



Delay Calculation in ChimeTime

Release 1

April 15, 2024

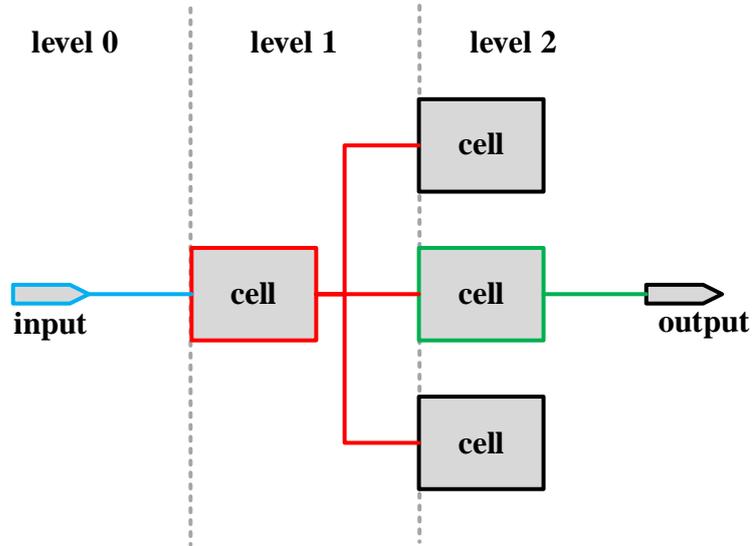
目录

base delay 计算部分	1
1 Stage 介绍	1
2 Net 建模: net RLC tree.....	2
3 Net 模型降阶	2
3.1 ROM	3
3.2 PI model	3
3.3 Ceff.....	3
4 Cell 模型	3
4.1 NLDM model	4
4.2 ccsd model	4
4.3 ccsn model	5
5 Correlation	5
6 Outlier	6
6.1 计算模型差异	6
6.2 Disable arc 引起的差异.....	7
Wave analysis 部分.....	8
1 Driver wave propagation.....	8
2 Sink wave compensation	8
3 Correlation	9
4 Outlier	10
Si 部分	12
1 Aggressor 的选取	12
1.1 Small coupling 过滤.....	12
1.2 快速 Bump 过滤.....	13
1.3 Window overlap 过滤.....	13
1.4 Virtual aggressor 构建.....	14
2 Mesh 构建与降阶	15
3 Bump 计算和注入电流计算.....	15
4 Victim si wave propagation.....	15
4.1 victim driver wave propagation	15
4.2 victim receiver wave propagation.....	15
4.3 victim ROP wave propagation	16
4.4 si delay 计算	16
5 alignment	16
6 Correlation	17
7 Outlier	18
7.1 PT 快速算法产生的 outlier.....	18
7.2 CHT 自身产生的 outlier	19

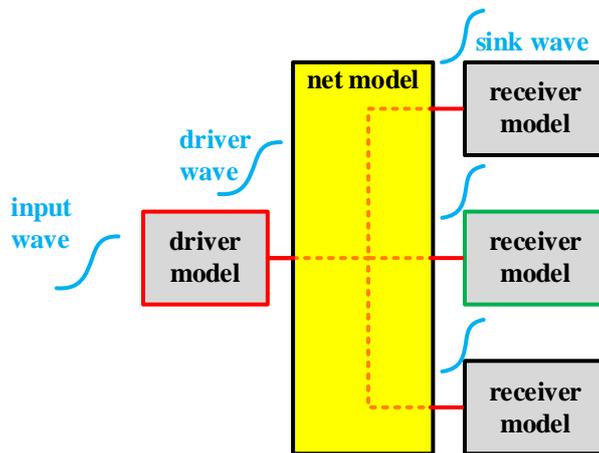
base delay 计算部分

1 Stage 介绍

Stage 是 timing 分析的基本单位，通常由一条 net 和它的 driving cell 组成，如下图中的红色部分和绿色部分分别是一个 stage。当 net 由 input port 驱动，且 input port 没有 driving cell 时，stage 只由该条 net 组成，如下图中的蓝色部分。



时序分析时，将整个 design 划分成一个一个 stage，然后根据信号传播的先后顺序划分等级，例如图中的 design 划分成了 3 个等级。由于下等级 stage 的输入波形或者 transition 需要在上等级 stage 计算完毕之后才能得出，因此 stage 之间存在依赖关系，等级更靠前的 stage 优先计算。前等级的全部 stage 计算完毕之后，下一等级的 stage 才能开始计算。同等级中的 stage 可以并行计算，以加快计算速度。

