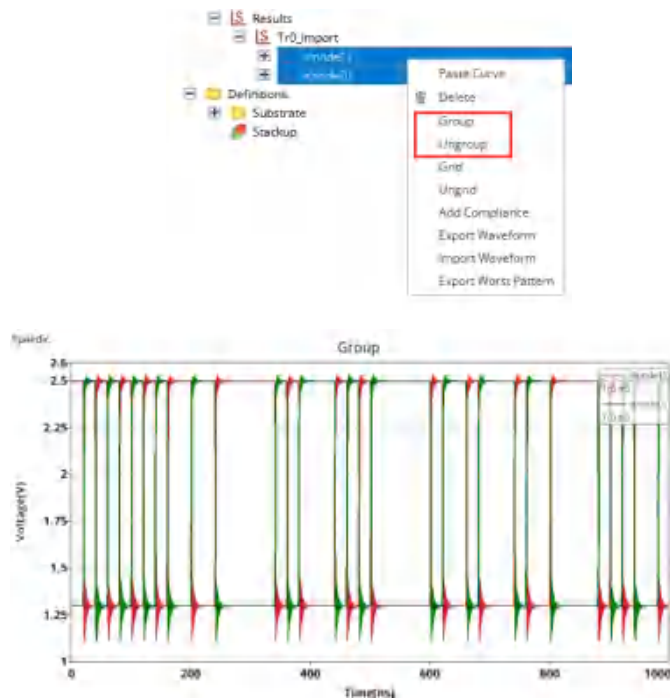


如果需要还原曲线，在画布上右击弹出后处理菜单栏，选择 Zoom 下的 Zoom Fit。



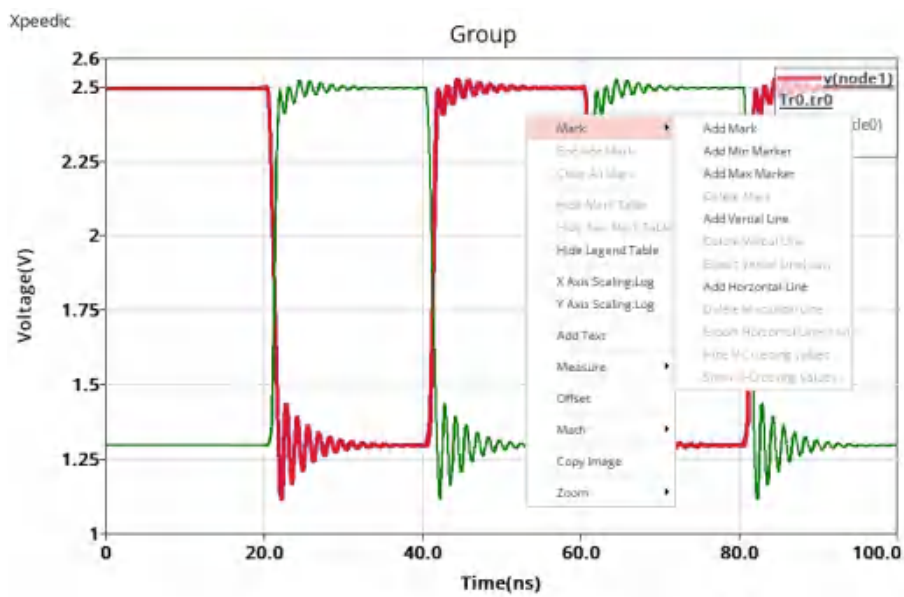
3.3.4.3 多个波形 Group

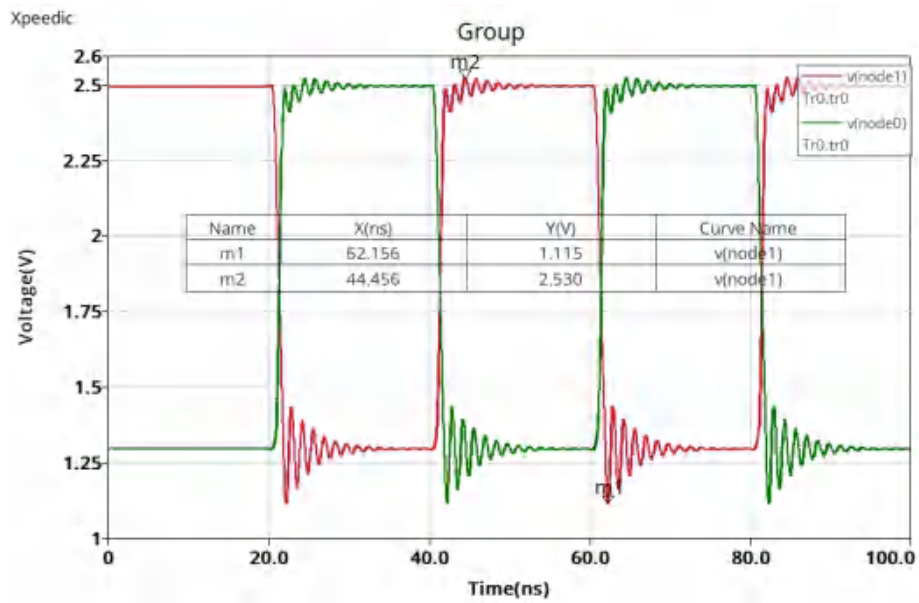
选中需要合并的曲线，右击弹出菜单栏，选择 Group，波形合并，Ungroup 波形取消合并。



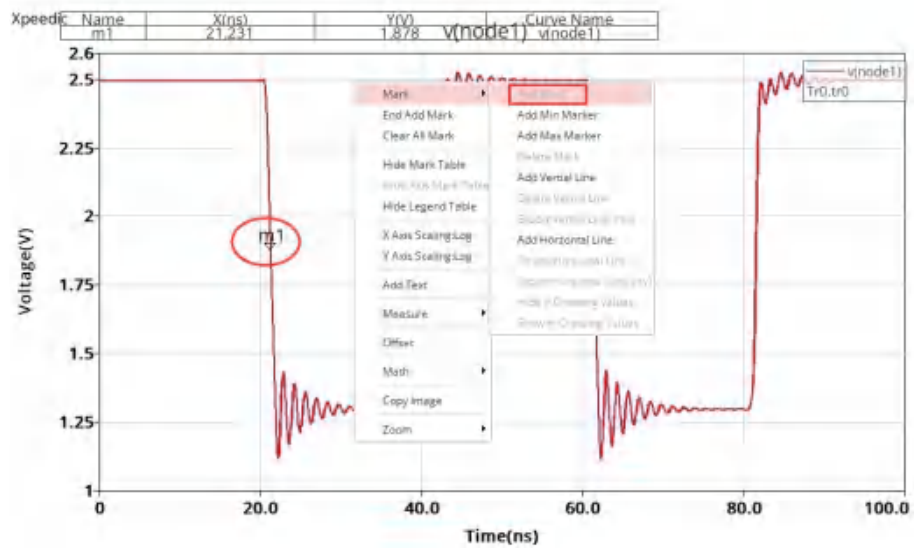
3.3.4.4 波形 Mark

鼠标选中单根曲线，曲线高亮，右击弹框选择 Mark。支持增加此曲线 Max 最大点标记，Min 最小点标记。Mark 点打在曲线上并以 table 表数据的方式显示具体坐标位置。



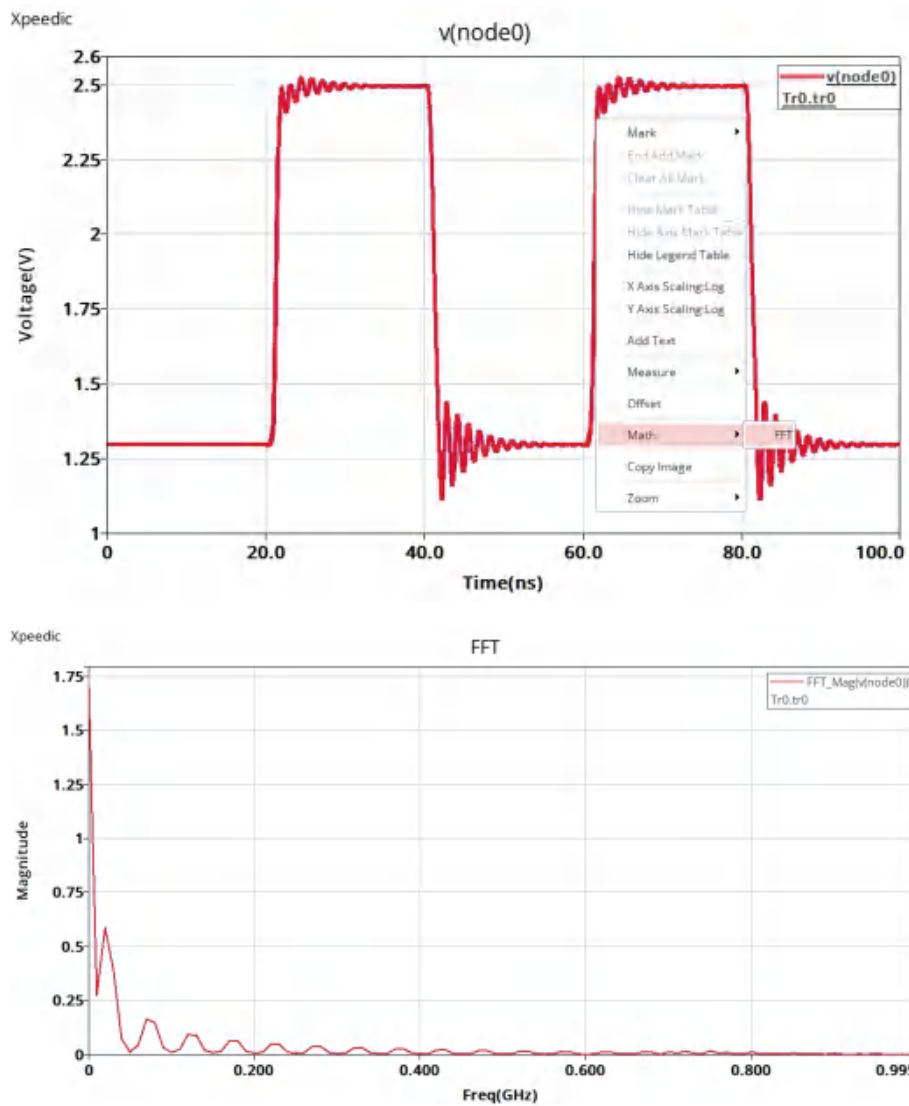


支持在曲线任意位置打 Mark 点，输出 table 数值。选中：



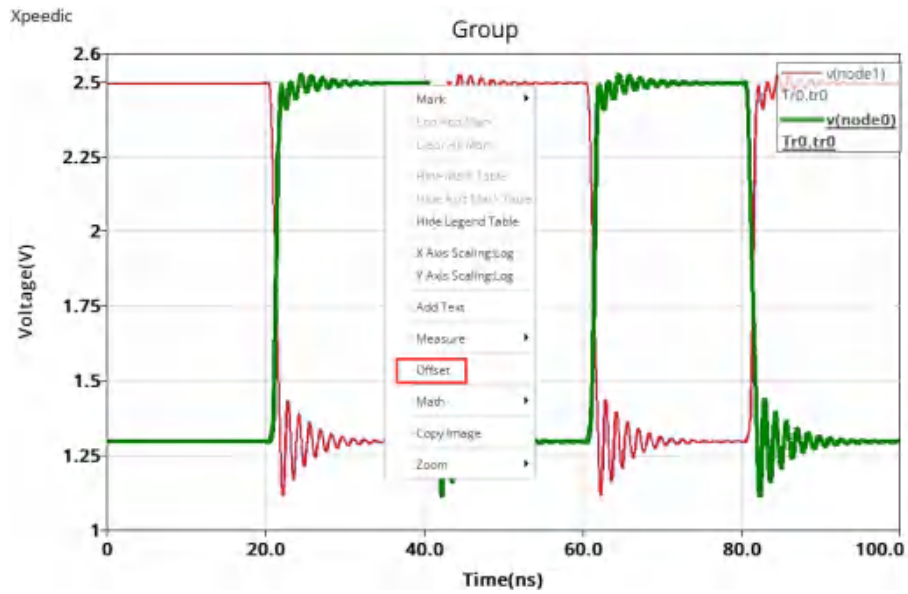
3.3.4.4 波形 FFT

选中曲线，右击弹出菜单栏，选择 Math FFT，进行傅里叶变换。

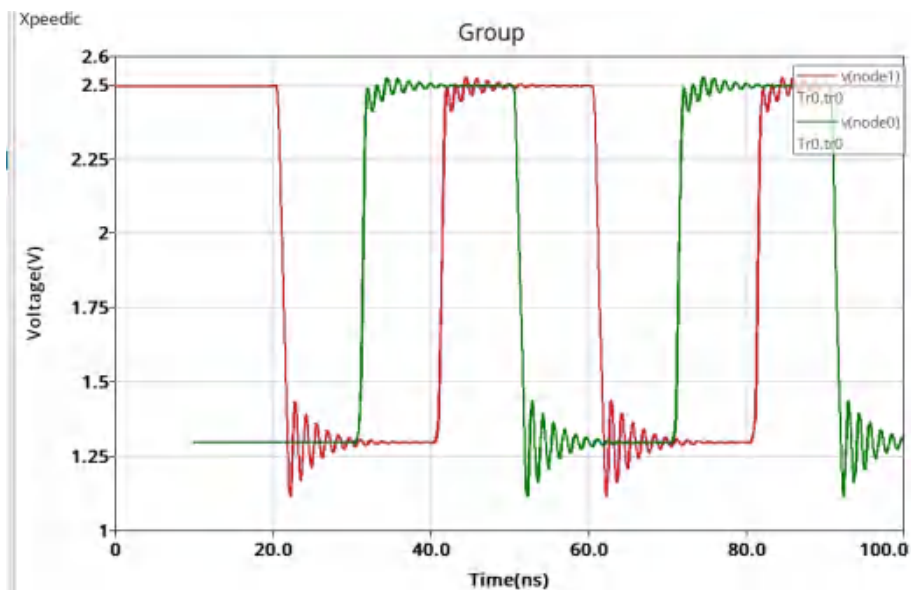
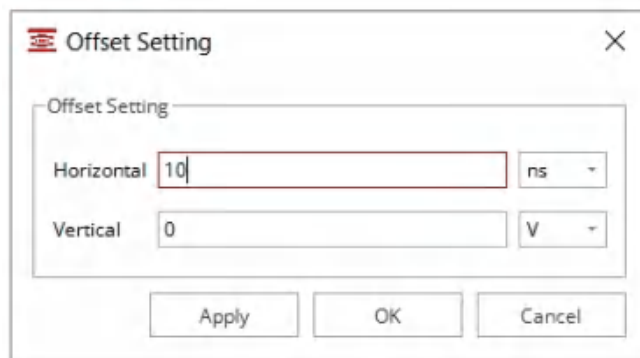


3.3.4.5 波形 Offset 偏移

选中其中一根曲线，右击弹出菜单栏，选择 Offset，进行此根线偏移操作。

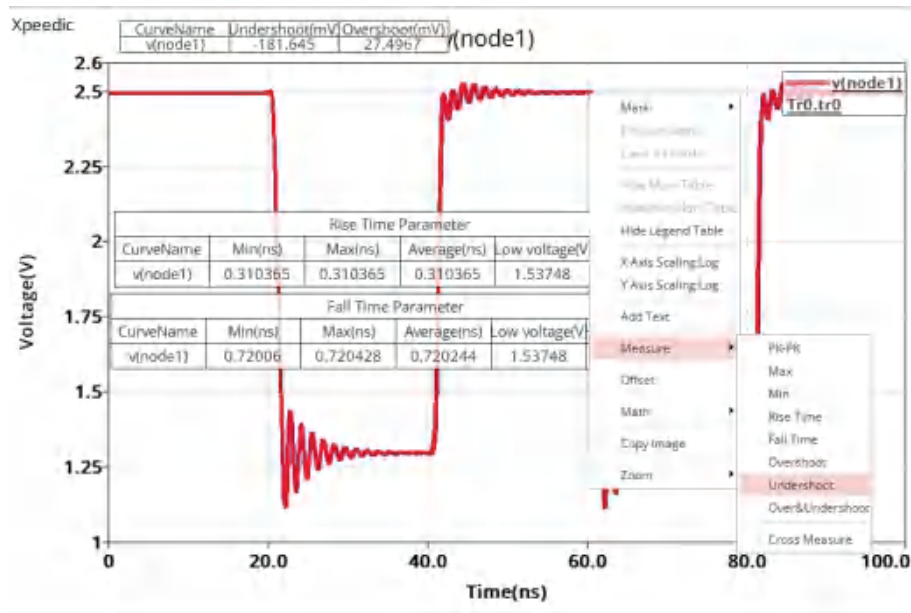


弹出 Offset Setting 界面，向右移 Horizontal 输入正值，向左移输入负值，时间单位可以切换。向上移 Vertical 输入正值，向下移输入负值，电压单位可以切换。



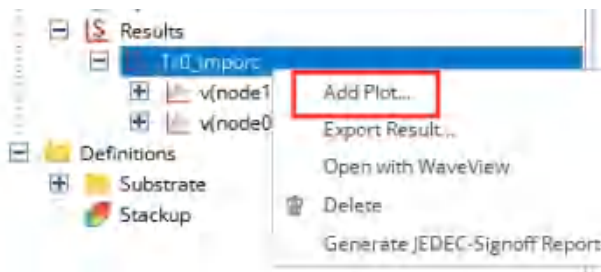
3.3.4.6 波形 Measure 测量

选中曲线，曲线高亮，右击弹窗菜单，选 Measure 进行曲线测量。支持 PK-PK、Max、Min、Rise Time、Fall Time、Overshoot、Undershoot。测量值会以 table 形式显示对应参数值。



3.3.4.7 曲线快捷公式计算

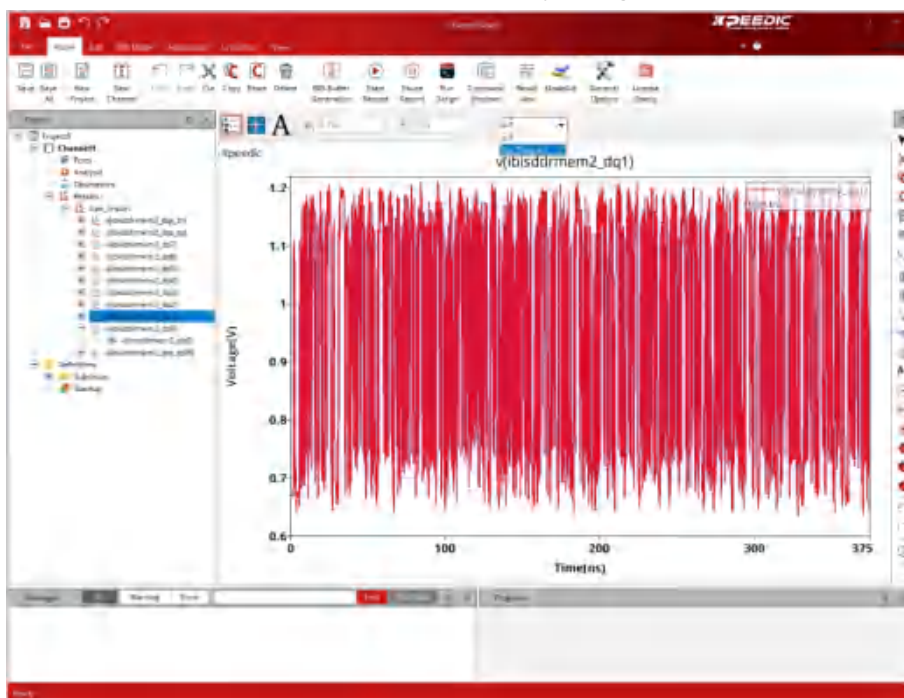
右击树节点 Results 下数据导入总节点，选择 Add Plot，弹出简易波形公式编辑弹框。



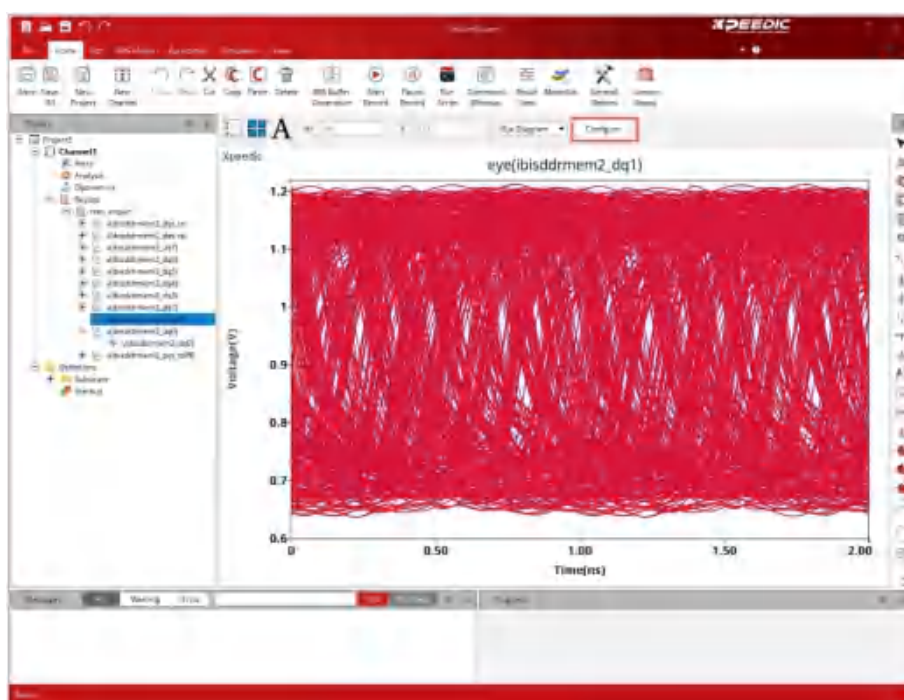
数据文件里的曲线会挂在下表，箭头拖入右侧公式编辑框，进行公式编辑，完成后点击 Add Plot，处理后的波形会挂在树节点下。

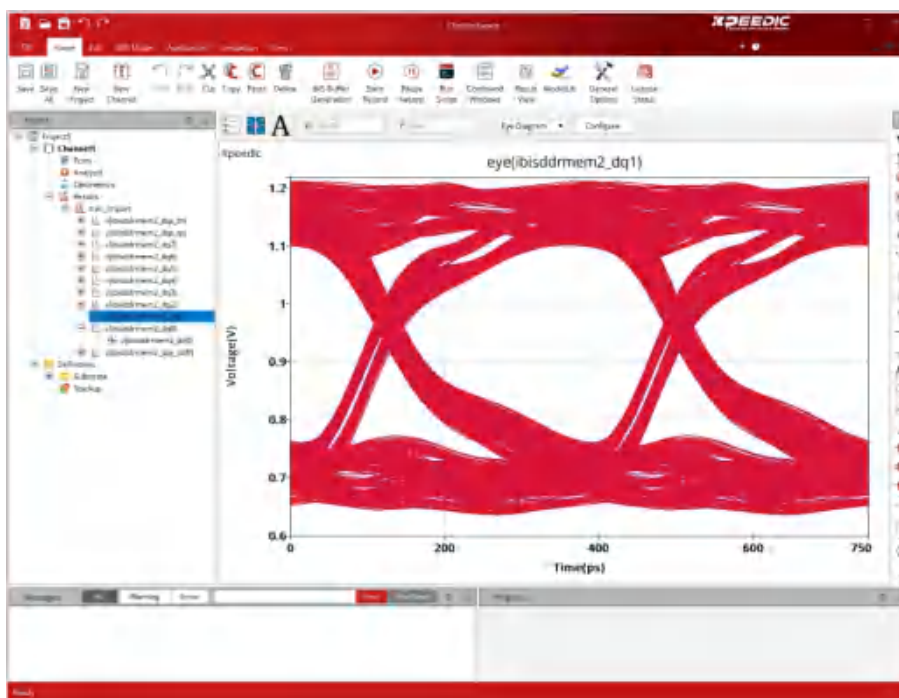
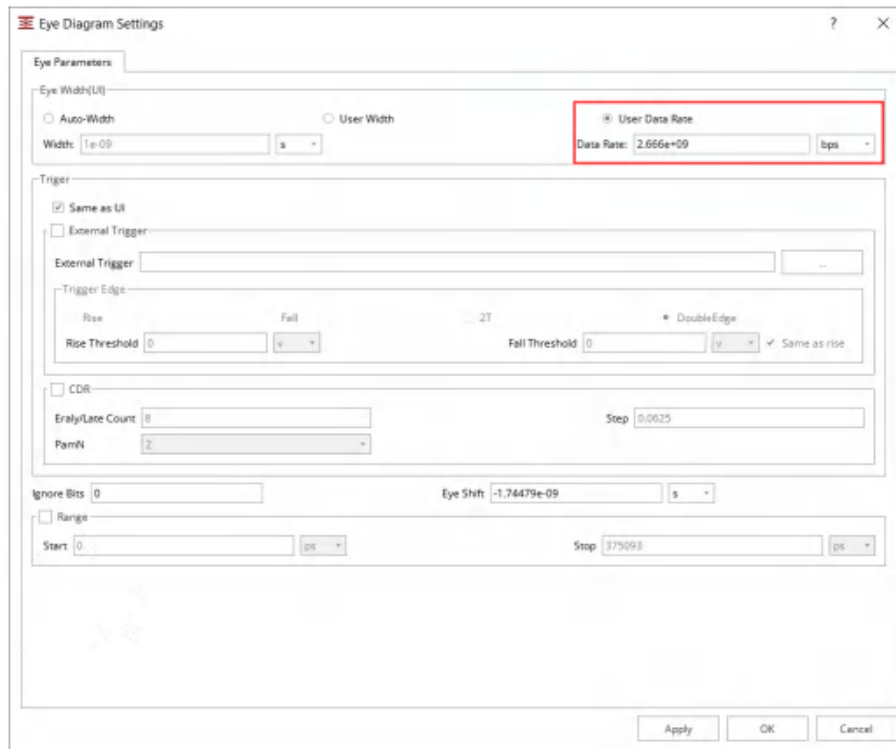
3.3.4.8 波形叠眼图

导入的时域波形，PLOT 上方，V-T 切换 Eye Diagram，曲线转换眼图。



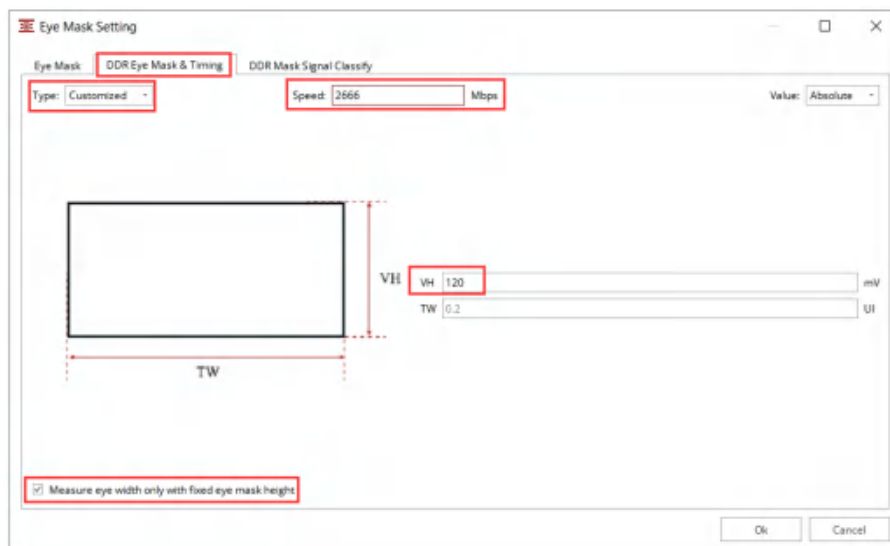
点击眼图上方快捷 Configure，弹出 Eye Diagram Setting 界面。Eye Parameters 勾选 User Data Rate 输入仿真速率。



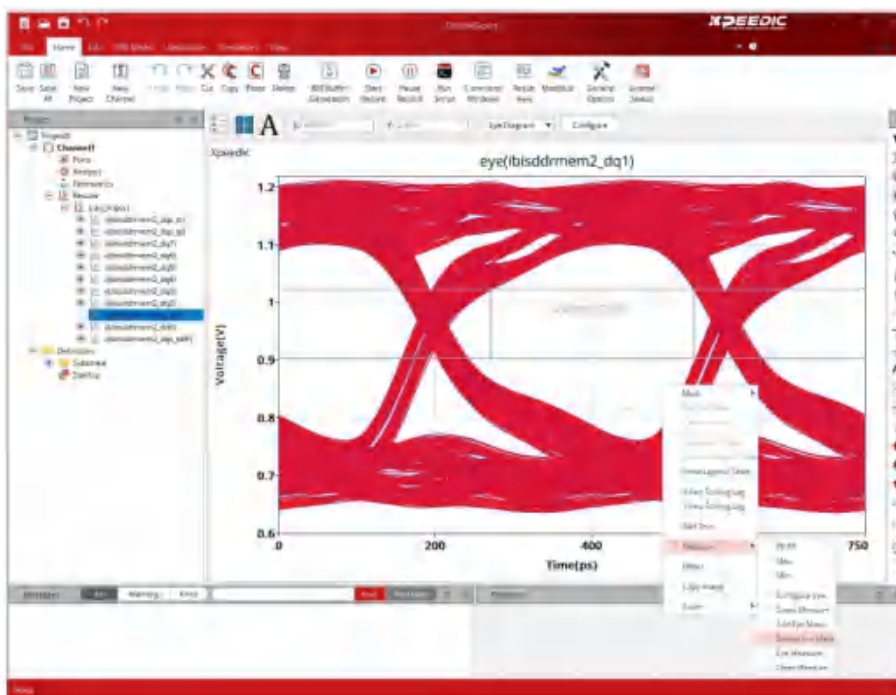


眼图测量的前提是，眼睛在中心，图幅里只能出现一个完整的眼睛。多个眼睛会影响测量。将眼图移到中心，继续点击眼图上方快捷 Configure，弹出 Eye Diagram Setting 界面。在 Eye Shift 编辑框，输入需要偏移的量，将眼睛移到中心。



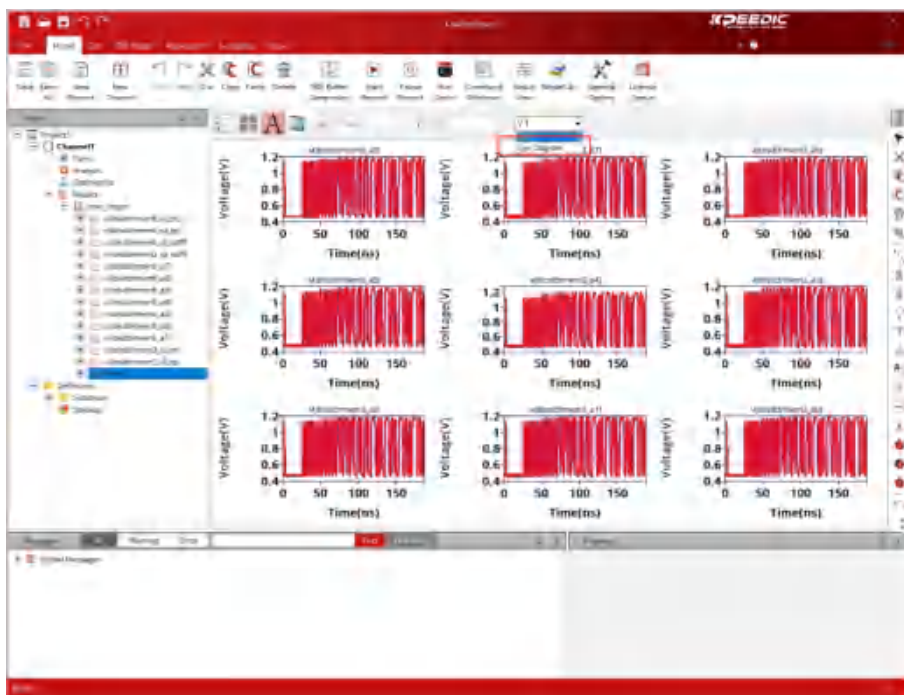
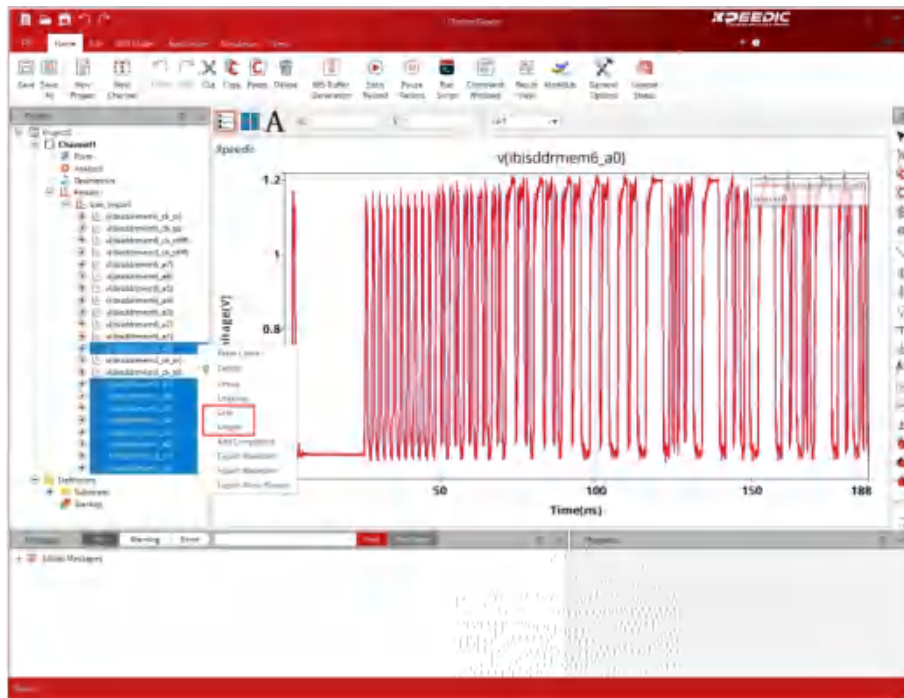


眼图中间的 Mask 显示固定眼高下的眼宽数据。单位为 ps。如果需要去掉 Mask，选择 Delete Eye Mask。

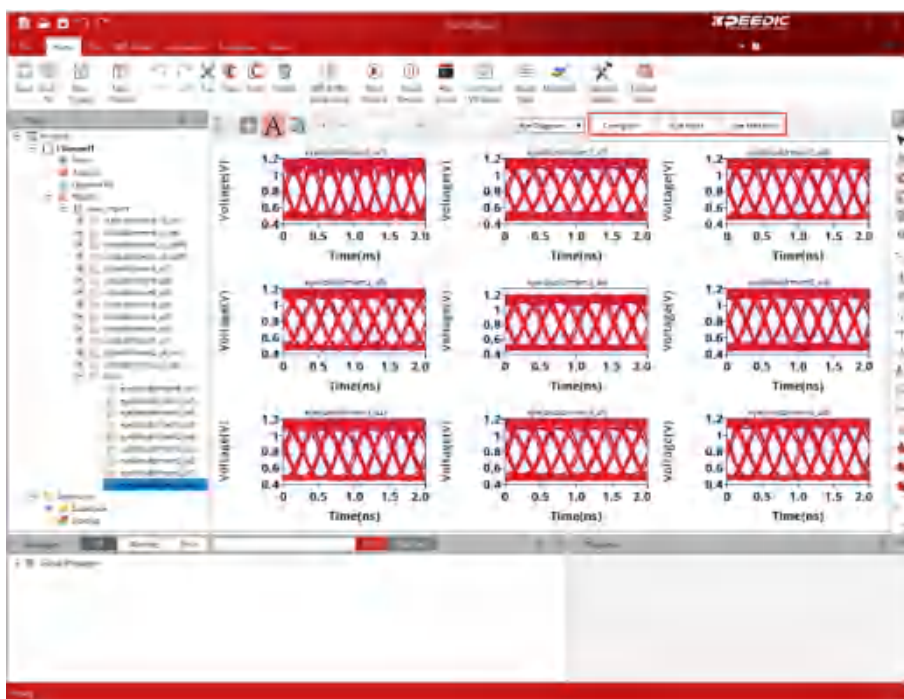


➤ 3.3.4.10.2 批量测量增加 Mask

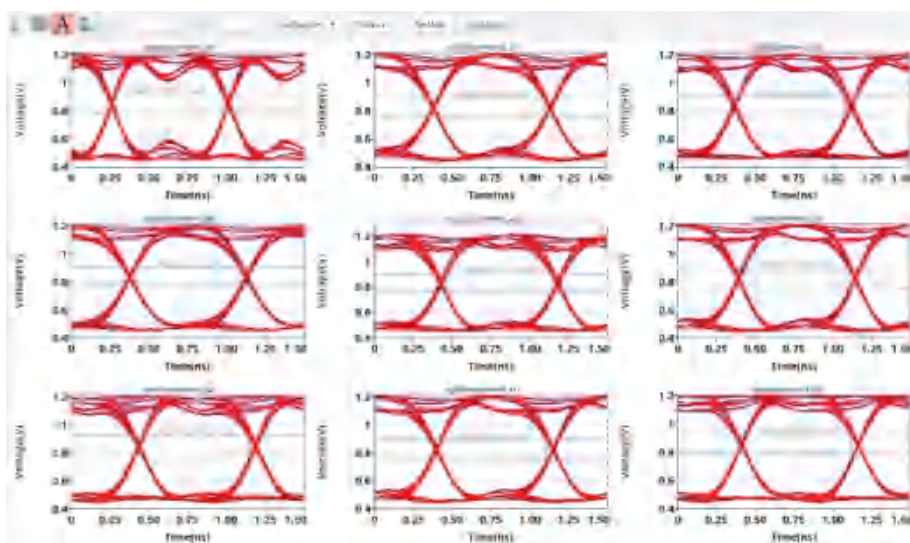
对需要批量观测的波形多选，右击选择 Grid。波形 Grid 显示，批量转眼图。



Grid 模式下 PLOT 上方的设置按钮对集体生效。



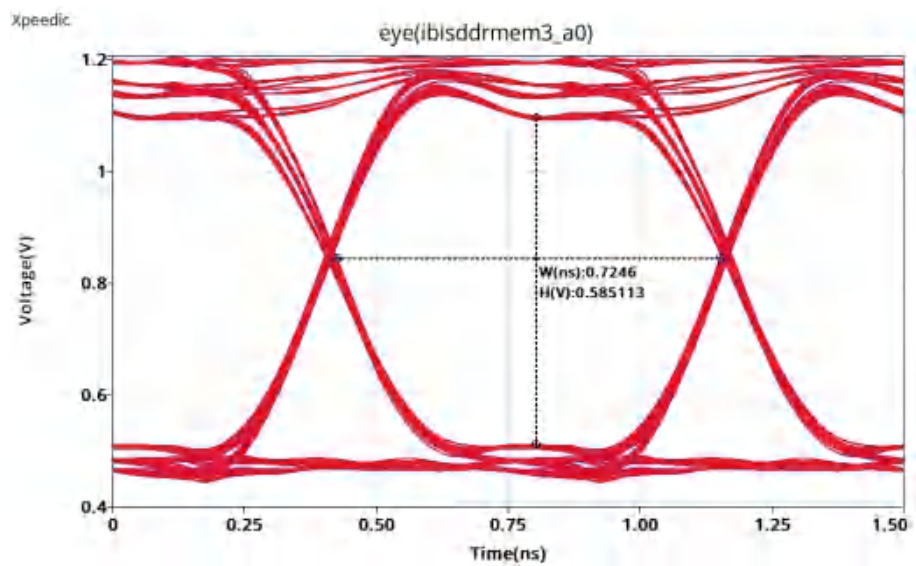
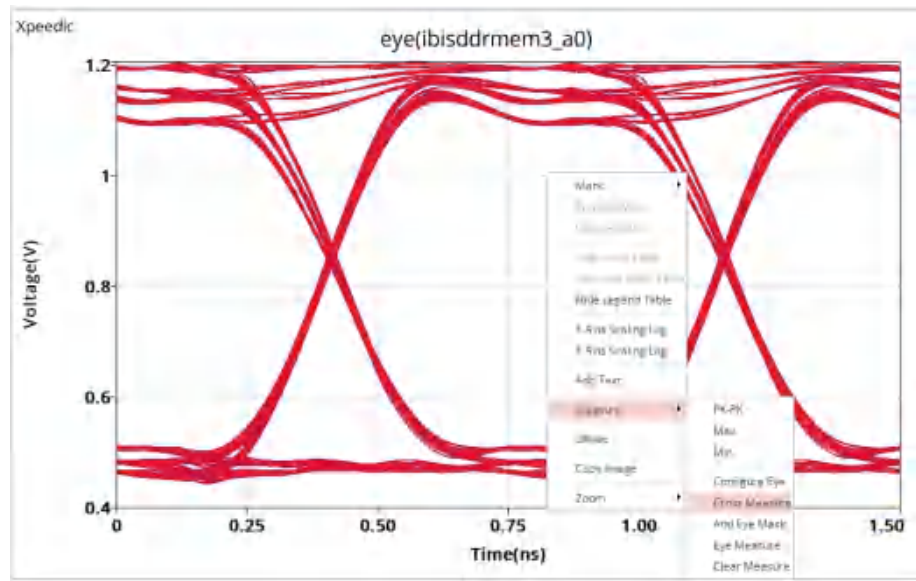
按照上面章节调整 Configure 里的 User Data Rate 和 Eye shift 调整好眼睛，支持批量给 Grid 里的眼图进行调节和增加 Mask。



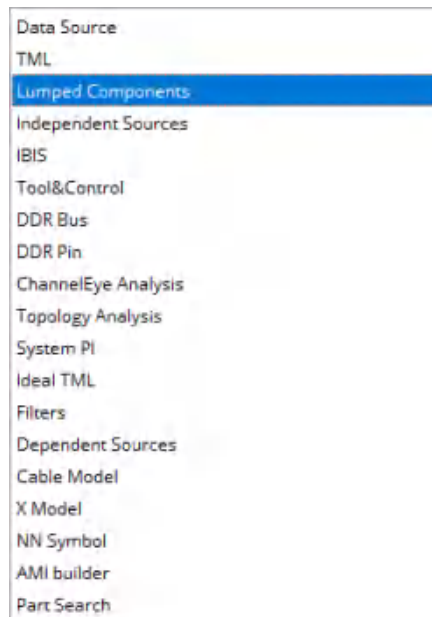
➤ 3.3.4.10.3 十字光标测量

同时支持光标追踪眼图，进行快速测量。

在眼图 PLOT 右击 Measure，选择 Cross Measure，鼠标放在眼图任意需要测量的位置出现十字光标，测量眼宽眼高。结束追踪选择 End Cross Measure。



4 器件与功能组件



4.1 仿真相关组件

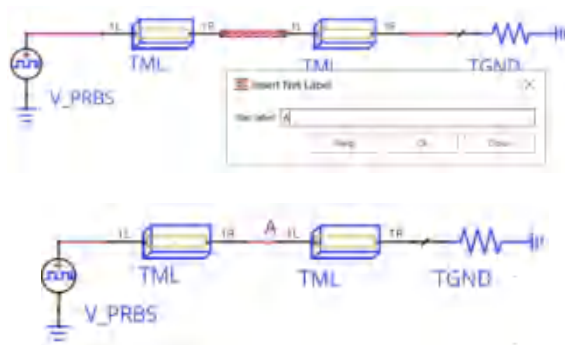
4.1.1 探针与网络标号 Netlabel



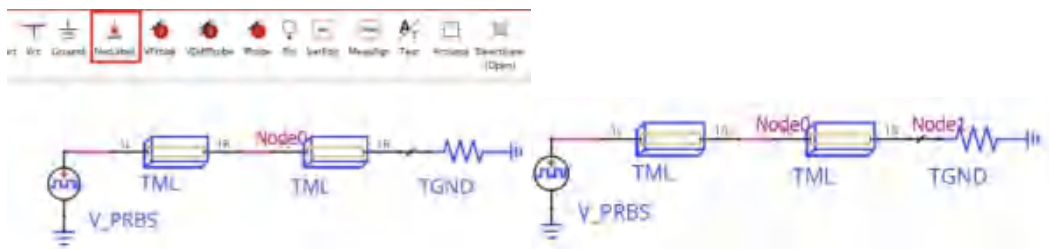
探针和 Netlabel 是位于 Edit 中。

探针包含电压探针、电流探针和差分电压探针。

Netlabel 添加方法有两种，一种是双击除 bus 线和接地线之外的走线，在编辑界面添加 probe 名称，从而新增一个 netlabel。如图所示：

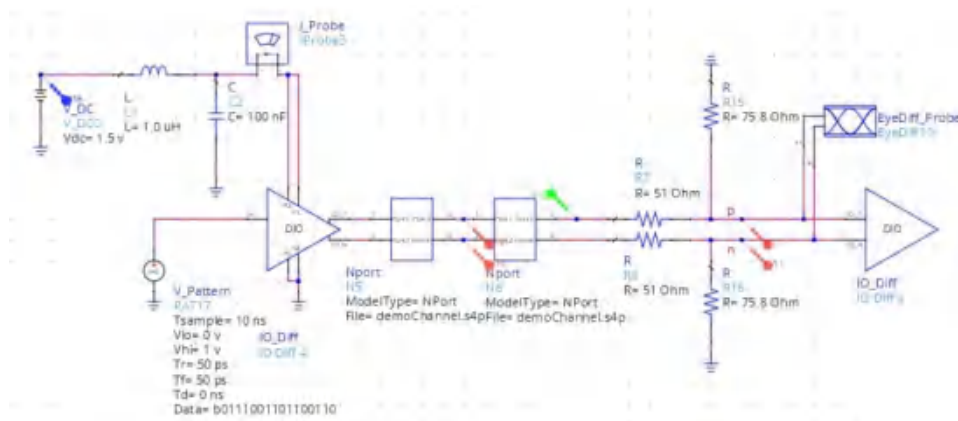


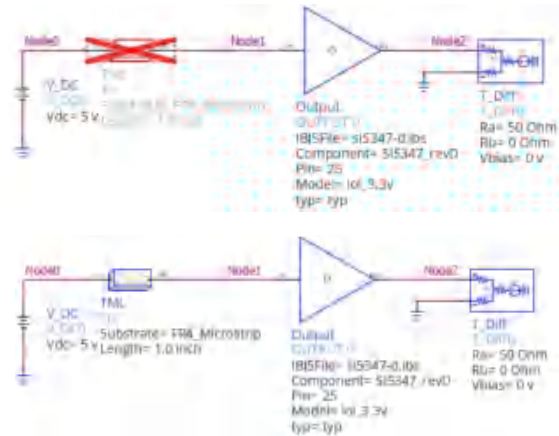
Netlabel 的另一种添加方法是，点击 Netlabel 选项后，再点击器件引脚或者走线（除 bus 线和接地线之外），会自动生成一个名字为 Node0 的 netlabel，继续点击其他器件或者走线可以递增添加 Netlabel: Node1 等，取消添加 Netlabel 需要右键选择 End Command ESC 或者按一下键盘的 Esc 即可。



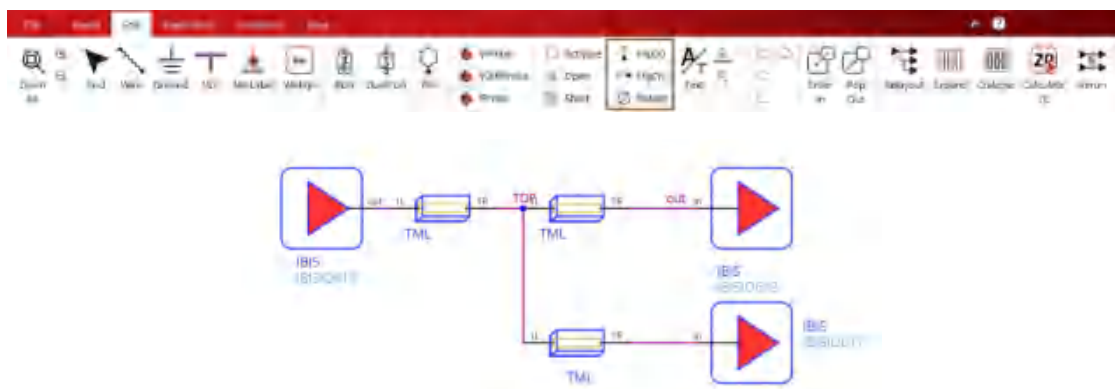
Nettable 修改名字的方法类似第一种添加 Nettable，也是通过双击线进行的。

探针的添加方法是通过点击相应探针后点击器件或者引脚添加的，不同点在于 IProbe 只能添加到器件的引脚上（非 bus 引脚），VProbes 是同时可以添加在引脚（非 bus 引脚）和线上的，VDiffProbe 添加和电压探针相同，但必须要一次添加一组探针才可以成功。具体使用如下图，可以探针和 Nettable 组合使用。

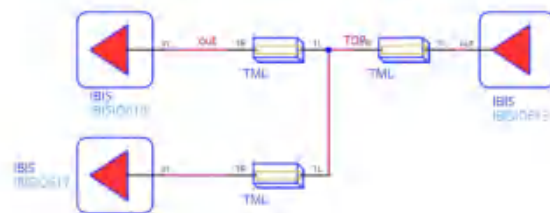




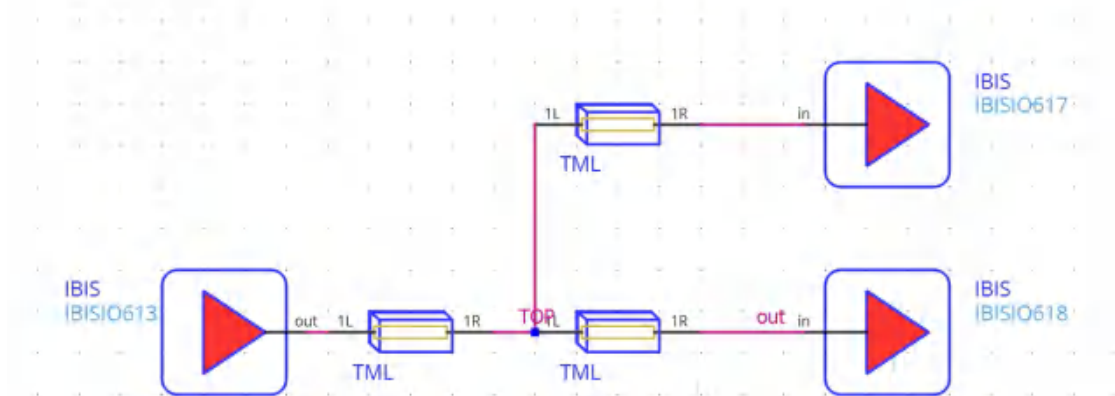
4.1.3 翻转



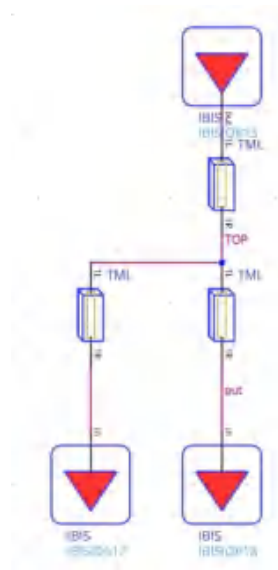
Filp(X) : 支持将单个或多个器件水平翻转



Filp(Y) : 支持将单个或多个器件垂直翻转



Rotate: 支持将单个或多个器件顺时针翻转 90 度。

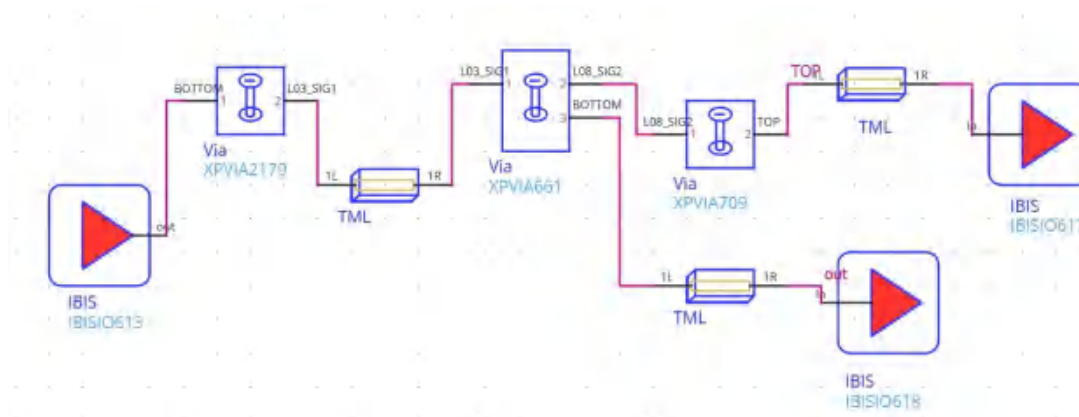


4.1.4 Xtop 相关组件

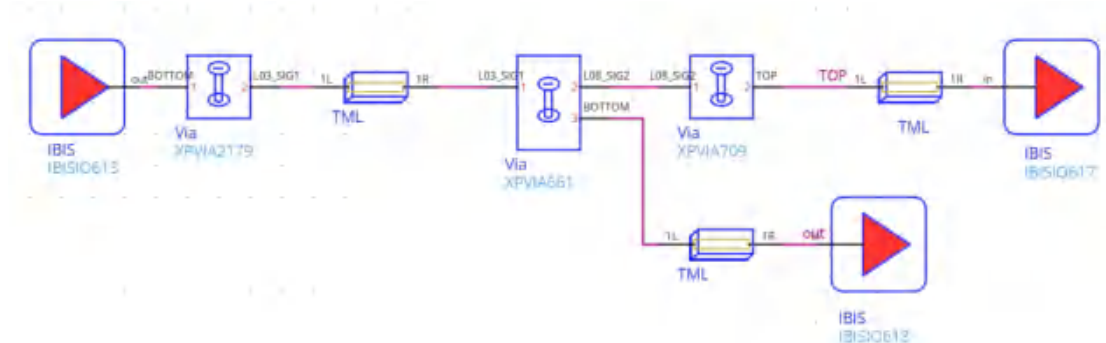


Relayout:

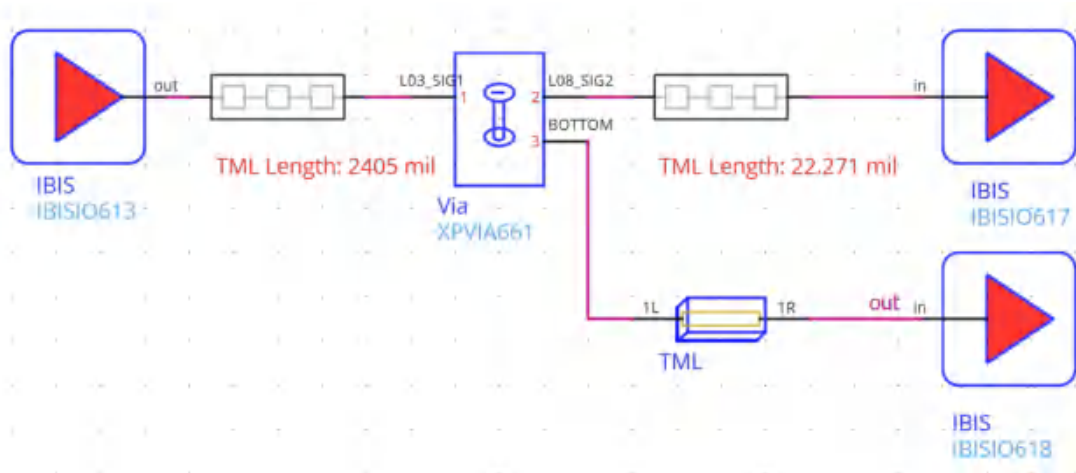
可将混乱的电路拓扑一键整理成树状排列结构:



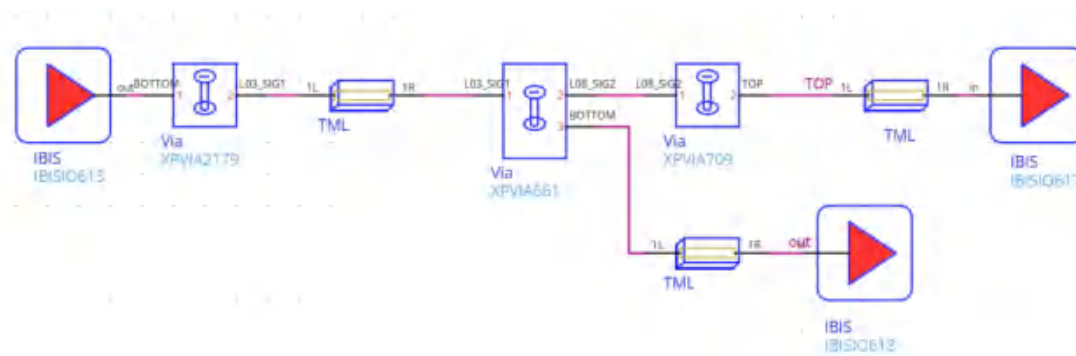
点击 relayout 后:



Collapse: 平铺显示的链路结构，可以折叠多段传输线和过孔，简化显示链路结构。



Expand:将折叠简化显示的链路恢复为原始的链路结构:



Rerun:

使用 Rerun 功能时，需要准备一份 batch simulation 的拓扑工程文件，同时需要使用管理员权限安装 ChannelExprt 的 exe。

在 AutoResult 中点击 simulate_result.html

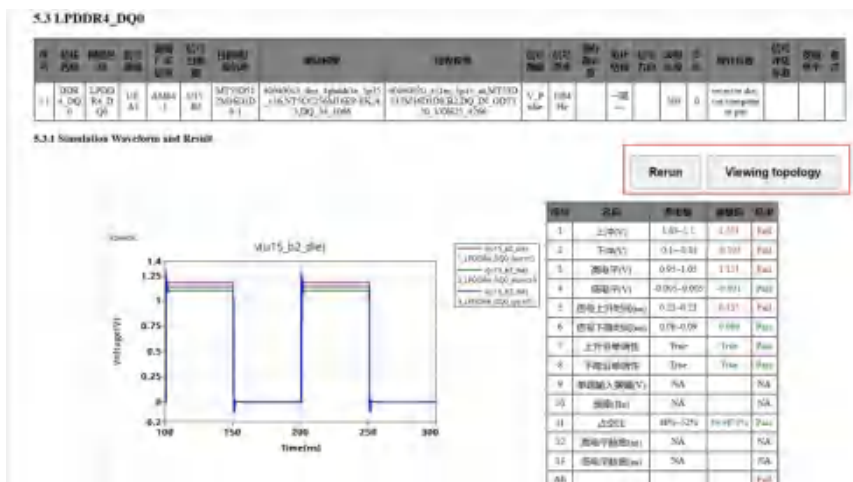
Name	Date modified	Type	Size
Suggestion	3/31/2025 1:40 PM	File folder	
Waveforms	3/31/2025 1:42 PM	File folder	
Compression.bat	3/31/2025 1:42 PM	Windows Batch file	1 KB
reportData.xml	3/31/2025 1:42 PM	XML Document	510 KB
simulate_result.docx	3/31/2025 1:42 PM	Microsoft Word Doc	12 MB
simulate_result.html	3/31/2025 1:42 PM	Chrome HTML Document	4.40 KB

在打开的网页中点击任意一个网络，会自动弹到对应的结果处。

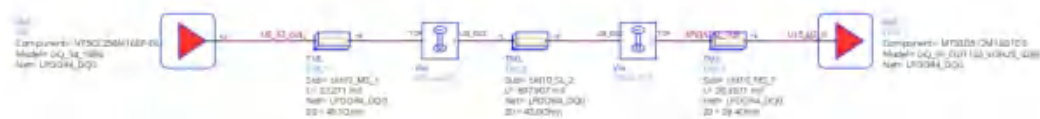
4 Summary

Net Count		Check Fail	Simulation Fail	Measure Fail
45		0	0	43

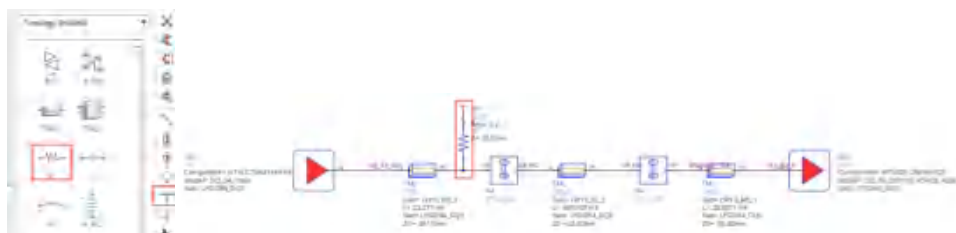
Excel Index	Index	Net	Check Result	Simulation Result	Measure Result
(1)	1.1	LPDDR4_DQ0	Pass	Pass	Fail
(1)	1.2	LPDDR4_DQ0	Pass	Pass	Fail
(2)	2.1	LPDDR4_DQ0	Pass	Pass	Fail
(3)	3.1	LPDDR4_DQ0	Pass	Pass	Fail
(4)	4.1	LPDDR4_CA0	Pass	Pass	Fail
(4)	4.2	LPDDR4_CA0	Pass	Pass	Fail
(5)	5.1	LPDDR4_CA0	Pass	Pass	Fail
(6)	6.1	LPDDR4_CA0	Pass	Pass	Fail



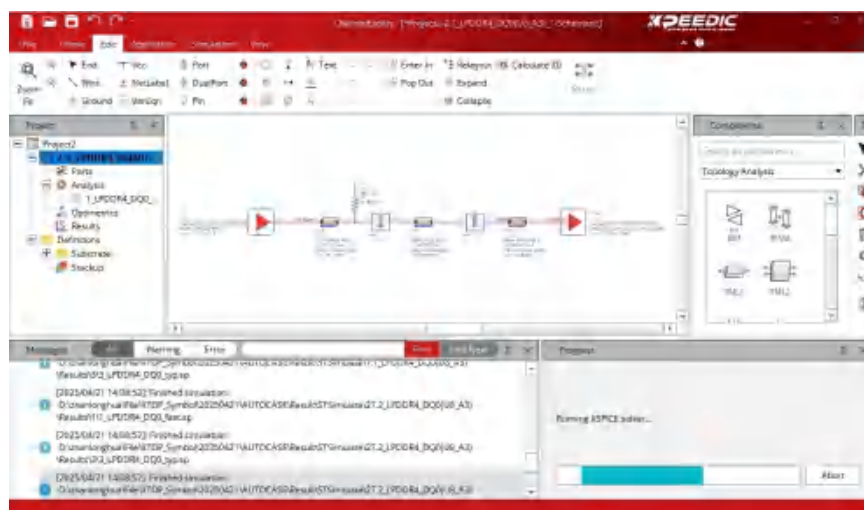
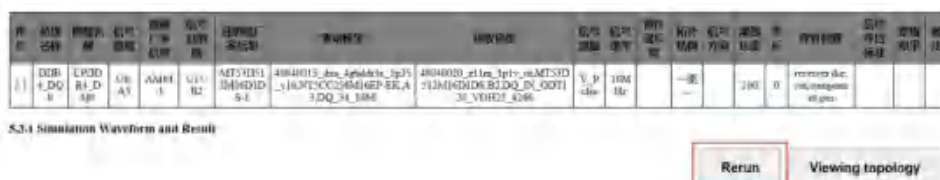
点击 Viewing topology，会自动打开 ChannelExpert 软件并展示拓扑结构。



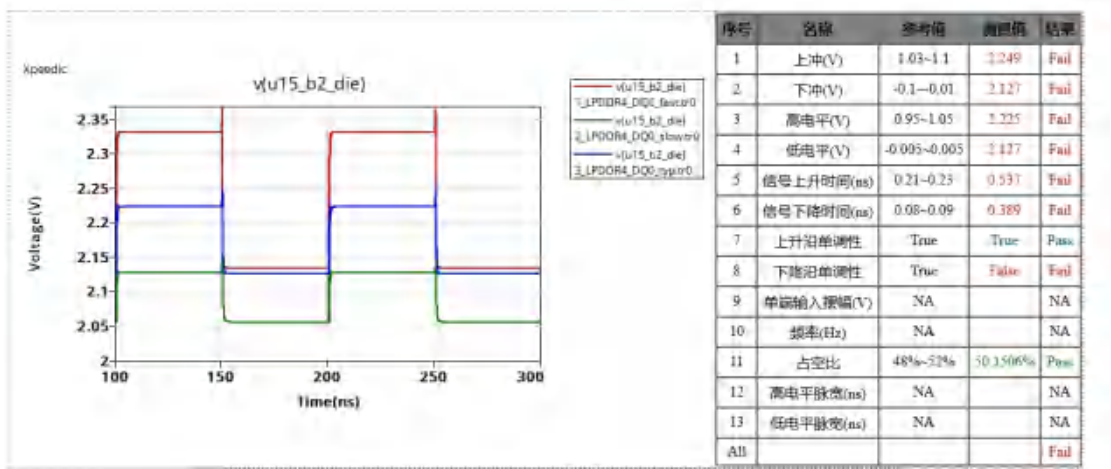
在 Topology Analysis 中选择 R，右边的器件库中选择 vcc，将其连接到拓扑结构中。



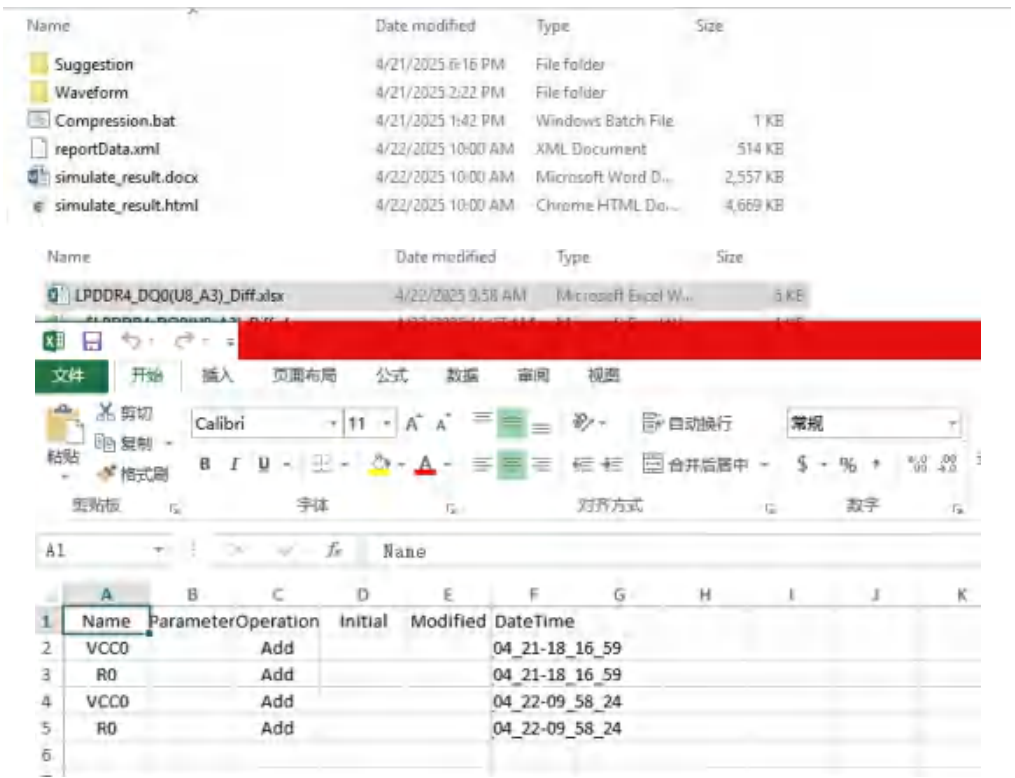
点击网页中相同网络中的 Rerun 按钮，触发批量更新报告功能。



等批量更新报告完成后，会自动弹出新的 html 报告，并更新测量数据和结果到新的报告中。



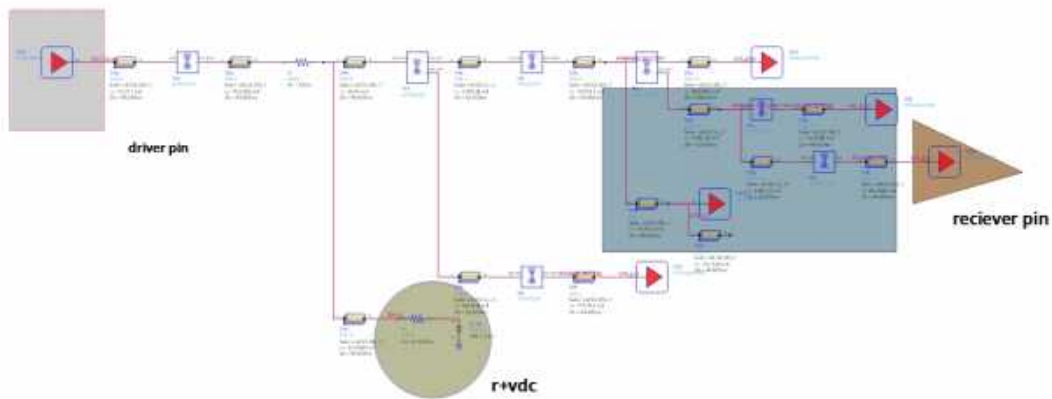
更新报告过程中的数据变更以 excle 的形式存放在 Suggestion 中。



4.1.5 图形注释

支持在原理图中加入注释型的信息，如矩形，多边形，圆，多线段，效果如下图所示：





4.2 SPICE 子电路



Spice 子电路符号

#	参数名	注释	范围	单位
1	SPICE File			
2	Current LIB			
3	SubcktPins			
4	Parameters			

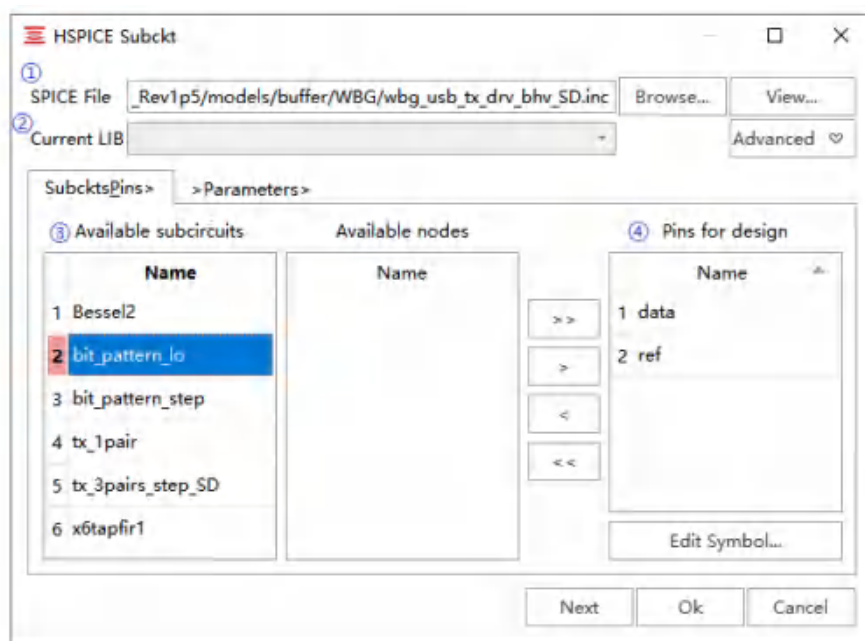
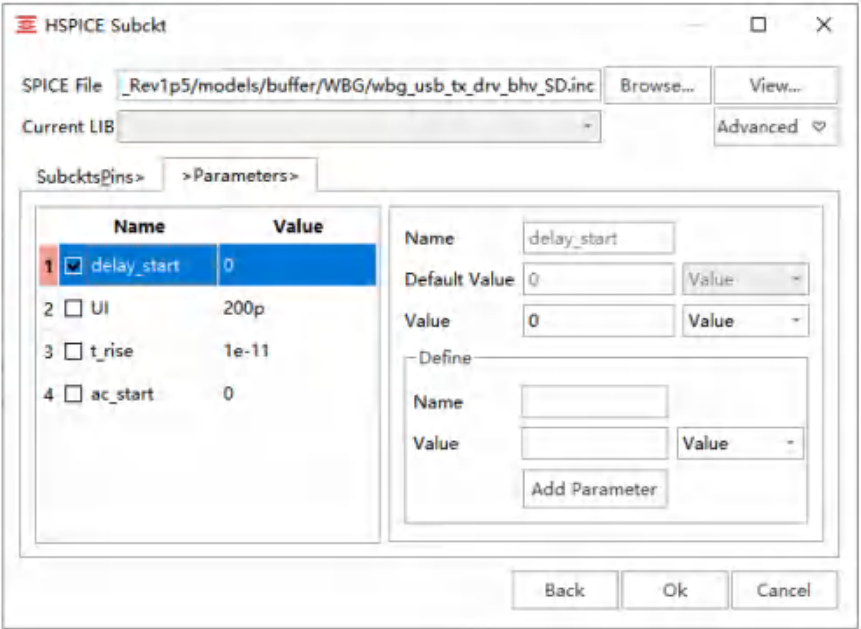


图1. Subckt 加载

从 component 栏找到对应的器件类将 subckt 拖入到原理图后，①点击 browse 加载需要仿真的 subckt 文件；②如果文件中包含 lib 则需要选择 lib；③选择需要的 subckt name，如下图选择的是 bit_pattern_IO；④将需要暴露的 pin 选择到 pins for design 窗口。

如果选择的 subckt 中存在参数或者用户需要额外定义参数可以在 Parameters 栏下进行操作。



4.3 LUMPED 集总模型

4.3.1 电阻 R

电阻符号如下：



图1. R 符号

#	属性名	默认值	范围	单位	注释
1	R	50Ω		Ω	电阻阻值

表1. R 属性表

4.3.2 电容 C

电容符号如下



图2. C 符号

#	属性名	默认值	范围	单位	注释
1	C	1		pF	电容容值

表2. C 属性表

4.3.3 电感 L

电感符号如下：



图3. L 符号

#	属性名	默认值	范围	单位	注释
1	L	1		nH	电感感值

表3. L 属性表

4.3.4 SRLC 串联器件

电阻电容电感串联器件 RLC 符号如下：

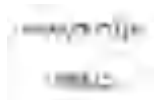


图4. SRLC 符号

#	属性名	默认值	范围	单位	注释
1	R	50		Ω	电阻阻值
2	L	1		nH	电感感值
3	C	1		pF	电容容值

表4. SRLC 属性表

4.3.5 单端端接电阻 TGND

单端端接电阻符号如下：



图5. TGND 符号

#	属性名	默认值	范围	单位	注释
1	R	50 Ω		Ω	电阻阻值

表5. TGND 属性表

4.3.6 差分端接电阻 T_Diff

差分端接电阻符号如下，其内部含有直流电压偏置。

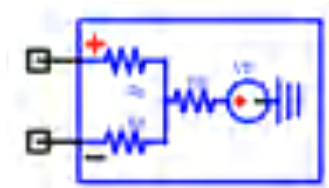


图6. T_diff 符号

#	属性名	默认值	范围	单位	注释
1	Ra	50Ω		Ω	Ra 阻值
2	Rb	0		Ω	Rb 阻值
3	Vb	0		V	直流偏置电压

表6. T_Diff 属性表

4.3.7 四端口巴伦 Balun4Port

巴伦 Balun 可以将差/共模信号与两个单端信号之间实现互相转换，其符号如下：



图7. Balun4Port 符号

其中 d 是差模信号端口，c 是共模信号端口，p/n 是单端信号的两个端口。
其各个端口的电压/电流满足：

$$\begin{aligned}
 V_d &= V_p - V_n \\
 i_d &= -(i_p - i_n)/2 \\
 V_c &= (V_p + V_n)/2 \\
 i_c &= -(i_p + i_n)
 \end{aligned}$$

式中各项含义如下：

V_d/i_d 分别是差模端口的电压/电流；

V_c/i_c 分别是共模端口的电压/电流；

V_p/i_p 分别是正端口的电压/电流；

V_n/i_n 分别是负端口的电压/电流。

如果用理想变压器来构建一个 balun，可以用以下等效拓扑

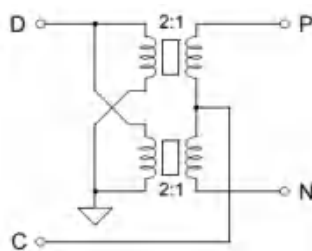


图8. 四端口 balun 等效电路

应用：对于差分通道可通过 balun 将信号分化为 differential 与 comm 分量，分别在对应的端口加入阻抗 R_d 、 R_c 与激励 V_d 、 V_c ，balun 的效果可以等效成下图所示：

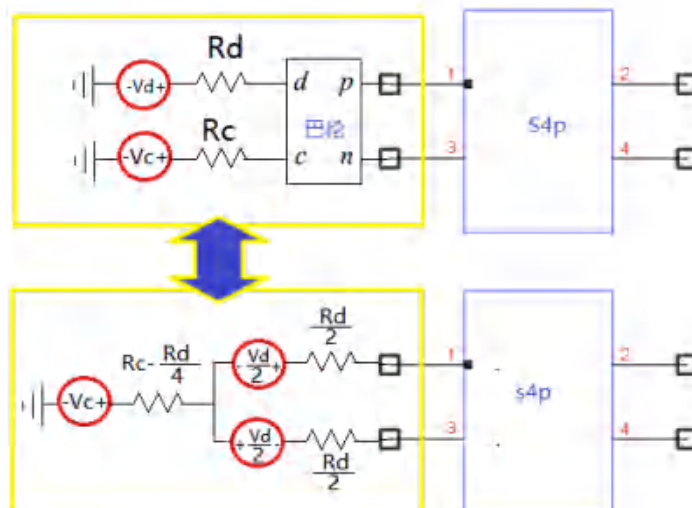


图9. Balun 等效作用

Balun 等效作用

当 $R_c = \frac{R_d}{4}$ 时，等效电路“Y”下方的电阻消失，总效果退化成两个单端信号的组合，即

常用的伪差分，真差分摆脱 $R_c = \frac{R_d}{4}$ 制约。

4.3.8 三端口巴伦 Balun3Port

如果不需要关注共模信号，则将共模端口接地处理，即电压 V_c 恒为 0，可以用三端口 Balun3Port，它的符号如下。



图10.Balun3Port 符号

其中 d 是差模信号端口， p/n 是单端信号的两个端口。

其各个端口的电压/电流满足：

$$\begin{aligned} V_d &= V_p - V_n \\ i_d &= -(i_p - i_n)/2 \\ V_p &= -V_n \\ i_p &= -i_n \end{aligned}$$

其中各项含义如下：

V_d/i_d 分别是差模端口的电压/电流；

V_p/i_p 分别是正端口的电压/电流；

V_n/i_n 分别是负端口的电压/电流。

如果用理想变压器来构建一个 3 端口 balun，可用以下等效拓扑

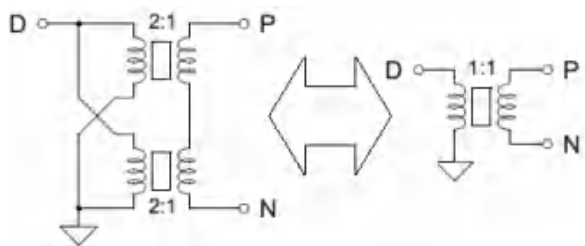


图11.三端口 balun 等效电路

4.4 独立源

4.4.1 V_Pattern 逻辑模式电压源



图12.V_Pattern 符号

#	属性名	默认值	范围	单位	注释
i	PAT				PAT 基本属性
1	Tui(Tsample)	1	>0	ns	每个符号的时间 UI
2	Vlow	0	>0	V	电压高值
3	Vhigh	1	>0	V	电压低值
4	Trise	50	>0	ps	上升沿时间
5	Tfall	50	>0	ps	下降沿时间
6	Tdelay	0		ns	延时
7	Data	b011100110 1100110	Binary		二进制码
8	RB	1	[1,length(Data)] 的整数		数据重复的起始位，如果小于 1 会自动当成 1。
9	R	1	≥-1 的整数		控制码流重复的次数。如果是-1 代表无限重复，小于-1 软件会自动设置为 0
ii	Encode				编码
10	No coder	1	Bool		无编码
11	DW8b10b	0	Bool		8b10b 编码
12	RD_INT	1	0 1		校验初始值
13	DW64b66b	0	Bool		64b66b 编码
14	TXC	00000000	Binary		64 比特数据会拆分为 8 个块的 8 比特的数据，TXC 含 8 比特二进制数据，用于指示每个块的类型： 0 表示对应的输入块是数据

					块；1 表示对应的输入块是控制块。
15	DW128b130b	0	Bool		128b130b 编码
16	DW128b132b	0	Bool		128b132b 编码
17	HEAD	DATA	DATA CONTROL		指示输入信号是数据信号还是控制信号。默认值为 DATA: DATA: 表示 128 位模式输入信号是数据信号。 CONTROL: 表示 128 位模式输入信号是控制信号。
iii	Modulation				多电平调制
18	NRZ		Bool		两电平
19	PAM3		Bool		三电平
20	PAM4		Bool		4 电平
21	PAM6		Bool		6 电平
22	PAM8		Bool		8 电平
23	PAM16		Bool		16 电平
24	Mapping		String		二进制映射到多电平调制的方式: PAM3: 0001021012202122 PAM8: 76450132 PAM16: 15;14;12;13;8;9;11;10;0;1;3;2; 7;6;4;5

表7. V_Pattern 属性表

4.4.2 V_PRBS 伪随机码电压源



图13.V_PRBS 符号

	属性名	默认值	范围	单位	注释
	PRBS				PRBS 基本属性
	Rate	1	>0	Gbps	码率
	Vlow	0	>0	V	电压高值
	Vhigh	1	>0	V	电压低值
	Trise	50	>0	ps	上升沿时间
	Tfall	50	>0	ps	下降沿时间
	Tdelay	0	>0	ns	延时
	Mode	User	User		模式选择

		Defined	Defined Maximal Length Bit File		
	Seed	21	>0 整数		种子
	Taps	[5,2]	整数序列		抽头位置
0	Register Length	8	>1		移位寄存器长度
1	Seed	170	>0		种子
i	Bit File				码型文件
0	Encode				编码
1	No coder	1	Bool		无编码
2	DW8b10b	0	Bool		8b10b 编码
3	RD_INT	1	0 1		校验初始值
4	DW64b66b	0	Bool		64b66b 编码
5	TXC	00000000	Binary		64 比特数据会拆分为 8 个块的 8 比特的数据，TXC 含 8 比特二进制数据，用于指示每个块的类型： 0 表示对应的输入块是数据块；1 表示对应的输入块是控制块。
6	DW128b130b	0	Bool		128b130b 编码
7	DW128b132b	0	Bool		128b132b 编码
8	HEAD	DATA	DATA CONTROL		指示输入信号是数据信号还是控制信号。 默认值为 DATA： DATA：表示 128 位模式输入信号是数据信号。 CONTROL：表示 128 位模式输入信号是控制信号。

ii	Modulation				多电平调制
9	NRZ		Bool		两电平
0	PAM3		Bool		三电平
1	PAM4		Bool		4 电平
2	PAM6		Bool		6 电平
3	PAM8		Bool		8 电平
4	PAM16		Bool		16 电平
5	Mapping		String		二进制映射到多电平调制的方式: PAM3: 0001021012202122 PAM8: 76450132 PAM16: 15;14;12;13;8;9;11;10; 0;1;3;2;7;6;4;5
v	Jitter				抖动
6	Rj Seed	1	>0 整数		RJ 种子
7	Rj RMS	0	≥0	ps	RJ 幅度
8	PJ Type	No PJ	No PJ sine sawtooth square		PJ 类型
9	PJ Amp	0	>0	ps	PJ 幅度
0	PJ Freq	01	>0	kHz	PJ 频率

表8. V_PRBS 属性表

4.4.3 V_DC 直流电压源



图14.V_DC 符号

#	属性名	默认值	范围	单位	注释
1	Vdc	5		V	直流电压值

表9. V_DC 属性表

4.4.4 I_DC 直流电流源



图15.I_DC 符号

#	属性名	默认值	范围	单位	注释
1	Idc	1		A	直流电流值

表10.I_DC 属性表

4.4.5 V_Pulse 脉冲电压源



图16.V_Pulse 符号

#	属性名	默认值	范围	单位	注释
1	Vlow	0	>0	V	电压高值
2	Vhigh	1	>0	V	电压低值
3	Trise	1	>0	ns	上升沿时间
4	Tfall	1	>0	ns	下降沿时间
5	Tdelay	0	≥0	ns	延时
6	Period	10	>0	ns	周期
7	Width	4	>0	ns	电压脉冲宽度
8	rmsjitter	10	≥0	ns	随机抖动幅值
8	perjitter	10	≥0	ns	周期抖动幅值
9	seed	21	>0 整数		种子

表11.V_Pulse 属性表

4.4.6 I_Pulse 脉冲电流源



图17.I_Pulse 符号

#	属性名	默认值	范围	单位	注释
1	I_low	0	>0	A	电流高值
2	I_high	1	>0	A	电流低值
3	Trise	1	>0	ns	上升沿时间
4	Tfall	1	>0	ns	下降沿时间
5	Tdelay	0	≥0	ns	延时
6	Period	10	>0	ns	周期
7	Width	4	>0	ns	电流脉冲宽度
8	rmsjitter	10	≥0	ns	随机抖动幅值
8	perjitter	10	≥0	ns	周期抖动幅值
9	seed	21	>0 整数		种子

表12.I_Pulse 属性表

4.4.7 V_Step 阶跃电压源



图18.V_Step 符号

#	属性名	默认值	范围	单位	注释
1	Vlow	0	>0	V	电压高值
2	Vhigh	1	>0	V	电压低值
3	Trise	1	>0	ns	上升沿时间
4	Tdelay	0	≥0	ns	延时

表13.V_Step 属性表

4.4.8 I_Step 阶跃电流源



图19.I_Step 符号

#	属性名	默认值	范围	单位	注释
1	I_low	1	>0	A	电流高值
2	I_high	2	>0	A	电流低值
3	Trise	1	>0	ns	上升沿时间

4	Tdelay	0	≥ 0	ns	延时
---	--------	---	----------	----	----

表14.I_Step 属性表

4.4.9 V_PWL 分段线性电压源



图20.V_PWL 符号

#	属性名	默认值	范围	单位	注释
1	Mode	Table	Table File		模式选择
2	Time#	10		ns	第 n 个时刻点
3	Voltage#	1		V	第 n 个时刻点对应的电压值
4	Wave File				时刻电压值文件

表15.V_PWL 属性表

4.4.10 I_PWL 分段线性电流源



图21.I_PWL 符号

#	属性名	默认值	范围	单位	注释
1	Mode	Table	Table File		模式选择
2	Time#	10		ns	第 n 个时刻点
3	Current#	1		mA	第 n 个时刻点对应的电流值
4	Wave File				时刻电流值文件

表16.I_PWL 属性表

4.4.11 V_AC 交流电压源



图22.V_AC 符号

#	属性名	默认值	范围	单位	注释
1	Vdc	1		V	直流偏置电压
2	Vac_mag	1	>0	V	AC 源幅值
3	Vac_pase	0		deg	AC 源相位
	Frequency	10		MHz	AC 源频率

表17.V_AC 属性表

4.4.12 I_AC 交流电压源



图23.I_AC 符号

#	属性名	默认值	范围	单位	注释
1	Idc	1		A	直流偏置电压
2	Iac_mag	2	>0	mA	AC 源幅值
3	Iac_pase	0		deg	AC 源相位
	Frequency	10		MHz	AC 源频率

表18.I_AC 属性表

4.4.13 V_Sin 正弦电压源



图24.V_sin 符号

#	属性名	默认值	范围	单位	注释
1	Vdc	0		V	直流偏置电压
2	Vamp	1		V	幅值
3	Frequency	100	>0	MHz	频率
4	Tdelay	1	≥ 0	ns	延时

表19.V_sin 属性表

4.4.14 I_Sin 正弦电流源



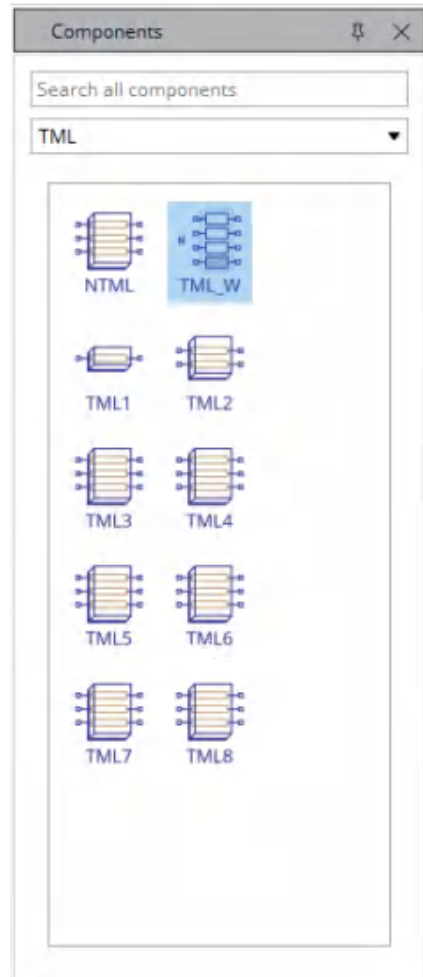
图25.I_sin 符号

#	属性名	默认值	范围	单位	注释
1	Idc	0		A	直流偏置电流
2	Iamp	1		A	幅值
3	Frequency	100	>0	MHz	频率
4	Tdelay	1	≥ 0	ns	延时

表20.I_sin 属性表

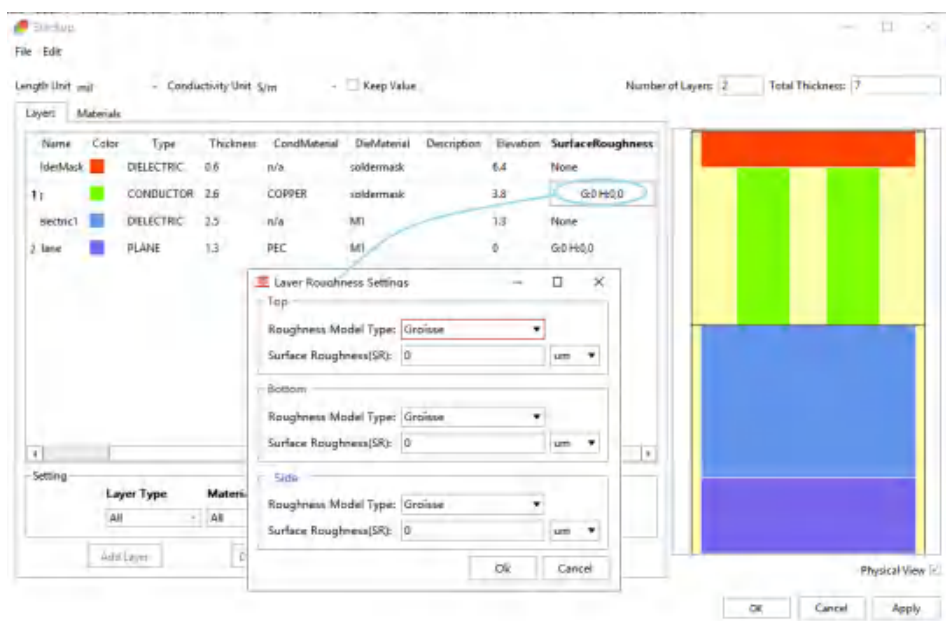
4.5 传输线 TML

4.5.1 基于叠层的结构的 2D 传输线



4.5.1.1 介质与导体粗糙度参数

粗糙度按 top、bottom、side 三个面分别配置粗糙度信息。



描述材料电学参数的几个物理量： $\rho\mu\sigma\epsilon$ ，分别是电阻率，磁导率，电导率，电容率。如果将传输线用等效电路来描述，则相关联的器件模型是RLGC，决定RLGC大小的材料物理量如下表所示。考虑到不同频率对应着不同的电流趋肤深度，导体表面粗糙度会影响R。

EM	TML	物理参数		材料参数	
电流	R	ρ / roughness		σ 电导率=5.8e7	$R' = R * K_{SR}$
	L	μ		μ 磁导率=1	
电场	G	σ		介质漏电=0	$\sigma_{complex} = \sigma + \omega * \epsilon_i$
		$\epsilon_{complex}$	ϵ_i	Df 损耗角=0.002	
	C	$= \epsilon_r - i * \epsilon_i$	ϵ_r	Dk 介电常数=3.3	

➤ 4.5.1.1.1 介质模型

● Multi-pole Debye Regular(一般多极点 Debye 模型)

多极点 Debye 介质模型由分布于实轴上的多个加权极点与一个常数共同决定，它不直接描述 Df 值，需要通过 $-\frac{imag(\epsilon(f))}{real(\epsilon(f))}$ 间接获得，其表达式如下所示：

$$\epsilon(f) = \epsilon(\infty) + \sum_{n=1}^N \frac{\Delta\epsilon_n}{1 + j \frac{f}{f_{rn}}}$$

下图表示的是一个有 16 个极点组成的 Multi-pole 模型，极点均匀分布在 log。右边分别是对应的 Dk 与 Df 值。

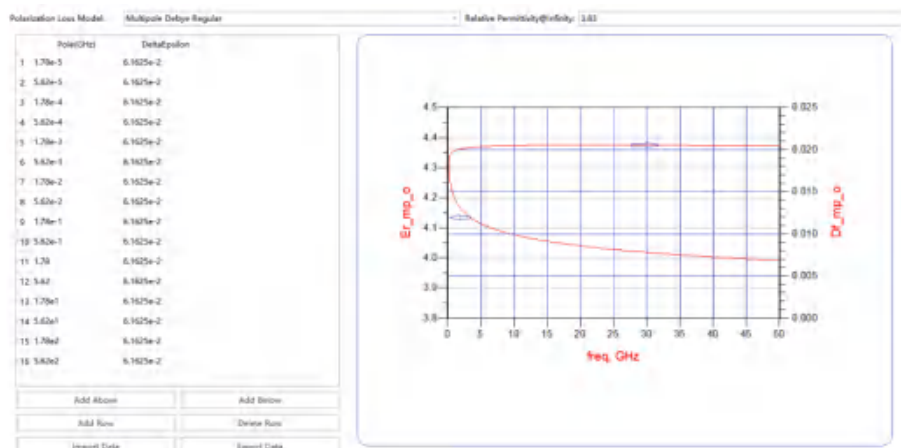


图26.Multipole Debye regular

● Multi-pole Debye(多极点 Debye 模型)

Multi-pole Debye 模型也是一种 Multi-pole Debye Regular 模型，输入变量从已知极点位置与权重换成用户已知频点的 DkDf 去反推全频段的 DkDf。

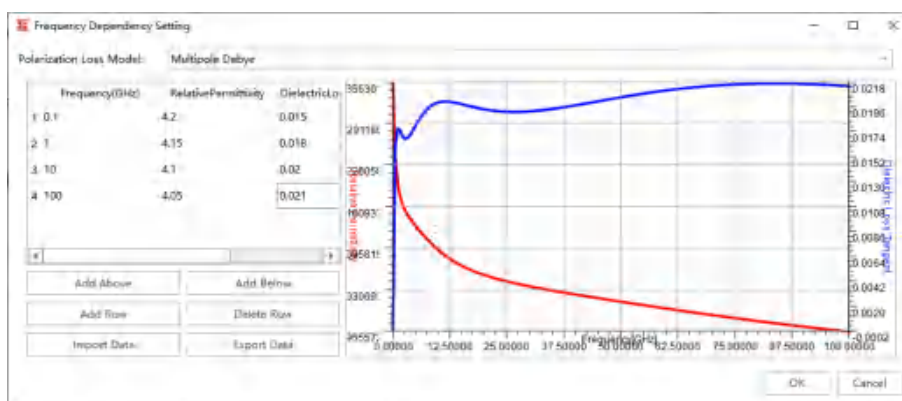


图27.Multipole Debye

■ Wideband Debye(宽带 Debye 模型/Djordjevic-Sarkar 模型)

当 ϵ_r 相等且 $n \rightarrow \infty$ 时，Multi-pole Debye 模型会趋近于 Wideband Debye 模型。也即

$\lim_{\epsilon_k \text{ 相等}, n \rightarrow \infty} \epsilon_r(\infty) + \sum_{k=1}^n \frac{\Delta \epsilon_k}{1 + i \frac{f}{f_{rk}}}$ 如下图所示:

$$\begin{aligned} \epsilon_{wd}(f) &= \epsilon(\infty) + \Delta \epsilon * \frac{1}{(m_2 - m_1)} \lg \frac{f_2 + jf}{f_1 + jf} = \epsilon(\infty) + \Delta \epsilon * \log \frac{f_2 + jf}{f_1 + jf} \\ &= \epsilon_r * (1 - j * D_f) = \epsilon_r * \left(1 - j * \frac{\sigma}{2\pi f \epsilon_r \epsilon_0} \right) \end{aligned}$$

其中

$$\begin{cases} \epsilon_r = \epsilon_{\infty} + \Delta \epsilon * \frac{1}{m_2 - m_1} * \frac{1}{\log_e(10)} * \frac{1}{2} \log_e \left(\frac{f_2^2 + f^2}{f_1^2 + f^2} \right) \\ \sigma = 2\pi f * \epsilon_0 * \Delta \epsilon * \frac{1}{m_2 - m_1} * \frac{1}{\log_e(10)} * \left(\tan^{-1} \left(\frac{f}{f_1} \right) - \tan^{-1} \left(\frac{f}{f_2} \right) \right) \end{cases}$$

$\epsilon_{wd}(f)$ 表达式中只有两个未知数， $\epsilon(\infty)$ 和 $\Delta \epsilon$ 。假设已知频率 f 对应的材料参数模型是 ϵ_r 、 $\tan \delta$ ，可以得到对应的 $\epsilon(\infty)$ 和 $\Delta \epsilon$ ，分别是

$$\varepsilon(\omega) = \varepsilon_{\infty} \left(1 + \tan \delta \cdot \frac{\operatorname{Re} \left(\ln \left[\frac{f_2 + jf}{f_1 + jf} \right] \right)}{\operatorname{Im} \left(\ln \left[\frac{f_2 + jf}{f_1 + jf} \right] \right)} \right)$$

$$\Delta \varepsilon = - \frac{\tan \delta \cdot \varepsilon_{\infty} \cdot \ln(10) \cdot [m_2 - m_1]}{\operatorname{Im} \left(\ln \left[\frac{f_2 + jf}{f_1 + jf} \right] \right)}$$

■ 用 Multi-pole Debye 模型来逼近 Wideband Debye 模型

Wideband Debye 是假设在 $\log f_1$ 与 $\log f_2$ 之间均匀分布无穷多个一样权重的极点，然而工程上只需要用有限个极点就可以较好地等效 wideband Debye 模型。

其中一种等效方法是在 $\log f_1 \sim \log f_2$ 之间均匀插入了 N 个极点，因极点在 $\log$ 空间均匀，故满足 $\frac{f_{rn}}{f_{rn-1}} = \left(\frac{f_2}{f_1}\right)^{1/N}$ ，第一个极点 $f_{r1} = f_1 * \left(\frac{f_2}{f_1}\right)^{1/2N}$ ，让每个极点的权重系数相等

$$\Delta \varepsilon(\text{multipole}) = \frac{1}{N} \Delta \varepsilon(\text{wideband}).$$

$$\text{最终可以表示为 } \varepsilon(f) = \varepsilon(\infty) + \frac{\Delta \varepsilon(\text{wideband})}{N} \sum_{n=1}^N \frac{1}{1 + j \frac{f}{f_{rn}}}$$

为了使 $\varepsilon(f)$ 曲线光滑，两个极点之间的距离需要满足 $\frac{f_{rn}}{f_{rn-1}} \leq \sqrt{10}$ ，也即 $N \geq 2 \lg \frac{f_2}{f_1}$ ， N 越大，曲线就越接近与 wideband debye 模型。当 $f_1=1\text{e}4, f_2=1\text{e}12$ 时，建议 $N \geq 16$ 。

Wideband 的是假设所有极点权重系数相等，因此对于有些材料不能同时很好地表示高低频的损耗，这时可以考虑放开每个极点的权重系数 $\Delta \varepsilon(\text{multipole})$ 相等的约束，让各个极点权重系数自由调节。

➤ 4.5.1.1.2 铜箔粗糙度模型

导体中电流随着频率的升高逐渐分布于导体的表面，从与电流方向垂直的横切面来看，足够高的频率时导体的中心部分几乎没有电流流过，电流有趋于集中在导体表面的趋势，当场强变成 $1/e$ 时的距离称为趋肤深度可表示为 $\delta(f) = \sqrt{\frac{1}{\pi f \mu \sigma}}$ ，可以看到频率越高则趋肤深度越小，越是可以看成电流都分布于导体表面。然而实际中的导体不是理想光滑的，当使用的导体粗糙度与随着频率而变小的趋肤深度可以比拟时，电流流过的路径会变得复杂，导体损耗加剧。如果要精确模拟导体损耗，需要引入粗糙度系数来对导体损耗做矫正，目前常用的模型有 Grosse 模型、Hammerstad 模型、Huray 雪球模型、炮弹球堆放 Huray 模型、半球模型。用于描述导体粗糙程度的物理有这样几种 R_a 、 R_z 、 R_q 。

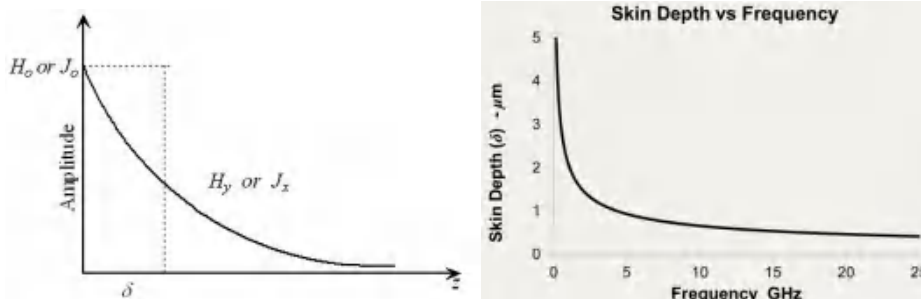


图28.H/J 场强与深度的关系

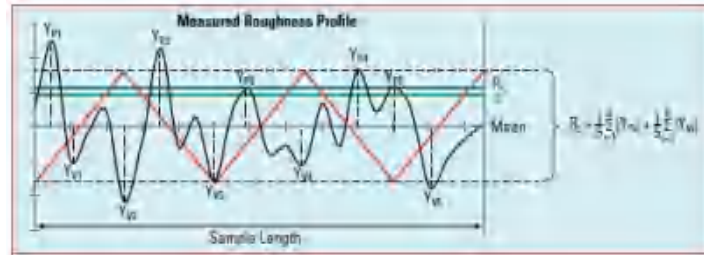


图29.粗糙度计量

类型	描述	
Ra	算术平均粗糙度	
Rz	峰谷平均	10 点测量值
Rq	均方根粗糙度	

表21.几个常见的粗糙度量

	Rz(um)	Ra	Rq	
基础	9.5/7.2			
VLP	3.0-4.0	0.3-0.35	0.30-0.40	
HVLP	2.0-3.0	0.25-0.3	0.25-0.35	
VLP2	1.5	0.15-0.17	0.15-0.20	
VLP3	1.0	0.12-0.15	0.12-0.17	
http://www.co-tech.com/_en/02_product/01_detail.php?MainID=2				

表22.铜箔类型

■ Hammerstad 模型及其修正模型

Hammerstad 模型于 1948 年提出，它是一种数据拟合方式确定的模型，模型输入只需要表面粗糙度（SR），然而在超过 5GHz 的频率时模型趋于饱和，使用受到很大限制。为了拓展模型的适用性增加粗糙度系数（RF）来抗饱和。

$$K_{SR} = 1 + \left(\frac{2}{\pi} * \tan^{-1} \left[1.4 \left(\frac{SR}{\delta(f)} \right)^2 \right] \right) * (RF - 1)$$

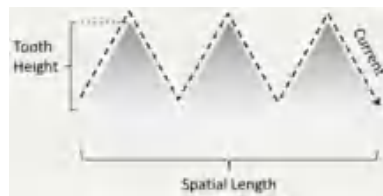


图30.Hammerstad 等效模型

Top Conductor Surface
 Roughness Model Type: ModifiedHammerstad
 Surface Roughness(SR): um
 Roughness Factor(RF):