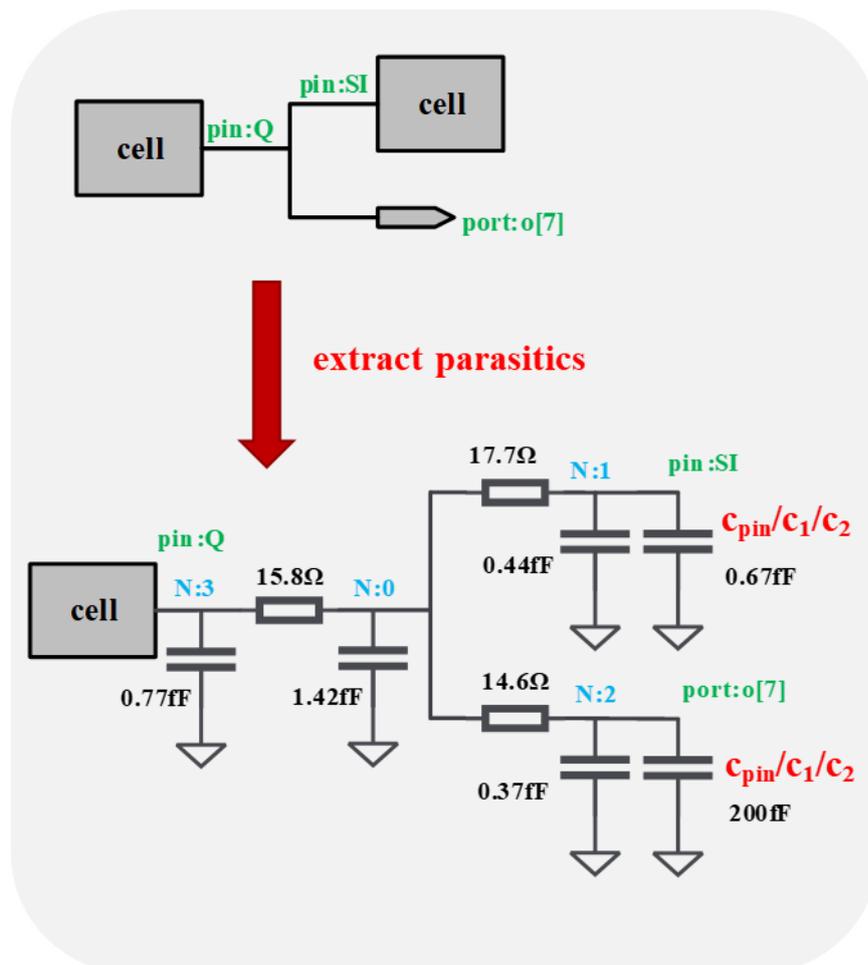


以 level1 的 stage 为例，简单介绍 stage 的计算方式。首先将 driver cell 用 driver model 替换，net 用 net model 替换，load cell 用 receiver model 替换，然后根据电路分析理论计算一个 stage 从输入到输出的信号传播，从而得到 delay 和 transition。driver model 有简单的 NLDM 模型，和更精确 ccs 模型，以及 ccsn 模型。这些都在 cell 对应的时序库中有定义，一般通过 stage 输入信号的 transition 和 stage 中 net 的 cap (对 NLDM 模型和 ccs 模型) 查表得到。CHT 全部支持三种 driver model，可根据参数 driver_cell_analysis_mode 的设定选择哪种

模型。receiver model 是指 receiver cell input pin 看进去的电容，也可以通过查找时序库得到。时序库对 pin 电容有多种定义方式：1.定义固定值，2.定义范围，3.定义查找表 receiver cap。精确程度：3>2>1。CHT 全部支持这三种 pin 电容，并根据参数 receiver_cell_analysis_mode 和 cap_usage 的设定，自由选择一种方式的 pin 电容。Net model 有多种，根据精简程度分为：ROM, PI model, effective cap (Ceff)。CHT 在兼顾速度与精度的情况下自动选择三种模型对 net 进行等效。下面先介绍 net model，再介绍 cell mode。

2 Net 建模：net RLC tree

Net 在实际芯片中通常由多段金属导线和通孔组成，金属导线可能跨越多个金属层。为保证精度，每段导线或通孔通常用多个电阻和电容进行建模，将每段的多个电阻和电容组合起来，最终得到一条 net 的集总参数模型——RLC tree。抽取 net 的寄生参数进行建模的过程叫做寄生参数提取，可由专门的软件完成，然后生成寄生参数文件，时序分析工具可以加载这些寄生参数文件获取 net 的寄生参数信息。从寄生参数文件中加载的 Net RLC tree 不包含 load cell 的 input pin 看进去的电容 (receiver cap)，因此在分析 stage 时序时需要为 net RLC tree 添加 receiver cap。下图中的 C_{pin} , C_1 , C_2 即为 receiver cap。



3 Net 模型降阶

为了保证精度，寄生参数提取得到的 Net RLC tree 的电阻电容非常多，这将使得计算量急剧增加，从而影响计算速度。为了提高计算速度，同时保证一定的精度，需要对 Net RLC tree 进行模型降阶，得到简单的等效模型。CHT 使用的等效模型主要有三种：ROM, PI model,

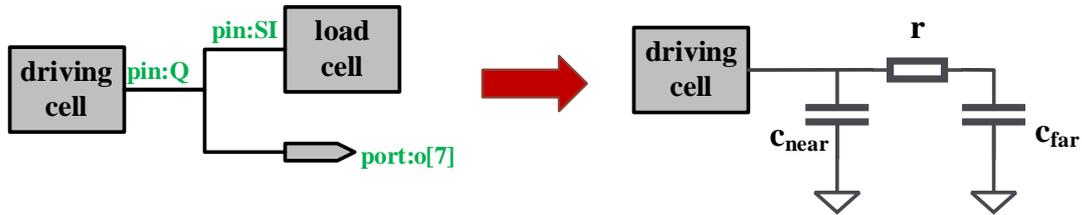
effective cap (Ceff)。CHT 在兼顾速度与精度的情况下自动选择三种模型对 net 进行等效。