图31.Modified Hammerstad 用户界面

■ Grosse 模型

Grosse 模型比传统 Hammerstad 模型使用的频率范围更宽，可以适用的频率大概是 7~10GHz。

$$K_{SR} = 1 + e^{-\left(\frac{\delta(f)}{2SR}\right)^{1.6}}$$



图32.Grosse 用户界面

■ Huray snowball 雪球模型

Huray SnowBall 模型将导体粗糙的表面用堆叠小球来模拟，如果能够得到模型中的系数，可以等效到很高的频段。其模型表述如下

$$K_{SRH}(f) = \frac{A_{\text{m\u00f6tte}}}{A_{\text{flat}}} + \frac{3}{2} \sum_{i=1}^j \frac{\left(\frac{N_i \times 4\pi a_i^2}{A_{\text{flat}}} \right)}{\left(1 + \frac{\delta(f)}{a_i} + \frac{\delta^2(f)}{2a_i^2} \right)}$$

Ni 表示半径是 ri 的球的个数，Ahex 是堆叠球体的平面面积，δ是趋肤深度。



图31.雪球模型

- ① 如果假设所有的球体体积相等，模型可以简化成下面的式子。

$$\begin{aligned} \frac{P_{\text{rough}}}{P_{\text{smooth}}} &= \frac{A_{\text{rough}}}{A_{\text{smooth}}} + \frac{3}{2} \left(\frac{4N\pi r^2}{A_{\text{flat}}} \right) \left(\frac{1}{1 + \frac{\delta(f)}{r} + \frac{\delta^2(f)}{2r^2}} \right) \\ &\approx 1 + \frac{3}{2} \cdot SR \cdot \left(\frac{1}{1 + \frac{\delta(f)}{r} + \frac{\delta^2(f)}{2r^2}} \right) \end{aligned}$$

KH 的最大值是 1+1.5*SR，如果 SR=2.9，则 KH=5.35

- ② 如果 i=2 那么则是 2 阶的雪球模型。

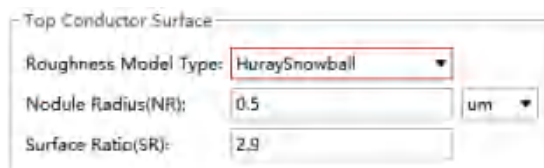
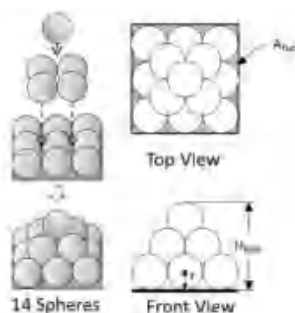


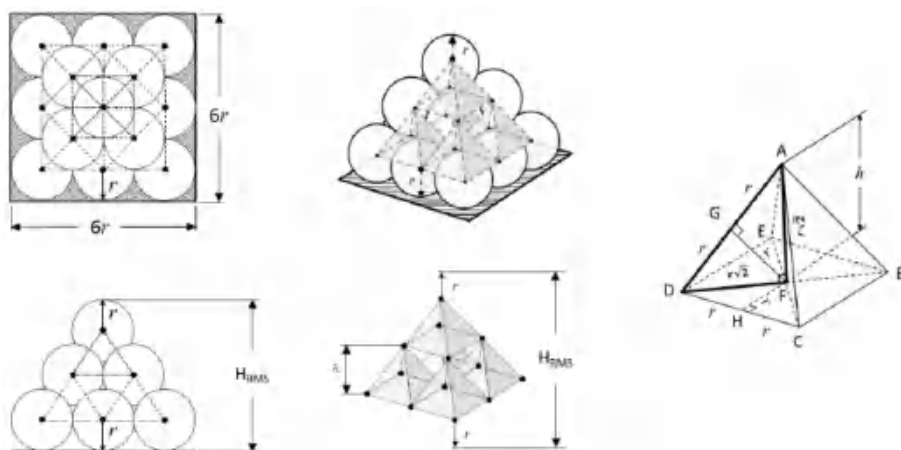
图33.Huray Snowball 用户界面

■ Cannonball-huray 炮弹模型

Huray Snowball 没有规定球体的堆积方式，如果将粗糙的表面用如下图所示的形式来等效，即最下面 9 个，第二层 4 个，第一层 1 个，一共 14 个球来等效，可以简化模型的输入参数，这种堆积方式称为炮弹堆积。



炮弹堆积模型



计算方式

根据图中的堆积方式可以推知 Huray Snowball 半径公式

$$r = \frac{H_{RMS}}{2(1+\sqrt{2})} = \frac{\frac{R_z}{2\sqrt{3}}}{2(1+\sqrt{2})} \approx 0.06R_z, \text{ 粗糙度系数 } SR = \frac{4N\pi r^2}{A_{flat}} = \frac{14\pi}{9} = 4.88$$

炮弹球模型只是一种特殊堆积的雪球模型，并不会比雪球模型更加的精确，只是提供了一种更方便的粗糙度表示方式，在一定范围内可以适用。

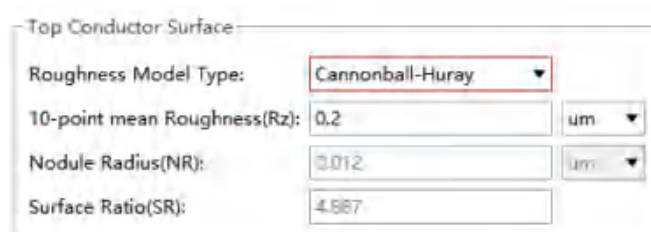


图32.Cannonball Huray 用户界面

■ 因果粗糙度模型

传输线的可以用 RLGC 模型来等效， $Z=R+j\omega L$ ， $Y=G+j\omega C$ 。

传播常数 $\gamma=\sqrt{Z*Y}=\alpha+j\beta$;

特征阻抗 $Z_0=\sqrt{Z/Y}=\sqrt{(R+j\omega L)(G+j\omega C)}$

Z 与 Y 均是具有实际物理意义的阻抗与导纳，不能违反因果性质，其中 Y 的因果性由满足因果的介质模型(wideband/Multipole)来保证。

当加入粗糙度系数时阻抗式子变成

$$Z=R*K_{sr}+j\omega L,$$

新的 Z 不再满足希尔伯特变换，破坏了因果性。正确的做法是

$$Z=(R+j\omega L)K_{sr},$$

粗糙度类型	Non-causal	causal
Groisse	$K_{SR} = 1 + e^{-\left(\frac{\delta(f)}{2SR}\right)^{1.6}}$	Non
Modified Hammerstad	$K_{SR} = 1 + \left(\frac{2}{\pi} * \tan^{-1} \left[1.4 \left(\frac{SR}{\delta(f)} \right)^2 \right] \right) * (RF - 1)$	$K = 1.0 + \frac{2}{\pi} * \left(\lg \frac{\sqrt{1+ix}}{1+\sqrt{ix}} - \tan^{-1} \sqrt{ix} \right) * (RF - 1)$ 其中 $x = 1.4 \left(\frac{SR}{\delta(f)} \right)^2$
Huray Snowball	$K_H(f) = 1 + \frac{3}{2} * SF * \left[\frac{1}{1 + \frac{\delta(f)}{r} + 0.5 \left(\frac{\delta(f)}{r} \right)^2} \right]$	$K_H(if) = 1 + \frac{3}{2} * SF * \left[\frac{1}{1 - 0.5(1-i) \frac{\delta(f)}{r}} \right]$
Cannonball-Huray	$K_{H'}(f) = 1 + 84 * \frac{\pi r^2}{A_{flat}} * \left[\frac{1}{1 + \frac{\delta(f)}{r} + 0.5 \left(\frac{\delta(f)}{r} \right)^2} \right]$	$K_{H'}(if) = 1 + 84 * \frac{\pi r^2}{A_{flat}} * \left[\frac{1}{1 - 0.5(1-i) \frac{\delta(f)}{r}} \right]$

4.5.2 W-element

在 component 栏将 TML-W 器件拖入到原理图中，选择需要的 w-element 后，可以设置需要的长度。

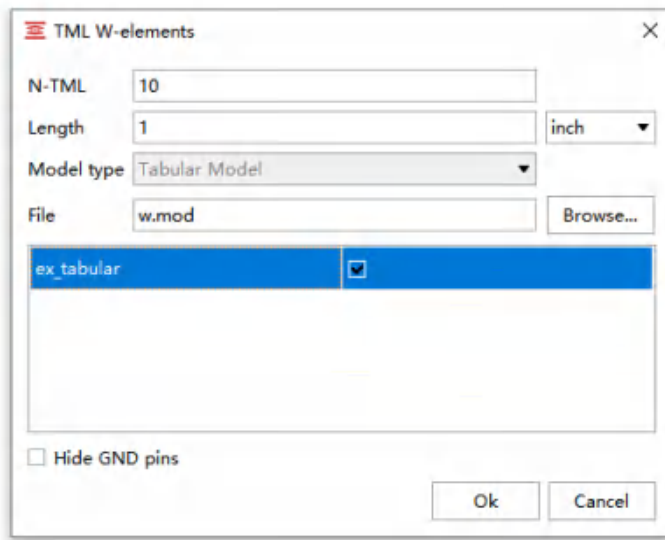


图33.w-elment

4.5.3 理想传输线 ideal TML

理想传输线可以用传播常数 γ 与长度来表示，它的 ABCD 参数可以表示成

$$\begin{vmatrix} A & B \\ C & D \end{vmatrix} = \begin{vmatrix} \cosh(\gamma L) & Z_c \sinh(\gamma L) \\ \frac{\sinh(\gamma L)}{Z_c} & \cosh(\gamma L) \end{vmatrix}.$$

传播常数可以表示为 $\gamma = \alpha + i\beta$,

其中 α 是传播常数的实部，称为衰减常数，单位是 Np/m 或 dB/m;

β 是传播常数的虚部，称为相位常数，单位是 rad/m。

4.5.3.1 TMLD

TMLD 理想单端传输线：通过特征阻抗 Z_c 与延时 Delay 来描述的单端无衰减理想传输线。



图34.TMLD 图标

	属性名	默认值	范围	单位	注释
	Zc	50	> 0	Ω	特征阻抗
	Delay	0.178	$\neq 0$	ns	传播延时
	Length	1000	$\neq 0$	mil	几何长度

	Vp	5600	> 0	mil/n s	传播速度
● 其中 Delay 与 Length 只能二选一, 另一个用于显示, 它们的关系满足 $\text{Delay}=\text{Length}/Vp$ 。					

表23.TMLD 属性表

4.5.3.2 CTMLD

CTMLD 理想差分传输线：通过差模特征阻抗 Z_d 、差模传播延时 Delay 与耦合系数 Kc 来描述的差分无衰减理想传输线

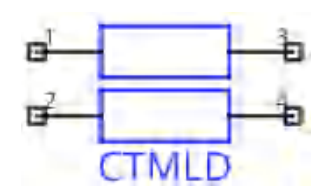


图35.CTMLD 图标

#	属性名	默认值	范围	单位	注释
1	Zd	100	>0	Ω	差模特征阻抗
2	Delay_odd	0.178	$\neq 0$	ns	差模传播延时
3	Length	1000	$\neq 0$	mil	几何长度
4	Vp_odd	5600	>0	mil/ns	差模传播速度
5	Kc	0	[0,1]		耦合系数
6	Kvp	1	>0		共模与差模的传播速度比例系数
耦合系数 Kc 表示差分对之间的耦合强度：可以用 $Kc=(4*Zc/Zd-1)/(4*Zc/Zd+1)$ 来计算，其中 Zc 是共模阻抗。如 Zd 是 100 Ω ，Zc 是 25 Ω ，则耦合系数是 0； Length 与 Delay_odd 只能二选一，另一个只用于显示，它们的关系满足 $\text{Delay_odd}=\text{Length}/Vp_odd$ ；					

表24.CTMLD 属性表

4.5.3.3 TMLE

TMLE 理想单端传输线：通过特征阻抗 Z_c 与指定频率 F 下的电长度 E 来描述的单端无衰减理想传输线。

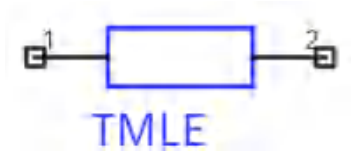


图36.TMLE 图标

#	属性名	默认值	范围	单位	注释
1	Zc	50	>0	Ω	特征阻抗
2	E	90	>0	deg	电长度

3	F	1	>0	GHz	电长度对应的频率
传输线对应的延时 Delay= 1/F*(E/360)					

表25.TMLE 属性表

4.5.3.4 CTMLE

CTMLE 理想差分传输线：通过特征阻抗 Z_e/Z_o 与指定频率 F 下的电长度 E 来描述的差分无衰减理想传输线。

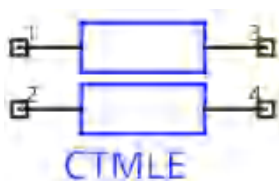


图37.CTMLE 图标

	属性名	默认值	范围	单位	注释
	Z_e	100	>0	Ω	差模特征阻抗
	Z_o	25	>0	Ω	共模特征阻抗
	E	90	>0	deg	电长度
	F	1	>0	GHz	电长度对应的频率
传输线对应的延时 Delay= 1/F*(E/360)					

表26.CTMLE 属性表

4.5.3.5 TMLP

TMLP 有损单端传输线：通过特征阻抗 Z_c 、几何长度 Length 与单位长度衰减系数 A 来描述的有损单端传输线

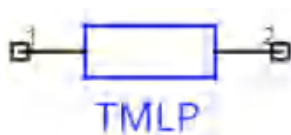


图38.TMLP 图标

#	属性名	默认值	范围	单位	注释
1	Z_c	50	>0	Ω	特征阻抗
3	Length	1000	$\neq 0$	mil	几何长度
4	K	4		无	相对介电常数
5	A	0.0001	dB/m	无	单位长度衰减系数
6	F	1	>0	GHz	衰减系数 A 对应的频率
传播常数: $\gamma = \alpha + i\beta$, 其中 $\alpha = A\sqrt{\frac{f}{F}}$, $\beta = \frac{2\pi f}{c/\sqrt{K}}$; 传播系数 $\gamma L = (\alpha + i\beta)L \Rightarrow \gamma L = \left(A\sqrt{\frac{f}{F}} + i\frac{2\pi f}{c/\sqrt{K}}\right)L$; 其中 f 是仿真频率					

表27.TMLP 属性表

4.5.3.6 CTMLP

CTMLP 有损差分传输线：通过特征阻抗 Z_e/Z_o 、几何长度 $Length$ 与单位长度衰减系数 A_e/A_o 来描述的有损差分传输线

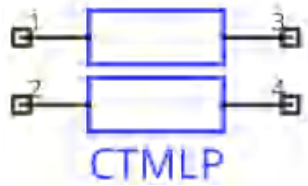


图39.CTMLP 图标

	属性名	默认值	范围	单位	注释
	Z_e	100	>0	Ω	差模特征阻抗
	Z_o	25	>0	Ω	共模特征阻抗
	$Length$	1000	$\neq 0$	mil	几何长度
	K_e	4	>0	无	差模的相对介电常数
	K_o	4	>0		共模的相对介电常数
	A_e	0.0001	dB/m	无	差模的单位长度衰减系数
	A_o	0.0001	dB/m	无	共模的单位长度衰减系数
	F	1	>0	GHz	衰减系数 A_e/A_o 对应的频率
传播常数: $\gamma = \alpha + i\beta$, 其中 $\alpha = A\sqrt{\frac{f}{F}}$, $\beta = \frac{2\pi f}{c/\sqrt{K}}$; 传播系数 $\gamma L = (\alpha + i\beta)L \Rightarrow \gamma L = \left(A\sqrt{\frac{f}{F}} + i\frac{2\pi f}{c/\sqrt{K}} \right) L$; 其中 f 是仿真频率					

表28.CTMLP 属性表

4.6 IBIS 模型

支持 IBIS 规范里的 17 种基本 buffer，以及单端组成的伪差分 buffer。其类型见下表罗列：

ID	Buffer 类型	单端	差分	备注
1	Input	√	√	
2	Output	√	√	
3	I/O	√	√	
4	3-state	√	√	
5	Open_drain	√	√	Axx 将 open_drain 与 open_sink 看成同类型
6	I/O_open_drain	√	√	
7	Open_sink	√	√	

8	I/O_open_sink	√	√	
9	Open_source	√	√	
10	I/O_open_source	√	√	
11	Input_ECL	√	√	
12	Output_ECL	√	√	
13	I/O_ECL	√	√	
14	3-state_ECL	√	√	
15	Series		搭配	可单独使用或用作差分 buffer 的 P/N 跨接
16	Series_switch		搭配	
17	Terminator	√	√	

表29.IBIS buffer 列表

如果将 buffer 按照类型相关的关系，可以制成下表：

	none	open drain	open sink	open source	ECL
Input	1#				11#
Output	2#	5#	7#	9#	12#
IO	3#	6#	8#	10#	13#
3-state	4#				14#
Series	15#				
Series_switch	16#				
Terminator	17#				

图40.ibis buffer 在 spice 中的序列号

在 ChannelExpert 中的 symbol 库如下图所示：

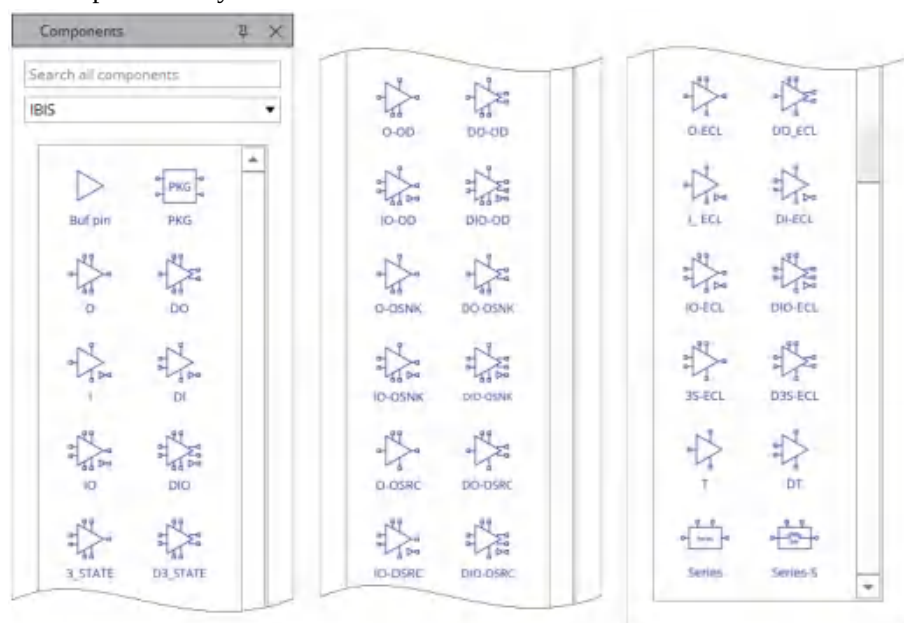


图41.IBIS 模型库

ibis 的 corner 配置参数 typ 的名称与数值见下图：

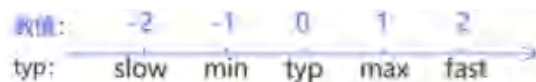


图42.Typ 的类型名与数值对应关系

如果 typ 的值是 typ、min 或 max 中的任一个，表示直接从 ibis 文件提取对应的列数据；如果 ibis 文件里不存在 min/max 数据，则使用 typ 数据；如果 typ 的值是 fast 或 slow，则使用 min 和 max 的数据组合。表 xxx 罗列了 fast 和 slow 所使用的具体数据类型。

参数	Fast	Slow
C_comp	min	max
Temp_Range	max	min
Voltage_Range	max	min
Pullup_Ref	max	min
Pulldown_Ref	min	max
POWR_Clamp_Ref	max	min
GND_Clamp_Ref	min	max
Rgnd	max	min
Rpower	max	min
Rac	max	min
Cac	min	max
Pulldown	max	min
Pullup	max	min
GND_Clamp	max	min
Power_Clamp	max	min
Ramp	max	min
Rising_waveform	max	min
Falling_waveform	max	min
V_fixture	max	min
Model Spec	max	min

表30.Fast 和 Slow 数据组合表

根据实际的使用需求，有一些通用的 IBIS 配置，如下图

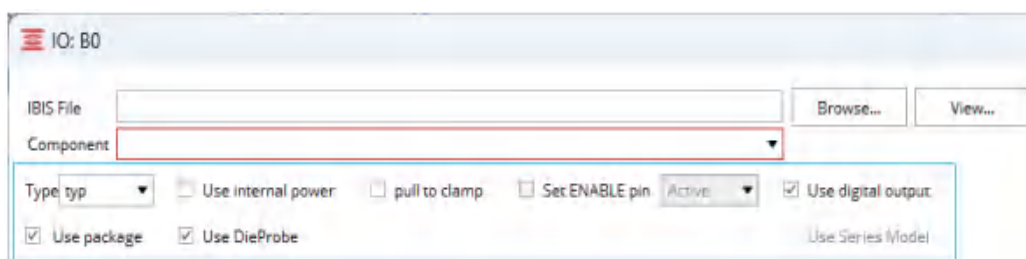


图43.IBIS 通用配置

各部分参数配置如下表

#	项	注释	默认	*备注
1	Type	Corner 选择	typ	Typ=[typ min max fast slow]

2	Use internal power	是否用内部电源	1	如果启用外部电源，则 pull 与 clamp 同时隐藏，@power=on
3	pull to clamp	Pull 是否关联到 clamp	0	一般情况下 pull 与 clamp 接到相同电位
4	set ENABLE pin	Enable 使能电平显示	0	由 ibis 文件中的 Enable=Active-High/Active-low 决定
6	Use digital output	是否要数字输出	0	数字判决后的电平
7	Use package	是否用 package	0	控制 package=0 1 2 3 4，详见 3.2.4 小节
8	Use DieProbe	是否露 Die Pad	0	

表31.IBIS 通用配置表

4.6.1 根据 pin 自动识别 buffer

自动识别器件 buf pin 可以根据用户选择的 Buffer 类型来自动匹配以上 15 种(除 series/series switch)之一的对应的 symbol。

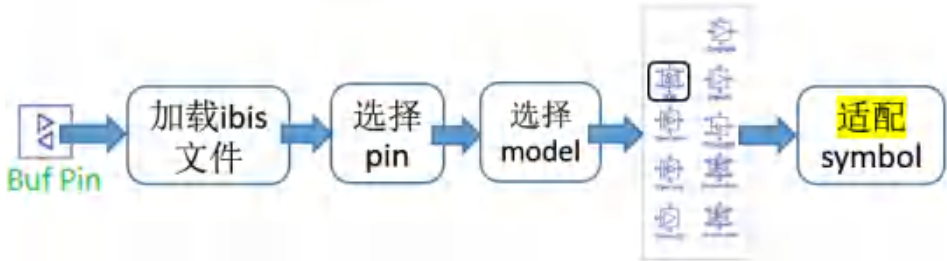


图44.Buf pin 使用流程

4.6.2 Buffer 类型

4.6.2.1 Output Buffer

Output 输出 buffer 的 symbol 如下：

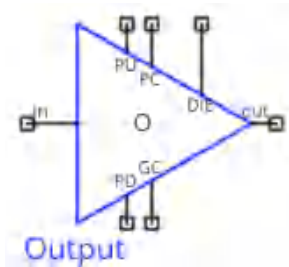


图45.Output 图标

其内部结构原理如下

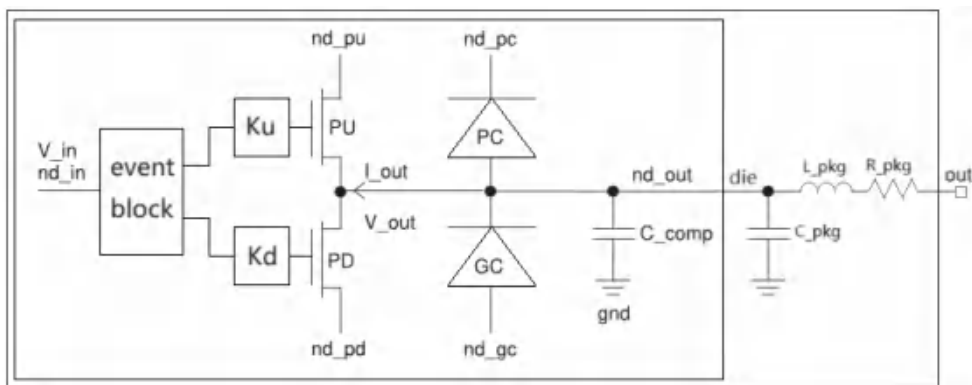


图46.Output 结构

nd_in 管脚是数字激励输入，在初始状态 $t=0$ 时与 0.5V 比较判决，在 $t>0$ 时刻与 0.2V/0.8V 迟滞比较判决，判决的逻辑值受到 Polarity 的影响，具体见下表。

Polarity=Non-Inverting		
初始状态 $t=0$		$V_{in} > 0.5?1:0$
$t>0$	$V_{in}>0.8$	state=1
	$V_{in}<0.2$	state=0
Polarity=Inverting		
初始状态 $t=0$		$V_{in} < 0.5?1:0$
$t>0$	$V_{in}>0.8$	state=1
	$V_{in}<0.2$	state=0

表32.Output 激励真值表

4.6.2.2 Input Buffer

Input 输入 buffer 的 symbol 如下：

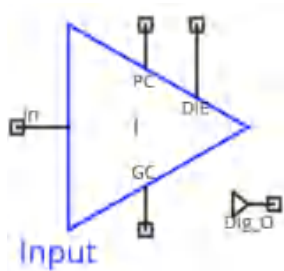


图47.Input 图标

其内部结构原理如下

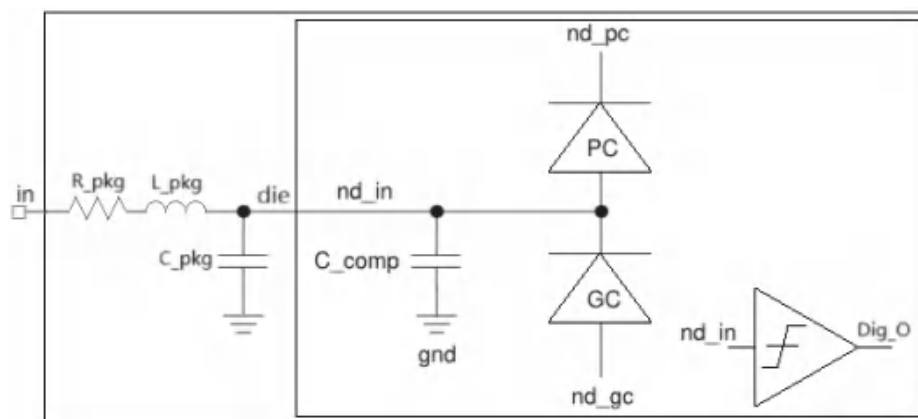


图48.Input 结构

数字逻辑输出管脚 Dig_O 的状态见下表

Polarity=Non-Inverting		
初始状态 t=0		$V(\text{Dig_O}) = V(\text{nd_in}) > (V_{\text{inh}} + V_{\text{inl}})/2$
t>0	$V(\text{nd_in}) > V_{\text{inh}}$	$V(\text{Dig_O}) = 1$
	$V(\text{nd_in}) < V_{\text{inl}}$	$V(\text{Dig_O}) = 0$
	$V_{\text{inl}} \leq V(\text{nd_in}) \leq V_{\text{inh}}$	$V(\text{Dig_O}) = 0.5$
Polarity=Inverting		
初始状态 t=0		$V(\text{Dig_O}) = V(\text{nd_in}) < (V_{\text{inh}} + V_{\text{inl}})/2$
t>0	$V(\text{nd_in}) > V_{\text{inh}}$	$V(\text{Dig_O}) = 0$
	$V(\text{nd_in}) < V_{\text{inl}}$	$V(\text{Dig_O}) = 1$
	$V_{\text{inl}} \leq V(\text{nd_in}) \leq V_{\text{inh}}$	$V(\text{Dig_O}) = 0.5$

表33. 数字管脚 Dig_O 输出

4.6.2.3 I/O Buffer

I/O 输入输出 buffer 的 symbol 如下:

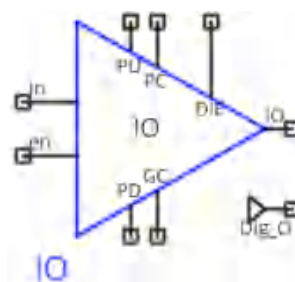


图49.I/O 图标

其内部结构原理如下

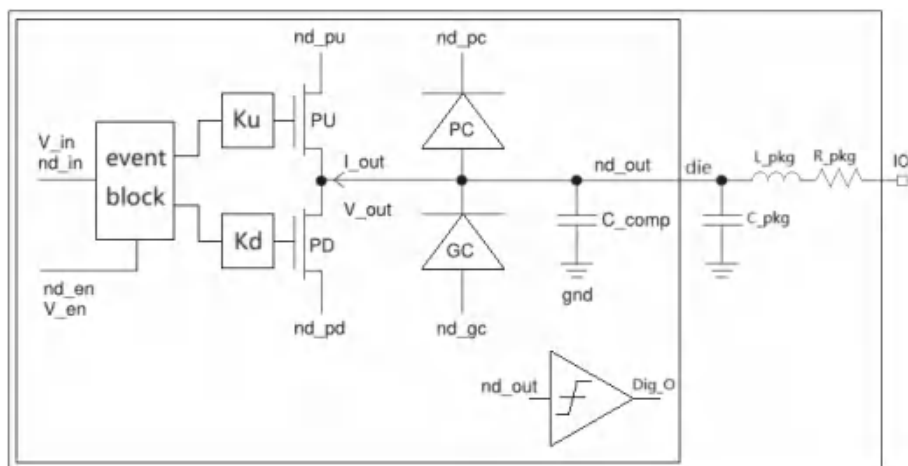


图50.I/O 结构

4.6.2.4 3-state Buffer

三态输出 buffer 的 symbol 如下:

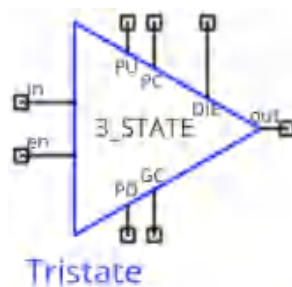


图51.3-sate 图标

其内部结构原理如下

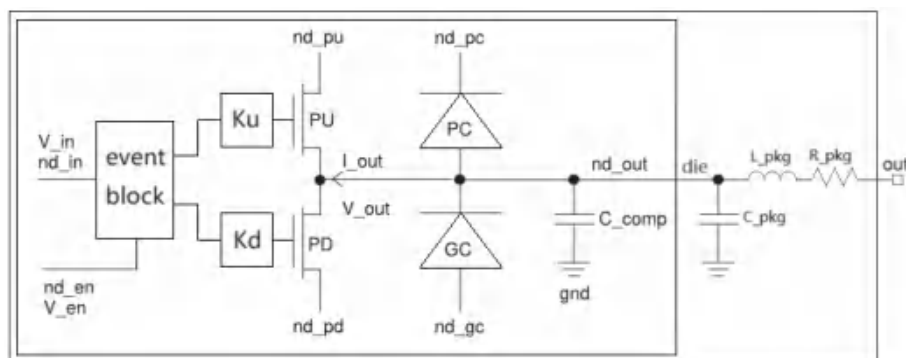


图52.3-state 结构

nd_in 管脚是数字激励输入，nd_en 是输出使能端，工作状态见下表。

	V_en		V_in	
	Active-High	Active-Low	Polarity=Non-Inverting	Polarity=Inverting
初始状态 t=0	>0.5 则使能	<0.5 则使能	>0.5 则输出高 <0.5 则输出低	>0.5 则输出低 <0.5 则输出高
	<0.5 不使能	>0.5 不使能	无效	无效
t>0	>0.8 则使能	<0.2 则使能	>0.8 则输出高 <0.2 则输出低	>0.8 则输出低 <0.2 则输出高

	<0.2 不使能	>0.8 不使能	无效	无效
--	----------	----------	----	----

表34.3-state 输出工作状态表

4.6.2.5 Open Drain/Sink, Open Source

它的 pin 与 output 完全相同，虽然他们缺省 pullup 或者 pulldown 支路，但是任然有对应的 pin。

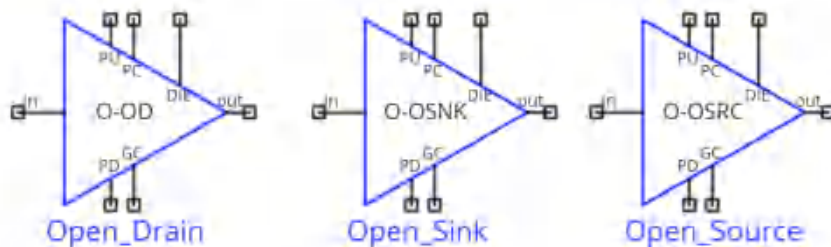


图53.Open Drain/Sink 与 Open Source 图标

Open Drain/Sink 内部结构原理如下，需要注意的是 Open drain/sink 没有 pullup 分支，因此不具备输出高电平的能力，一般需要配置上拉电阻。

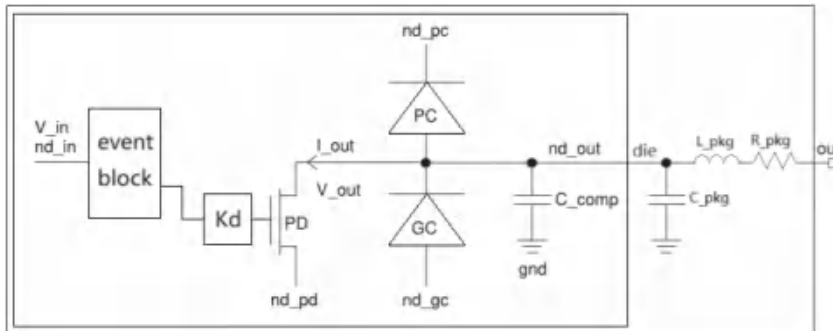


图54.Open Drain/sink 结构

Open Source 内部结构原理如下，需要注意的是 Open Source 没有 pulldown 分支，因此不具备输出低电平的能力，一般需要配置下拉电阻。

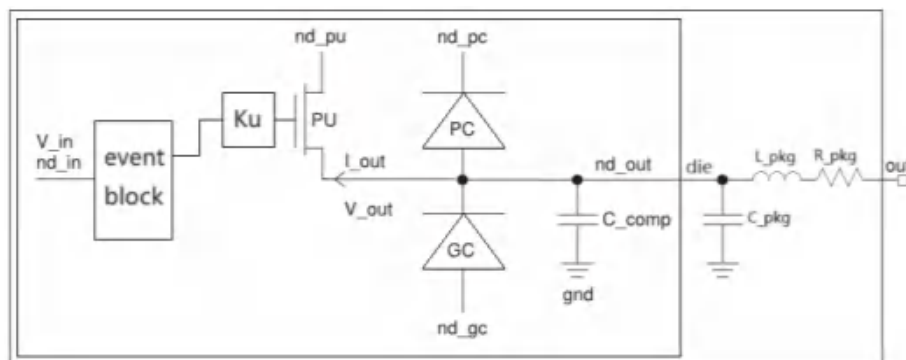


图55.Open source 结构

4.6.2.6 I/O Open Drain/Sink, I/O Open Source Buffer

它的 pin 与 I/O 完全相同，虽然他们缺省 pullup 或者 pulldown 支路，但是任然有对应的 pin。

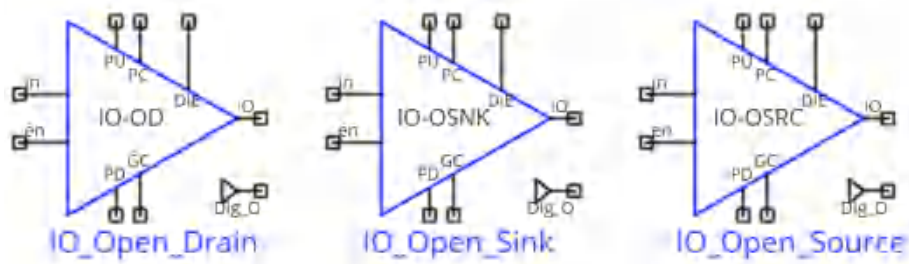


图56.I/O Open Drain/Sink 与 I/O Open Source 图标

I/O Open Drain/Sink 内部结构原理如下，需要注意的是 Open Drain/Sink 没有 pullup 分支，因此不具备输出高电平的能力，一般需要配置上拉电阻。

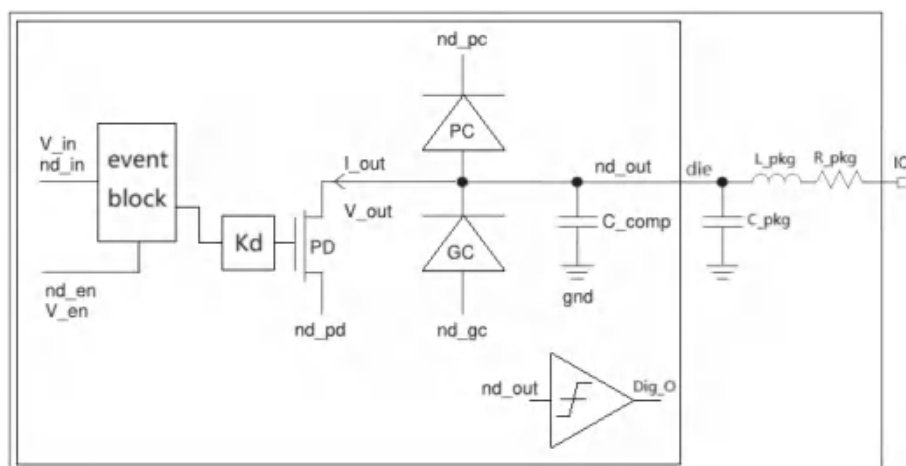


图57.I/O Open Drain/Sink 结构

I/O Open Source 内部结构原理如下，需要注意的是 OpenSource 没有 pulldown 分支，因此不具备输出高电平的能力，一般需要配置下拉电阻。

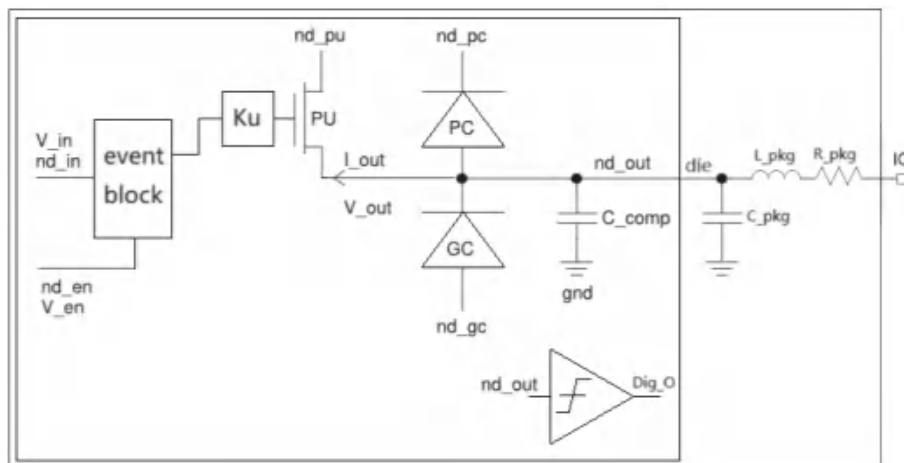


图58.I/O Open Source 结构

4.6.2.7 Output_ECL Buffer

ECL 输出 buffer 的 symbol 如下：

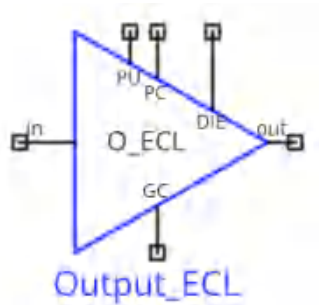


图59.Output ECL 图标

其内部结构原理如下，为了模拟 ECL 的电流输出特性，IBIS 规范将 PD 分支并联到 PU 分支上。

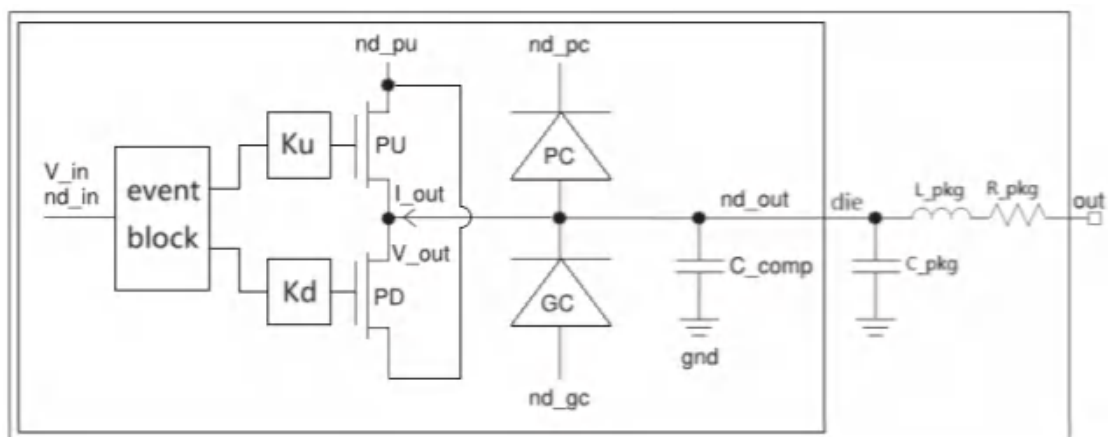


图60.Ouput ECL 结构

4.6.2.8 Input_ECL Buffer

ECL 输入 buffer 的 symbol 如下:

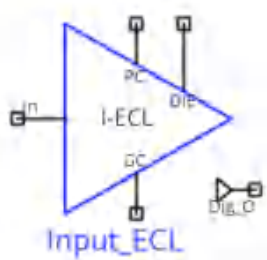


图61.Input ECL 图标

其内部结构原理如下

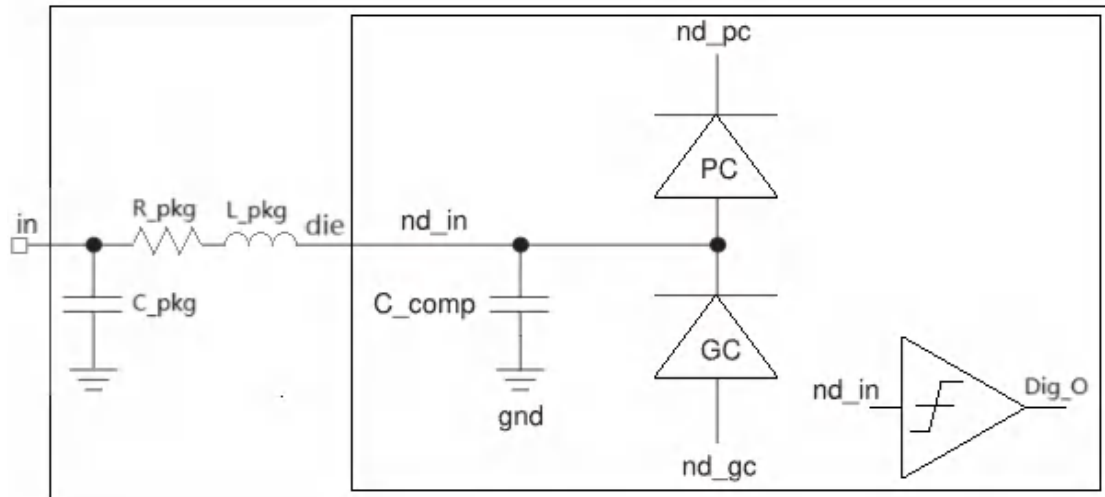


图62.Input ECL 结构

4.6.2.9 I/O_ECL Buffer

ECL 输入输出 buffer 的 symbol 如下:

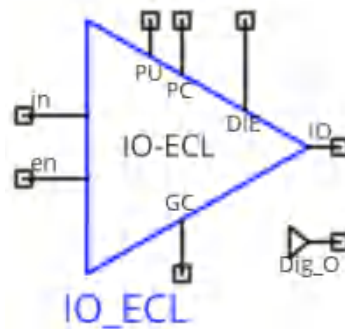


图63.I/O ECL 图标

其内部结构原理如下，为了模拟 ECL 的电流输出特性，IBIS 将 PD 分支连接到 PU 上。

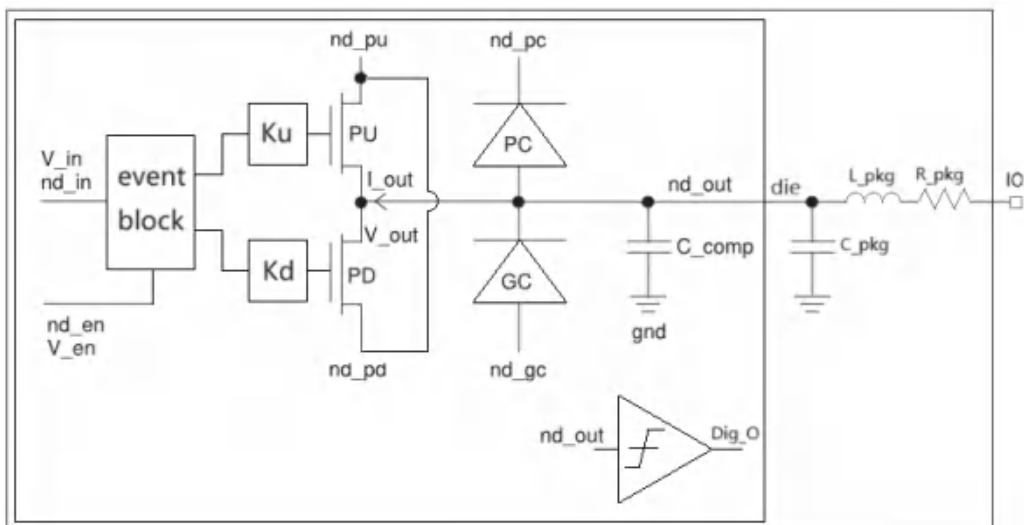


图64.I/O ECL 结构

4.6.2.10 3-state_ECL Buffer

ECL 三态输出 buffer 的 symbol 如下：

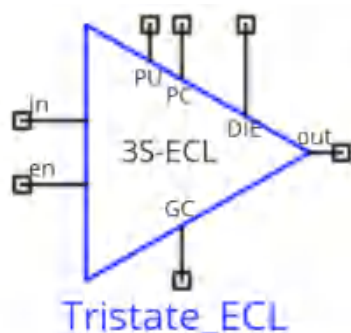


图65.3-satate ECL 图标

其内部结构原理如下，为了模拟 ECL 的电流输出特性，IBIS 将 PD 分支连接到 PU 上。

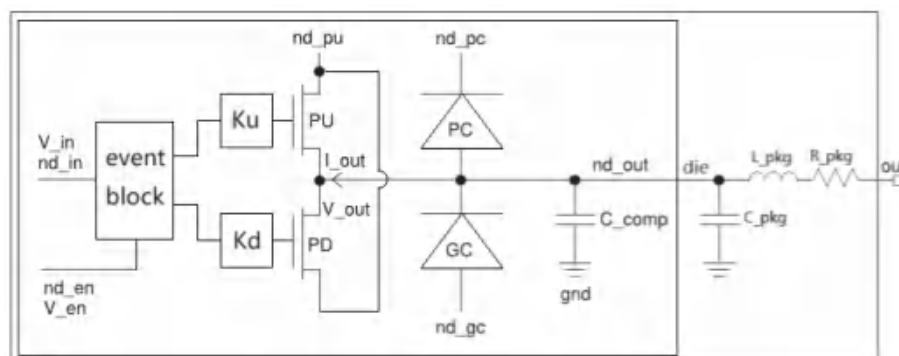


图1. 3-state ECL 结构

4.6.2.11 Terminator

端接 buffer 的 symbol 如下：

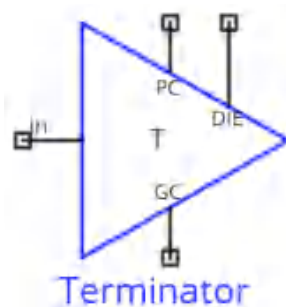


图66.Terminator 图标

它与 input 相比多了[Rgnd], [Rpower], [Rac], [Cac], 少了数字逻辑判决部分，其内部结构原理如下

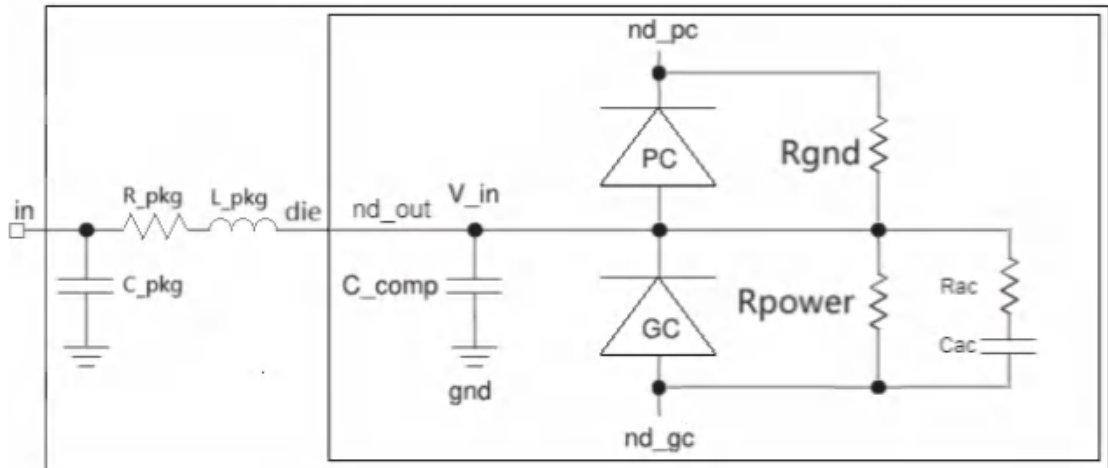


图67.Terminator 结构

4.6.2.12 Series

Series buffer 的 symbol 如下:



图68.Series 图标

它包含[R Series], [L Series], [Rl Series], [C Series], [Lc Series], [Rc Series], [Series Current],[Series MOSFET]。

其内部结构原理如下

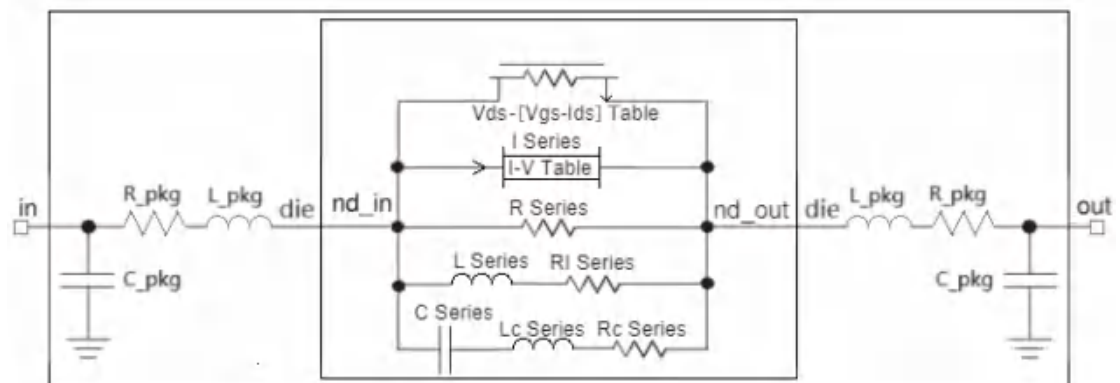


图69.Series 结构

4.6.2.13 Series_switch

串联开关 buffer 的 symbol 如下:

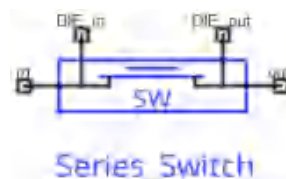


图70.Series switch 图标

节点总数 2。它包含[R Series], [L Series], [Rl Series], [C Series], [Lc Series], [Rc Series], [Series Current],[Series MOSFET]，与 Series Buffer 相比，它多了[Series MOSFET]的使能控制关键字 ss_state。

其内部结构原理如下：

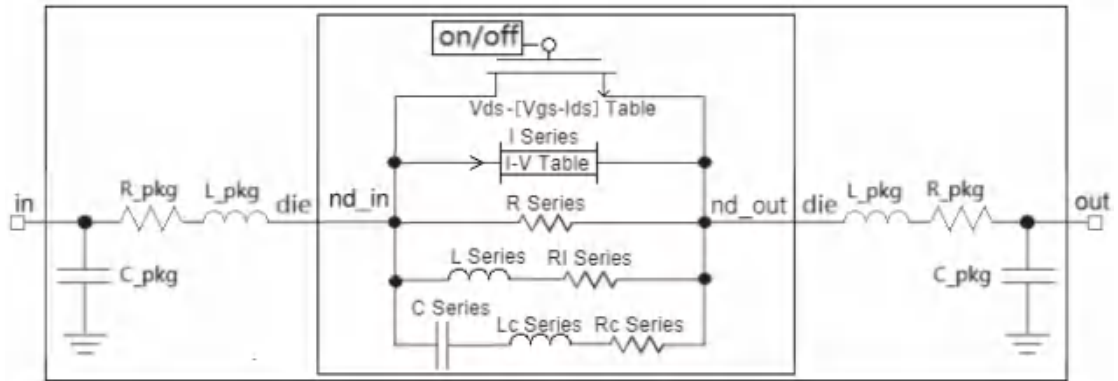


图71.Series Switch 结构

4.6.3 差分 buffer 模型

差分 buffer 的 symbol 如下：

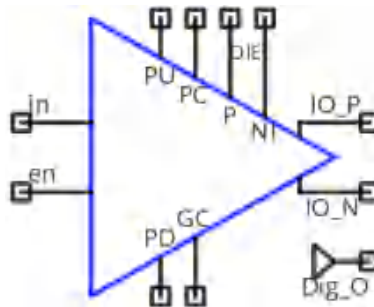


图72.差分 buffer 图标

差分 buffer 内部的 P 和 N 管脚分别对应了 1 个单端 buffer，可能会在 P 和 N 直接跨接一个 Series/Series switch 类 buffer，用户可以控制使能。对于存在 nd_in 的 buffer，N 会在内部将 P 的反向作为激励，对于有 Enable 使能的 buffer，P 和 N 同时控制。

其内部结构原理如下

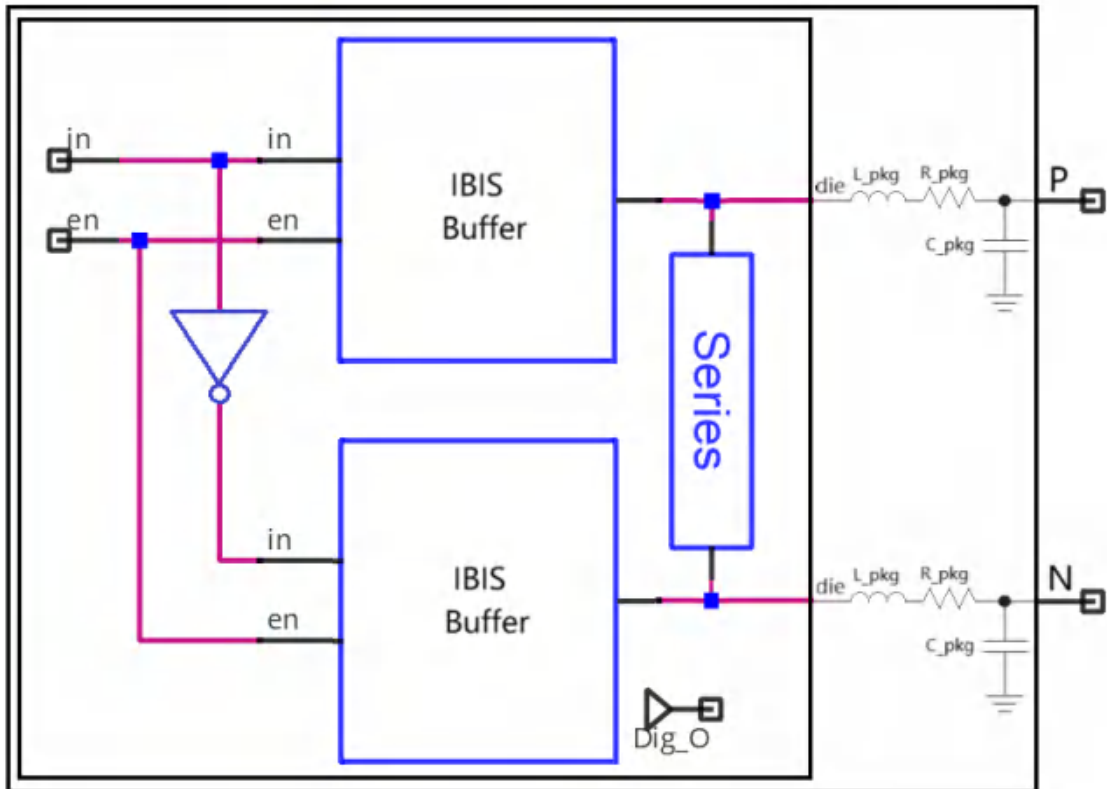


图73.差分 buffer 结构

nd_in 管脚是数字激励输入，在初始状态 $t=0$ 时与 0.5V 比较判决，在 $t>0$ 时与 0.2V/0.8V 迟滞比较判决，判决的逻辑值受到 Polarity 的影响，具体见下表。

Polarity=Non-Inverting		
初始状态 $t=0$		$V_{in} > 0.5?1:0$
$t>0$	$V_{in}>0.8$	state=1
	$V_{in}<0.2$	state=0
Polarity=Inverting		
初始状态 $t=0$		$V_{in} < 0.5?1:0$
$t>0$	$V_{in}>0.8$	state=0
	$V_{in}<0.2$	state=1

表35.Output 激励真值表

如果差分 buffer 有数字逻辑输出管脚 Dig_O，则其输出状态见下表

Polarity=Non-Inverting		
初始状态 $t=0$		$V(\text{Dig_O})=V(P,N)>0?1:0$
$t>0$	$V(P,N)>V_{diff}$	$V(\text{Dig_O})=1$
	$V(P,N)<-V_{diff}$	$V(\text{Dig_O})=0$
	$-V_{diff}\leq V(P,N)\leq V_{diff}$	$V(\text{Dig_O})=0.5$
Polarity=Inverting		
初始状态 $t=0$		$V(\text{Dig_O})=V(P,N)<0?1:0$
$t>0$	$V(P,N)>V_{diff}$	$V(\text{Dig_O})=0$
	$V(P,N)<-V_{diff}$	$V(\text{Dig_O})=1$
	$-V_{diff}<V(P,N)<V_{diff}$	$V(\text{Dig_O})=0.5$

表36. 数字管脚 Dig_O 输出

4.6.4 PKG 模型

IBIS PKG 可以提供用户将 ibs 文件中的指定 pin 的 pkg 参数作为独立 symbol 加到电路中的对应位置。

- 当只考虑 pin 自身的寄生参数而不考虑耦合时，采用[Package]或[Pin]所描述的 R、L、C 参数组成如下图所示的电路， R_{ii} 是自阻对应 R_{pkg} 值， L_{ii} 是自感对应 L_{pkg} ， C_{ii} 是自容对应 C_{pkg} 。
- 当需要考虑各个 pin 之间的耦合能量时，采用[Package Model]所描述的 [Inductance Matrix] [Capacitance Matrix] [Resistance Matrix]矩阵参数。

具体如下表所示

	类型	电阻	电感	电容
Package=1	[Package]	R_{pkg}	L_{pkg}	C_{pkg}
Package=2	[Pin]	R_{pin}	L_{pin}	C_{pin}
Package=3		Resistance matrix	Inductance matrix	Capacitance matrix
	Full_matrix	√	√	√
	Banded_matrix	√	√	√
	Sparse_matrix	√	√	√
Package=4	S 参数			
● 当选择 distributed 时将集总 RLC 转换成分布式的传输线 ● C 矩阵是 M 矩阵，L、R 是逆 M 矩阵 ● C_{ij} 定义为当导体“i”的电压是 1 V 且所有其他导体是 0V 时导体“j”上感应的电荷 ● L_{ij} 定义为当导体“i”的电流改变 1A/s 且所有其他导体是 0A/s 时导体“j”上感应的电压，$L_{ii} \geq \sum L_{ij} (i \neq j)$，$L_{ii} > 0$。 ● $R_{ii} > 0$，$R_{ii} > R_{ij}$				

图74.PKG 参数表

4.7 SYSTEMPI 电源完整性模型

4.7.1 VRM 模型

VRM 模型需要描述电源的动态响应特性，通过描述它的等效输出阻抗可间接描述，如下图所示是 1 阶 VRM 模型，有理想电压源、电阻、电感串联得到。



图75.VRM 符号

其内部等效电路如下

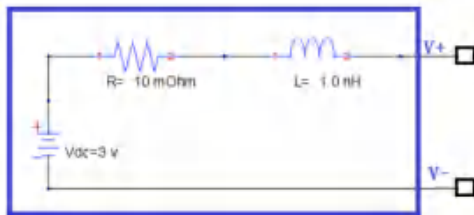


图76.VRM 内部等效电路

#	属性名	默认值	范围	单位	注释
1	Vdc	1.2		V	直流电压
2	Rs	1		mΩ	串联电阻
3	Ls	0.1		nH	串联电感

表37.VRM 参数表

4.7.2 CPM 模型



图77.CPM 符号

4.8 滤波器

支持多种经典的模拟滤波器类型，包括贝塞尔（Bessel）、巴特沃斯（Butterworth）、切比雪夫（Chebyshev）、高斯（Gaussian）和椭圆（Elliptic）滤波器是，每种类型有其独特的频率响应特性。以下是它们的主要特性对比：

类型	通带平坦性	过渡带陡峭性	相位线性
Bessel	好	差	好
Butterworth	好	一般	一般
Chebyshev	有纹波	较好	差
Gaussian	好	一般	较好
Elliptic	有纹波	好	差

图78.滤波器性能比对

可以看到 Bessel 与 Gaussian 的相位线性很好，群延迟是相位对频率的导数，因此群延迟的平坦性 $G_{dpass} = GD(0)/GD(F_p)$ 是衡量系统信号保真度的重要参数，其中 $GD(0)$ 是滤波器在零频率下的群延迟； $GD(F_p)$ 是滤波器在通带边缘频率下的群延迟。

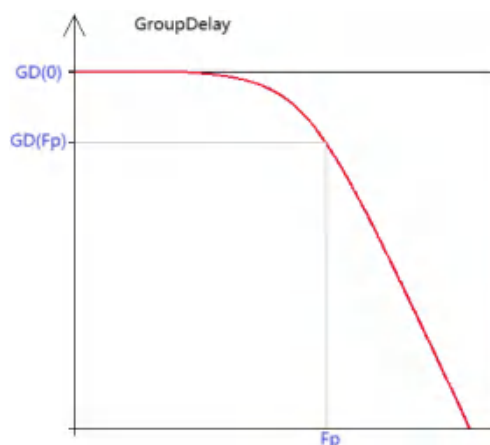


图79.群延时

滤波器 S 矩阵可以表示为如下，

$$\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = \begin{bmatrix} RL * (-1)^{\text{StopType}==\text{short}?1:0} & H_{tf} * \alpha \\ H_{tf} * \alpha & S_{11} * (-1)^{\text{OrderType}==\text{even}?1:0} \end{bmatrix}$$

其中：H_{tf} 是通过滤波综合算法得到的传递函数；RL 是反射能量；α 是通带衰减系数；StopType 是阻带开短路表现，当是 short 时回损相位反向；OrderType 是滤波器阶数，当是偶数时 S₂₂ 与 S₁₁ 相位相反。

在当前滤波器功能中是用 Fcenter，BWpass/BWstop 来表示带通/带阻滤波器的频率特性，可以通过如下关系式映射到滤波器的通带/阻带边缘频率，

$$Fp1 = Fcenter - BWpass/2$$

$$Fp2 = Fcenter + BWpass/2$$

$$Fs1 = \sqrt{Fcenter^2 - (BWpass/2)^2 + (BWstop/2)^2} - BWstop/2$$

$$Fs2 = \sqrt{Fcenter^2 - (BWpass/2)^2 + (BWstop/2)^2} + BWstop/2$$

其中：Fp1/Fp2 分别是通带的边缘频率；Fs1/Fs2 是阻带的边缘频率。

4.8.1 LPF 低通滤波器

低通滤波器（LPF）允许低频信号通过，同时衰减高频信号。其特性是信号在通带（即低于截至频率 F_{pass} 的频率范围内）几乎不受影响，而在阻带（即高于截至频率 F_{stop} 的频率范围内）被显著削弱甚至完全衰减。



图80.LPF 符号

#	属性名	默认值	范围	单位	滤波器类型					注释
					Bessel	Butterworth	Chebyshev	Gaussian	Elliptic	
1	Fpass	1	>0	GHz	√	√	√	√	√	通带边缘频率

2	Apass	3.0103	[0.01,3.0103]	dB	√	√	√ 1	√		通带衰减
3	Gdpass	0.9	(0,1)		√			√		通带群延迟
4	Fstop	1.2	≥Fpass	GHz		√	√		√	阻带边缘频率
5	Astop	20	>Apass	dB		√	√		√	阻带边缘衰减
6	StopType	open	Open Short							阻带输入阻抗类型: OPEN 或 SHORT
7	N	0	[0,15]							滤波器阶数, 如果 N = 0, 则内部自动计算阶数
8	IL	0		dB						通带插入损耗
9	Zin	50	>0	Ohm						输入端口参考阻抗
10	Zout	50	>0	Ohm						输出端口参考阻抗

表38.LPF 参数表

4.8.2 BPF 带通滤波器

带通滤波器（BPF）是一种允许特定频率范围内信号通过的滤波器，以 Fcenter 为中心的 BWpass 范围内的信号通过，BWstop 范围外的信号被显著削弱甚至完全衰减。

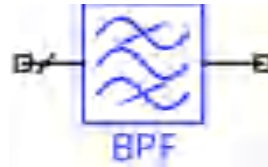


图81.BPF 符号

#	属性名	默认值	范围	单位	滤波器类型					注释
					Bessel	Butterworth	Chebyshev	Gaussian	Elliptic	
1	Fcenter	1.5	>0	GHz	√	√	√	√	√	中心频率
2	BWpass	1	< 2*Fcenter 且>0	GHz	√	√	√	√	√	通带边到边宽度
3	Apass	3.0103	[0.01,3.0103]	dB	√	√	√ 1	√		通带衰减
4	Gdpass	0.9	(0,1)		√			√		通带群延迟
5	BWstop	1.2	>BWpass	GHz		√	√		√	阻带边到边频率
6	Astop	20	>Apass	dB		√	√		√	阻带边缘衰减
7	StopType	open	Open Short							阻带输入阻抗类型: OPEN 或 SHORT
8	N	0	[0,15]							滤波器阶数, 如果 N = 0, 则内部

					自动计算阶数 通带插入损耗 输入端口参考阻抗 输出端口参考阻抗
9	IL	0		dB	
10	Zin	50	>0	Ohm	
11	Zout	50	>0	Ohm	

表39.BPF 参数表

4.8.3 HPF 高通滤波器

高通滤波器（HPF）则与低通滤波器相反，它允许高频信号通过，而衰减低频信号。在截至频率 F_{stop} 以下的低频信号被削弱，而在截至频率 F_{pass} 以上，信号几乎无衰减。

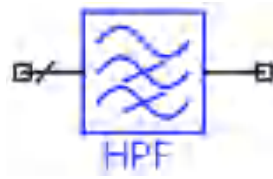


图82.HPF 符号

#	属性名	默认值	范围	单位	滤波器类型					注释
					Bessel	Butterworth	Chebyshev	Gaussian	Elliptic	
1	Fpass	1	$\geq F_{stop}$		√	√	√	√	√	
2										
3	Apass	3.0103	[0.01,3.0103]	dB	√	√	√ 1	√		通带衰减
4	Gdpass	0.9	(0,1)		√			√		通带群延迟
5	Fstop	0.8	>0			√	√		√	
6	Astop	20	>Apass	dB		√	√		√	阻带边缘衰减
7	StopType	open	Open Short							阻带输入阻抗类型: OPEN 或 SHORT
8	N	0	[0,15]							滤波器阶数，如果 N = 0，则内部自动计算阶数
9	IL	0		dB						通带插入损耗
10	Zin	50	>0	Ohm						输入端口参考阻抗
11	Zout	50	>0	Ohm						输出端口参考阻抗

表40.HPF 参数表

4.8.4 BSF 带通滤波器

带阻滤波器（BSF）是一种不允许特定频率范围内信号通过的滤波器，以 Fcenter 为中心，BWstop 范围内的信号被显著削弱甚至完全衰减，BWpass 范围外的信号通过。



图83.BSF 符号

#	属性名	默认值	范围	单位	滤波器类型					注释
					Bessel	Butterworth	Chebyshev	Gaussian	Elliptic	
1	Fcenter	1.5	>0	GHz	√	√	√	√	√	中心频率
2	BWpass	1.2	< 2*Fcenter 且>0	GHz	√	√	√	√	√	通带边到边宽度
3	Apass	3.0103	[0.01,3.0103]	dB	√	√	√ 1	√	√	通带衰减
4	Gdpass	0.9	(0,1)	无	√			√		通带群延迟
5	BWstop	1	<BWpass	GHz		√	√		√	阻带边到边频率
6	Astop	20	>Apass	dB		√	√		√	阻带边缘衰减
7	StopType	open	Open Short							阻带输入阻抗 类型: OPEN 或 SHORT
8	N	0	[0,15]							滤波器阶数， 如果 N = 0， 则内部自动计算阶数
9	IL	0		dB						通带插入损耗
10	Zin	50	>0	Ohm						输入端口参考阻抗
11	Zout	50	>0	Ohm						输出端口参考阻抗

表41.BSF 参数表

4.9 受控源模型

	VCVS	CCCS	VCCS	CCVS
基本	√	√	√	√
Laplace 拉普拉斯多项式	√		√	
PZR 极零点形式	√		√	
PR 极余形式	√		√	
支持仿真类型	Transient, S-parameters, Channel-eye			

表42.受控源支持列表

4.9.1 E-[VCVS]

E-[VCVS]是电压控制电压源，包含输入电阻与输出串联电阻。其 symbol 如下图所示



图84.VCVS 符号

#	属性名	默认值	范围	单位	注释
1	gain	1			增益
2	delay	0	>0	s	延时
3	Rin	1e20		Ω	输入电阻
4	Rout	0		Ω	输出电阻

表43.VCVS 属性表

4.9.2 F-[CCCS]

F-[CCCS]是电流控制电流源，包含输入电阻与输出并联电阻。其 symbol 如下图所示

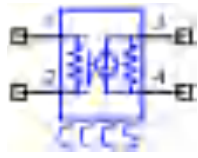


图85.CCCS 符号

#	属性名	默认值	范围	单位	注释
1	gain	1			增益
2	delay	0	>0	s	延时
3	Rin	0		Ω	输入电阻

4	Rout	1e20		Ω	输出电阻
---	------	------	--	----------	------

表44.CCCS 属性表

4.9.3 G-[VCCS]

G-[VCCS]是电压控制电流源，包含输入电阻与输出并联电阻。其 symbol 如下图所示

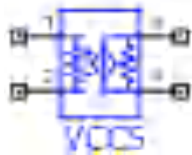


图86.VCCS 符号

#	属性名	默认值	范围	单位	注释
1	gain	1		S	增益
2	delay	0	>0	s	延时
3	Rin	1e20		Ω	输入电阻
4	Rout	1e20		Ω	输出电阻

表45.VCCS 属性表

4.9.4 H-[CCVS]

H-[CCVS]是电流控制电压源，包含输入电阻与出串联电阻。其 symbol 如下图所示

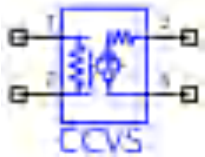


图87.CCVS 符号

#	属性名	默认值	范围	单位	注释
1	gain	1		Ω	增益
2	delay	0	>0	s	延时
3	Rin	0		Ω	输入电阻
4	Rout	0		Ω	输出电阻

表46.CCVS 属性表

4.9.5 E/G 传递函数表达式

4.9.5.1 Laplace—拉普拉斯多项式形式

传递函数分子分母表示成多项式的形式

$$H(s) = \frac{k_0 + k_1s + \dots + k_ns^n}{d_0 + d_1s + \dots + d_ms^m}$$

4.9.5.2 Pole-Zero—极零点形式

传递函数分子分母表示成零极点形式

$$H(s) = \frac{a \cdot (s + \alpha_{z1} - j2\pi f_{z1}) \cdot (s + \alpha_{z1} + j2\pi f_{z1}) \dots (s + \alpha_{zn} - j2\pi f_{zn})(s + \alpha_{zn} + j2\pi f_{zn})}{b \cdot (s + \alpha_{p1} - j2\pi f_{p1}) \cdot (s + \alpha_{p1} + j2\pi f_{p1}) \dots (s + \alpha_{pm} - j2\pi f_{pm})(s + \alpha_{pm} + j2\pi f_{pm})}$$

4.9.5.1 Foster Pole-Residue—极余形式

传递函数分子分母表示成极余形式，极余数都互为共轭

$$H(s) = k_0 + k_1 s + \sum_i \left(\frac{\text{Re}\{A_i\} + j \cdot \text{Im}\{A_i\}}{s - (\text{Re}\{p_i\} + j \cdot \text{Im}\{p_i\})} + \frac{\text{Re}\{A_i\} - j \cdot \text{Im}\{A_i\}}{s - (\text{Re}\{p_i\} - j \cdot \text{Im}\{p_i\})} \right)$$

5 结果显示后处理

5.1 函数与运算表格

分类	Function Name	Description
Math	pow(x,y)或者 x**y	x 的 y 次方
	sqr(x)	x 开平方
	sqr()	Square of the selected simulation quantity.
	root(x,y)	nth root function.
	abs(x)	绝对值
	exp(x)	e 的 x 次方
	fmod(x,y)	x%y 取余
	log(x)或 ln	e 为底 x 对数
	log10(x)	10 为底 x 的对数
	sin(x)	正弦
	cos(x)	余弦
	tan(x)	正切
	asin(x)	反正弦
	acos(x)	反余弦
	atan(x)	反正切(数学 -pi/2~pi/2)
	atan2(x,y)	反正切(工程 -pi~pi)
	sinh(x)	双曲正弦
	cosh(x)	双曲余弦
	tanh	Hyperbolic tangent.
	asinh	Inverse hyperbolic sine.
	acosh	Inverse hyperbolic arc cosine.
	atanh	Inverse hyperbolic tan.
	floor(x)	向下取整
	ceil(x)	向上取整

	round(x)	四舍五入
	sgn	Sign extraction.
复数	real(y)或 re(y)	y 的实部
	imag(y)或 im(y)	y 的虚部
	cmplx(re, im)	A complex number, where re is the real part and im is the imaginary part.
	conjg	Conjugate of the complex number.
	mag(y)	y 的模值
	phase(y)	y 的弧度相位
	ang_rad(y)/ang(y)	y 的弧度相位
	ang_deg(y)/arg(y)	y 的角度相位
	unwarp_arg_rad(y)	y 的非折叠弧度相位
	unwarp_arg_deg(y)	y 的非折叠角度相位
	cang_rad(y)	同 unwarp_ang_rad(y)
	cang_deg(y)	同 unwarp_ang_deg(y)
求 dB 运算	dB(x)	$20 \cdot \log_{10}(x)$ to base 10.
	dB20	$20 \cdot \log(x)$ to base 10.
	dBm(x)	$10 \cdot \log_{10}(x) + 30$.
	dBW(x)	$10 \cdot \log_{10}(x)$.
	dB10	$10 \cdot \log(x)$ to base 10.
	dBc	Decibels relative to the carrier. It is the power ratio of the signal to a carrier signal. Gives the relative signal strength.
测量与最值运算	max	Returns maximum value of the simulation quantity.
	max2	Maximum value of the two simulation quantities. For example, max2(a,b) will plot maximum of a and b for a particular instance.
	min	Returns the minimum value of the simulation quantity.
	min2	Minimum value of the two simulation quantities. For example, min2(a,b) will plot minimum of a and b for a particular instance.

	XAtYMax	Threshold crossing time: report first time (x value) at which an output quantity crosses YMax.
	XAtYMin	Threshold crossing time: report first time (x value) at which an output quantity crosses a user definable threshold.
	XAtYVal	Returns the X value at the first occurrence of Y value.
	YAtXMax	Returns the X value at maximum value of Y.
	YAtXMin	Returns the Y value at minimum value of X.
	YAtXVal	Returns the Y value at the first occurrence of X value.
	bandwidth	Returns the 3dB bandwidth of the selected simulation quantity. For bandwidth, the calculation is based on 3dB below the maximum peak.
	avg	Returns the average of the values of the selected quantity. $avg = (\text{Area between the curve and the X-axis}) / (\text{X length of the curve})$
	avgabs	Returns the mean of the absolute value of the selected quantity.
	mean	Returns the average in the set of quantities selected. $mean = \text{sum}(\text{all y-value}) / (\text{number of y-values})$
高级	cum_integ	The cumulative integral function returns a set of values that have the same length as the original set of points (the first element will always be zero). Element I of the set returned by cum_integ is the integral of elements 1 through I of the original data set.
	cum_sum	The cumulative sum function returns a data set that has the same length as the original set of points. Element I of the set returned by cum_sum is the sum of

		elements 1 through I of the original data set.
	deriv	Derivative of first parameter with respect to second parameter.
	integ	Integral of the selected quantity. Uses trapezoidal area.
杂	min(y,y,...)	最小
	max(y,y,...)	最大
	plot_vs(vectorY,vectorX)	plot_vs(yValues,xValues)输出整图
	~x	取反
基本运算	+	加
	-	减
	*	乘
	**	幂
	/	除
逻辑运算	!x	取非
		或
	&&	与
	==	等
	!=	不等
	<	小于
	<=	小于等于
	>	大于
	>=	大于等于
条件运算	if(,)	条件分支 if(cond_exp,true_exp, false_exp).
Arrays/Matrix	inverse(m_Matrix)	矩阵求逆 inverse({1,2,3},{2,6,4},{3,4,5})
	transpose(m_Matrix)	矩阵转置
	matrixdot(matrix1,matrix2)	两个矩阵的乘积
	expandMatrix({M0,M1})	expandMatrix 函数
	permute({4.S(2,1),5.S(2,1),6.S(2,1),7.S(2,1)})	将 {4.S(2,1),5.S(2,1),6.S(2,1),7.S(2,1)} 生成矩阵
	X[12,:]	获取数组第 12 个及以后所有数据
	{M1,M2,M3,M4,...}	多个数组组成矩阵
	X(row,column)/X(row)	从数组/矩阵中抽取对应位置的元

		素
	X[pos]	从数组/矩阵中抽取对应位置元素
	S(2,1)[3::6,:]	等价于 {4.S(2,1),5.S(2,1),6.S(2,1),7.S(2,1)}

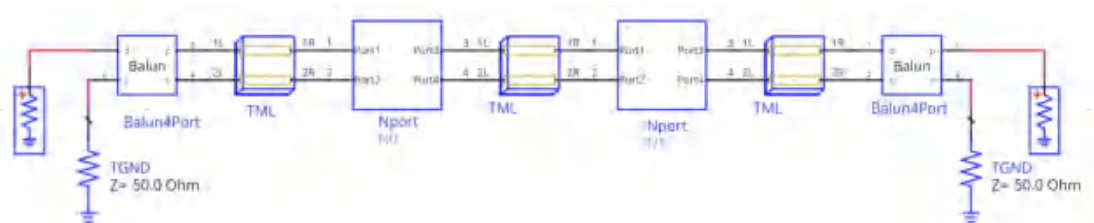
6 应用场景与案例

6.1 S 参数级联仿真

将 S 参数、传输线、BALUN 级联

6.1.1 搭建 S 参数、传输线、Balun 级联链路

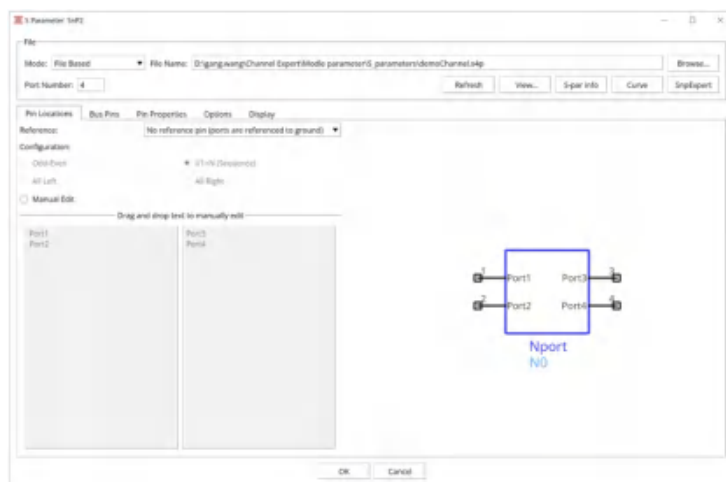
分别从 Components 下 TML 元件窗口拖入 TML、Lumped Components 原件窗口拖入 Balun4，选择 Data Source 下 Nport 与 TML 下 TML 器件作为通道，建立连接关系，并在器件两端增加 GND 与 Port：



6.1.2 电路参数设置

双击 Nport 器件，Nport 器件界面包括 Model、File Name、S 参数引脚信息、Port Number 数。通过 Browse 导入 S 参数，S 参数的引脚基本信息可在 Pin Location、Bus Pins、Pin Properties 处查看；Refresh 是重新加载 S 参数，View 是查看 s*p 文件信息，S-par info 是查看 S 参数的具体信息，Curve 和 SnpExpert 是查看 S 参数的联通关系，选在在 ChannelExpert 软件或在 Snp Expert 软件中展示

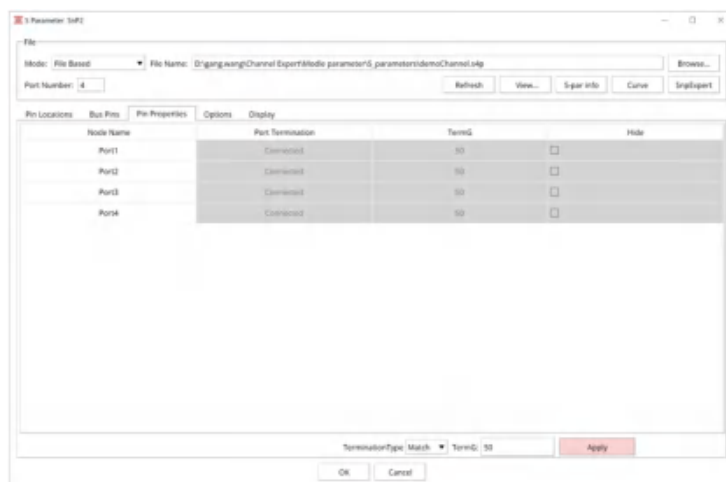
引脚位置信息可通过 Pin Properties 处查看，Configuration 处展示的是 S 参数引脚的显示方式，包括四种；可以通过选择 Manual Edit 自行对引脚进行编辑，在 Drag and drop text to manually edit 处通过直接对 pin 脚拖拽，完成位置变换：



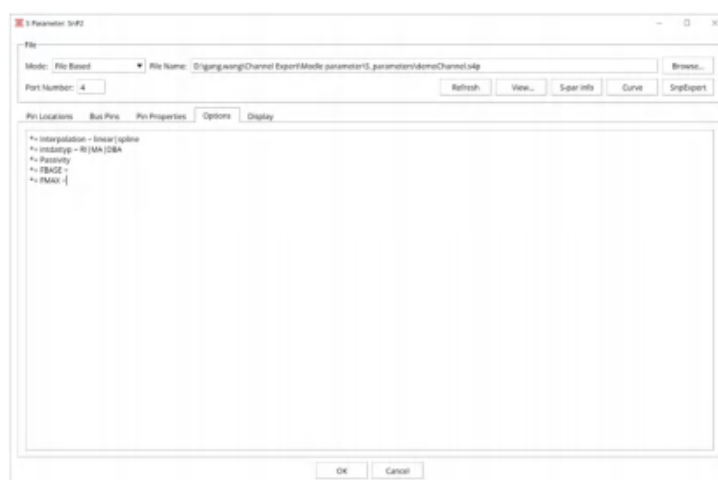
Bus Pins 界面可以查看 pin index 和 pin Name，也可以对 pin 脚进行组 Bus:



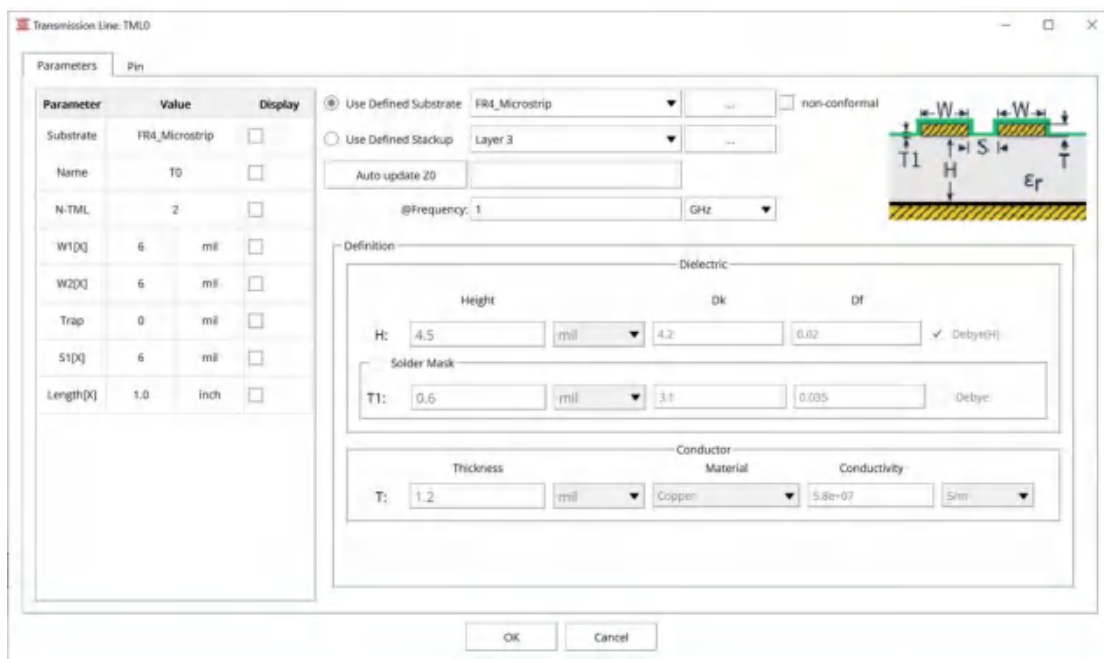
Pin Properties 可以对各个 pin 的引脚状态进行判断，如已经是连接状态，连接的器件是什么，也可通过勾选 hide 界面来选择是否在原理图中展示:



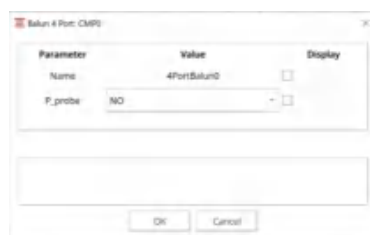
Options 处可以在仿真时对仿真器增加限定条件，如增加 FBASE、FMAX 等：



TML 器件界面包括 Substrate 材料信息和叠层信息；Substrate 选项下，可以有两种方式选择材料信息，一是通过系统提供的材料库，二是通过导入材料文件来导入材料库；Impedance 阻抗值可以人为编辑，频率通过软件计算得出；页面左上角显示的是 TML 的 Parameter，每根线需要放在的导体层，每根线的宽度、线与线之间的间隙、线的总长度。这里线与线之间的间隙规则是上一根线的右侧到下一根线的左侧距离器件基本信息；Pin Locations 可以选择引脚的位置显示方式：

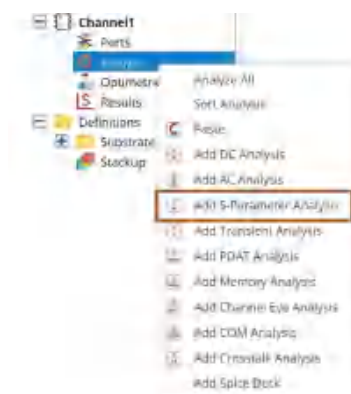


Balun4 器件界面包括 Name 以及是否打开 P_probe:

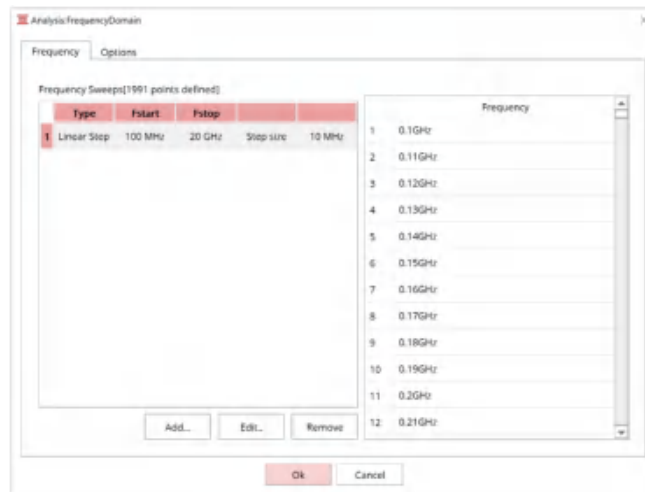


6.1.3 参数仿真设置

右击树节点 Analysis 增加 S-Parameter Analysis



弹出仿真设置界面，点击 Add 或 Edit，设置扫描的 Type（如 Linear Step 线性仿真）、设置 S 参数仿真的频率范围(Fstart、Fstop、Step size)，扫描的频率数会在界面右侧显示：

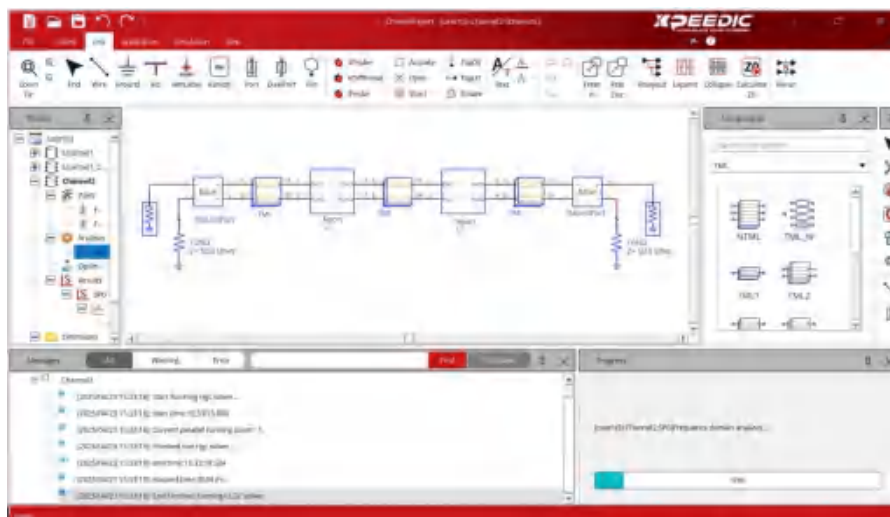


点击 Options 选项，可以选择插值方式、插值类型，具体项目如下所示：

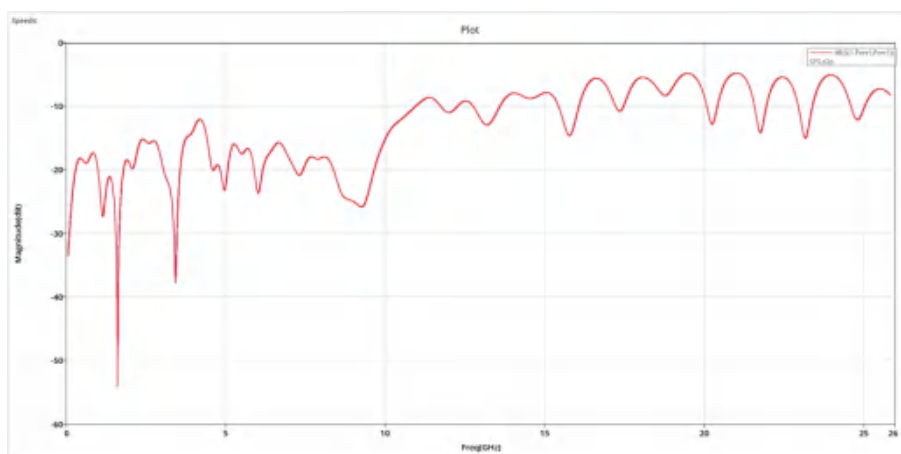


6.1.4 S 参数仿真结果

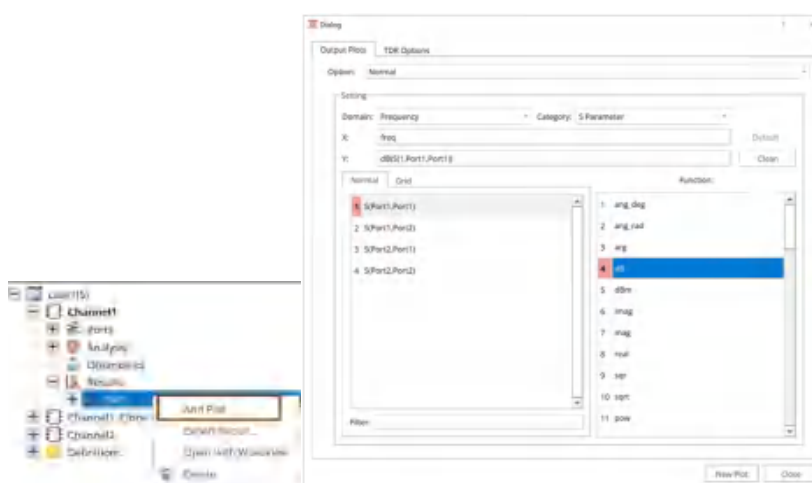
设置完成后，在 SP0 树节点上右键，选择 analysis 开始仿真，下方 Messages 会显示当前 solver 的 log 信息，进度条展示当前的运行状态：

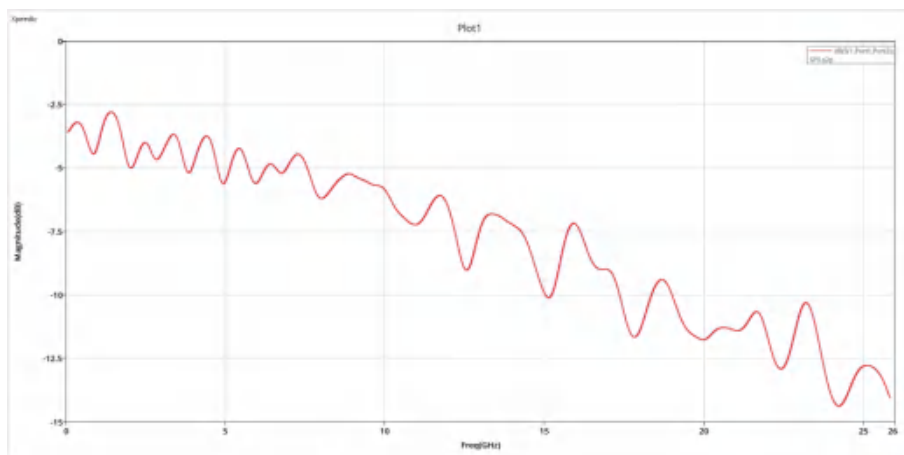


仿真结束，树节点自动挂出 S11 曲线图，利用内置的 S 参数结果，可以观测到链路的 S 参数仿真结果，仿真结束后的 S 参数仿真曲线如下：



结果默认展示的是 S11，可以树节点下右键 SP0，选择 Add Plot，在弹出的界面，选择需要查看的 S 参数（如选择 S12），点击 New Plot，得到结果：



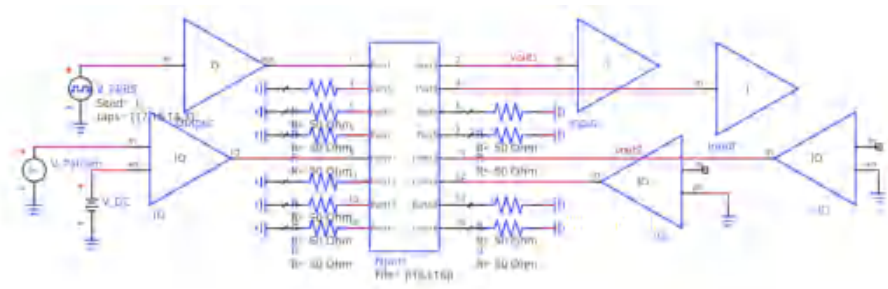


需要拿出生成的结果，右键 results，选择 Open Results Folder。通过对 S 参数曲线分析，可以判别链路通道信息情况。

6.2 IBIS 通用电路拓扑 TRAN 仿真

6.2.1 搭建 Tran 仿真链路

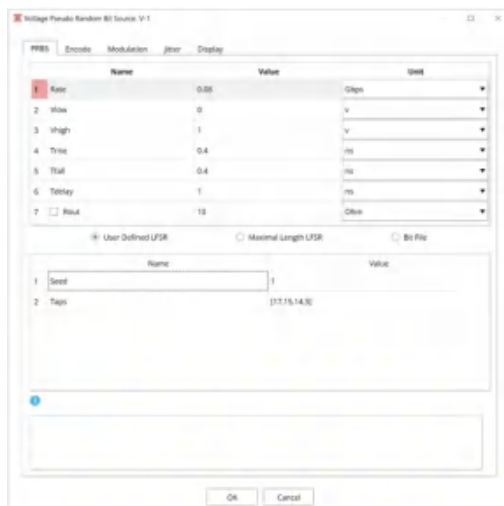
分别从元件窗口拖入 Independent Source 源、Buf Pin、N_port，建立连接关系，仿真电路如下图所示。从输入端不同的 Source 源输出信号，经过 O/I/O IBIS 器件、S 参数通道，由接收端 I/O IBIS 器件接受信号，在需要观察的节点加入 Probe 节点用来观测电压曲线。



6.2.2 电路参数内部设置

Source 源输入端包括 PRBS 电压基本设置、Encoder、Modelation 与 Jitter 四部分：

PRBS 源包括速率、上升沿下降沿时间、延时等基本设置，勾选 Rout 可以阻抗值进行设置；进行瞬态分析时，可在界面下方设置码型：



Encode, 编码设置, 目前支持 DW8b10b、DW64b66b、DW128b130b、DW128b132b。

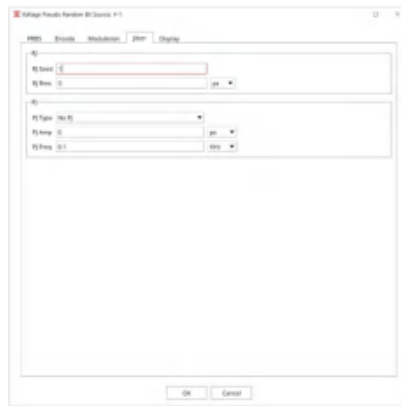


Modulation, 选择调制模式, 默认 NRZ, 支持 PAM3、PAM4、PAM6、PAM8、PAM16:



Jitter 增加抖动设置。分随机抖动 RJ 和无周期抖动 PJ

RJ 下可以设置 Seed 和 Sigma 值; PJ 下可以设置 Type、Amp 和 Freq。

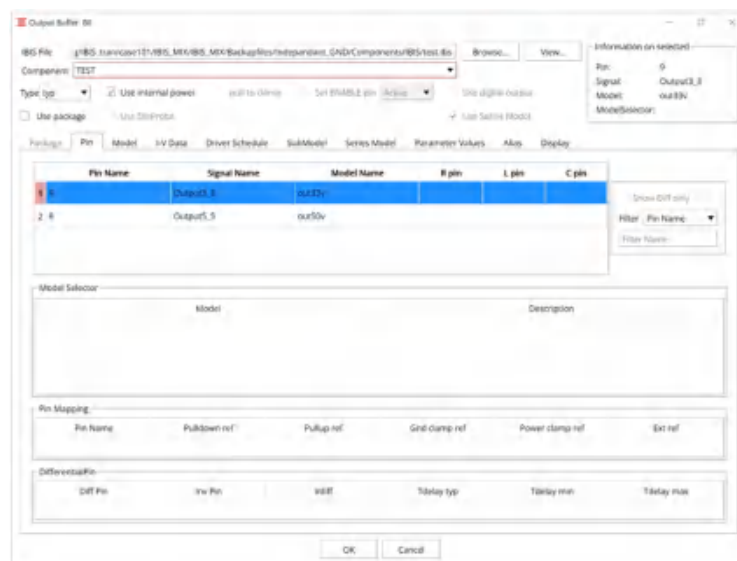


电阻设置:



IBIS 文件由 Browse 按钮导入，ibs 文件的具体内容可通过点击 View 按钮查看；导入后的 IBIS 文件路径和文件名在 IBIS file 处显示、IBIS 器件的 Components 和 Type 信息可选项下查看；勾选 Use internal power 表示使用内部电源，Set ENABLE pin 是设置使能端状态，有两种选项，一是设为 Active，另一种是 Active；勾选 Use package 是选择使用内部封装；勾选 Use DieProbe 是使用封装的内部引脚；勾选 Use digital output 是使用数字信号引脚输出；勾选 Use Series Model 是差分信号对中的耦合部分的模型形态；

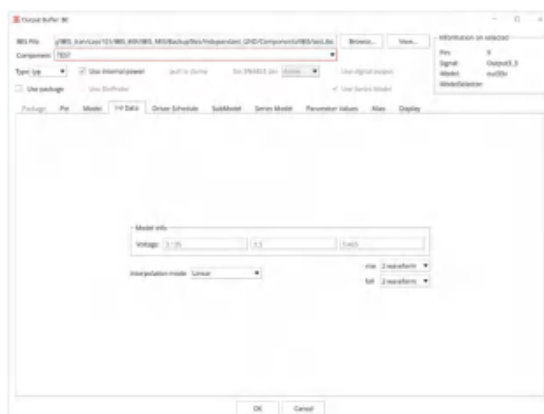
IBIS Pin 的 Name、Signal name、Model Name 可在 Pin 脚信息页下查看，在界面上可以手动选择器件中的 pin 脚和需要用到的 Model；界面右侧可以通过勾选 Show Diff only 按钮选择只显示差分信号，可以选择 Filter 显示方式（Pin Name、Signal Name、Model Name 三种显示方式中的一个），也可以通过 Filter Name 搜索 Pin name；



Model 信息页包括 Model Information 和 Die Capacitances，包括 ibis 的 mode type、Polarity、Enable、电压信息等：

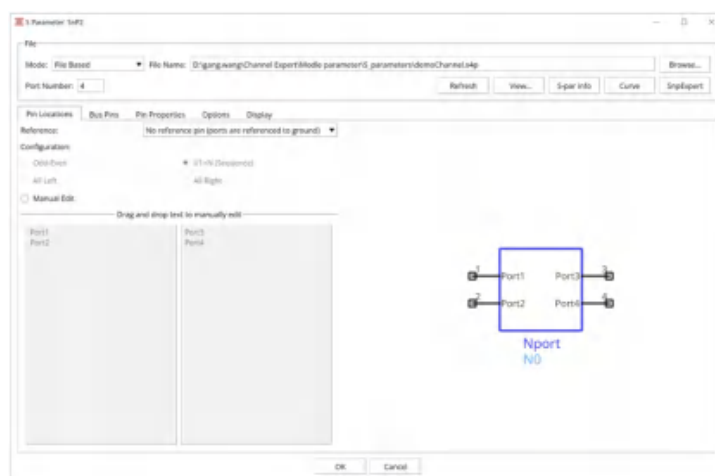


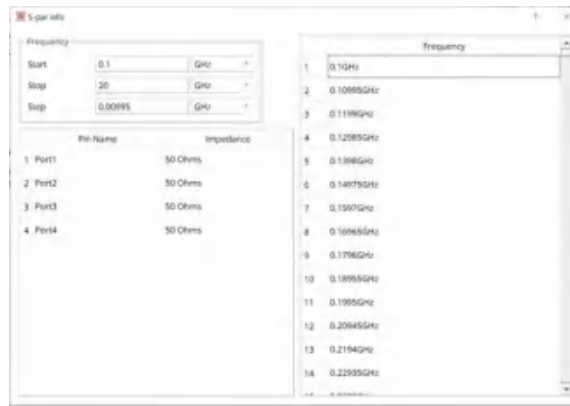
I-V Data 信息页包括 Model info（Voltage 值），插值方式等：



双击 Nport 器件，Nport 器件界面包括 Model 、File Name、S 参数引脚信息、Port Number 数。通过 Browse 导入 S 参数，S 参数的引脚基本信息可在 Pin Location、Bus Pins、Pin Properties 处查看；Refresh 是重新加载 S 参数，View 是查看 s*p 文件信息，S-par info 是查看 S 参数的具体信息，Curve 和 SnpExpert 是查看 S 参数的联通关系，选在在 ChannelExpert 软件或在 Snp Expert 软件中展示

引脚位置信息可通过 Pin Properties 处查看，Configuration 处展示的是 S 参数引脚的显示方式，包括四种；可以通过选择 Manual Edit 自行对引脚进行编辑，在 Drag and drop text to manually edit 处通过直接对 pin 脚拖拽，完成位置变换：

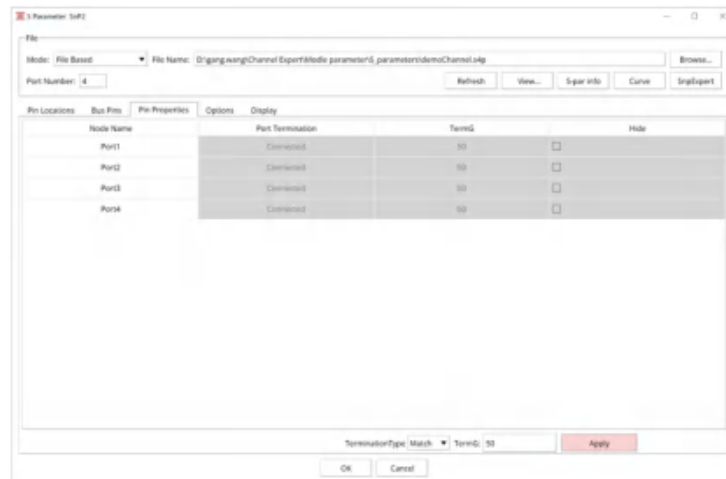




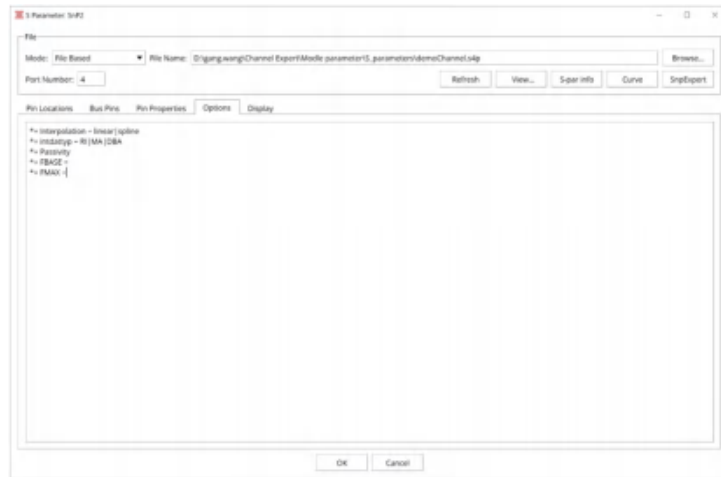
Bus Pins 界面可以查看 pin index 和 pin Name，也可以对 pin 脚进行组 Bus:



Pin Properties 可以对各个 pin 的引脚状态进行判断，如已经是连接状态，连接的器件是什么，也可通过勾选 hide 界面来选择是否在原理图中展示:

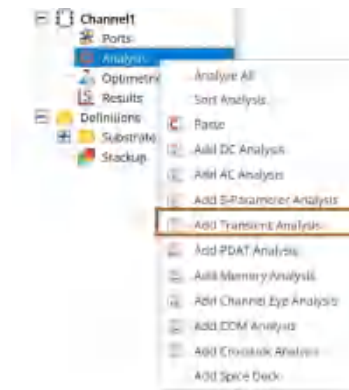


Options 处可以在仿真时对仿真器增加限定条件，如增加 FBASE、FMAX 等:

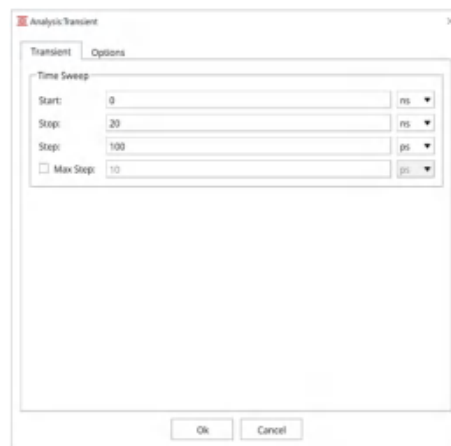


6.2.3 Tran 仿真设置

右击树节点 Analysis 增加 Transient Analysis

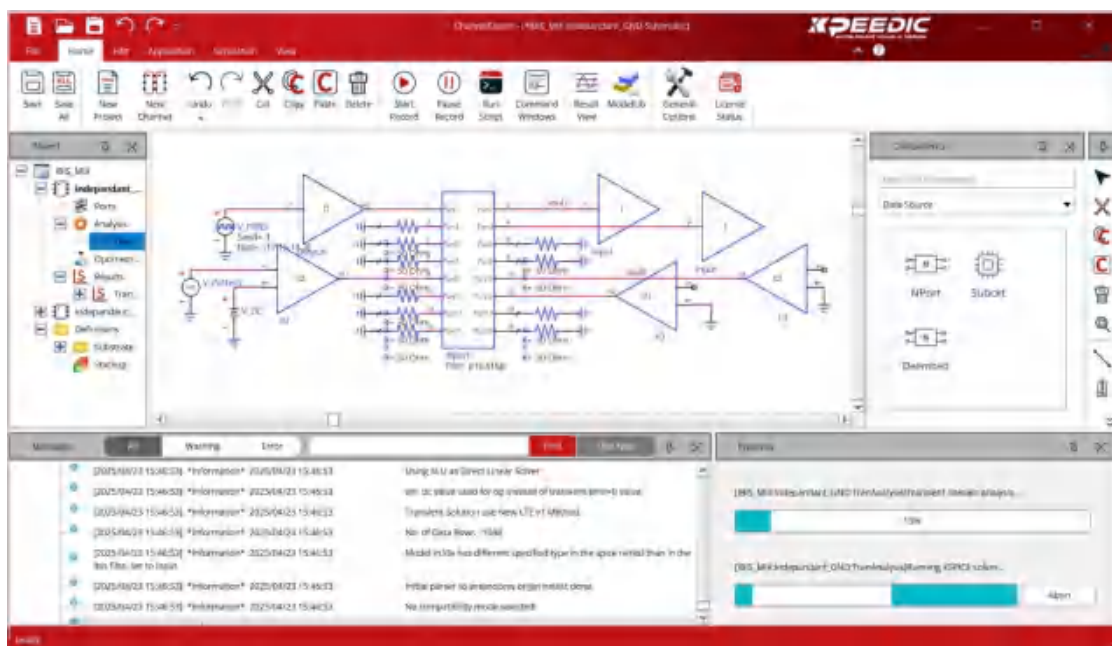


设置 Tran 仿真的频率扫描范围（Start、Stop、Step），如果步长不满足，也可勾选 Max Step:

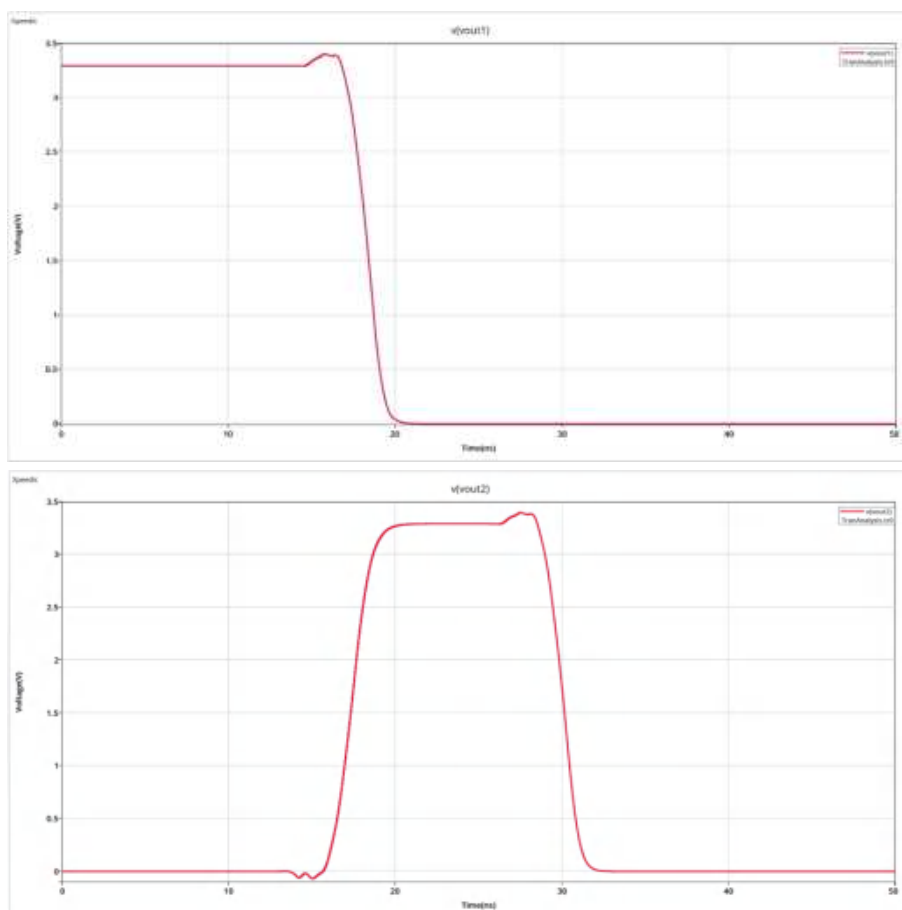


6.2.4 Tran 仿真结果

设置完成后，在 TranAnalysis 树节点上右键，选择 Analysis，开始仿真，下方 Messages 会显示当前 solver 的 log 信息，进度条展示当前的运行状态:



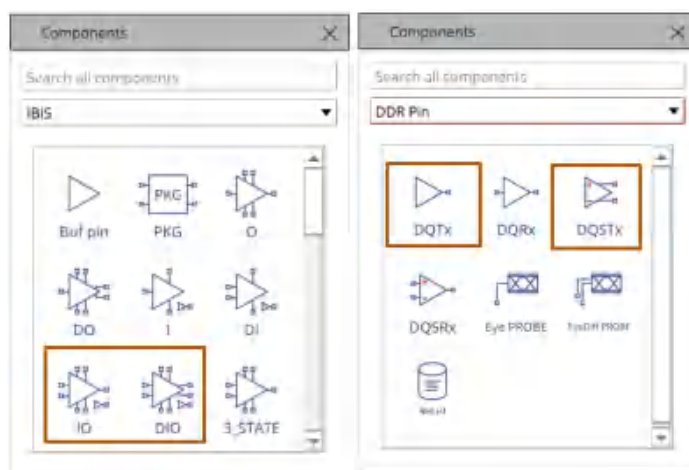
仿真结束，各个节点（Vout1、Vout2）的输出电压-时间（V-T）曲线会直接挂载出来，结果如下图所示：



6.3 DDR 并行链路仿真

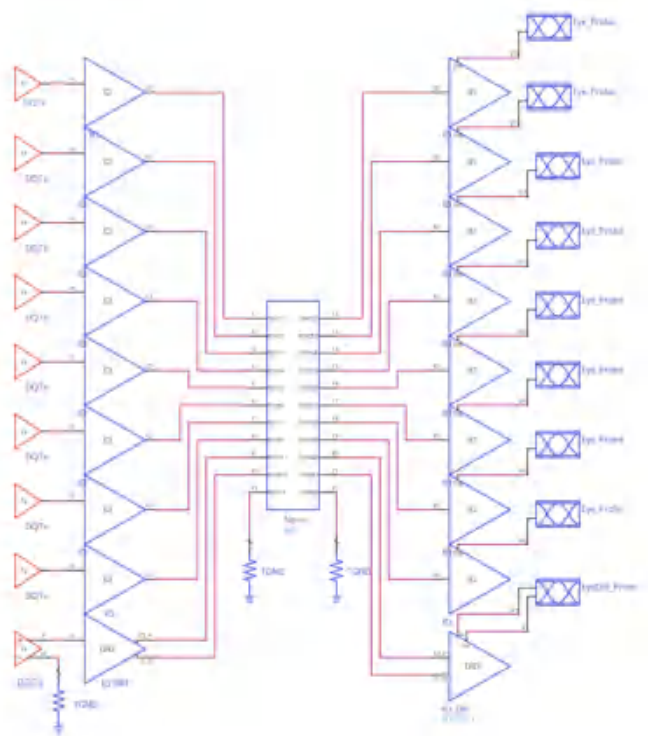
6.3.1 单 DDR buffer 传统模式

Components 的 IBIS 模型下包含 ibs 参数的加载器，可以用作 S 参数通道的传输情况；TML 模型下包括不同引脚的 TML 器件；Lumped 器件下包括 Balun4 模型。各器件如下所示：



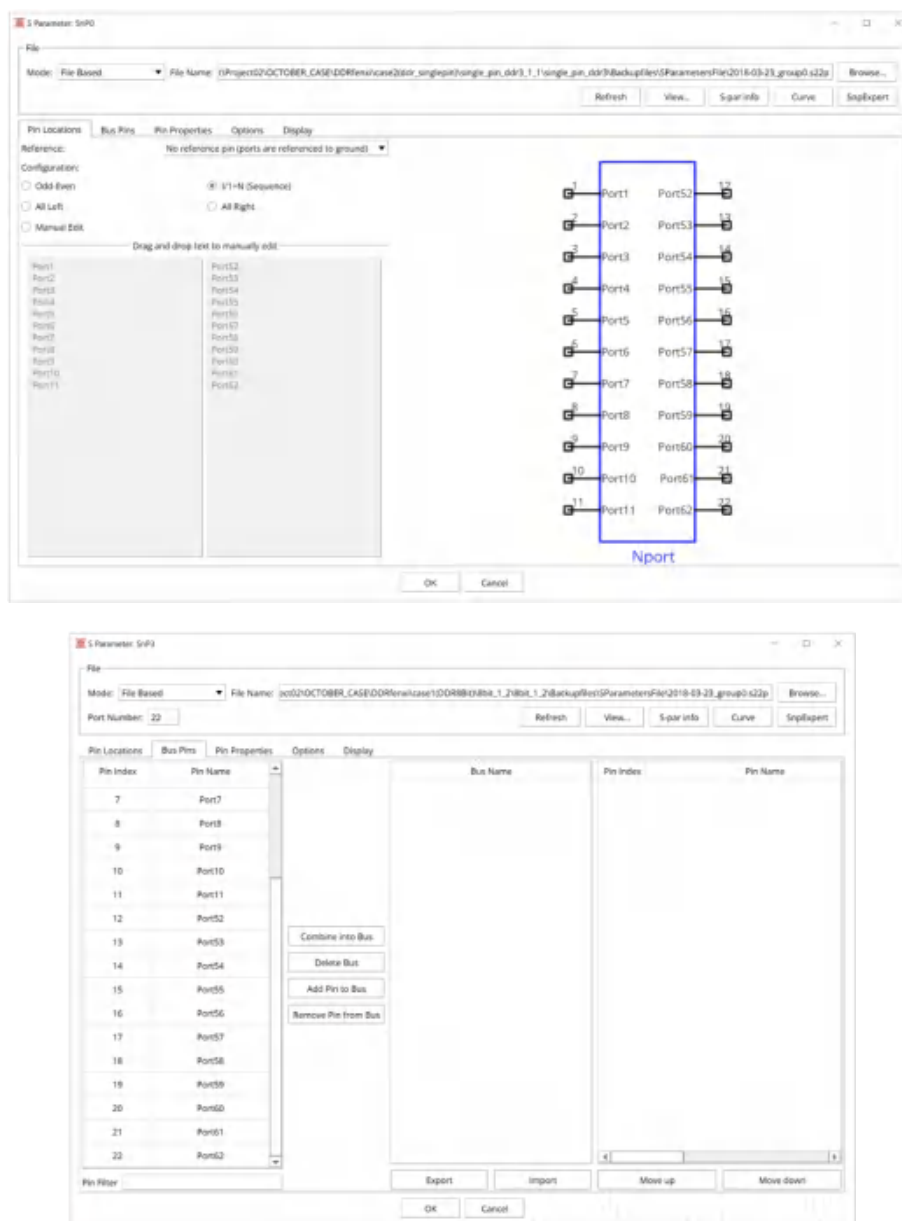
6.3.1.1 搭建 DDR 单 Pin 仿真链路

分别从元件窗口拖入 DQTx、DQRx、IO/DIO 与通道，建立连接关系，DDR 单 Pin 仿真电路如下图所示。从输入端 DQTx 输出信号，经过 IO 器件 S 参数通道，由接收端接受信号：



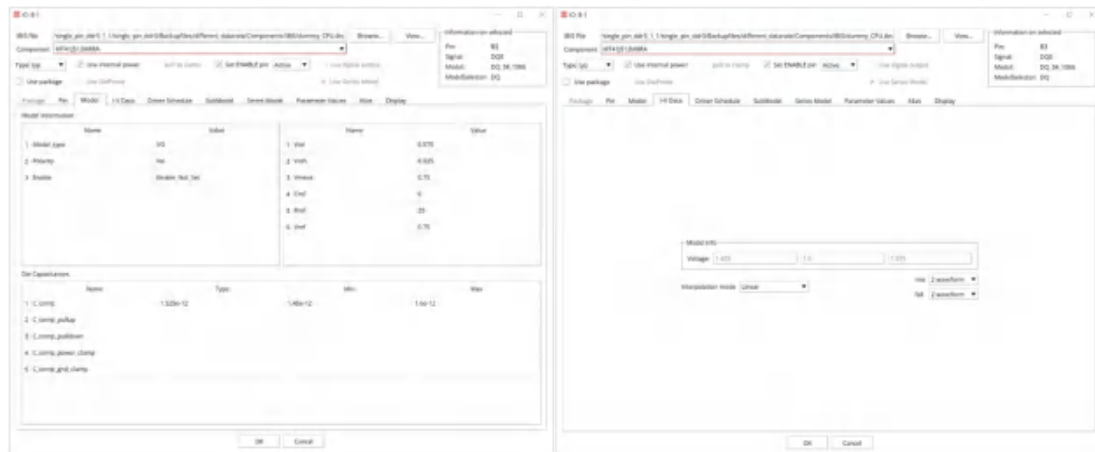
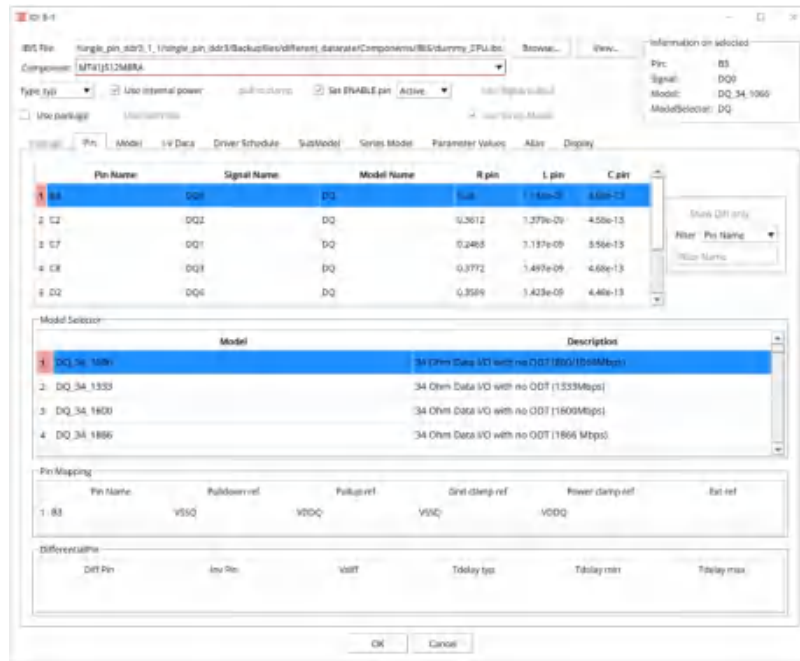
6.3.1.2 电路内部设置

Nport 器件界面包括 File Name、Model、S 参数引脚信息、Port Number 数。通过 Browse 导入 S 参数，基本信息可在 Pin Location 处查看，引脚连接关系可任意通过 Pin Properties 处查看，Options 处可以在仿真时对仿真器增加限定条件：

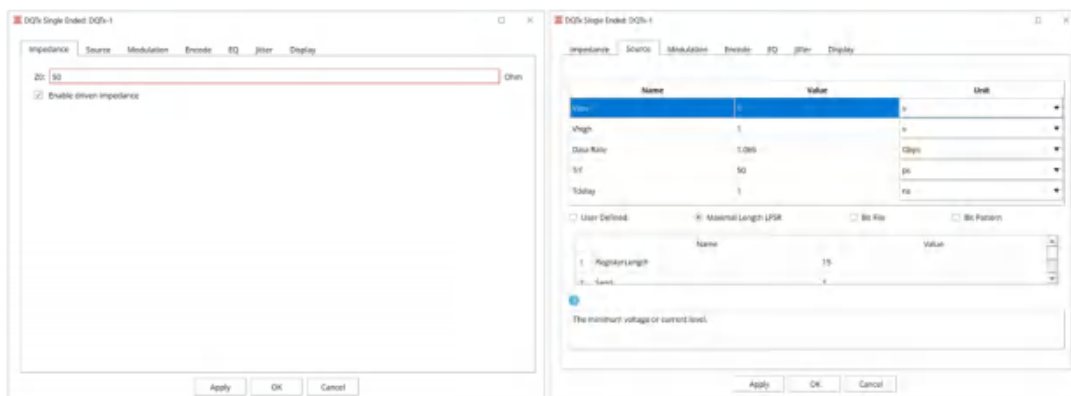


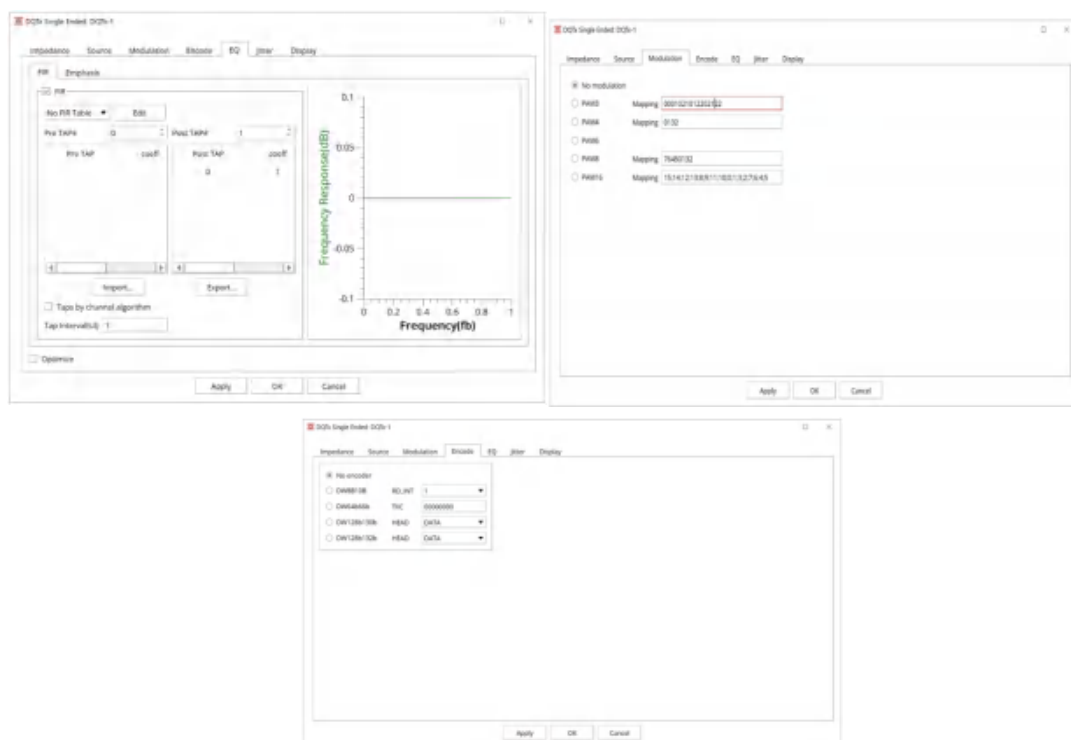
IO 器件界面包括 IBIS File、Component、Type 信息；选择是否使用 ibs 的 internal power、set ENABLE pin 选择将 ENABLE pin 的状态选为 Active 或者使用 Disable、是否使用 Package；

在 Pin 信息栏可以查看导入的 ibs 所包含的 pin 脚信息和 Model 等，Model 信息栏可以查看 Model Type 等信息；I-V Data 栏可以查看 Model info 信息：

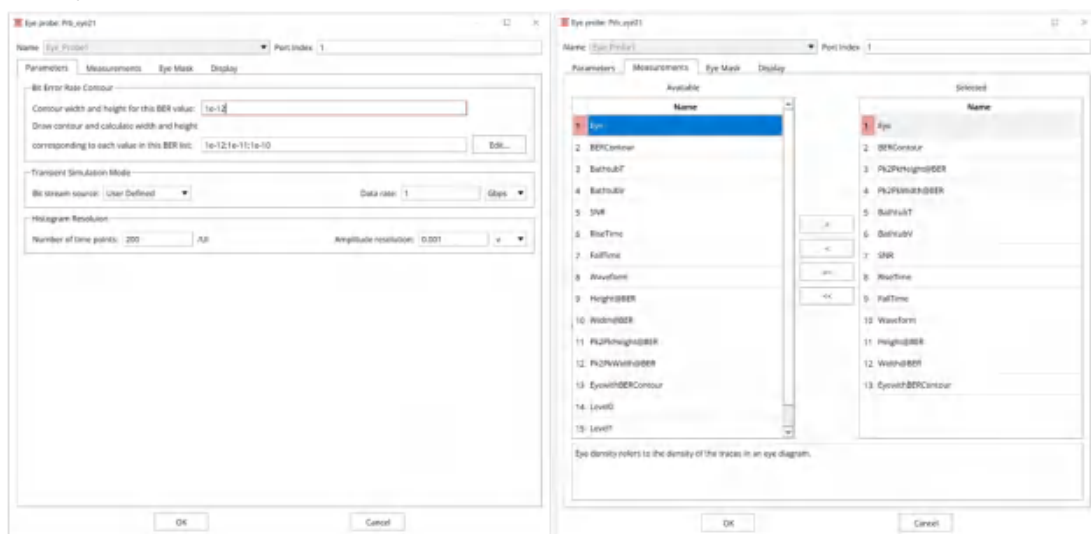


DQTX 包括 Impedance，可以对阻抗值进行设置；Source，可以设置电压幅值、速率、上升沿下降沿时间等；Modulation，PAM3、PAM4 等；Encode；EQ 均衡、Jitter 抖动等：





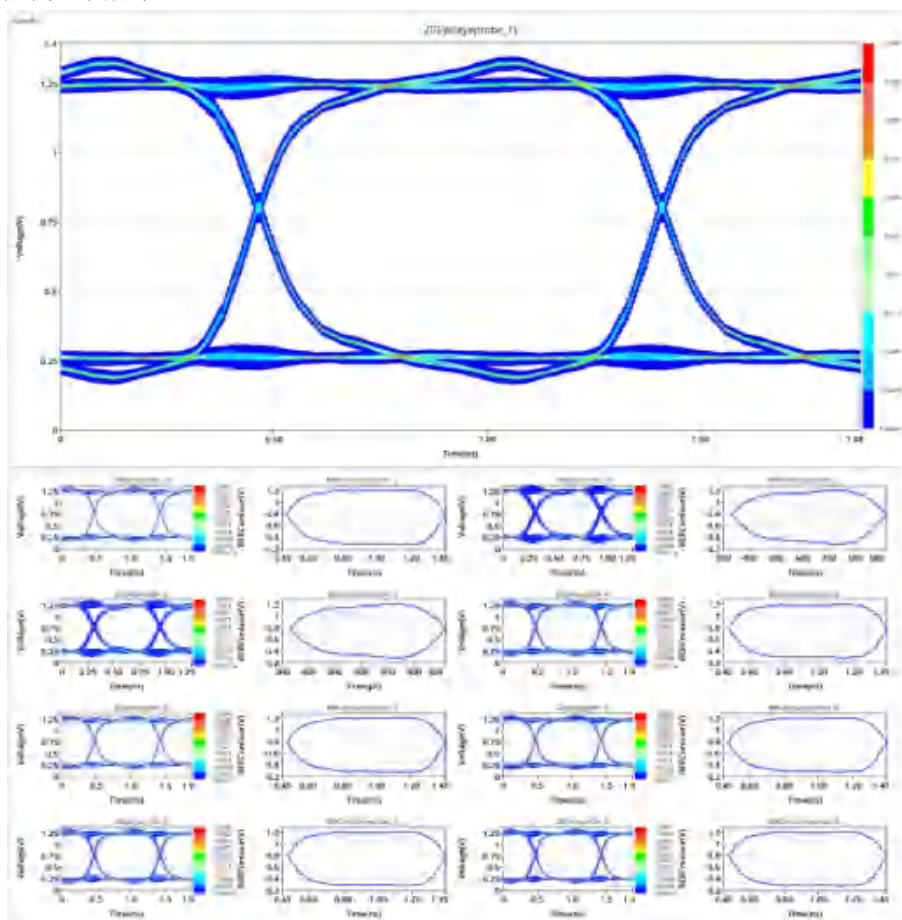
设置 Eye_probe 属性，包括 Bit Error Rate Contour、Transient Simulation Mode、Histogram Resolution 等；选择 Eye_probe 要输出的内容，目前 Eye_probe 可以输出的内容包含 eye 与 BERContour 等；选择是否在产生的结果中添加眼图测量的 Mask：





6.3.1.3 仿真结果

眼图结果如下所示：



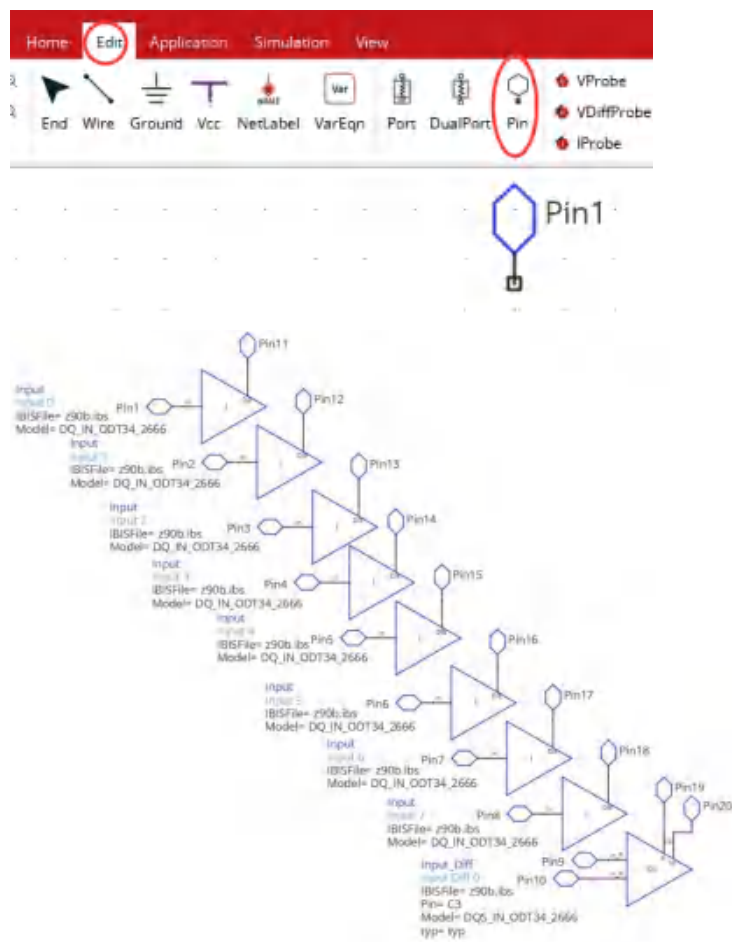
通过对眼图以及 BER 曲线分析，可以判别 DDR 电路的稳定性和可靠性。

6.3.2 DDR 单 Pin 层次化

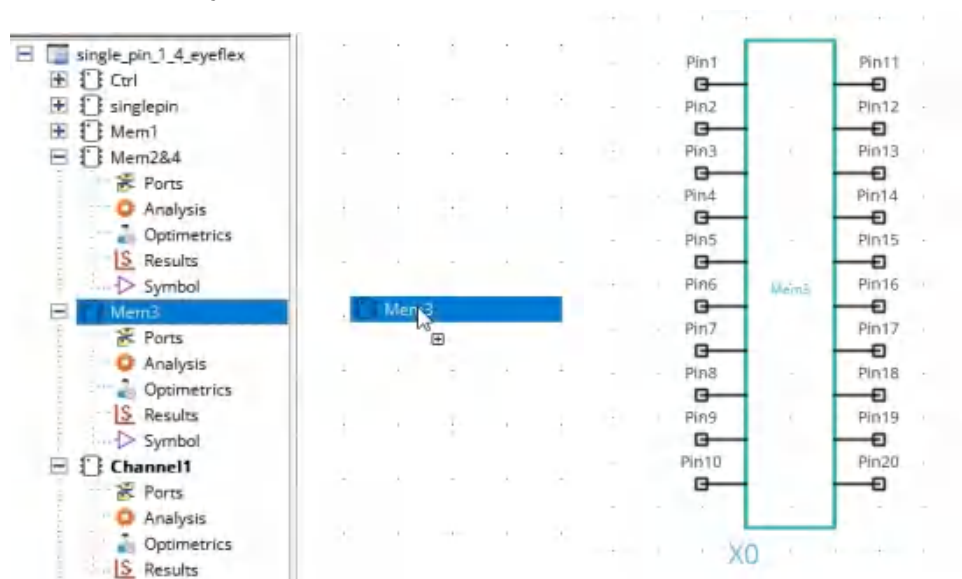
当器件较多时，为了原理图整体更清晰、整洁，我们支持了原理图的层次化。下面以常用的 DDR 层次化链路为例，同时结 EyeFlex 示波器整体输出结果。

6.3.2.1 层次化搭建原理图

搭建好底层基础链路，作为子模块，给需要露出来的端口增加 Pin。



子电路模块，也就是一个独立 channel，拖入到需要放置的其他 channel 里，会在新的原理图里生成 Symbol。



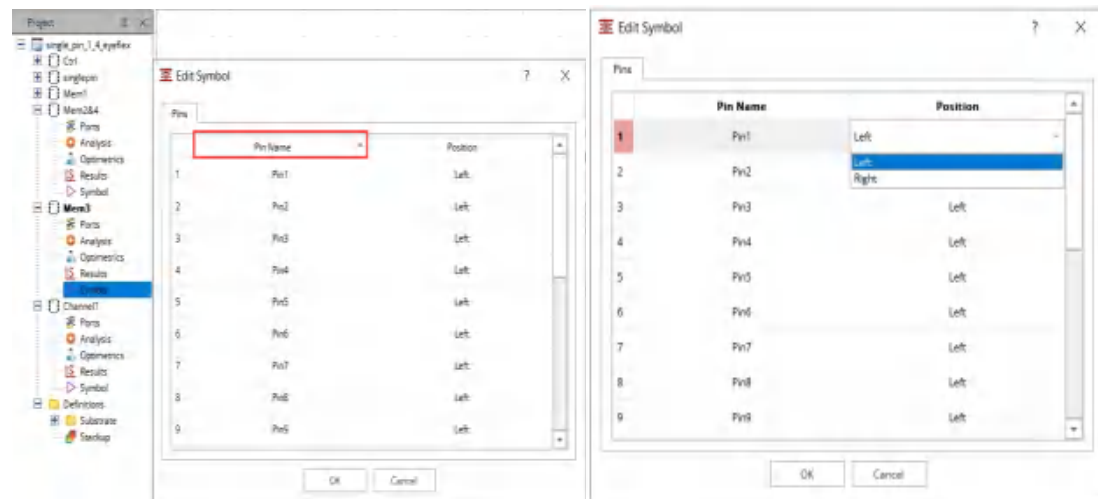


单击 Symbol，快捷键 Shift+E，或者选择 Edit 菜单下的 Enter In ，进入到底层查看

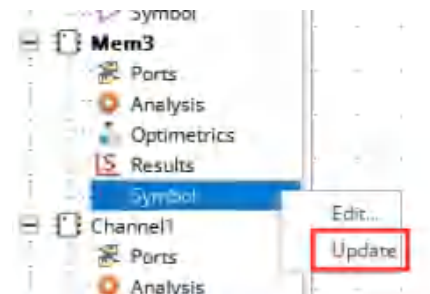


电路。快捷键 B，或者选择 Edit 菜单下的 Pop Out ，返回顶层。

Symbol pin 位置支持调换，右击树节点 Symbol，弹出 Edit 界面，Pin Name 单击，进行排序。Position 下拉给 pin 调整左右位置。

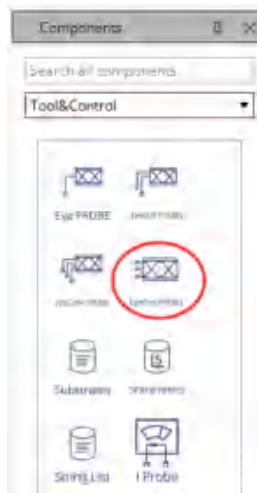


调整好 Symbol，右击树节点 Symbol，选择 Update，自动刷新顶层原理图。



6.3.2.2 EyeFlex 示波器

Flex EyeProbe 主要应用在 DDR 单 Pin Memory 统计仿真电路和常规瞬态 Transient 电路中，可以同时观测多路信号的眼图，并可将时钟抖动考虑进眼图。

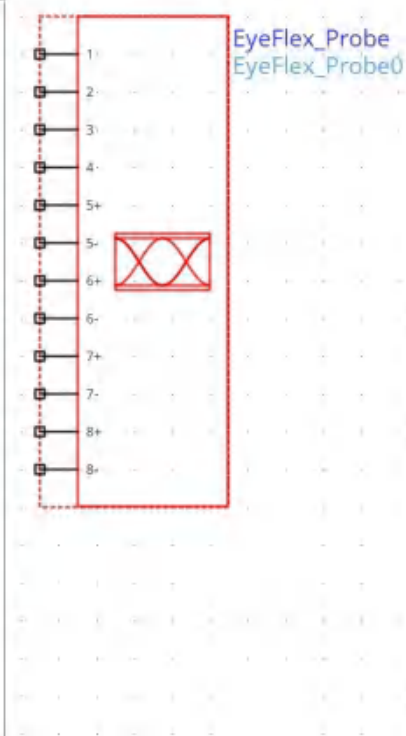
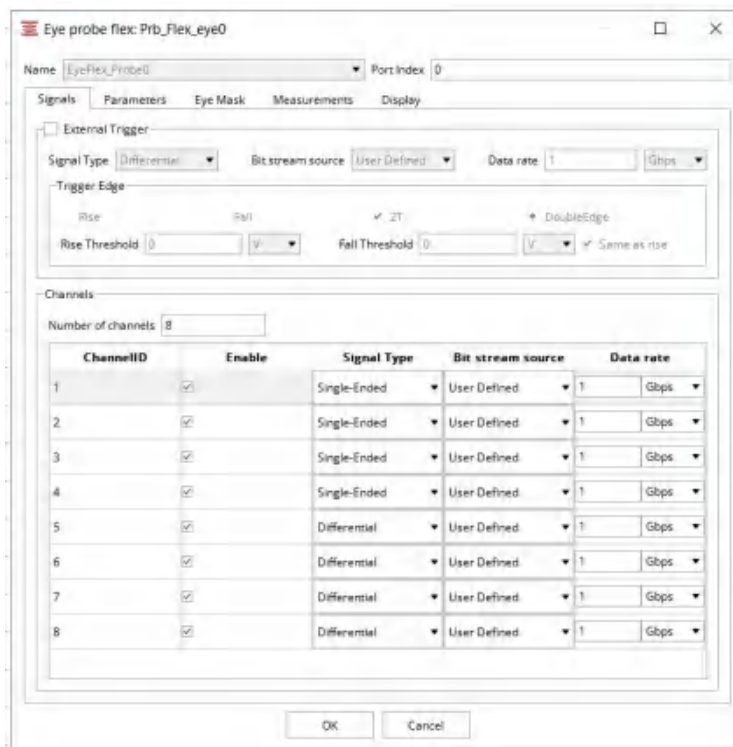


Number of channels 设置通道个数

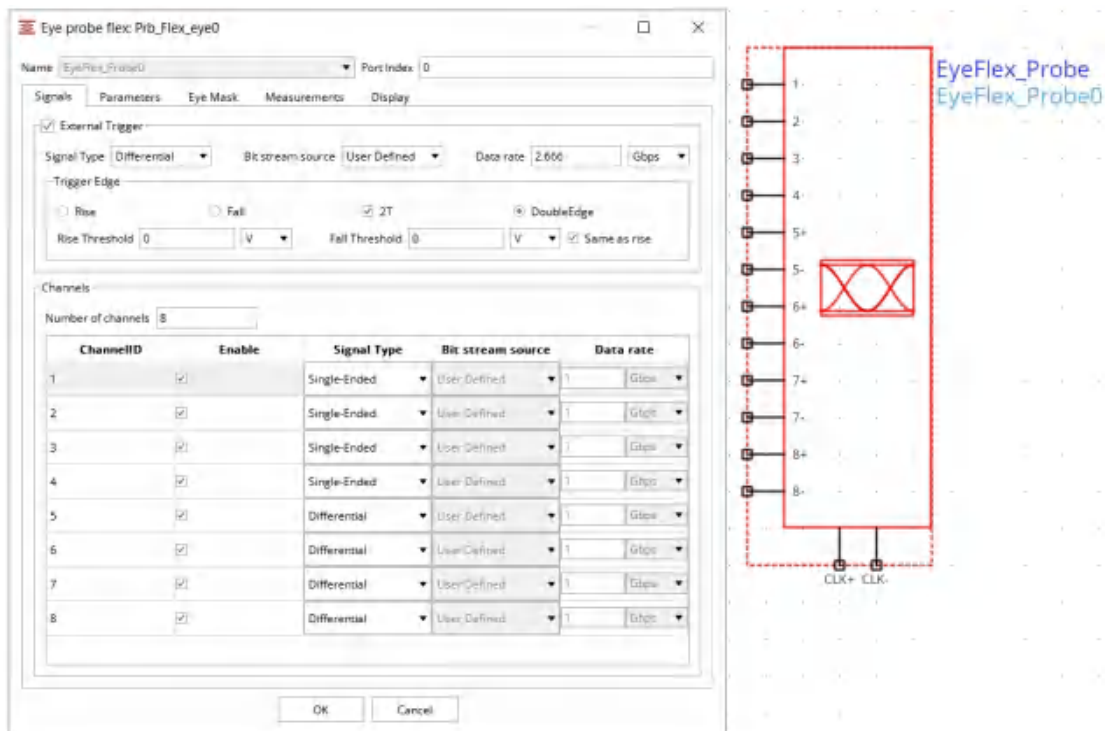
Signal Type 配置差分或者单端信号

Bit stream source 自动检测连接的信号源

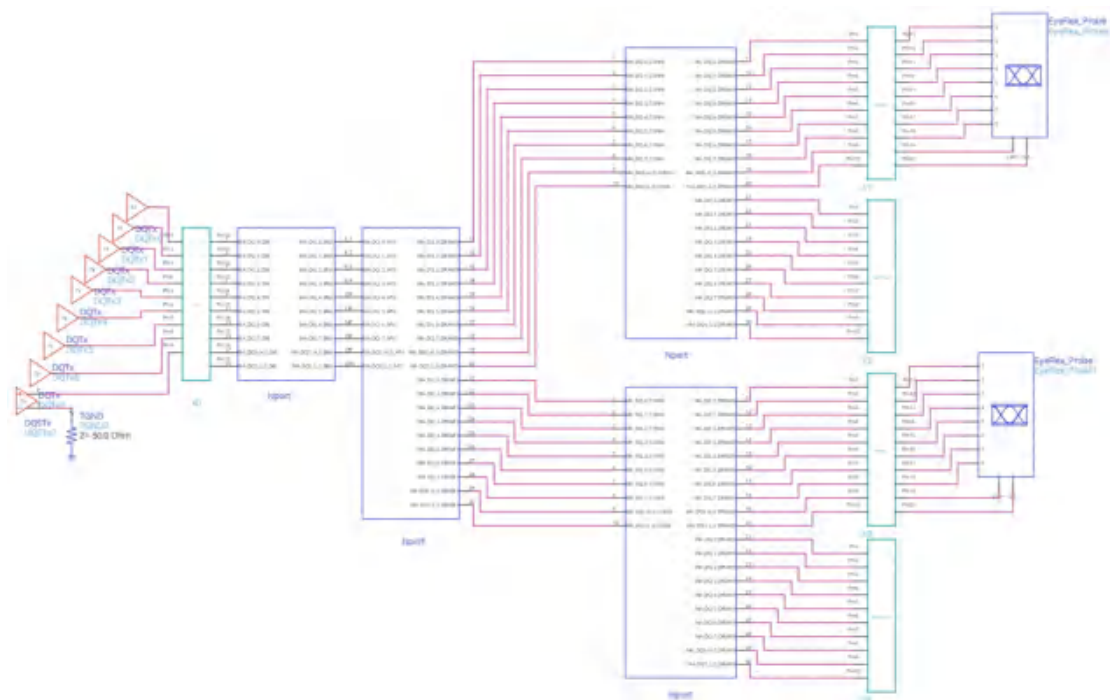
Data rate 瞬态仿真中，每路按照自己的 Data rate 叠眼图，如果勾选 External Trigger，则统一按照 clk 的触发速率叠眼图；Memory 统计模式不使用 Data rate 参数，以发射端内置速率为准。



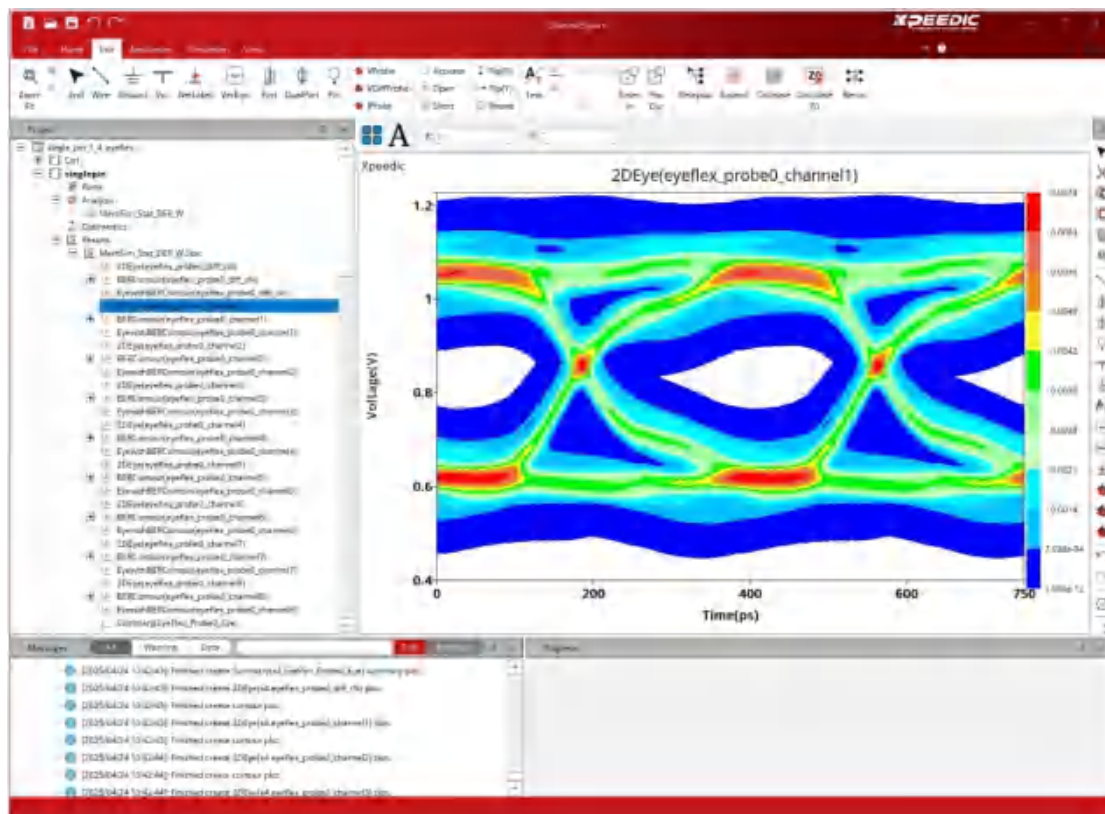
勾选 External Trigger 外部时钟触发，此时信号单独控制的 Bit stream source 和 Data rate 置灰，使能当前时钟信号设置。Trigger Edge 有 Rise 上升沿触发、Fall 下降沿触发、2T 触发、DoubleEdge 双边沿触发。Symbol 对应时钟管脚产生。



下图为 DDR 一驱四典型链路



仿真输出



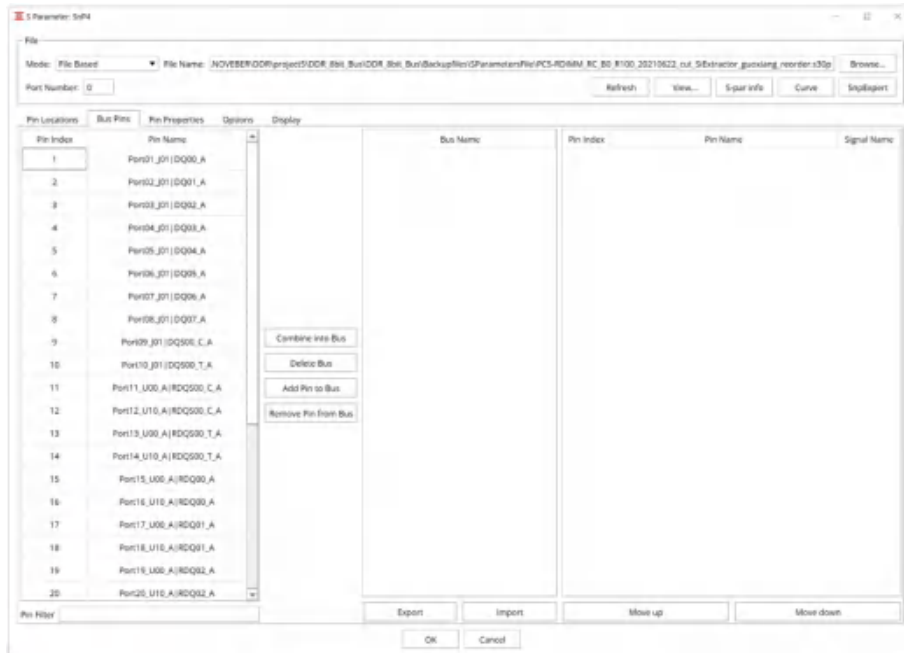
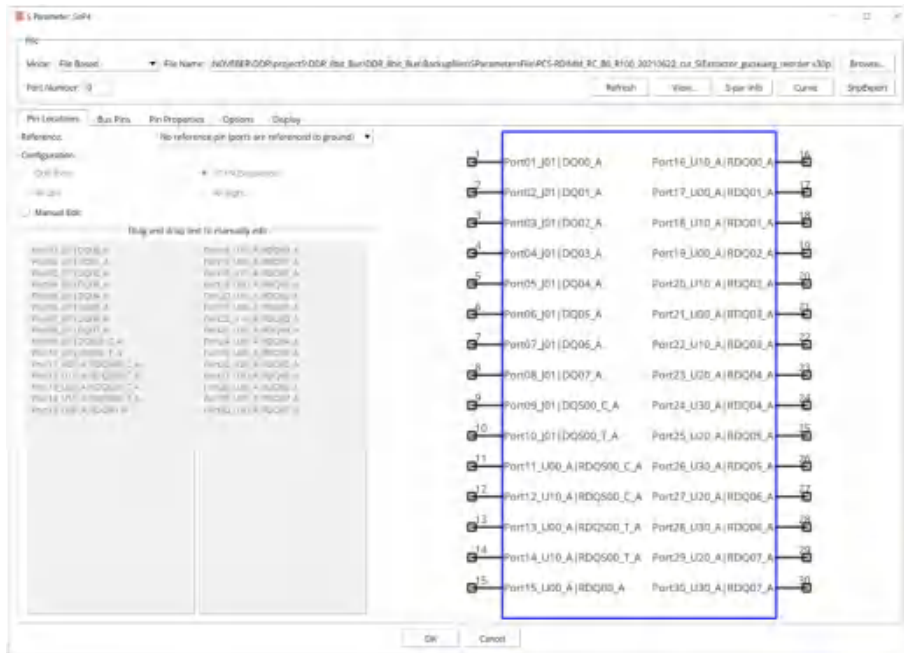
EyeFlex 示波器多信号同时输出 Summary

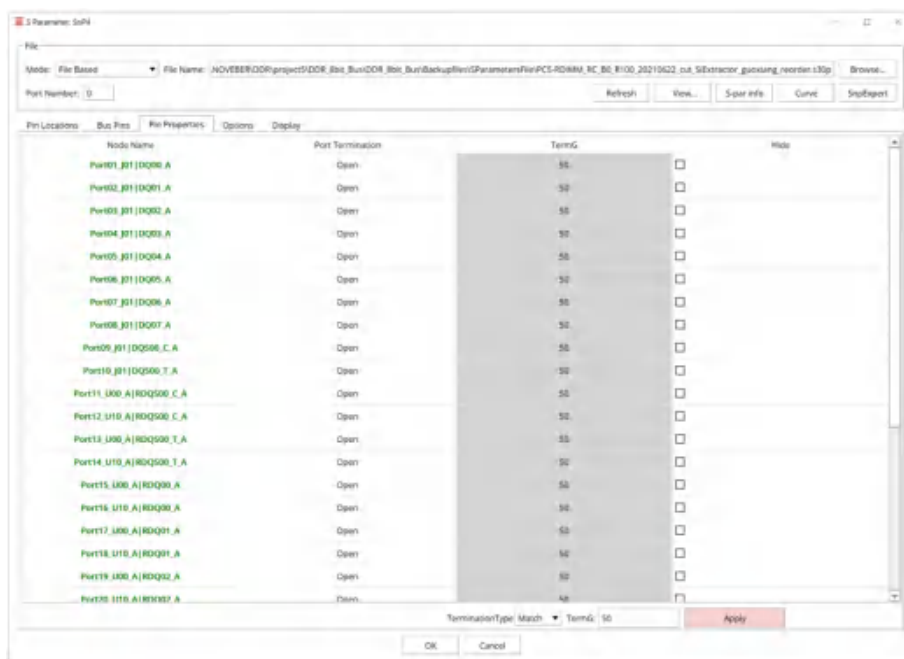
Stat Eye Summary		
Name	PK2PKHeight(V)	PK2PKWidth(ps)
x4.eyeflex_probe0_diff_clk	0.611023	370.708
x4.eyeflex_probe0_channel1	0.0798284	106.97
x4.eyeflex_probe0_channel2	0.101398	123.438
x4.eyeflex_probe0_channel3	0.0876863	110.221
x4.eyeflex_probe0_channel4	0.0968256	120.97
x4.eyeflex_probe0_channel5	0.149633	210.765
x4.eyeflex_probe0_channel6	0.126249	214.566
x4.eyeflex_probe0_channel7	0.153367	215.679
x4.eyeflex_probe0_channel8	0.136911	226.001

Export CSV OK

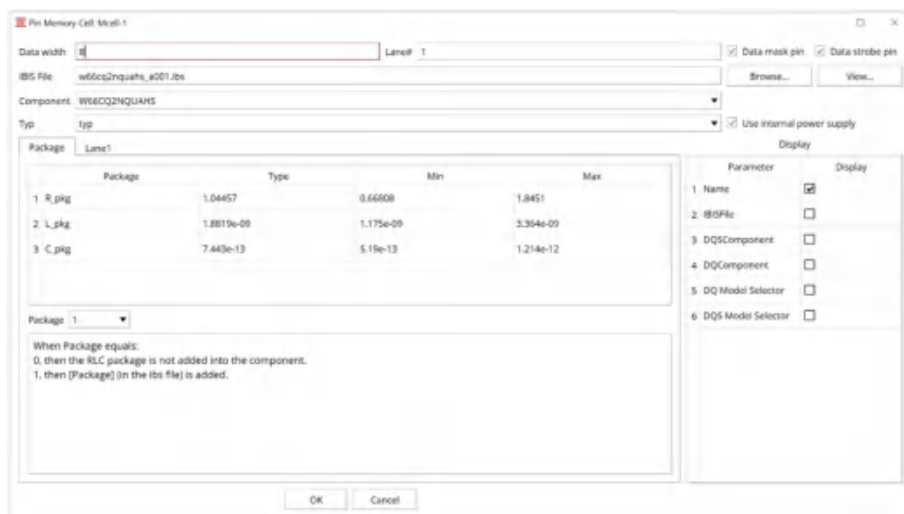
6.3.3 DDR 数据 lane 组模式

DDR BUS 的理想模型下包含了发送接收接收器，可以用作评估 DDR 通路的收发情况。DDR BUS 下的 Ctrl 与 Mem 如下所示：

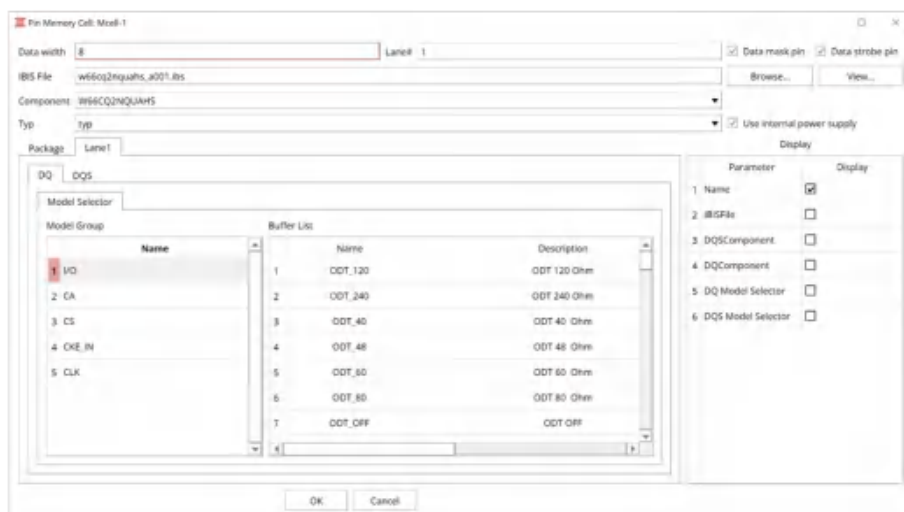




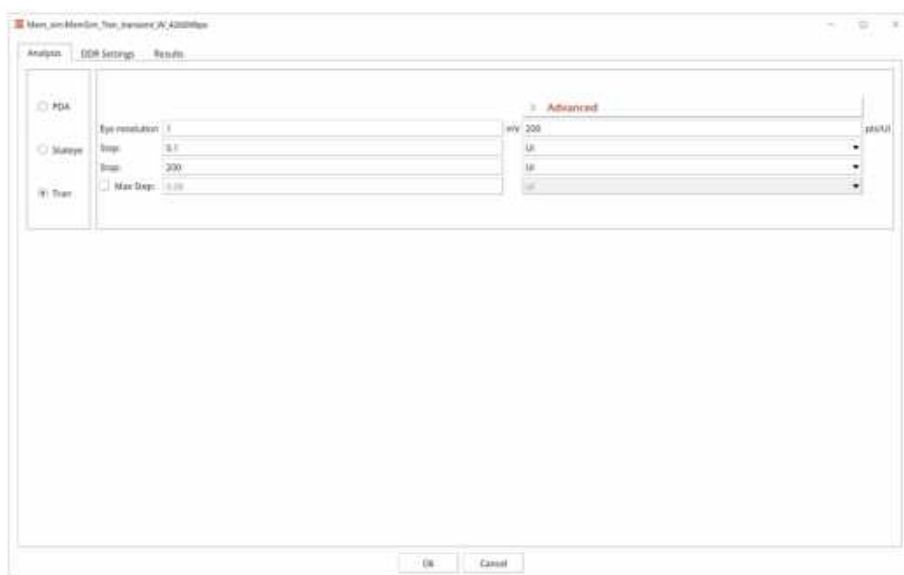
Ctrl/Mem 端：加载完可以使用的 IBIS File 后，选择合适的 Component 和 Typ(可通过勾选 Use internal power supply 选择是否使用内部电源)，Data width 设置数据线长度，Lane# 设置 Rank 数，Data mask pin 为是否显示数据线屏蔽信号；Data strpbe pin 为是否显示 DQS 时钟信号数据线；



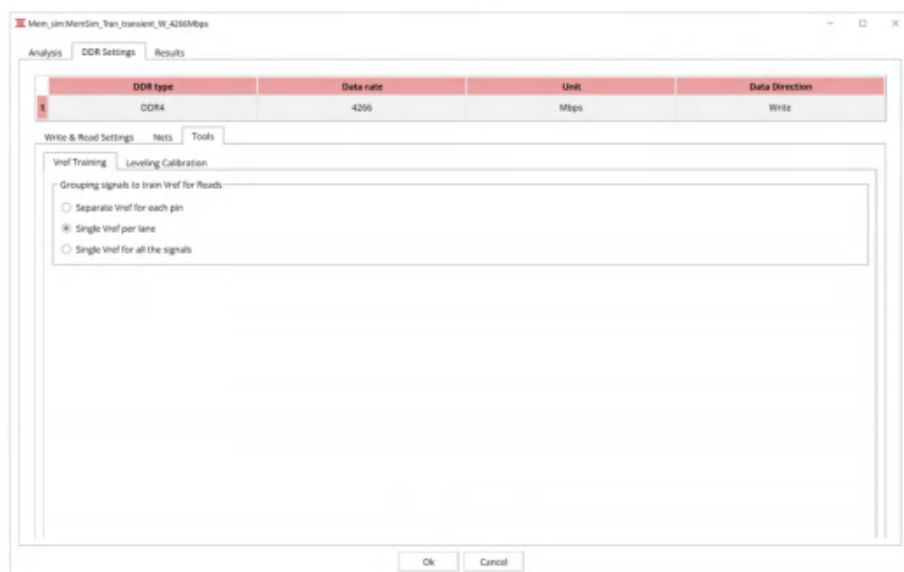
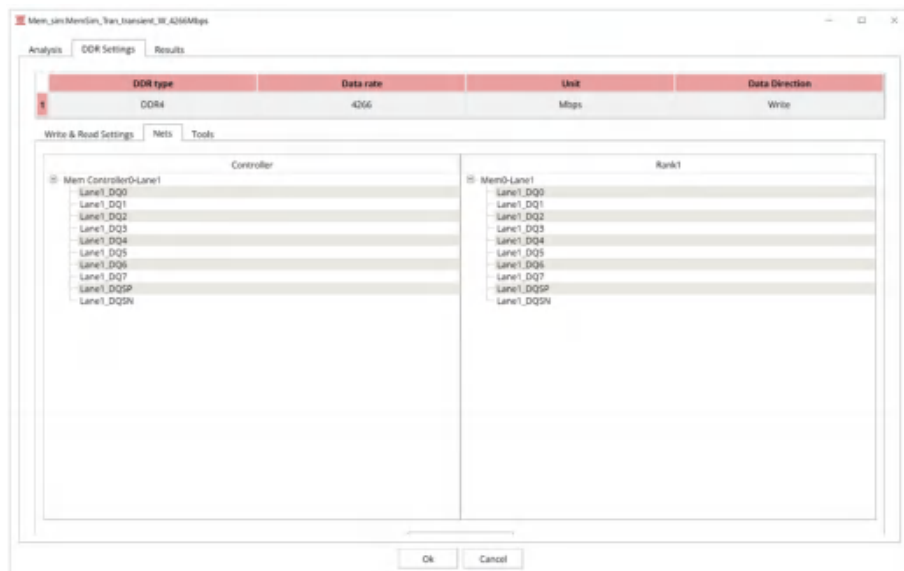
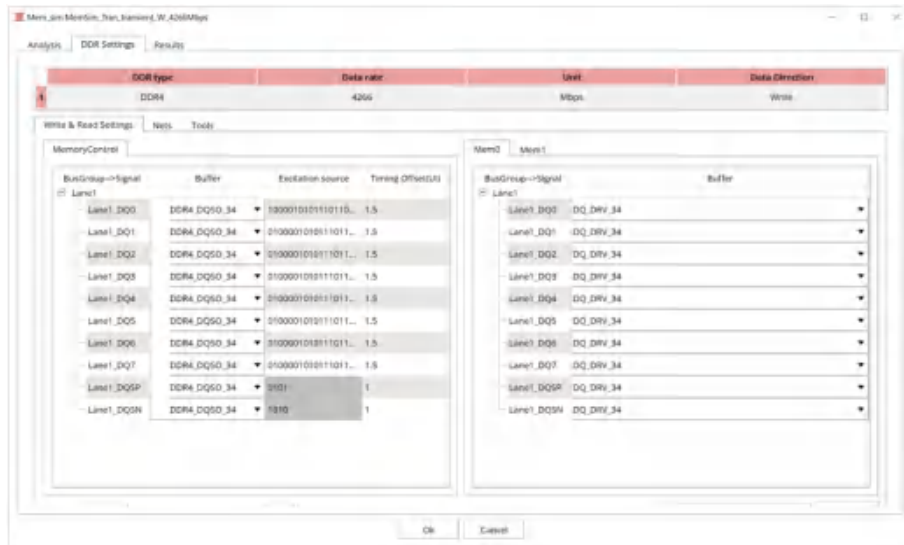
在 Lane 界面处配置 DQ 与 DQS 信号，选择需要配置的 Buffer 组，Display 勾选要展示的项目：



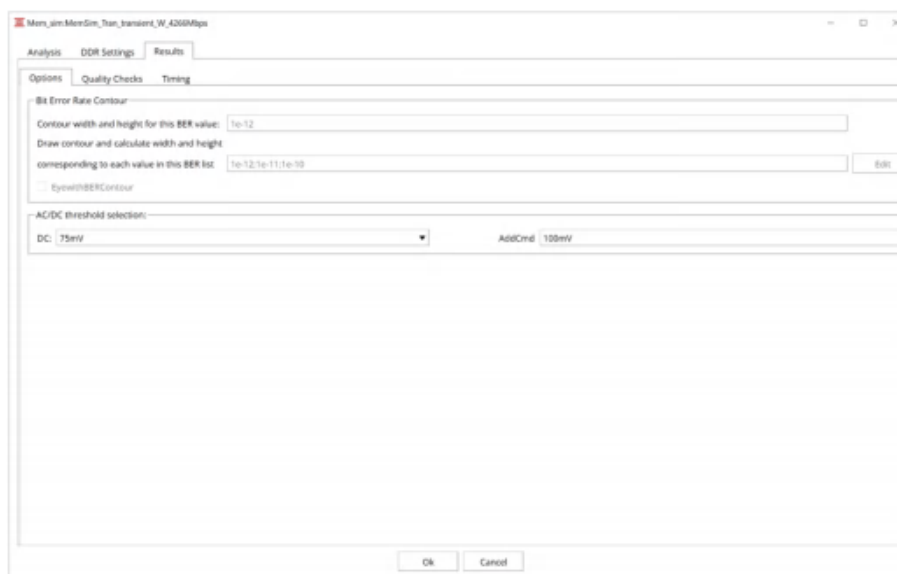
设置 Analysis 的仿真类型(Tran)，设置 Eye resolution、仿真范围（Step 和 Stop 范围）；



在 DDR Setting 下设置 DDR Type 与 DDR rate；通过 Write&Read Settings 界面，设置输入输出 buffer、输入信号的 Excitation source、Timing Offset；通过 Nets 界面可以查看输入输出端的对应关系：

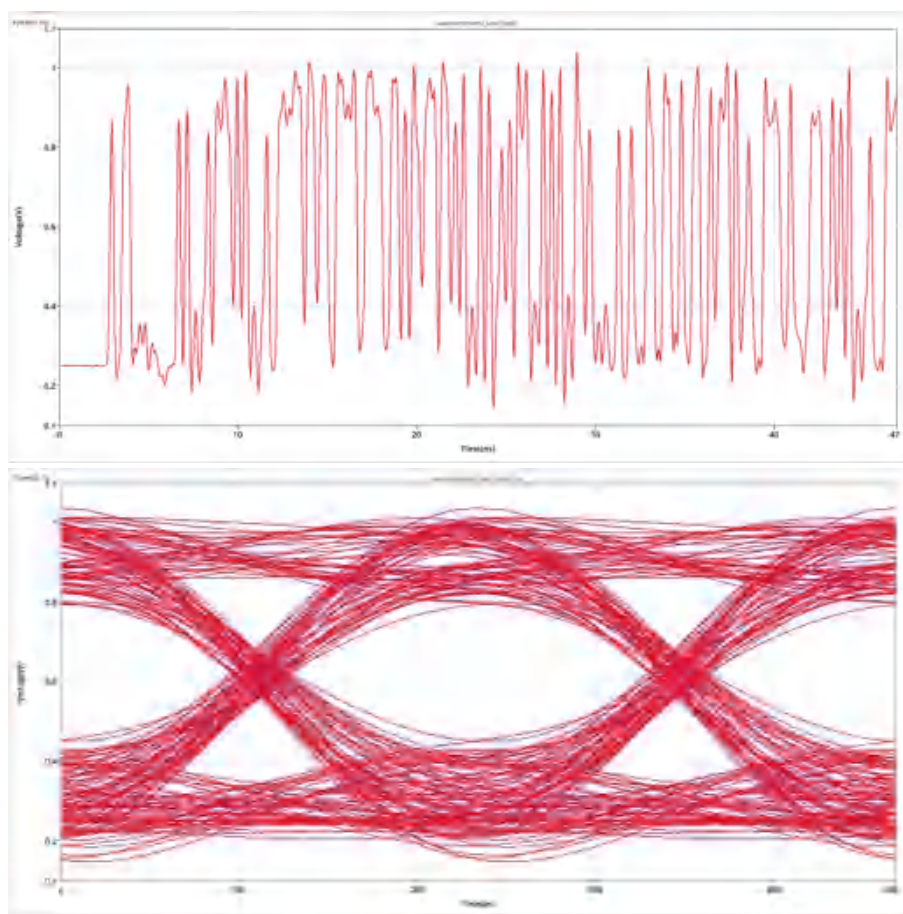


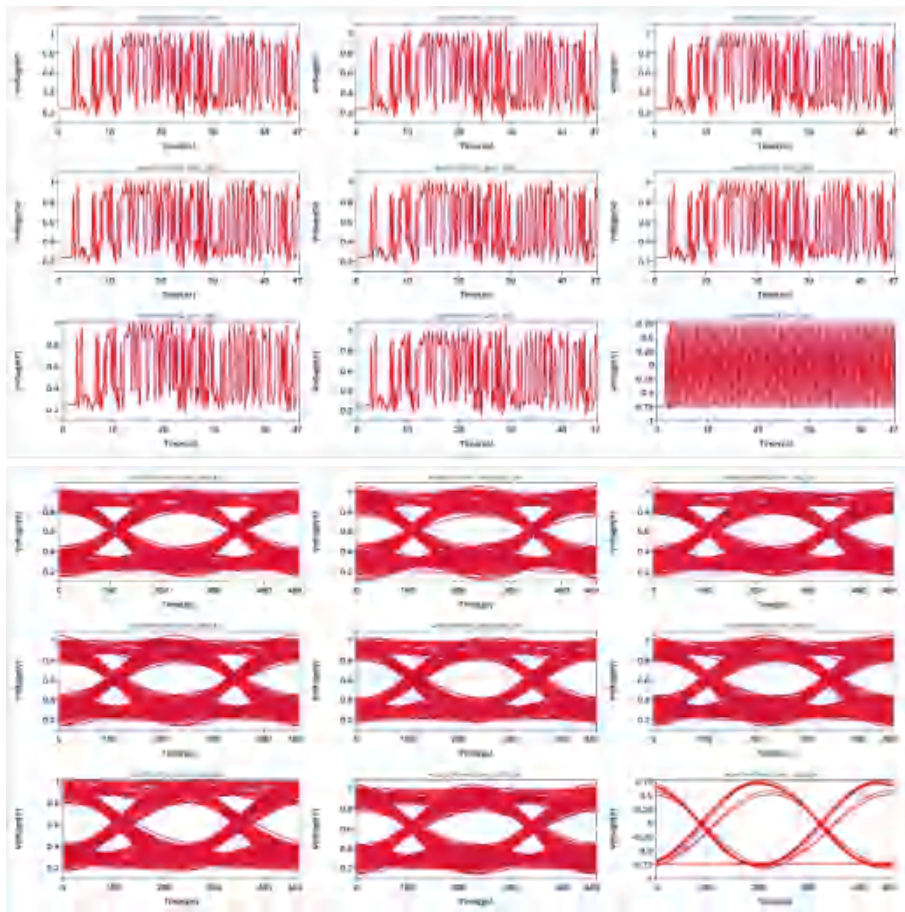
Result 界面可以配置不同的 Options，包括 Bit Error Rate Contour、corresponding to each value in this BER list 等：



6.3.3.3 仿真结果

通过 DDR-8Bit 仿真链路，利用内置的眼图观测和电压观测，可以观测到 DDR 链路的仿真结果，仿真结束后显示 Mem 端输出的眼图以及各个节点的输出电压-时间（V-T）曲线如下图所示：





通过对 VT 曲线和叠眼图分析，可以判别 DDR 通路的收发情况。

6.3.4 DDR 的 Bus 模式

以 DDR4 为例，从电路搭建到仿真，查看结果并输出报告。

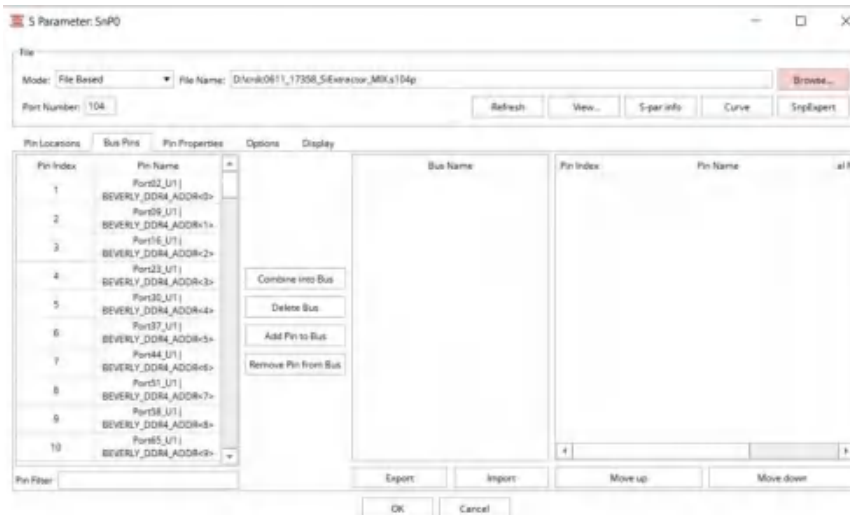
6.3.4.1 仿真链路搭建

S 参数通道 bus 及连通关系配置

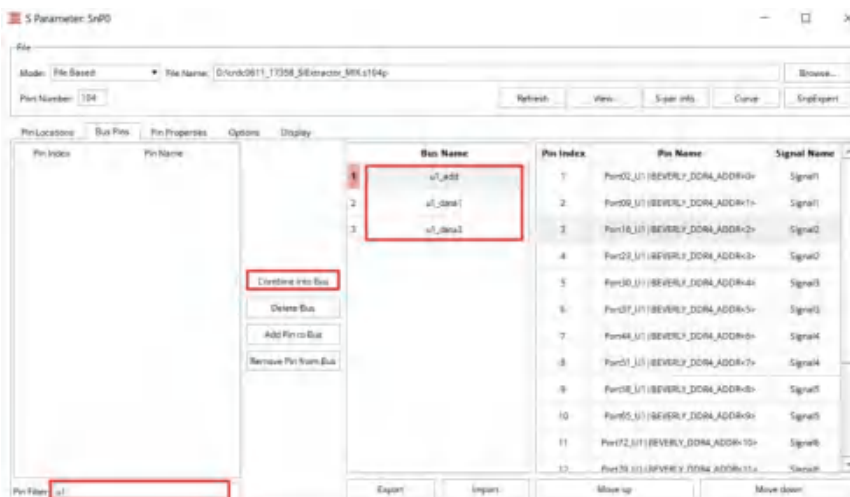
1、 给 DDR 加载通道模型，可以是 S 参数或者 TML。在元件大类 DataSource 下拖动 S 器件到原理图中。



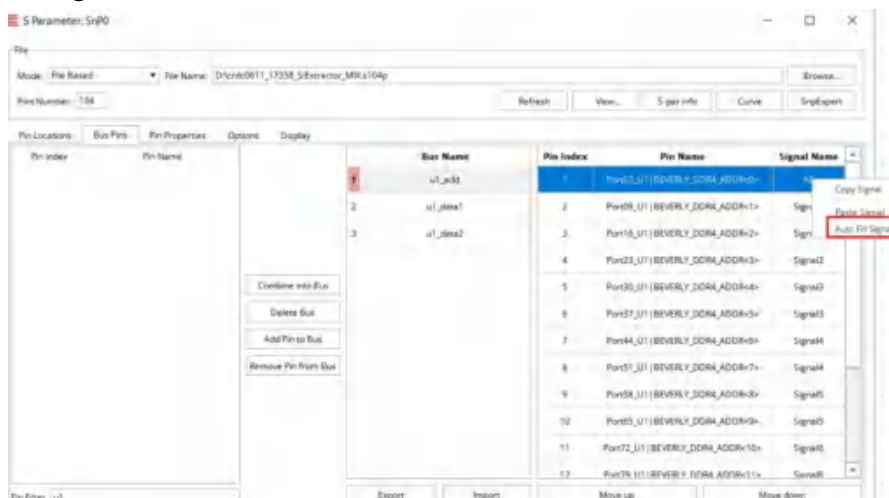
2、 双击 S 器件，加载 S 参数，模型的 pin 需要以 bus 总线方式来显示，选 BusPins。



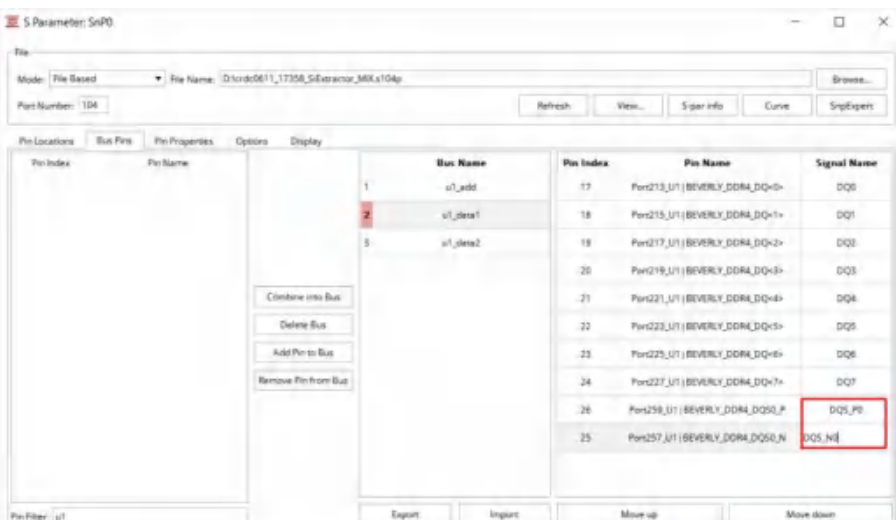
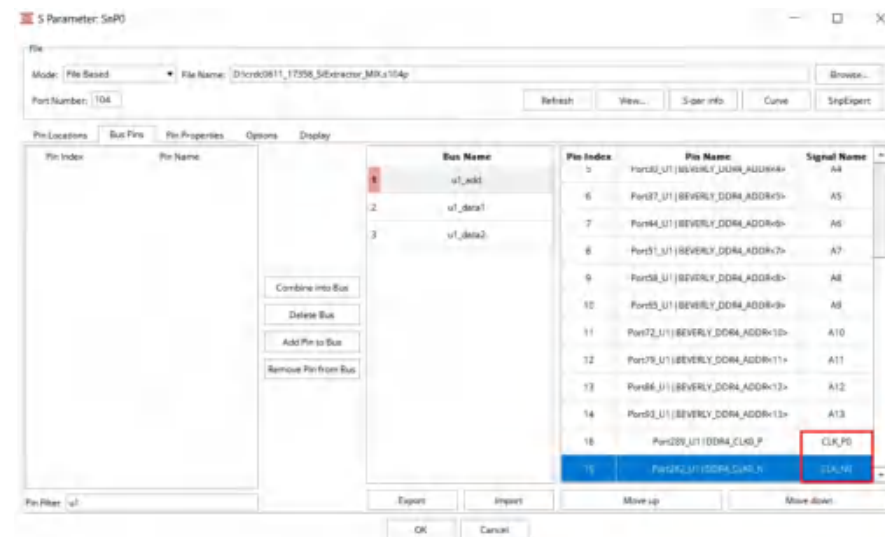
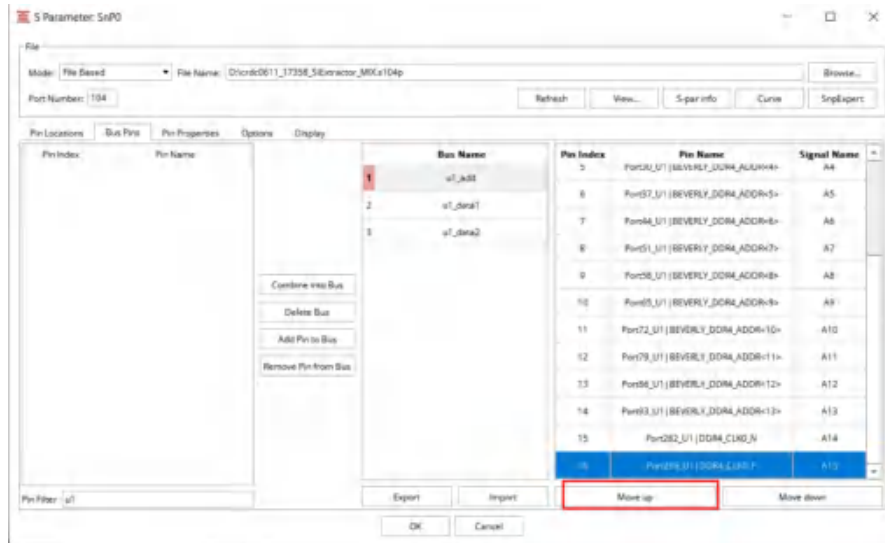
3、 根据 S 参数组 bus，将需要的 pin 加如到对应的 bus 里。通过 PinFilter 选择相同 net
组 bus，点击 Combine into Bus，Busname 可自定义编辑。

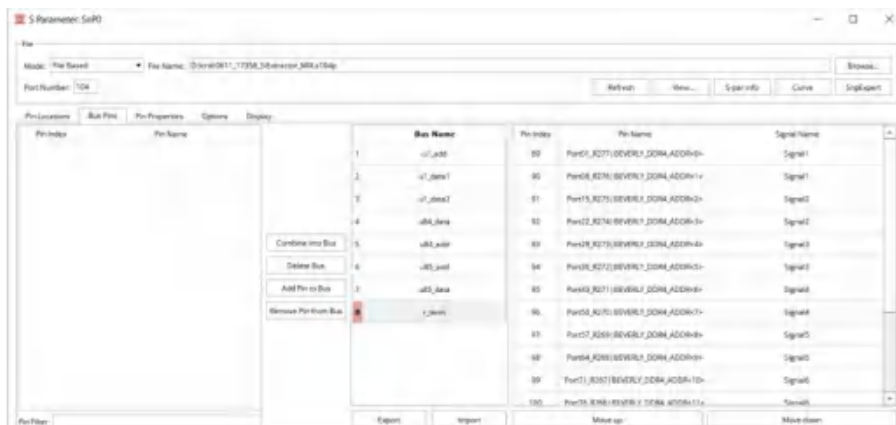


4、 Signal Name 可以编辑，设置一个之后可以批量自动填充，操作是右键点击 Auto fill signal，完成自动批量填充。

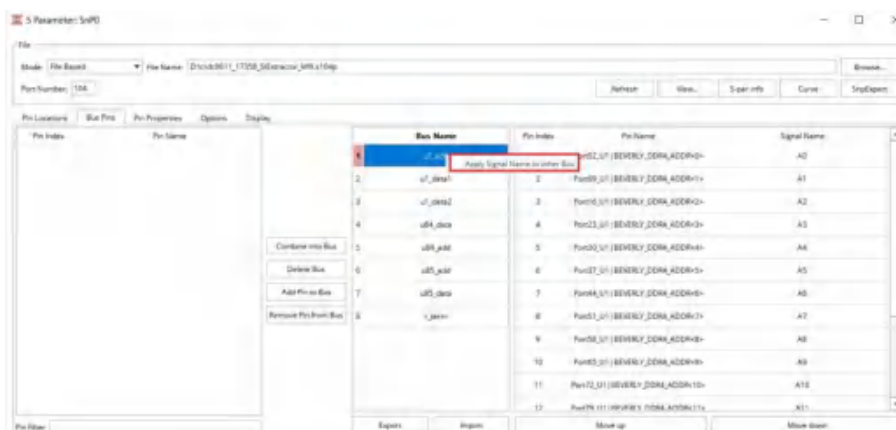


5、 组好 bus 的 pin，根据需要的顺序选中 pin，点击底部 moveup、movedown 调整排序。

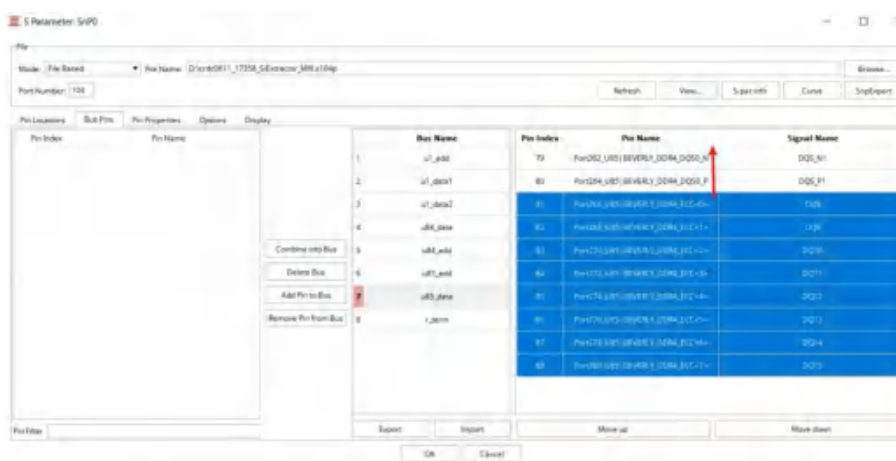




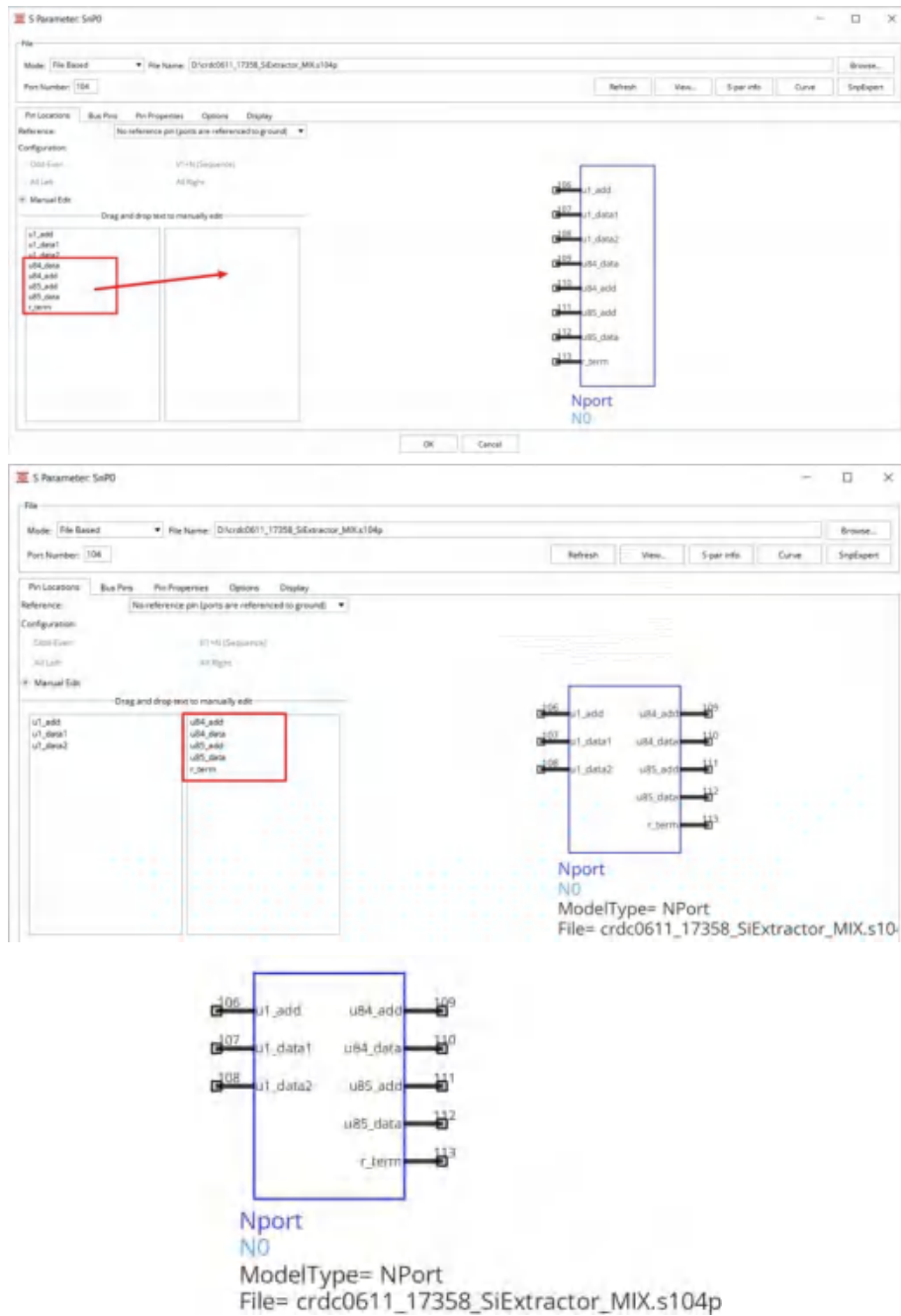
9、连通关系可以批量进行设置。在 u1_add 右键 Apply Signal Name to other Bus，可将连通的 bus 进行连通，对应连通的 bus 的 signal name 会自动变成相同的名字。



10、将作为 ctrl 端的所有 bus 都执行上一步的操作，再检查 pin 是否需要位置移动

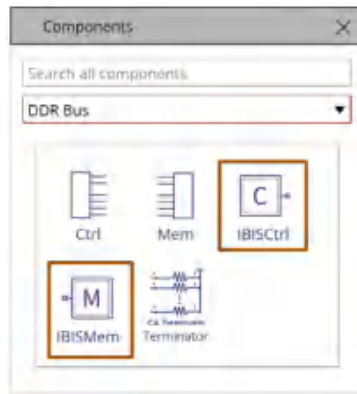


11、bus 全部调整和设置完成后，切换到 Pin location，会自动勾选 ManualEdit，手动拖拽 bus 需要摆放的位置，右侧实时显示原理图。点击 OK，产生整齐的 symbol。

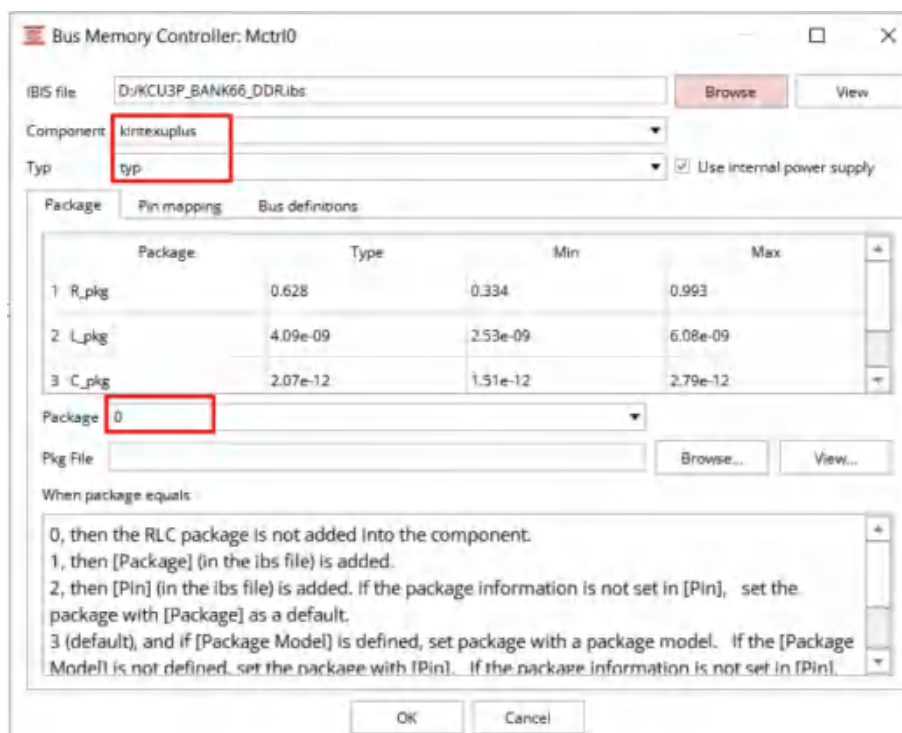


6.3.4.2 IBISCtrl 和 IBISMem 设置

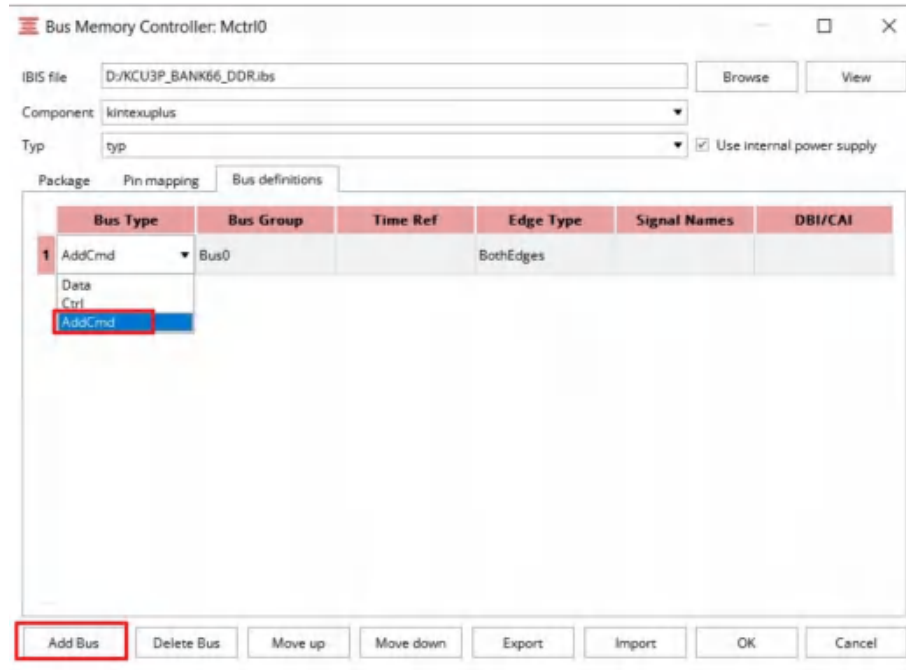
- 1、 在元件大类 DDRBus 下拖动一个 IBISCtrl 与若干个 IBISMem 到原理图中。



2、双击 IBISCtrl，点击 BrowseIBISfile 导入芯片模型。选择 ibis 文件中需要用到的芯片型号，选择 Coner 和合适的 Package。



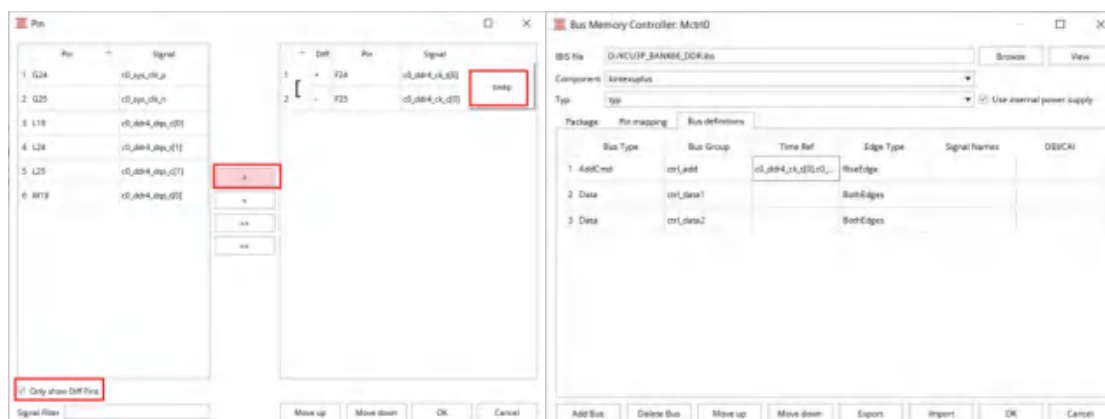
2、切换到 Busdefinitions，BusType 支持 Data、Ctrl、AddCmd；EdgeType 支持 RiseEdge、FallEdge、BothEdges。以一个 AddCmd 触发边沿 RiseEdge 和两个 data 双边沿触发为例。



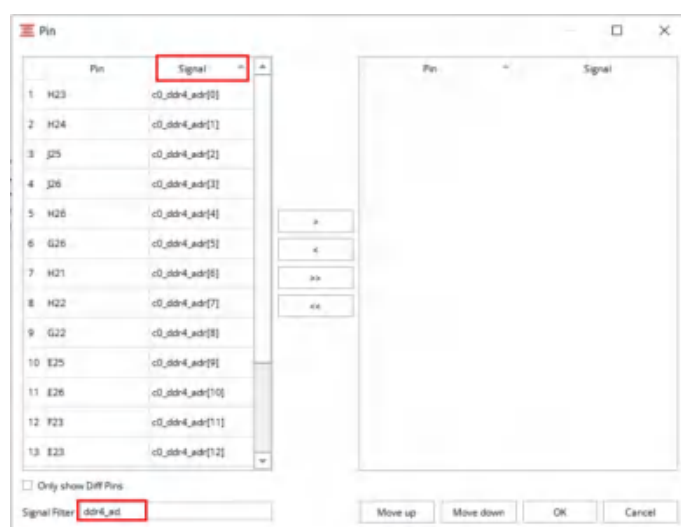
3、 点击 Add bus 可以新增多个 bus，Bus type 为 addcmd 时，Edge type 选择 FallEdge; Bus type 为 data 时，Edge type 选择 Bothdge， Bus Group 可以自定义设置名称。



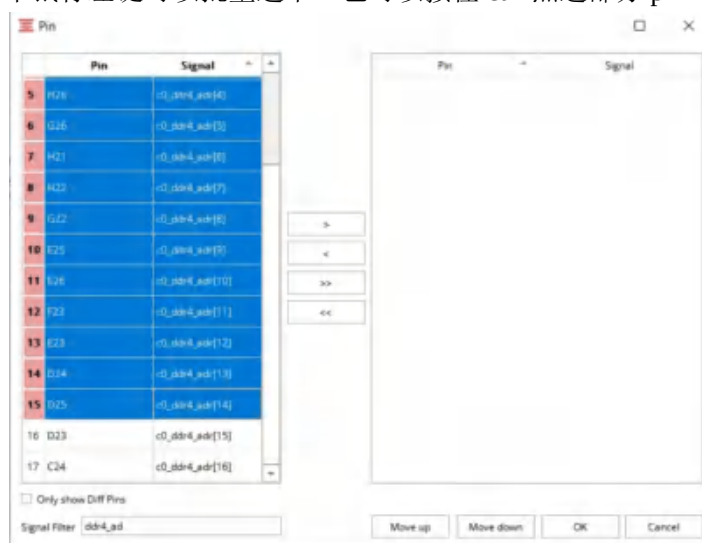
4、 双击 TimeRef，选择差分时钟信号，点击箭头>，加入右侧 Pin，如果要调整 pin 顺序，点击 swap 进行 PN 交换。如果没有合适的差分对，可以去勾选 show diff 手动选择需要的 pin 组成差分对，选择完成后，点击 OK。



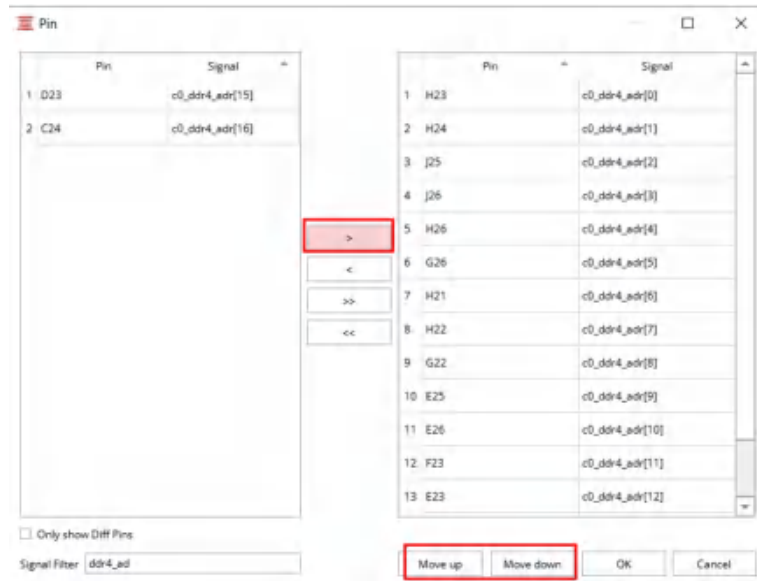
5、双击 SignalNames 下空白的单元格，在弹出的 pin 窗口中，在左侧 Signal Filter 中输入过滤字段会自动刷新展示匹配的结果，点击过滤后的表头的 Signal 字段可以排序。



1、 选中一个 pin 后按住键盘 shift 不松手，滚动滚轮直到出现最后一个需要的 pin，点击一下鼠标左键可以批量选中。也可以按住 ctrl 点选部分 pin。



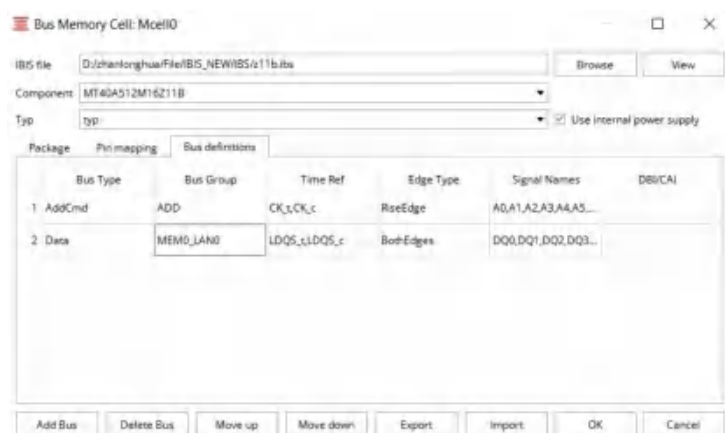
2、 点击向右箭头可以批量移动至右边，移动过去的 pin 可以点击向左箭头<从右侧删除，也可以进行 move up，move down 选项。完成后，点击 ok。



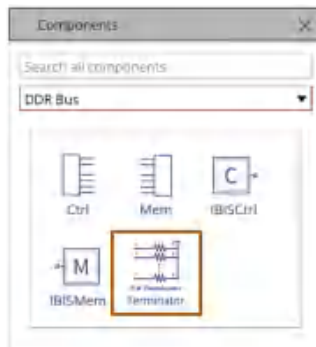
3、再按照配置 add 一样的操作配置 data 信号的 Time Ref 和 Signal Name