3.1 ROM

ROM 为高精度的降阶模型，采用业界公认的 Arnoldi 算法进行等效降阶，可指定阶数，默认为 4 阶模型，即保留 net RLC tree 的 4 个极点和留数。有研究表明，对于 463 个电阻和 470 个对地电容的 net RLC tree，4 阶模型依然可以保证极高的精度，波形传播与原始 net RLC tree 几乎没有区别。

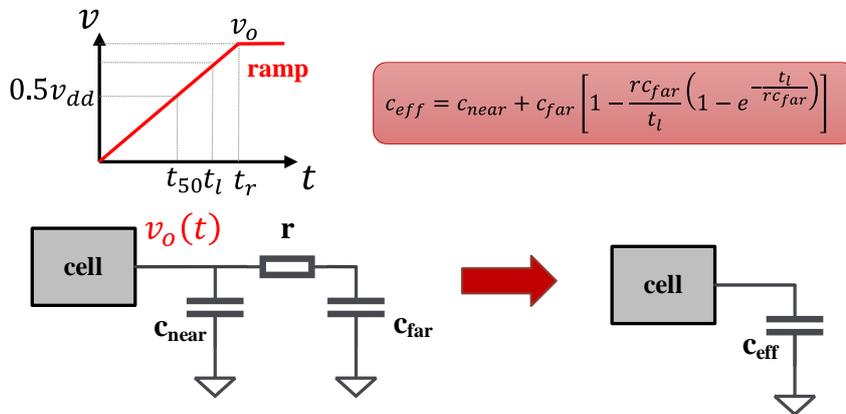
3.2 PI model

PI model 即用两个电容和一个电阻的 π 型网络将 net 进行等效，属于简单的 2 阶模型，通过 moment matching 技术求得。相比 ROM 损失了一定的精度，但模型更加简单，使用更加方便，主要用在求解 driving cell 的 output pin wave 过程中对简单 net 进行等效。



3.3 Ceff

为了方便查找表，可对 PI model 进行进一步简化等效。计算方法有多种，通常通过平均电流相等可将 PI model 等效为一个等效电容 Ceff。如下图所示，假设 cell 的输出波形是一个 ramp 波形，令 t_l 时刻内的平均电流相等，则可以用公式得到 Ceff。



Ceff 可以直接用于 cell model 查表，非常方便。通常一条 net 的 RLC tree 非常复杂，且 cell 的输出波形也不是理想的 ramp 波形，因此在整个波形期间都用一个 Ceff 等效，通常是不准确的。但是对于一些简单的 net，比如电阻很小的 net，用 Ceff 不会带来很大的误差，且能够非常方便快捷的查表计算。

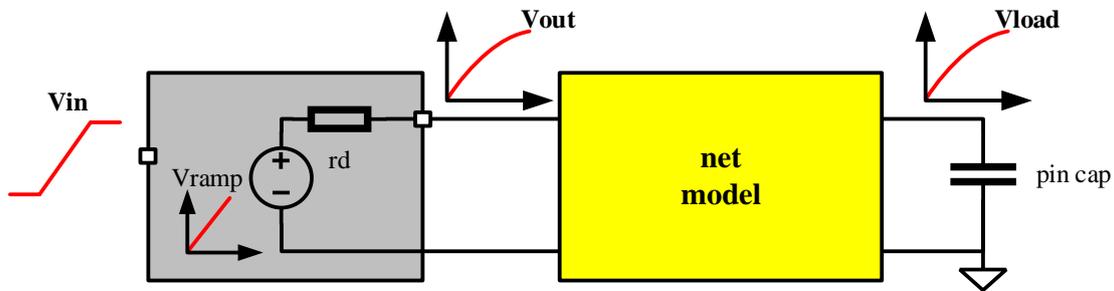
4 Cell 模型

Cell 模型是将 cell 进行等效替换，分为 driver model 和 receiver model。在 net driver 端用 driver model 替换，在 receiver 端用 receiver model 替换。CHT 支持三种 driver model: NLDM model, ccsc model, ccscn model，可根据参数 driver_cell_analysis_mode 的设定选择哪种模型。

receiver model 支持 pin cap 固定值, pin cap 范围, 查找表 receiver cap, 可根据 receiver_cell_analysis_mode 和 cap_usage 的设定进行选择。