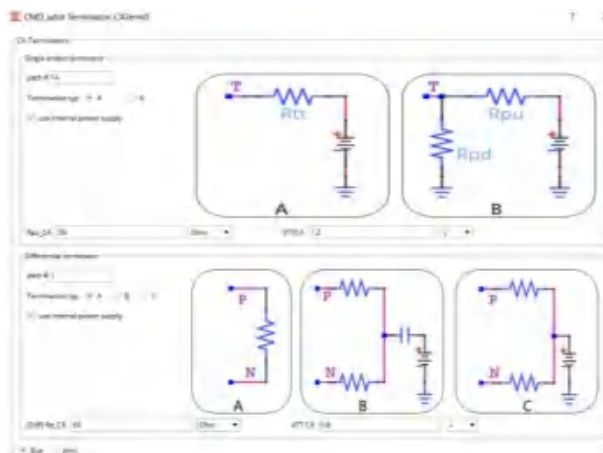
4、mem 端配置与 ctrl 端相同，依次设置 mem 端的信号。如果 mem 端使用的配置相同可以复制生成一个相同的 mem。



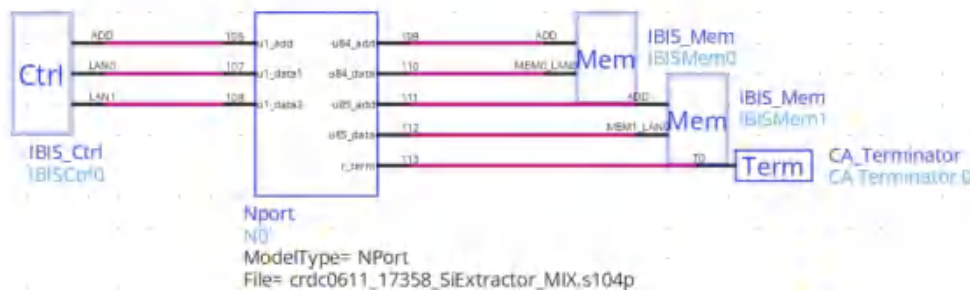
10、从元件 DDR bus 中拖入 CA Terminator。



11、双击 CA_Terminator，可以设置单端和差分的信号的电阻，电压等参数。



11、点击 ctrl 暴露出来的 pin 会自动生成线与其他 pin 脚相连，或者进行碰撞连接。



12、连线完成后，可以双击 bus 线查看 pin mapping 关系，可以在 setting 界面对 pin mapping 进行导入导出，移动等设置。

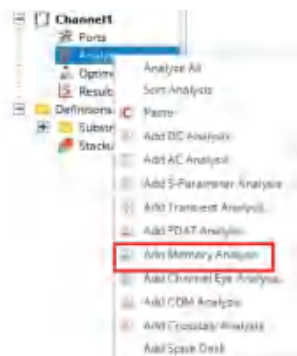
Bus Pin Mapping Setting

Pin Name (IBISCtrl)	Signal Name (IBISCtrl)	Pin Name (NPort)	Signal Name (NPort)	Probe
H23	c0_ddr4_addr[0]	Port0_U11	BEVERLY_DDR4_ADDR<0>	☐
H24	c0_ddr4_addr[1]	Port0_U11	BEVERLY_DDR4_ADDR<1>	☐
J25	c0_ddr4_addr[2]	Port1_U11	BEVERLY_DDR4_ADDR<2>	☐
J26	c0_ddr4_addr[3]	Port2_U11	BEVERLY_DDR4_ADDR<3>	☐
H26	c0_ddr4_addr[4]	Port3_U11	BEVERLY_DDR4_ADDR<4>	☐
G26	c0_ddr4_addr[5]	Port7_U11	BEVERLY_DDR4_ADDR<5>	☐
H21	c0_ddr4_addr[6]	Port4_U11	BEVERLY_DDR4_ADDR<6>	☐
H22	c0_ddr4_addr[7]	Port5_U11	BEVERLY_DDR4_ADDR<7>	☐
G22	c0_ddr4_addr[8]	Port6_U11	BEVERLY_DDR4_ADDR<8>	☐
E25	c0_ddr4_addr[9]	Port7_U11	BEVERLY_DDR4_ADDR<9>	☐

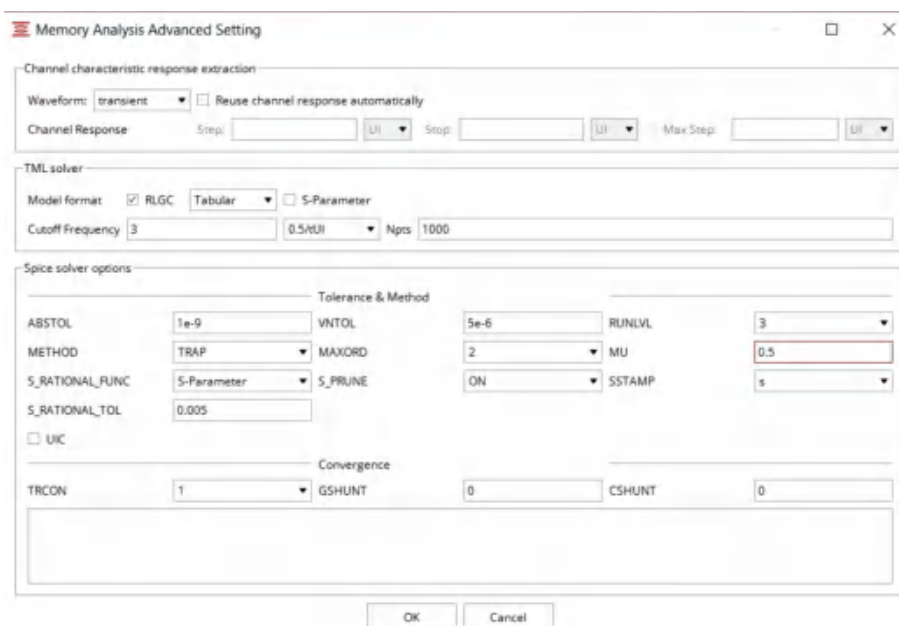
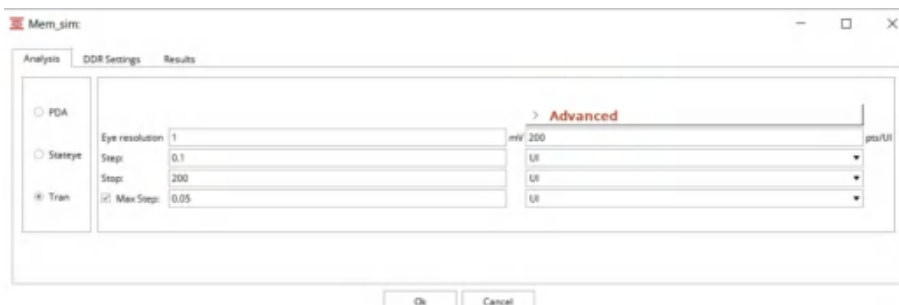
Buttons: Export, Import, Reset, Move up, Move down, By Pin Index, By Pin Name, By Signal, OK, Cancel

6.3.4.3 设置仿真参数

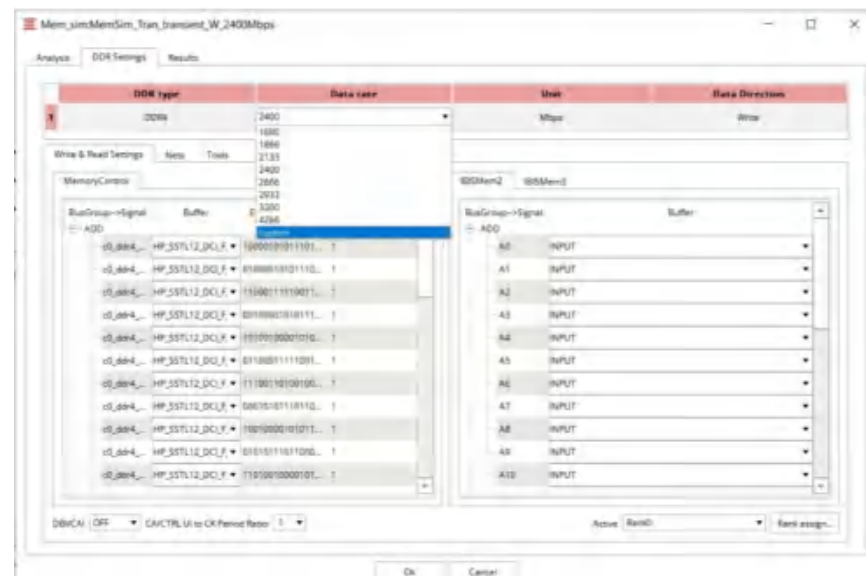
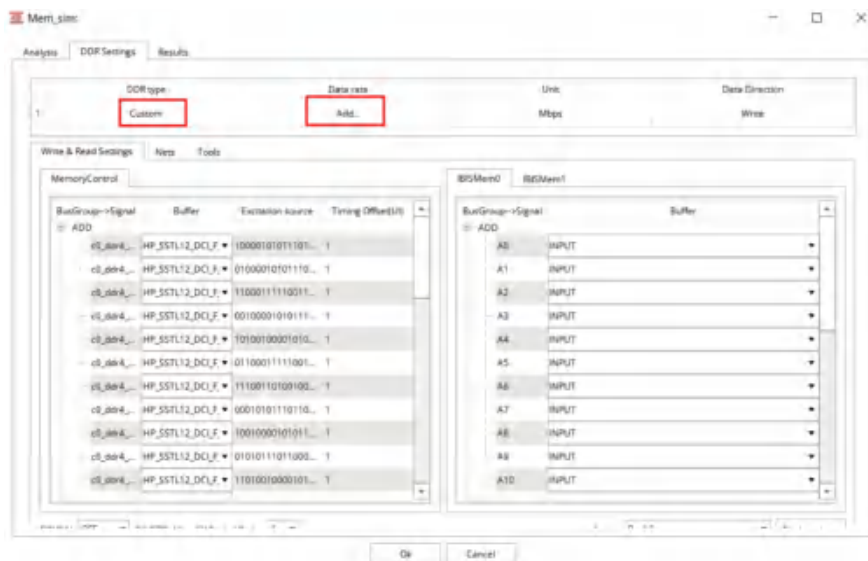
1、 树节点右键点击 Add Memory Analysis



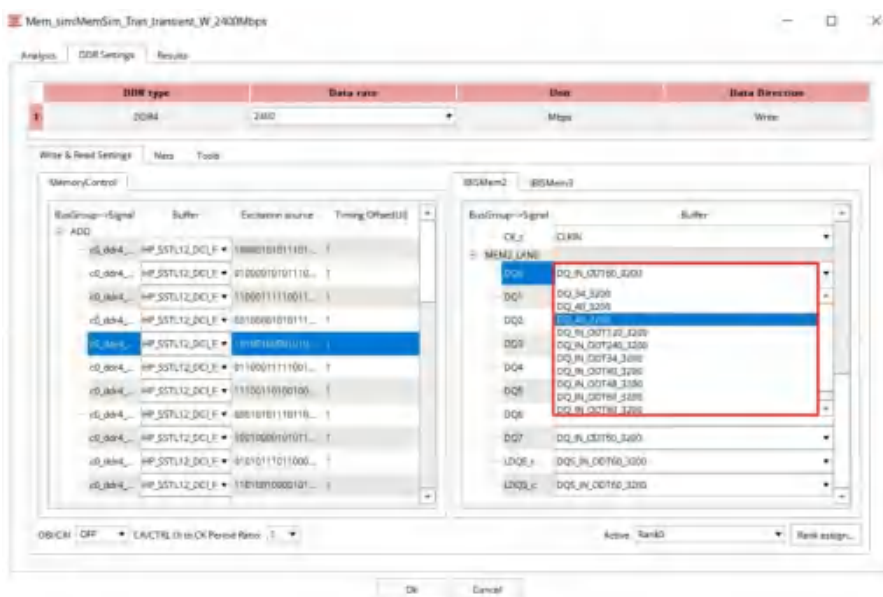
2、 仿真类型选择 Tran，仿真类型如果选择 Stateye 或者 PDA 时，是无法使用出报告功能的。可以设置 stop、step 等参数，点击 Advanced 可以设置仿真相关的参数。



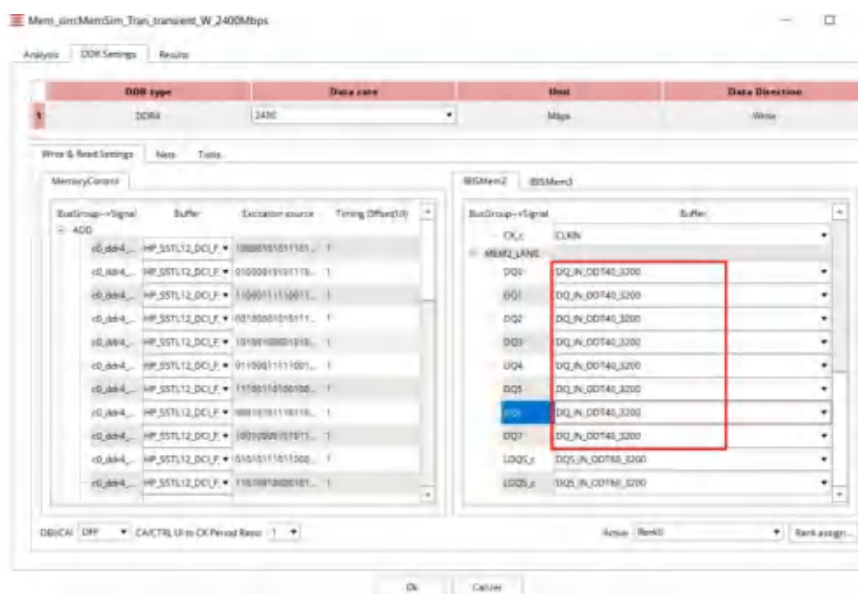
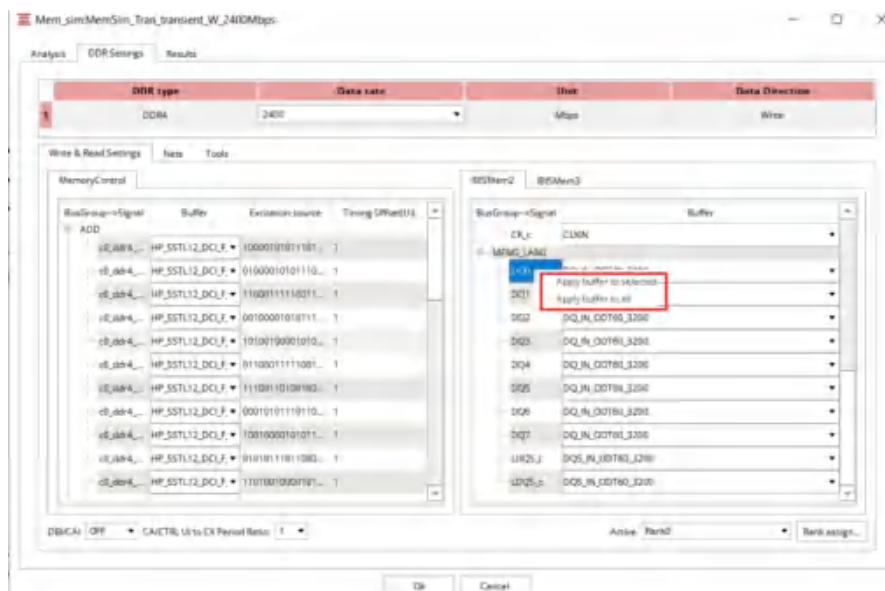
3、 DDR Settings 中设置仿真类型和速率及方向，默认 type 为 Custom 时，在结果节点是不会展示 view HtmlReport，点击 Custom 可以切换不同类型的 type，目前包括 DDR3-1.5V, DDR31.35V, DDR31.25V, DDR4, DDR5，切换 ddr type 后可以选择对应协议的速率。Data Direction 可以选择 WRITE 和 READ 两种。



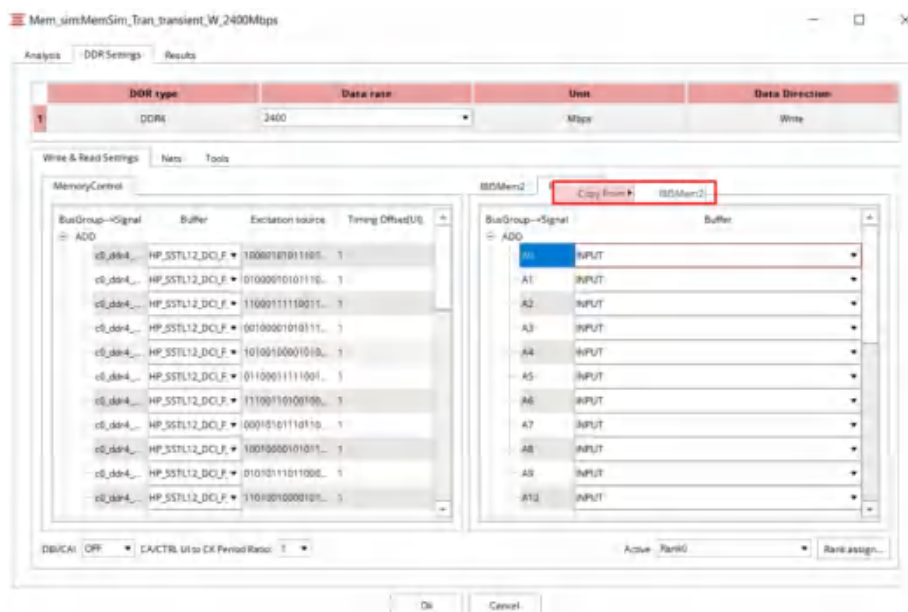
4、针对单个 buffer,点击需要编辑的 buffer, 可以对 buffer 进行切换。



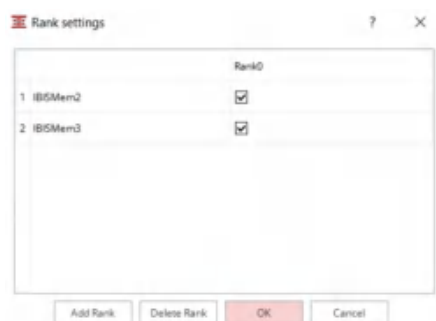
5、 点击信号名右键会出现 Apply buffer to selected/Apply buffer to all。当选中了多个信号时，可以选择应用到选中，会将选中的 buffer 批量修改，如果选择 Apply buffer to all，会将同类型的当前 mem 或者 ctrl 端的信号批量修改。Dqs 或者 clk 的批量修改方法一样。



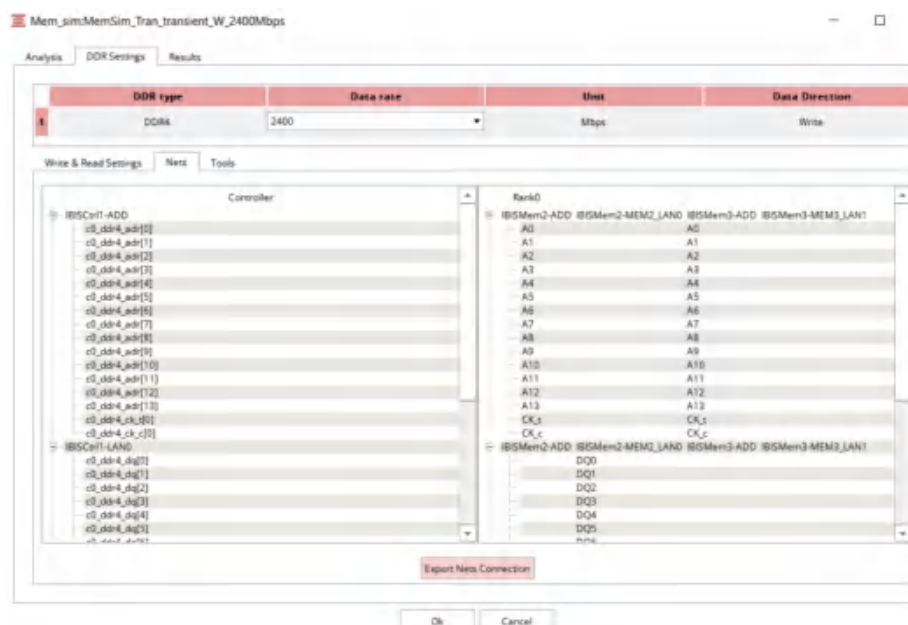
6、 在未配置的 mem 名称上右键，可以应用其他 mem 的配置到自己，点击 copy from 后面的 mem，可以自动将选择的 mem 的 buffer 信息全部同步到当前的 mem 中。



7、 点击 Rank Assign 可以对颗粒端进行分配。



8、 点击 nets，可以查看连接状态



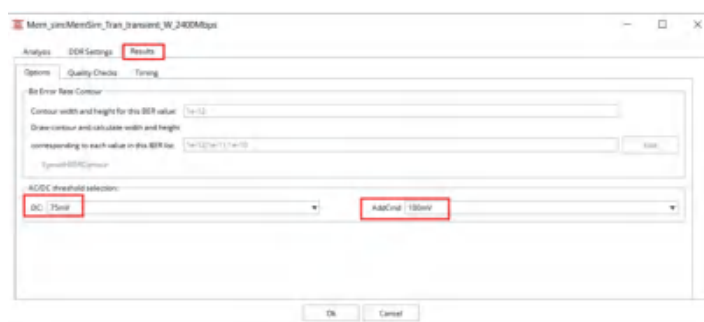
9、 点击 Tools，可以配置 Verf 策略，选择训练的时候是以单个信号还是 Lan 等模式。



10、点击 Leveling Calibration，可以设置是否开启校准补偿，并针对读写，地址数据信号分别设置校准策略。



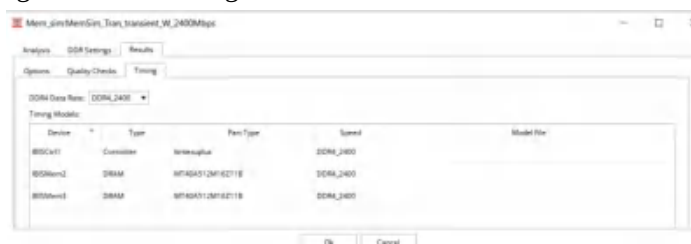
11、点击 Result，可以设置 AC/DC threshold



12、点击 Quality Check，可以设置 JEDEC 相关参数，去勾选后可以自定义 jedec 参数。

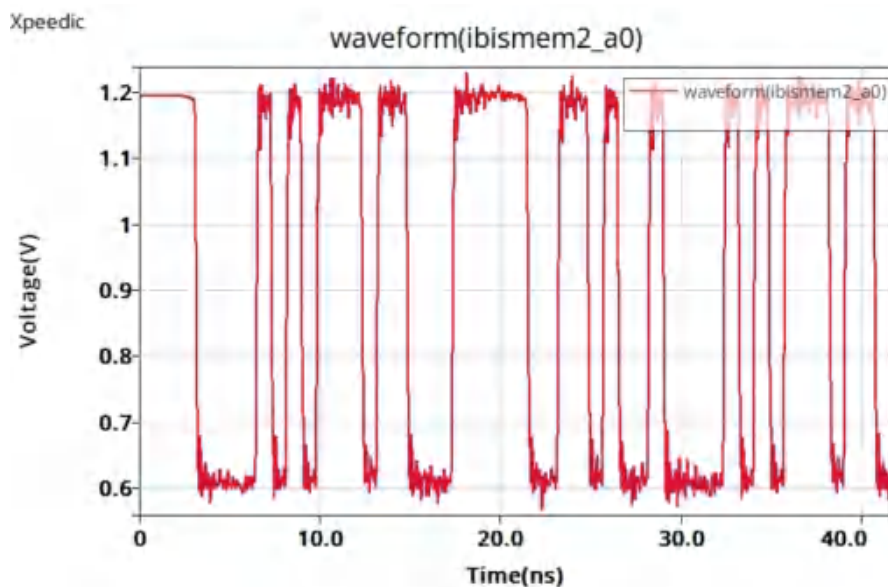


13、点击 Timing 可以查看 Timing models。

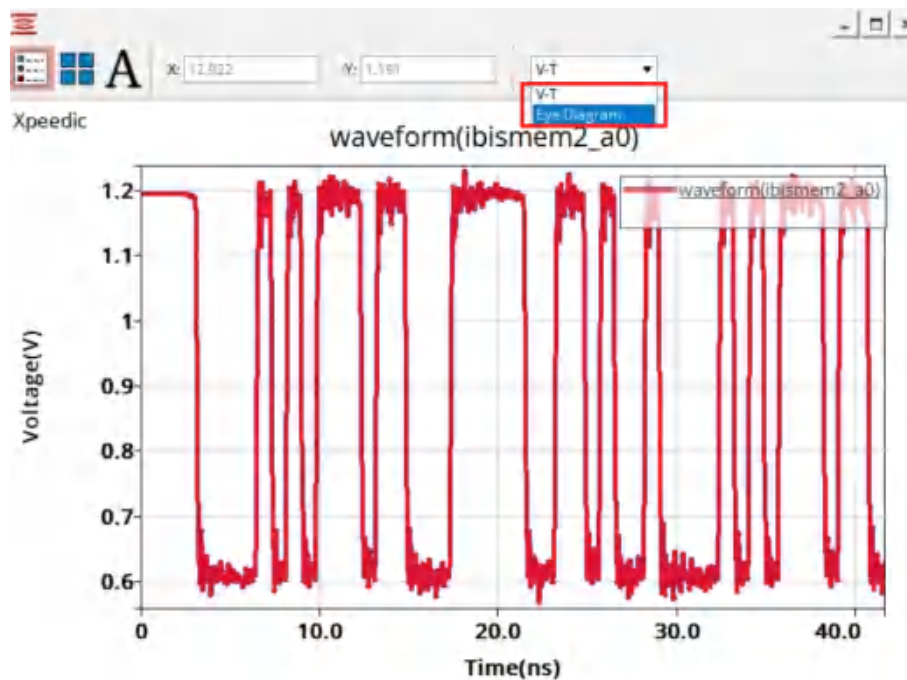


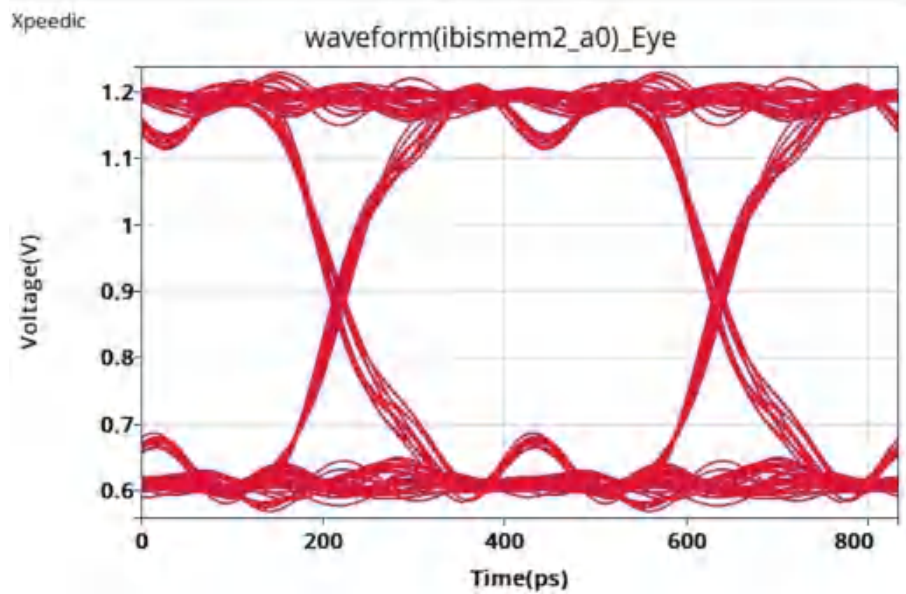
6.3.4.4 仿真及出报告

- 1、 启动仿真后，待仿真完成可以查看 dq 信号的 VT 及眼图情况。

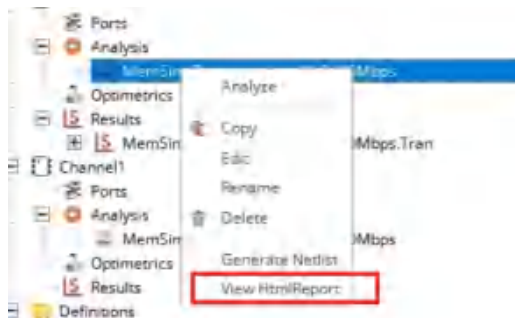


- 2、 点击 VT，选择 Eye Diagram，可查看眼图。



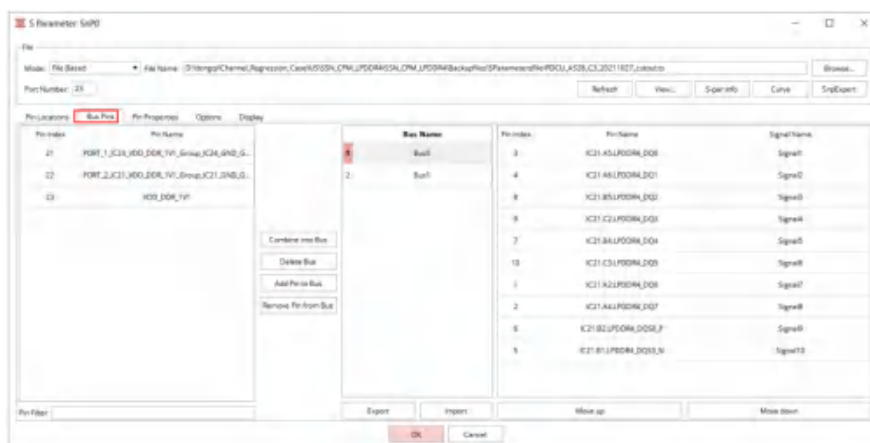


3、 点击仿真节点右键，点击 View HtmlReport，会自动生成本次仿真的 HTML 报告并在浏览器中打开。



4、 在浏览器中可以查看 html 报告，在报告中可以查看建立保持时间，eyemask 等。

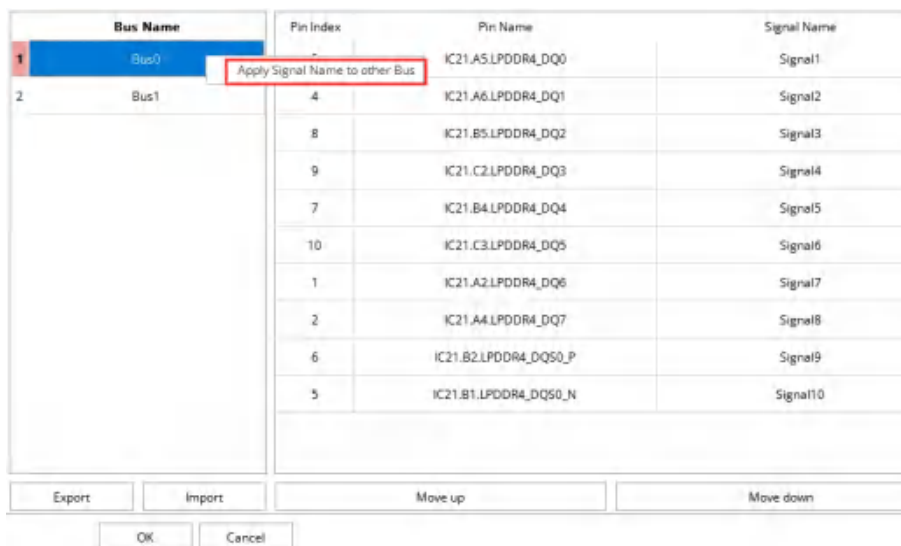
Signal	Type	Length	Delay	Setup	Hold	Eye Mask	Setup Time
1	Signal	1000000	1000000	1000000	1000000	1000000	1000000
2	Signal	1000000	1000000	1000000	1000000	1000000	1000000
3	Signal	1000000	1000000	1000000	1000000	1000000	1000000
4	Signal	1000000	1000000	1000000	1000000	1000000	1000000
5	Signal	1000000	1000000	1000000	1000000	1000000	1000000
6	Signal	1000000	1000000	1000000	1000000	1000000	1000000
7	Signal	1000000	1000000	1000000	1000000	1000000	1000000
8	Signal	1000000	1000000	1000000	1000000	1000000	1000000
9	Signal	1000000	1000000	1000000	1000000	1000000	1000000
10	Signal	1000000	1000000	1000000	1000000	1000000	1000000
11	Signal	1000000	1000000	1000000	1000000	1000000	1000000
12	Signal	1000000	1000000	1000000	1000000	1000000	1000000
13	Signal	1000000	1000000	1000000	1000000	1000000	1000000
14	Signal	1000000	1000000	1000000	1000000	1000000	1000000
15	Signal	1000000	1000000	1000000	1000000	1000000	1000000
16	Signal	1000000	1000000	1000000	1000000	1000000	1000000
17	Signal	1000000	1000000	1000000	1000000	1000000	1000000
18	Signal	1000000	1000000	1000000	1000000	1000000	1000000
19	Signal	1000000	1000000	1000000	1000000	1000000	1000000
20	Signal	1000000	1000000	1000000	1000000	1000000	1000000



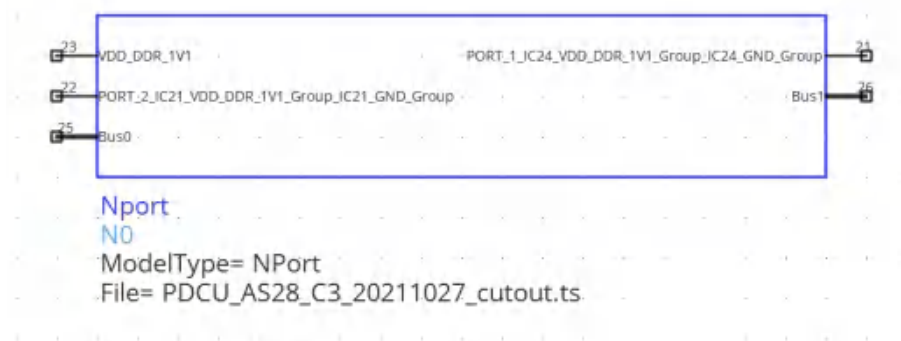
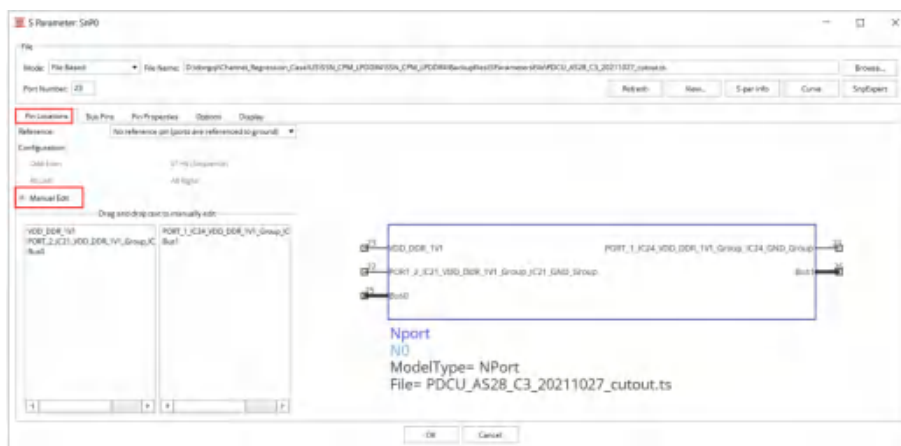
先对第一组 Bus 进行相关操作，后续可通过连通性填充其余 Bus。Bus 内 Signal Name 填写第一个，右击弹出菜单栏选择 Auto Fill signal，自动填充后面的 Signal Name。

Pin Index	Pin Name	Signal Name
3	IC21.A5.LPDDR4_DQ0	Signal1
4	IC21.A6.LPDDR4_DQ1	Signal2
8	IC21.B5.LPDDR4_DQ2	Signal3
9	IC21.C2.LPDDR4_DQ3	Signal4
7	IC21.B4.LPDDR4_DQ4	Signal5
10	IC21.C3.LPDDR4_DQ5	Signal6
1	IC21.A2.LPDDR4_DQ6	Signal7
2	IC21.A4.LPDDR4_DQ7	Signal8
6	IC21.B2.LPDDR4_DQ50_P	Signal9
5	IC21.B1.LPDDR4_DQ50_N	Signal10

只需要设置好第一个 Bus，在 Bus Name 列右击当前配置好的 Bus，弹出菜单栏，选择 Apply Signal Name to other Bus。内置算法自动判断 S 参数连通性，其余 Bus Signal Name 自动填充。所有 Bus Net 关系自动配好。

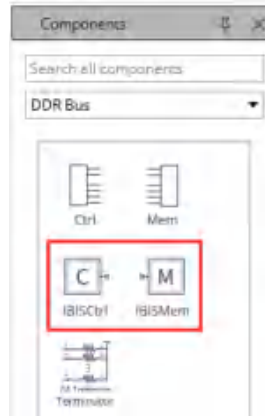


电源 Pin 不需要组 Bus，在 S 参数 Pin Locations Manual Edit，手动拖动想放置的位置。配好后点击 OK，Symbol 产生。

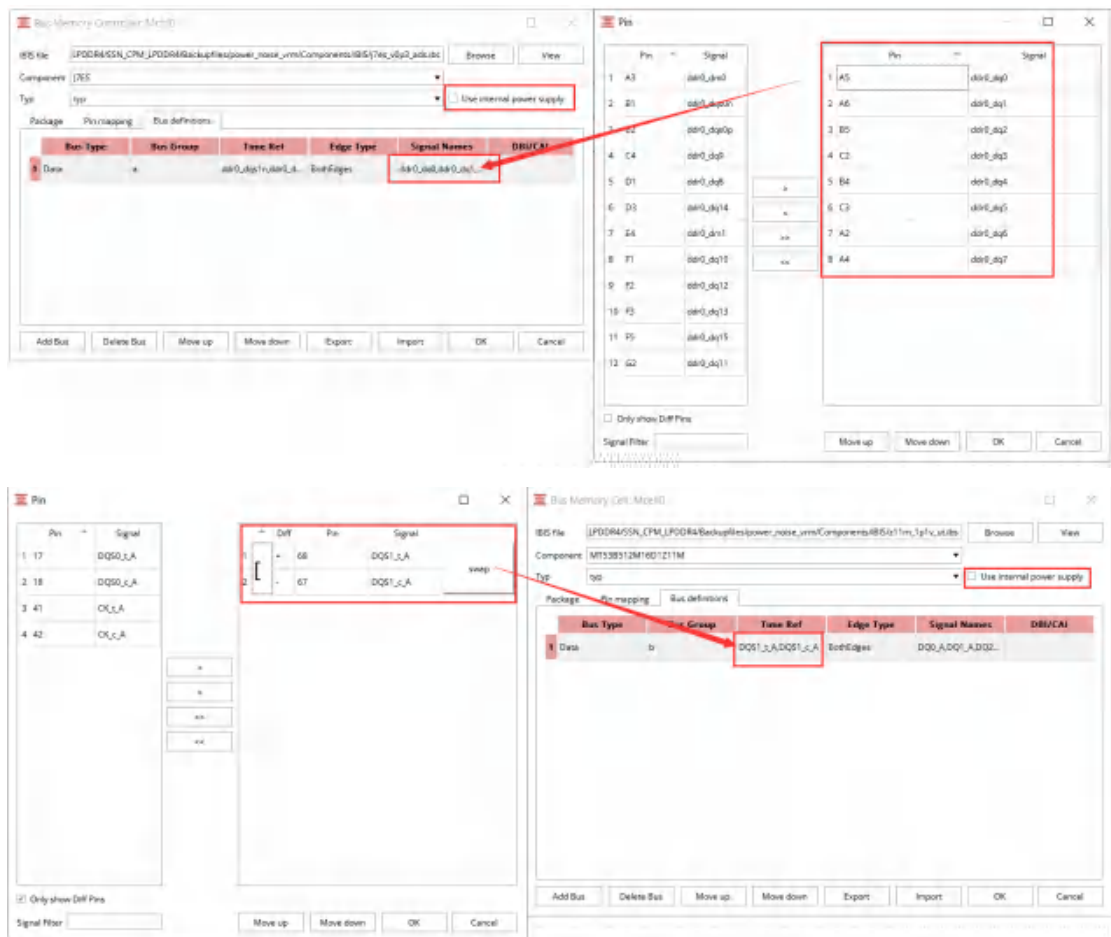


➤ 配置 DDR 控制器与内存

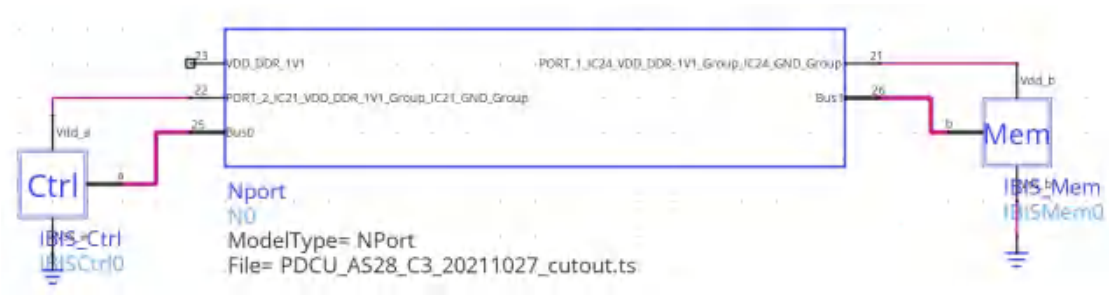
打开 Components 下 DDR Bus 器件库，加入 IBISCtrl IBISMem 器件。



双击 IBISCtrl 和 IBISMem 器件，导入 IBIS 文件，在 Bus definitions 双击 Bus Type 选择 Data，输入 Bus Group。双击 Time Ref 选择时钟 pin，双击 Signal Names 选择数据 pin。双击 Edge Type 选择 BothEdges。去掉 Use internal power supply 的勾选，使用外部电源。

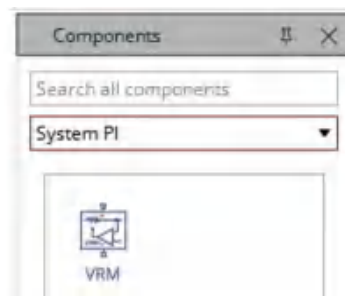


Ctrl 和 Mem Vdd 端口接通 Vdd pin，电路如下：

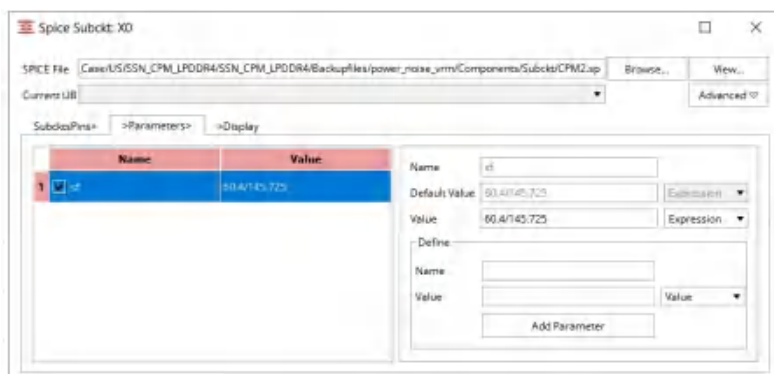
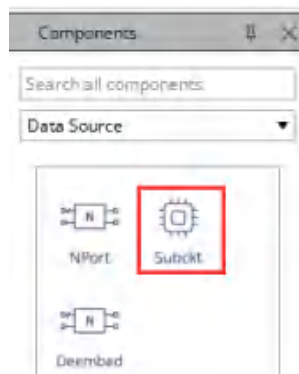


➤ 配置 VRM

打开 Components 下 System PI 器件库，加入 VRM 器件，外部供电。电压 Vdc 保持和 IBIS 一致。



打开 Components 下 Data Source 器件库，选择 Subckt，芯片电源模型 CPM 以子电路形式，即 PWL 电流模型加载进 Subckt。



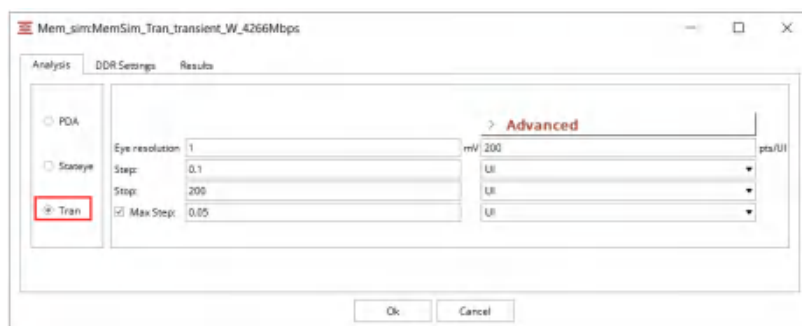
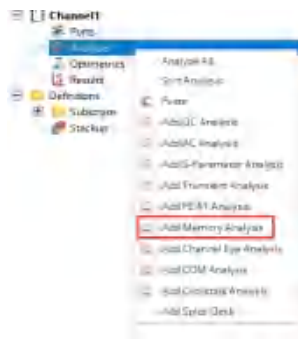
```
* CRM Port Name | Average current (A) |
* BAR_0_0_VDD 145.725 366.652 0.76
* Average power = 110.751 W.
.param sf='60.4/145.725'
Icoursig1 BAR_0_0_VDD BAR_0_0_VSS pw1
+ 0.000000ps '126.57033*sf'
+ 10.000000ps '180.99627*sf'
+ 20.000000ps '169.48512*sf'
+ 29.999999ps '198.25139*sf'
+ 40.000000ps '235.73247*sf'
+ 50.000001ps '276.55052*sf'
+ 59.999998ps '309.80646*sf'
+ 69.999999ps '328.90367*sf'
+ 80.000000ps '332.56399*sf'
+ 90.000001ps '326.85733*sf'
+ 100.000001ps '319.53071*sf'
```

完整电路如下，在需要观测波形的单 Pin 处双击连线增加 Net 节点。Bus 内部的观测节点双击 Bus 粗线可以进行勾选。



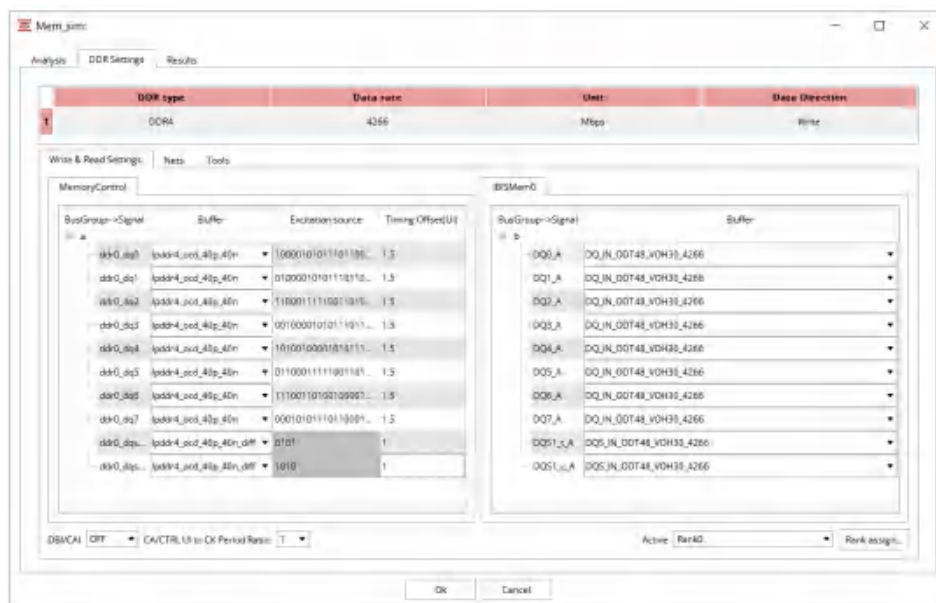
6.3.5.2 创建 Tran 仿真

树节点右击 Analysis 弹出菜单栏，选择 Add Memory Analysis。弹出 Mem sim 窗口，勾选 Tran 仿真。

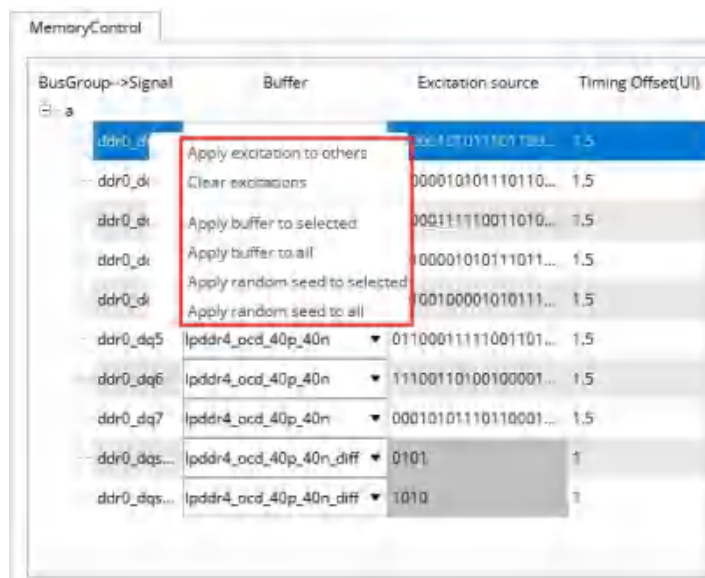


DDR Settings 设置 DDR 速率，读写方向。

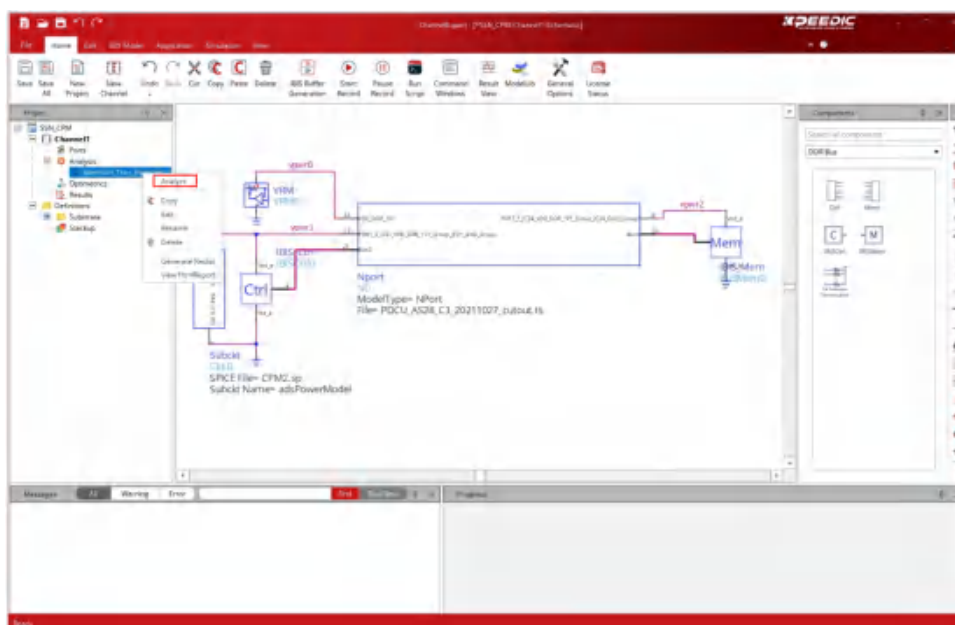
配置 Ctrl 和 Mem 端需要用到的 Buffer



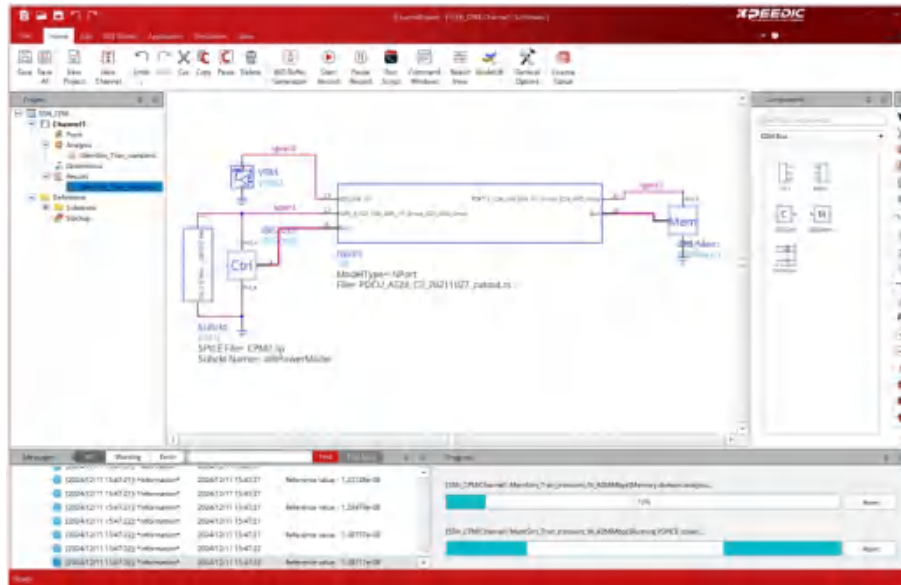
支持批量配 Buffer 和码型，右击当前选中 Signal，弹出菜单栏，应用到全部。



全部配置好，右击树节点增加的 MemoryAnalysis，点击 Analyze，进入仿真。

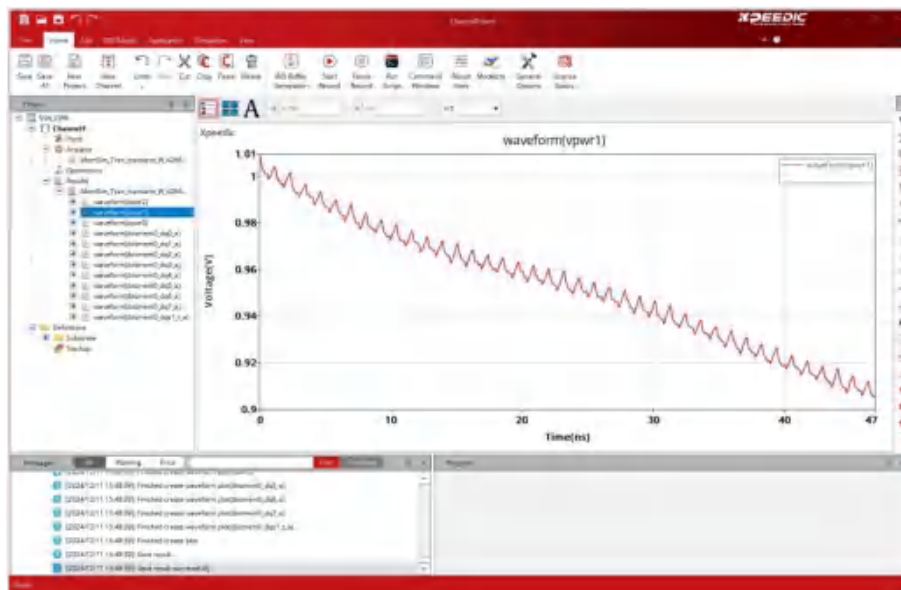


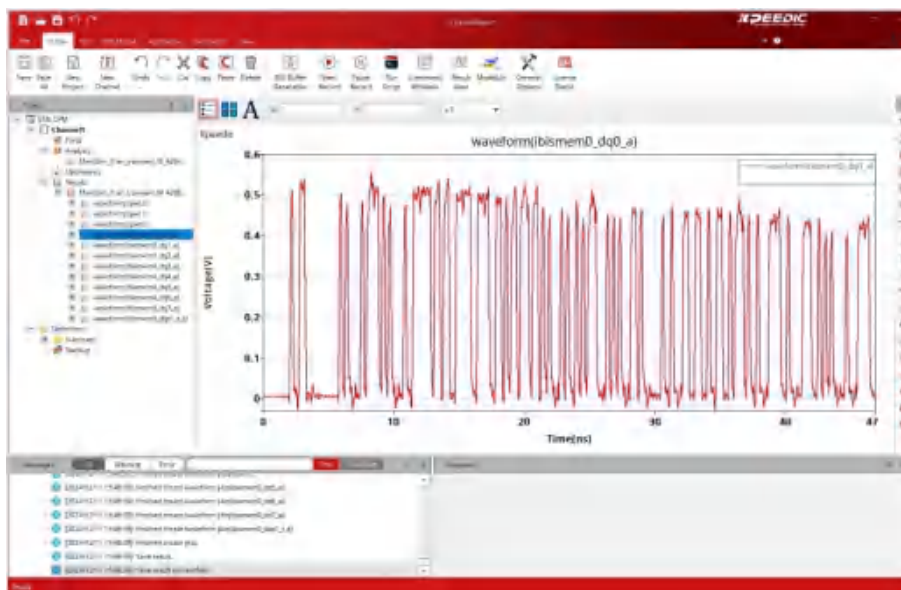
XSPICE Solver 解析中。



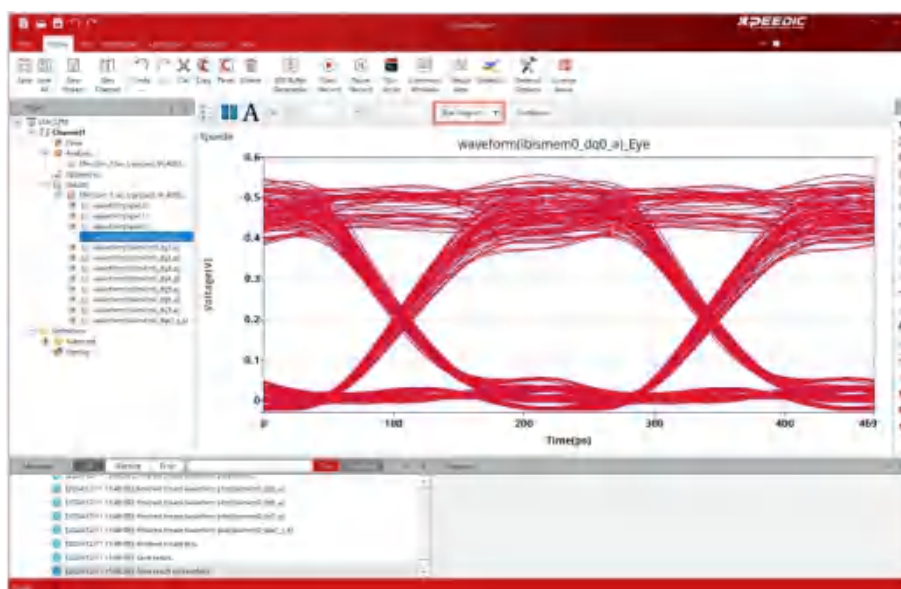
6.3.5.3 瞬态波形和眼图显示

仿真结束，tran 波形挂在树节点下，点击浏览。

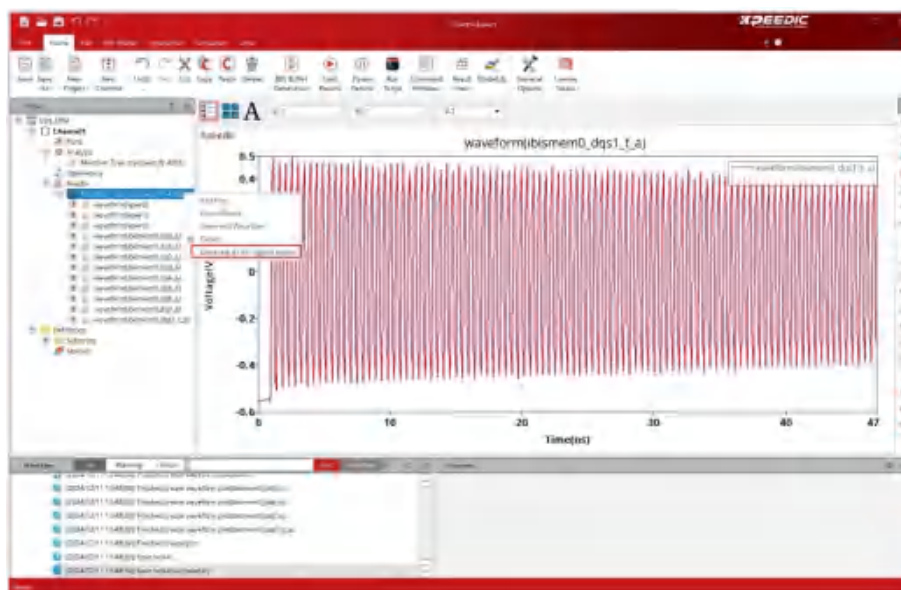




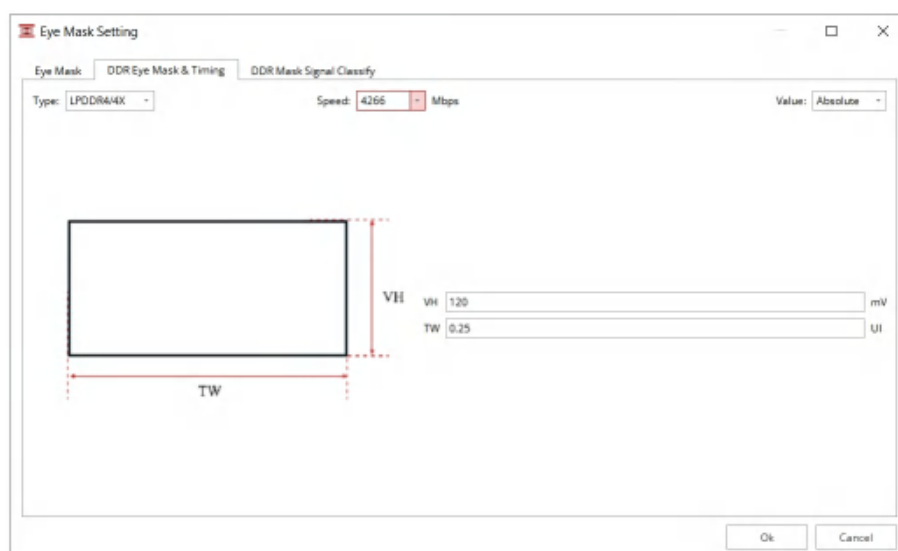
对 DQ 数据波形叠眼图，波形 PLOT 界面 VT 转 Eye Diagram。



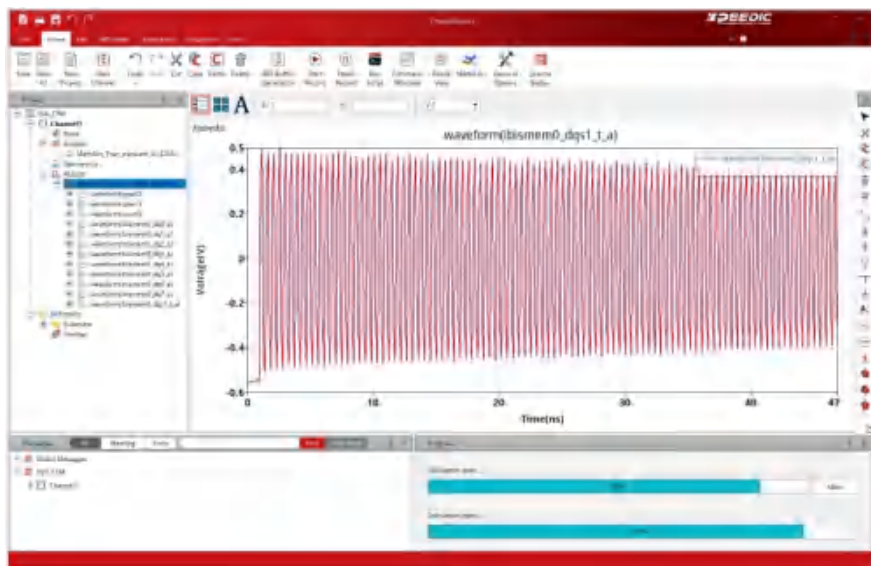
批量测量眼图数据，右击树节点 Results 下的仿真节点，选择 Generate JEDEC-Signoff Report。



弹出眼图模板选择对应的协议和速率。



点击 OK，进度条进入 Caculation Jedec。

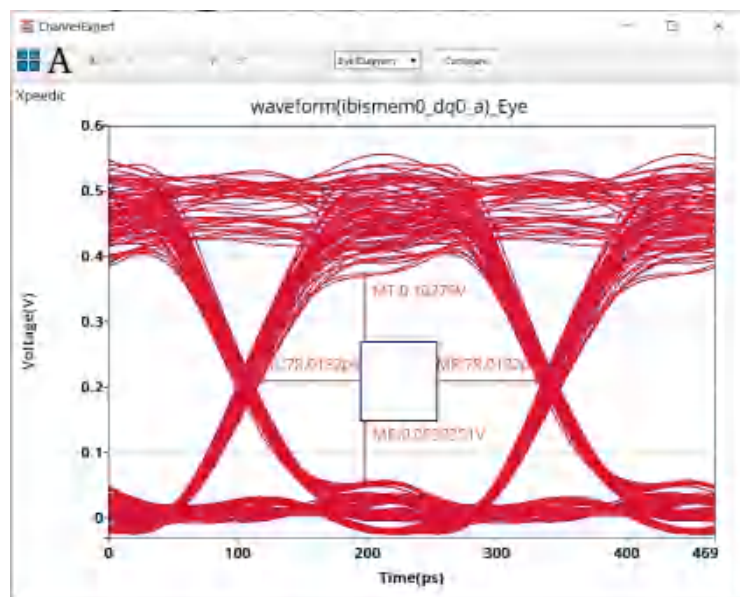


眼宽眼高 table 表生成。

The screenshot shows the 'Xpedic Report' window. At the top, there is a diagram of a signal waveform. Below the diagram is a table with 12 rows of data. The table columns are: Signal, Value, Mask Height, Mask Width, Eye Height, Eye Width, Margin, Pass/Fail, An group setup, An hold setup, An group delay, and Hold time (ps).

Signal	Value	Mask Height	Mask Width	Eye Height	Eye Width	Margin	Pass/Fail	An group setup	An hold setup	An group delay	Hold time (ps)
1 waveform(qs1)	1.00233	0.12	0.25	0.00047222	34.5351	-100.026	Fail	nan	nan	nan	nan
2 waveform(qs1)	0.905748	0.12	0.25	0.00070118	35.0668	-100.011	Fail	nan	nan	nan	nan
3 waveform(qs2)	1.12794	0.12	0.25	0.00030053	63.7816	-100.088	Fail	nan	nan	nan	nan
4 waveform(bismem0_dqs1_a)	0.203760	0.12	0.25	-0.116136	214.639	155.558	Pass	nan	nan	nan	nan
5 waveform(bismem0_dqs1_a)	0.205917	0.12	0.25	-0.394025	216.832	143.011	Pass	nan	nan	nan	nan
6 waveform(bismem0_dqs1_a)	0.205768	0.12	0.25	-0.393827	215.829	131.288	Pass	nan	nan	nan	nan
7 waveform(bismem0_dqs1_a)	0.205985	0.12	0.25	-0.391708	214.747	125.505	Pass	nan	nan	nan	nan
8 waveform(bismem0_dqs1_a)	0.205522	0.12	0.25	-0.388495	215.082	141.344	Pass	nan	nan	nan	nan
9 waveform(bismem0_dqs1_a)	0.204418	0.12	0.25	-0.386335	215.389	120.018	Pass	nan	nan	nan	nan
10 waveform(bismem0_dqs1_a)	0.207819	0.12	0.25	-0.391732	215.337	137.355	Pass	nan	nan	nan	nan
11 waveform(bismem0_dqs1_a)	0.208881	0.12	0.25	-0.318876	214.216	136.518	Pass	nan	nan	nan	nan
12 waveform(bismem0_dqs1_a)	0.200331891	0.12	0.25	-0.390001	227.837	189.761	Pass	nan	nan	nan	nan

点击任意 DQ 信号，弹出眼图。

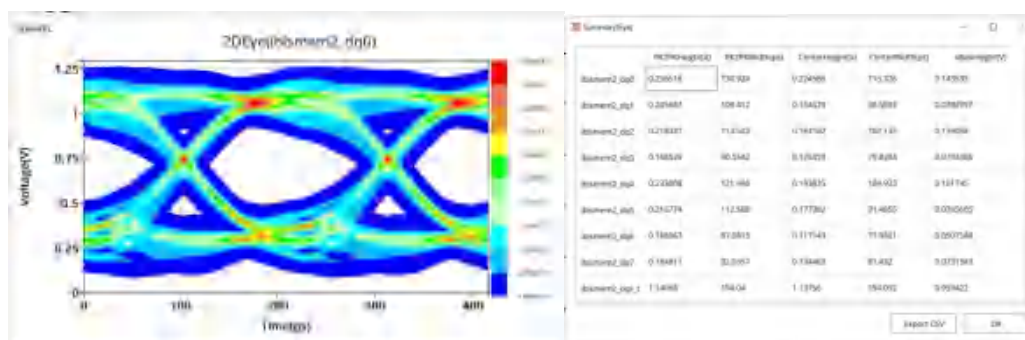


6.3.6 DDR 统计仿真新功能支持及测量方式

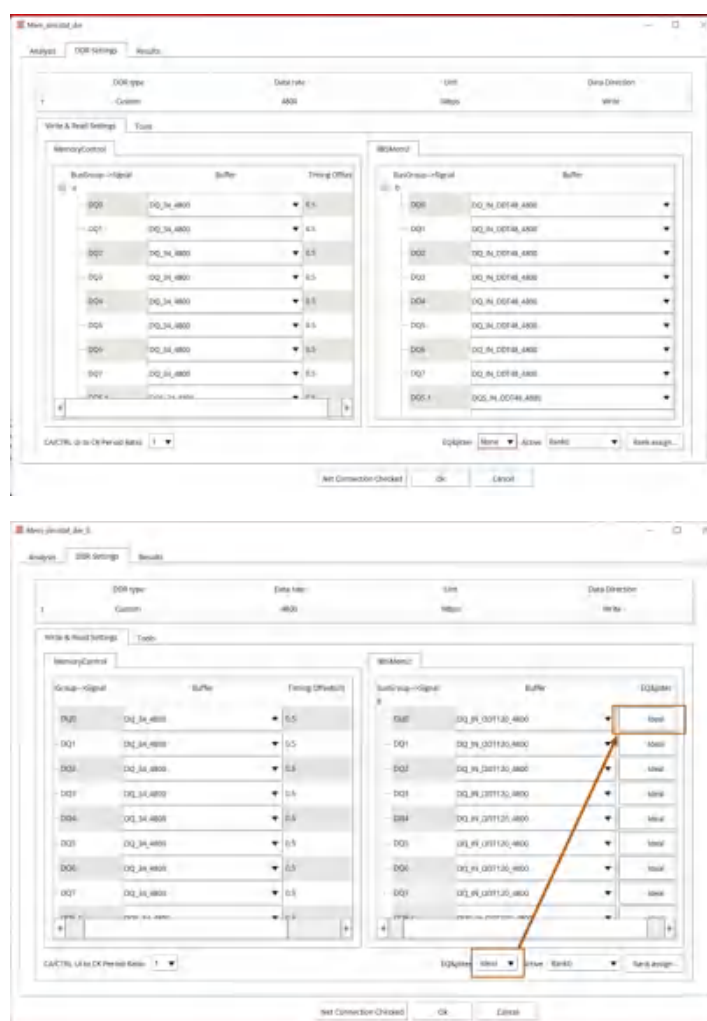
6.3.6.1 CTLE 寻优

使用 CTLE 曲线可以对眼图进行优化，Channel 使用优化方法如下。

若原始 DDR 的工程的眼图以及眼高眼宽信息如下：

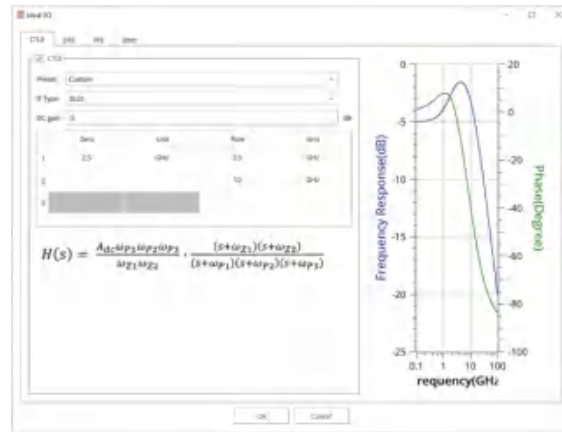


按照章节 6.3.4.3 节步骤操作，进入 DDR Settings 界面，EQ&Jitter 选项默认为 None，将其勾选为 Ideal，在 Buffer 后面，会弹出 Ideal 选项，点击 Ideal 按钮，进入 Ideal EQ 参数设置界面：

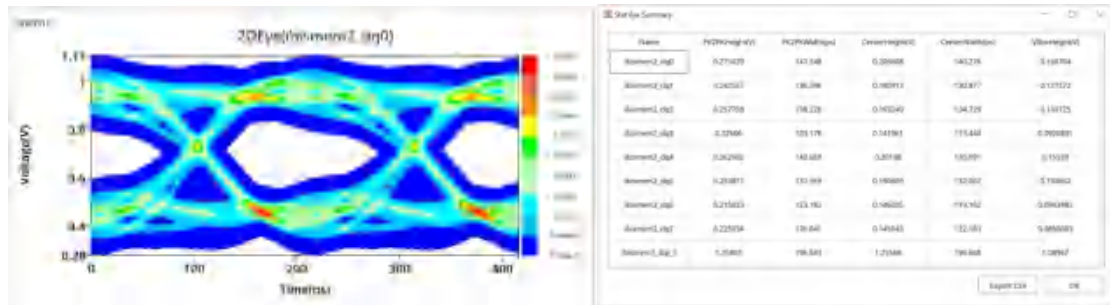


CTLE 优化曲线，可以选择软件内置的曲线优化，也可以由用户导入 CTLE 曲线文件：

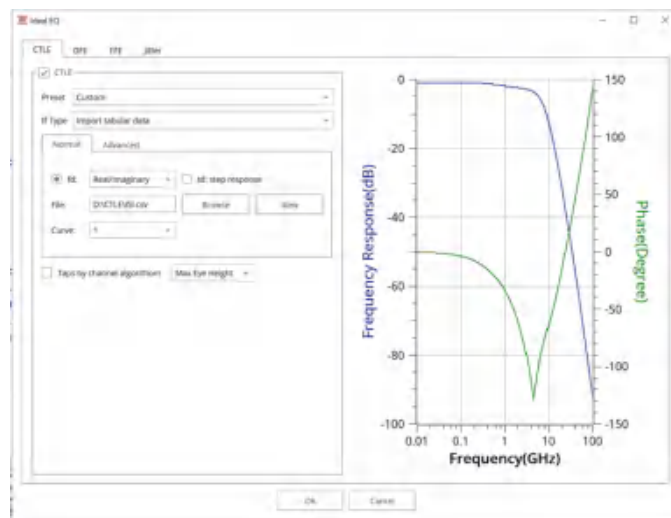
以内置优化曲线为例，tf type 选择 3p2z，点击 OK，启动仿真，



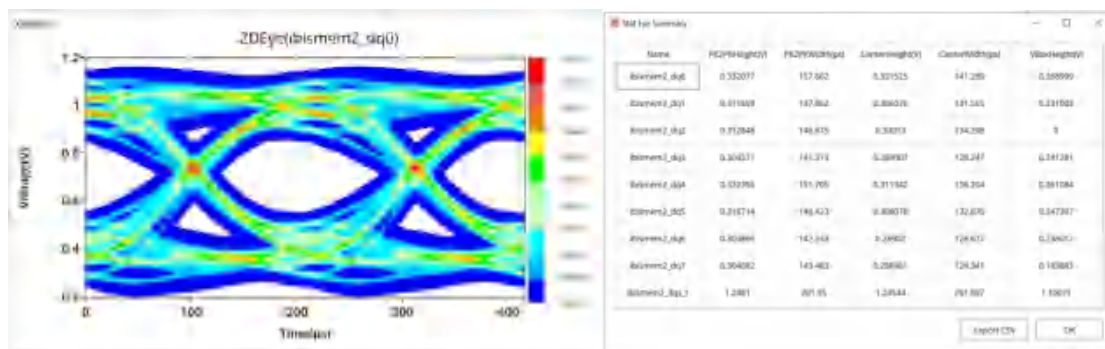
仿真后结果如图所示：



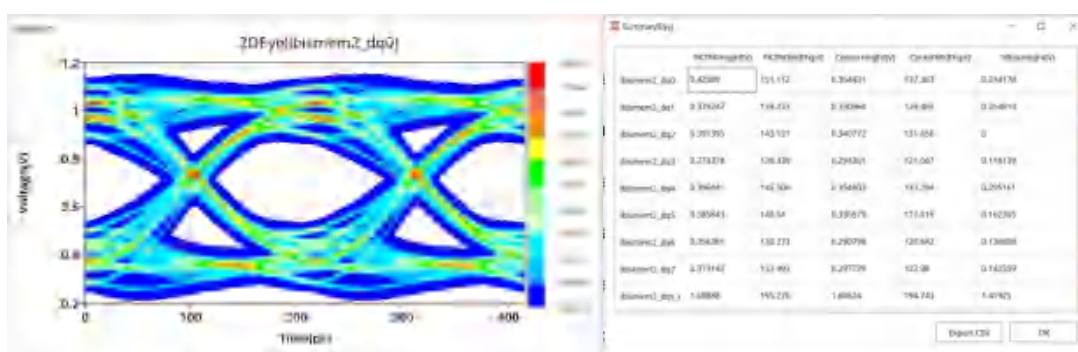
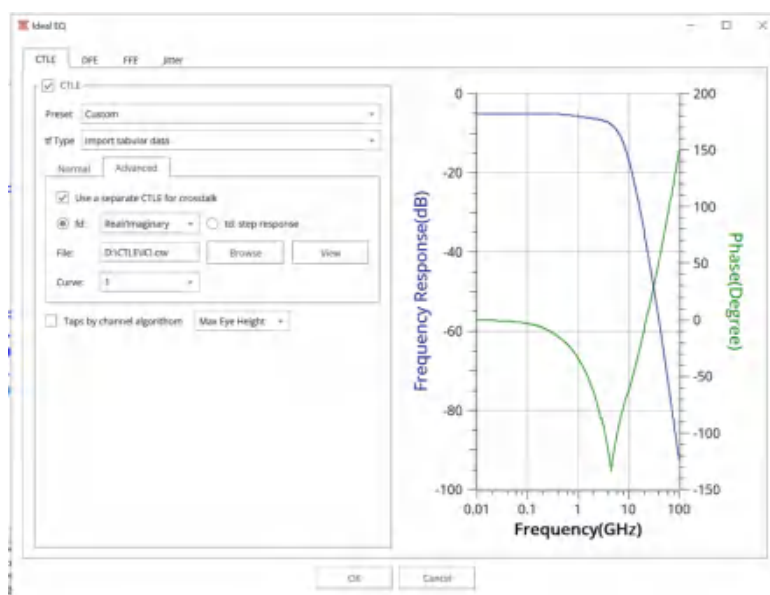
Tf Type 若选为 Import tabular data，则点击 Browse 按钮，导入 CTLE 曲线文件，选择曲线类型，点击 OK，进行仿真，仿真结果如下图所示：



运行结果如下图所示：



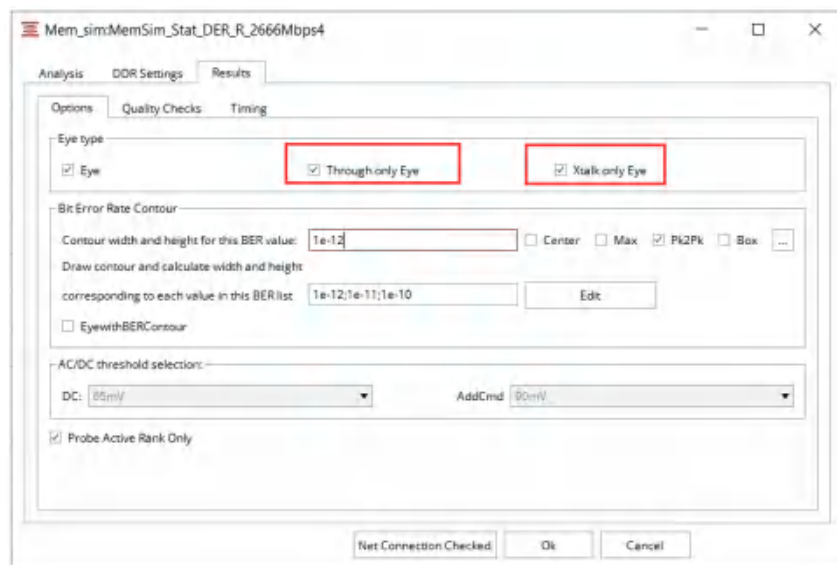
如果用户自己导入曲线优化该选项，Channel 提供了 ISI 与 ICI 曲线单独优化功能，即对于 ICI 信号，导入另外的曲线进行单独优化，该功能通过 Advanced 选项实现，勾选 Use a separate CTLE for crosstalk，导入 ICI 优化曲线，点击 OK，进行仿真，结果如下：



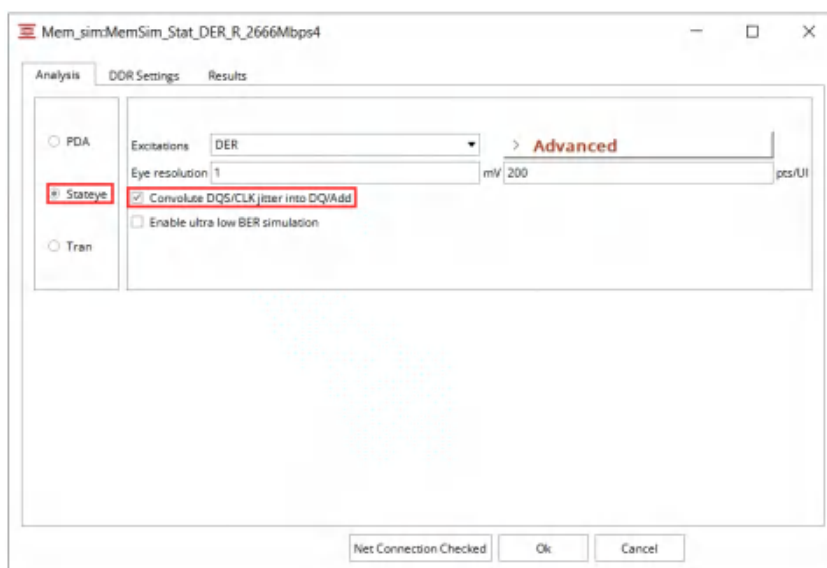
6.3.6.2 主通道与串扰通道拆开分析，引入时钟抖动

常规输出的眼图为主信号和串扰信号综合作用的结果，同时也支持将主信号和串扰信号眼图分开输出，能够更直观的反应出信号特性。

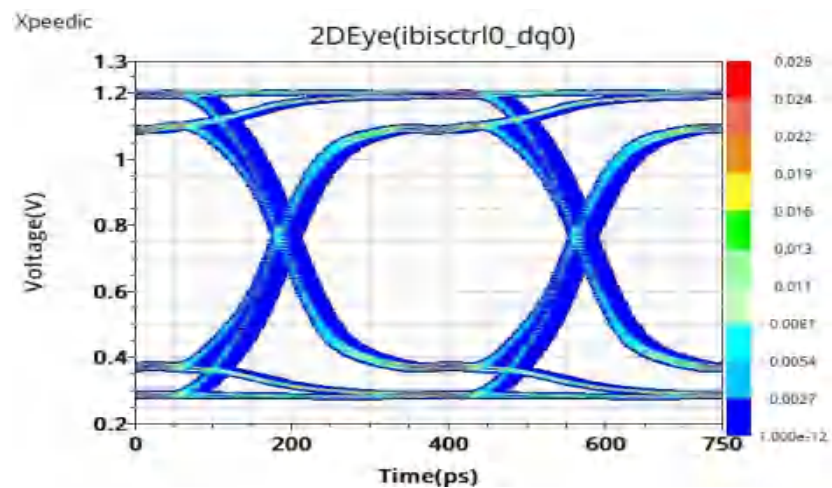
Memory 统计仿真编辑界面，切换到 Results。勾选 Eye 输出的是主信号和串扰信号综合作用下的眼图，勾选 Through only Eye 输出主信号眼图，勾选 Xtalk only Eye 输出串扰信号眼图。



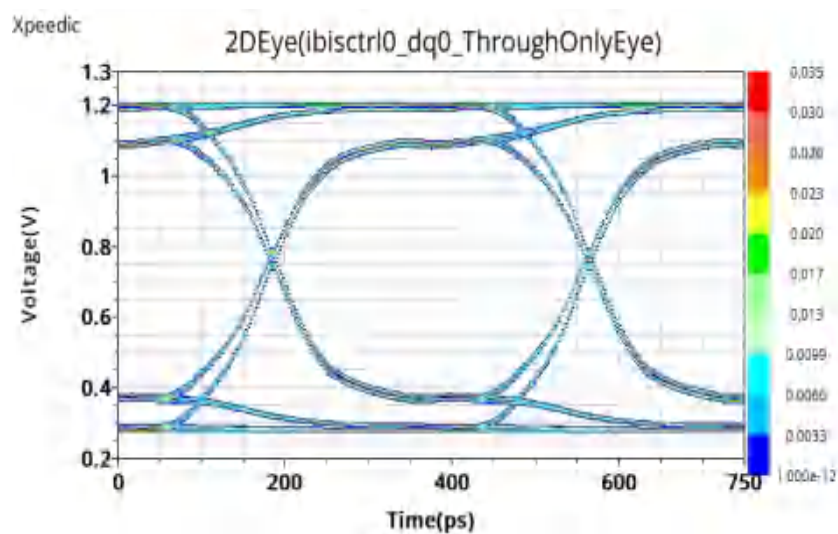
支持将时钟抖动考虑进信号分析，勾选 Analysis 下的 Convolute DQS/CLK jitter into DQ/Add。



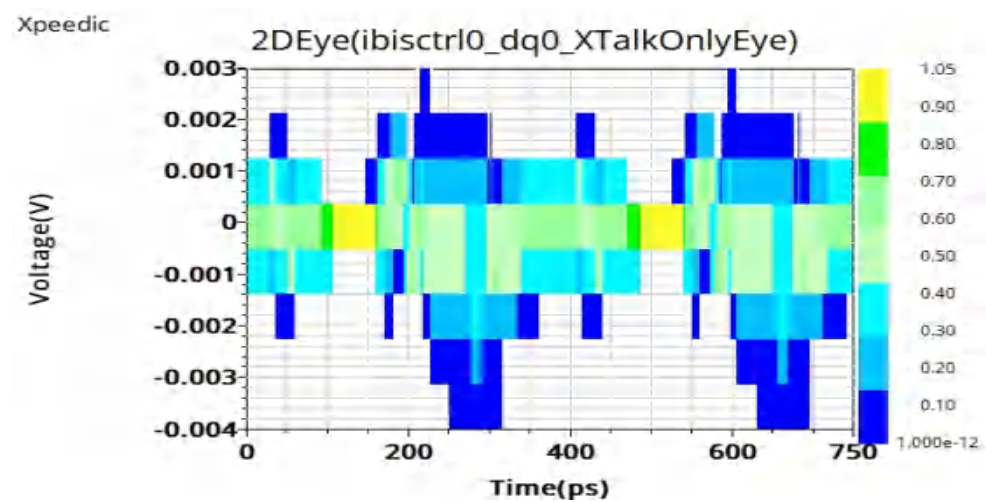
下图为加入时钟抖动的统计眼图



下图为只有主信号的统计眼图

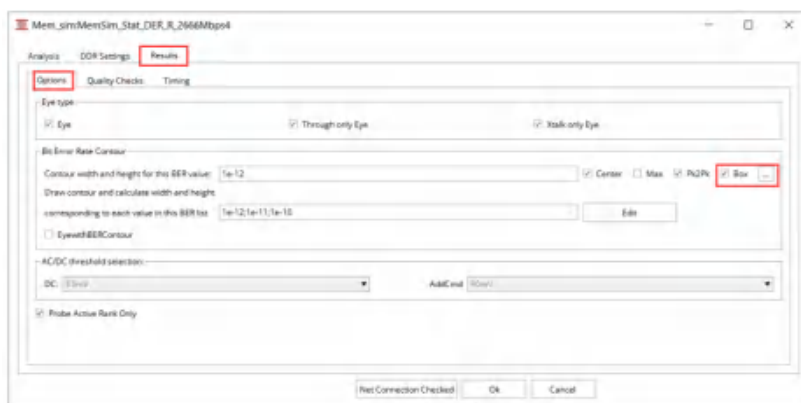


下图为只有串扰号的统计眼图



6.3.6.3 Hbox、Vbox 测量模式

Results 下面勾选 Box 则使能测量模式

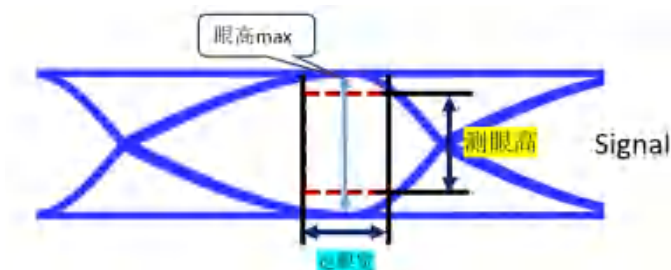


点击展开按钮，弹出 Box 参数设置界面

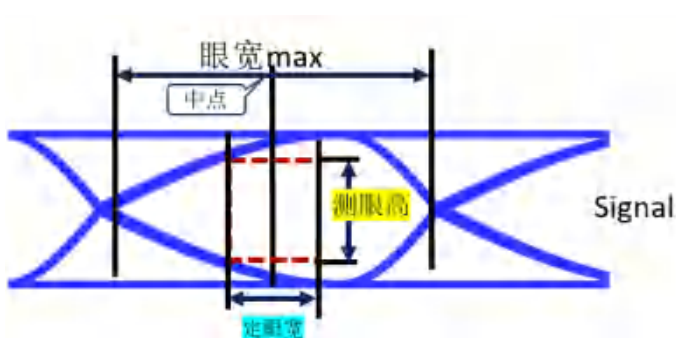


勾选 Height，测量方式为定宽测高

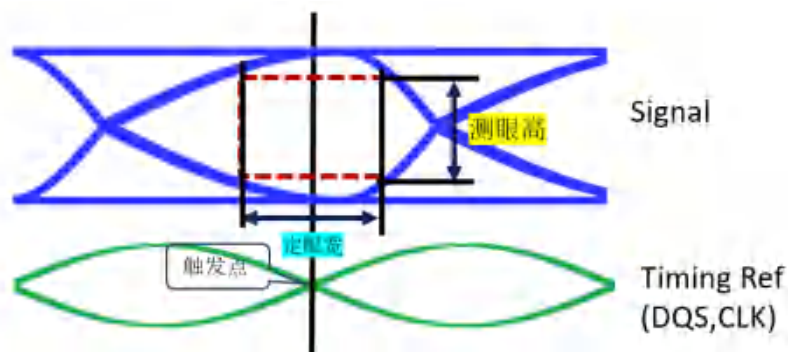
Max eye vertical point 以眼高最大对应的水平位置为基准



Center of max eye width 以眼宽最大处的中点为基准

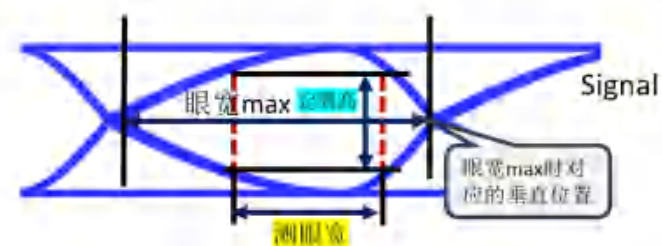


Trigger of time reference 以 timing ref(DQS,CLK)触发点为基准

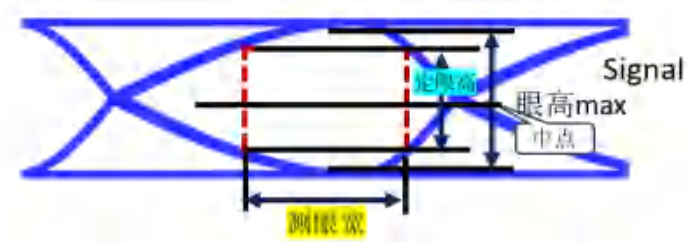


勾选 Width，测量方式为定高测宽

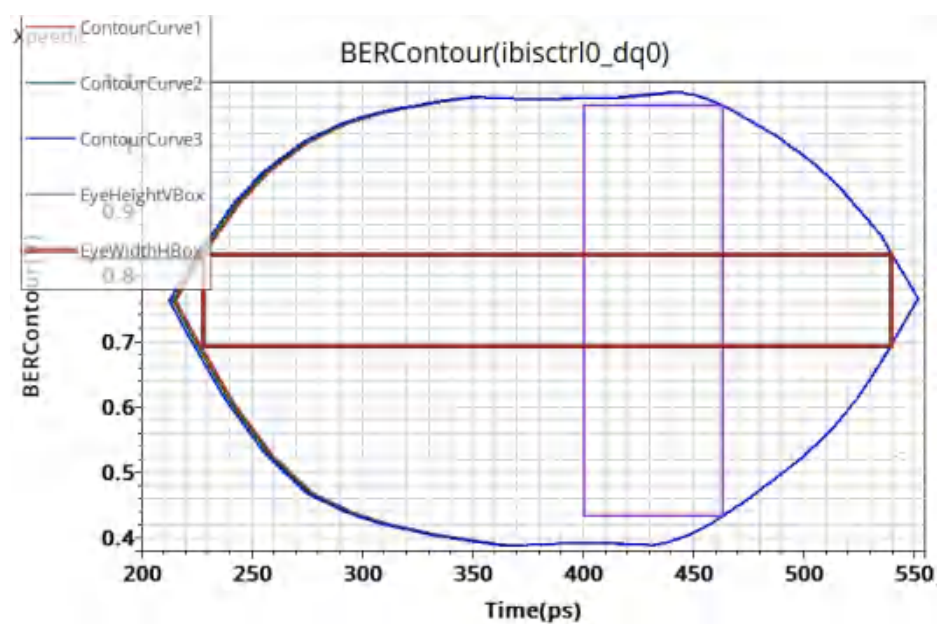
Max Eye horizontal point 以眼宽最大处对应的垂直位置为基准



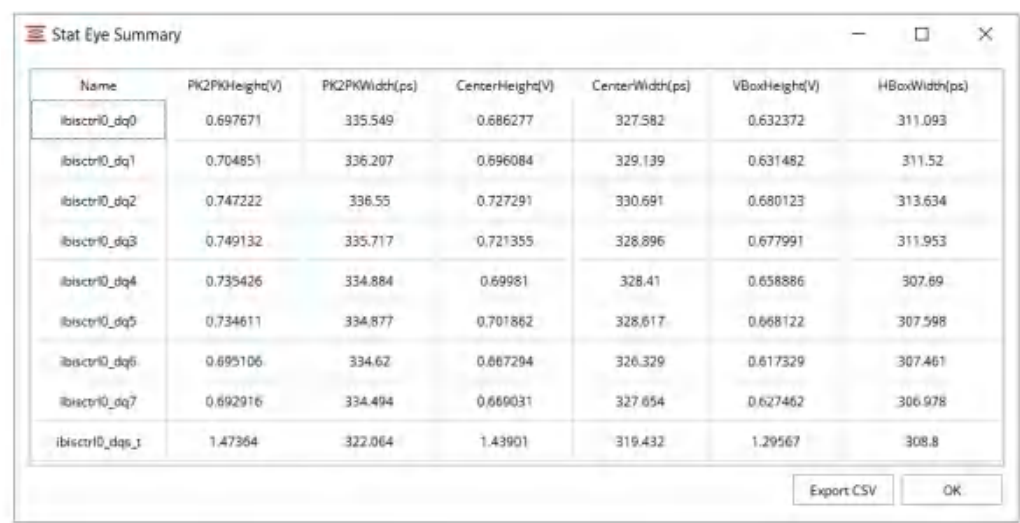
Center of max eye Height 以眼高最大处的中点为基准



Output Box 使能，Box 输出到统计眼图的 BerContour 中



VBoxHeight 定宽测高、HBoxWidth 定高测宽的数据通过 Summary 整体输出

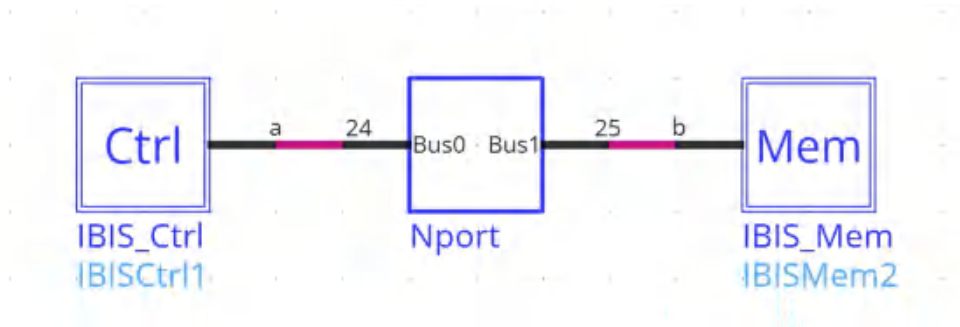


Name	PK2PKHeight(V)	PK2PKWidth(ps)	CenterHeight(V)	CenterWidth(ps)	VBoxHeight(V)	HBoxWidth(ps)
IbisCtrl0_dq0	0.697671	335.549	0.686277	327.582	0.632372	311.093
IbisCtrl0_dq1	0.704851	336.207	0.696084	329.139	0.631482	311.52
IbisCtrl0_dq2	0.747222	336.55	0.727291	330.691	0.680123	313.634
IbisCtrl0_dq3	0.749132	335.717	0.721355	328.896	0.677991	311.953
IbisCtrl0_dq4	0.735426	334.884	0.69981	328.41	0.658886	307.69
IbisCtrl0_dq5	0.734611	334.877	0.701862	328.617	0.668122	307.598
IbisCtrl0_dq6	0.695106	334.62	0.667294	326.329	0.617329	307.461
IbisCtrl0_dq7	0.692916	334.494	0.669031	327.654	0.627462	306.978
IbisCtrl0_dqs_1	1.47364	322.064	1.43901	319.432	1.29567	308.8

Export CSV OK

6.3.7 DDR buffer 扫描

原始工程如下：



原始的 Buffer 数据设置如下：

Mem_simstat_der

Analysis

DDR Settings

Results

	DDR type	Data rate	Unit	Data Direction
1	Custom	4800	Mbps	Write

Write & Read Settings

Tools

MemoryControl

BusGroup->Signal	Buffer	Timing Offset(UI)
a		
DQ0	DQ_34_4800	▼ 0.5
DQ1	DQ_34_4800	▼ 0.5
DQ2	DQ_34_4800	▼ 0.5
DQ3	DQ_34_4800	▼ 0.5
DQ4	DQ_34_4800	▼ 0.5
DQ5	DQ_34_4800	▼ 0.5
DQ6	DQ_34_4800	▼ 0.5
DQ7	DQ_34_4800	▼ 0.5
DQS_1	DQS_34_4800	▼ 0.5

IBISMem2

BusGroup->Signal	Buffer
b	
DQ0	DQ_IN_COT120_4800 ▼
DQ1	DQ_IN_COT120_4800 ▼
DQ2	DQ_IN_COT120_4800 ▼
DQ3	DQ_IN_COT120_4800 ▼
DQ4	DQ_IN_COT120_4800 ▼
DQ5	DQ_IN_COT120_4800 ▼
DQ6	DQ_IN_COT120_4800 ▼
DQ7	DQ_IN_COT120_4800 ▼
DQS_1	DQS_IN_COT120_4800 ▼

CA/CTRL UI to CK Period Ratio: 1 ▼

EQ&Iter

None ▼

Active

Rank0 ▼

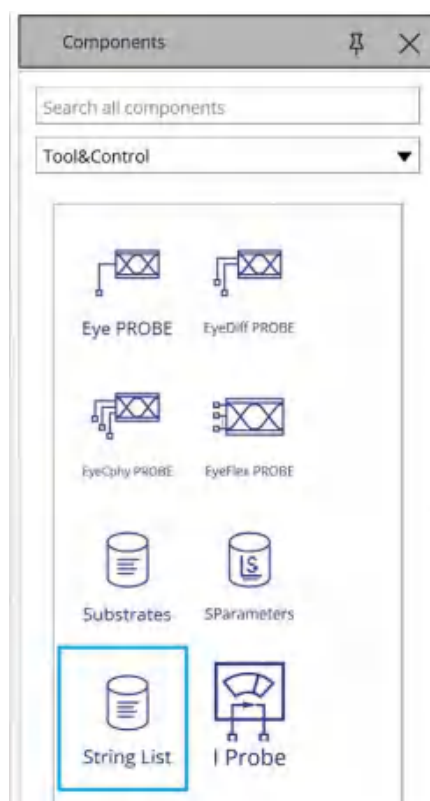
Rank assign...

Net Connection Checked

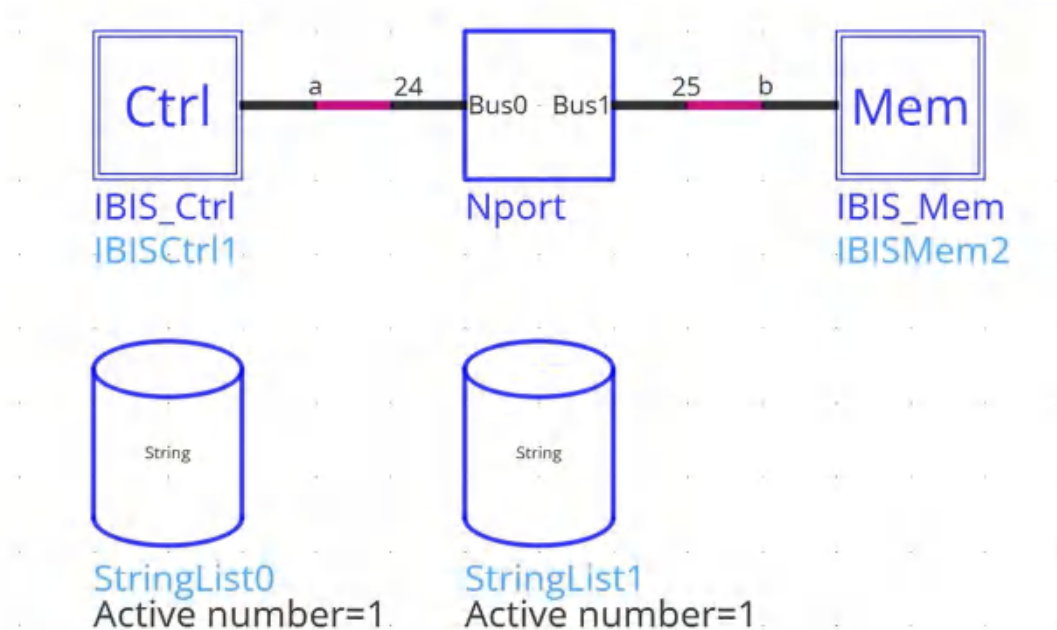
OK

Cancel

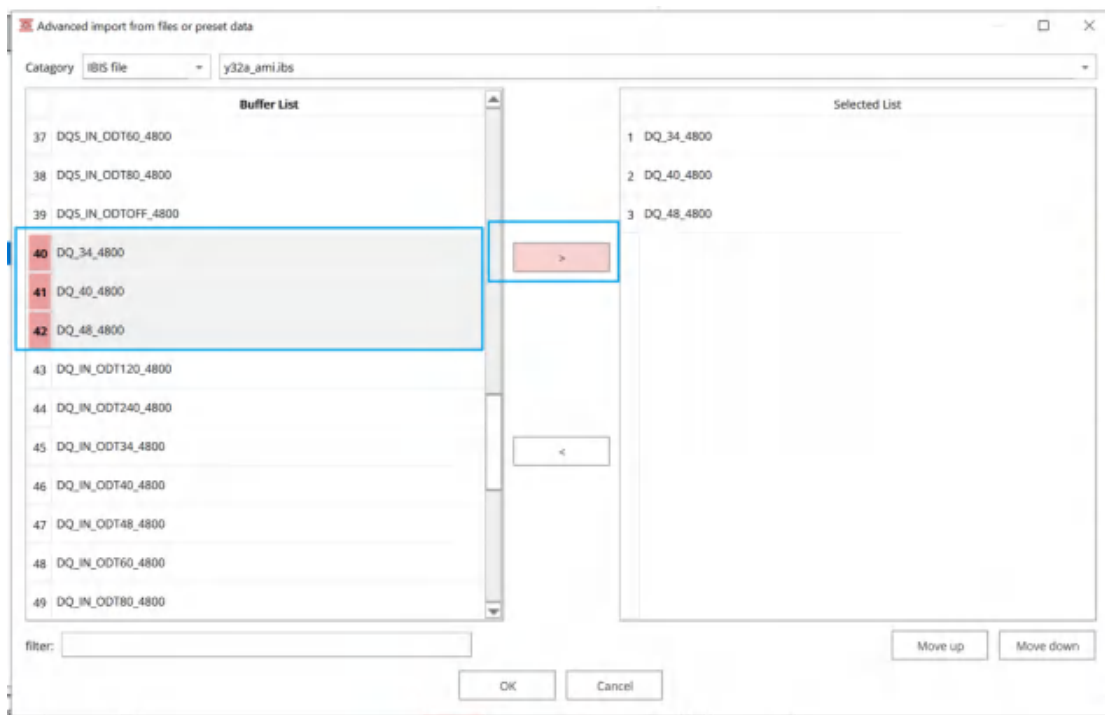
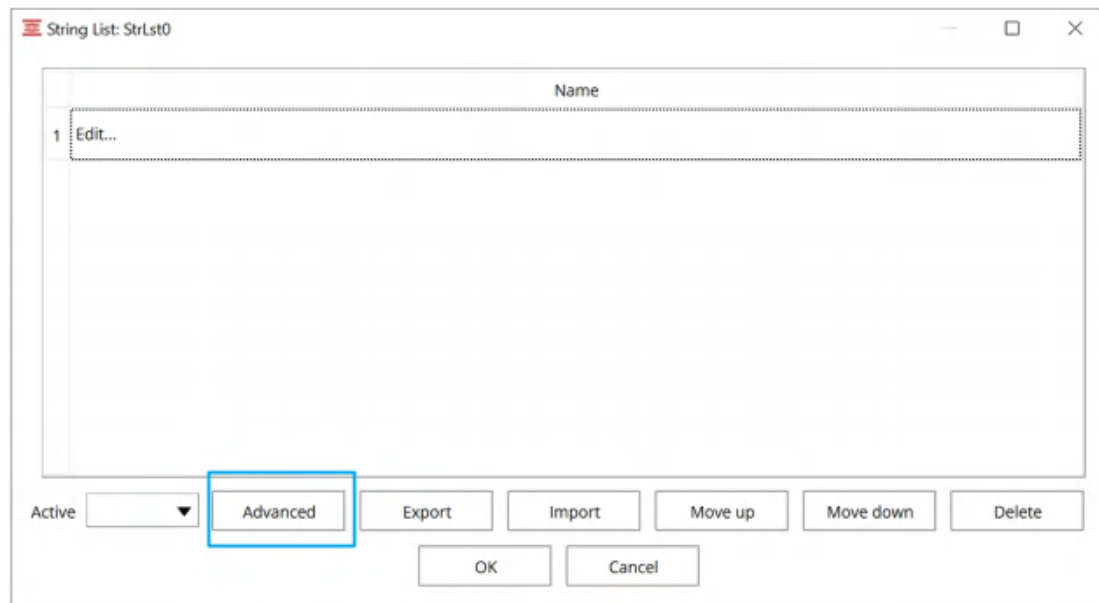
在 Components 界面处选择 Tool&Control，选择 String List，将两个 String List 拖入原理图中：

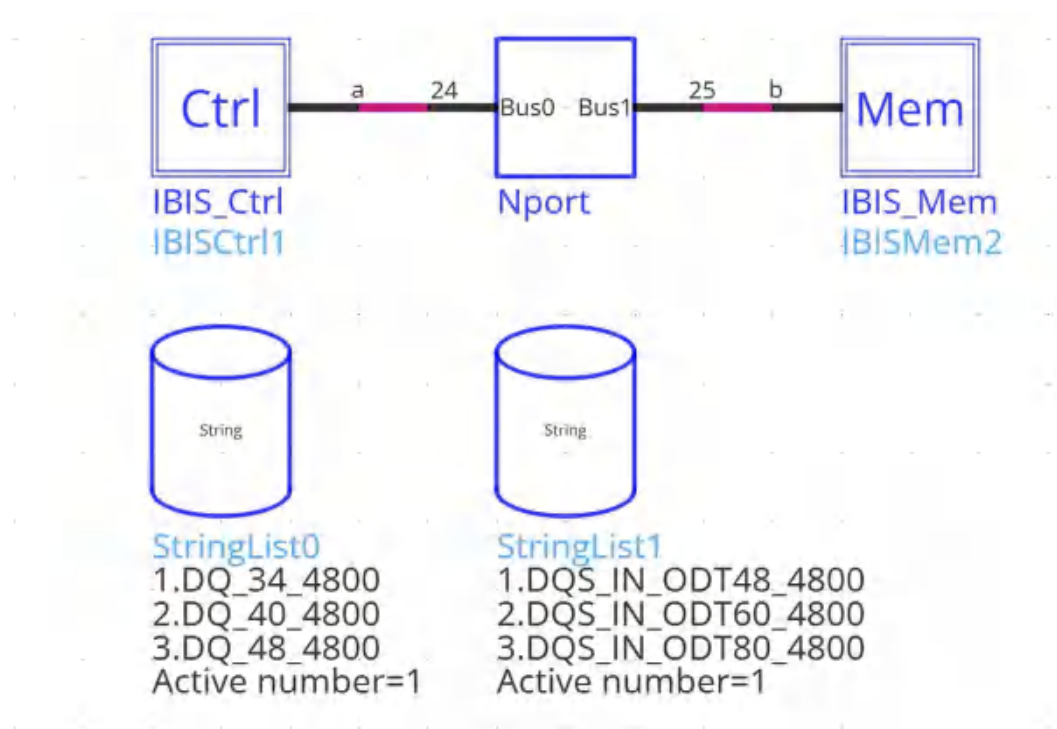


选择完成后界面如下：

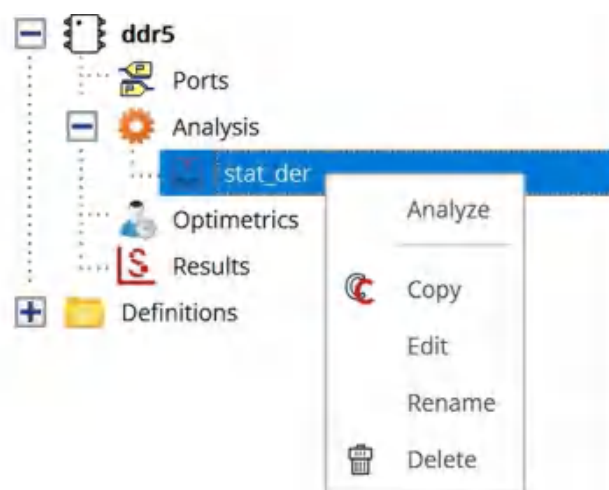


双击 StringList0 器件，点击 Advanced，选择需要扫描的 Buffer，点击向右的按钮，将其选择到 Selected List，点击 OK；StringList1 器件进行相同的操作，操作完成后效果图如下图所示：

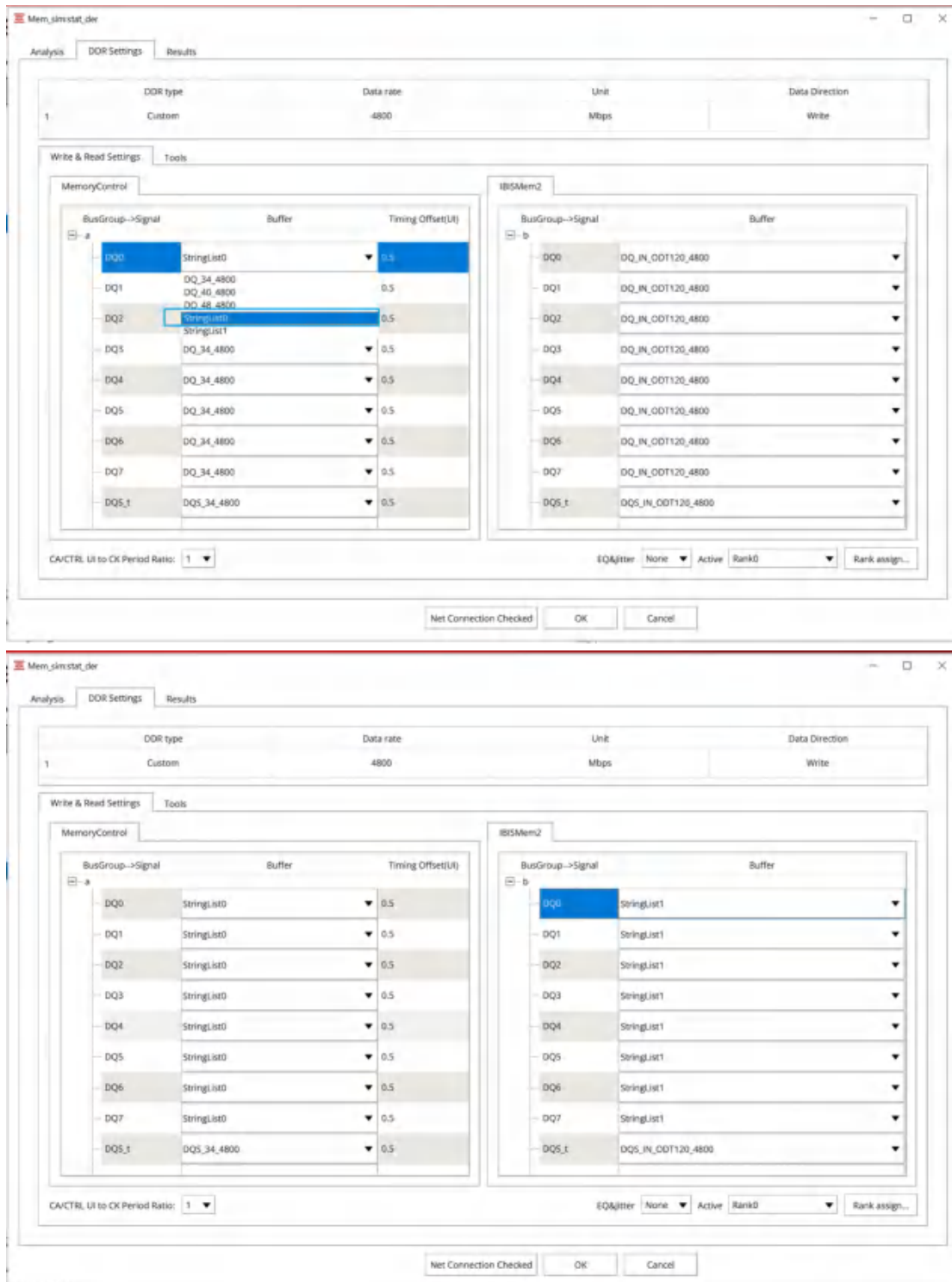




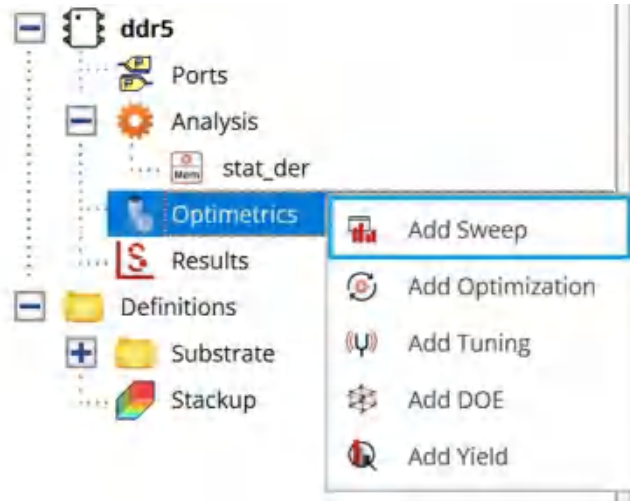
树节点下右键仿真节点，点击 Edit，



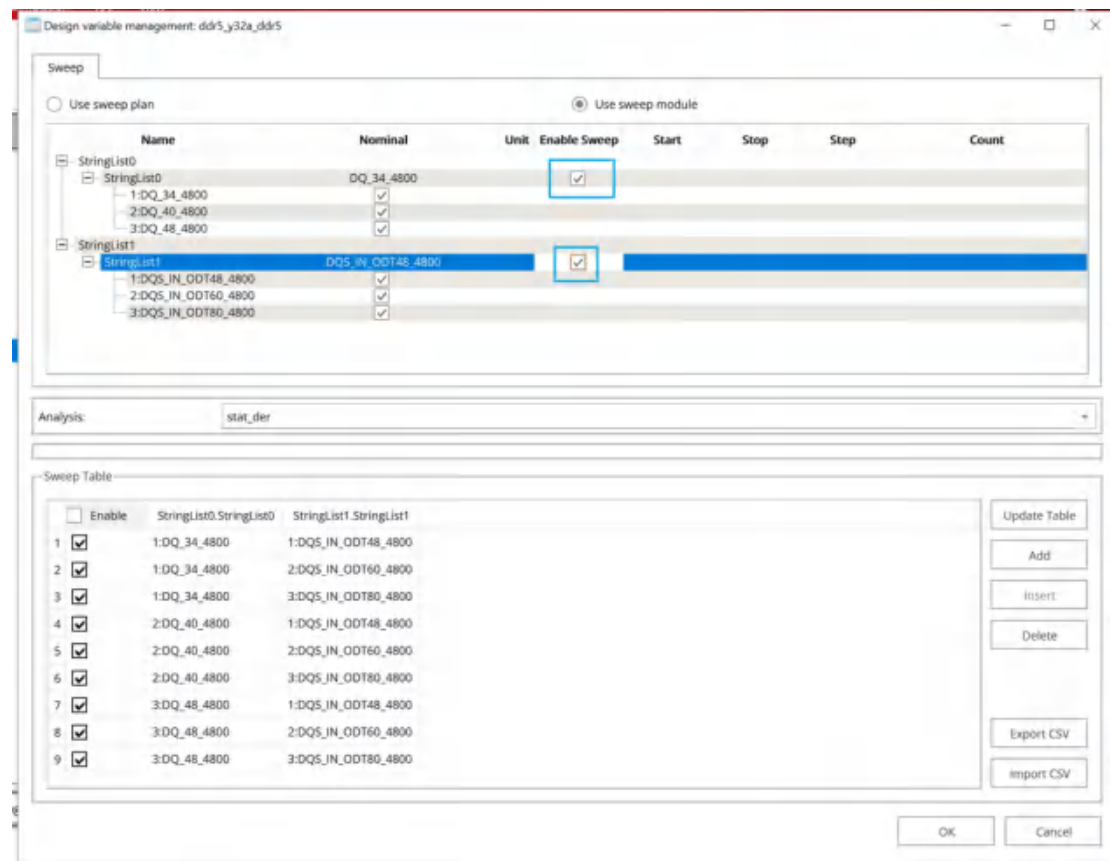
点击 DDR Settings，进入 buffer 选择界面，将 Ctrl 端的 • 的 Buffr 改为 StringList0，将 Memory 端的 buffer 改为 StringList1:



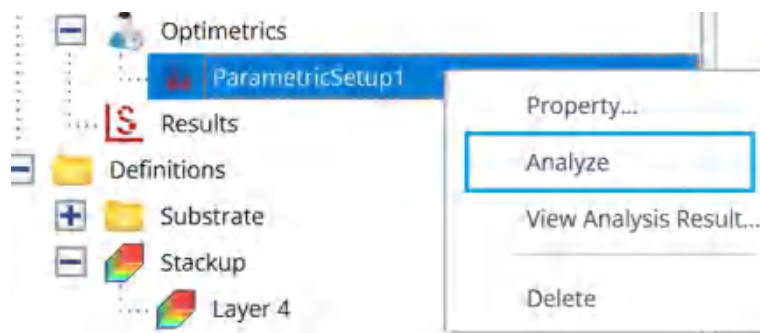
右键点击 Optimetrics，选择 Add Sweep，：



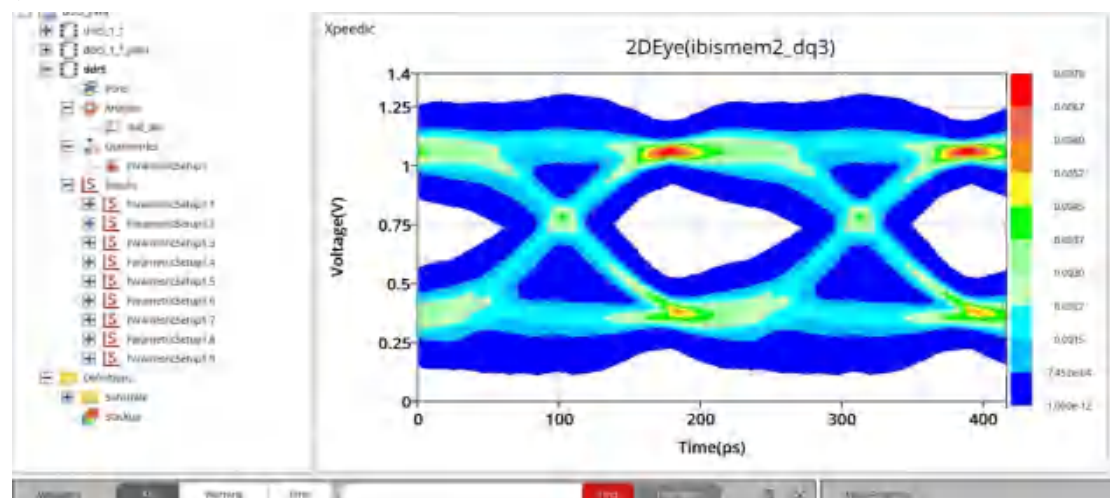
在弹出的界面中将 Enable Sweep 勾选，



右键点击 ParametricSetup1，点击 Analysis 开始仿真：

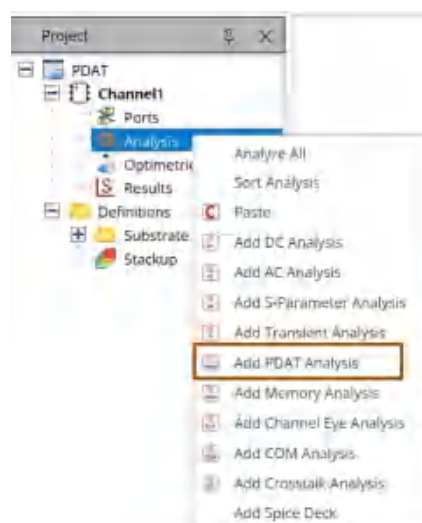


仿真结束后，各个 buffer 的仿真结果会在 Results 节点下显示，本例共扫描 9 组，因此树节点下显示 9 组结果：

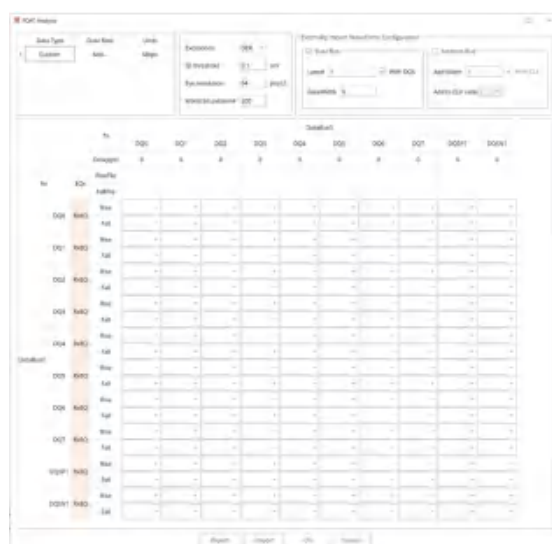


6.4 用 PDAT 分析工具分析 DDR 最坏情况

6.4.1 设置 PDAT 仿真参数



树节点下右键点击 Analysis，点击 Add PDAT Analysis，弹出仿真设置界面：



该界面包括四部分，首先设置 Data Type 和 Data Rate 部分，可以点击下拉框选择软件提供的 type 和 Rate 或通过下拉框中 Custom 选项由用户自己输入；

设置仿真参数部分，Excitations 下拉框可以选择激励方式，默认为 DER 方式，设置 ISI threshold 码间串扰阈值、Eye resolution 眼图分辨率、Worst bit pattern 最坏码型数，默认为 0.1mv、64pts/UI、200；

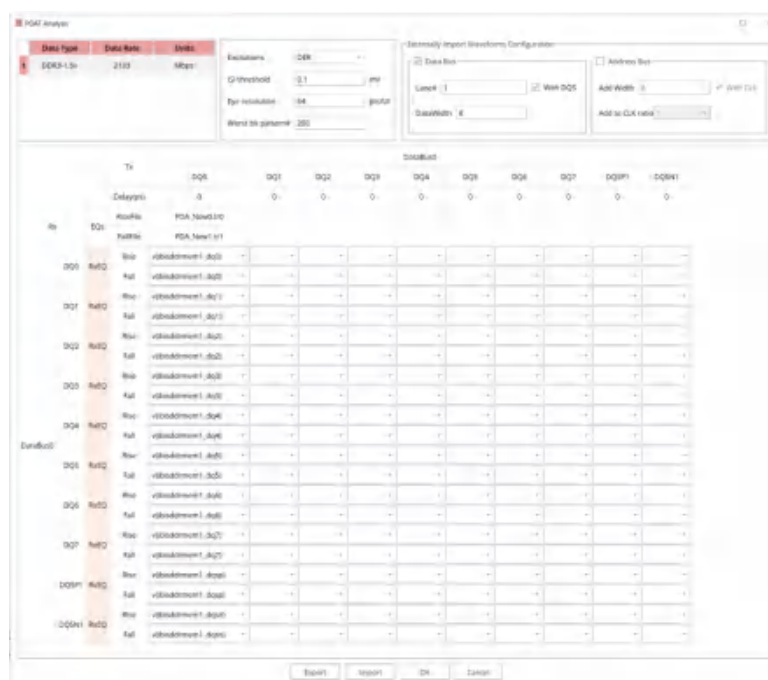
Externally Import Waveform Configuration 部分选择是对 Data Bus(数据线)、Address Bus(地址线)分析，Data Width/Add Width 为 Bus 线路数，勾选 With DQS/With CLK 为是否添加时钟信号，Lane/Add to CLK ratio 为时钟信号路数，可通过下拉框选择，也可由用户自己输入；

DataBus0 表格为 Tr0 文件导入与编辑表，该表格以列为整体查看，每一列代表的是该列通道与其他通道的信号关系，DQ 为数据线信号，DQS 为时钟信号数据；与每行中同名的（如 DQ0 与 DQ0）为主信号传输，其他名通道均为串扰信号；TX 为输入端，RX 为接收端，Delay 延时时间，RiseFile 上升沿 Tr0 文件，FallFile 下降沿 TR0 文件，Rise 表示上升沿信号，Fall 表示下降沿信号；以每一列为单位，选择每一行的名字与每一列名字对应关系相同的主信号

设置完成后界面示例如下：



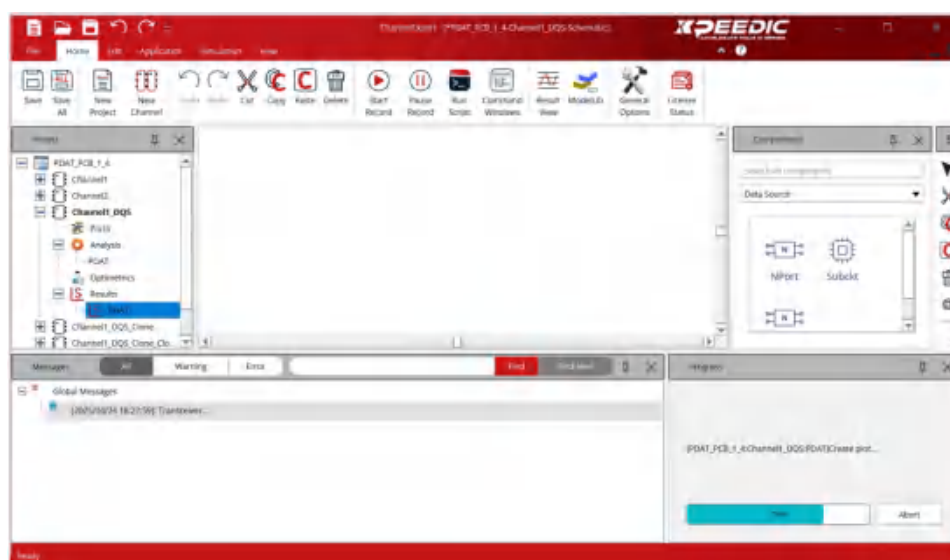
注：如果仅导入了一列信号，软件也可以进行仿真，此时产生的结果仅是该列信号的主信号以及其他路信号对该路的串扰影响，以 DQ0 为例：



6.4.2 PDAT 计算结果

在 PDAT 树节点下右键，选择 Analysis，进行仿真：

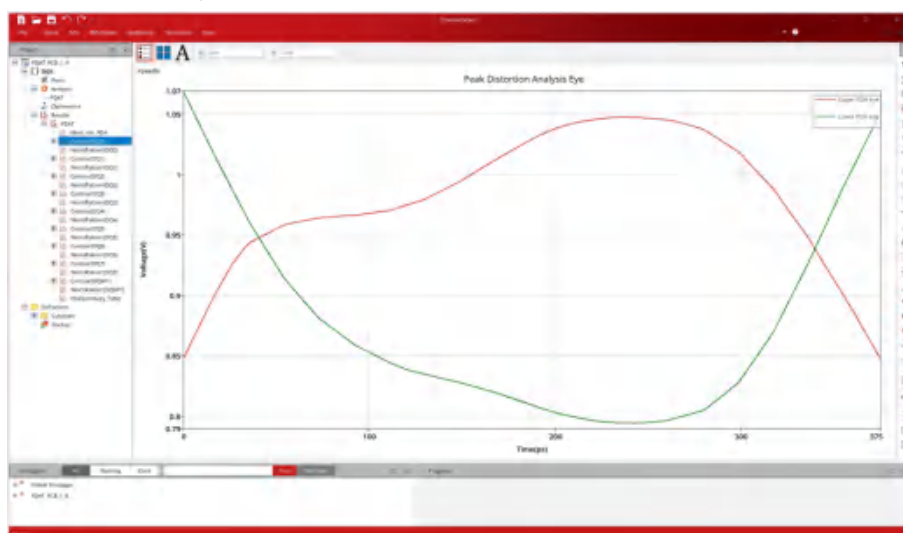
下方 Messages 会显示当前 solver 的 log 信息，进度条展示当前的运行状态。



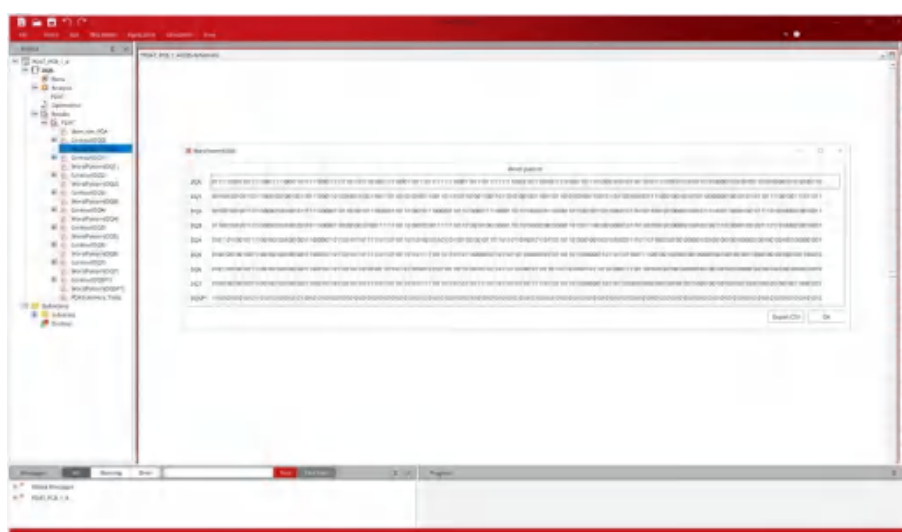
仿真结束，树节点下挂出 Mem_Sim_PDA 节点，可以显示各个 PDA 曲线的眼高、眼宽图，可通过 Eye Type 切换：



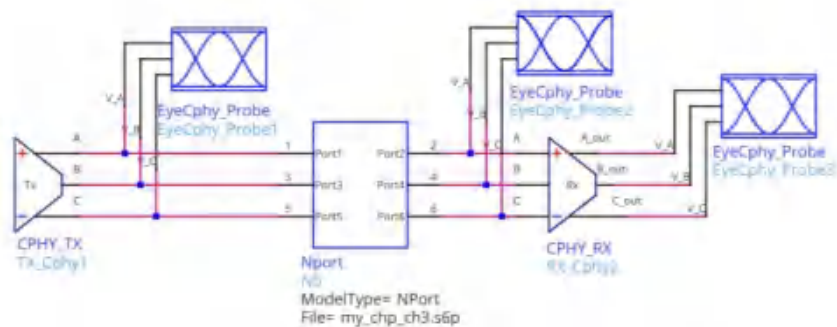
点击树节点下的 Contour，可以查看每个通路的 PDA 眼图，该眼图由 Upper PDA eye 曲线与 Lower PDA eye 两条曲线组成：



点击树节点下 WorstPattern，查看每个通道的最重要的码型：



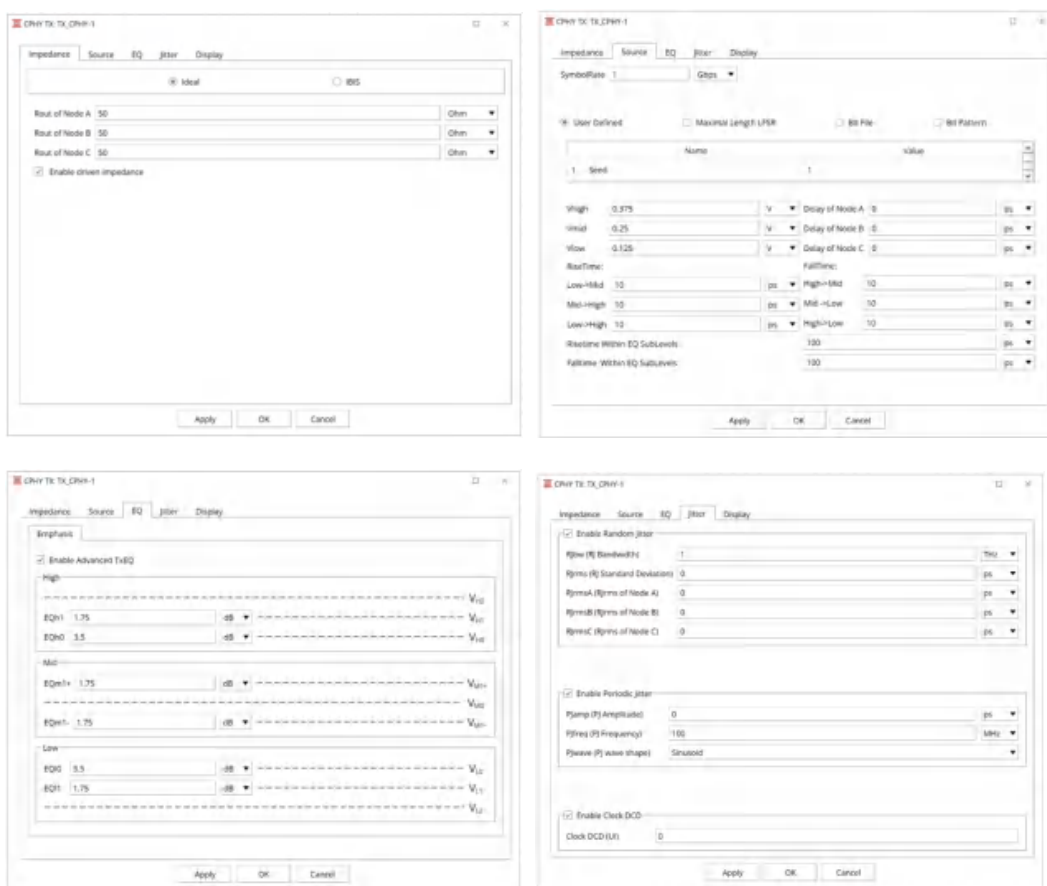
点击树节点下 PDASummary_Table，查看每个通道的串扰 Xtalk 和 ISI 值，以及每个通道的眼高眼宽峰峰值：



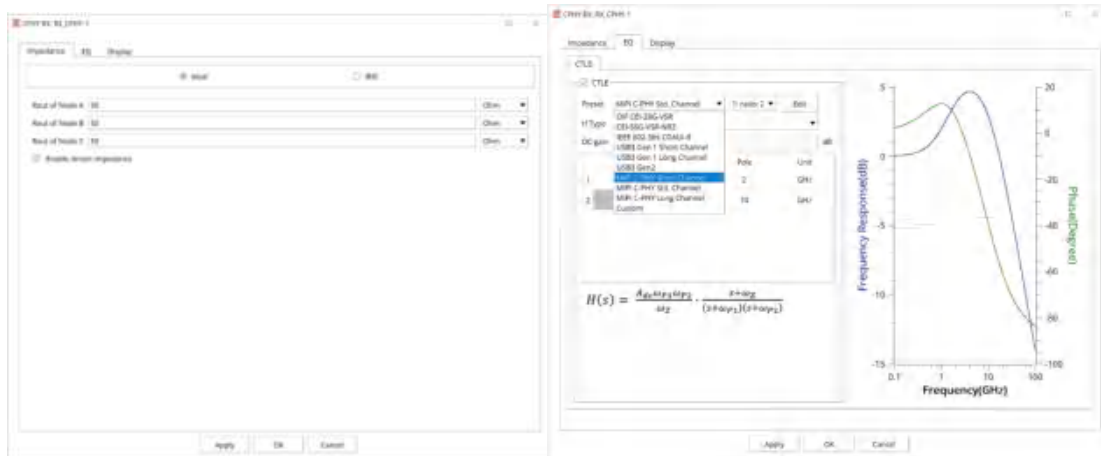
6.5.1.2 Cphy TX 与 Cphy RX 端内部设置

Cphy TX 输入端包括驱动强度、SymbolRate 与码型、EQ；Cphy RX 端包括驱动强度和 EQ。

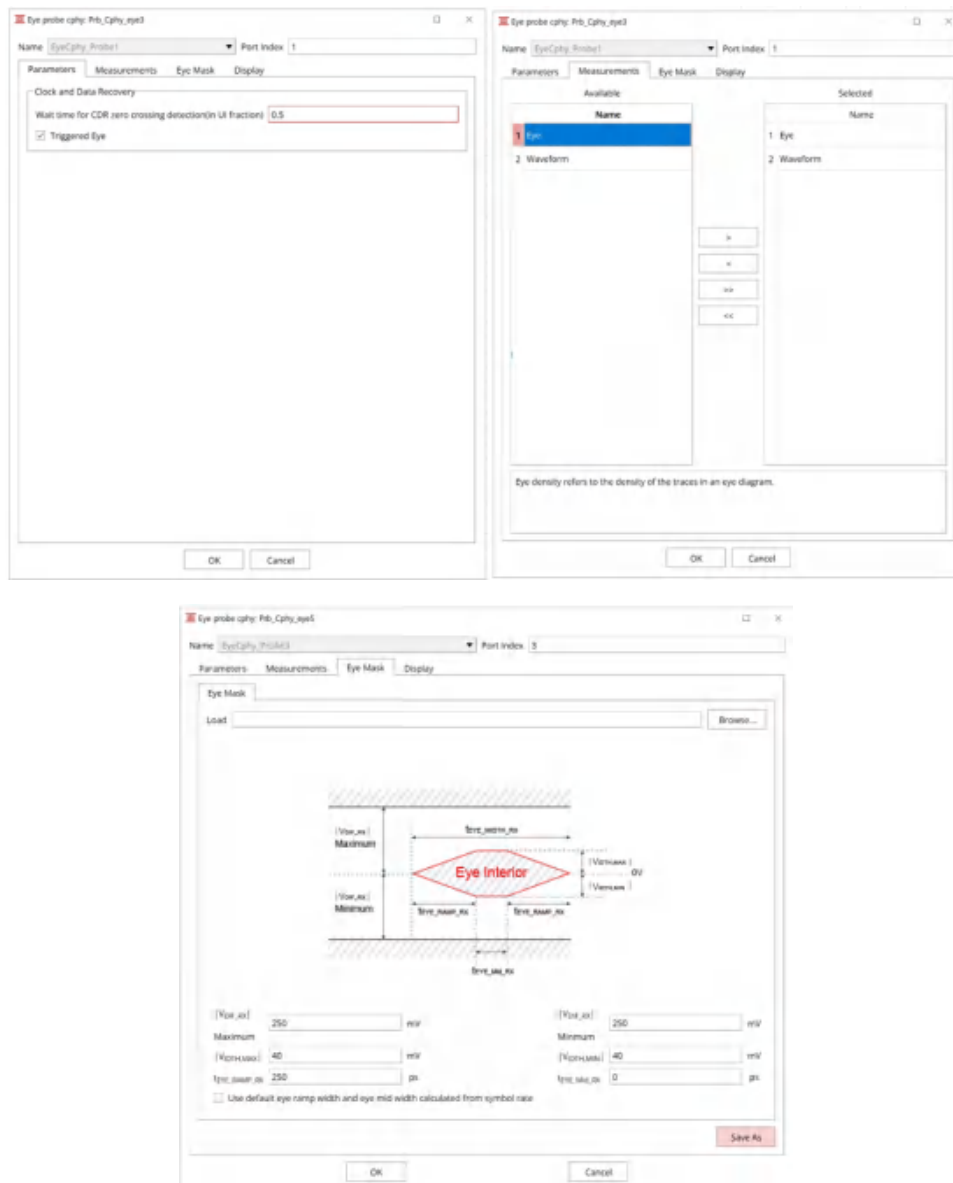
设置 Tx 的码率与码型以及驱动电压，驱动电气属性通过 Vhigh、Vmid、Vlow 以及边沿来描述；设置 Tx 的 EQ：



设置 Rx 的电气特性与 Rx 的均衡，Rx 内置均衡有标准协议的 Short、std、Long 等类型。

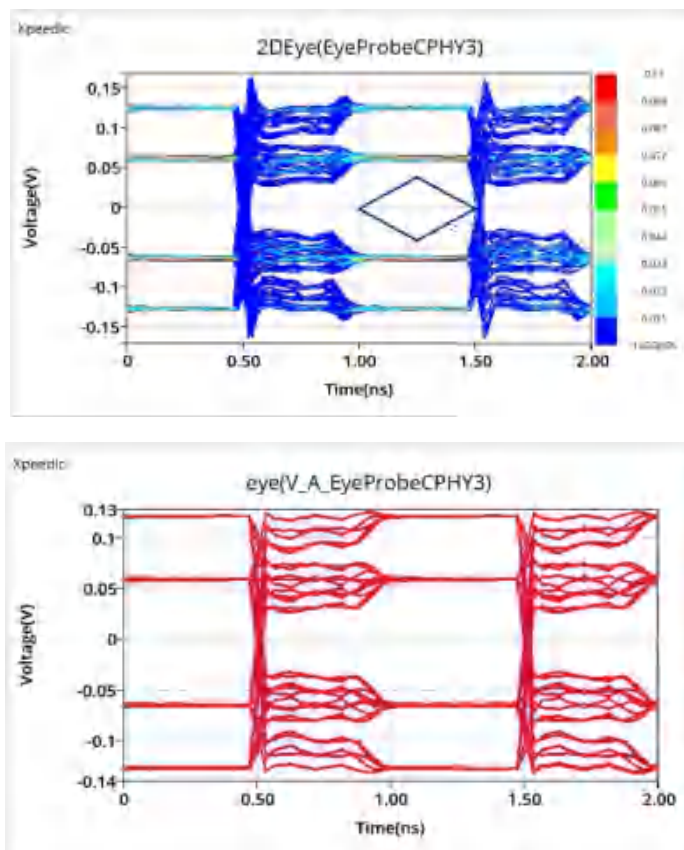


设置 C-phy probe 属性与 C-phy 的眼图触发方式，如果勾选 Triggered Eye 则按照标注 C-phy 协议恢复出触发时钟后再折叠眼图，如果不勾选则按照等周期叠加的方式；选择 C-PHY 要输出的内容，目前 C-phy 可以输出的内容包含 eye 与 Waveform：



6.5.1.3 仿真结果

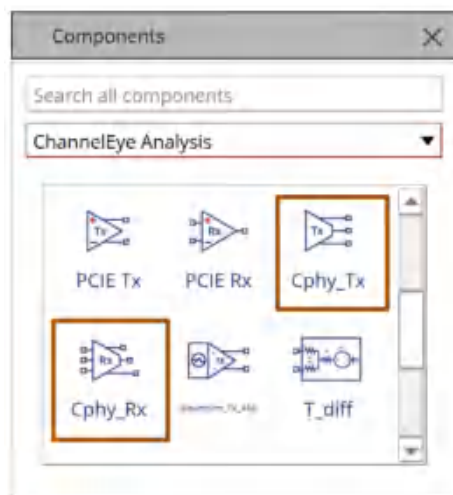
通过 Cphy 仿真链路，利用 Eye Probe 内置的眼图观测和电压观测，可以观测到 Cphy 链路的仿真结果，仿真结束后显示 RX 端输出的眼图以及各个节点的输出电压-时间（V-T）曲线如下图所示：



通过对眼图以及 VT 曲线分析，可以判别 C-PHY 链路的收发情况。

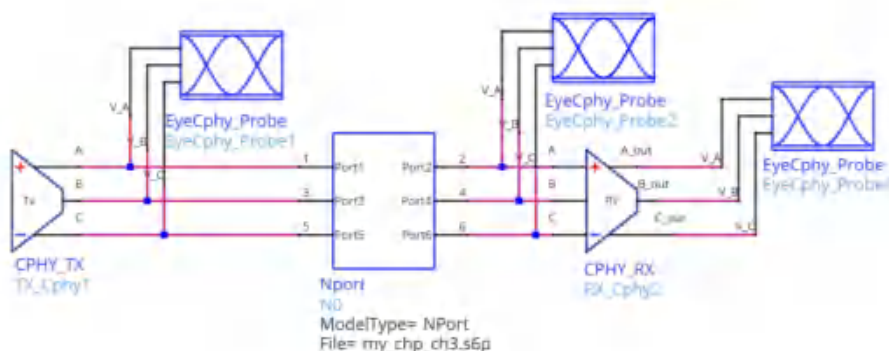
6.5.2 基于 IBIS 模型的 MIPI CPHY 仿真

Transceiver 的理想 buffer 模型下包含了 MIPI C-phy 发送接收接收器，可以用作评估 C-PHY 链路的收发情况。ChannelEyeAnalysis 下的 C-PHY 如下所示：



6.5.2.1 搭建仿真链路

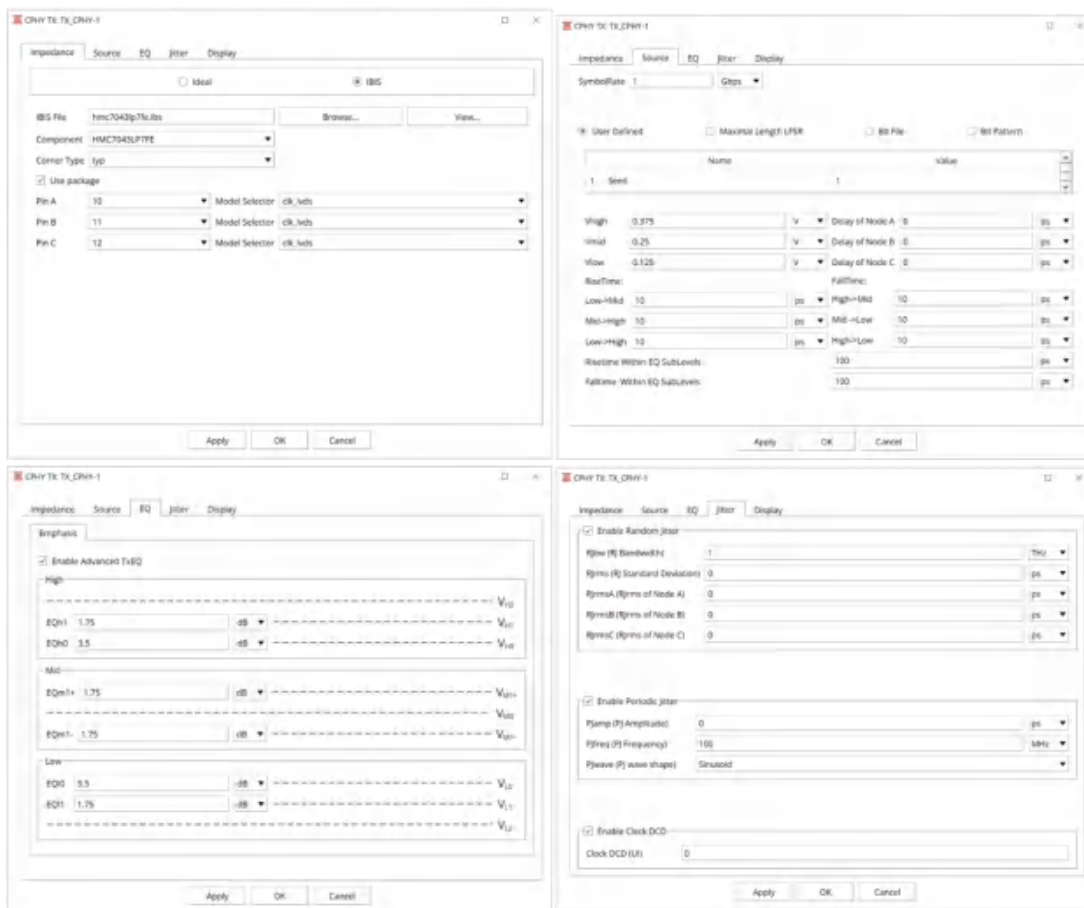
分别从元件窗口拖入 Tx/Rx buffer 与通道，建立连接关系， Cphy 仿真电路如下图所示。从输入端 Cphy-TX 输出信号，经过 S 参数通道，由接收端 Chhy-RX 接受信号，在 Cphy-RX 端的输出得到结果，在需要观察的节点加入 C-phy probe 图标用来观测波形。



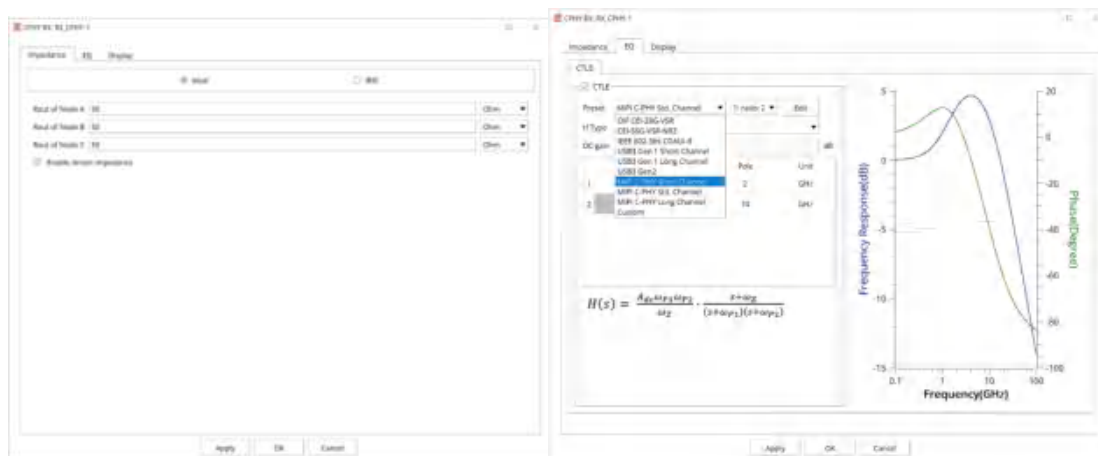
6.5.2.2 CPHY TX CPHY RX 端内部设置如下：

CPHY TX 输入端包括驱动强度、SymbolRate 与码型、EQ；CPHY RX 端包括驱动强度和 EQ。

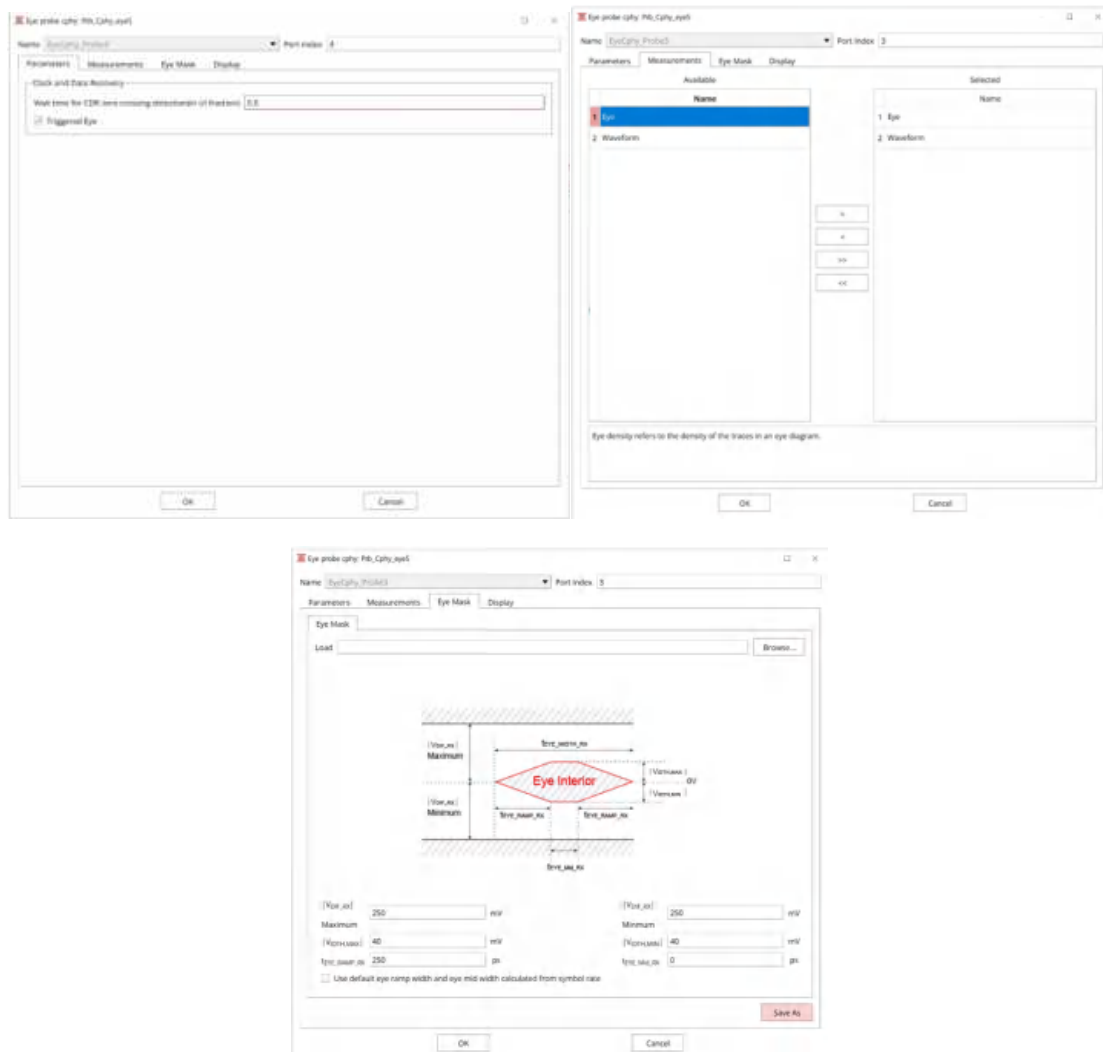
设置 Tx 的码率与码型以及驱动电压，驱动电气属性通过 Vhigh、Vmid、Vlow 以及边沿来描述；设置 Tx 的 EQ：



设置 Rx 的电气特性与 Rx 的均衡，Rx 内置均衡有标准协议的 Short、std、Long 等类型。

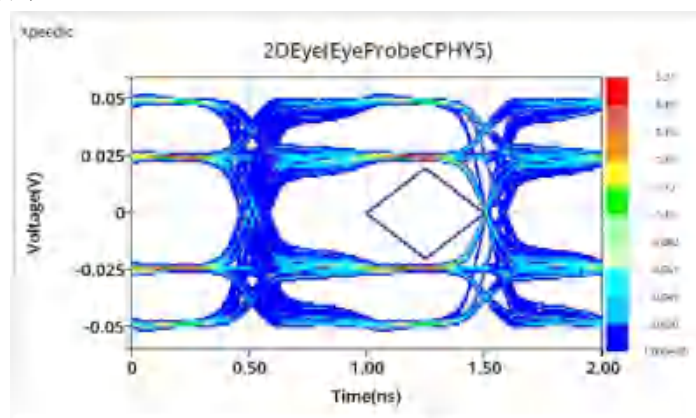


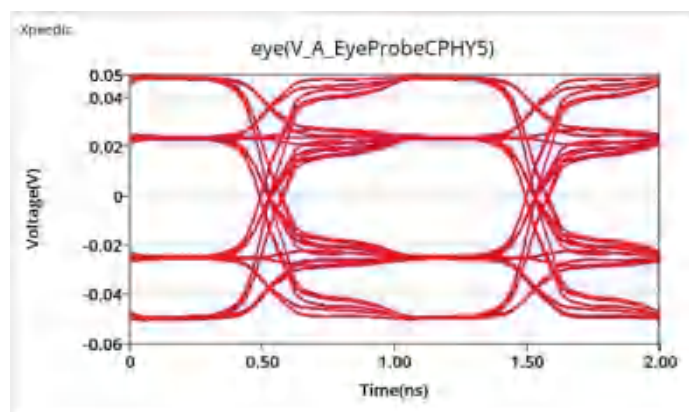
设置 C-phy probe 属性与 C-phy 的眼图触发方式，如果勾选 Triggered Eye 则按照标注 C-phy 协议恢复出触发时钟后再折叠眼图，如果不勾选则按照等周期叠加的方式；选择 C-PHY 要输出的内容，目前 C-phy 可以输出的内容包含 eye 与 Waveform:



6.5.2.3 C-PHY 仿真结果

通过 Cphy 仿真链路，利用 Eye Probe 内置的眼图观测和电压观测，可以观测到 Cphy 链路的仿真结果，仿真结束后显示 RX 端输出的眼图以及各个节点的输出电压-时间（V-T）曲线如下图所示：



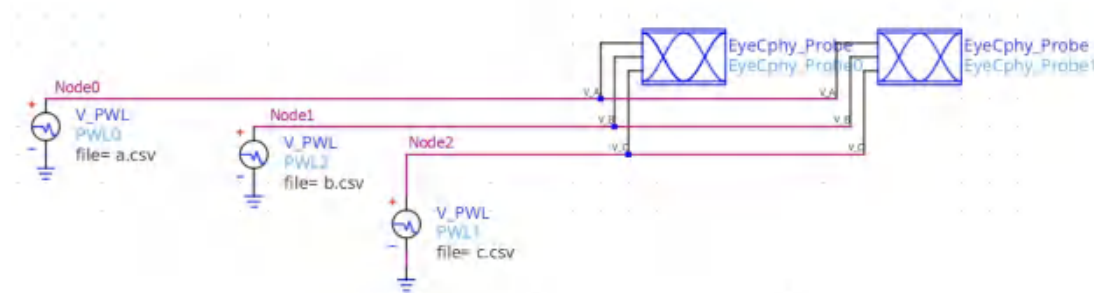


通过对眼图以及 VT 曲线分析，可以判别 C-PHY 链路的收发情况。

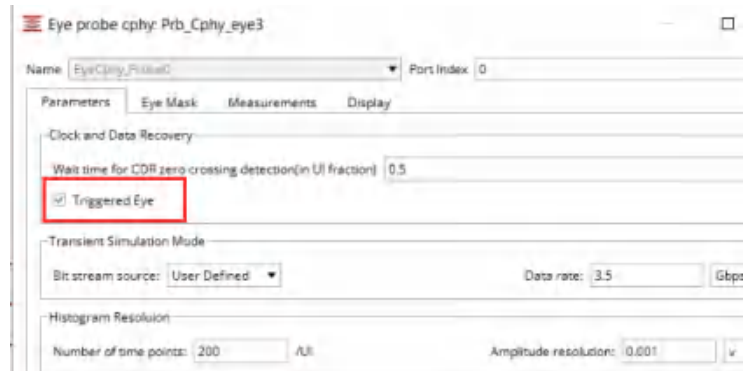
6.5.3 CPHY 的瞬态仿真分析

6.5.3.1 通过 C-PHY EyeProbe 观测眼图

使用 V-PWL 自定义 File 作为信号源，搭建 CPHY 链路



EyeCphy probe 设置，Trigger Eye 选择，设置曲线叠眼图的速率



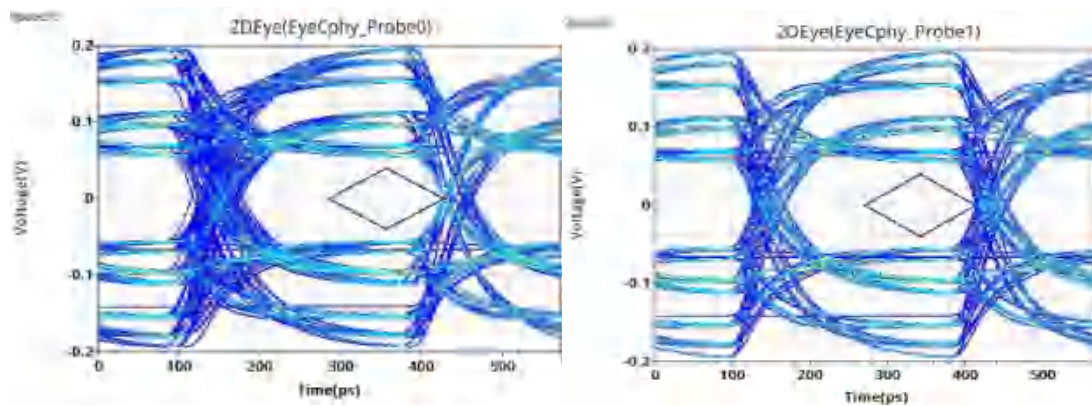
增加瞬态仿真分析



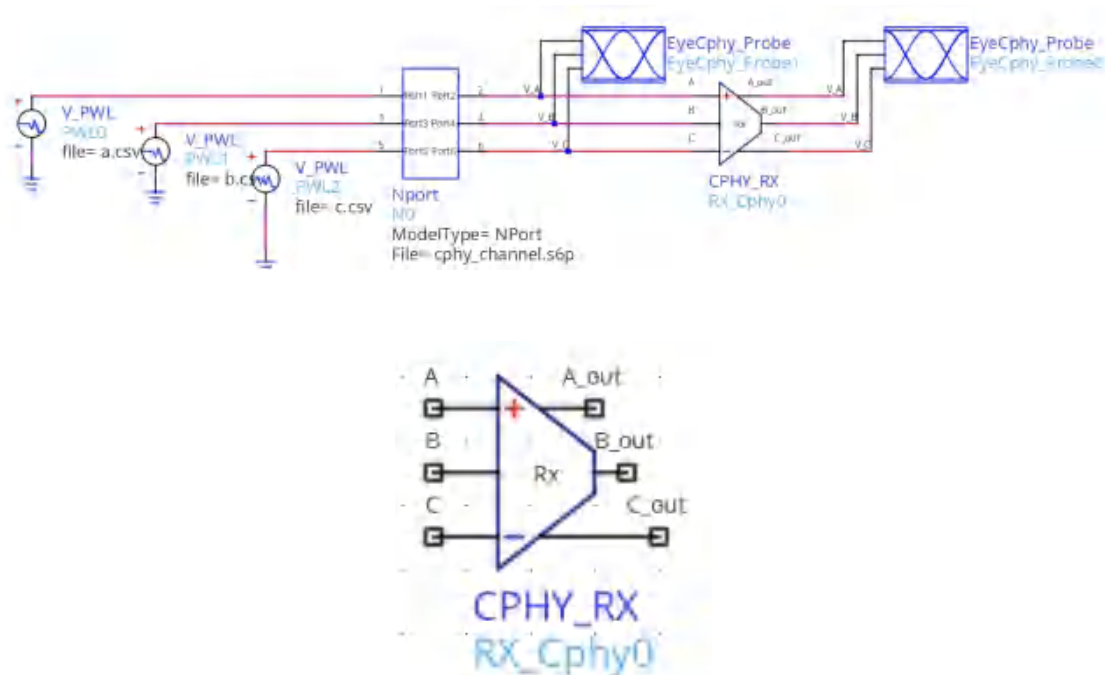
得到瞬态 2D 眼图

Triggered Eye 使能

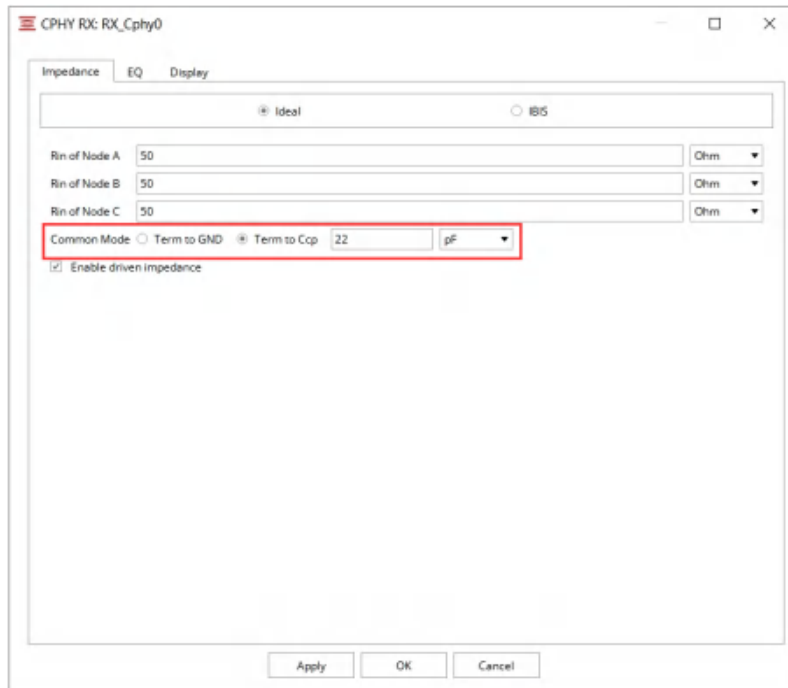
Triggered Eye 不使能



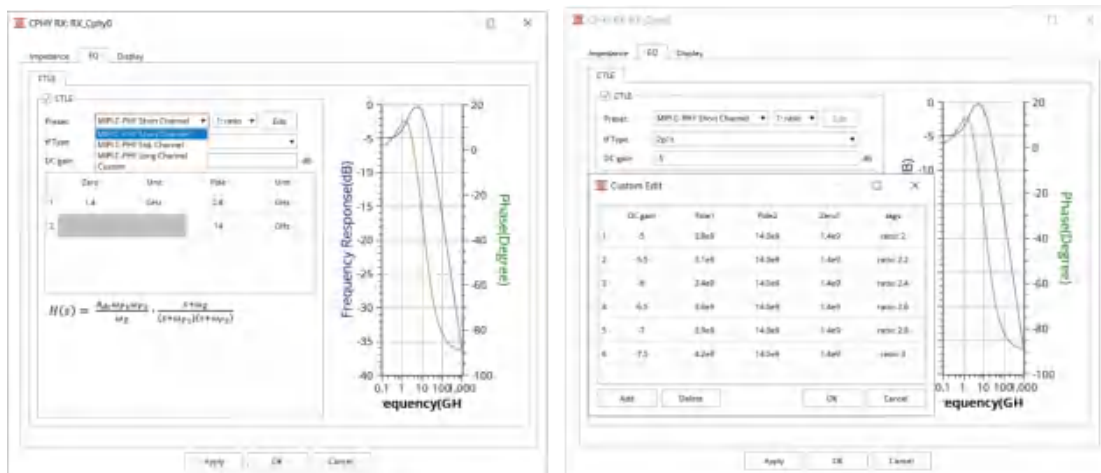
6.5.3.2 C-PHY Rx 的 CTLE 均衡功能



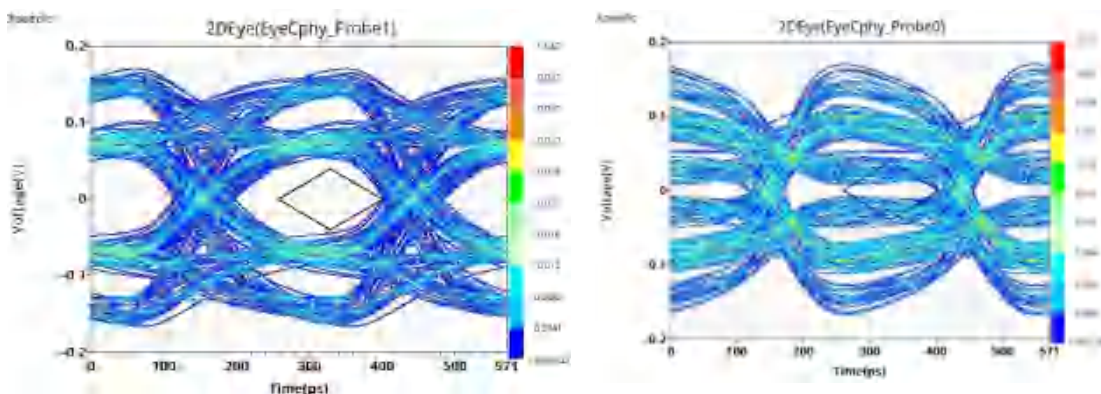
Rx CPHY 器件里的 Impedance 增加共模电压端接控制，勾选 Term to GND 端接到地，勾选 Term to Ccp 通过电容端接到地。



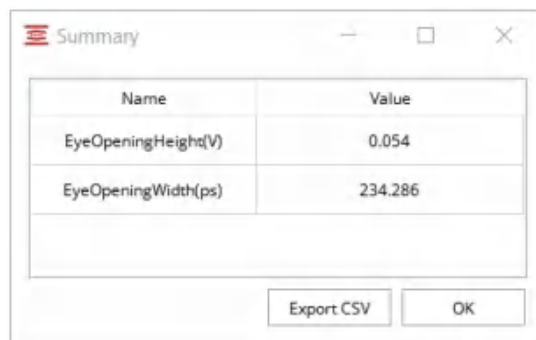
CPHY Rx 器件 CTLE 嵌入 CPHY 标准协议



结果如下:



眼宽眼高 summary 输出



Name	Value
EyeOpeningHeight(V)	0.054
EyeOpeningWidth(ps)	234.286

Export CSV OK

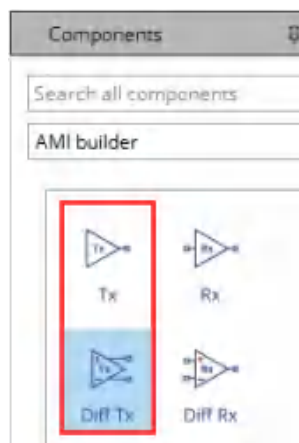
6.6 SERDES 眼图仿真分析

6.6.1 AMI builder

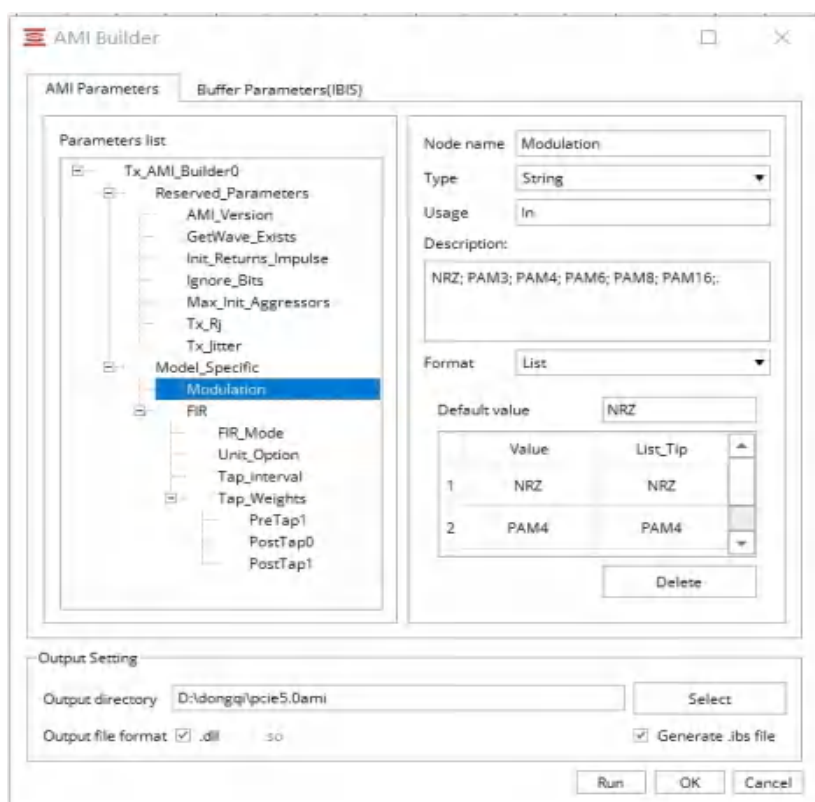
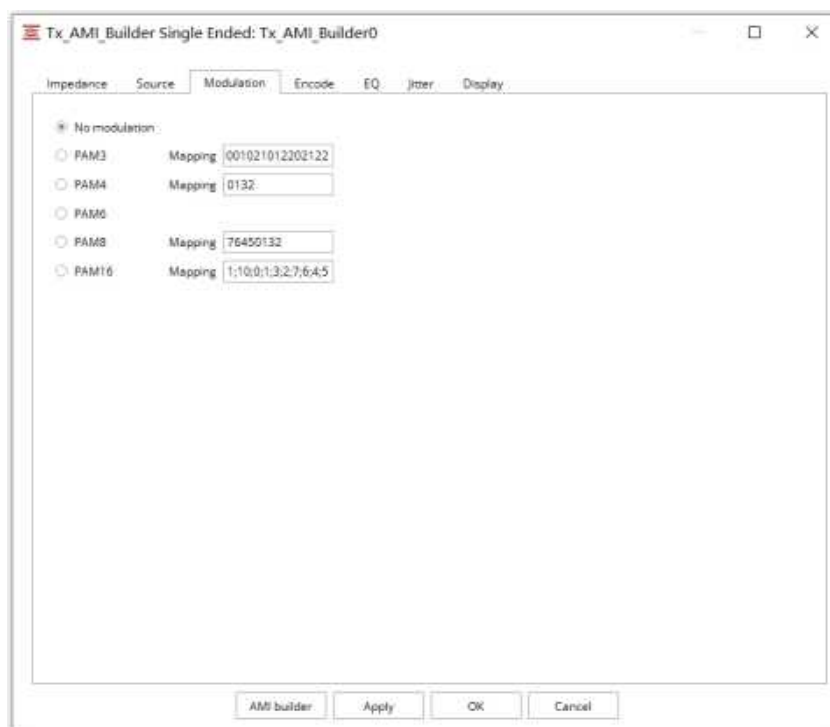
AMI 模型是 IBIS 标准的扩展，它能更好地捕捉信道中的复杂动态特性，主要用于解决传统 IBIS 模型无法描述的复杂均衡算法和信号处理功能。通过将这些功能算法化，AMI 模型可以更准确地模拟系统在实际工作中的行为。帮助工程师评估和优化系统在整个系统中的性能，确保信号成功地在不同芯片之间进行传输，从而提高系统的可靠性和稳定性。

6.6.1.1 Tx AMI build

在 Components 下找到 AMI builder 模块，支持单端和差分的 Tx。



双击 Tx 器件，在 Modulation 下选择信号编码方式，点击 AMI build，进入参数编界面。目前支持 NRZ、PAM3、PAM4、PAM6、PAM8、PAM16 编码方式。



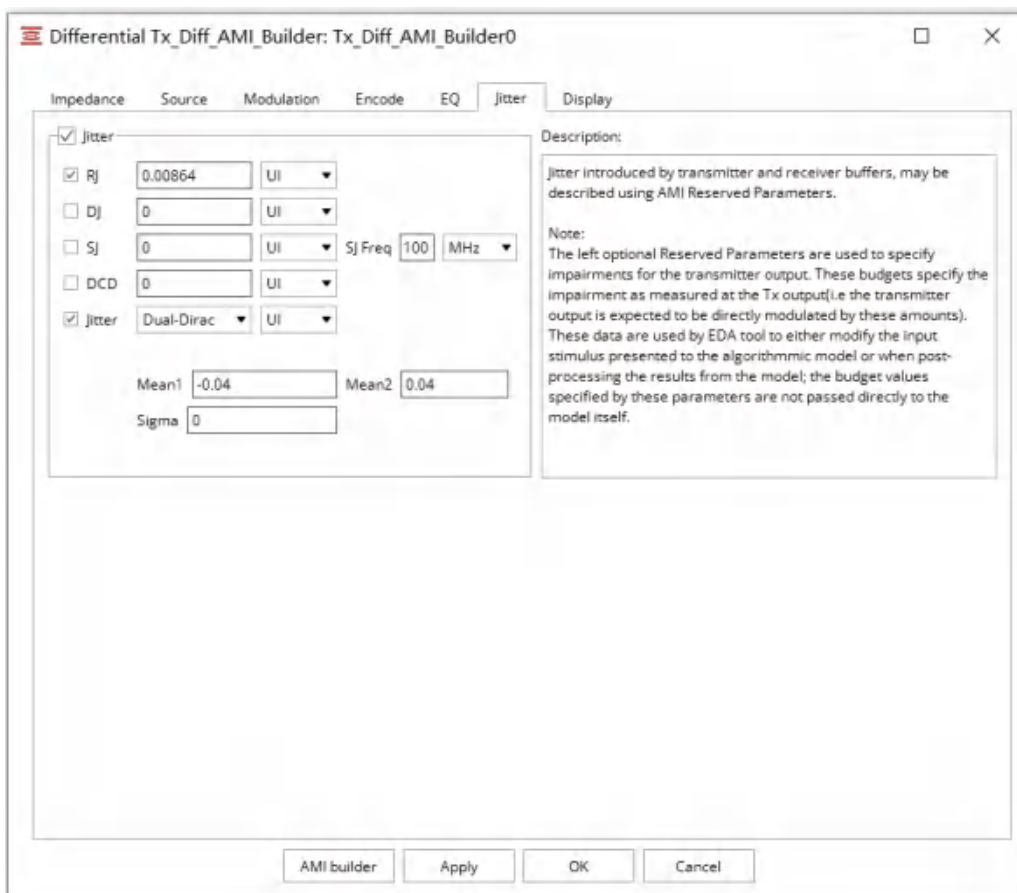
根据 IBIS 规范，目前 Reserved Parameter 支持了 AMI_Version、GetWave_Exists、Init_Return_Impulse、Ignore_Bits、Max_Init_Aggressors，具体对应的数据类型格式如下表，X 表示 IBIS 规范中需要支持的选项。

Reserved Parameter	Data Type					Data Format									
						Value	Range	Corner	List	Increment	Steps	Gaussian	Dual-Dirac	DJRj	Table
	Float	UI	Integer	String	Boolean										
AMI_Version				X		X									
GetWave_Exists					X	X									
Ignore_Bits			X			X									
Init_Returns_Impulse					X	X									
Max_Init_Aggressors			X			X									
Use_Init_Output					X	X									
Resolve_Exists					X	X									
Model_Name				X		X									
Special_Param_Names				X											X
Component_Name				X		X									
Signal_Name				X		X									
Rx_Decision_Time	X	X				X									
DC_Offset	X					X									
Rx_Use_Clock_Input				X		X			X						

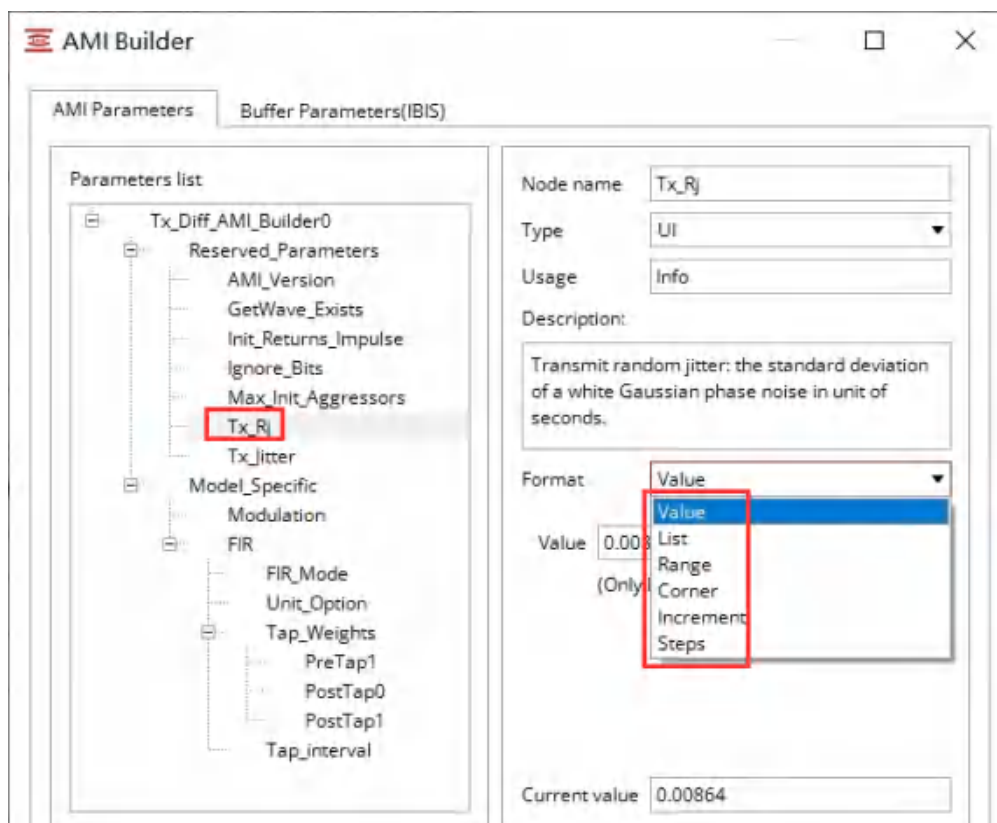
双击 Tx 器件，选择 Jitter，支持的 jitter 类型和格式，如下表，X 代表 IBIS 规范里需要支持的类型。

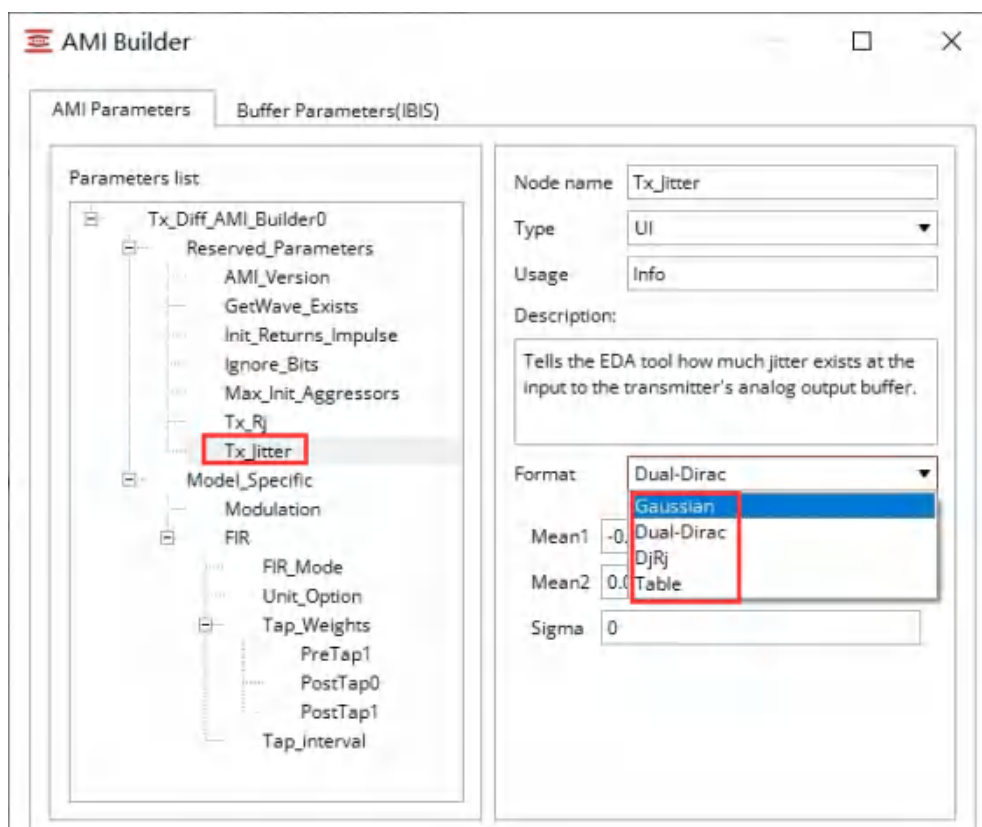
Table 26 – Allowable Data Types for Jitter and Noise Reserved Parameters						Data Format									
						Value	Range	Corner	List	Increment	Steps	Gaussian	Dual-Dirac	DJRj	Table
Reserved Parameter	Float	UI	Integer	String	Boolean										
Rx_Clock_PDF	X	X											X	X	X
Rx_Clock_Recovery_DCD	X	X				X	X	X	X	X	X				
Rx_Clock_Recovery_Dj	X	X				X	X	X	X	X	X				
Rx_Clock_Recovery_Mean	X	X				X	X	X	X	X	X				
Rx_Clock_Recovery_Rj	X	X				X	X	X	X	X	X				
Rx_Clock_Recovery_Sj	X	X				X	X	X	X	X	X				
Rx_DCD	X	X				X	X	X	X	X	X				
Rx_Df	X	X				X	X	X	X	X	X				
Rx_Noise, Rx_GaussianNoise	X					X	X	X	X	X	X				
Rx_UniformNoise	X					X	X	X	X	X	X				
Rx_Receiver_Sensitivity	X					X	X	X	X	X	X				
Rx_Rj	X	X				X	X	X	X	X	X				
Rx_Sj	X	X				X	X	X	X	X	X				
Tx_DCD	X	X				X	X	X	X	X	X				
Tx_Dj	X	X				X	X	X	X	X	X				
Tx_Filter	X	X										X	X	X	X
Tx_Rj	X	X				X	X	X	X	X	X				
Tx_Sj	X	X				X	X	X	X	X	X				
Tx_Sf_Frequency	X					X	X	X	X	X	X				

勾选 Jitter，灰的参数会高亮，在需要加入的 jitter 类型前打勾。Description 是参数描述区，用于提示用户参数的用途。



点击下方的 Ami builder，进入参数设置界面，可以看到不同的数据格式。





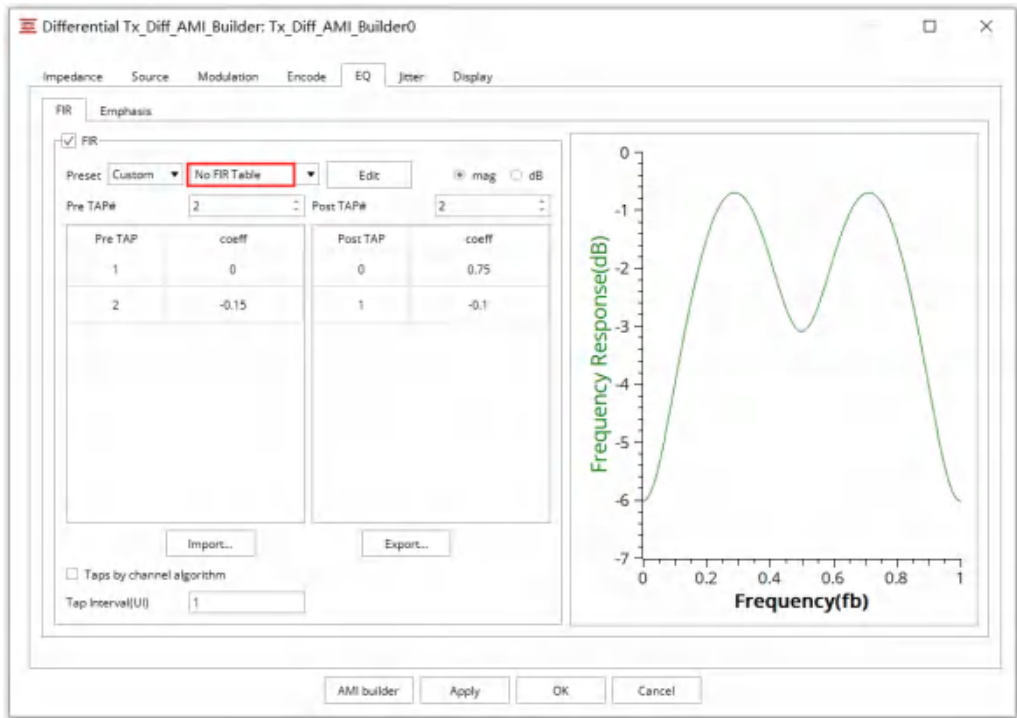
Jitter 对应的不同数据格式及写法，如下：

Format	Data Type					
	Float	UI	Integer	String	Boolean	Tap
Corner	X	X	X	X	X	X
DjRj	X	X				
Dual-Dirac	X	X				
Gaussian	X	X				
Increment	X	X	X			X
List	X	X	X	X	X	X
Range	X	X	X			X
Steps	X	X	X			X
Table	X	X	X	X	X	
Value	X	X	X	X	X	X

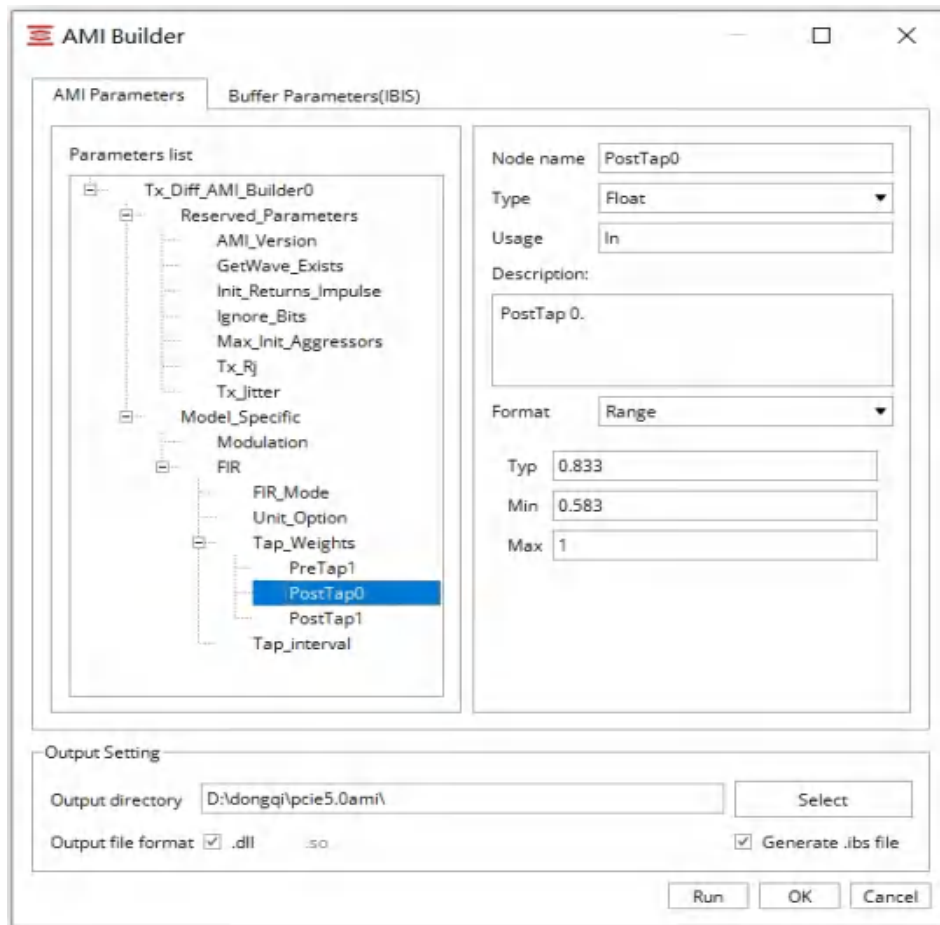
format
Value <value>
Range <typ value> <min value> <max value>
List <value> <value> <value> <value> ... <value>

List_Tip <entry><entry><entry><entry>...<entry>
Corner <typ value> <slow value> <fast value>
Increment <typ> <min> <max> <delta>
Steps <typ> <min> <max> <# steps>
Table
leaf Labels

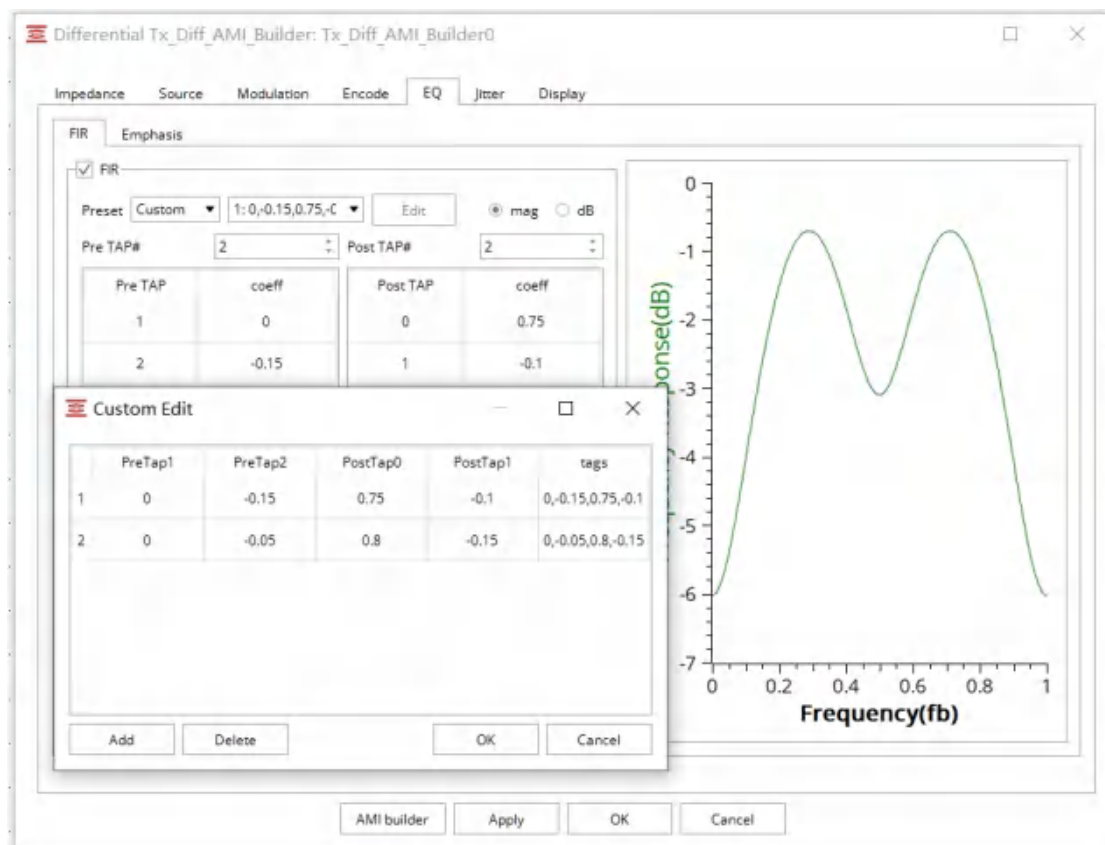
双击 Tx 器件，切换到 EQ 界面。TX EQ 支持 De-emphasis, Pre-emphasis , FIR。FIR 均衡不使用 Table 时，增加前抽头和后抽头的个数，填入系数。右侧曲线会根据公式自动变换。FIR 均衡 taps 可以自适应。



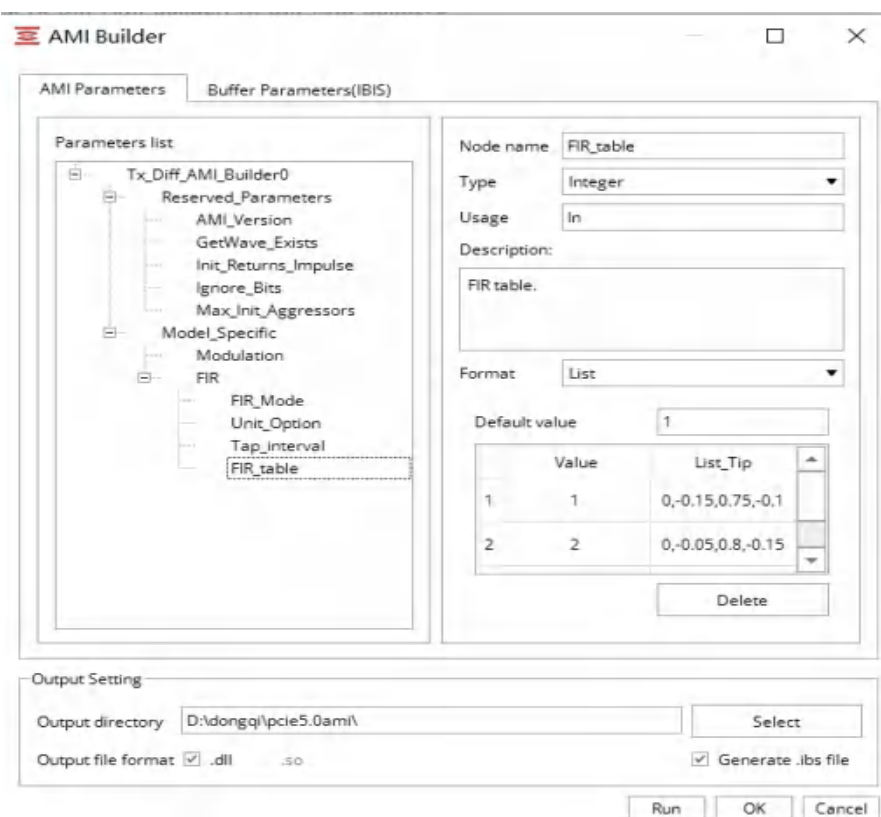
点击下方的 Ami builder，进入参数设置界面，下图抽头系数是 Range 的数据格式。



FIR 均衡也可以使用 Table 编辑，点击 Edit，弹出 Custom Edit 界面，填补提前设置好抽头个数对应的系数参数值，tags 一列为注释，可填可不填。如果需要增加点击 Add，增加多条参数组合，使用时在下拉框选择。



点击下方的 Ami builder ，进入可更改设置的参数，此处为 list 格式，Default value 填写一个默认值。



Tx ami build 填写结束，Tx IBIS 我们自带理想模板，模板里的参数也可支持编辑。

AMI Builder

AMI Parameters Buffer Parameters (IBIS)

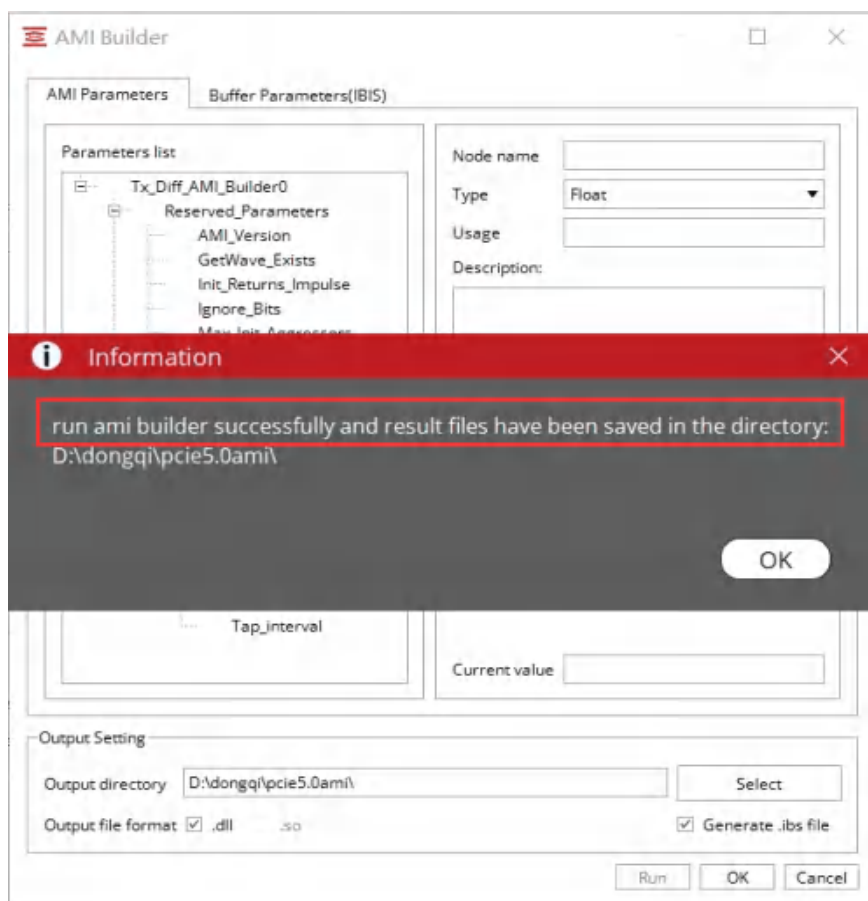
	Property Name	TYP	MIN	MAX
1	Voltage Range(V)	1	0.9	1.1
2	C_comp(pF)	0.2	0.18	0.22
3	Rising Time(ps)	10	10	10
4	Falling Time(ps)	10	10	10
5	PU Impedance(Ohm)	50	45	55
6	PD Impedance(Ohm)	50	45	55

Output Setting

Output directory:

Output file format: ☒ .dll ☐ .so ☒ Generate .ibis file

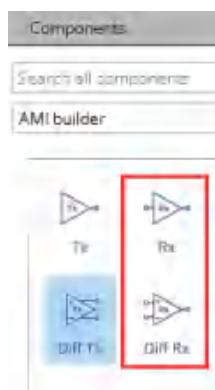
IBIS 和 AMI 确认设置后，Output directory select 创建的路径，Output file format 在 Window 系统下编译出.dll 文件，Linux 系统下编译出.so 文件。勾选 Generate .ibis file 同时会创建出 IBIS 文件，不勾选则只创建 ami、dll、so 文件。点击 Run，开始创建。注意必须在有 VS 编译软件的环境下才能编译出 dll。



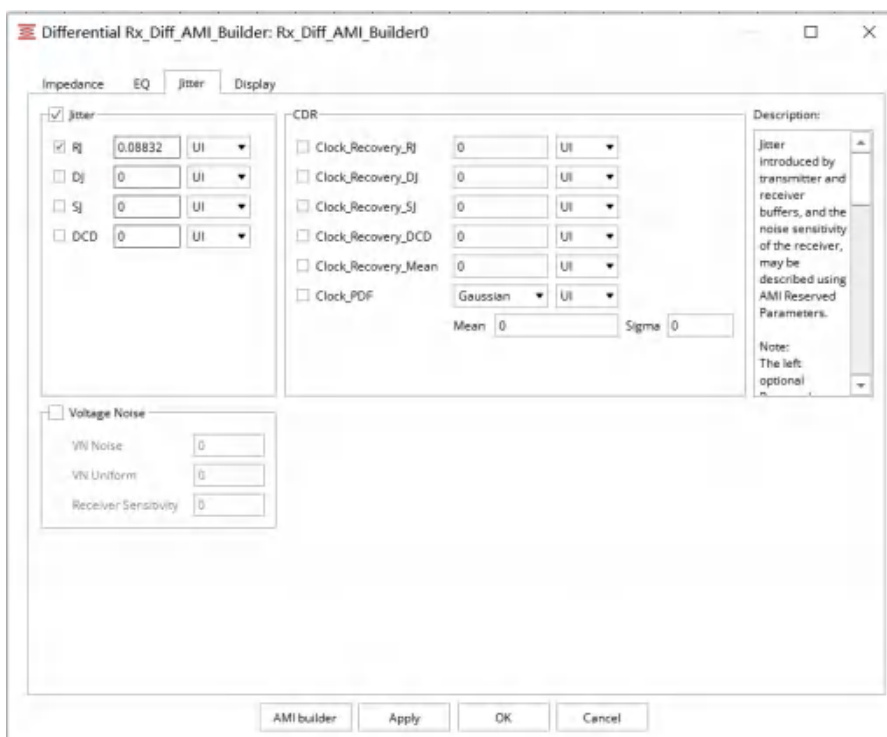
弹出对话框，提示编译成功。

6.6.1.2 Rx AMI build

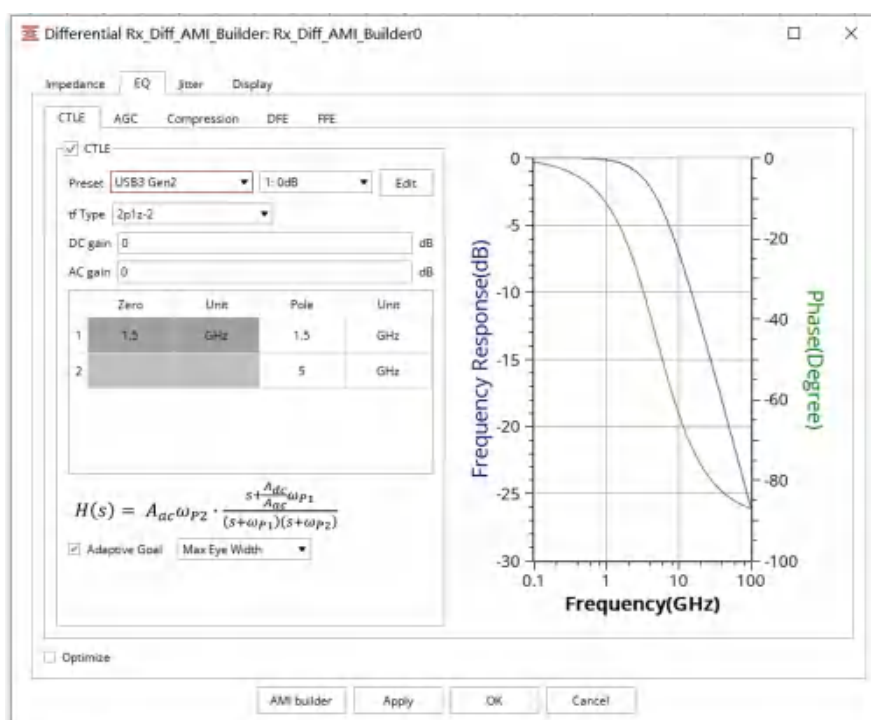
在 Components 下找到 AMI builder 模块，支持单端和差分的 Rx。



双击 Rx 器件，切换 Jitter 界面。勾选 Jitter，灰的参数会高亮，在需要加入的 jitter 类型前打勾。



双击 Rx 器件，切换到 EQ 界面。支持 FFE、CTLE、DFE、AGC、Compression。CTLE 支持多个 COM 协议。也可以自定义 Custom 进行系数编辑。CTLE 均衡设置，可依据对应通信协议，设置 CTLE 工作模式，支持定参数和自适应两种模式。