4.1 NLDM model

NLDM 即非线性延迟模型, 是非常简单的模型, 具有速度快的优点, 在 28nm 以上的节点能保证较好的精度。其 driver model 通过 cell input transition 和 output net cap 直接查表得到 cell delay 和 cell output transition。然后构建戴维南等效电压源模型, 使得用戴维南等效电压源产生的 cell out wave 相对于 cell input wave 的 cell delay 和 out transition 与查 NLDM 表得到的值相同。Receiver model 通过定义固定值或者定义范围等效。通过将 receiver pin cap 接入 net RLC tree 再进行模型降阶, 结合戴维南等效电压源, 可以计算得到 receiver pin 处的波形。于是可以计算得到 net 的 delay 和 receiver pin 的 transition。Receiver pin 的 transition 作为下一个 stage 的 cell input transition, 用来查表确定下一个 stage 的 driver model。当 driver_cell_analysis_mode 和 receiver_cell_analysis_mode 都设置为 basic 时, CHT 采用这种模式计算 stage 的时序信息。



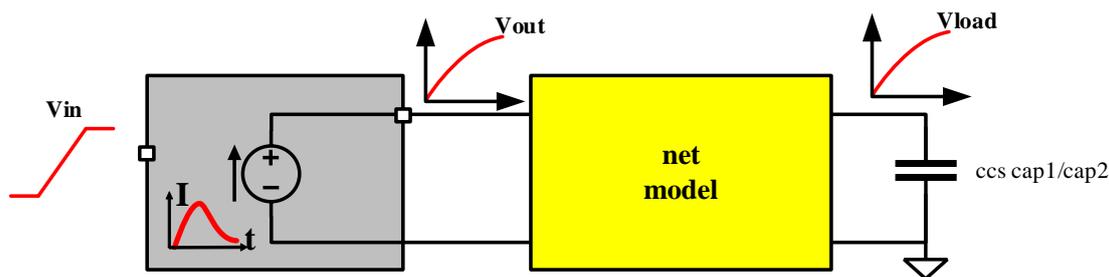
4.2 ccsd model

随着工艺节点的降低, net 电阻增大, NLDM 模型变得不够精确, 同时 miller 电容的反馈效应明显增强, 波形变得扭曲, 在整段波形计算过程中都使用相同的 receiver pin cap 也将损失精度。于是有了更加精确的 ccsd 模型。

ccsd 模型的 driver model 将 cell 等效为电流源, 通过 driver cell input transition 和 driver cell output net cap 查找 ccs current 表得到电流波形, 因此即使 net 电阻增大也不会影响精度。考虑到 miller 电容的反馈效应, Receiver model 通过 2 段 pin cap 等效, 前半段用 c1, 后半段用 c2, 以提高精度。同样, c1 和 c2 通过 receiver cell input transition 和 receiver cell output net cap 查表得到。前面提到, 由于整个波形计算过程中都用 ceff 对 net 进行等效是不够准确的。CHT 在 driver pin 波形计算过程会根据 net 的 ROM 模型实时更新 ceff 的值, 从而更新查找表得到的电流值, 使得计算的 driver pin 波形更加准确。由于有两个 receiver cap, CHT 会根据 2 个 receiver cap 的差异大小, 自动判定是否创建两个 ROM, 以保证精度与速度。当创建两个 ROM 时, 波形前半段使用 c1 创建的 ROM 进行计算, 后半段使用 c2 创建的 ROM 进行计算。

driver pin 波形得到之后, 通过与 net ROM 进行卷积计算即可得到 receiver pin 波形, 从而计算得到 net 的 delay 和 receiver pin 的 transition。Receiver pin 的 transition 作为下一个 stage 的 driver cell input transition, 用来查表确定下一个 stage 的 driver model。

由于 receiver cap 查表需要用到 load cell input pin transition, 而 load cell input pin transition 的计算又需要用到 receiver cap, 因此 cht 在使用 ccsd 模型时需要进行迭代计算, 直到前后 2 次迭代得到的 net 的总电容差异收敛到一定的范围。

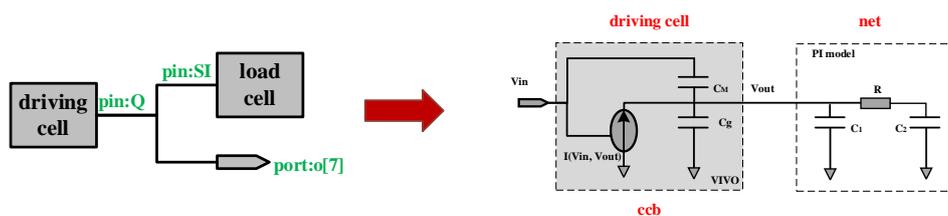


4.3 ccsn model

ccsn model 即 vivo 模型，也即 lib 中的 ccb 部分。通过直接给出随输入和输出电压变化的电流表，以及 miller 电容，更加精确且灵活的描述一个 cell。由于 ccsn 模型直接给出随输入和输出电压变化的电流表，因此可以将任意扭曲的 cell 输入波形传播到 cell 的输出端。这是 Ccsd 模型不具备的优势，因为 ccsd 模型由 input transition 和 output net cap 查表得到，而 input transition 不能反应输入波形的实际形状。因此，CHT 使用 Ccsn model 进行 waveform analysis 和 si analysis。