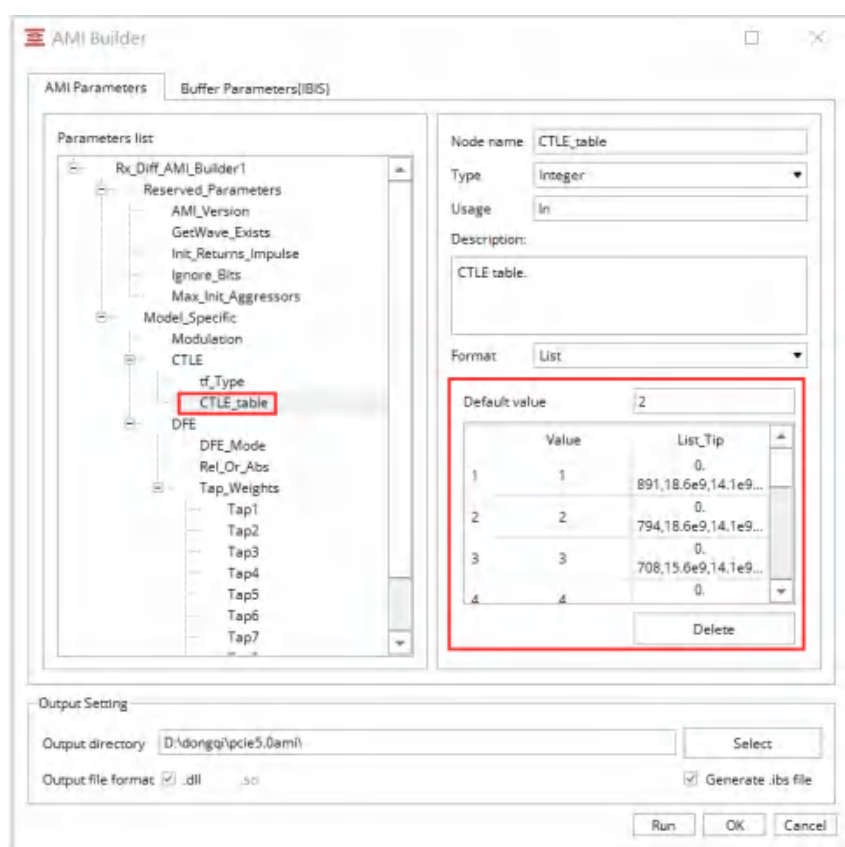


填写相关协议系数，taps 为注释，可写可不写。



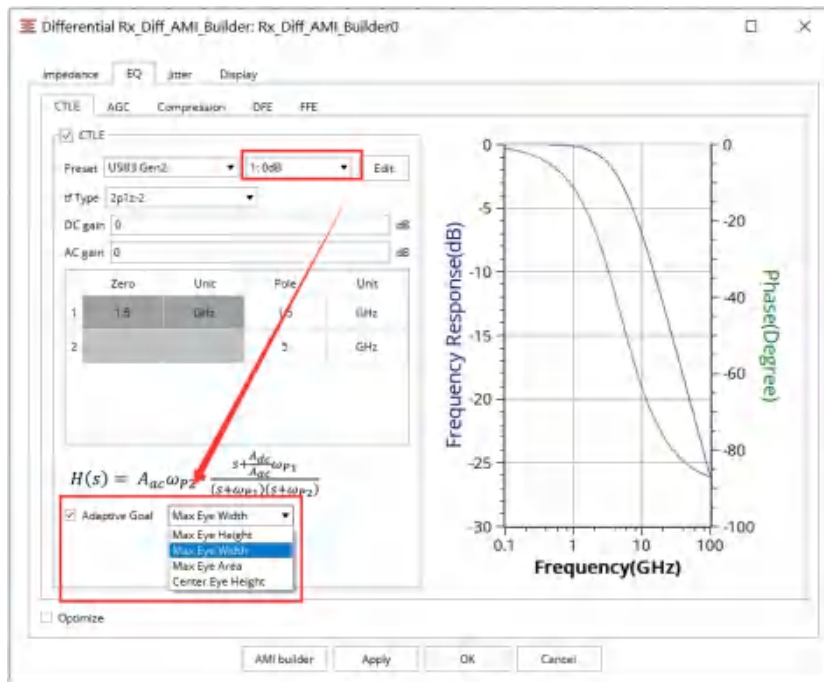
	DC gain	Pole1	Pole2	Zero1	tags
1	0.891	18.6e9	14.1e9	8.31e9	0.891,18.6e9,14.1e9,8.31e9
2	0.794	18.6e9	14.1e9	7.1e9	0.794,18.6e9,14.1e9,7.1e9
3	0.708	15.6e9	14.1e9	5.68e9	0.708,15.6e9,14.1e9,5.68e9
4	0.631	15.6e9	14.1e9	4.98e9	0.631,15.6e9,14.1e9,4.98e9
5	0.562	15.6e9	14.1e9	4.35e9	0.562,15.6e9,14.1e9,4.35e9
6	0.501	15.6e9	14.1e9	3.82e9	0.501,15.6e9,14.1e9,3.82e9
7	0.447	15.6e9	14.1e9	3.43e9	0.447,15.6e9,14.1e9,3.43e9
8	0.389	15.6e9	14.1e9	3e9	0.389,15.6e9,14.1e9,3e9
9	0.355	15.6e9	14.1e9	2.67e9	0.355,15.6e9,14.1e9,2.67e9

点击下方的 Ami builder，进入参数设置界面，下图 CTLE_table list 数据格式。Default value 填写一个默认值。

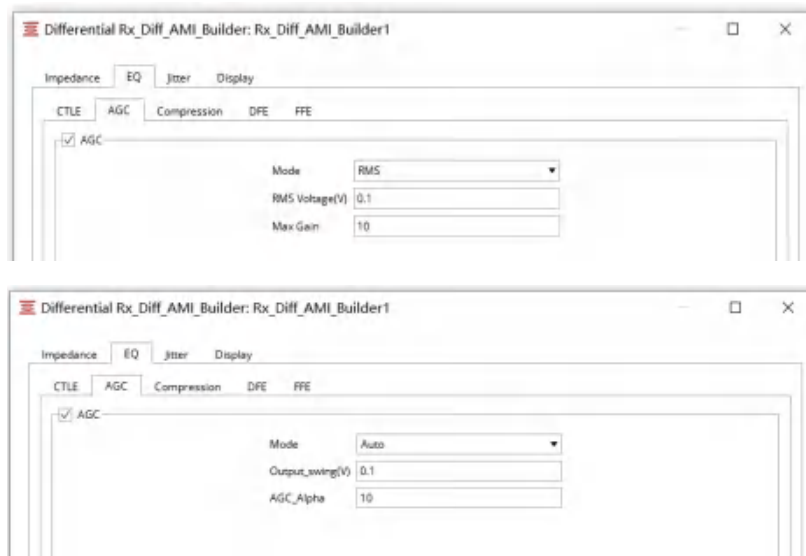


CTLE 定参数模式下可设置直流增益、零极点值，定参数模式下支持调用频域幅度文件,支持以下性能指标(Max Eye Height、Max Eye Width、Max Eye Area、Center Eye Height) 输出;

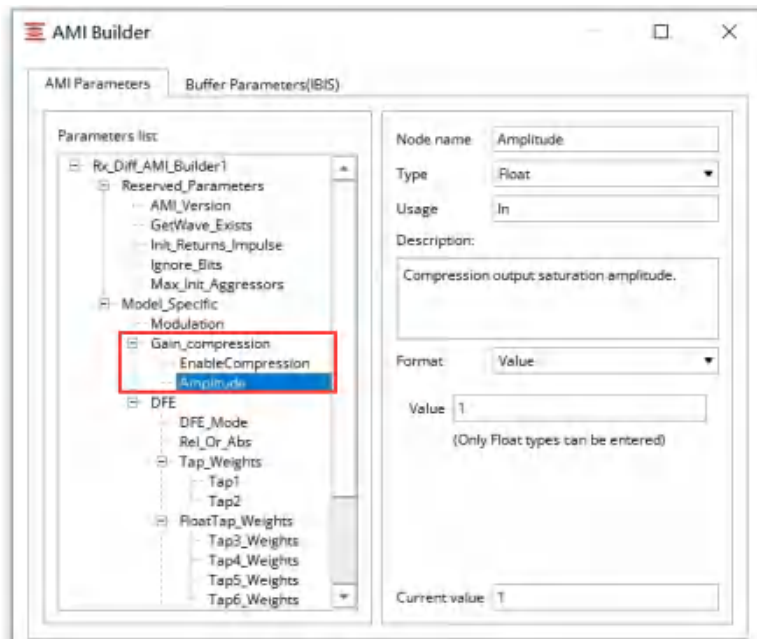
在 Preset 的增益下拉状态下 Adaptive Goal 高亮，勾选 Adaptive Goal 使能根据性能指标对协议里的参数进行扫描。



AGC 自动增益设置与 Saturating Amplifier 饱和放大限制设置。

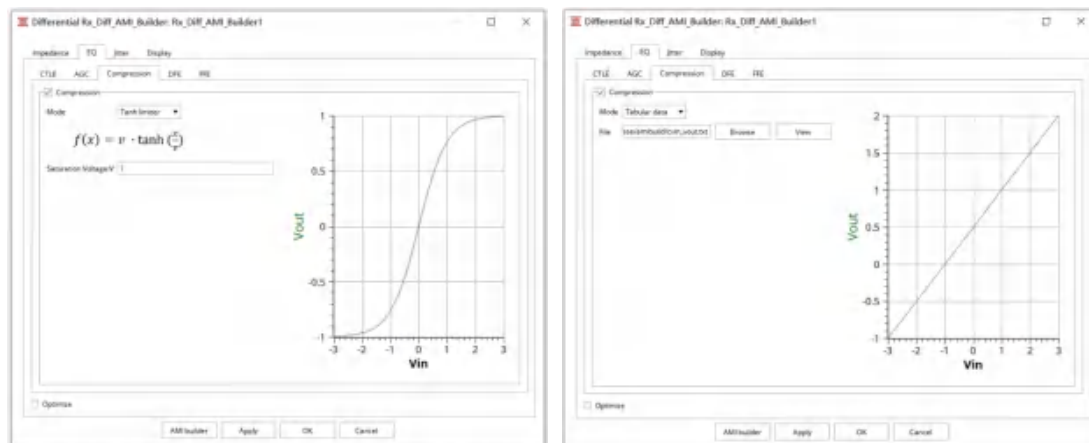


点击 AMI builder，查看编辑参数

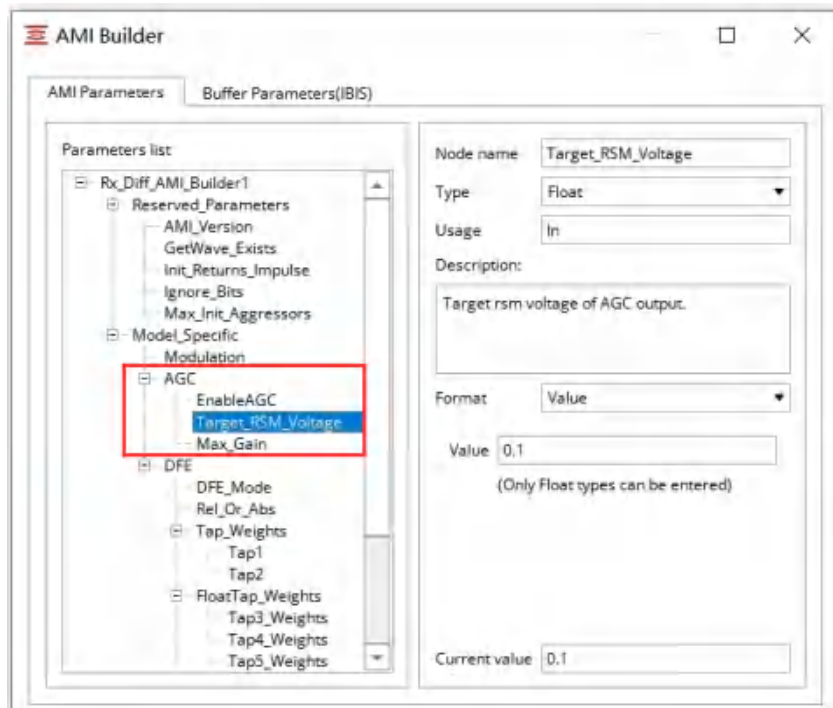


勾选 Compression 使能 AnalogIn 与 AnalogOut 设置稳定电平电压、边沿跳变时间、端接阻抗与端接电容。

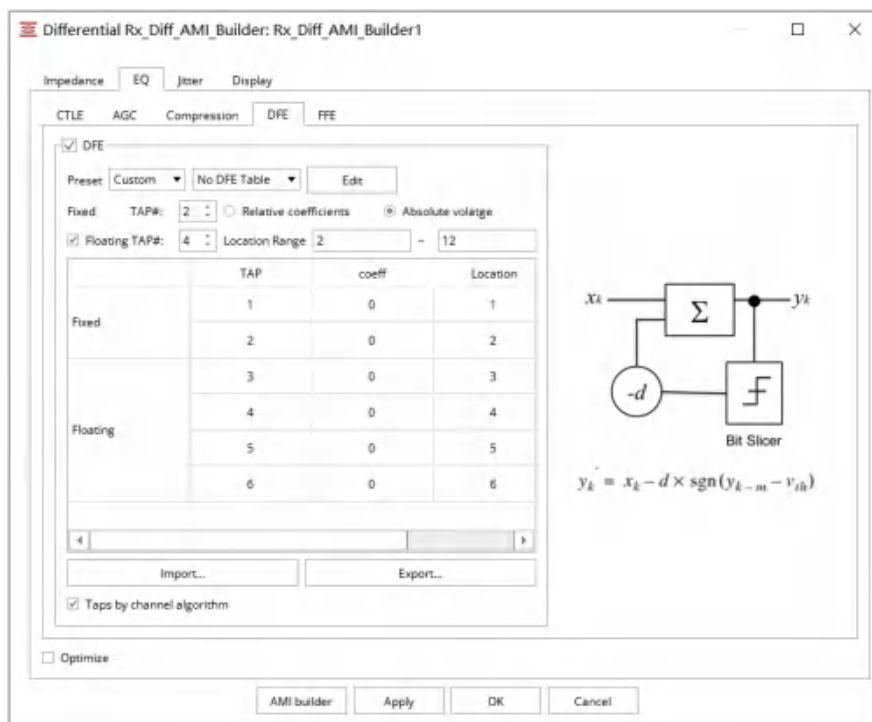
Mode 支持 Tanh limiter, Tabular data。



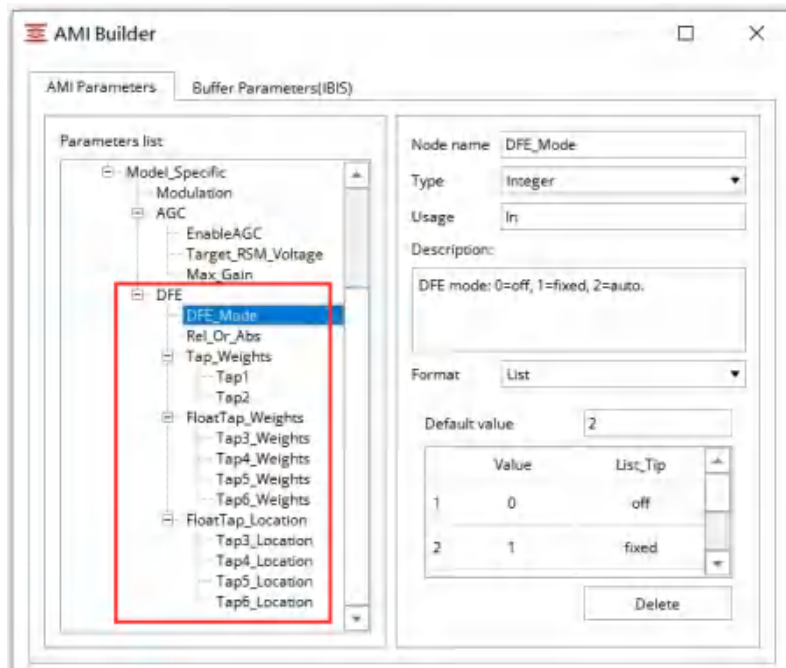
点击 AMI builder, 查看编辑参数



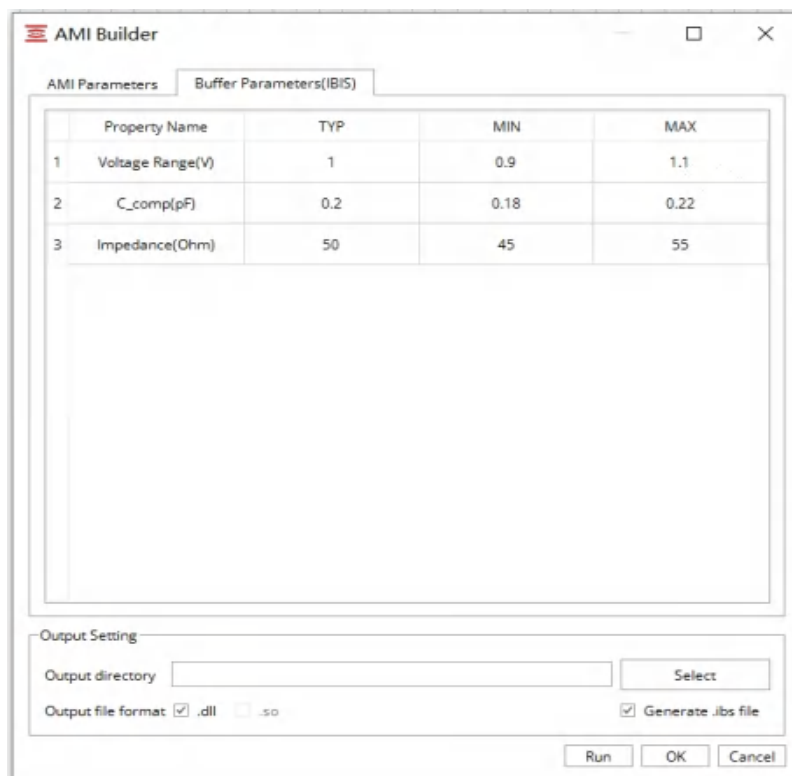
DFE 支持 Fixed Tap 和 Floating Tap，Floating Tap 勾选使能，设置抽头数和抽头系数，设置 Location Range。勾选下方的 Taps by channel algorithm，自适应最优值。



点击 AMI builder，查看编辑参数



Rx ami build 填写结束，Rx IBIS 自带理想模板，模板里的参数也可支持编辑。

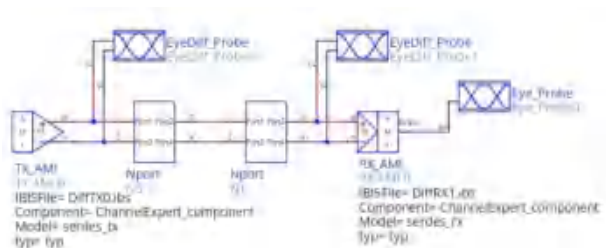


IBIS 和 AMI 确认设置后，Output directory select 创建的路径，Output file format 在 Window 系统下编译出.dll 文件，Linux 系统下编译出.so 文件。勾选 Generate .ibis file 同时会创建出 IBIS 文件，不勾选则只创建 ami、dll、so 文件。点击 Run，开始创建。注意必须在有 VS 编译软件的环境下才能编译出 dll。

此电脑 > Data (D:) > dongqi > pcie5.0ami

名称	修改日期	类型	大小
DiffRX-1	2025/4/23 11:21	IBS 文件	2 KB
DiffTX-1	2025/4/23 11:21	IBS 文件	2 KB
Rx_Diff_AMI_Builder1	2025/4/23 11:22	AMI 文件	5 KB
Rx_Diff_AMI_Builder1.dll	2025/4/23 11:22	应用程序扩展	36,389 KB
Tx_Diff_AMI_Builder0	2025/4/23 11:21	AMI 文件	3 KB
Tx_Diff_AMI_Builder0.dll	2025/4/23 11:21	应用程序扩展	42 KB

6.6.2 基于 IBIS AMI 模型的仿真

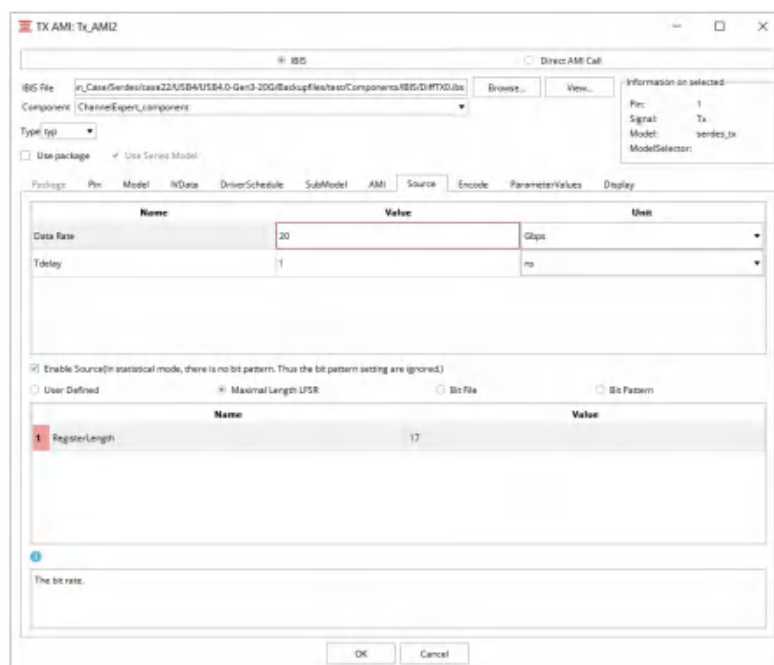


6.6.2.1 Tx AMI 设置

双击 Tx AMI 器件，勾选 IBIS，导入 IBIS File。

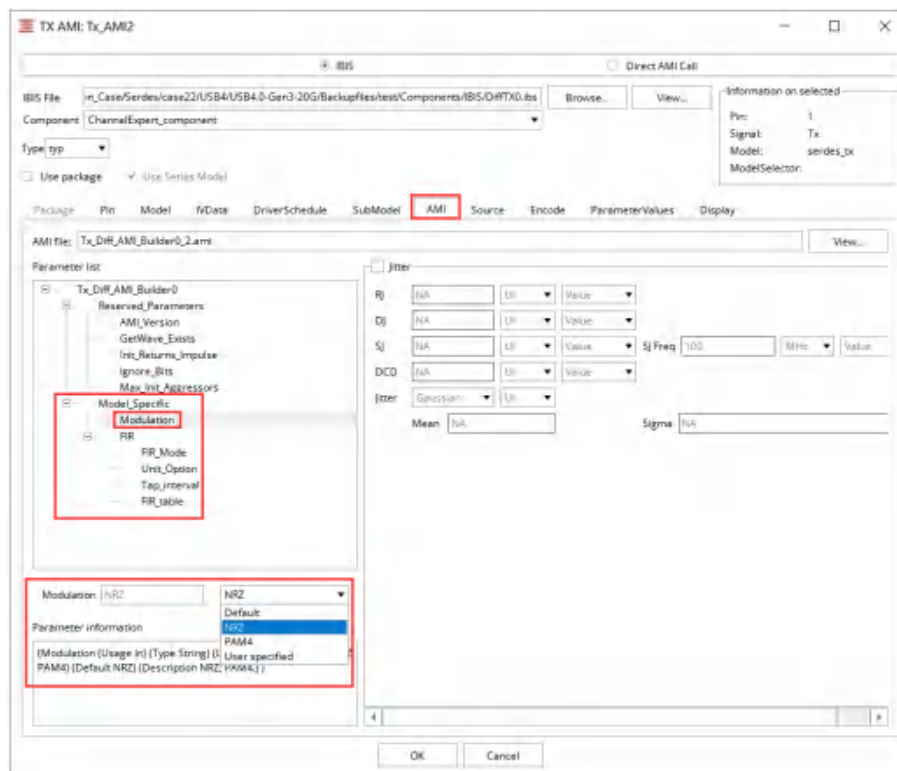
Source 里设置 Data Rate，设置模型仿真速率。

Bit by bit 需要输入码型，在 Source 界面选择码型。

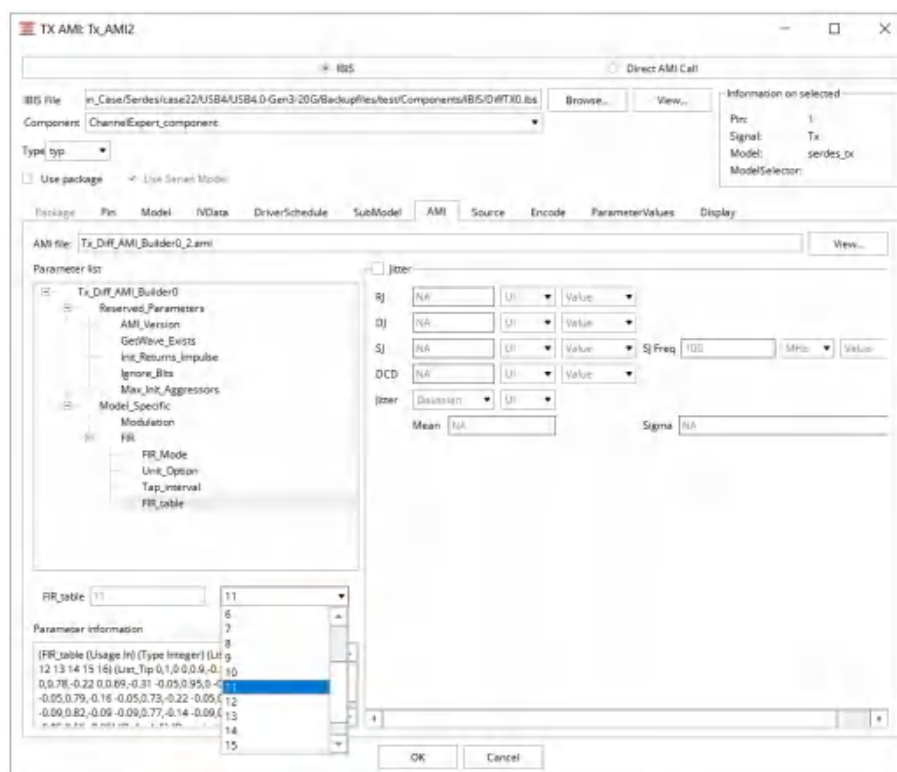


Tx AMI 界面挂载当前 IBIS 下 AMI 里的参数

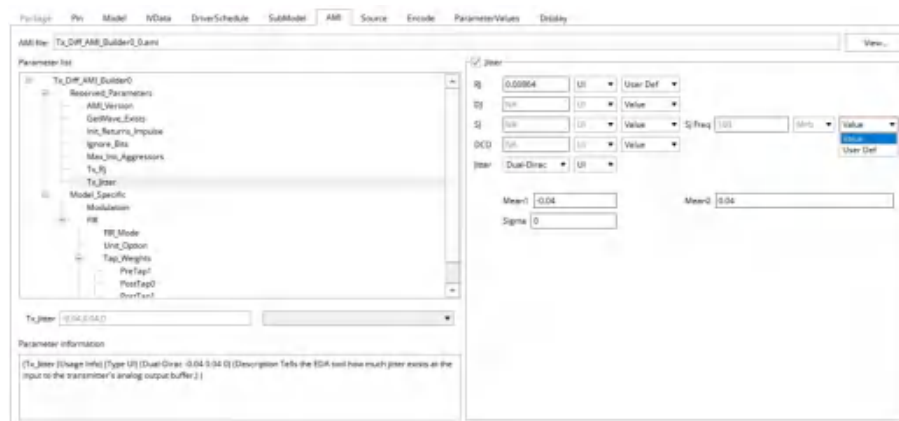
Reserved Parameters 下的指标是 AMI 里的固定设置，除了 Jitter 可以通过右侧界面更改，其他属性不能更改。Model Specific 下的属性参数都可以编辑更改。Modulation 调制方式可以通过下拉切换 NRZ 和 PAM4。



Tx 端 AMI 模型如果带 FIR 均衡，可以下来选择对应参数，如果需要自定义，下拉选择 User specified 通过编辑框输入 value 和字符串。



Tx AMI model 自带 Jitter 会自动写入右侧，可以通过勾选开启是否注入抖动。也可以对 Tx 端 jitter 进行参数编辑。



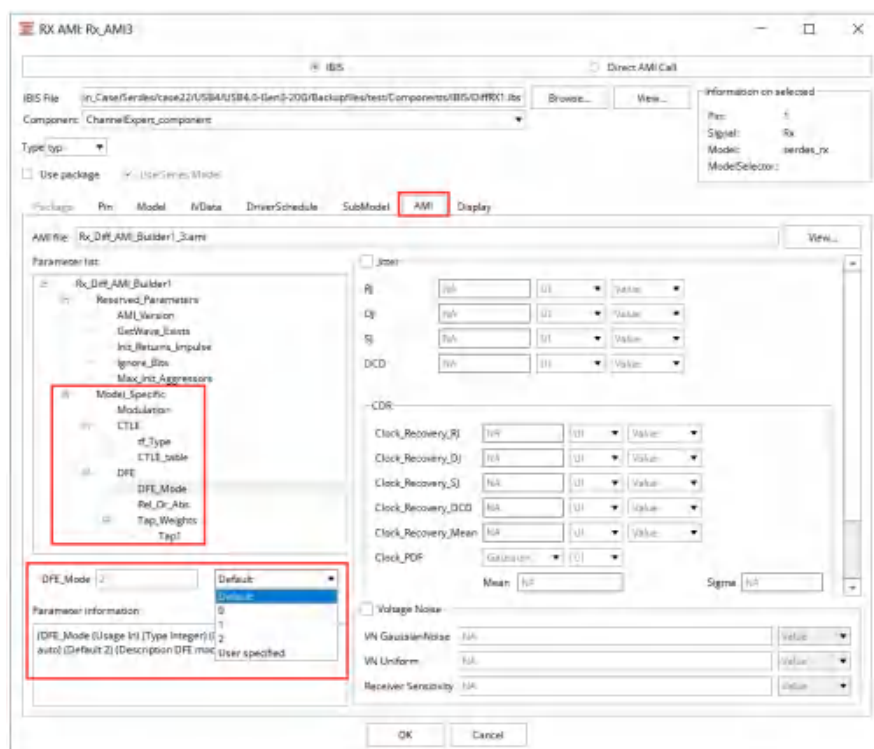
6.6.2.2 Rx AMI 设置

双击 Rx AMI 器件，勾选 IBIS，导入 IBIS File。

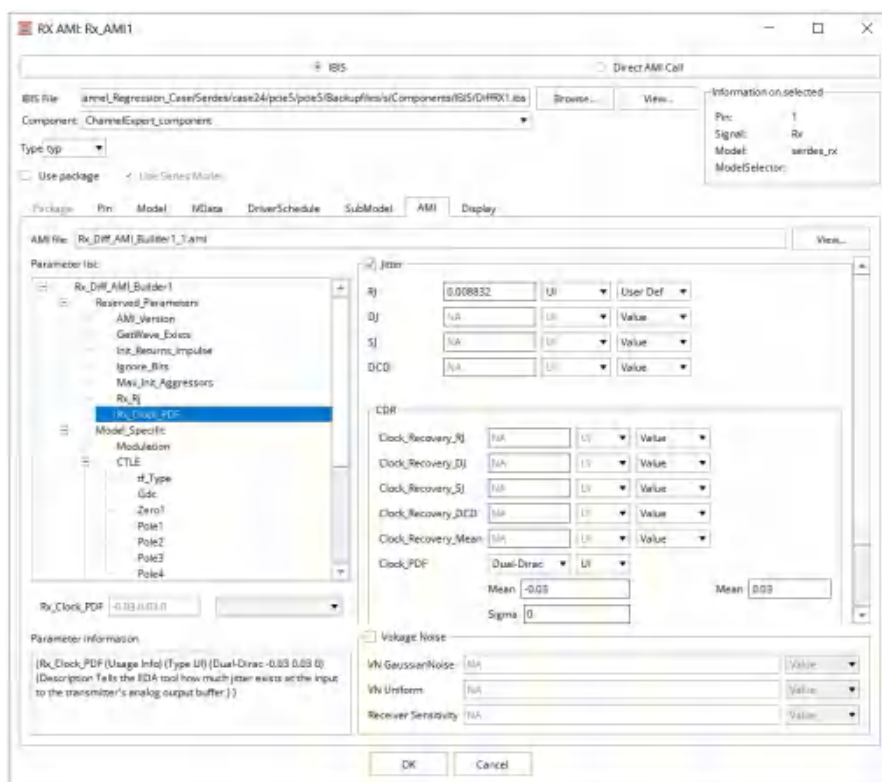
Rx AMI 界面挂载当前 IBIS 下 AMI 里的参数

Reserved Parameters 下的指标是 AMI 里的固定设置，除了 Jitter 可以通过右侧界面更改，其他属性不能更改。Model Specific 下的属性参数都可以编辑更改。

Rx 端主要的均衡有 CTLE、DFE、FFE，可以对 Model 里 AMI 参数下拉更改。



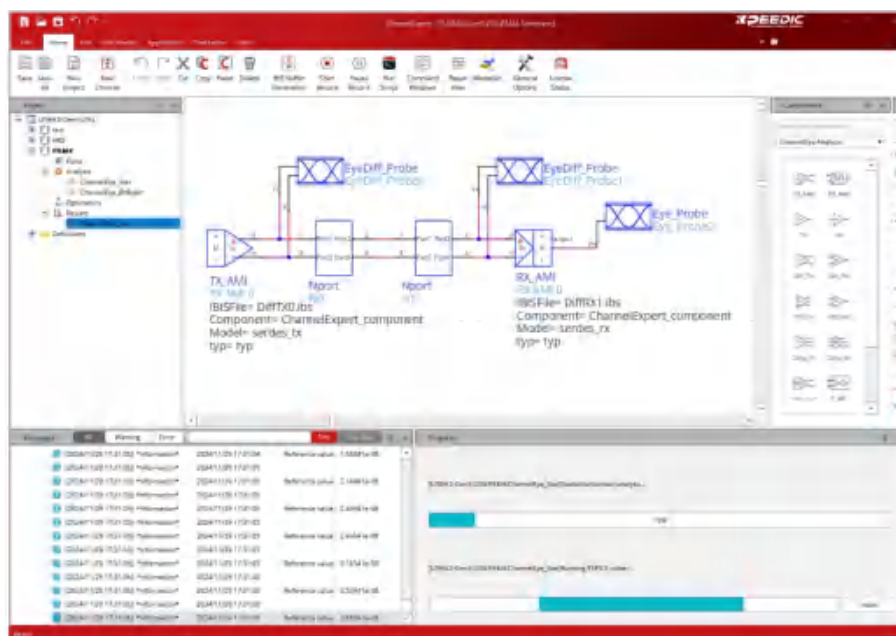
Rx AMI model 自带 Jitter 会自动写入右侧，可以通过勾选开启是否注入抖动。也可以对 Rx 端 jitter 进行参数编辑。Rx AMI Jitter 增加了 CDR 和垂直噪声的种类。



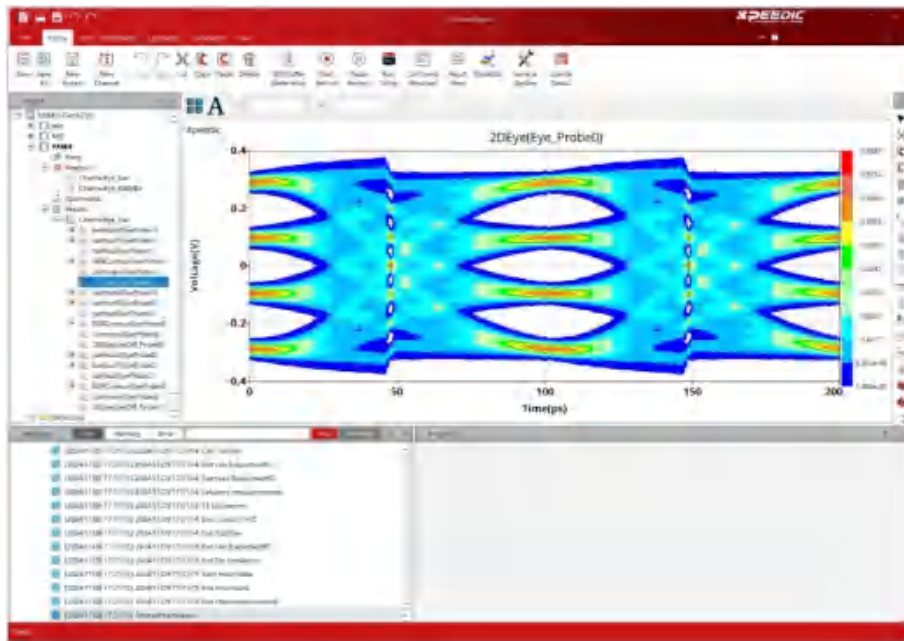
6.6.2.3 ChannelEye 仿真配置

链路搭建完成，增加 ChannelEye 仿真，操作步骤同 TxRx 章节。

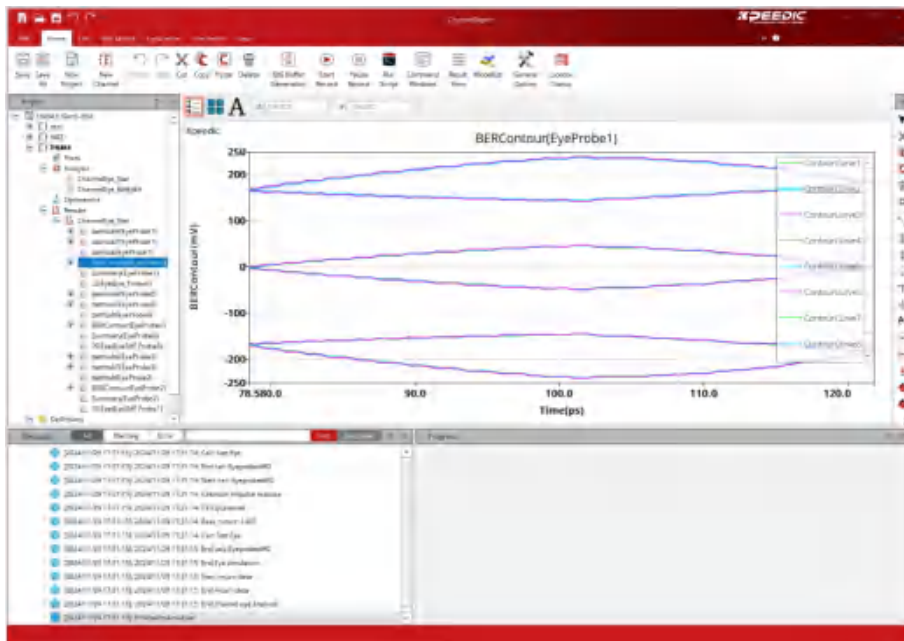
右击 ChannelEye_Stat 弹出菜单栏，点击 Analyze 进入仿真界面。下方 Messages 会显示当前 solver 的 log 信息，进度条展示当前的运行状态。



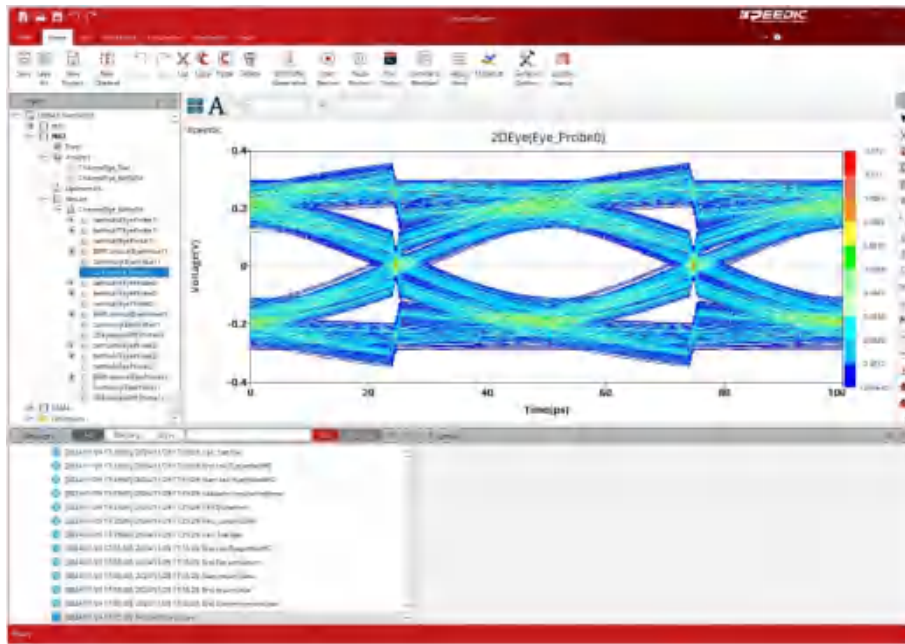
仿真结束，树节点自动挂出 Eye Probe 里添加的曲线类型。下图为 Stat 仿真 PAM4 的 Eye 图。



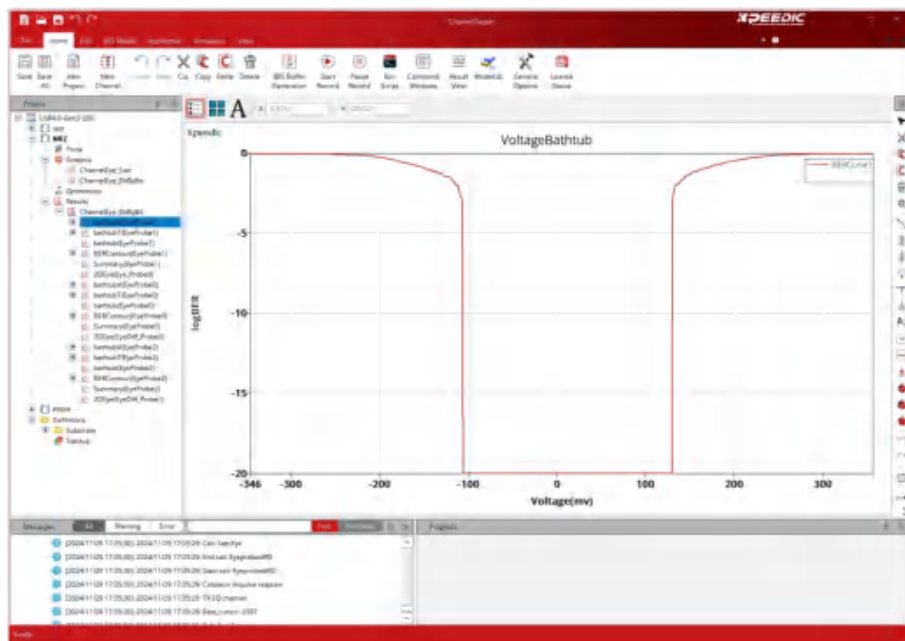
可以通过对 Results 的切换来浏览曲线，下面显示的是 Stat 仿真 PAM4 的 BERContour 图。



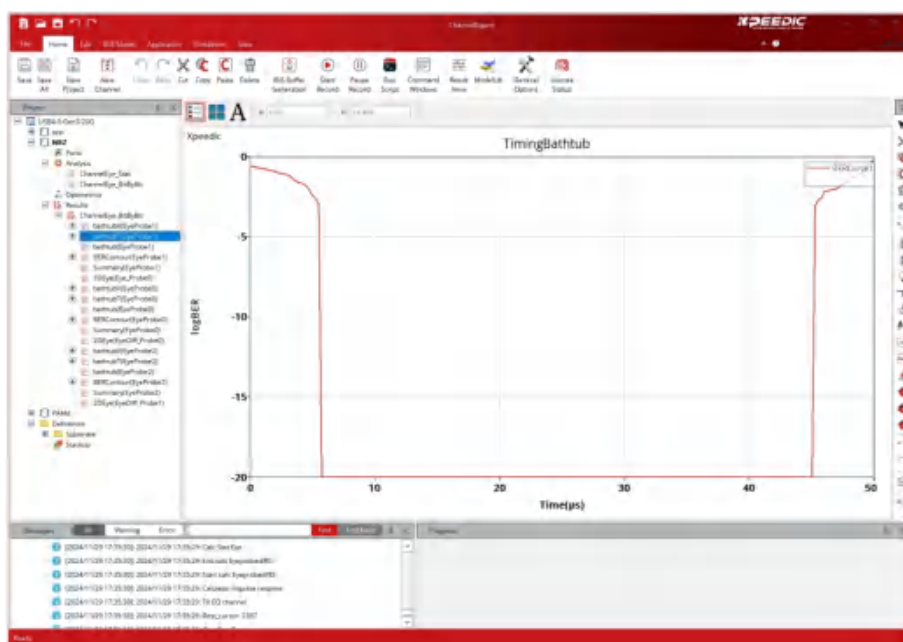
点击树节点下的 Summary，弹出眼高眼宽的 table 数值。PAM4 有上中下三只眼睛，数据会全部统计。



下图为 Bit by bit 仿真的 NRZ 的电压浴盆曲线。



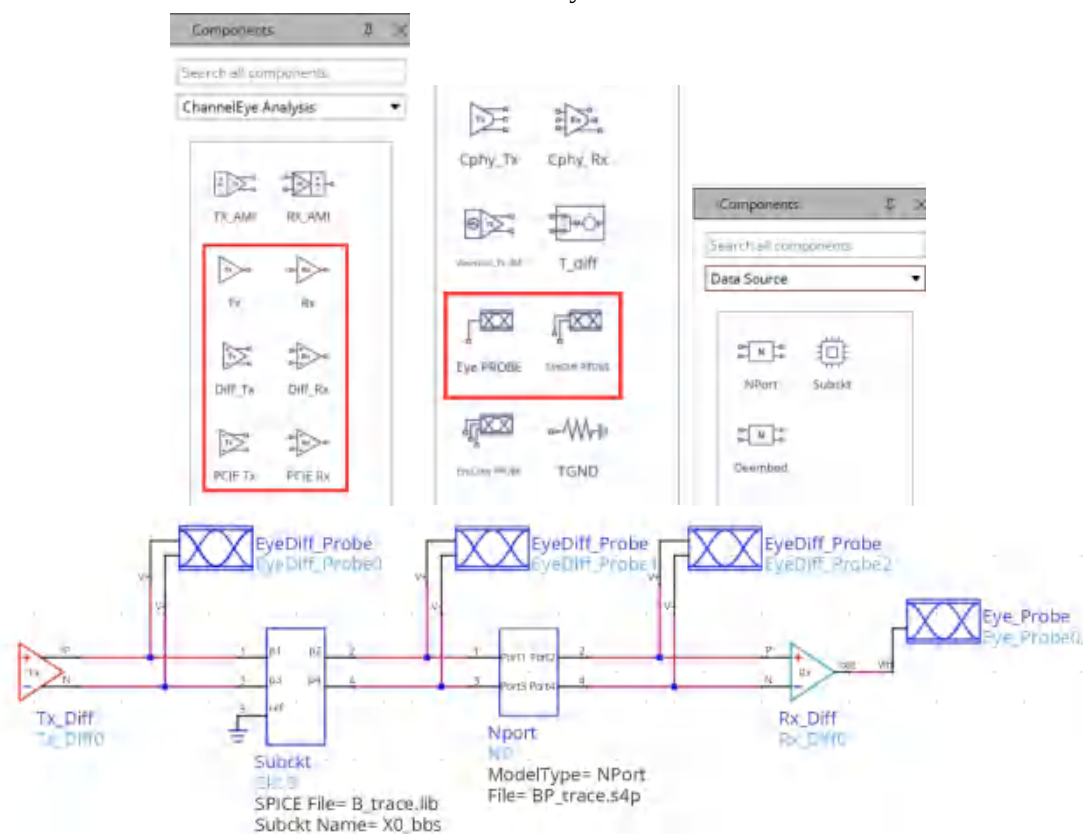
下图为 Bit by bit 仿真的 NRZ 的时间浴盆曲线。



6.6.3 基于理想 Tx Rx 模型的仿真

6.6.3.1 搭建链路

打开 Components 下 ChanneEye Analysis 器件库，选择 TxRx 收发器件，选择 Data Source 下 NPort、Subckt 作为链路通道，增加 Eye PROBE 在需要观测的节点处。



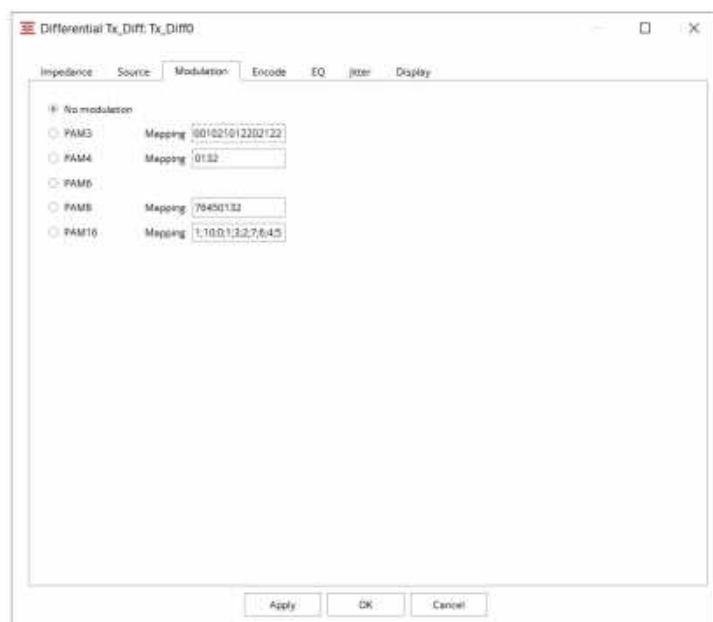
6.6.3.2 Tx 器件设置

双击 Tx 器件，Source 里设置 Data Rate 仿真速率、Trf 上升下降时间、高低电平、Tdelay 延迟时间等。当进行 bit by bit 瞬态仿真，设置下方的码型。

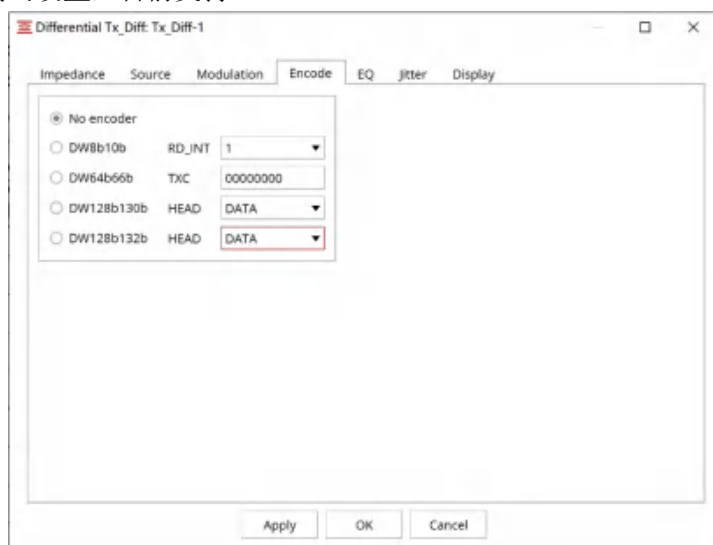
Impedance, 设置 Tx 阻抗值。默认值单端为 50 Ohm，差分为 100 Ohm。

勾选 Enable driven impedance，表示以定义阻抗为准。不勾选阻抗为开路，编辑框变灰。

Modulation, 选择调制模式，默认 NRZ，支持 PAM3、PAM4、PAM6、PAM8、PAM16。

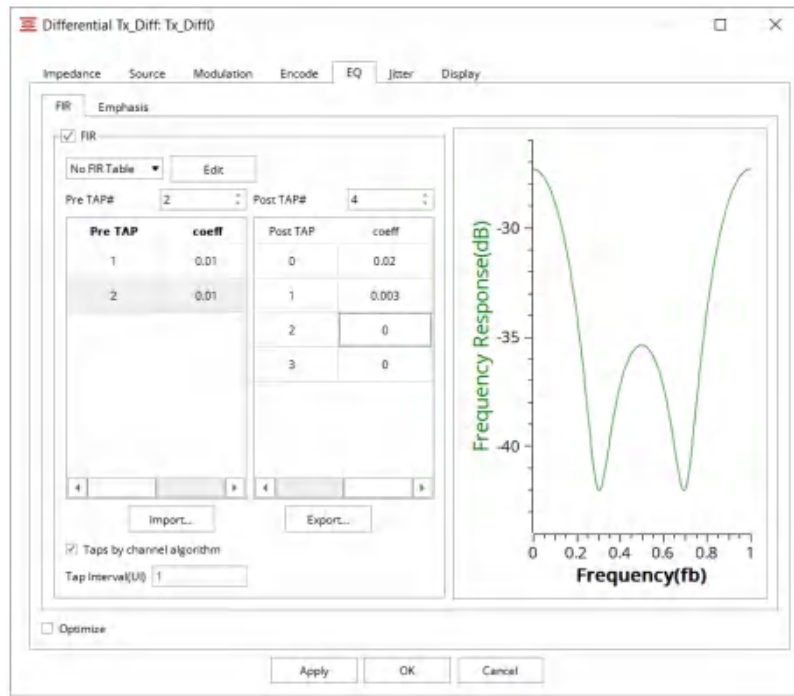


Encode, 编码设置, 目前支持 DW8b10b、DW64b66b、DW128b130b、DW128b132b。

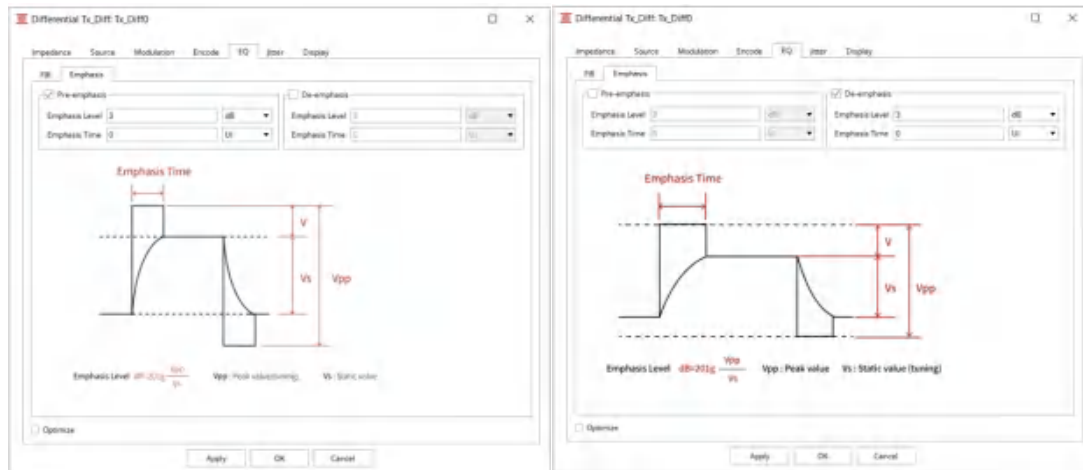


EQ, 均衡设置, Tx 端的均衡有 FIR 和 Emphasis 两种。

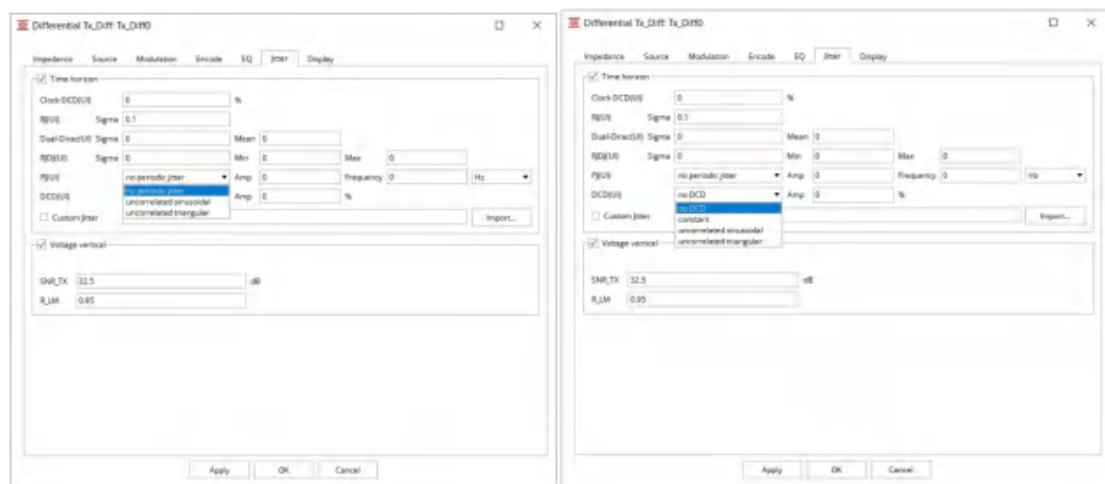
FIR 设置前抽头和后抽头, 系数可编辑, Tap Interval 触头间隔设置, 一般设置 0-1UI。Tap Interval 为 1 时, 等同于 Rx 端的 FFE。也可以通过勾选 Taps by channel algorithm, 在进行一次仿真后返回系统的最优系数 value。



Emphasis 均衡有预加重和去加重两种。



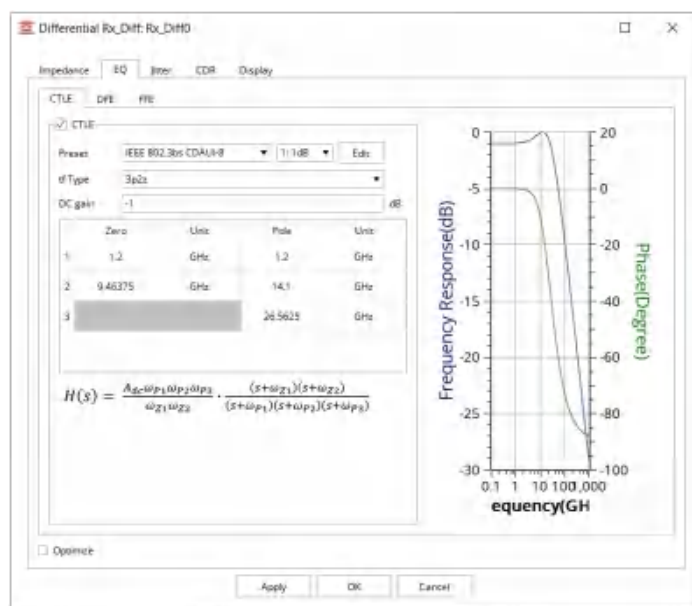
Jitter 增加抖动设置。分 Time horizon 水平抖动和 Voltage Vertical 垂直抖动。Clock DCD: 设置%UI 值;
 RJ: 随机抖动, 设置 Sigma 值;
 Dual-Dirac: 设置 Sigma 值, Mean 值;
 RJDJ: 确定区域的随机抖动, 设置 Min、Max 范围值, 设置 Sigma 值;
 PJ: no periodic jitter 无周期性的抖动 uncorrelated sinusoidal, 不相关的正弦曲线 (需要设置 Frequency 值), uncorrelated triangular 不相关的三角形;
 DCD: no DCD、constant、uncorrelated sinusoidal、uncorrelated triangular。
 支持 Import Custom Jitter。



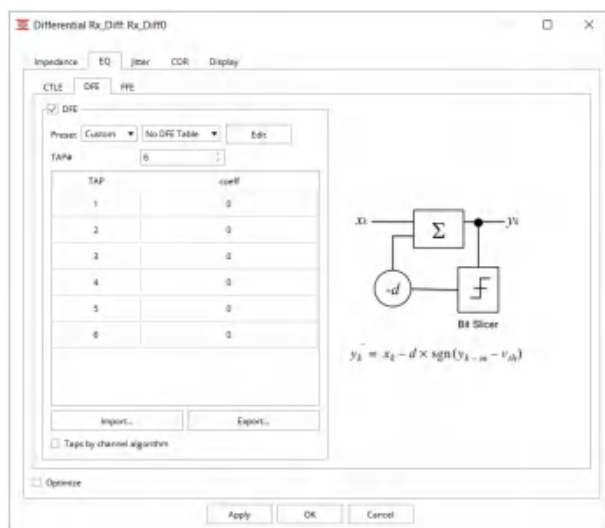
6.6.3.3 Rx 器件设置

双击 Rx 器件，进入 Rx 的 EQ 均衡类型选择。均衡有 CTLE、DFE、FFE 三种。

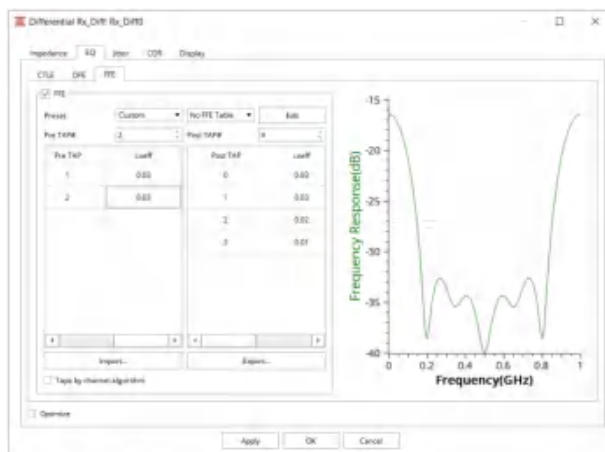
CTLE 内嵌多种协议，可以通过下拉 Preset 选择。Tf Type 下拉选择多种零极点的组合设置，根据参考公式，下方 table 表数据会有不同，同时右侧曲线对应变化。



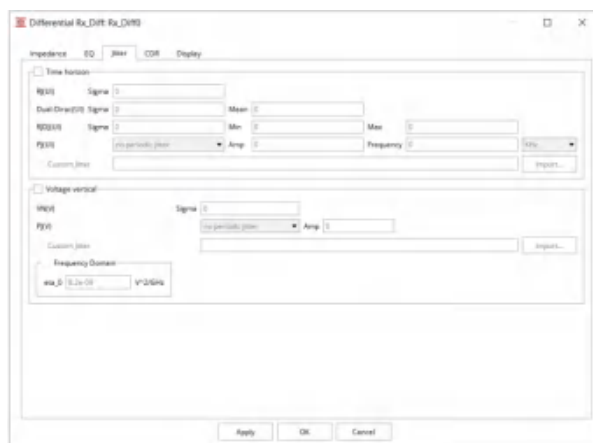
DFE 减少码后干扰的均衡，类似 FFE Post 均衡。



FFE 与 FIR 相同设置相同。设置系数 coeff 值，合理的设置是主抽头 0 系数最大，所有系数绝对值的和接近 1。



Jitter 增加抖动。VN: 电压噪声抖动，设置 Sigma 值。

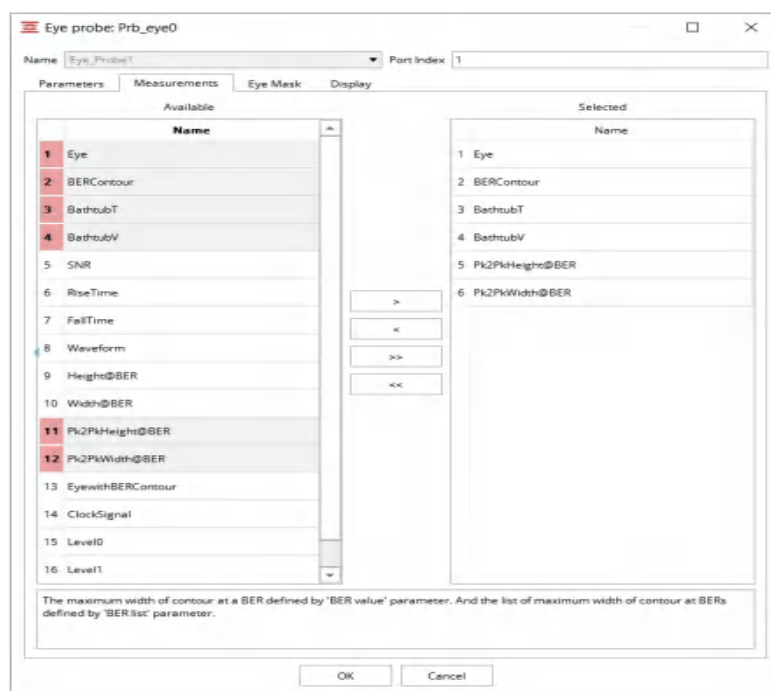


6.6.3.4 Eye Probe 器件设置

双击 Eye Probe 器件，Measurements 支持以下数据类型输出：

- 1、Eye 眼图；
- 2、BERContour 特定误码率下眼图内轮廓曲线；
- 3、BathtubT 时间浴盆曲线；

- 4、BathtubV 电压浴盆曲线；
- 5、SNR 信噪比；
- 6、Rise Time 上升时间；
- 7、Fall Time 下降时间；
- 8、Waveform 差分曲线；
- 9、Waveform_com 共模曲线；
- 10、Height@BER 特定误码率下的水平中心眼高；
- 11、Width@BER 特定误码率下的垂直中心眼宽；
- 12、Pk2PkHeight@BER 特定误码率下的眼高峰峰值；
- 13、Pk2PkWidth@BER 特定误码率下的眼宽峰峰值；
- 14、EyewithBERContour 眼图内轮廓与眼图合成曲线；
- 15、ClockSignal 时钟信号曲线（bit by bit 触发时钟信号时才会输出）；
- 16、Level0 逻辑低电平；
- 17、Level1 逻辑高电平；



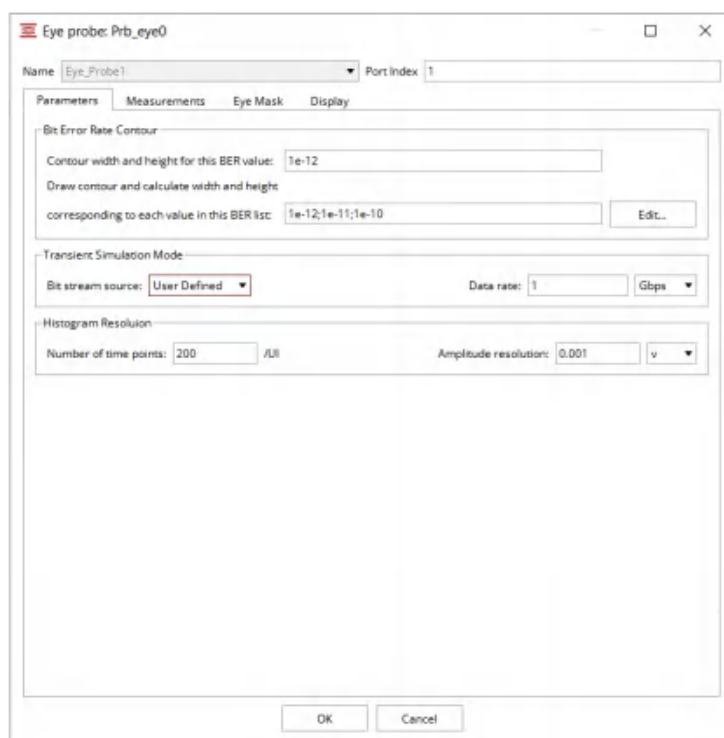
Parameters 设置

- 1、Bit Error Rate Contour 误码率等值线：

Contour width and height for this BER value 设置眼高眼宽对应误码率，Summary 输出的眼高眼宽值为此特定误码率下的值；

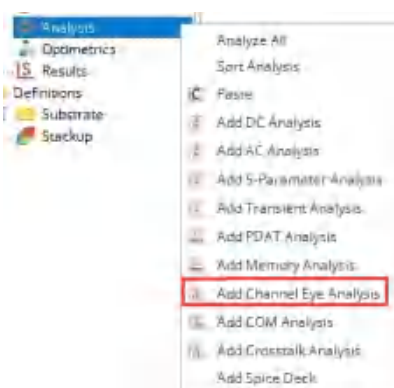
Each value in this BER list 设置不同误码率绘制 Contour 曲线，支持 Edit。可以自定义多个误码率，对应应在 BERContour 曲线图里；

- 2、Transient Simulation Mode 用此器件，需要设置 Data rate；
- 3、Histogram Resoluion 采样点和分辨率设置



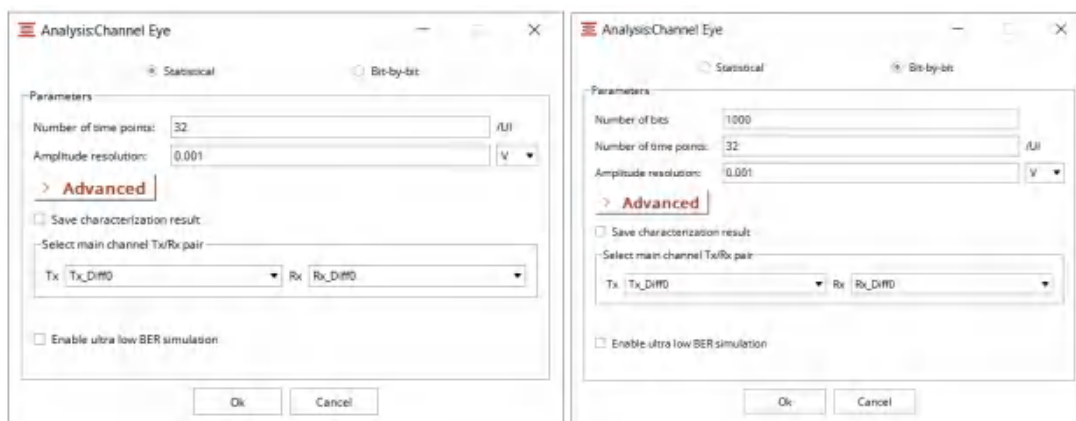
6.6.3.5 Channel Eye 仿真设置

右击树节点 Analysis 增加 Channel Eye Analysis

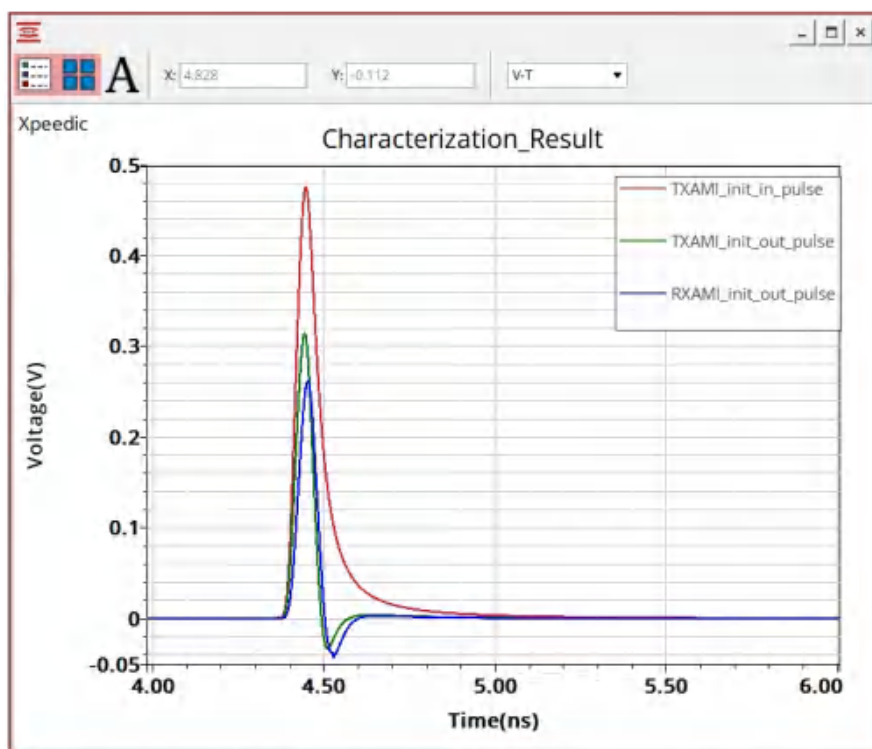


弹出仿真设置界面，仿真分 Statistical 统计仿真和 Bit-by-bit 逐比特仿真。

- 1、设置 Number of time points 采样点数和 Amplitude resolution 分辨率。
- 2、Select main channel Tx/Rx pair，当有多个 Tx/Rx，选择主通道是哪一路。

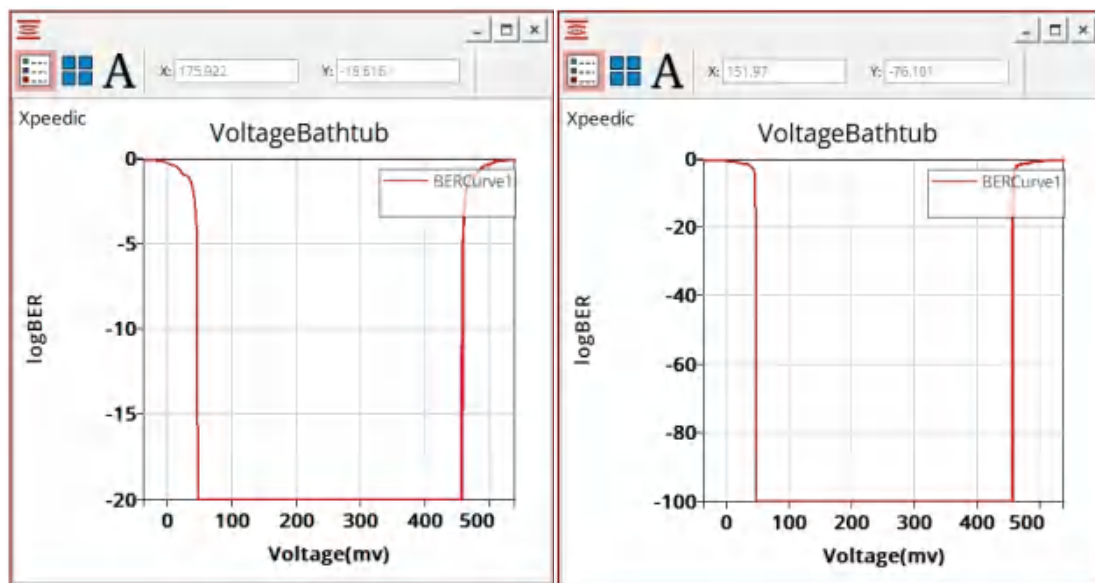


3、勾选 Save characterization result，统计仿真会输出 TXAMI init in pulse 曲线、TXAMI init out pulse 曲线、RXAMI init out pulse 曲线。



4、Bit-by-bit 仿真 Number of bits 设置仿真时间长度。

5、勾选 Enabel ultra low BER simulation，浴盆曲线盆底最低达到 1e-100，默认不勾选，不勾选浴盆曲线盆底最低 1e-20。



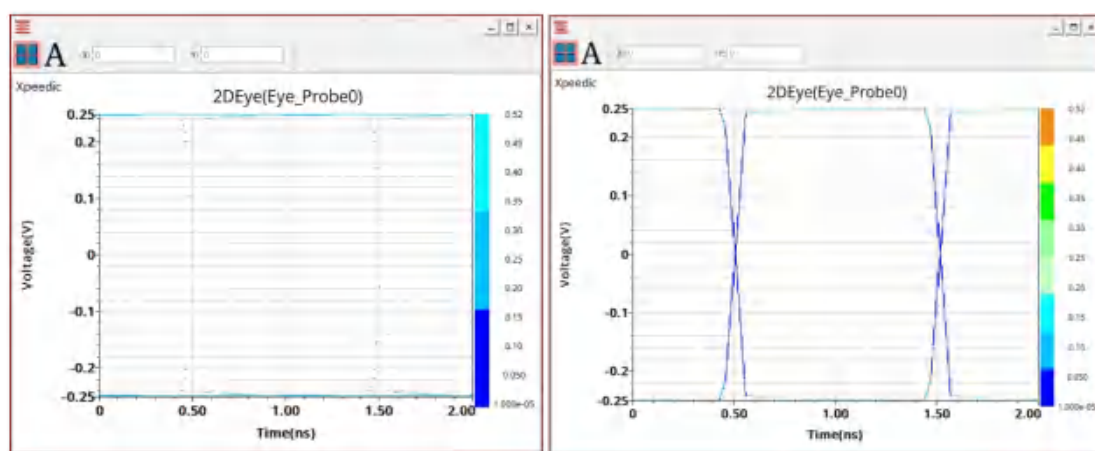
点击 Advanced 为仿真过程高级调节窗口。

Max response length 最大响应时长；

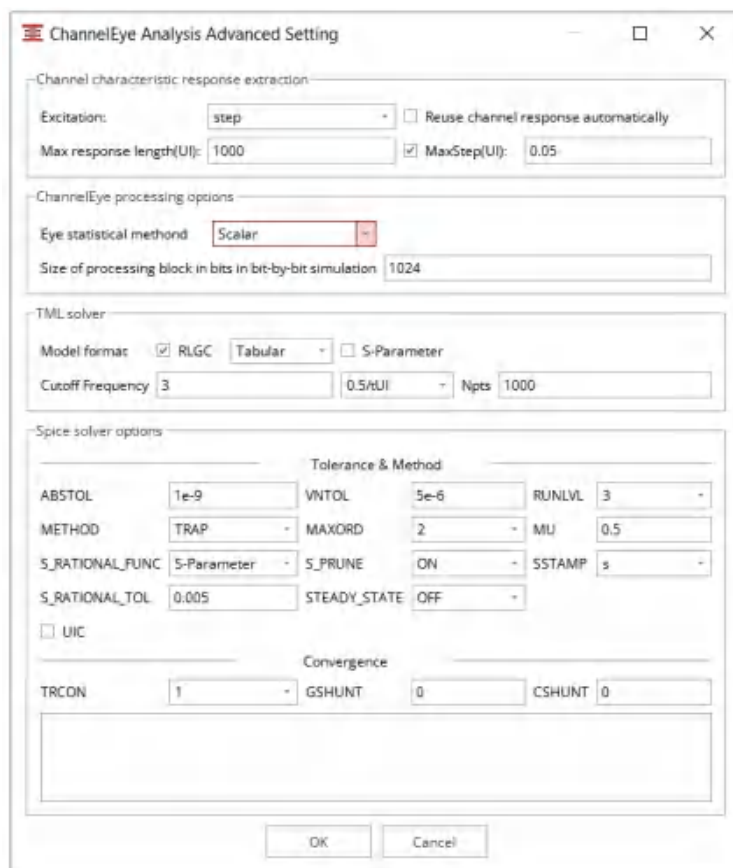
MaxStep 自适应最大步长；

Reuse channel response automatically, 复用 spice solver 产生的阶跃曲线；

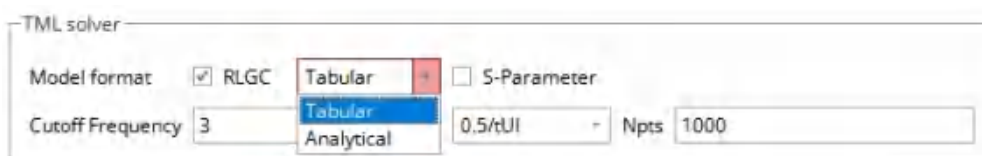
Eye statistical method 默认 Scalar 标量法，统计点的概率密度；Vector 矢量法，统计线的概率密度；



Size of processing block in bits in bit-by-bit simulation, 将码流波形数据按 block 的 bit 数量，分段传递给 AMI dll 函数调用。



TML solver，参与时域仿真的传输线勾选 RLGC，网表以子电路.mod 形式输送。RLGC 分为 Tabular 表格形式和 Analytical 公式两种。勾选 S-Parameter，网表以 S 参数形式输送。Cutoff Frequency 参与时域仿真的传输线以 S 参数输送时的截断频率。Npts，截断后的 S 参数的点数。



6.6.3.6 Spice Option 说明

The image shows a 'Spice solver options' dialog box. It is divided into two main sections: 'Tolerance & Method' and 'Convergence'.
Under 'Tolerance & Method':
- ABSTOL: 1e-9
- VNTOL: 5e-6
- RUNLVL: 3 (dropdown)
- METHOD: TRAP (dropdown)
- MAXORD: 2 (dropdown)
- MU: 0.5
- S_RATIONAL_FUNC: S-Parameter (dropdown)
- S_PRUNE: ON (dropdown)
- SSTAMP: s (dropdown)
- S_RATIONAL_TOL: 0.005
- STEADY_STATE: OFF (dropdown)
- There is an unchecked checkbox for UIC.
Under 'Convergence':
- TRCON: 1 (dropdown)
- GSHUNT: 0
- CSHUNT: 0
At the bottom, there is a large empty rectangular box for manual editing of options.

ABSTOL 电流的绝对误差容限，默认 1e-9A；

VNTOL 电压的绝对误差容限，默认 5e-6V；

RUNLVL 的值越高，准确性越高，模拟运行时间越长；

METHOD 数据积分方法，分 Trap 梯形和 Gear；

MAXORD 数据积分方法的最大阶数；

MU 设置梯形积分的阻尼因子；

S_RATIONAL_FUNC 有三个模式，S-Parameter 即 ifft 网表里用 0 表示，RFM 网表里用 1 表示，BBS 网表里用 2 表示；

S_PRUNE ON 代表 S 参数 Open 的端口去掉，OFF 保留；

SSTAMP 矩阵填充方式，有 S 参数和 Y 参数两种；

S_RATIONAL_TOL 当 S_RATIONAL_FUNC 为 RFM 时生效，调整 RFM 精度；

STEADY_STATE 阶跃的稳态检测 OFF 关闭 ON 生效；

TRCON 瞬态仿真过程中收敛行为的自动控制；

GSHUNT 每个节点（node）添加一个连接到地（ground）的导纳（或电导）；

CSHUNT 电容接地。

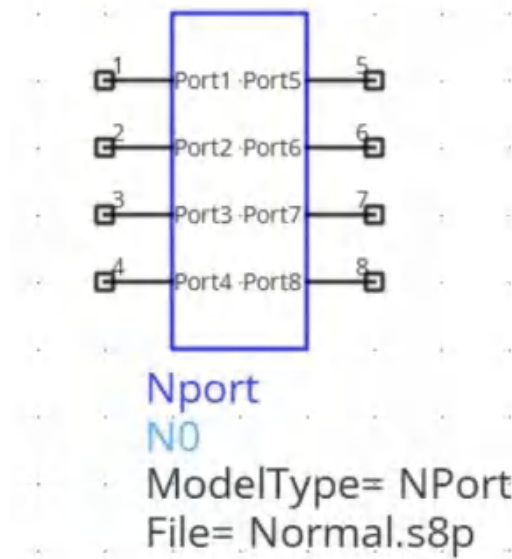
界面下的方框可以手动编辑增加 option。

6.7 COM 通道操作裕度分析

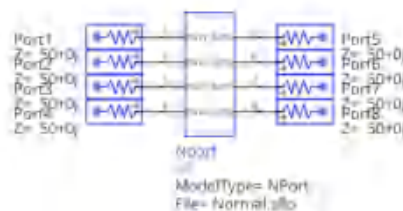
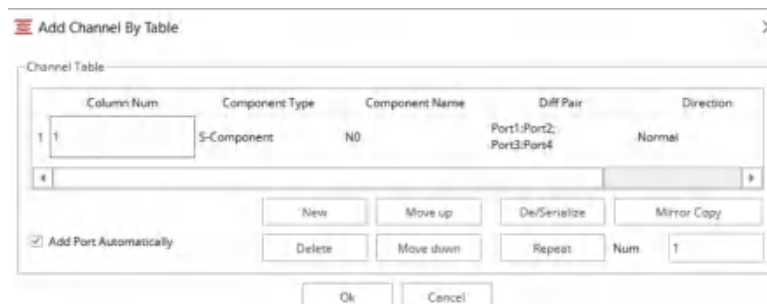
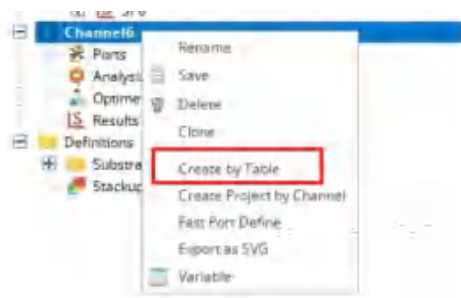
针对 com 分析，需要准备 S 参数文件。

6.7.1 搭建仿真链路

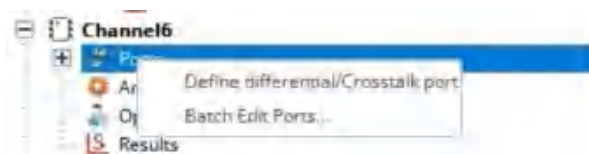
从元件库 Data Source 下拖入 Nport 到画布中，加载 S 参数文件。



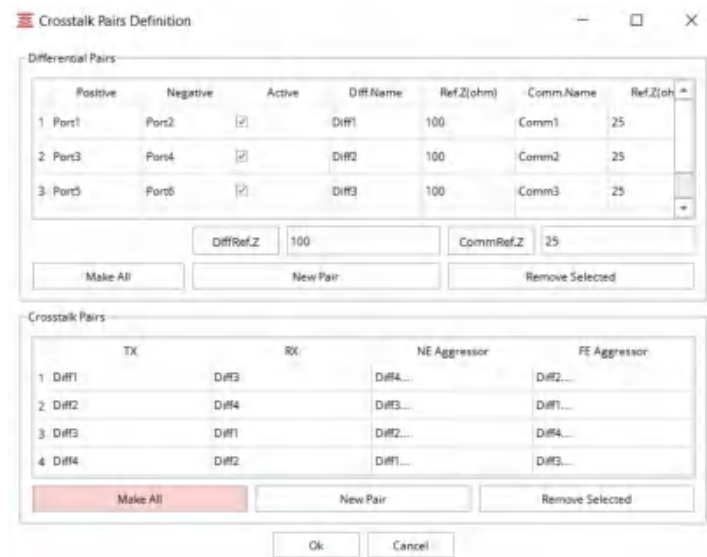
在 Channel 节点右键选择 Create By Table 选项，使用默认配置后确认，完成 Nport 的 port 自动添加。



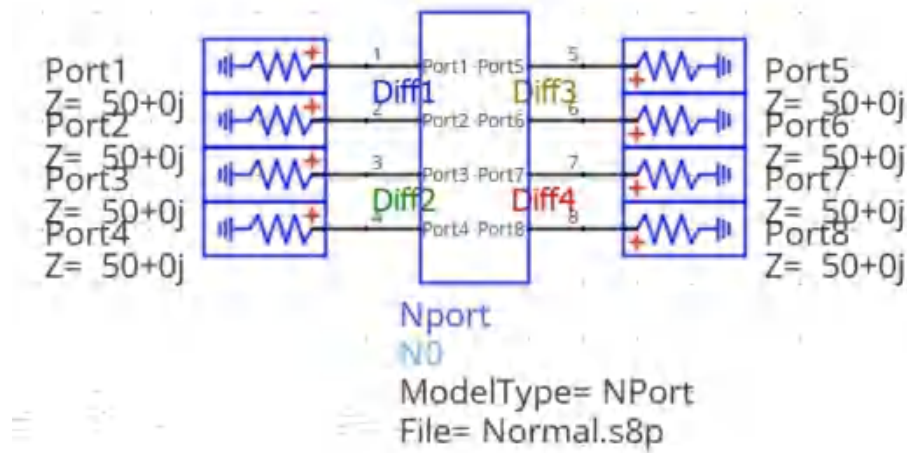
在左侧树节点 Port 处右键，选择 Define differential/Crosstalk port 选项。



在差分信号配置界面，在 Differential Pairs 下点击 Make All，再点击 Crosstalk Pairs 下的 Make All。也可以通过 New Pair 自定义差分对设置，同时自定义差分对阻抗等相关参数。

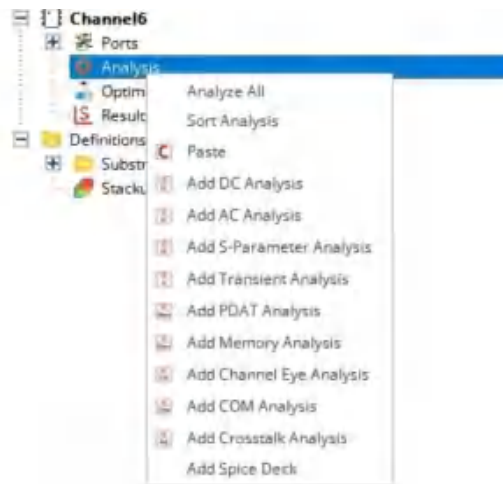


Com 仿真链路搭建完成，如图所示：

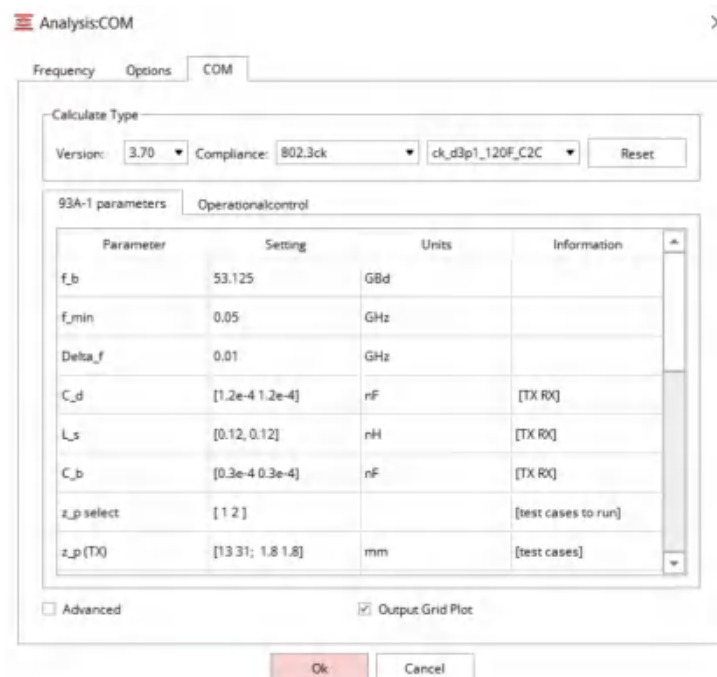


6.7.2 设置 COM 协议参数

添加 COM 仿真。

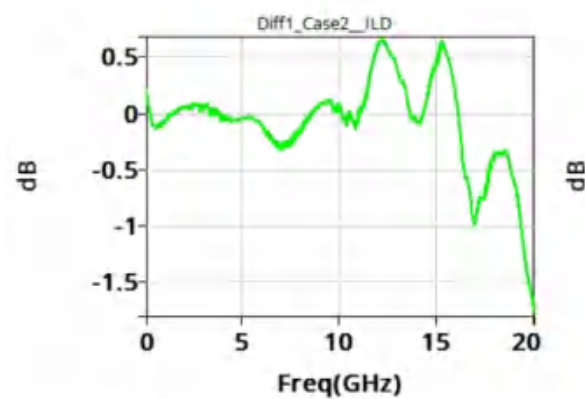


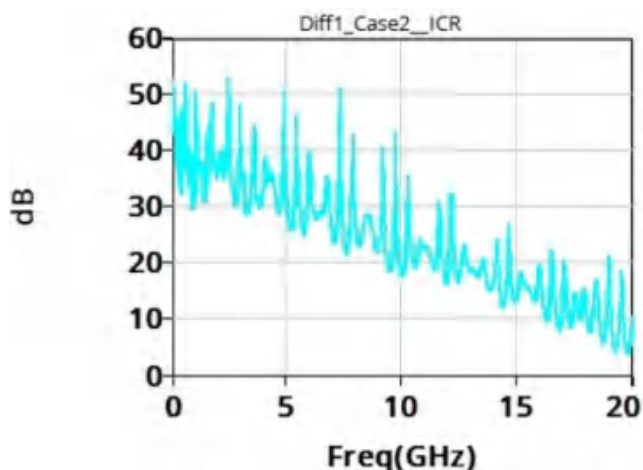
COM 仿真界面设置仿真协议参数。



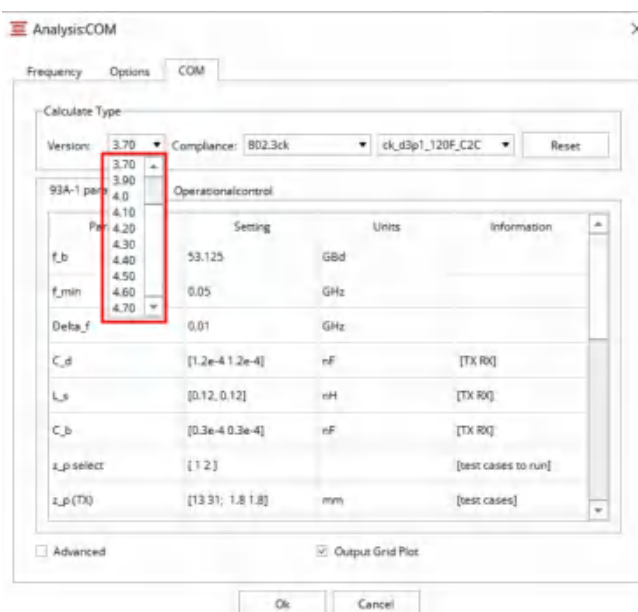
6.7.3 COM 仿真结果查看

完成仿真后，可在 Result 中查看 ILD，ICR 等结果。





编辑仿真界面，切换协议类型，可以仿真生成不同的结果。

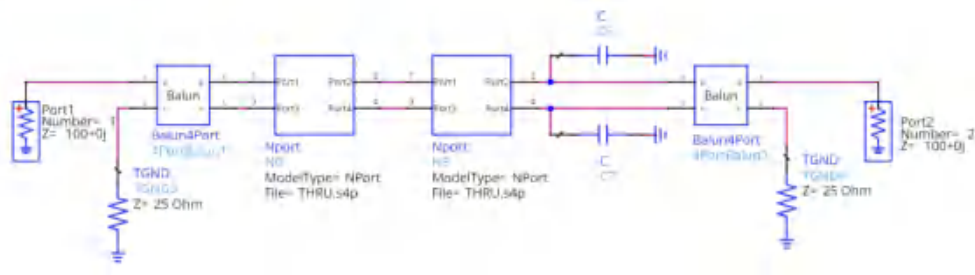


6.8 用户自定义后处理计算—ICN

6.8.1 搭建 S 参数仿真链路

分别从元件窗口拖入 TML、Balun4 与通道，建立连接关系，S 参数仿真电路如下图所示。Balun4 器件与 TML 器件与 Nport 器件级联，并在两端连接 Port 器件进行 S 参数分析。

分别从 Components 下 TML 元件窗口拖入 TML、Lumped Components 原件窗口拖入 Balun4、电容 C，选择 Data Source 下 Nport 器件作为通道，建立连接关系，并在器件两端增加 GND 与 Port：



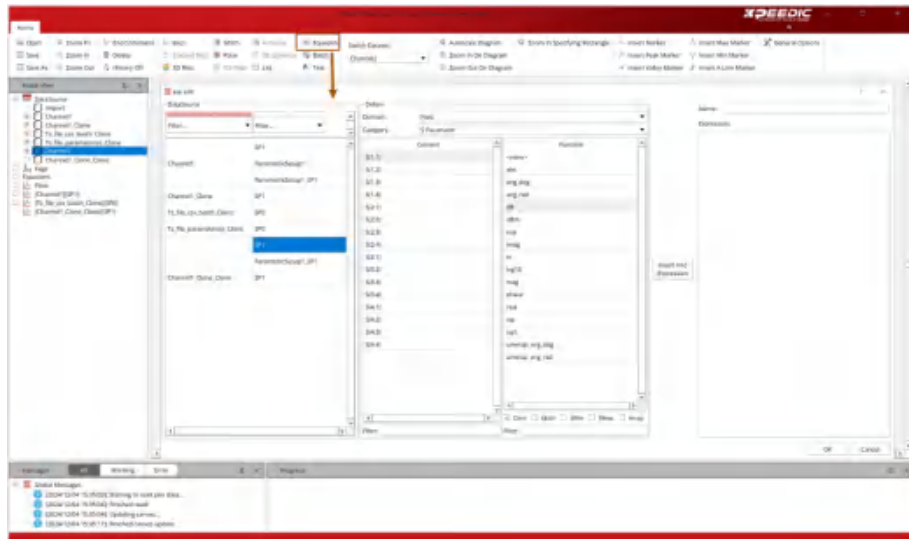
搭建电路结束后，对该电路进行 S 参数仿真；

6.8.2 电路内部设置

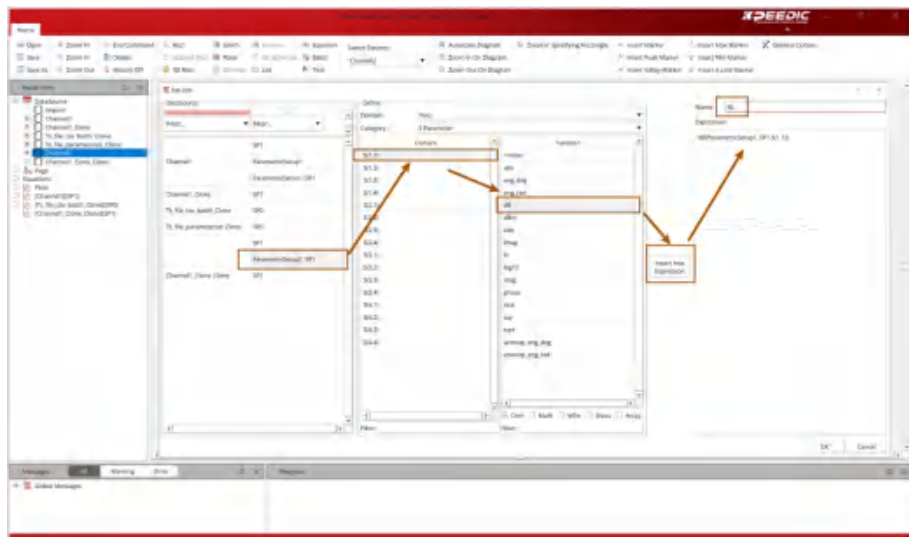
仿真结束后，点击菜单栏 **Result View** 进入后处理编辑界面，初次弹出时，后处理界面会默认显示仿真产生的 S 参数曲线：



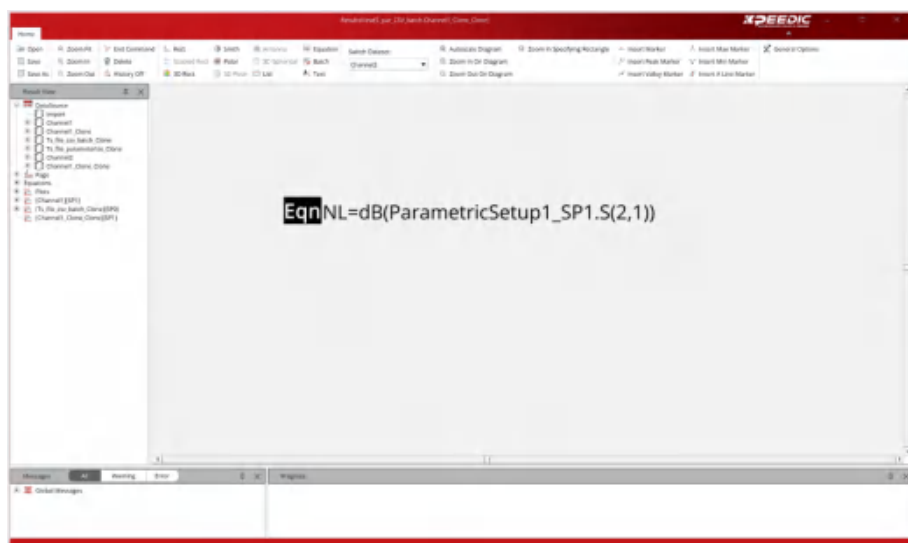
点击菜单栏 **Equation** 按钮，弹出公式编辑界面：Data Source 处显示的是工程名以及创建的仿真，Define 处显示的是具体的 S 参数结果，Function 处是软件自带的公式，包括 Cmn、Math、Means、Array 四种，界面右侧为公式编辑处：



选择需要计算的公式，通过 Insert Into Expression，可以将公式导入，也可以手动输入，在 Name 处输入公式的名字：



完成后，在后处理界面显示编辑的公式：注，如果公式正确，Eqn 为黑色显示，如果公式错误，则 Eqn 为红色显示。



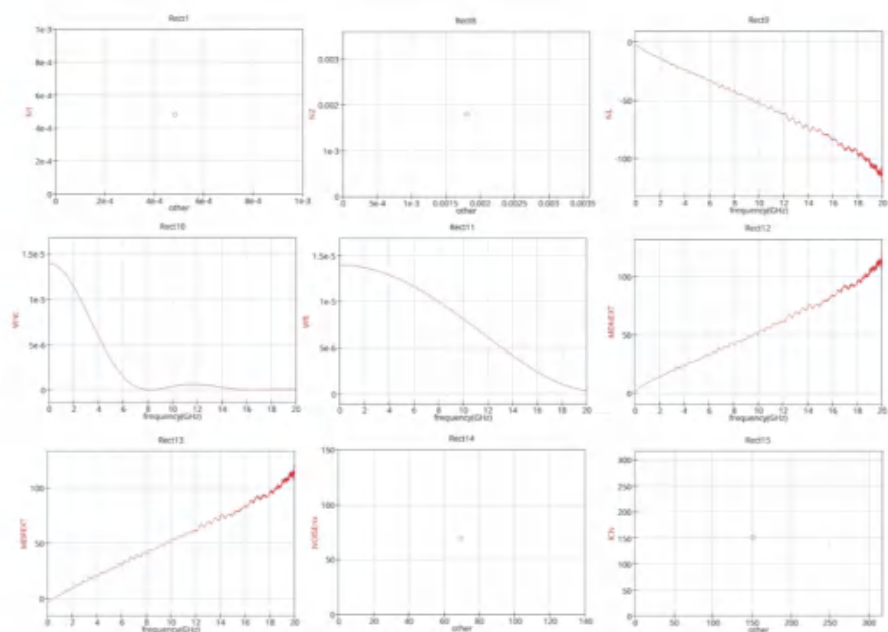
1

[illegible]

输入 ICN 计算公式:

[illegible]

229 / 307



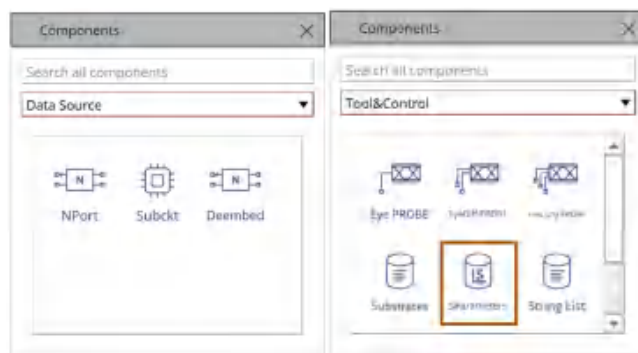
通过对曲线和对数值分析，可以用于评估多个串扰源对单一接收器的干扰影响。

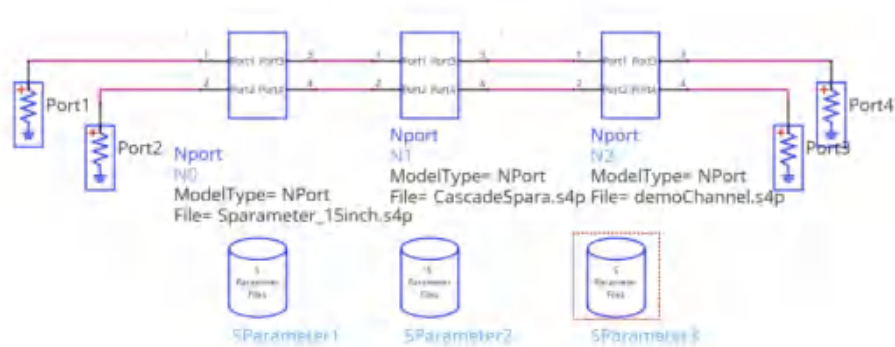
6.9 高级仿真分析

6.9.1 参数扫描 Sweep

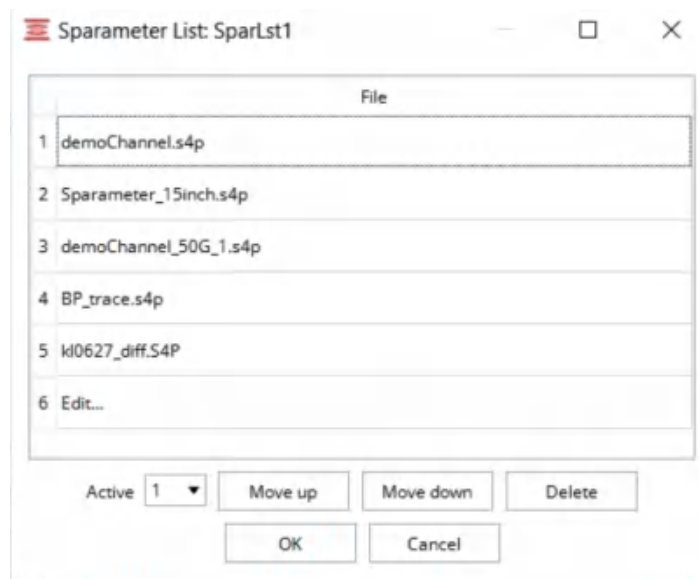
6.9.1.1 搭建链路并设置扫描参数

1、从元件库创建多个 Nport 到原理图上，分别加载 S 参数，添加 port 后连线。从 Tool&Control 中拖取 S parameter list 到原理图上搭建仿真链路。