CHT 使用 ccsn 传播 cell input 波形到 cell output 的计算方式如下图所示：先将 cell 用 ccsn 模型替换，net 用 pi model 替换，得到右侧电路；然后将 pi model 的两个电容（c1 和 c2）以及 miller 电容和对地电容（cg）分别用欧拉近似替换成线性器件；再根据输入波形和输出波形的初始条件采用迭代的方式求解线性方程得到 cell 输出波形。对地电容 cg 的增加是为了提高模型精度，通过 ccb 内部的 output_voltage_rise/fall 表中最后两个点求得。cell input 波形一般由上级 stage 计算得到，如果只有 cell input transition，则根据时序库中的 normlized wave 表和 cell input transition 内插得到 cell input 波形。

cell 输出波形得到之后，通过与 ccsd 相同的计算方式，得到 receiver pin 波形，从而计算得到 net 的 delay 和 receiver pin 的 transition。Receiver pin 的 transition 作为下一个 stage 的 cell input transition 继续下一级的时序计算。



5 Correlation

下面 2 张表分别展示了 CHT 和 PT 的 correlation，以及和 SPICE 的 correlation。从表中可以看出，CHT 的 base delay 计算精度非常高，correlation 全部都在 99% 以上。

表 5-1 CHT 和 PT 的 base slack correlation

index	design	corner	process	setup slack ±5%	hold slack ±3%
1	ca53_cpu_wrap	ff	S28	100.00%	100.00%
2	ca53_cpu_wrap	ss	S28	100.00%	100.00%
3	GPU	ff	S28	100.00%	100.00%
4	GPU	ss	S28	100.00%	100.00%
5	MEDIA	ff	S28	100.00%	100.00%

6	MEDIA	ss	S28	100.00%	100.00%
7	VDH	ff	S28	99.92%	100.00%
8	VDH	ss	S28	100.00%	100.00%
9	ca53_cpu	ff	T28	100.00%	99.99%
10	ca53_cpu	ss	T28	100.00%	100.00%
11	DPD	ff	T28	100.00%	100.00%
12	DPD	ss	T28	100.00%	100.00%
13	ETH	ff	T28	100.00%	100.00%
14	ETH	ss	T28	99.87%	99.97%
15	mali	ff	T28	100.00%	100.00%
16	mali	ss	T28	100.00%	100.00%
17	PCIE	ff	T28	100.00%	100.00%
18	PCIE	ss	T28	100.00%	100.00%
19	PERI	ff	T28	100.00%	99.99%
20	PERI	ss	T28	100.00%	100.00%
21	R52	ss	T28	100.00%	100.00%
22	SA	ff	T28	99.99%	100.00%
23	SA	ss	T28	99.93%	100.00%
24	SOC	ff	T28	100.00%	100.00%
25	SOC	ss	T28	100.00%	100.00%
26	VENC	ff	T28	100.00%	100.00%
27	VENC	ss	T28	100.00%	100.00%

表 5-2 三星 7ns 工艺下 CHT 与 SPICE 的 base path correlation

Corner	Type	Total Pahts	Avg Diff	Criteria 1 ($\pm 3\%$) passing ratio	Criteria 2 ($\pm 4.5\%$) passing ratio
SSPG/0.71/125C	Setup	1000	0.47%	100.0%	100.0%
	Hold	1000	-0.25%	100.0%	100.0%
FFPG/0.79/125C	Setup	1000	0.19%	100.0%	100.0%
	Hold	1000	-0.52%	100.0%	100.0%
SSPG/0.61/-25C	Setup	1000	0.86%	100.0%	100.0%
	Hold	1000	-0.90%	100.0%	100.0%
FFPG/0.90/-25C	Setup	1000	0.77%	99.8%	100.0%
	Hold	1000	0.39%	100.0%	100.0%

6 Outlier

6.1 计算模型差异

在 CHT 和 pt 的 driver 和 receiver 模型参数都设置成 advanced, 且 cell 有 ccsc model 时, CHT 采用 ccsc model 计算 cell 的 delay 和 transition, 而 PT 采用 NLDM model 计算 cell 的 delay 和 transition, 从而导致了少许 stage 有较大差异。

如下表所示, 在 cell input transition 差异不大的情况下, CHT 用 NLDM 计算方法将得到与 PT 非常一致的结果, 而 CHT 用 ccsc 计算方法得到的 delay 和 cell output transition 都与 PT

存在很大差异。

		delay max rise (ns)			delay max fall (ns)		
		cell input trans	cell out trans	cell delay	cell input trans	cell out trans	cell delay
PT	value	0.1565	0.0358	0.0785	0.1504	0.0357	0.0769
CHT	value	0.1596	0.0188	0.0629	0.1496	0.0183	0.0607
CCSD	diff (CHT - PT)	0.0031	-0.0170	-0.0156	-0.0008	-0.0173	-0.0161
	diff ratio (diff/PT)	2.0%	-47.5%	-19.9%	-0.5%	-48.6%	-21.0%
CHT	value	0.1499	0.0369	0.0765	0.1418	0.0344	0.0726
NLDM	diff (CHT - PT)	-0.0066	0.0011	-0.0019	-0.0086	-0.0013	-0.0043
	diff ratio (diff/PT)	-4.2%	3.1%	-2.5%	-5.7%	-3.6%	-5.6%

对于 net delay 小于一定值（如 3ps）的情况，CHT 为了加快计算速度，将采用 NLDM model 计算。而 PT 依旧采用 ccsc model 计算，从而导致了少许 stage 有较大差异。此问题在最新版本中有改善。

如下表所示，在 cell input transition 差异不大的情况下，CHT 用 ccsc 计算方法将得到与 PT 非常一致的结果，而 CHT 用 NLDM 计算方法得到的 delay 和 cell output transition 都与 PT 存在很大差异。

		delay max rise (ns)			delay max fall (ns)		
		cell input trans	cell out trans	cell delay	cell input trans	cell out trans	cell delay
PT	value	0.0375	0.1171	0.1000	0.0497	0.0711	0.0769
CHT	value	0.0383	0.1182	0.0989	0.0488	0.072	0.0656
CCSD	diff (CHT - PT)	0.0008	0.0011	-0.0011	-0.0009	0.0009	-0.0113
	diff ratio (diff/PT)	2.1%	0.9%	-1.1%	-1.8%	1.3%	-14.6%
CHT	value	0.0383	0.1153	0.0736	0.0488	0.0692	0.0428
NLDM	diff (CHT - PT)	0.0008	-0.0018	-0.0264	-0.0009	-0.0019	-0.0341
	diff ratio (diff/PT)	2.1%	-1.5%	-26.4%	-1.8%	-2.7%	-44.3%

6.2 Disable arc 引起的差异

当某些 cell 的 arc 被全部 disable 之后，后续的 stage 将没有任何信号传播过来。对于这种情况，PT 将后续每个 stage 的 input trans 都视为 0 计算 delay 和 transition，transition 并不往后传播。而 CHT 则只将后续第一个 stage 的 input trans 视为 0 进行计算，再下一级的 stage 将用上一级 stage 计算出来的 input transition 进行计算。从而导致出现较大的 outlier。

如下表的例子，由于 cell input transition 的差异，cell delay 存在很大的差异。但是当 PT 的 cell input transition 设置成跟 CHT 一致时，cell delay 与 CHT 非常一致。

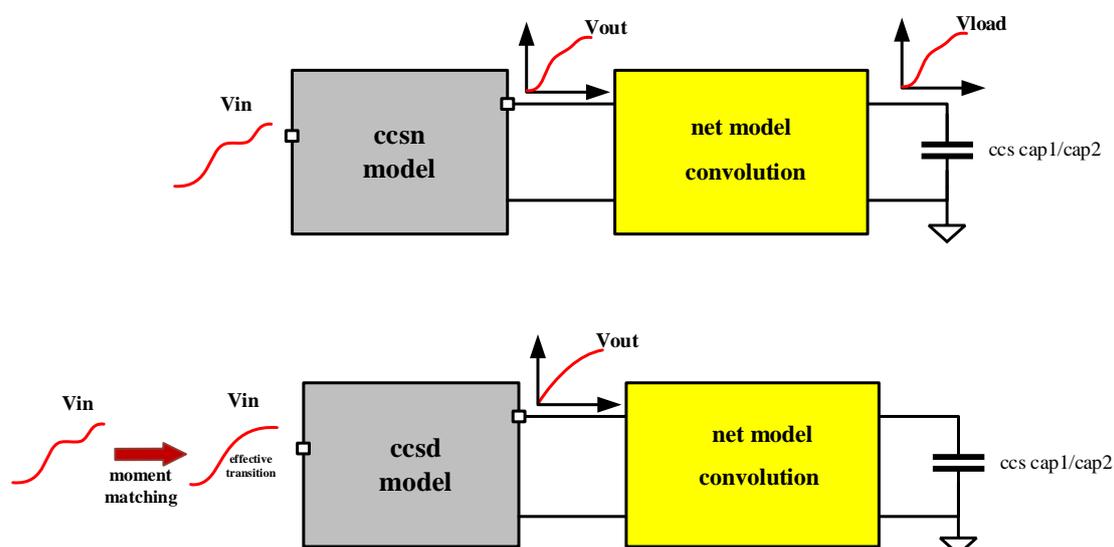
		delay max rise (ns)			delay max fall (ns)		
		cell input trans	cell out trans	cell delay	cell input trans	cell out trans	cell delay
PT		0.0000	0.1551	0.0694	0.0000	0.1566	0.0702
PT with same trans		0.1836	0.1782	0.1782	0.1836	0.1805	0.1782
CHT		0.1836	0.1683	0.1831	0.1887	0.1832	0.1832

Wave analysis 部分

Wave analysis 是分析时序路径的波形效应，如波形的长尾效应和米勒效应等。当 cell 的输入波形明显偏离库表征过程中所使用的波形时，如果不考虑这些影响，将可能导致延迟计算不准确。在 16 纳米及更小的尺寸下，波形效应比较严重，因此 waveform analysis 非常重要。CHT 从两个方面来考虑波形效应：driver wave propagation 和 sink wave compensation。