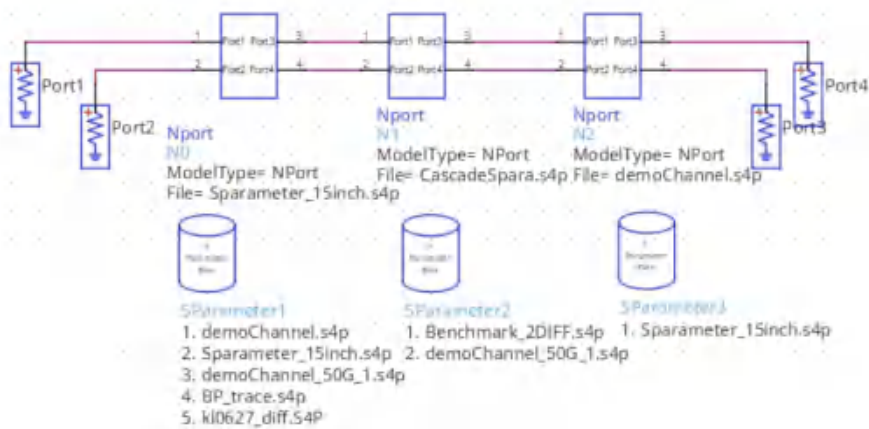




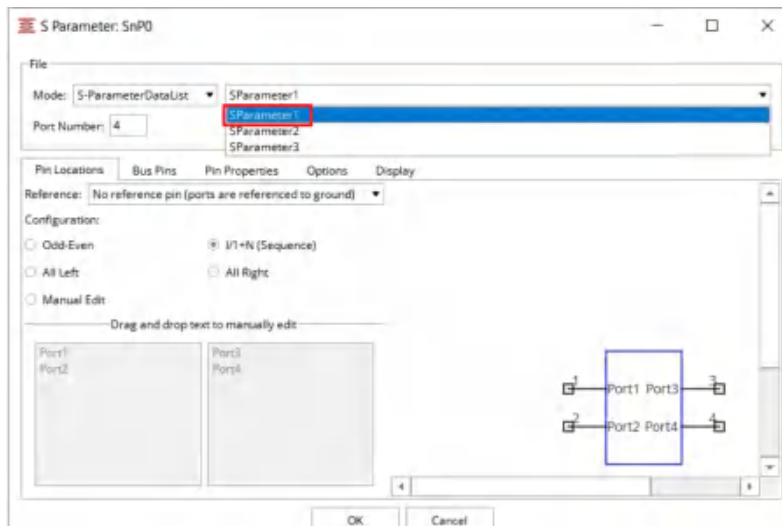
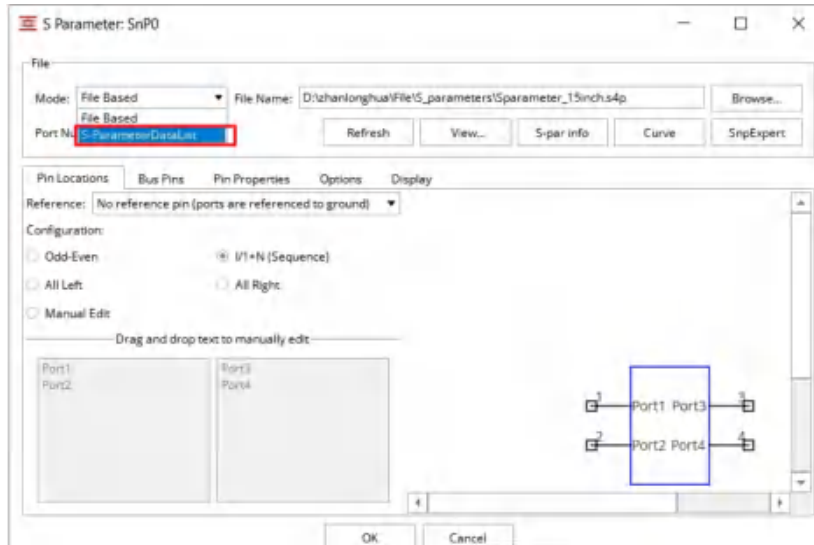
2、双击 Sparameter1，点击 Edit，可以给桶加载多个 S 参数。可以对加载的 S 参数文件做 move up，move down 操作，也可点击 active 选择使能的 S 参数是哪一个。



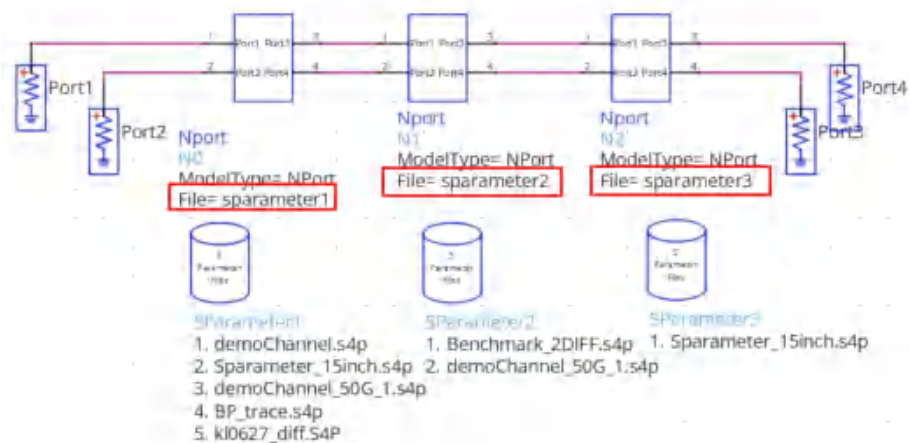
3、依次对 S parameter list 进行编辑配置。



5、 点击 N0，选择 S parameterDataList，选择 SParameter1。

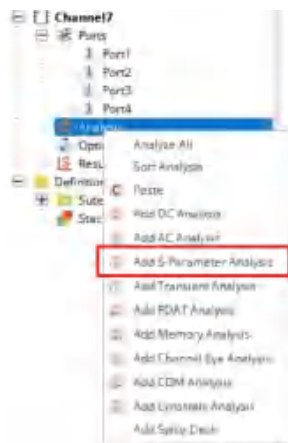


6、依次设置 N1 和 N2。

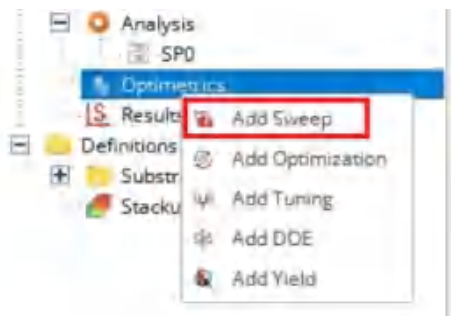


6.9.1.2 创建 sweep 扫描仿真

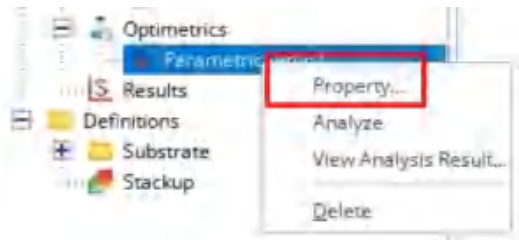
1、在 Analysis 右键选择 Add Sparameter Analysis，并设置仿真参数。



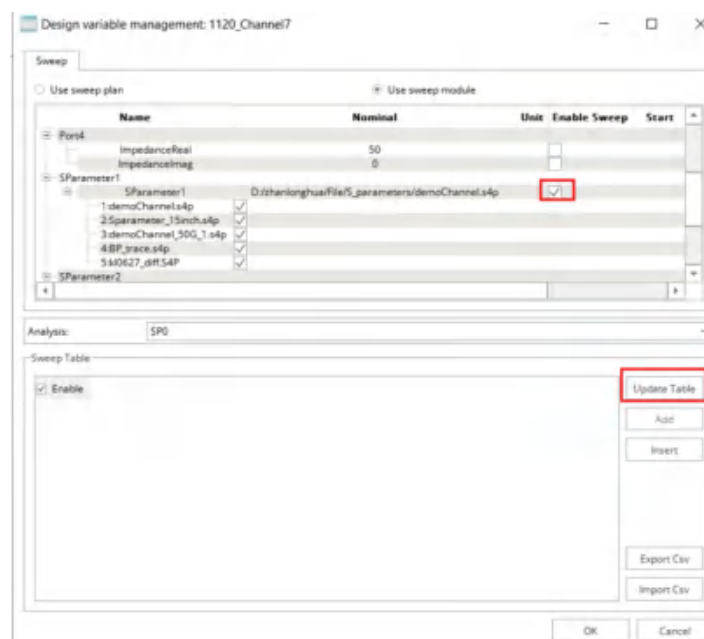
2、在 Optimetrics 右键点击 Add Sweep。

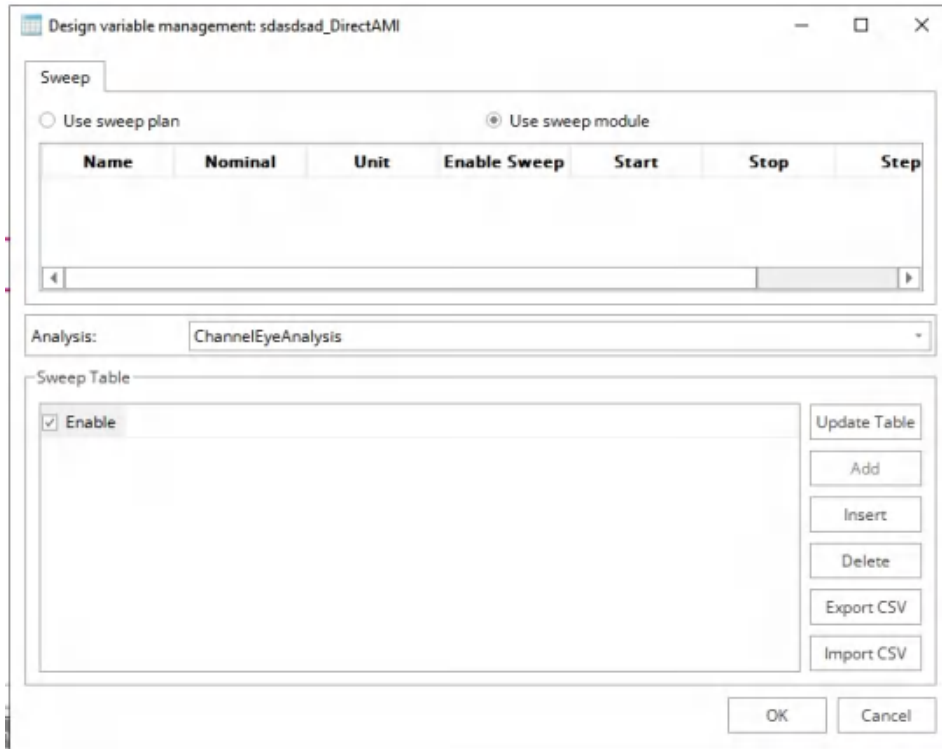


3、在 ParameterSetup1 右键点击 Property。

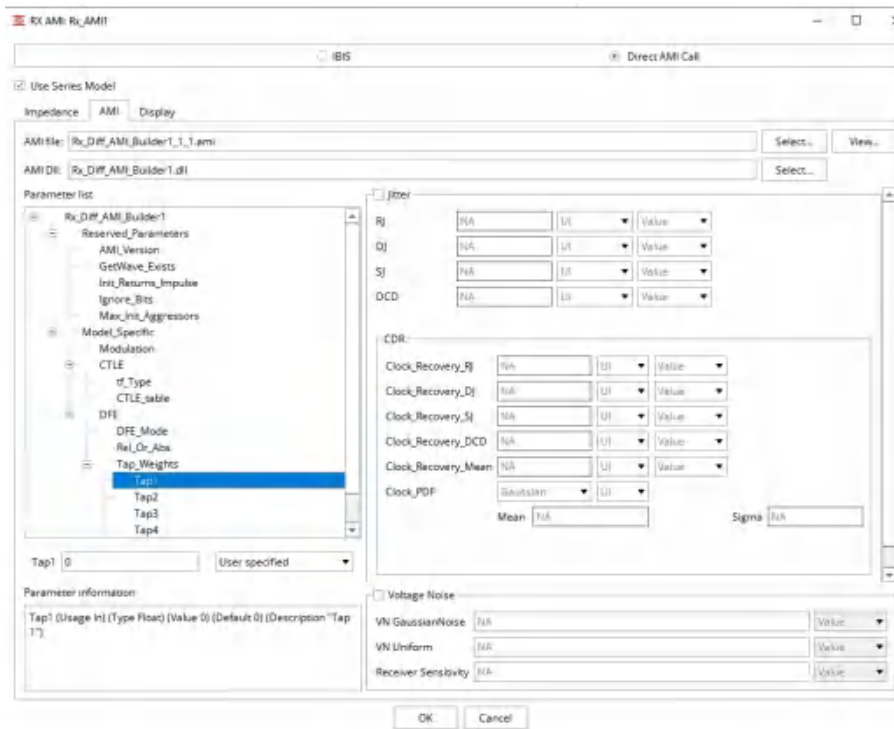


3、勾选需要仿真的参数后面的 enable，点击 update 更新到 sweep table 中。

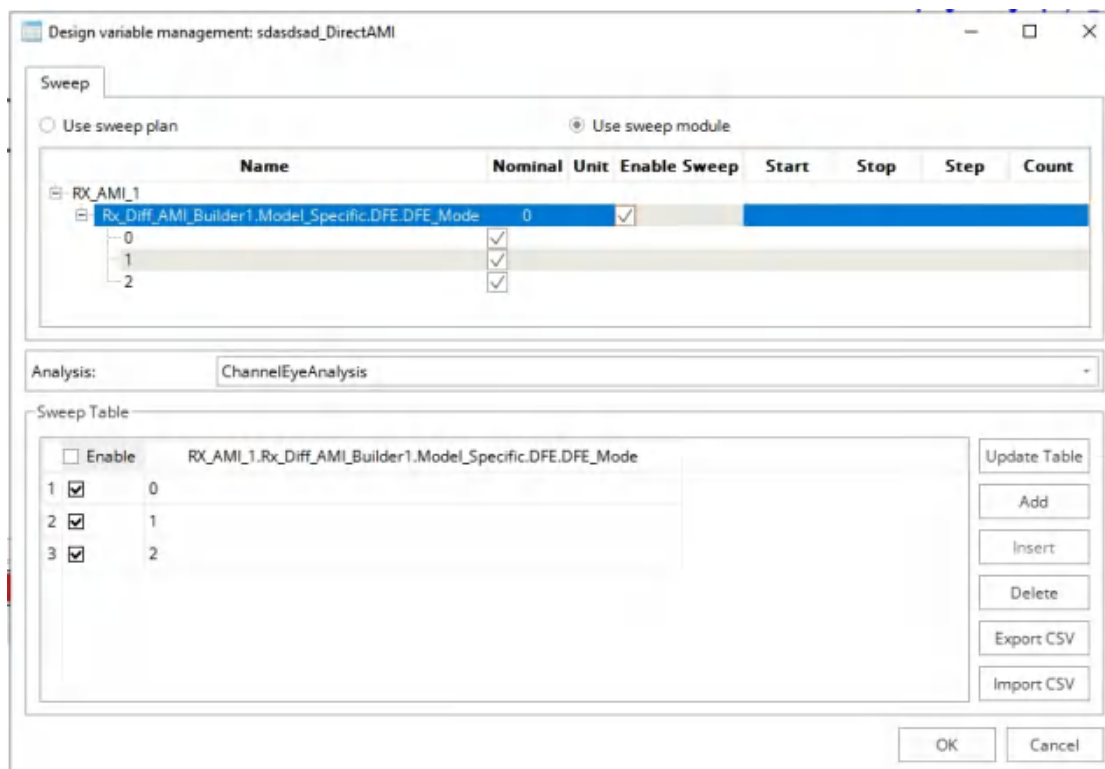




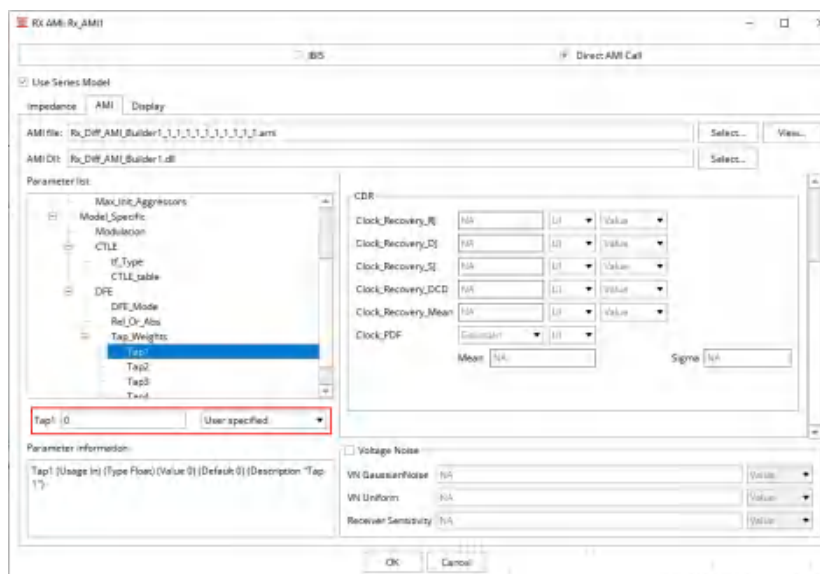
对于 ami 的参数需要加入 sweep 的，点击需要使用的参数，切换类型为 user specified。

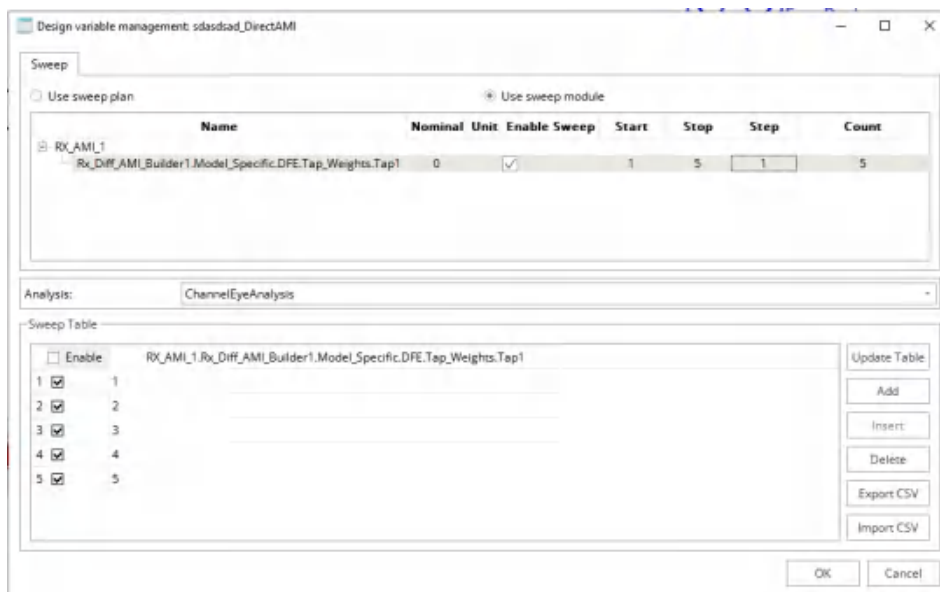


模式切换完成后，点击 ok，在 sweep 的属性配置界面中就会出现该参数。



ami 参数为 float 类型时配置如下



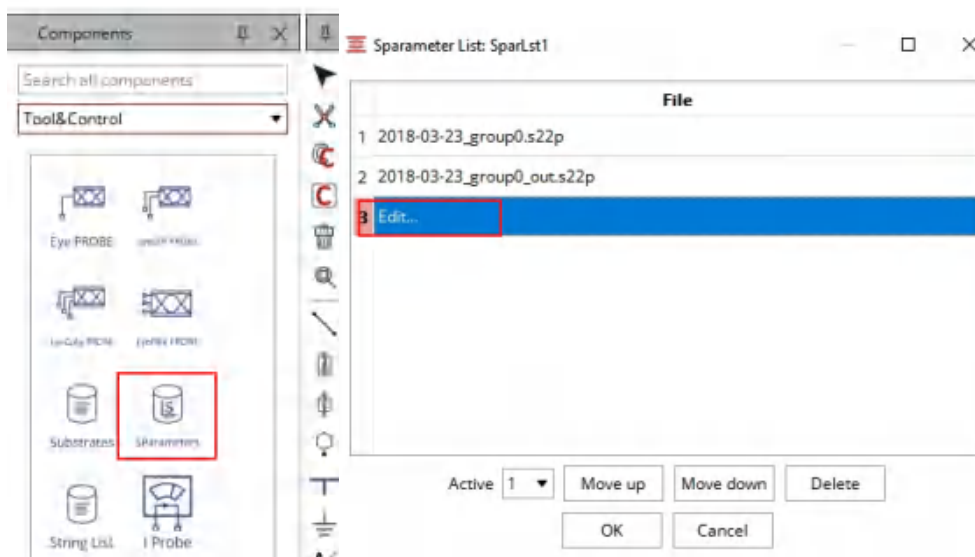


配置完成后，扫描和查看结果操作同 6.9.1.1-2。

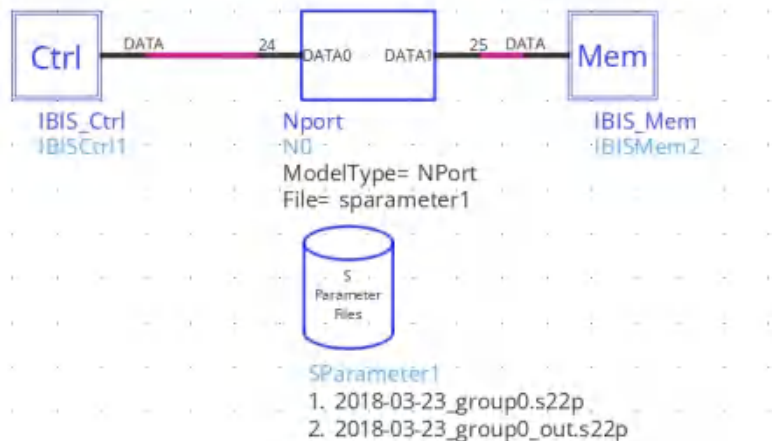
6.9.1.5 ddr sweep reuse

ddr sweep

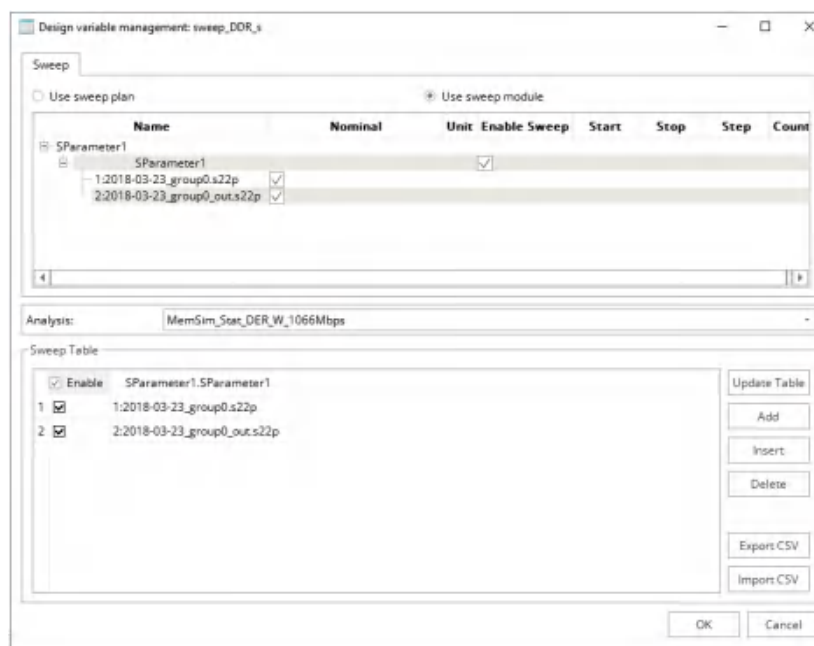
ddr 仿真通道为 S 参数时，需要做 sweep 仿真时，需要从器件库中拖拽 S parameters 到原理图中。双击 Edit 加载 S 参数。



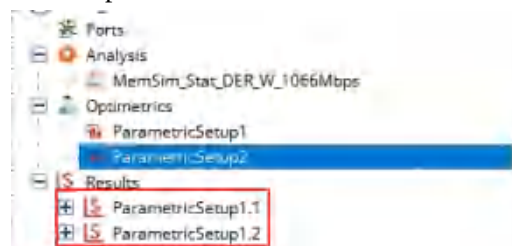
搭建的完整链路如下。



添加 sweep 仿真节点后，在仿真节点处右键编辑配置如下。

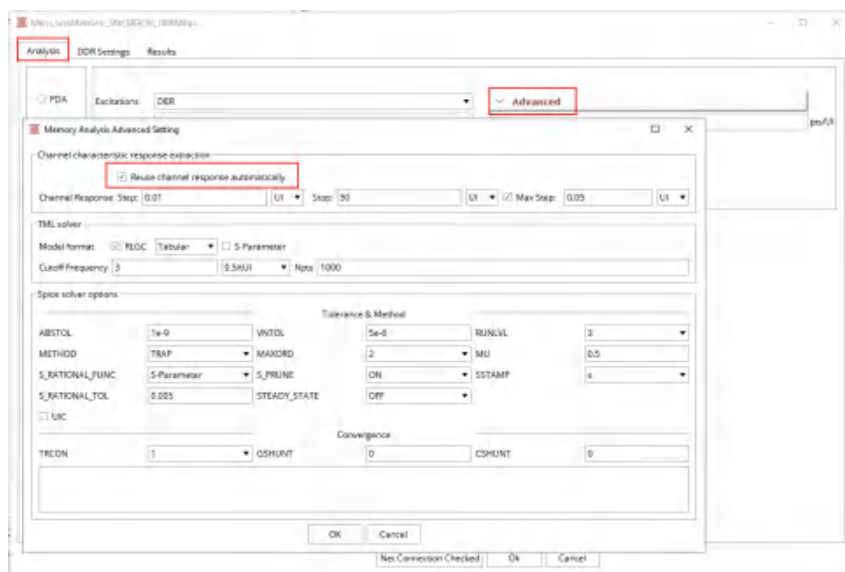


仿真配置完成后，启动 sweep 仿真。

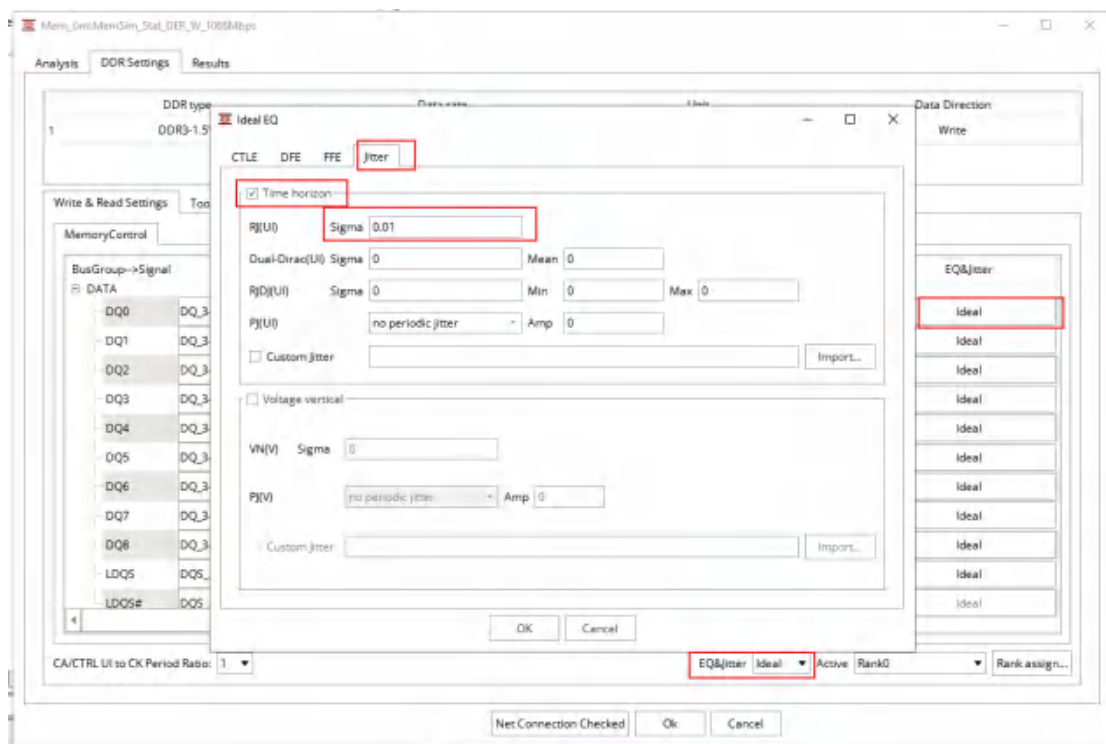


使用 ddr sweep reuse 功能，需要所使用的仿真节点已经产生 sweep 优化的结果，否则在配置 reuse 时会报错。

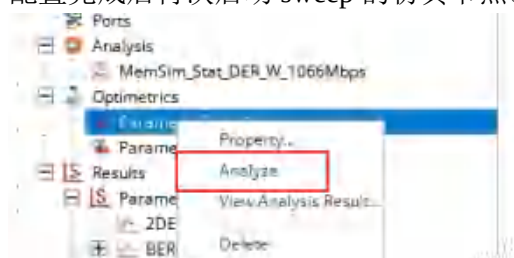
当仿真结果存在时，点击 normal flow 的仿真节点右键，在仿真节点 Analysis 中点击 advanced，勾选 reuse channel response aotomatically。



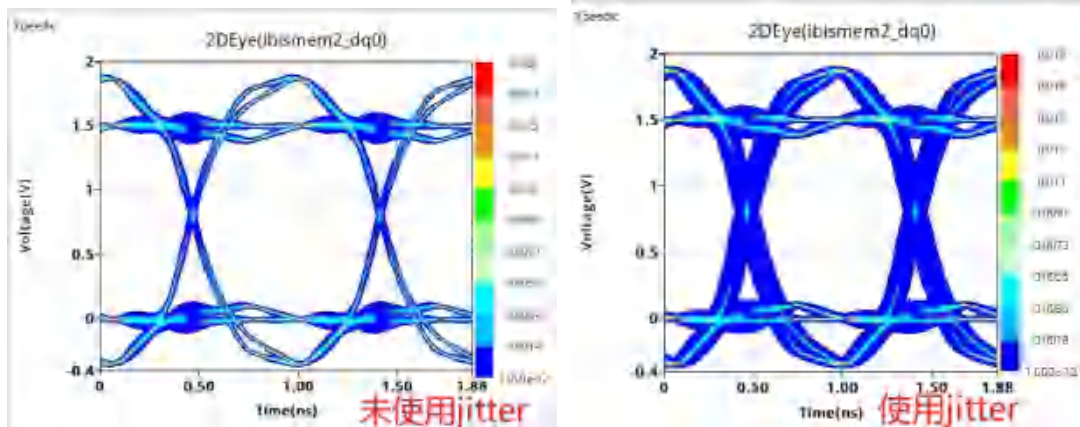
在 DDR setting 中，EQ&Jitter 选择 Ideal，点击需要更改配置的信号后面的 Ideal，在弹出的配置界面中选择 Jitter，勾选使用 Time horizon，配置相关的参数后点击确认，也可以配置使用 CTLE/DFE/FFE。



配置完成后再次启动 sweep 的仿真节点。



仿真完成后即可查看使用 jitter 的参数话结果。



6.9.2 参数优化 OPT

6.9.2.1 OPT 仿真链路搭建及设置

使用 1 中搭建的 Normal flow 仿真链路进行 opt 分析。

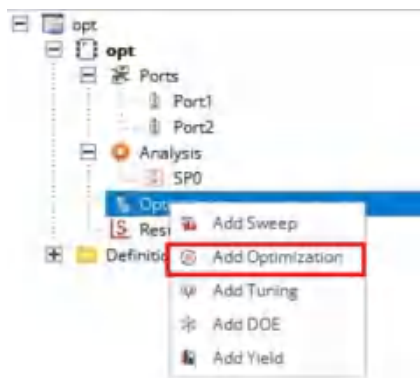
工程树节点选择 Optimetrics，点击菜单栏的 Simulation，选择 Design Variable。



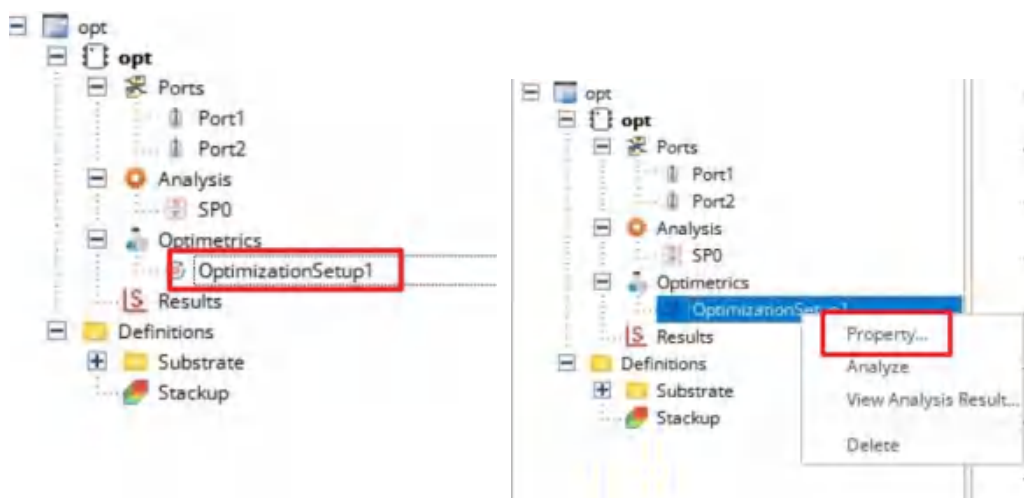
在 design variable 设置界面选择 Optimization，在需要参与优化的参数后面的 enable 列中打钩，完成后点击 ok。



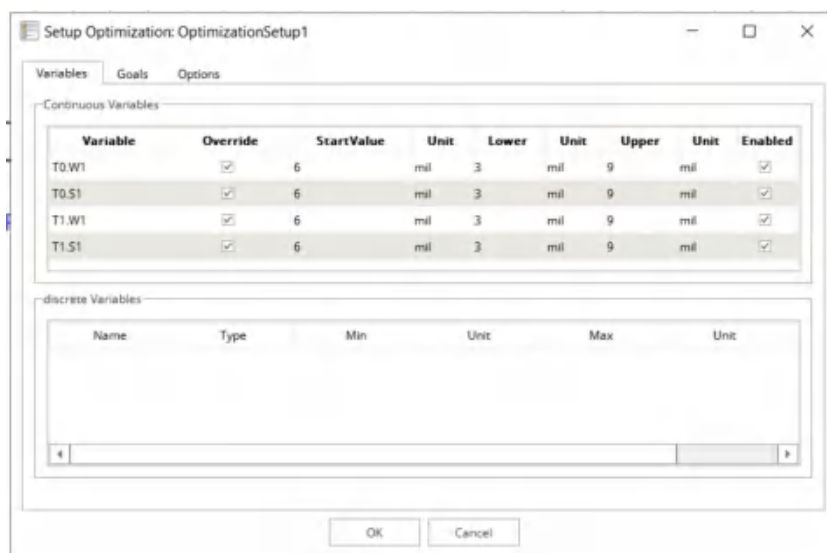
在工程节点下的 Optimetrics 下右键，选择 Add Optimization。



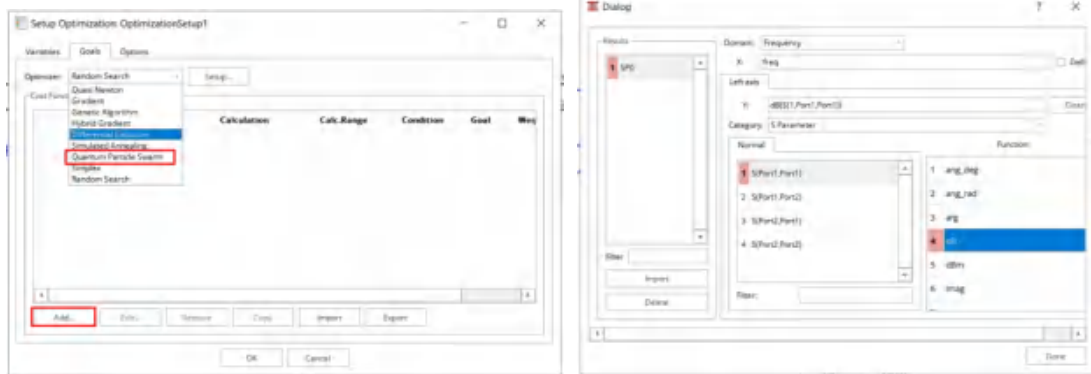
在生成的 OptimizationSetup1 右键，点击 Property。



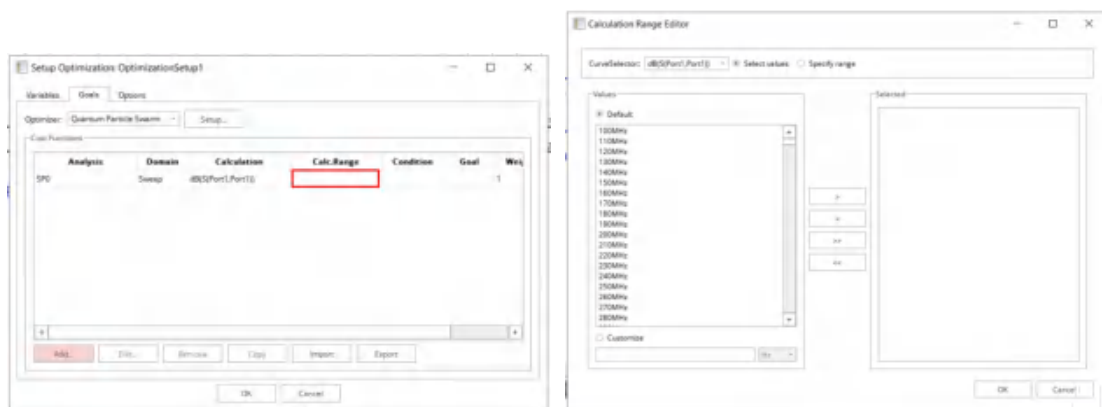
在 OPT 属性编辑界面 Variable 中会显示选择的属性参数，



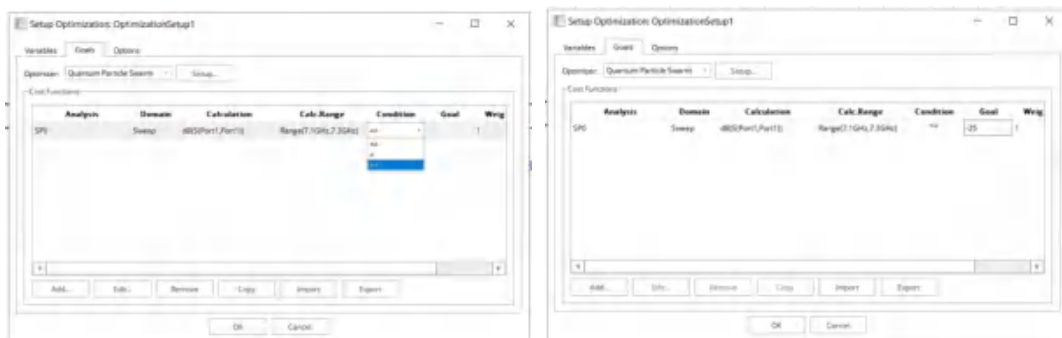
点击 Goals，选择算法类型后点击 add。对弹窗中选择结果节点后点击 Done。



点击 Calc.Range 下面的空白处，可以设置手动选择频点方式，自定义频点方式或者频率范围方式，以选择范围方式为例。

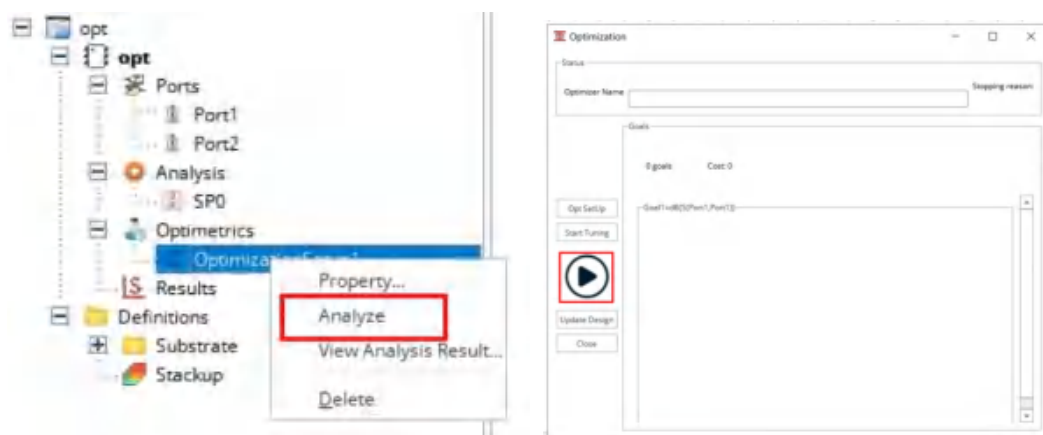


设置频率范围后点击 ok。点击 condition 选择目标达成的条件，在 goal 下设置目标值。在目标编辑界面可以同时添加多个目标，添加目标方式相同。可以复制、删除、导出、导入目标。

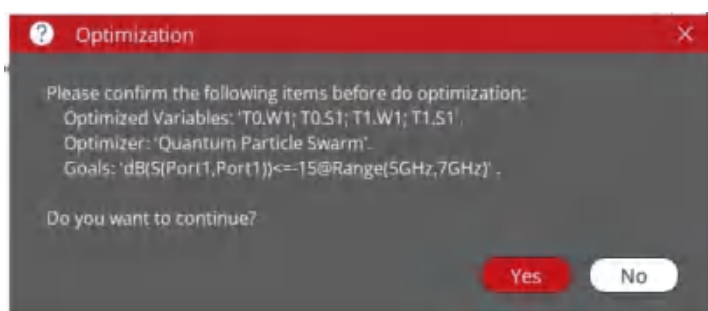


6.9.2.2 OPT 仿真及结果查看

在 OPT 仿真节点右键选择 Analyze，在仿真界面点击启动选项，开始进行 opt 优化。



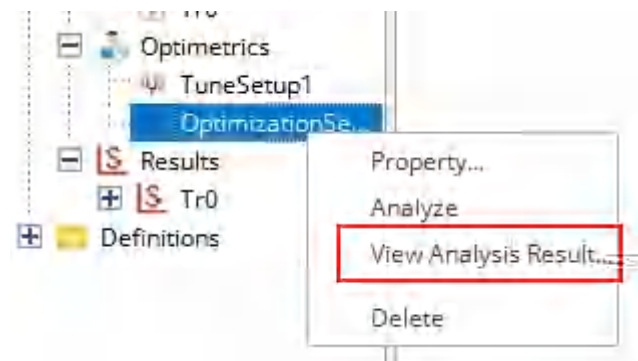
启动仿真后，会弹出界面展示当前次仿真的参数和目标设置情况，点击 Yes 继续。

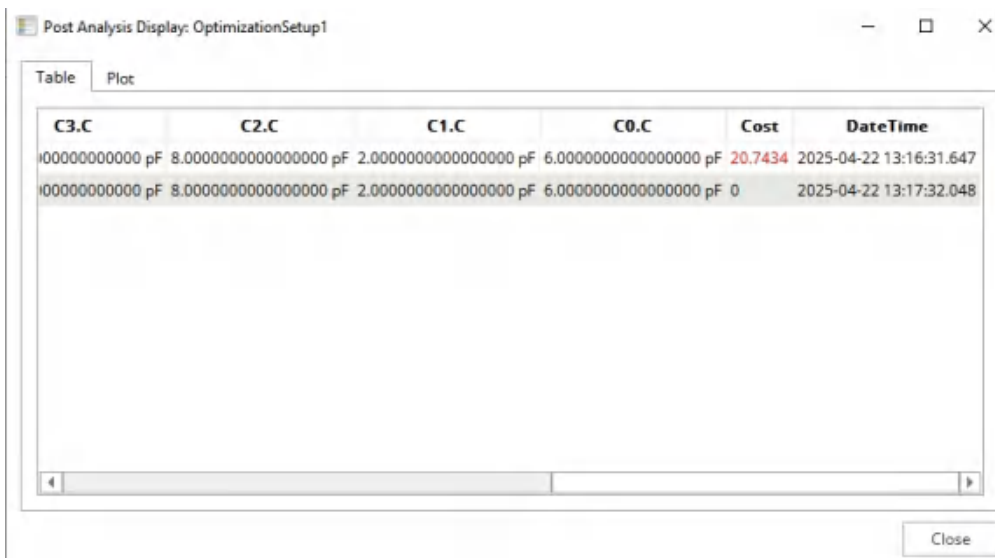


仿真启动后，会在 opt 界面动态展示优化结果，显示轮询次数、误差值等信息。



仿真完成后可以在仿真节点 View Analysis Result，查看结果。





Post Analysis Display: OptimizationSetup1

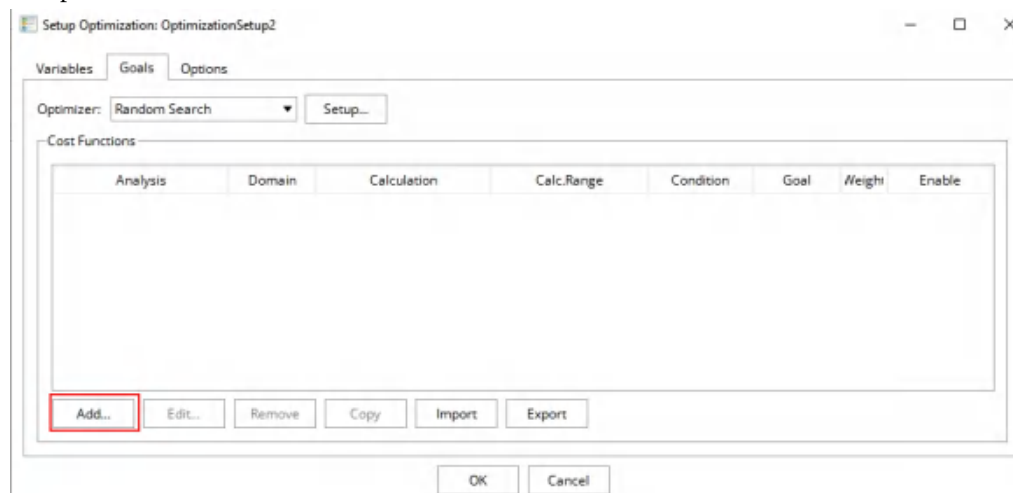
C3.C	C2.C	C1.C	C0.C	Cost	DateTime
100000000000 pF	8.0000000000000000 pF	2.0000000000000000 pF	6.0000000000000000 pF	20.7434	2025-04-22 13:16:31.647
100000000000 pF	8.0000000000000000 pF	2.0000000000000000 pF	6.0000000000000000 pF	0	2025-04-22 13:17:32.048

Close

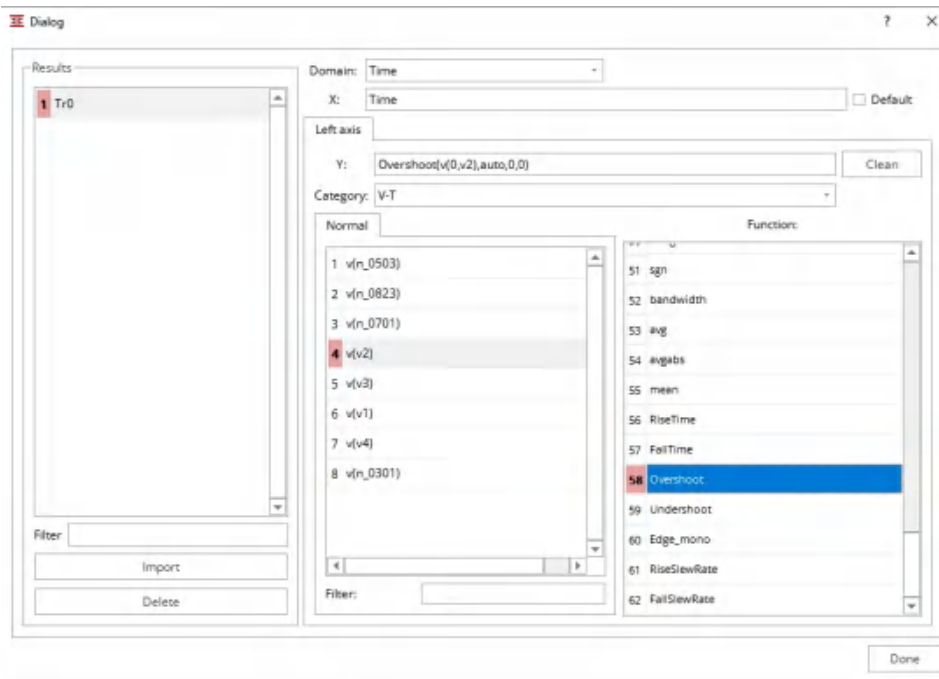
6.9.2.3 OPT 仿真支持测量函数

时域仿真需要使用测量函数作为目标时，可以在 goals 中添加 Overshoot(上冲)/Undershoot (下冲)/FailTime (下降时间)/RiseTime (上升时间)/Edge_mono (单调性)，添加方式参考如下操作。

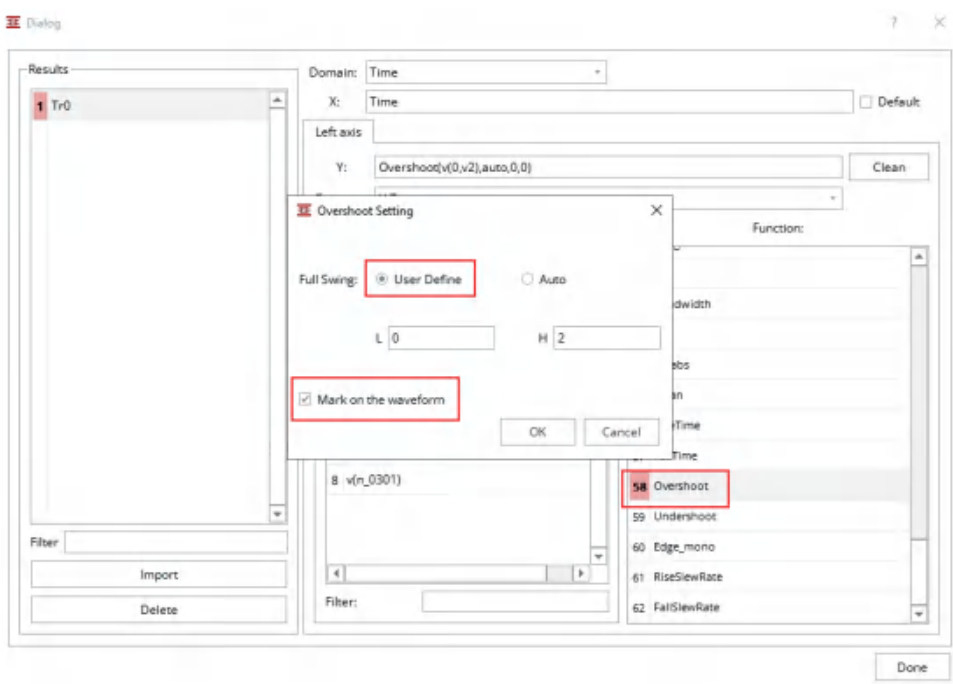
在 opt 目标配置界面点击 Add。



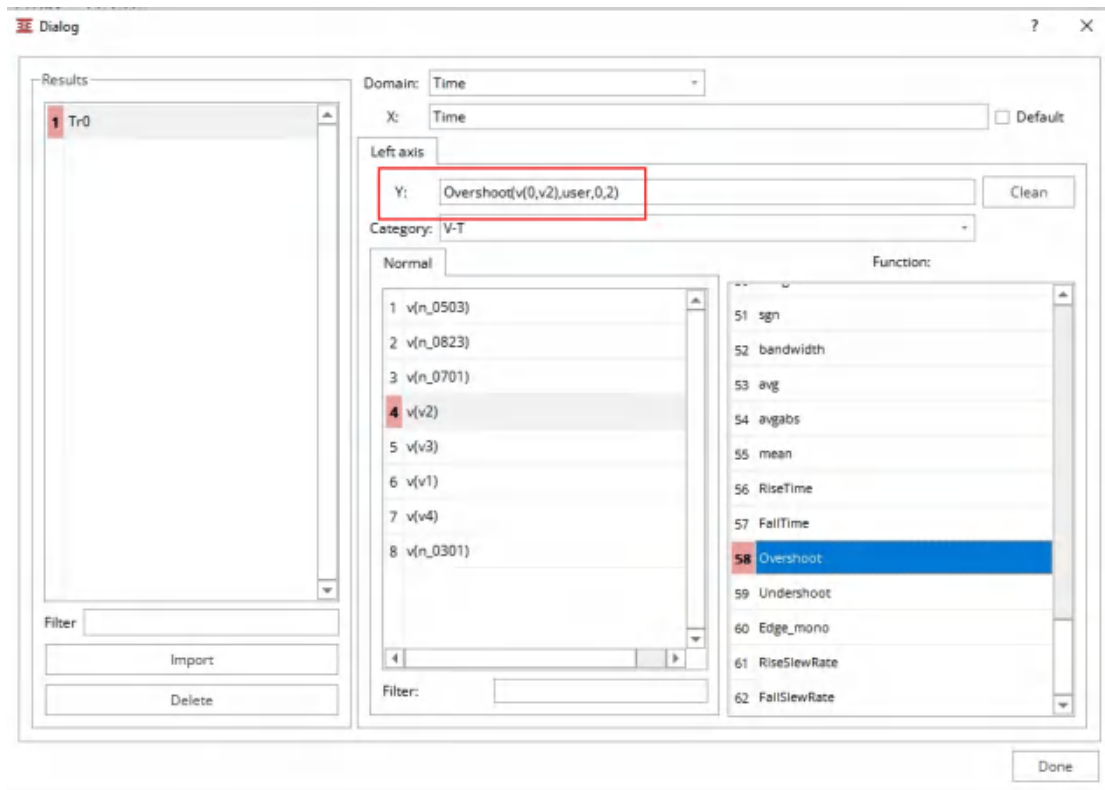
选择需要的目标曲线，比如 v(v2)，在 function 中选择 Overshoot。



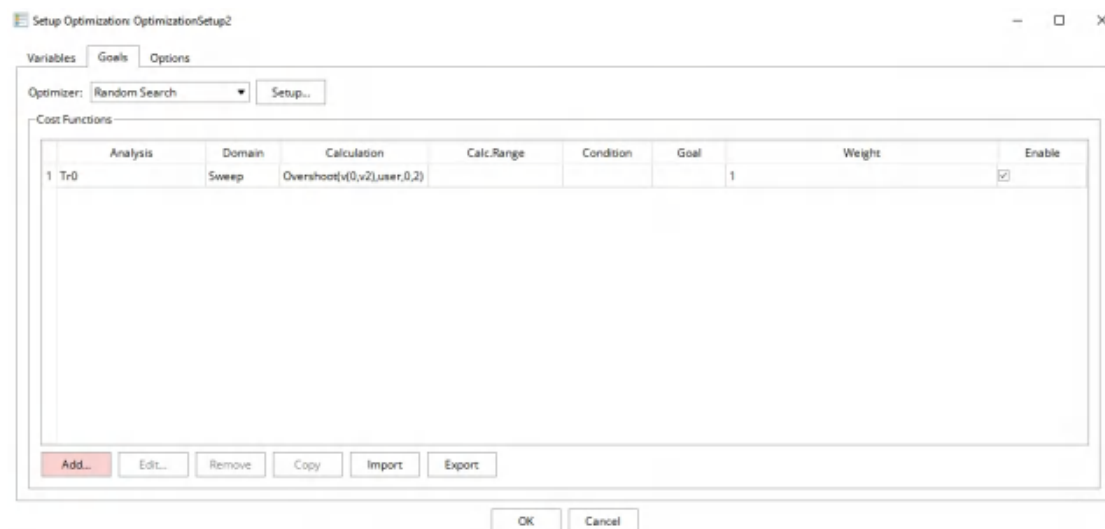
在 function 中点击 Overshoot，在弹出的配置界面模式选择 User Define，L 代表低电平，H 代表高电平，单位为：V，取值需要参考 Normal flow 的结果高低电平，示例中分别取值为 L=0v，H=2v，勾选 Mark on the waveform 后，会将测量的结果标识在结果曲线上，完成后点击 ok。



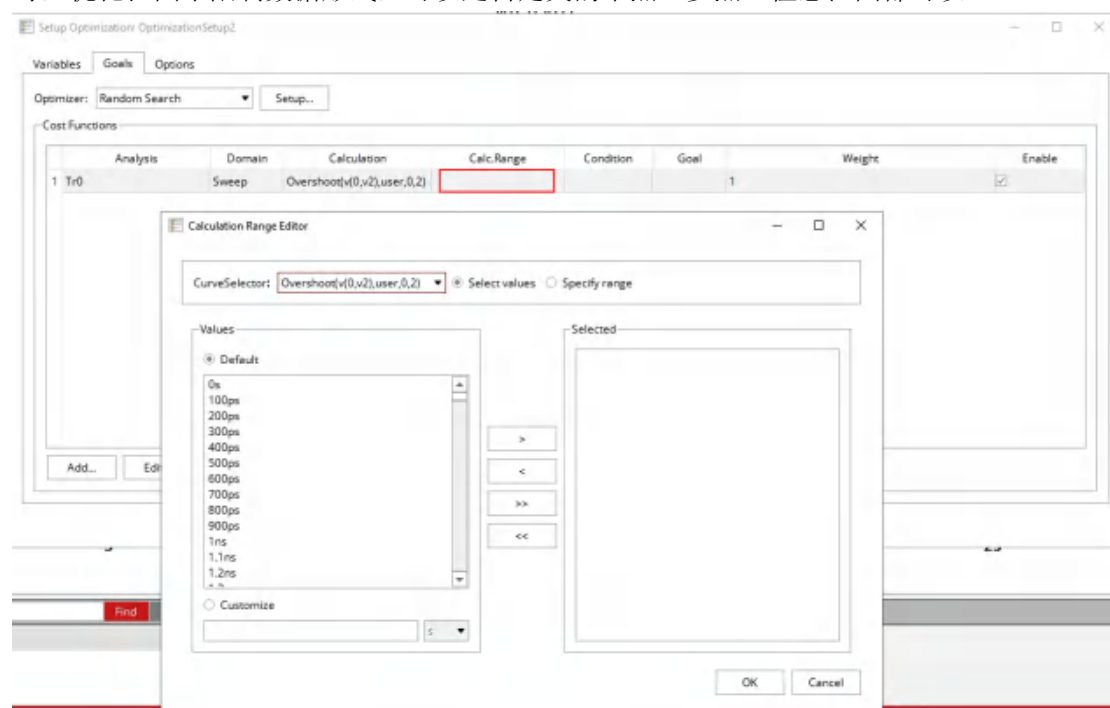
Y 中会显示当前的测量函数表达式。



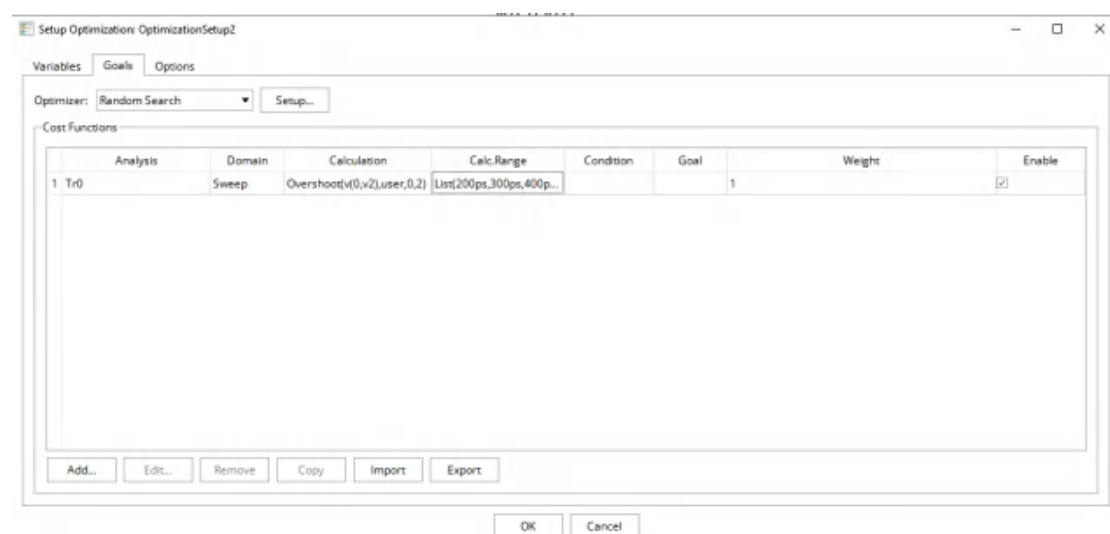
点击 Done，会生成一条目标数据。



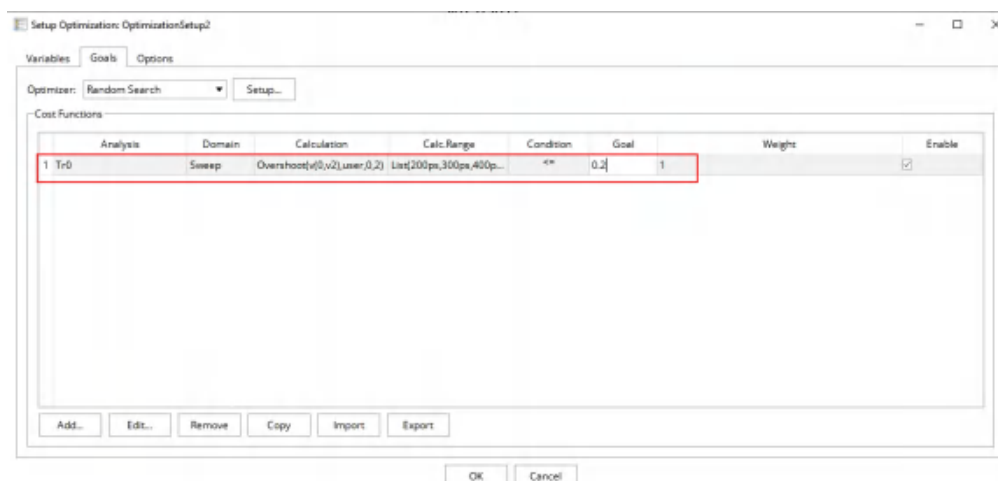
点击 Calc.Range 下的空白处，配置优化范围。需要注意的是，当优化的目标是测量函数时，优化范围不限制数据形式，可以是自定义的单点、多点、任意范围都可以。



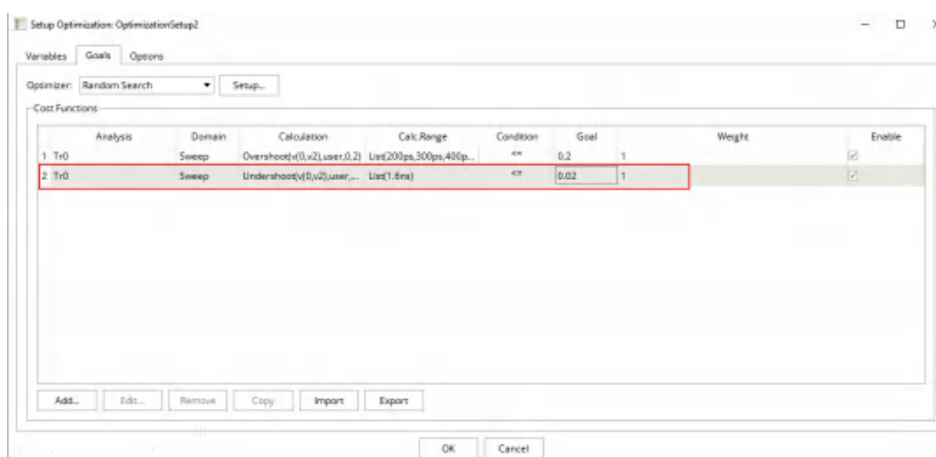
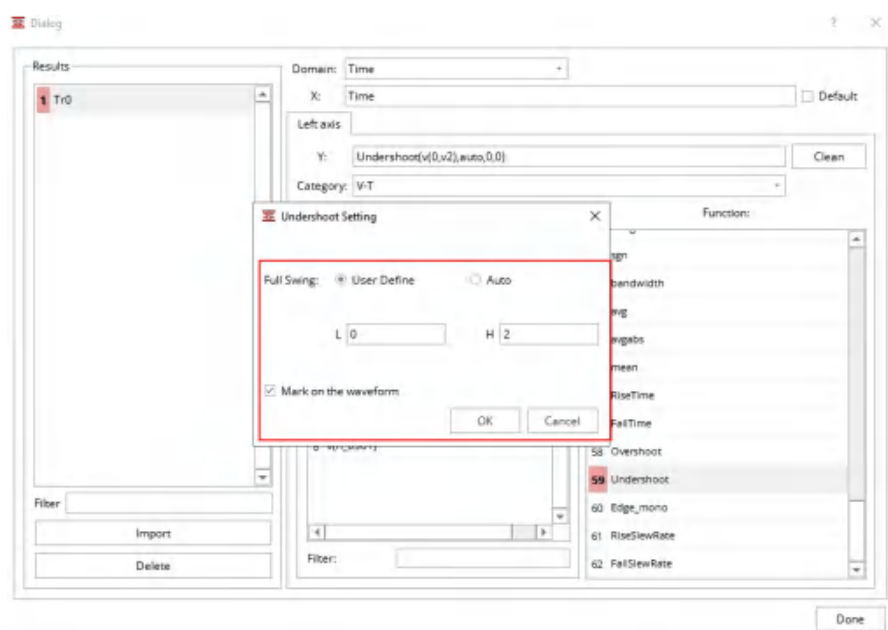
此处选择多个频点为例。选择任意几个频点后点击向右的单箭头选择。完成后点击 Done.



点击 Condition 下的空白处配置条件，点击 goal 下的空白处输入数值。当配置的测量函数时上冲或者下冲时，goal 的默认单位为 V，缺省；当配置的测量函数是下冲或者上冲时，默认单位为 ps，缺省。

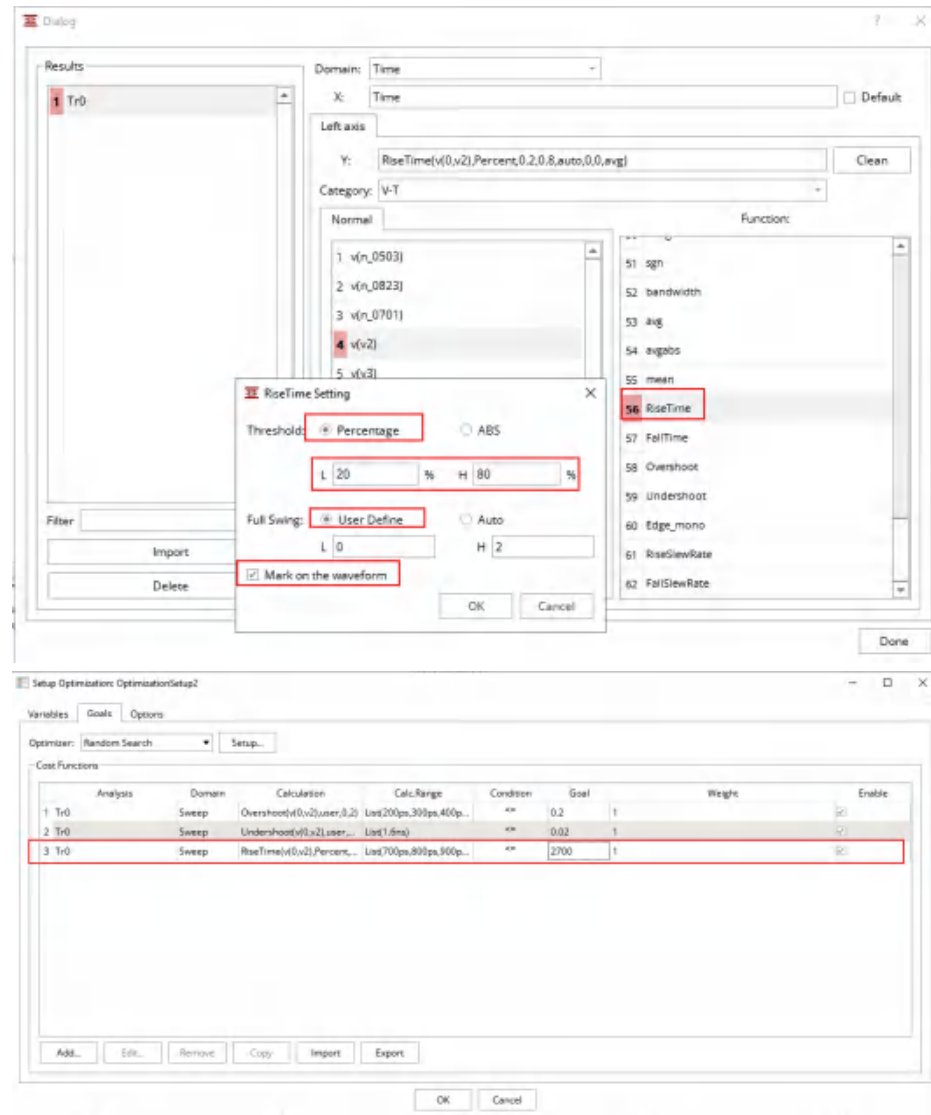


下冲配置和上冲配置类似。

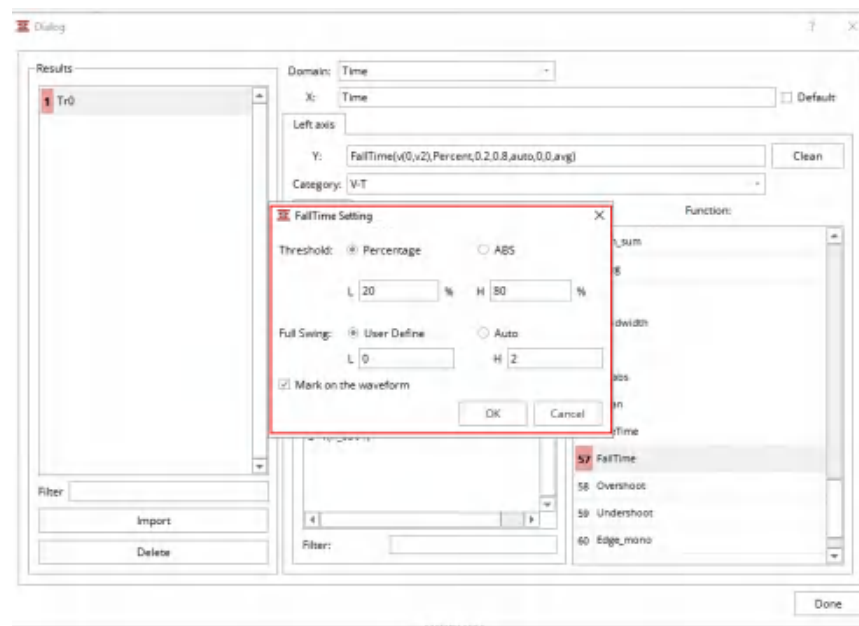


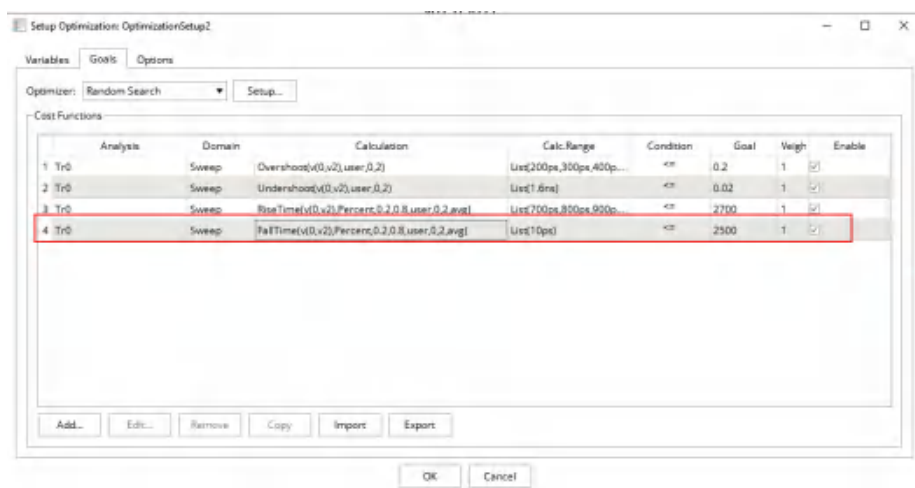
上升时间的 Threshold 选择 Percentage 模式，范围从 20%-80%，Full Swing 使用自定义，低电平 L，单位 V 缺省，高电平 H，单位 V 缺省，设置的数据参考 normal flow 的结果，勾选 Mark on the waveform 表示会将测量的结果展示在波形曲线上。上升下降时间的目标配置中，goal 的默认单位是 ps，缺省。

RiseTime 的测量函数配置如下：



FallTime 的配置如下：





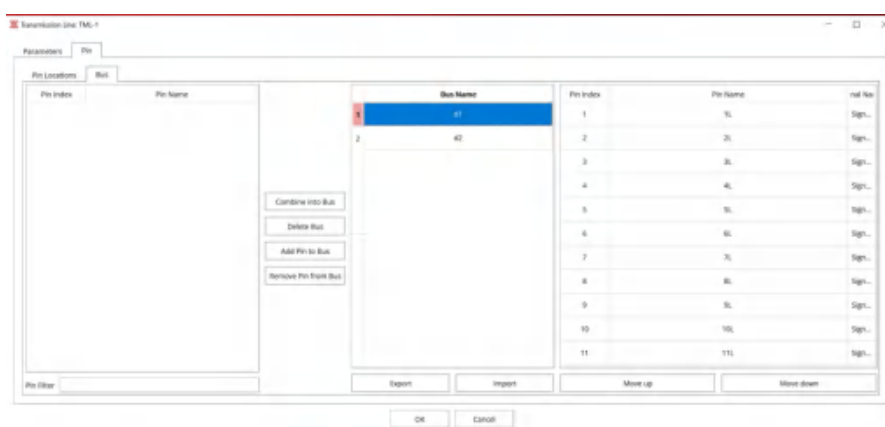
6.9.3 实验设计 DoE

6.9.3.1 搭建 ddr 仿真链路设置 DOE 分析

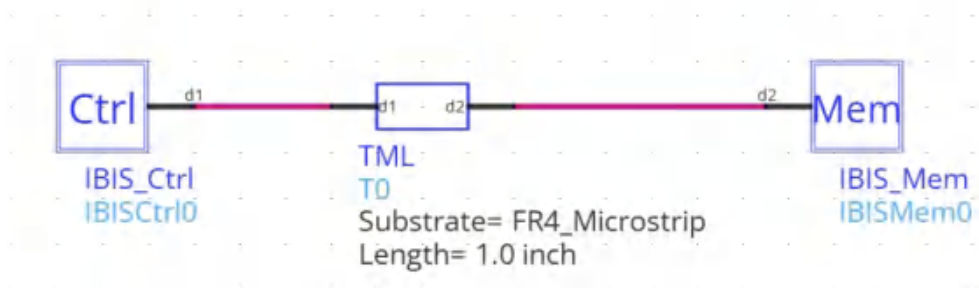
从 TML 下拖拽 ntml 到原理图上，设置 NTML 的值为 11。



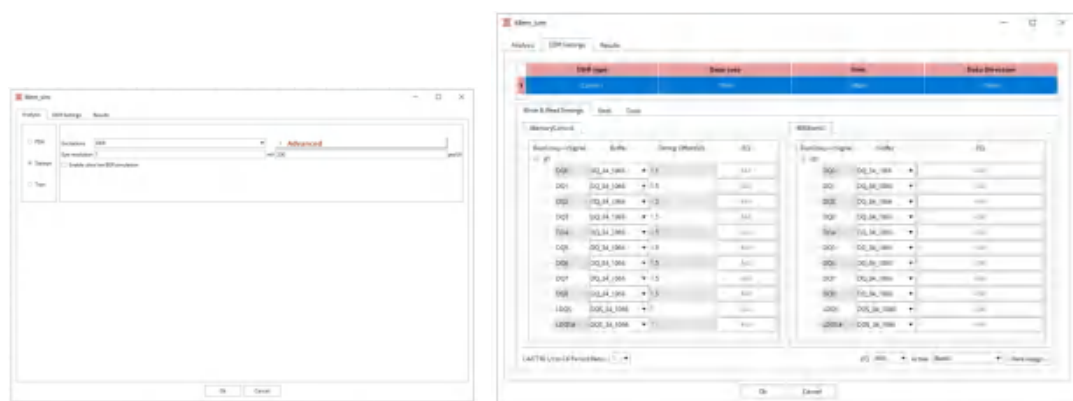
在 BUS pin 中选择一部分 pin，点击 Combine into Bus。使用同样的操作将剩余的 pin 组 bus。



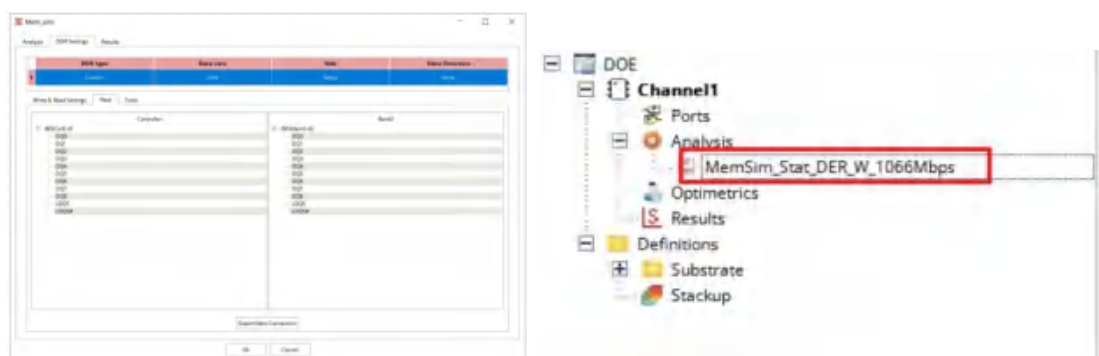
使用第十一章 11.1 节方式搭建 ddr 仿真链路的 ctrl 和 mem。搭建原理图如图所示。



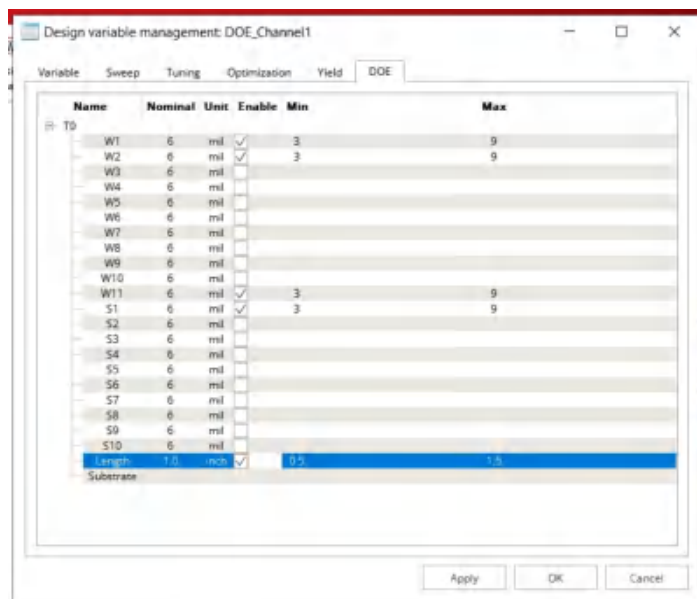
参照 11.2 的步骤添加 ddr stat 仿真。



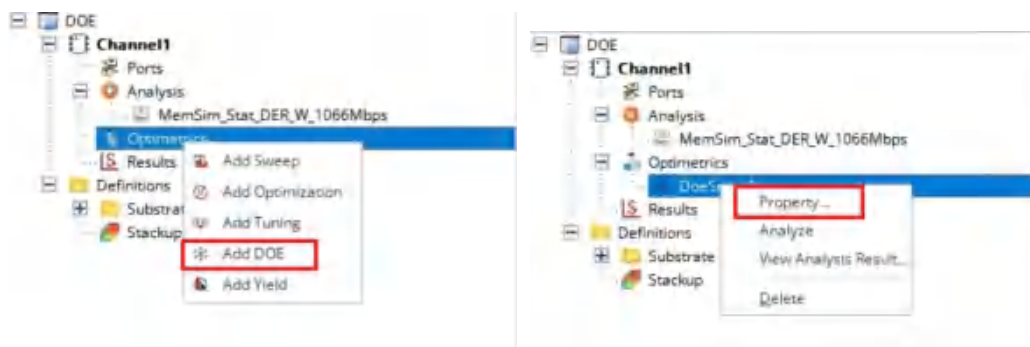
可以在 nets 中查看 pin 连接关系。完成后会生成仿真节点。



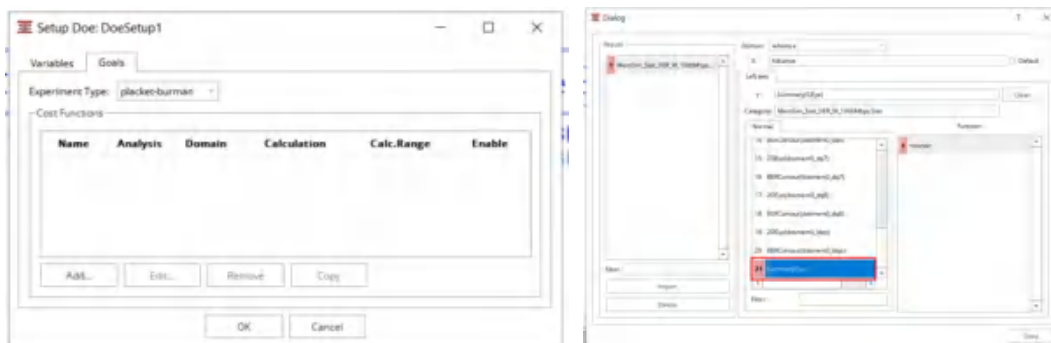
参照 18.1 的步骤，设置 doe 仿真参数。



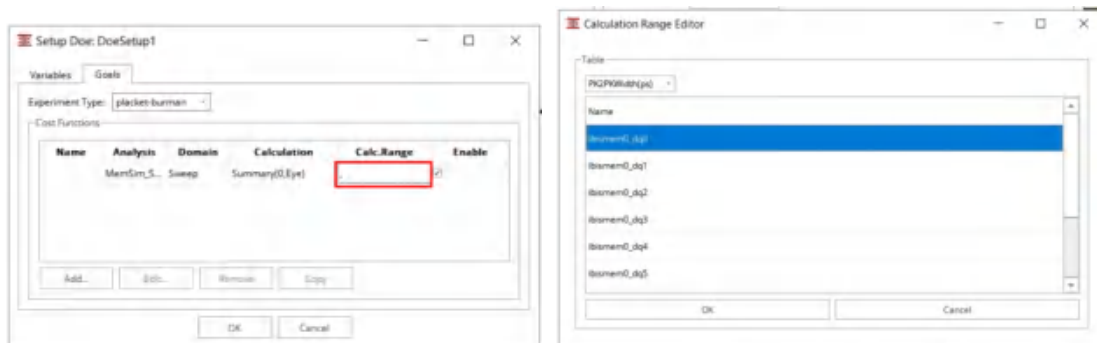
点击工程节点的 Optimetrics 右键选择 Add DOE。点击 DoeSetup1 右键选择 Property。



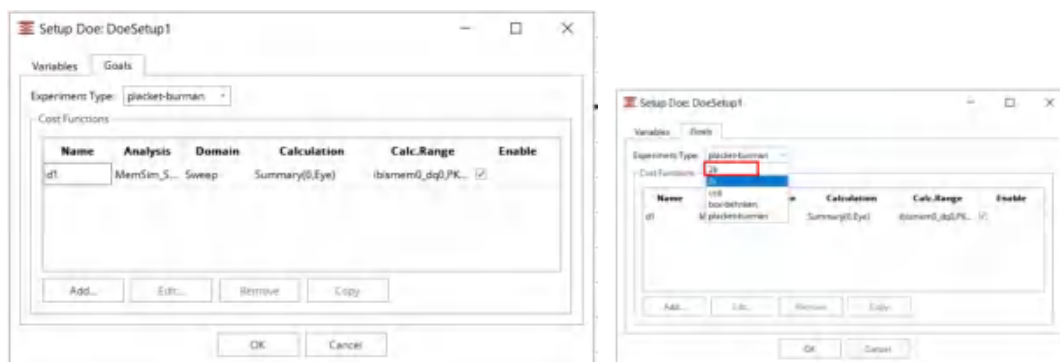
设置目标前需要先进行 Normal flow 仿真。在目标设置界面会显示当前选择的参数。Goals 中使用 18.1 中相同的方法设置目标。点击 Add，选择 Summary 后点击 Done。



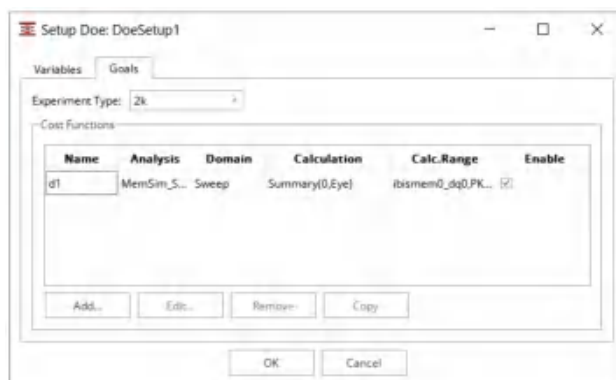
点击 Calc.Range 下的“,”，设置目标信号。



再设置一个目标名字之后，在算法的下拉列表中选择 2k。

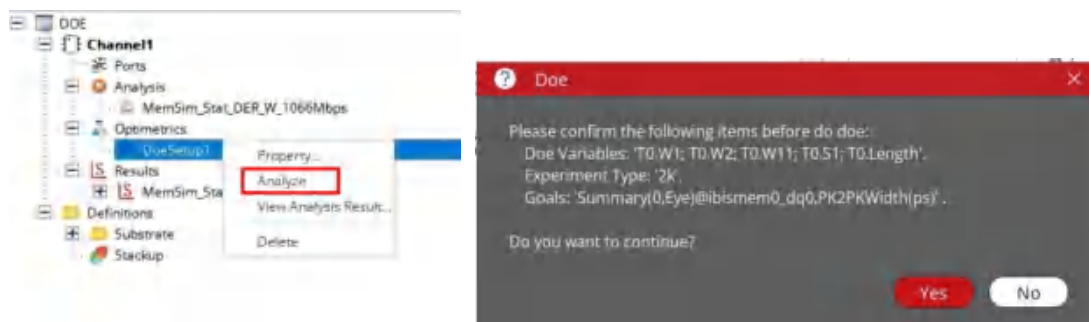


如图完成设置后点击 ok。

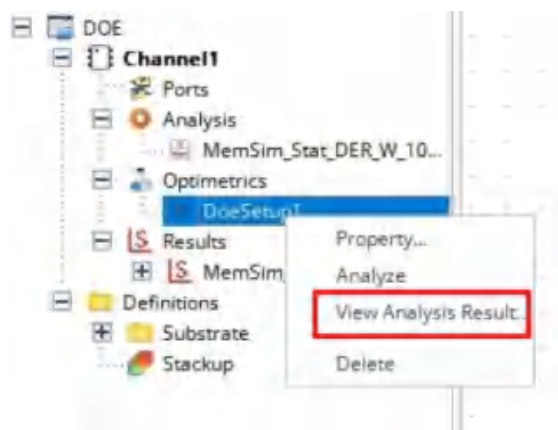


6.9.3.2 DOE 仿真及结果查看

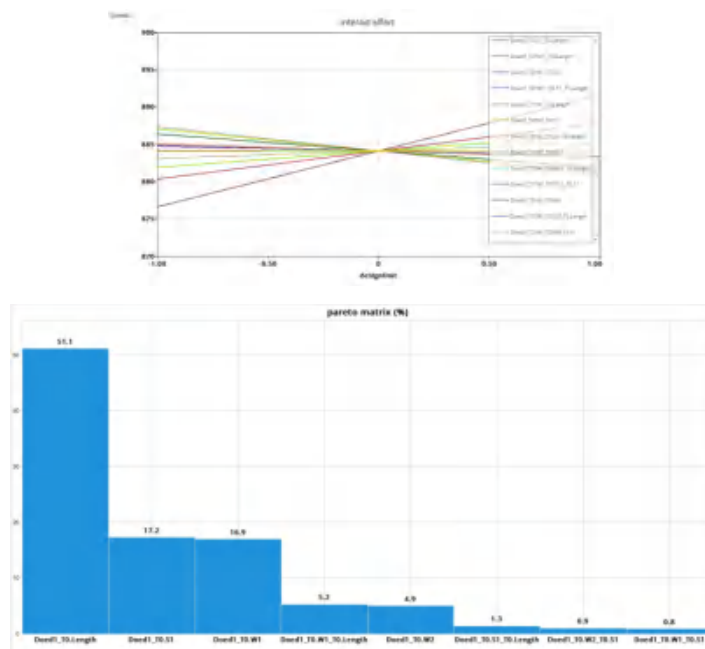
在 DoeSetup1 右键，会弹出提示窗口显示当前设置的目标信息。



点击 Yes 后开始 Doe 分析，等待分析完成后，在 Doe 节点右键选择 View Analysis



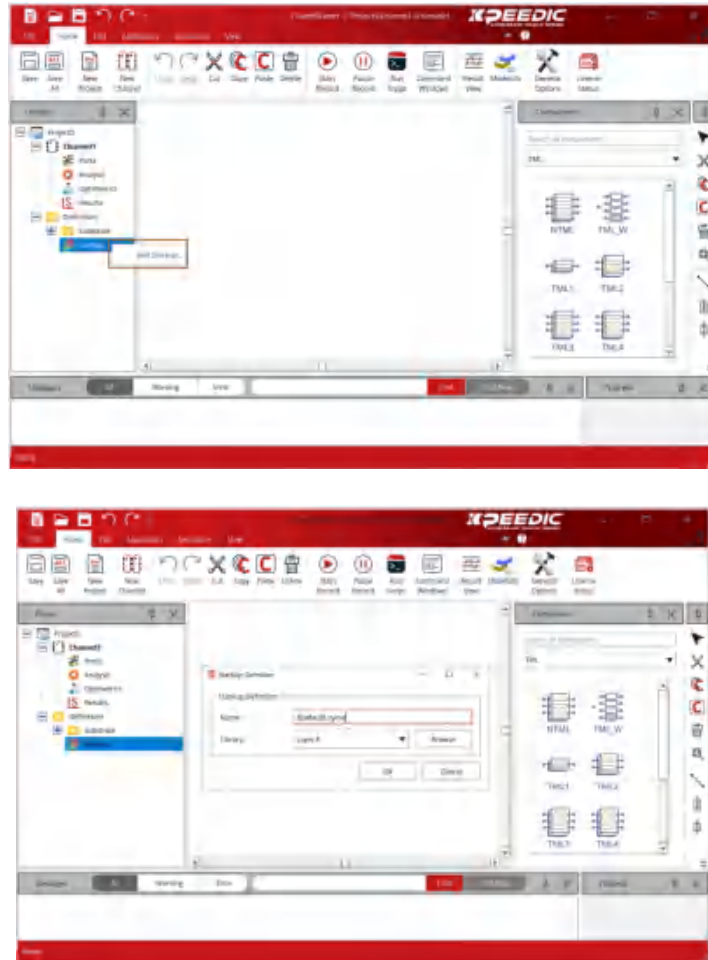
Result:



6.10BROADSIDE 传输线创建与仿真

6.10.1 增加叠层信息

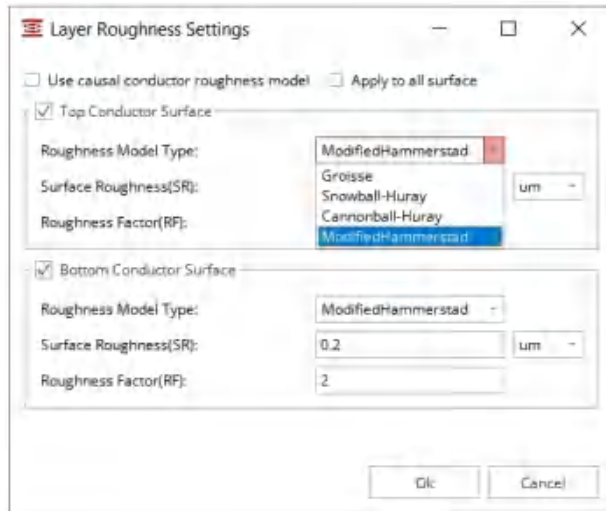
跨层传输线一般需要设置多个叠层，右键工程树节点下的 Stackup，选择 Add Stackup，会弹出叠层的材料信息，进行选择。



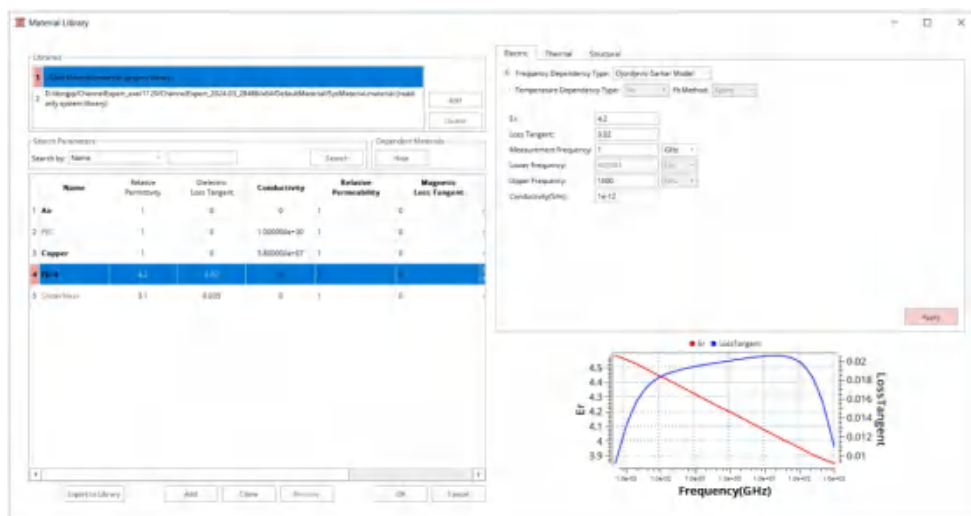
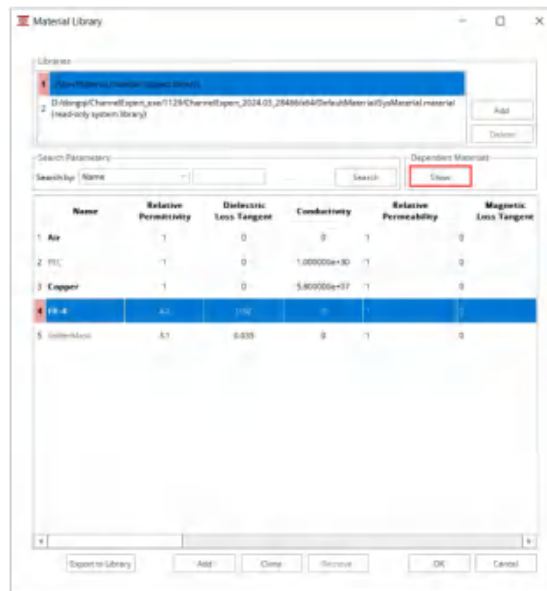
选择好 Layer，点击 OK，点击 Browse，弹出材料框，设置导体、地平面、介质等厚度、粗糙度属性。



设置粗糙度 Roughness，鼠标点进去，可以对每一层进行粗糙度设置。目前支持的粗糙度类型有 Hammerstad、Snowball-Huray、Cannonball-Huray。传输线的上下表面，可以分开设置，也可以一起设置。

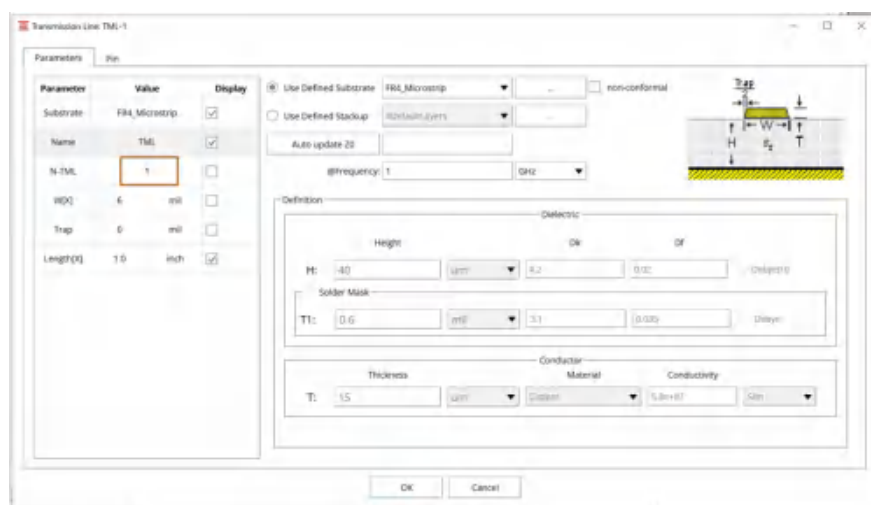


单击上一步的材料框右下角 Material Library，弹出材料 Dk Df、电导率等信息。选中介质材料，单击 Show，设置频变等属性。

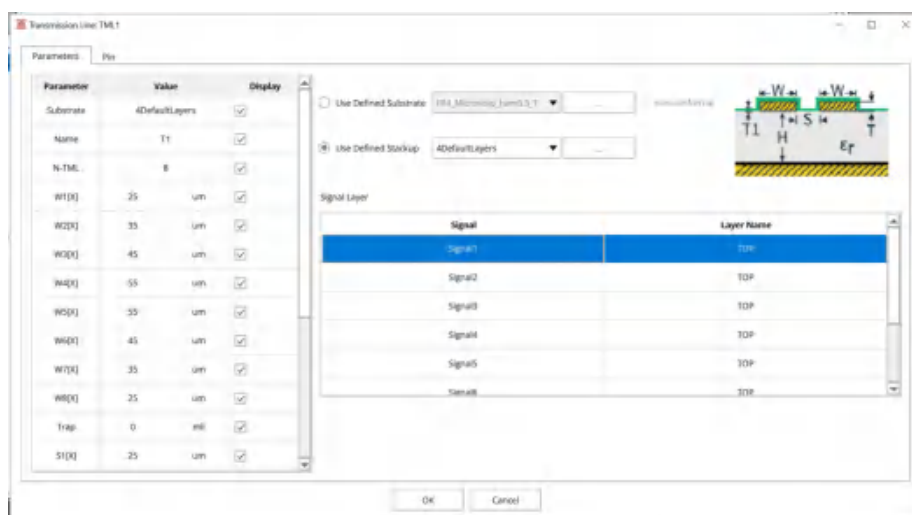


6.10.2 搭建传输线原理图

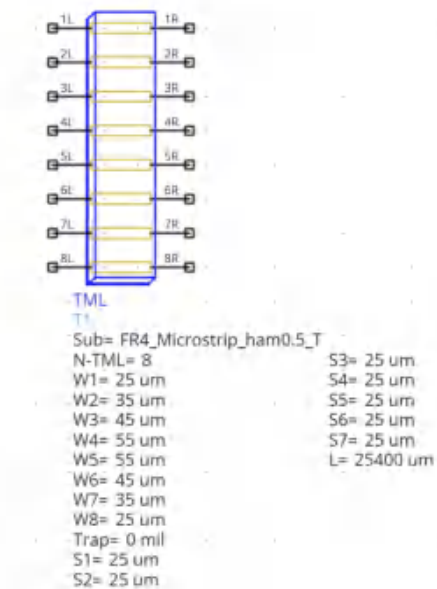
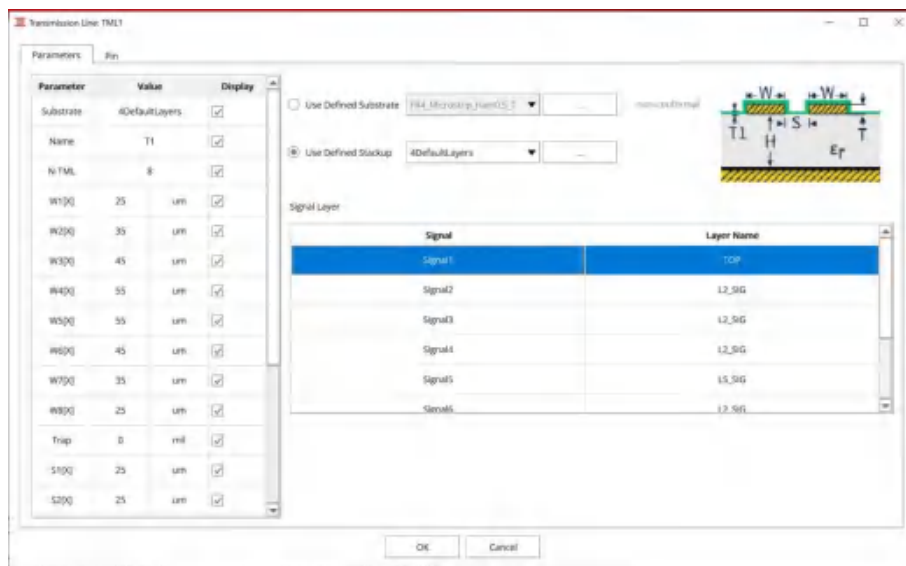
点击 Components TML 器件库里的 NTML，弹出传输线设置界面，这里可以增加任意多根线。在 N-TML 输出需要的传输线个数。对应的 symbol 和属性同时发生改变。



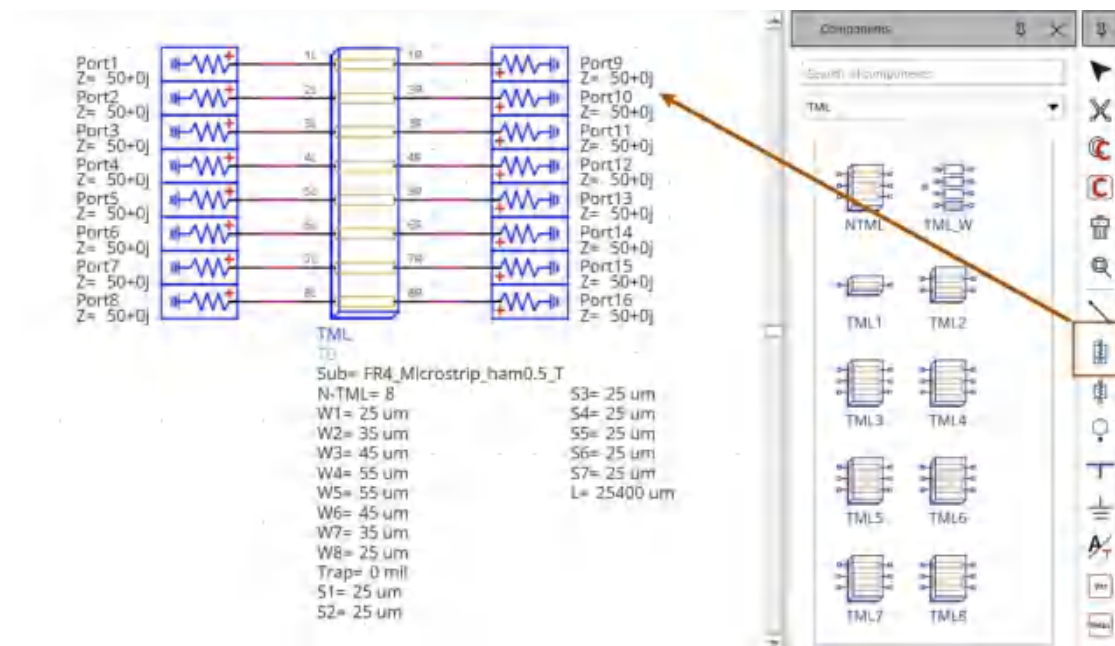
在右侧勾选已经添加好的叠层



分配每根线需要放在的导体层，设置每根线的宽度、线与线之间的间隙、线的总长度。这里线与线之间的间隙规则是上一根线的右侧到下一根线的左侧距离，可以为负值。

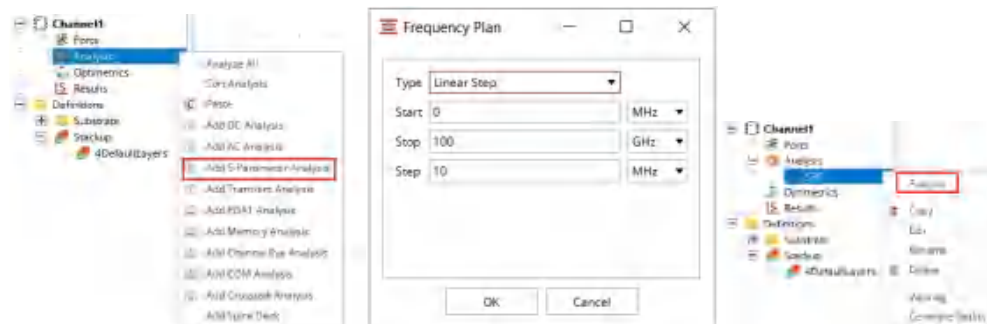


设置好后点击 OK，可以 display 所有信息，传输线设置好后，增加 port 进行频域仿真

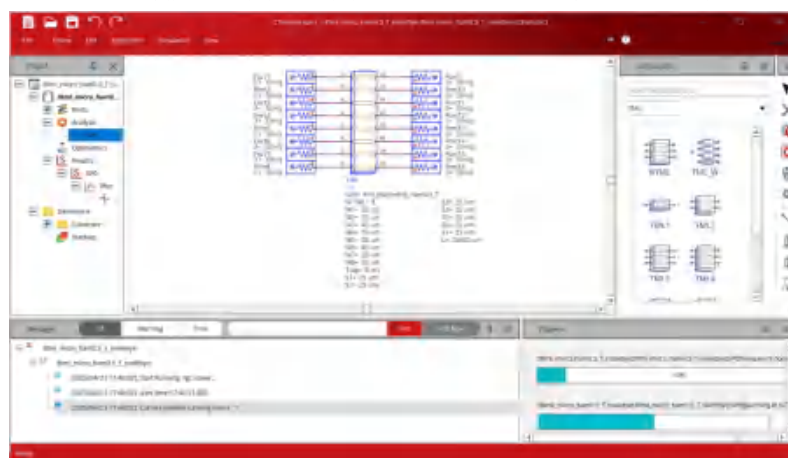


6.10.3 传输线仿真

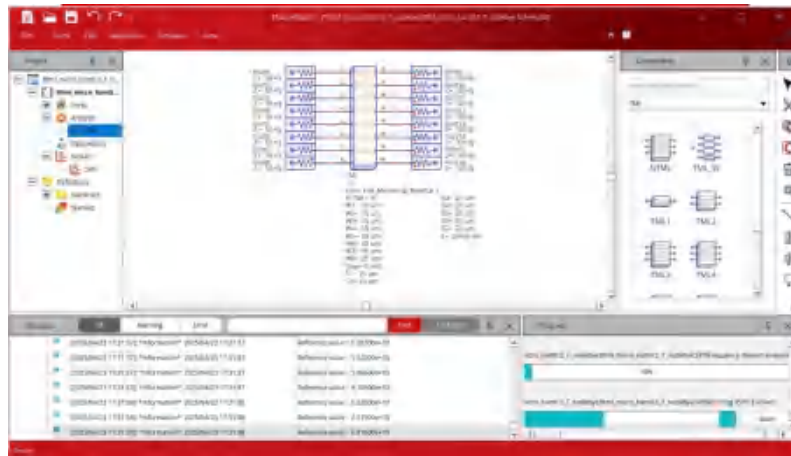
树节点 Add S-Parameter Analysis，弹出仿真设置。



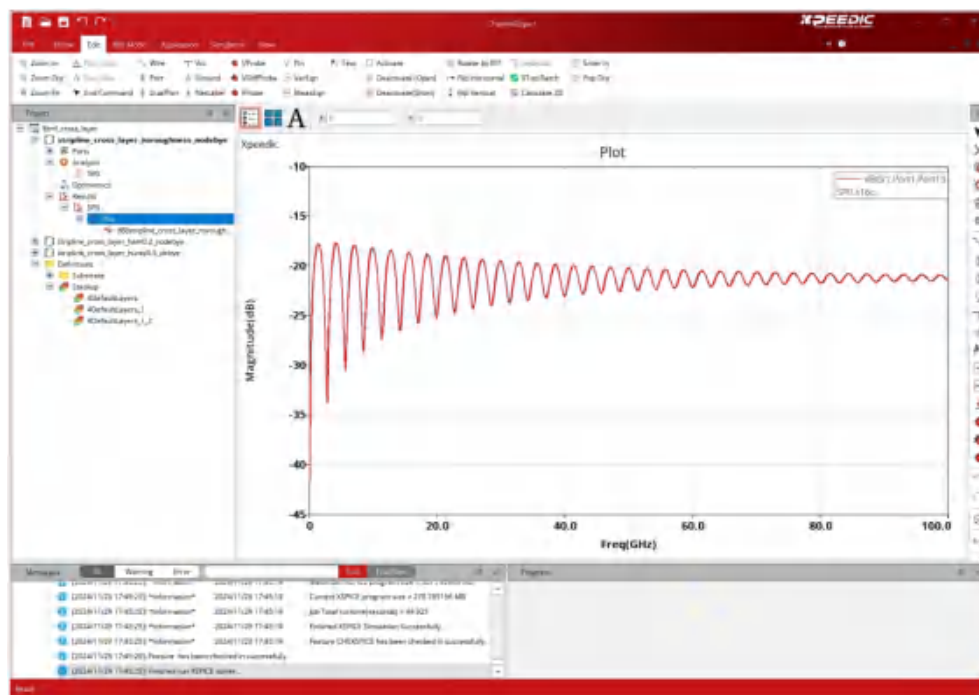
设置好后，点击 Analyze 进入仿真界面，首先生成的是传输线 cfg 材料文件和 mesh，内部调用 X2D solver 进行传输线仿真，仿真成功生成对应的传输线 S 参数。



前端拿到传输线生成文件，会写成网表子电路形式，传递给 xspice solver，做电路仿真。

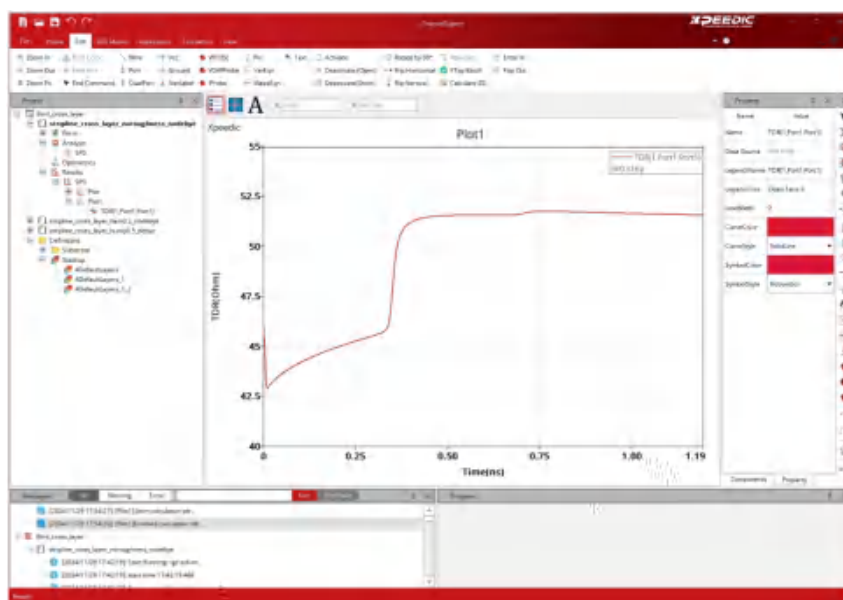
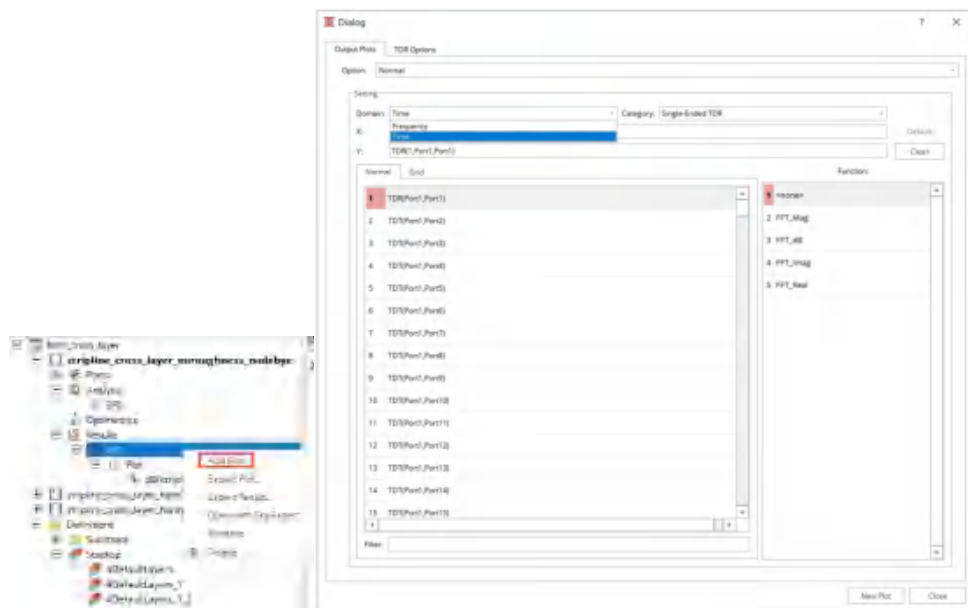


仿真结束，得到电路 result。默认挂在树下的 Plot 是 S11 回损。

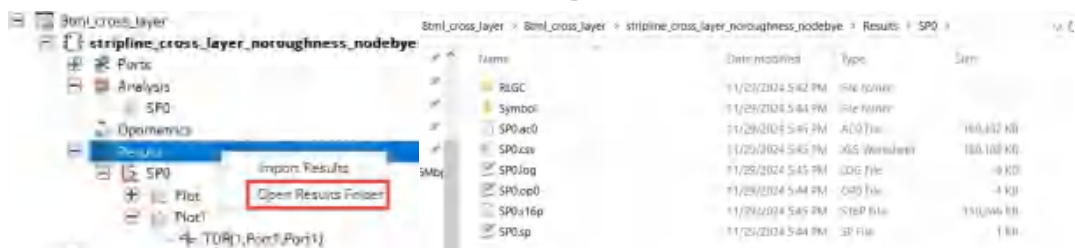


6.10.4 仿真结果查看

右键 result 下的仿真分析结果，Add Plot 弹出绘图界面，可以查看 TDR、插损等 Plot。



需要拿出生成的结果，右键 results，选择 Open Results Folder。

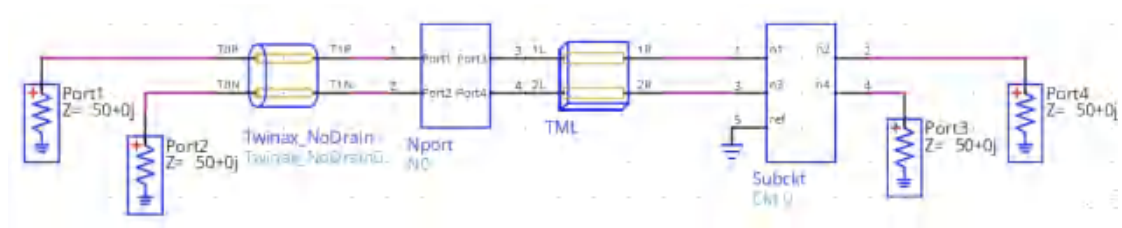


6.11 CABLE 创建与仿真

6.11.1 频域仿真

6.11.1.1 搭建 CABLE 的仿真链路

从元件库中分别拖入 Cable, Nport, tml, subckt 及 port 依次放置到画布中, 依次连接起来, 如图所示。

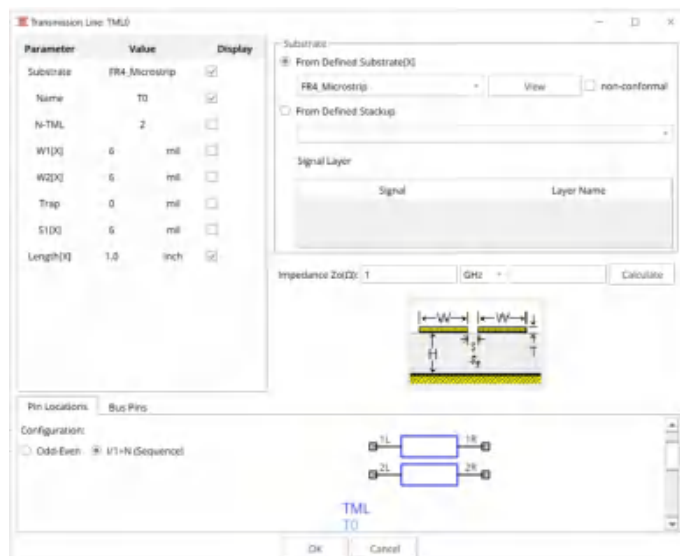


6.11.1.2 配置 CABLE 参数

双击 Cable 的 NoDrain 元件, 可在编辑界面设置长度, 材料, Sig, 厚度, sig 的偏移量等相关参数:

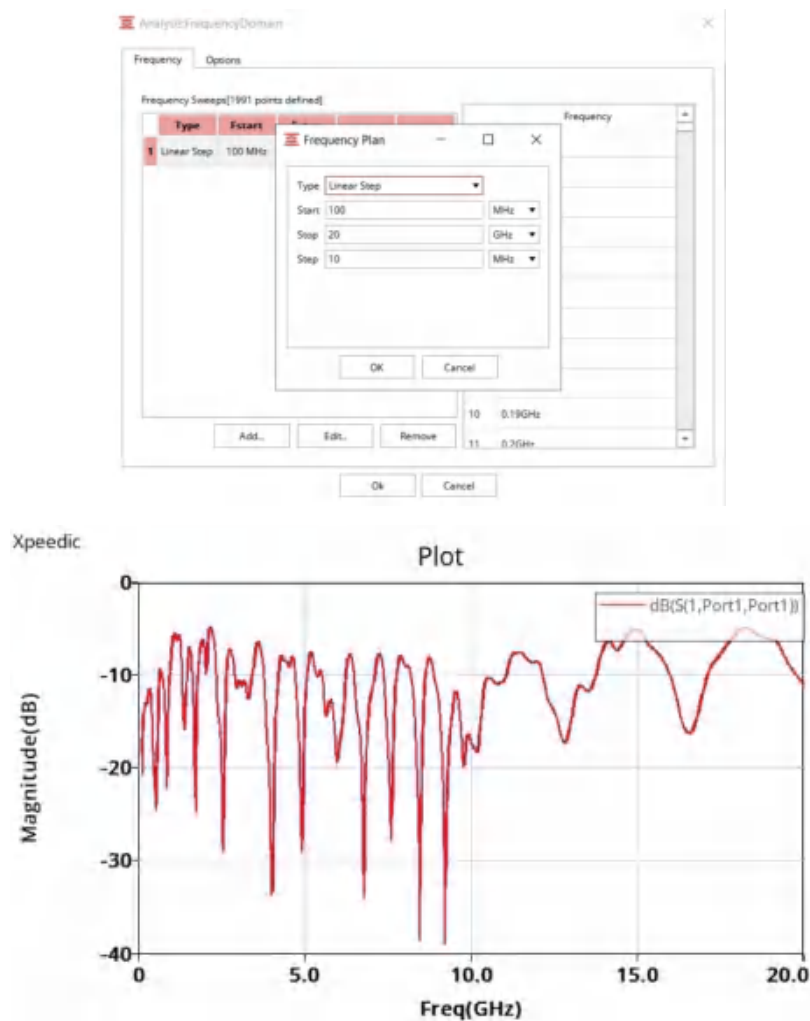


双击 Nport, 点击 Browse 加载需要的 S 参数。双击 TML, 在编辑界面可以配置 W、length, 叠层等参数。Subckt 元件同样需要加载子电路文件。



6.11.1.3 添加仿真，启动仿真后查看结果

在仿真配置中，可以设置 type 为 Linear step、Linear Count 等四种模式，可分别设置起始、终止、步长频率。



仿真完成后在工程路径下会生成一个 cable 的项目工程文件，在 RLGC 文件夹中会有 cfg、mesh、rlgc 及 s*p 文件生成。

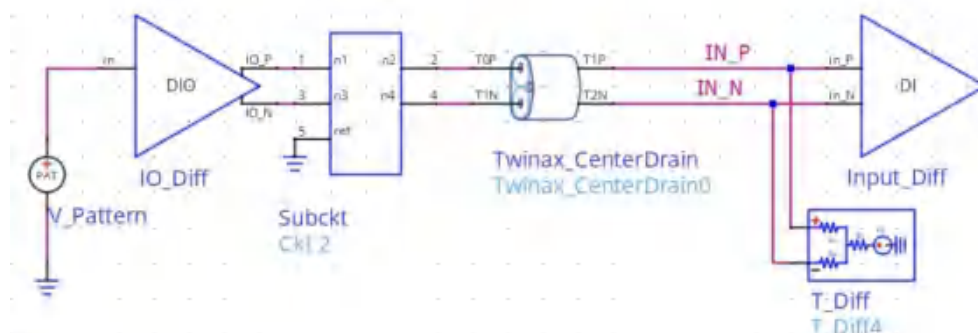
Name	Date modified	Type	Size
Cable_1120	11/21/2024 7:32 PM	File folder	
Cable_1120.cable	11/21/2024 7:37 PM	XPEEDIC CableExp...	1 KB

> ChannelExpert > Daily > 2024 > 11 > 1120 > CableModel > Cable_1120 > Channel5_Twinax_NoDrain0			
Name	Date modified	Type	Size
Channel5_Twinax_NoDrain0.cbt	11/21/2024 7:37 PM	CBT File	8 KB
Channel5_Twinax_NoDrain0.cfg	11/21/2024 7:33 PM	CFG File	1 KB
Channel5_Twinax_NoDrain0.log	11/21/2024 7:33 PM	LOG File	5 KB
Channel5_Twinax_NoDrain0.mesh	11/21/2024 7:33 PM	MESH File	.615 KB
Channel5_Twinax_NoDrain0.rlgc	11/21/2024 7:33 PM	RLGC File	561 KB
Channel5_Twinax_NoDrain0.s4p	11/21/2024 7:33 PM	S4P File	1,576 KB

6.11.2 时域仿真

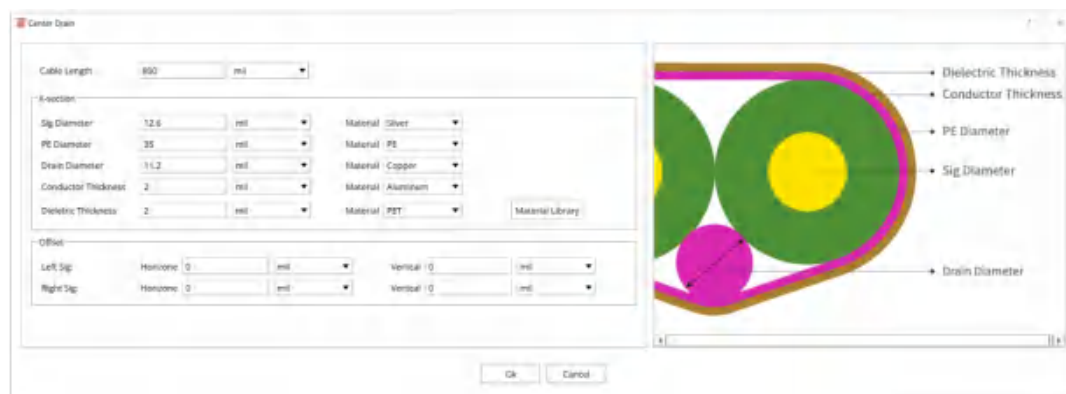
6.11.2.1 搭建带 cable 器件的时域仿真电路

从软件元件库 cable 下选择 CenterDRain，DataSource 下选择 subckt，从 IBIS 下选择 IO_Diff 与 Input_Diff，从 Independent Sources 下选择 Pattern 源，分别拖入到画布中，如图：

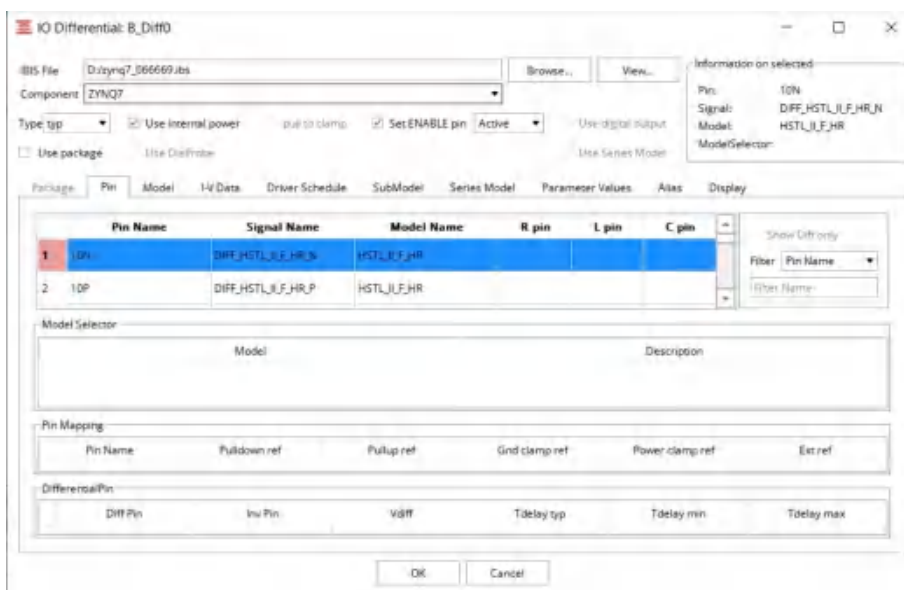


6.11.2.2 配置时域仿真电路

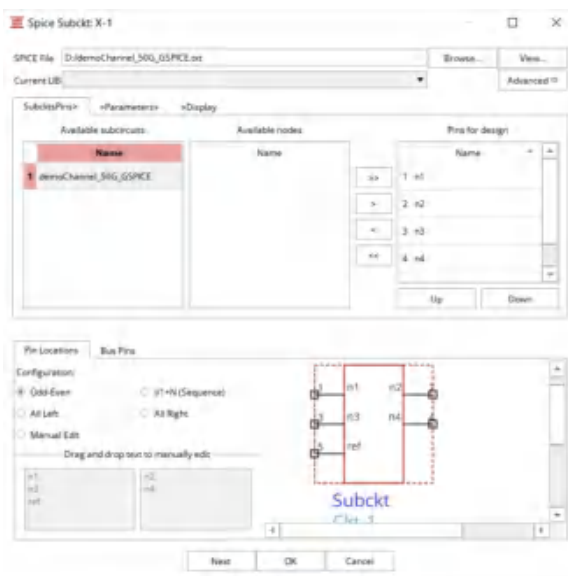
1、设置 Cable 在的 Length, Sig, Thickness, offset 等参数。



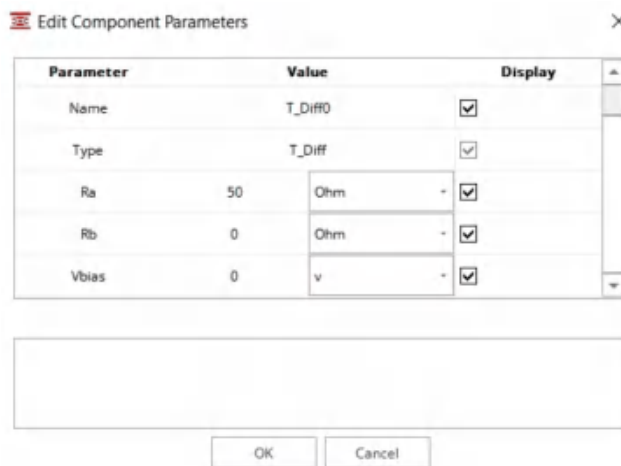
2、IBIS 器件需要分别加载合适的 IBIS 文件，并选择合适的 pin, Type 类型，是否使用内部电源，是否使用封装，是否输出 DieProbe 都可以自定义设置。需要展示的参数可以在 Display 中设置。



3、Subckt 需要加载子电路文件。

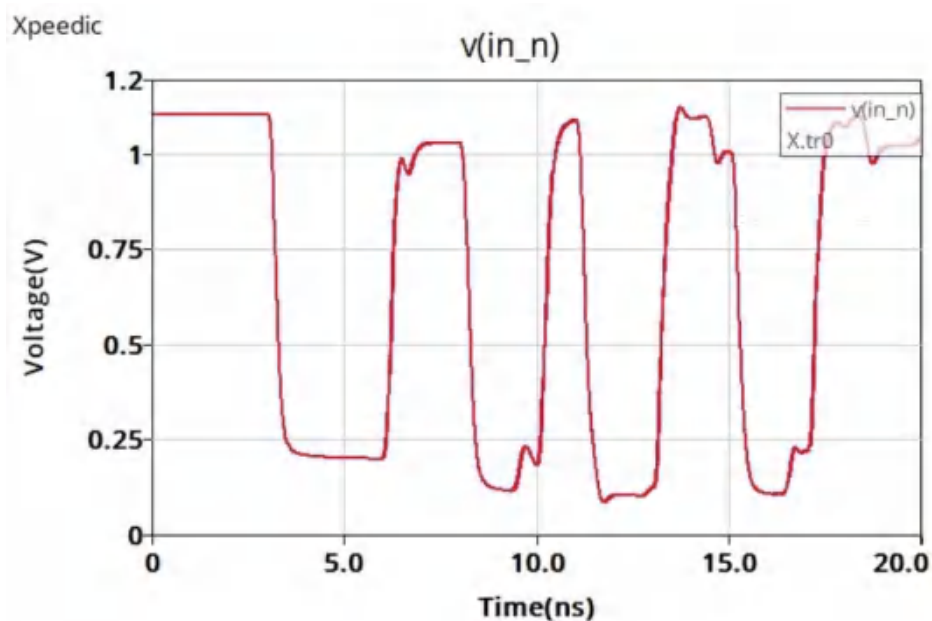


4、T_Diff 可以设置 Ra, Rb, 电压以及展示参数信息。



6.11.2.3 添加时域仿真及仿真

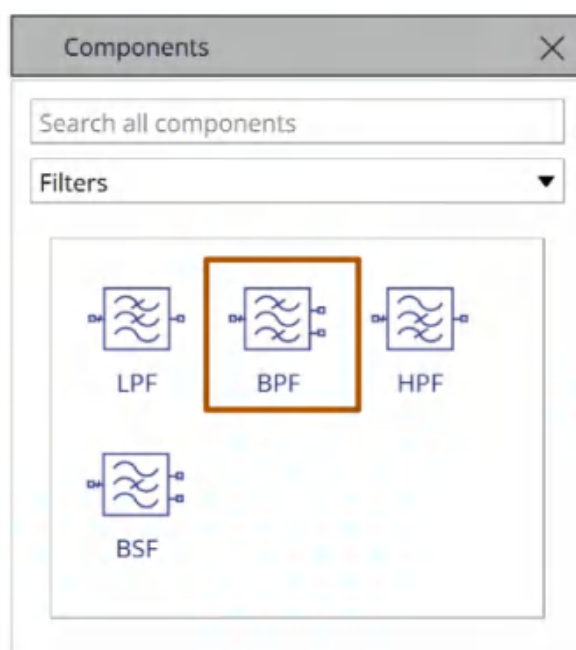
在工程树仿真节点添加时域仿真，仿真可以 probe 节点查看相应观测点的数据。



6.12 行为级滤波器 FILTER 仿真

6.12.1 搭建 BPF 频域仿真电路

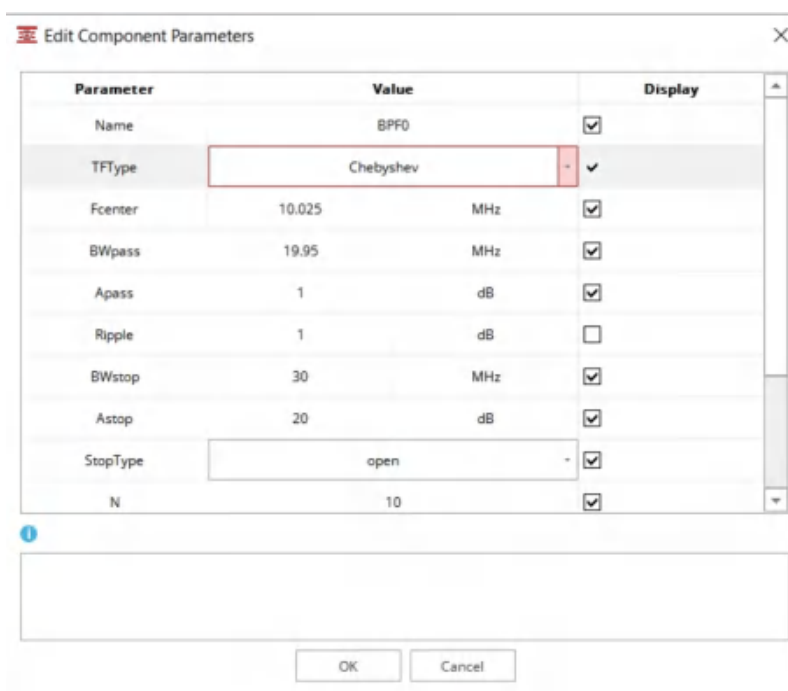
从元件窗口 Filter 中拖 BPF 到原理图，再从右侧工具栏中加入 Port，并连接如图所示。





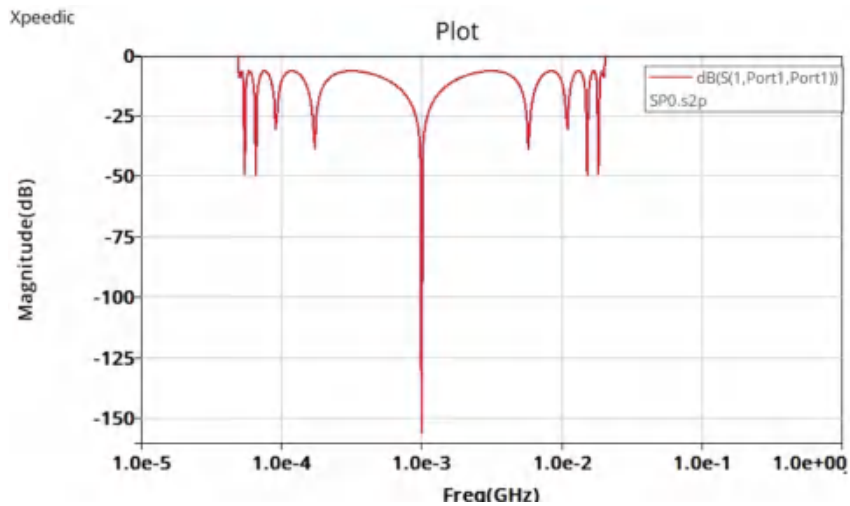
6.12.2 配置滤波器器件属性

双击原理图 BPF，在弹出的属性编辑窗口中对属性进行编辑。在 Tftype 中可分别选择 Bessel/ButterWorth/Chebyshev/Gaussian/Elliptic，不同 Tftype 下可分别设置对应的 Fcenter 等参数。



6.12.3 BPF 元件的 s_paramater 仿真

在工程节点 Analysis 下添加 S_paramater 仿真，并查看仿真结果。



6.13 SERDES PRELYOUT 仿真

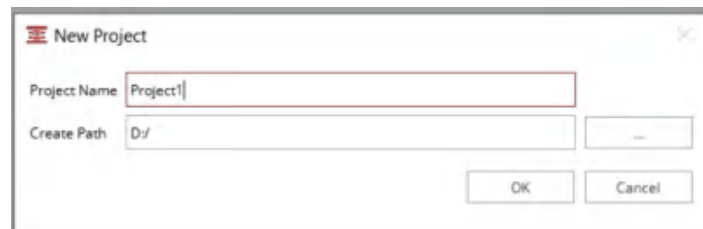
6.13.1 非 TML 模板类 Pre 仿真

6.13.1.1 搭建普通 pre 仿真链路

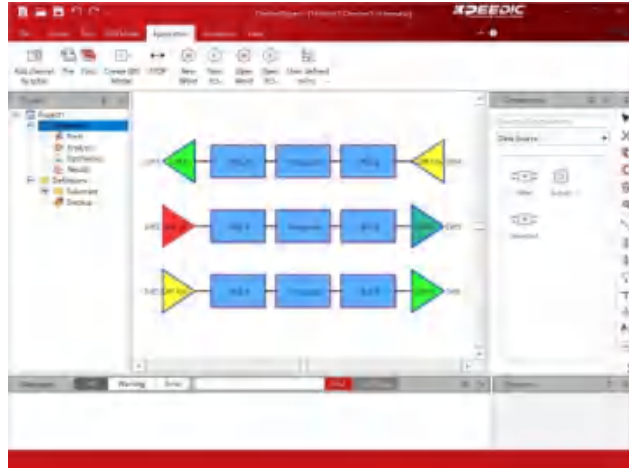
点击菜单栏窗口的 Application，选择 pre 选项进入到 pre 仿真通道设置界面，可以分别设置协议、拓扑类型、NEXT/FEXT 串扰通道个数。



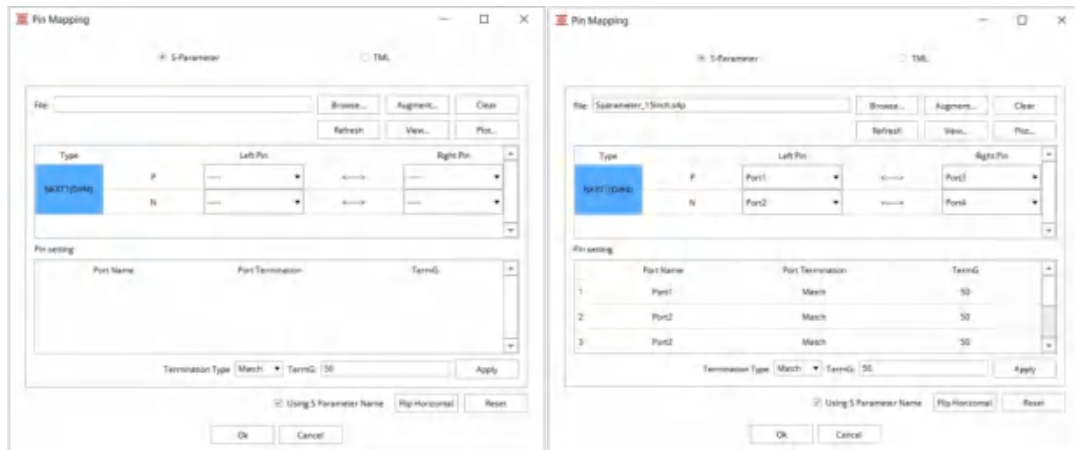
设置完成后点击 ok，进入到工程创建设置界面，可以设置项目的名称和项目保存路径



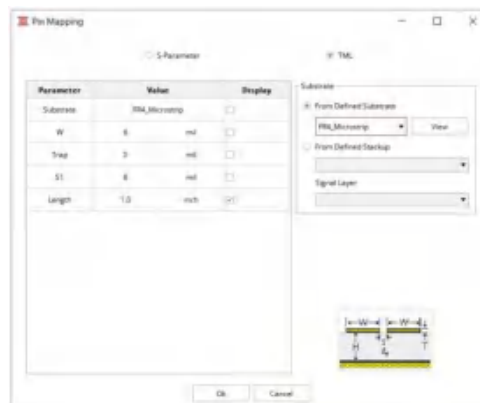
点击 ok 后会在展示出 pre 拓扑连接关系的原理图。中间红色 DIFF_TX 所在路径是主通道，主通道上方是近端串扰（NEXT），主通道下方是远端串扰（FEXT），蓝色矩形是通道的 element。



双击蓝色 element，可以对 element 进行模型设置。在 pin mapping 界面可以设置 element 为 S 参数类型或者 tml 类型。当需要设置成 S 参数类型时，点选 S parameter 前面的选择按钮，点击 Browse 加载需要的 S 参数文件。

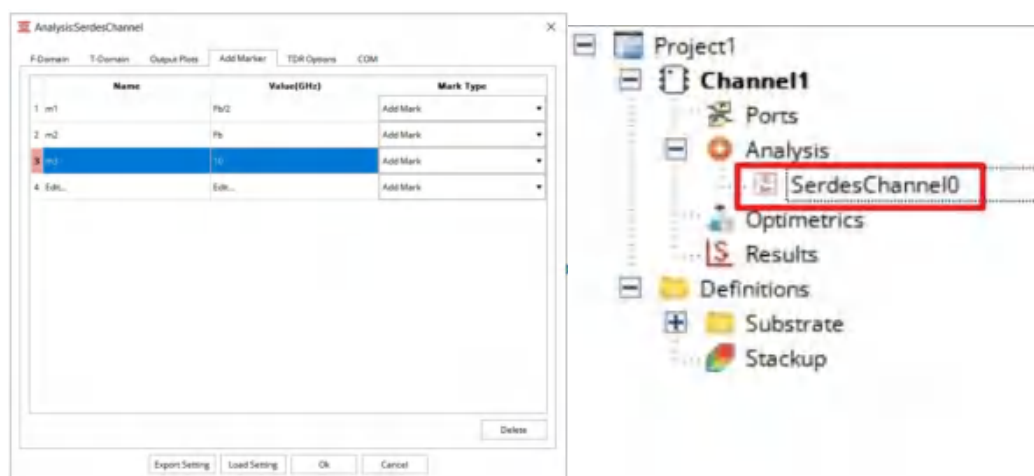


当选择的类型为 TML 时，可以在该界面设置 tml 的 w、trap、length 的属性参数。



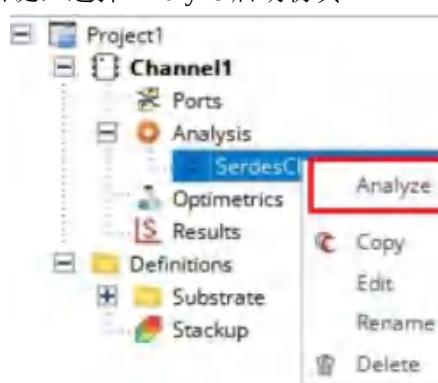
S 参数或者 TML 设置完成后，会刷新展示状态。S 参数会显示 S 参数名称，并变成蓝色外框，实例化后的 tml 也会变成蓝色外框。如果需要对 element 进行组合操作，可以分别选中两个通道各一个 element，右键选择 Combine Element，会在组合一起的两个 element 上用虚线连接起来。

当需要添加 mark 点时，可以在 Add Marker 界面添加。双击 Name 下的 Edit 添加名字，双击 Value 下面的 Edit 可以添加频率值。配置完成后点击 ok，生成仿真节点。

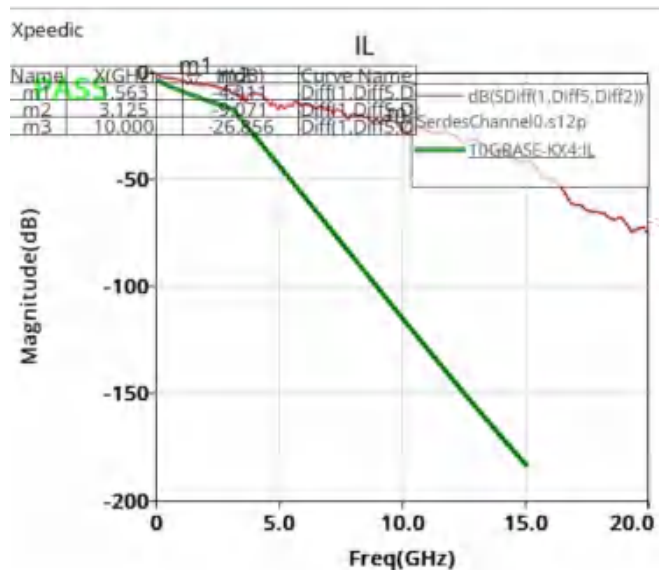


6.13.1.3 查看仿真结果

在添加的仿真节点上右键，选择 Analyze 启动仿真。



仿真完成后，会在 result 节点挂载仿真结果。

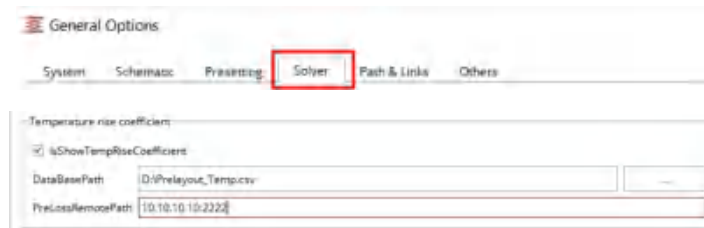


6.13.2 TML 模板类 Pre 仿真

6.13.2.1 损耗收集的仿真链路搭建

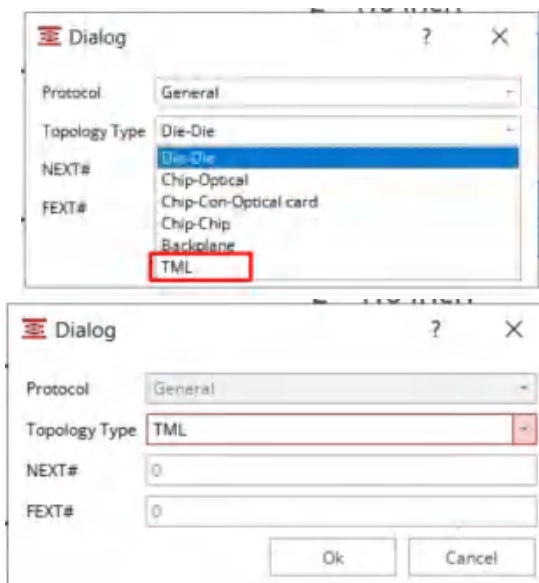
损耗收集需要先配置损耗收集参数的文件和 libmanager 的地址。

选择菜单栏 Home-General Options ，选择 Solver，勾选展示温升影响的选项，在 DataBasePath 中添加损耗收集参数的文件，在 PreLossRemotePath 中添加 libmanager 的 ip 和端口，设置完成后点击 ok 保存配置。

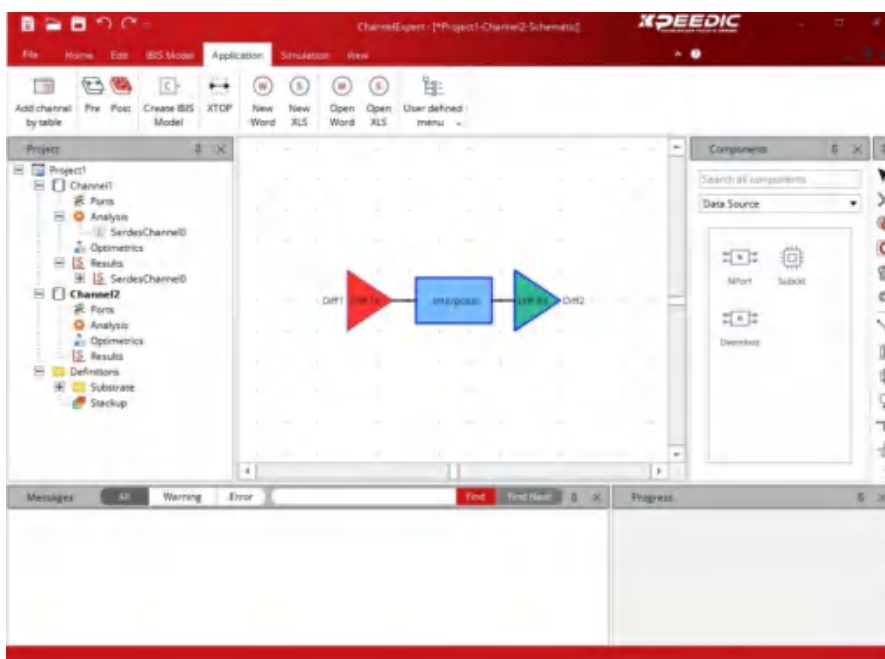


损耗收集需要关联一个普通的 serdes 节点，创建普通的 serdes 仿真节点同 17.1.1。

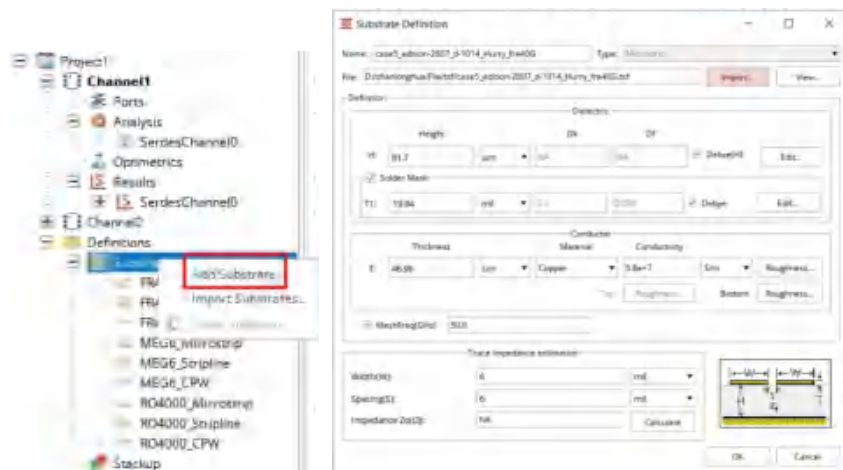
点击菜单栏窗口的 Application，选择 pre 选项进入到 pre 仿真通道设置界面。点击 Topology Type，选择 TML，创建 TML 模板类的仿真工程。TML 模板类的仿真工程不需要设置串扰通道个数。



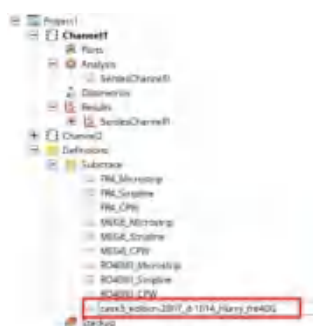
点击 ok 后会生成一个新的 channel 节点并创建拓扑结构。



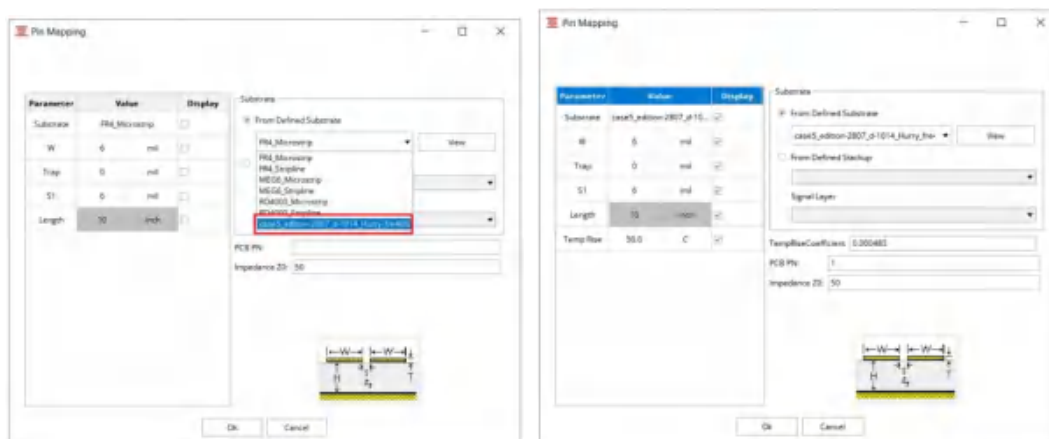
点击工程节点下的 Substrate，右键选择 Add Substrate。在 substrate 界面，点击 Import 导入叠层配置文件。



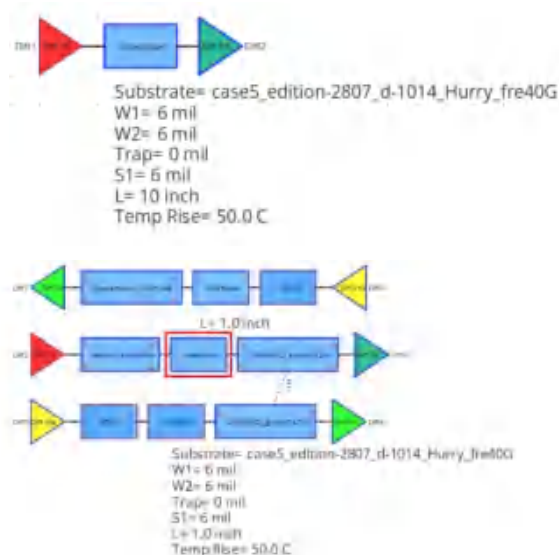
导入完成后会在工程节点 Substrate 下面新增一条叠层记录。



双击 tml 模板类节点的 element，在编辑界面 Substrate 下拉列表中选择带温升的 Substrate，tml 属性列中会显示出温升，右侧也会展示温升系数。设置一个 PCB PN 值后点击 ok 完成 tml 设置。



Tml 类设置完成后，同样的方法需要设置非 tml 类的 channel 节点，选择一个 element 设置相同的 substrate 和 tml 参数。

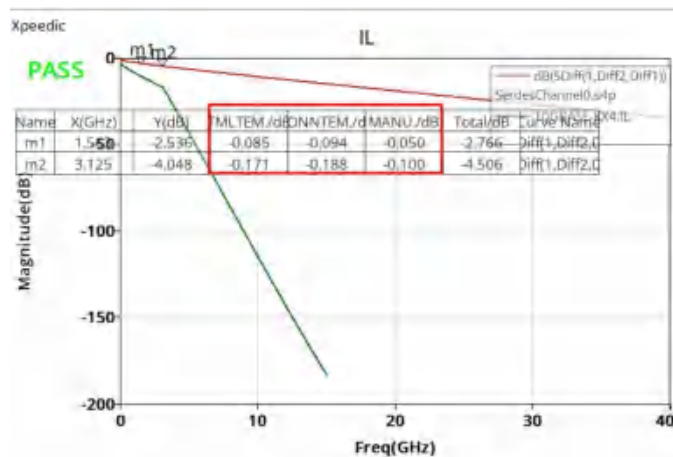


6.13.2.2 损耗收集仿真

损耗收集仿真节点设置方法同 6.13.1.2。

6.13.2.3 损耗收集结果查看

仿真完成后在 IL 中会展示出温升损耗、连接损耗等结果。在 libmanager 中也会展示对应的记录。

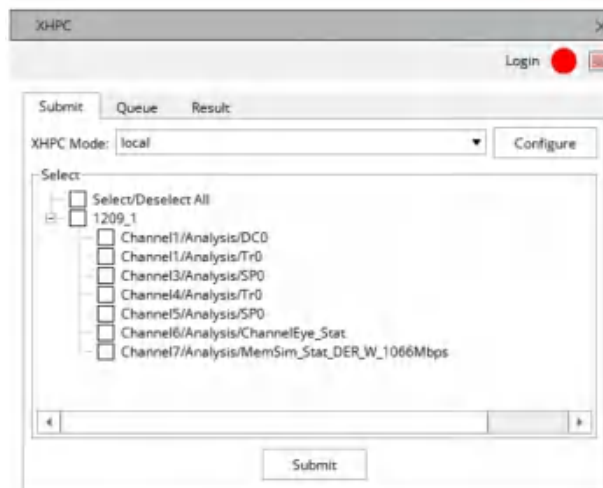
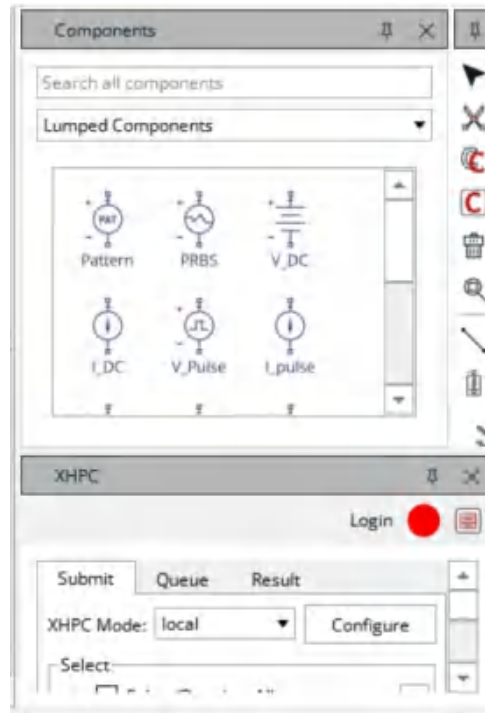


6.14 XHPC

点击 Simulation，选择菜单中的 XHPC,即可使用 XHPC 功能。



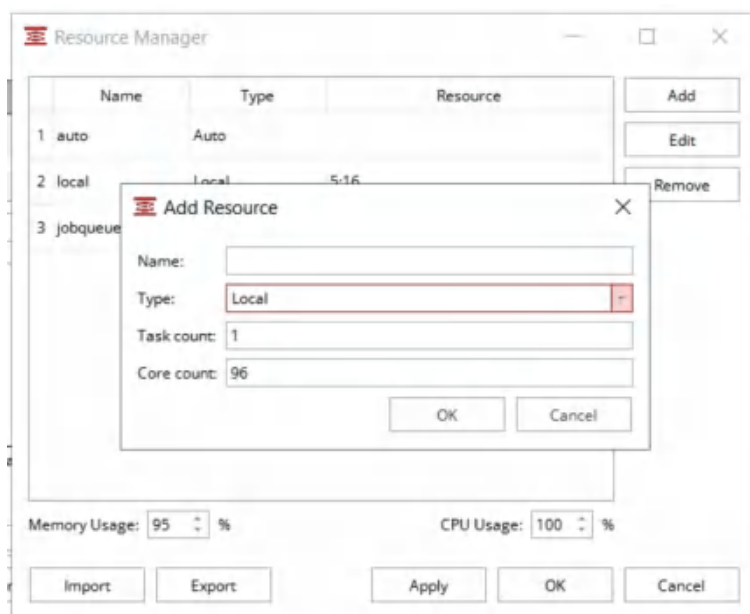
点击 xhpc 窗口，可以拖动到画布上任意位置。



点击 Configure，可以对 xhpc 模式进行配置。

点击 Add，新增一条 resource 记录。Name 中自定义名称，Type 中可以选择 Local、MSPI、Jobqueue 模式。

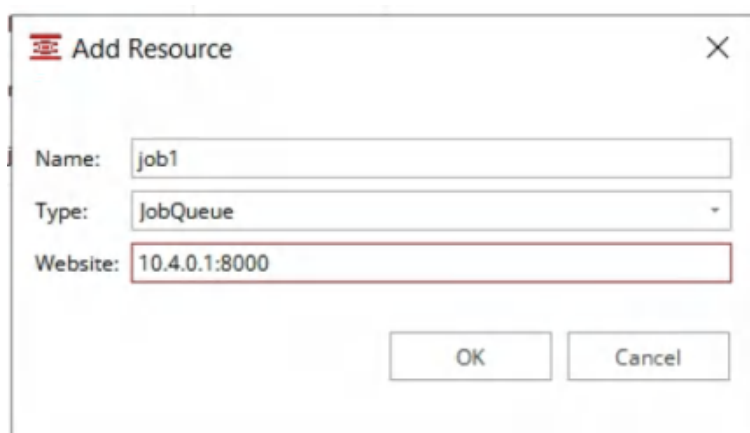
当模式为 Local 模式时，需要设置任务数和核数以及 name。



当模式为 MSPI 时，可以添加多条记录，需要分别设置 ip，task counts，core counts。

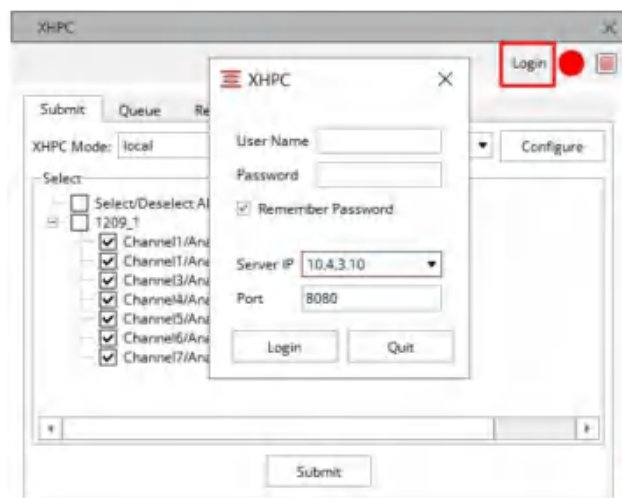


当模式为 JobQueue 时，需要设置 ip+端口以及名称。



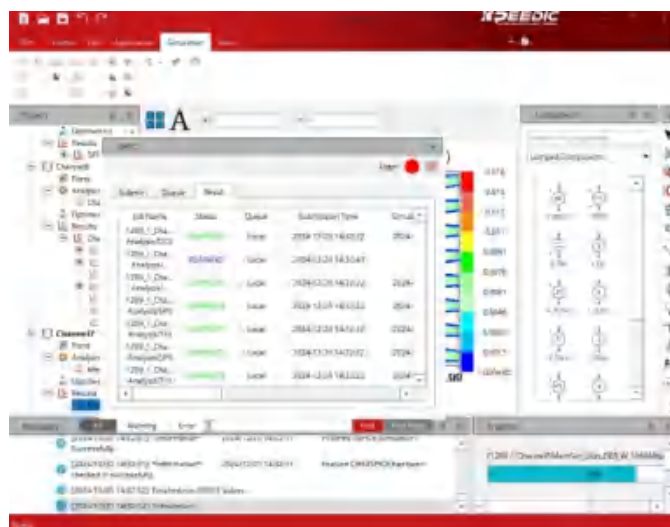
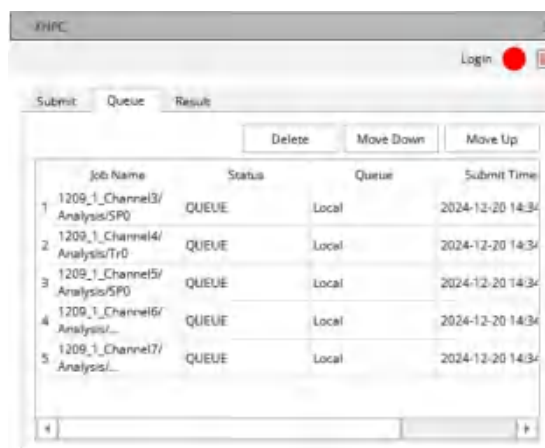
配置完成后，在 XHPC model 中可以切换模式。

当选择非 local 模式时，需要 login 操作。点击 login，配置用户名、密码、ip，port 信息。



第一次点击 Select all，可以勾选全部任务，再次点击会取消全部选中。也可以自定义选择某几条任务运行。选中任务后，点击 Submit，提交后会触发仿真。

Queue 中会显示队列中的任务，在 result 中会展示当前任务的状态信息，complete 表示完成，Running 表示运行中。



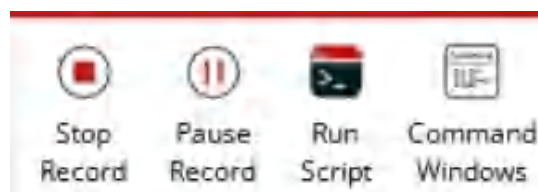
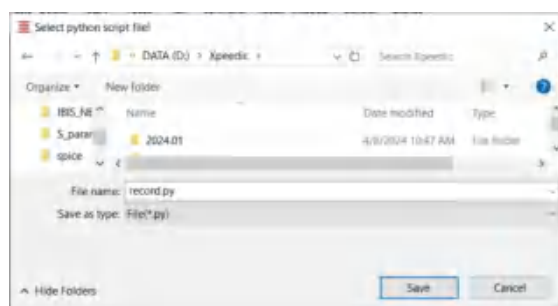
6.15 脚本录制

6.15.1 录制操作

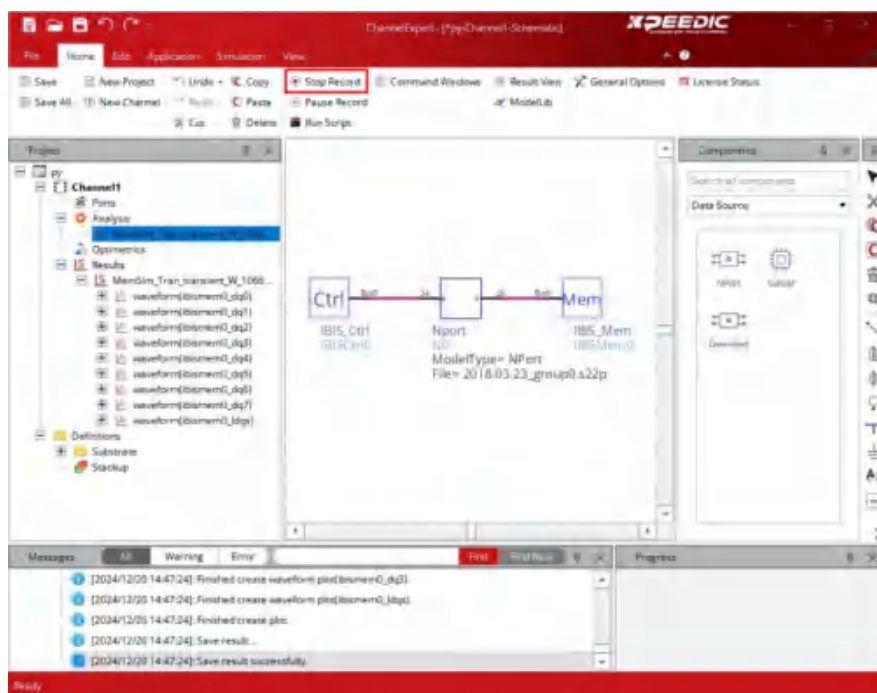
打开软件后，点击 Start Record。



选择脚本保存路径，点击 save。开始启动录制后，启动选项会变成 Stop Record。



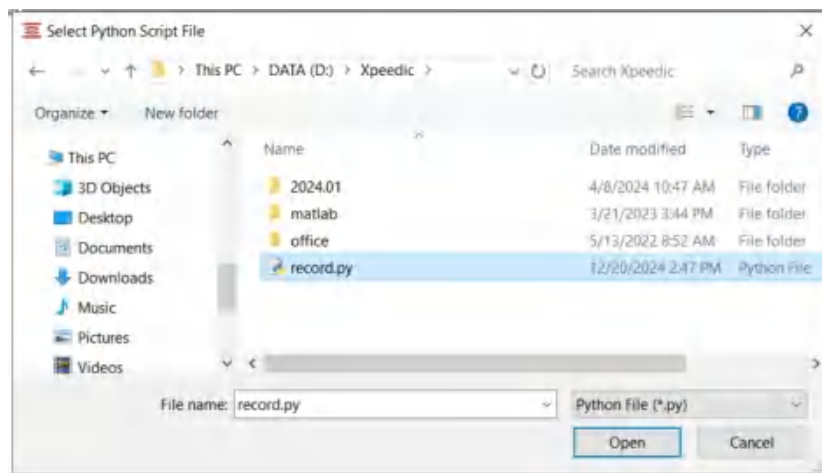
在原理图中做新建工程和仿真等操作后点击 Stop Record。



6.15.2 回放操作

回放如果选择在原 py 文件中的工程所在路径，回放会提示工程已经存在。可以修改 py 中的工程路径，或者删除工程再进程回放操作。

点击 Run Scripts，选择录制工程的 py 文件，点击 open。

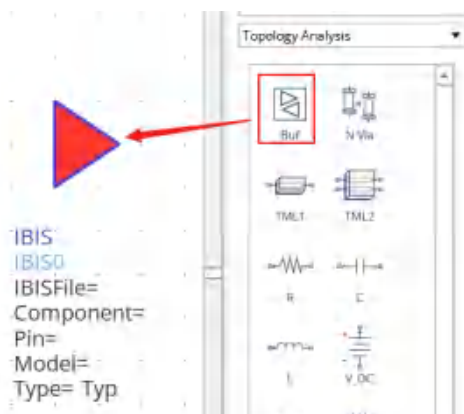


脚本执行完成后，会将录制的所有操作重新执行一遍。

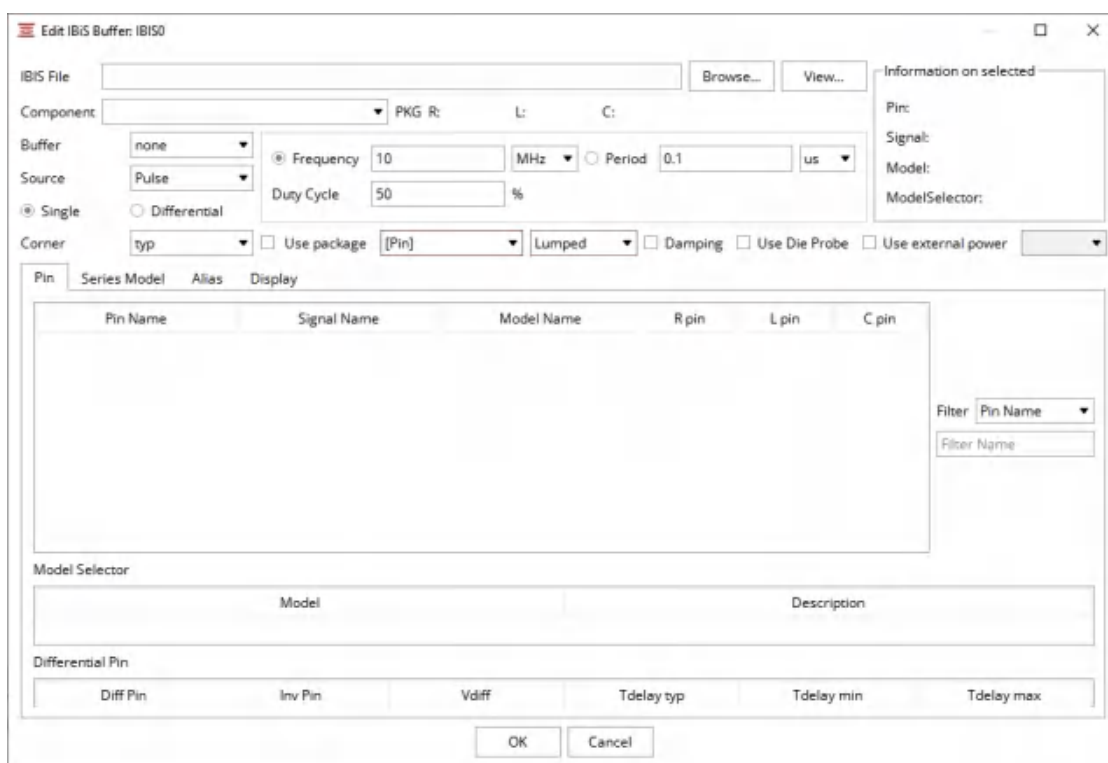
6.16 XTOP 模块仿真

6.16.1 Buf 和 Nvia 创建

在 Topology Analysis 中选择 Buf 器件，拖拽到原理图画布上，会生成一个空的没有赋 ibis 模型的新的 ibis 器件。



双击 IBIS 器件进入到编辑界面

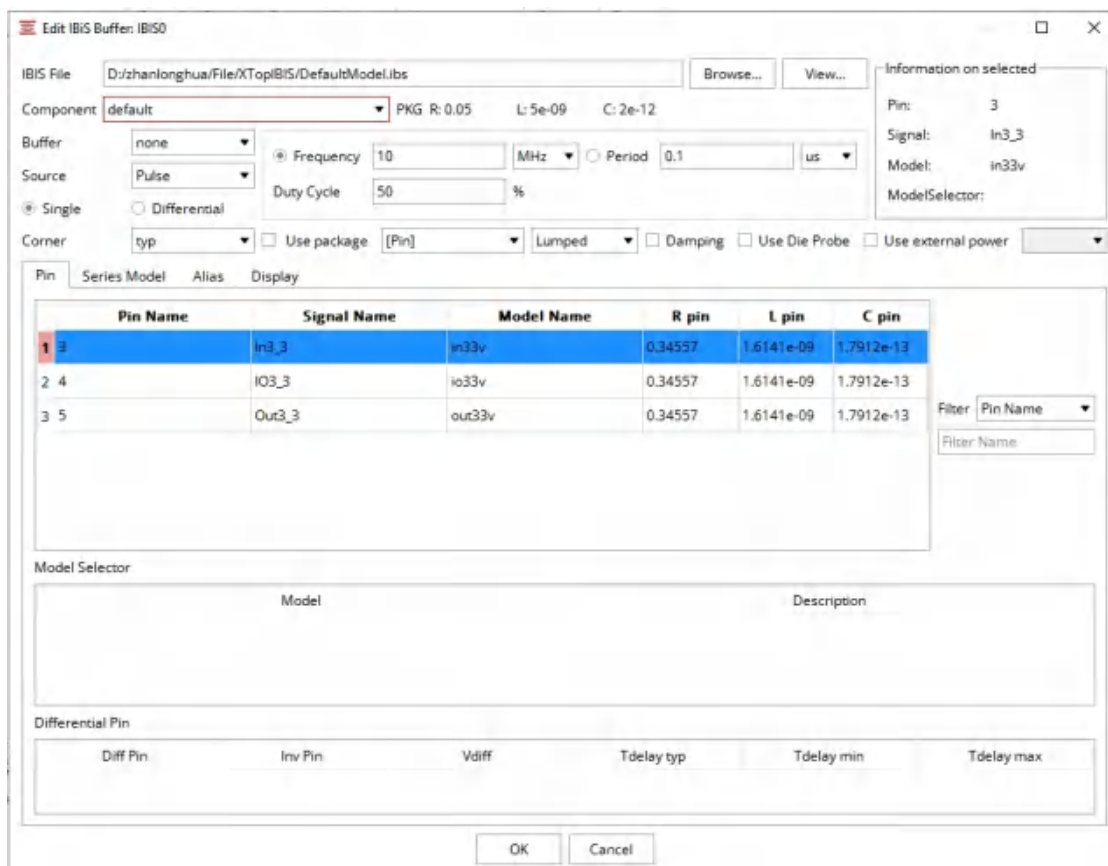


Browse...: 点击 browse 加载 ibis 文件

View: 点击 view 可以快速查看加载的 ibis 文件的内容

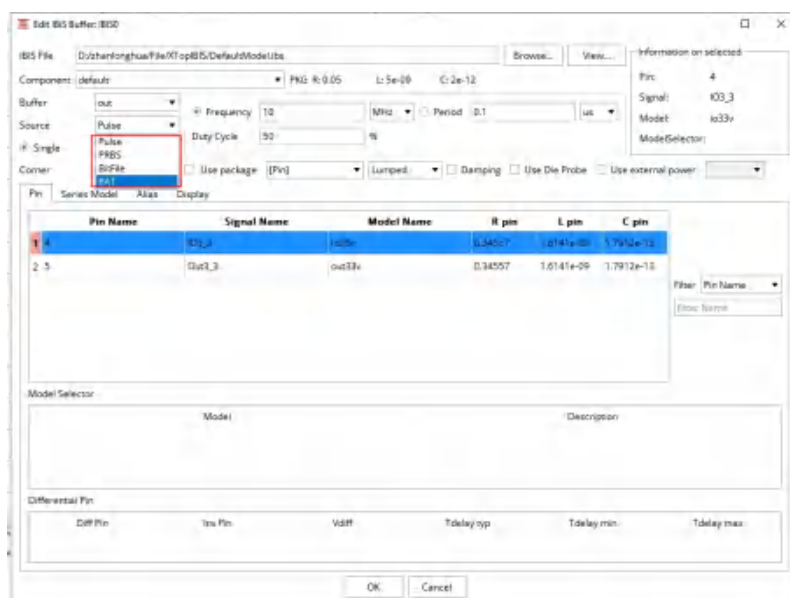
Component: 当 ibis 加载完成后会自动读取当前 ibis 的 component 内容

Buffer: 默认为 None

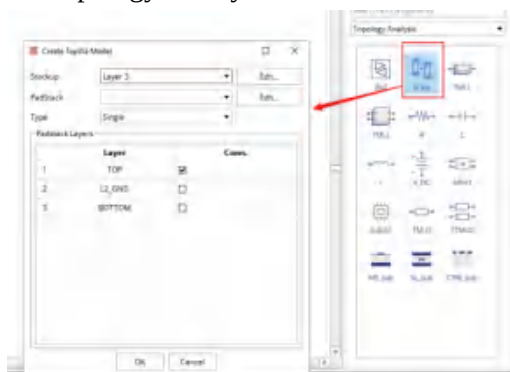


当 buffer 类型选择 out 时，作为 controller 端的 ibis 输出；当 buffer 类型选择 in 时，作为 receiver 端的 ibis 输入。模式可以选择单独或者差分类型，可以手动定义是否使用 package/die probe/internal power 等选项。如果选择 pin/package 下面的 lumped 时，可以额外使用 Damping，在网表中加一个阻尼电阻。Corner 支持 typ/min/max/fast/slow/GlobalCorner。

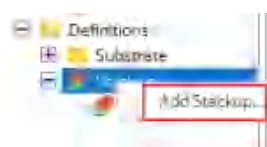
当 ibis 作为输入时，需要配置激励源，当前支持 Pulse/PRBS/Bitfile/PAT 四种激励。



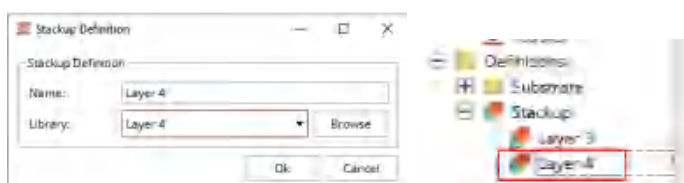
在 Topology Analysis 中选择 N via 器件，拖拽到画布中。



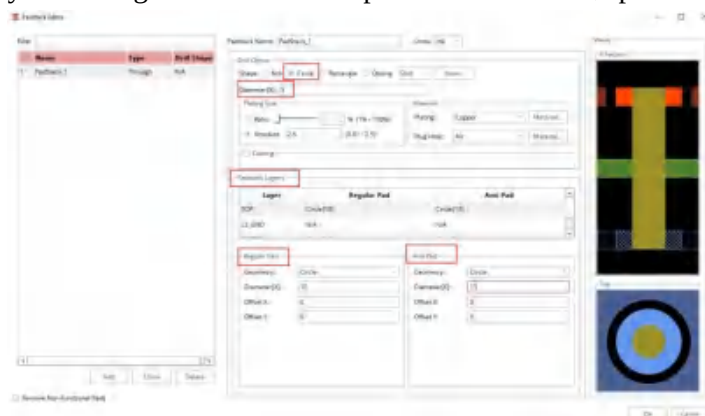
可以使用预置的 stackup，或者重新添加一个 stackup。在树节点的 Stackup 处点击鼠标右键，选择 Add Stackup。



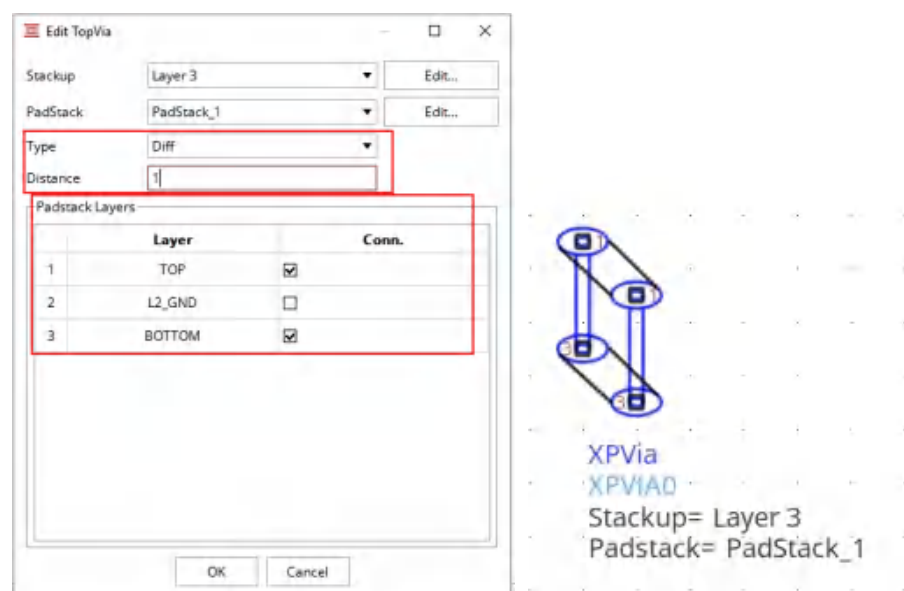
配置 name 和 Library 后点击 ok。



点击 padstack 后面的 Edit，编辑 padstack。在编辑界面配置 shape/Diameter/Padstack Layers 的 Regular Pad 和 Anti pad，可以定义多个 padstack。



在 via 的编辑界面可以选择单端孔或者差分孔，同时需要配置选择过孔的 layer 层位置，当选择差分孔时，需要配置 distance。

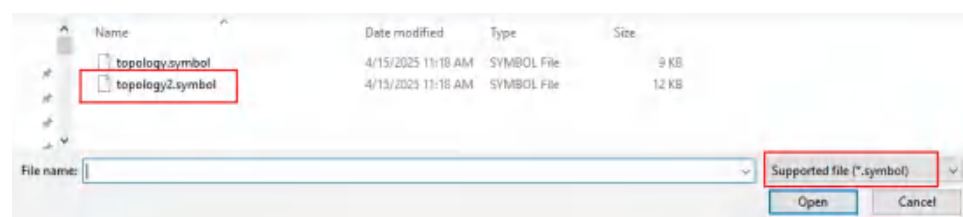


6.16.2 Xtop Rerun

在 Application 中选择 XTOP



选择 topology2.symbol，即可查看拓扑结构。



[illegible]

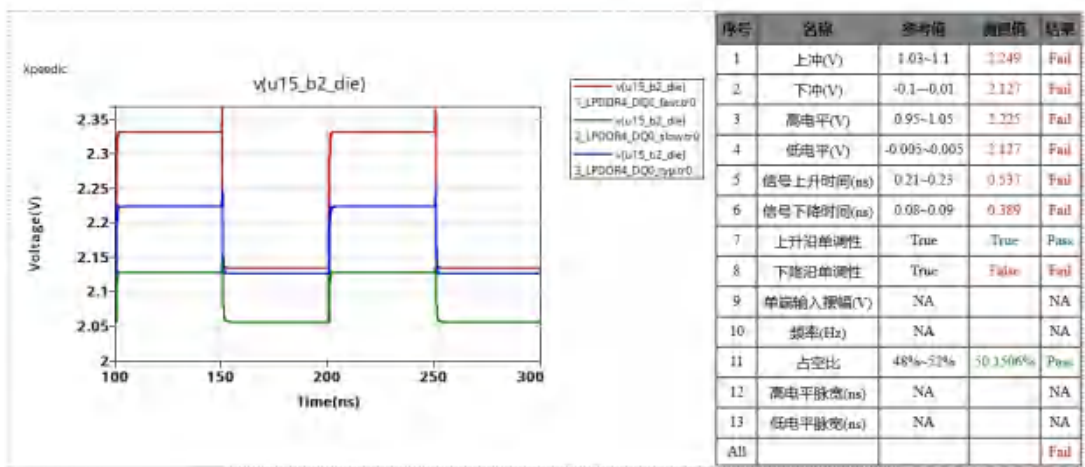
Run Viewing topology

[illegible][illegible]

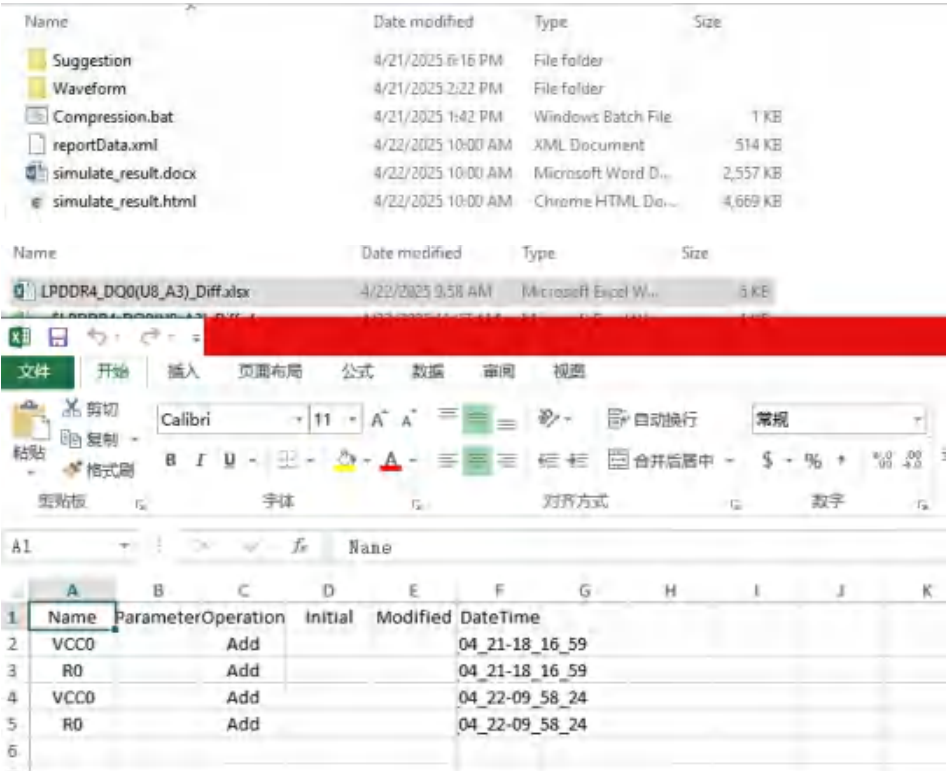
Run	Viewing topology
-----	------------------



等批量更新报告完成后，会自动弹出新的 html 报告，并更新测量数据和结果到新的报告中。

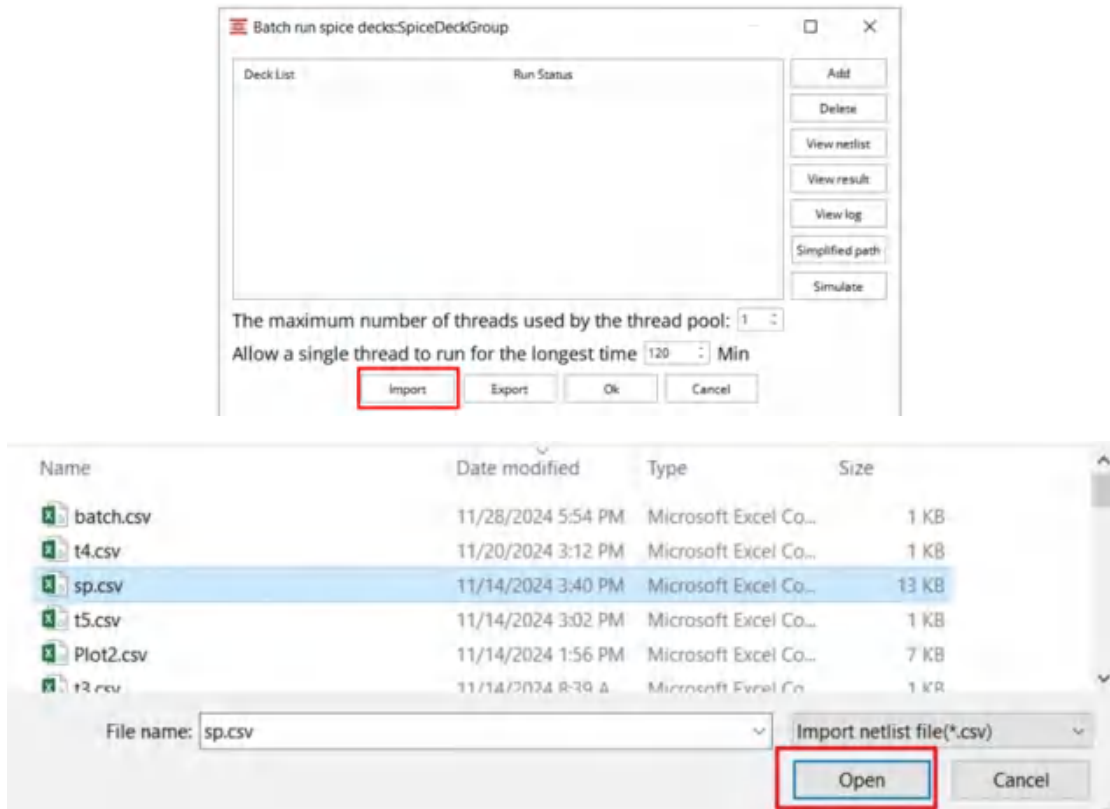


更新报告过程中的数据变更以 excle 的形式存放在 Suggestion 中。



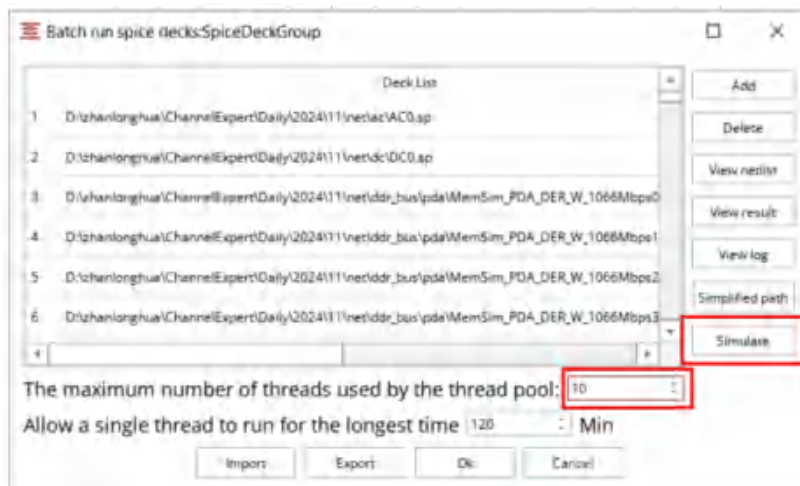
6.17 批量仿真网表

若需要同时仿真多个网表，则可使用 SpiceDeck 功能：

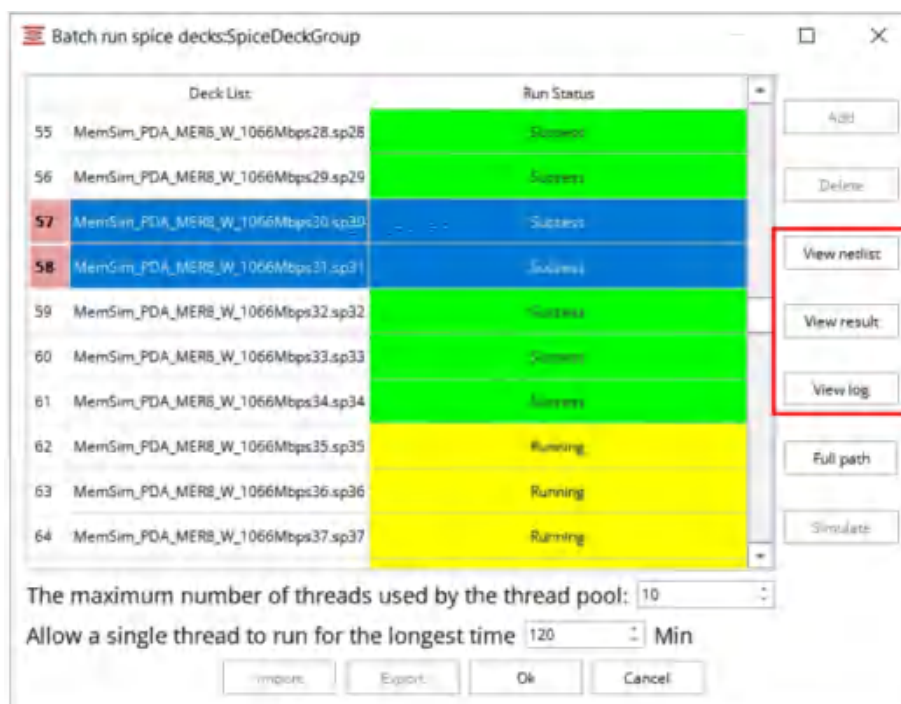
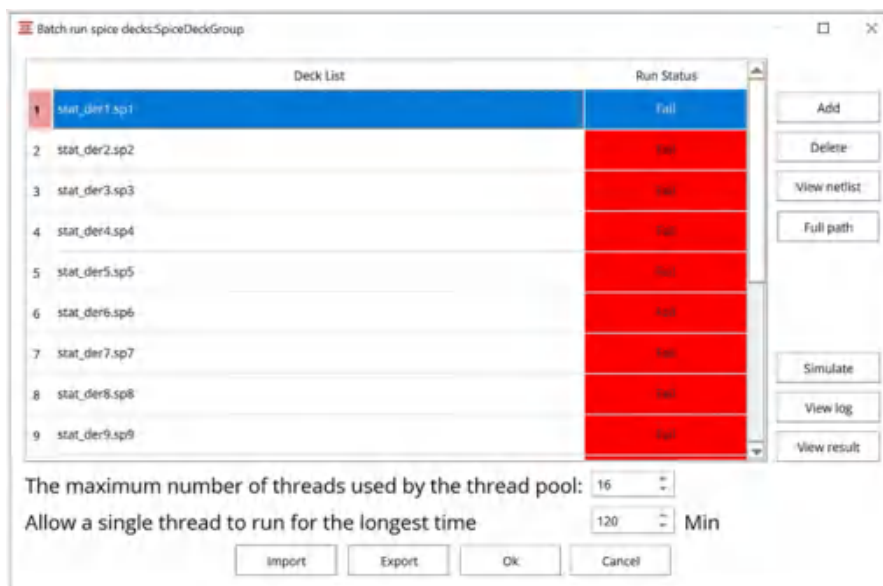


该界面中，Delete 为删除网表，View netlist 为查看网表内容，Full path 为查看网表所在全路径，Simulate 为开始仿真，View log 为查看仿真过程产生的 log，View result 为查看仿真完成的结果，The maximum number of threads used by the thread pool 为最大一次性仿真网表数量，Allow a single thread to run for the longest time 为允许单个网表运行的最大时间，若超过该时间，则停止仿真。

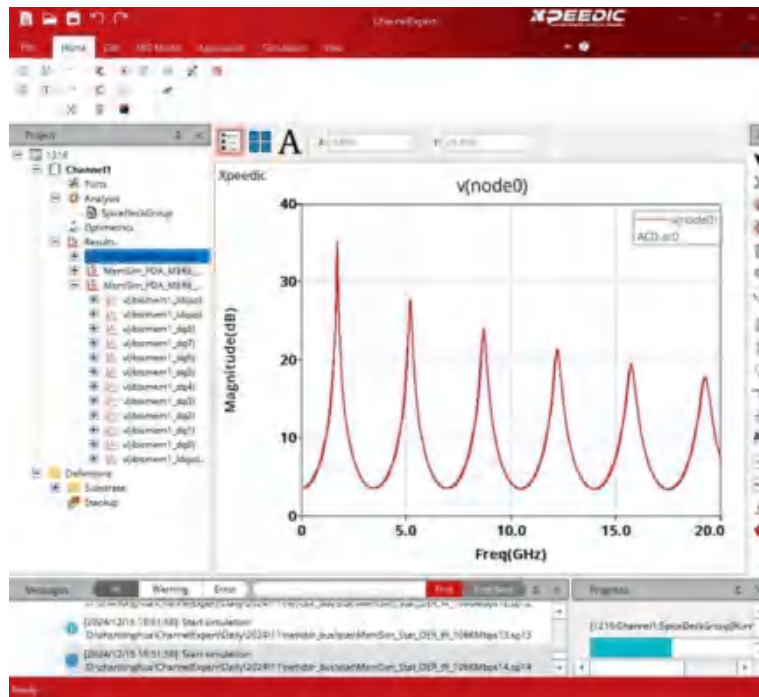
点击 Simulate，开始仿真，



仿真结束后，会在界面展示运行状态（Success/Fail/Error）：



点击 View，查看仿真结果：



7 开源软件声明

The source code comprises of both Open Source and proprietary software components.

The Open Source components are:

Qt 5.15.11 (Windows)

Qt for Application Development lets you create applications for desktop and mobile platforms. It contains all the Qt APIs and the Qt Creator IDE for seamless cross-platform development.

Qt for Application Development is dual-licensed under commercial and open source licenses. The commercial Qt license gives you the full rights to create and distribute software on your own terms without any open source license obligations. With the commercial license you also have access to the official Qt Support and close strategic relationship with The Qt Company to make sure your development goals are met.

Qt for Application Development is also available under GPL and LGPLv3 open source licenses. Qt tools and some libraries are only available under GPL. See the comparison chart for details. The Qt open source licensing is ideal for use cases such as open source projects with open source distribution, student/academic purposes, hobby projects, internal research projects without external distribution, or other projects where all (L)GPL obligations can be met.

Qt 5.7.0 (Linux)

Qt for Application Development lets you create applications for desktop and mobile platforms. It contains all the Qt APIs and the Qt Creator IDE for seamless cross-platform development.

Qt for Application Development is dual-licensed under commercial and open source licenses. The commercial Qt license gives you the full rights to create and distribute software on your own terms without any open source license obligations. With the commercial license you also have access to the official Qt Support and close strategic relationship with The Qt Company to make sure your development goals are met.

Qt for Application Development is also available under GPL and LGPLv3 open source licenses. Qt tools and some libraries are only available under GPL. See the comparison chart for details. The Qt open source licensing is ideal for use cases such as open source projects with open source distribution, student/academic purposes, hobby projects, internal research projects without external distribution, or other projects where all (L)GPL obligations can be met.

Clipper2 1.0.0

The Clipper Library (including Delphi, C++ & C# source code, other accompanying code, examples and documentation), hereafter called the "Software", has been released under the following license, terms and conditions:

Boost 1.79

Boost Software License - Version 1.0 - August 17th, 2003

http://www.boost.org/LICENSE_1_0.txt

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the Software covered by this license to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-

INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

QtXlsx 0.3

MIT License

Copyright (c) 2013-2014 Debao Zhang hello@debao.me

All right reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MPFR 2.4.2

MPFR is free. It is distributed under the GNU Lesser General Public License (GNU Lesser GPL), version 3 or later (2.1 or later for MPFR versions until 2.4.x). The library has been registered in France by the Agence de Protection des Programmes under the number IDDN FR 001 120020 00 R P 2000 000 10800, on 15 March 2000. This license guarantees your freedom to share and change MPFR, to make sure MPFR is free for all its users. Unlike the ordinary General Public License, the Lesser GPL enables developers of non-free programs to use MPFR in their programs.

MPC 0.8.1

The library is built upon and follows the same principles as GNU MPFR. It is written by Andreas Enge, Mickaël Gastineau, Philippe Théveny and Paul Zimmermann and is distributed under the GNU Lesser General Public License, either version 3 of the licence, or (at your option) any later version (LGPLv3+). The GNU MPC library has been registered in France by the Agence pour la Protection des Programmes on 2003-02-05 under the number IDDN FR 001 060029 000 R P 2003 000 10000.

GMP 4.3.2

Since version 6, GMP is distributed under the dual licenses, GNU LGPL v3 and GNU GPL v2. These licenses make the library free to use, share, and improve, and allow you to pass on the result. The GNU licenses give freedoms, but also set firm restrictions on the use with non-free programs.

Boost 1.79

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CGAL 4.14.3

CGAL is licensed under GNU LESSER GENERAL PUBLIC LICENSE (LGPL) Version 3.0. See GNU LESSER GENERAL PUBLIC LICENSE (LGPL) for a complete listing of the GNU LESSER GENERAL PUBLIC LICENSE.

Virtually any software may use Eigen. For example, closed-source software may use Eigen without having to disclose its own source code. Many proprietary and closed-source software projects are using Eigen right now, as well as many BSD-licensed projects.

Eigen 3.3.7

Eigen is Free Software. Starting from the 3.1.1 version, it is licensed under the MPL2, which is a simple weak copyleft license. Common questions about the MPL2 are answered in the official MPL2 FAQ.

Earlier versions were licensed under the LGPL3+.

Note that currently, a few features rely on third-party code licensed under the LGPL: `SimplicialCholesky`, AMD ordering, and `constrained_cg`. Such features can be explicitly disabled by compiling with the `EIGEN_MPL2_ONLY` preprocessor symbol defined. Furthermore, Eigen provides interface classes for various third-party libraries (usually recognizable by the `<Eigen/*Support>` header name). Of course you have to mind the license of the so-included library when using them.

Virtually any software may use Eigen. For example, closed-source software may use Eigen without having to disclose its own source code. Many proprietary and closed-source software projects are using Eigen right now, as well as many BSD-licensed projects.

Python 3.8.8rc1

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 3.9.2 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 3.9.2 alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright © 2001-2021 Python Software Foundation; All Rights Reserved" are retained in Python 3.9.2 alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 3.9.2

or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 3.9.2.

4. PSF is making Python 3.9.2 available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 3.9.2 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 3.9.2 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 3.9.2, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python 3.9.2, Licensee agrees to be bound by the terms and conditions of this License Agreement.

KLayout 0.25.3

KLayout is published under GNU public license GPL version 2 or any later version (www.gnu.org) in compliance with the requirements for using the Qt open source licence. It may be copied and distributed freely.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of Merchantability or fitness for a particular purpose. Please use it AT YOUR OWN RISK.

Any artwork created with KLayout is yours. KLayout can be used freely for any purpose, commercial or private, production or education.

VTK 7.1.1

VTK is an open-source toolkit licensed under the BSD license.

Copyright (c) 1993-2008 Ken Martin, Will Schroeder, Bill Lorensen

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither name of Ken Martin, Will Schroeder, or Bill Lorensen nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

PythonQt 3.2

PythonQt is distributed under the LGPL 2.1 license. It can be used in commercial applications when following the LGPL 2.1 obligations.

The build system of PythonQt makes use of a modified version of the LGPL'ed QtScript generator, located in the "generator" directory.

See <https://code.qt.io/cgit/qt-labs/qtscriptgenerator.git> for details on the original project. Thanks a lot to the QtJambi guys and the QtScript Generator project for the C++ parser and Qt typesystem files!

The PythonQt wrappers generated by the generator located in the "generated_cpp" directory are

free to be used without any licensing restrictions.

The generated wrappers are pre-generated and checked-in for 5.0, 5.3, 5.4 and 5.6, so you only need to build and run the generator when you want to build additional wrappers or you want to upgrade/downgrade to another Qt version. You may use the generator to generate C++ bindings for your own C++ classes (e.g., to make them inheritable in Python), but this is currently not documented and involves creating your own typesystem files.

onemkl 2022.0.3

Intel Simplified Software License (Version April 2018)

Copyright (c) 2018 Intel Corporation.

Use and Redistribution. You may use and redistribute the software (the "Software"), without modification, provided the following conditions are met:

- * Redistributions must reproduce the above copyright notice and the following terms of use in the Software and in the documentation and/or other materials provided with the distribution.

- * Neither the name of Intel nor the names of its suppliers may be used to endorse or promote products derived from this Software without specific prior written permission.

- * No reverse engineering, decompilation, or disassembly of this Software is permitted.

Limited patent license. Intel grants you a world-wide, royalty-free, non-exclusive license under patents it now or hereafter owns or controls to make, have made, use, import, offer to sell and sell ("Utilize") this Software, but solely to the extent that any such patent is necessary to Utilize the Software alone. The patent license shall not apply to any combinations which include this software. No hardware per se is licensed hereunder.

Third party and other Intel programs. "Third Party Programs" are the files listed in the "third-party-programs.txt" text file that is included with the Software and may include Intel programs under separate license terms. Third Party Programs, even if included with the distribution of the Materials, are governed by separate license terms and those license terms solely govern your use of those programs.

DISCLAIMER. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT ARE DISCLAIMED. THIS SOFTWARE IS NOT INTENDED FOR USE IN SYSTEMS OR APPLICATIONS WHERE FAILURE OF THE SOFTWARE MAY CAUSE

PERSONAL INJURY OR DEATH AND YOU AGREE THAT YOU ARE FULLY RESPONSIBLE FOR ANY CLAIMS, COSTS, DAMAGES, EXPENSES, AND ATTORNEYS' FEES ARISING OUT OF ANY SUCH USE, EVEN IF ANY CLAIM ALLEGES THAT INTEL WAS NEGLIGENT REGARDING THE DESIGN OR MANUFACTURE OF THE MATERIALS.

LIMITATION OF LIABILITY. IN NO EVENT WILL INTEL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. YOU AGREE TO INDEMNIFY AND HOLD INTEL HARMLESS AGAINST ANY CLAIMS AND EXPENSES RESULTING FROM YOUR USE OR UNAUTHORIZED USE OF THE SOFTWARE.

No support. Intel may make changes to the Software, at any time without notice, and is not obligated to support, update or provide training for the Software.

Termination. Intel may terminate your right to use the Software in the event of your breach of this Agreement and you fail to cure the breach within a reasonable period of time.

Feedback. Should you provide Intel with comments, modifications, corrections, enhancements or other input ("Feedback") related to the Software Intel will be free to use, disclose, reproduce, license or otherwise distribute or exploit the Feedback in its sole discretion without any obligations or restrictions of any kind, including without limitation, intellectual property rights or licensing obligations.

Compliance with laws. You agree to comply with all relevant laws and regulations governing your use, transfer, import or export (or prohibition thereof) of the Software.

Governing law. All disputes will be governed by the laws of the United States of America and the State of Delaware without reference to conflict of law principles and subject to the exclusive jurisdiction of the state or federal courts sitting in the State of Delaware, and each party agrees that it submits to the personal jurisdiction and venue of those courts and waives any objections. The United Nations Convention on Contracts for the International Sale of Goods (1980) is specifically excluded and will not apply to the Software.

*Other names and brands may be claimed as the property of others.

Pybind11 2.6.2

Copyright (c) 2016 Wenzel Jakob <wenzel.jakob@epfl.ch>, All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Please also refer to the file `.github/CONTRIBUTING.md`, which clarifies licensing of external contributions to this project including patches, pull requests, etc.

7za 19.00

7-Zip is free software with open source. The most of the code is under the GNU LGPL license. Some parts of the code are under the BSD 3-clause License. Also there is unRAR license restriction for some parts of the code. Read 7-Zip License information.

You can use 7-Zip on any computer, including a computer in a commercial organization. You don't need to register or pay for 7-Zip.

7-Zip Copyright (C) 1999-2021 Igor Pavlov.

The licenses for files are:

1) 7z.dll:

- The "GNU LGPL" as main license for most of the code
- The "GNU LGPL" with "unRAR license restriction" for some code
- The "BSD 3-clause License" for some code

2) All other files: the "GNU LGPL".

Redistributions in binary form must reproduce related license information from this file.

Note:

You can use 7-Zip on any computer, including a computer in a commercial organization. You don't need to register or pay for 7-Zip.

PETSc 3.15.0

2-clause BSD license

Copyright (c) 1991-2021, UChicago Argonne, LLC and the PETSc Development Team

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This license DOES NOT apply to any software that may be obtained via the --download-package option of the PETSc configuration. Each of those packages are covered by their own licenses.

SLEPc 3.15.1

BSD-3-Clause

Copyright (c) 2002-2021, Universitat Politecnica de Valencia, Spain All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1.Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2.Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. This license DOES NOT apply to any software that may be obtained via the --download-package option of the SLEPc configuration. Each of those packages are covered by their own licenses.

pyDOE 0.3.5

BSD License (BSD License (3-Clause))

Copyright (c) 2014, Abraham D. Lee

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

mimalloc 209

MIT License

Copyright (c) 2018-2021 Microsoft Corporation, Daan Leijen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,

EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Numpy 1.19.4

Copyright (c) 2005-2021, NumPy Developers.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- * Neither the name of the NumPy Developers nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MS-MPI 10.1.2

MIT License

Copyright (c) Microsoft Corporation. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.