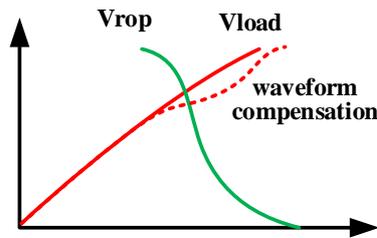
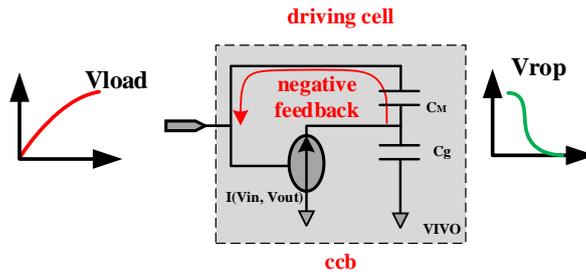
1 Driver wave propagation

CHT 进行波形传播的方式主要有两种：当存在 ccsn 模型时，使用 ccsn 模型直接将 driver cell input 波形传播到 driver cell output。当没有 ccsn 模型时，通过 moment matching 技术将 driver cell input 波形和时序库中的 normalized wave 进行波形匹配，得到等效的 effective transition，然后通过 effective transition 查找 ccsd 表进行波形传播。



2 Sink wave compensation

Driver wave 得到之后，通过 net 的降阶模型，使用卷积计算得到 receiver pin 处的波形 (sink wave or receiver wave)。由于 load cell 的 miller cap 存在负反馈效应，为了提高精度，CHT 对 sink wave 进行了校正。校正的基本原理是电流相等。首先，将未校正的 sink wave 通过 ccsn 波形传播到一个 ccb 后面，得到 ROP wave。计算 ROP wave 与 sink wave 的重叠度。如果存在较大的重叠度，说明 ROP wave 通过 miller cap 对 sink wave 存在较强的负反馈效应。此时需要对 sink wave 进行校正。根据 ROP wave 和 sink wave 计算放大系数 K ，进而得到增强的等效对地电容 $cap=k*\text{miller cap}$ 。由于电容增大，在电流相等的前提下，达到相应的电压值的时间将增加。因此对 sink wave 的时间点进行调整，得到最终调整后的 sink wave。调整后的 sink wave 将继续用 ccsn 模型或者 effective transition 传播至下一级。



3 Correlation

下表展示了 CHT 和 PT 在考虑 waveform propagation 和 si 效应之后的 slack correlation。从表中可以看出，CHT 的 waveform propagation 的整体精度也是非常高的，除个别 case（红色标出）低于 99% 以外，其余 case 的 correlation 全部都在 99% 以上。

表 3-1 CHT 和 PT 的 waveform (含 si 效应) slack correlation

index	design	corner	process	setup slack $\pm 5\%$	hold slack $\pm 3\%$
1	MAIA	ff	C1X	100.00%	100.00%
2	MAIA	ss	C1X	100.00%	100.00%
3	MAIA	tt	C1X	100.00%	100.00%
4	ROCKET	ff	C1X	99.95%	99.98%
5	ROCKET	ss	C1X	97.91%	99.84%
6	ROCKET	tt	C1X	99.97%	99.98%
7	TCVR	ff	C1X	99.58%	98.41%
8	TCVR	ss	C1X	99.95%	99.73%
9	TCVR	tt	C1X	99.94%	99.29%
10	TW5GRFDS	ff	C1X	99.92%	99.96%
11	TW5GRFDS	ss	C1X	99.70%	100.00%
12	TW5GRFDS	tt	C1X	99.89%	99.99%
13	TWST	ff	C1X	99.88%	100.00%
14	TWST	ss	C1X	99.94%	99.96%
15	TWST	tt	C1X	99.96%	99.99%
16	AUDIO	ff	S12	99.89%	100.00%
17	AUDIO	ss	S12	99.89%	100.00%
18	DSS	ff	S12	99.88%	100.00%
19	DSS	ss	S12	99.99%	100.00%
20	enyo	ff	S12	100.00%	100.00%
21	enyo	ss	S12	99.99%	99.99%
22	HSDT0	ff	S12	93.15%	100.00%

23	HSDT0	ss	S12	99.73%	100.00%
24	AUDIO	ff	S14	99.90%	99.99%
25	AUDIO	ss	S14	99.81%	99.99%
26	ccpu	ff	S14	99.13%	99.74%
27	ccpu	ss	S14	99.89%	99.54%
28	HBM	ff	S7	96.17%	99.48%
29	HBM	ss	S7	99.45%	99.92%
30	h32	ss	T7	99.00%	99.73%

4 Outlier

PT 的 waveform propagation 方法未知, 可能与 CHT 的方法存在较大差异, 因此 waveform propagation 之后的结果存在较多的 outlier。这些 outlier 大致可以分为 3 种类型:

第一种类型: CHT 自身产生的 outlier。一般情况是从某一级开始, CHT 与 PT 的 input transition 差异较小但是 input waveform 尾巴阶段存在较大差异, 从而导致该级之后的 delay 和 transition 计算出现了较大差异, 进而影响后续 stage。

如下表的例子。该 stage cell input transition 差异非常小, 且从波形上看差异也非常小。但是 cell 的 output transition 和 cell delay 都存在很大差异。但是 CHT 的值偏悲观。从某种意义上来说, CHT 更能保证 waveform 悲观度。

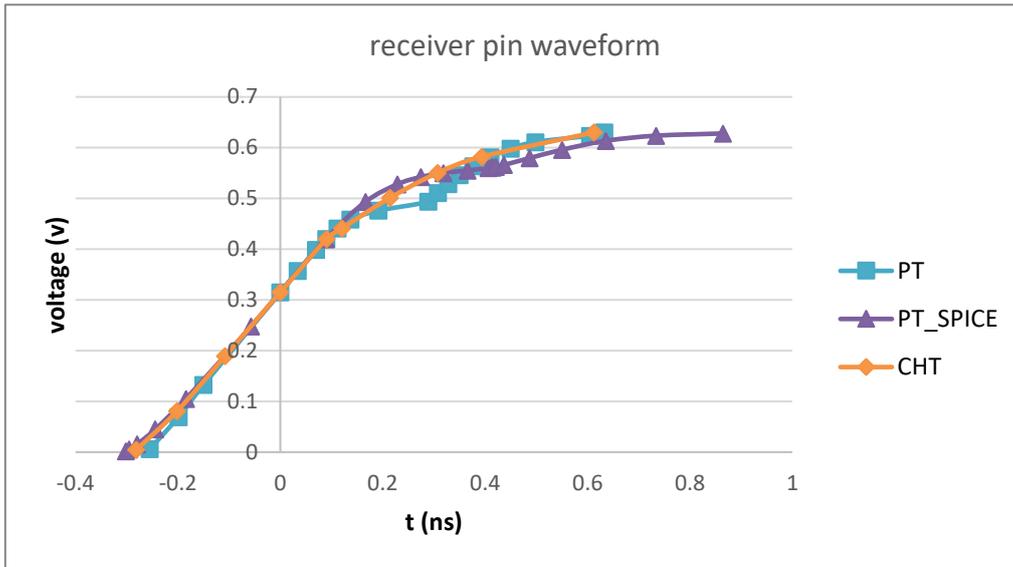
		delay max rise (ns)		
		cell input trans	cell out trans	cell delay
PT	value	0.337	0.495	0.312
CHT	value	0.341	0.623	0.400
	diff (CHT - PT)	0.004	0.128	0.088
	diff ratio (diff/PT)	1.2%	25.9%	28.2%

第二种类型: CHT 与 PT spice 仿真结果相差较小, 而 PT 与 PT spice 仿真结果相差较大的情况。

如下表的例子, PT 与 PT spice 的差异有 19ps, 占比 8.8%

		delay max rise (ns)		
		cell input trans	cell out trans	cell delay
PT SPICE	value	0.082	0.324	0.217
PT	value	0.082	0.241	0.198
	diff (PT - PT_SPICE)	0.000	-0.083	-0.019
	diff ratio (diff/PT_SPICE)	0.0%	-25.6%	-8.8%

第三种类型: CHT/PT 与 PT spice 仿真结果都相差较大, 如下图所示其中一个 stage receiver pin 的波形对比, 该 stage input pin waveform 非常接近, 但 receiver pin 的波形与 spice 仿真结果差异都比较大。



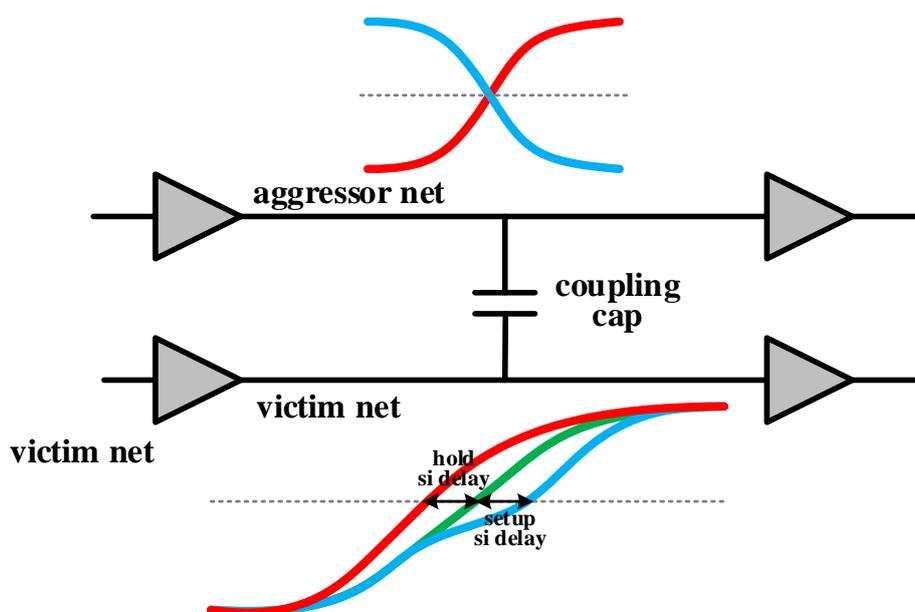
而该 receiver pin 波形将引起下一个 stage 的 delay 出现很大的差异，如下表所示。PT 与 spice 的 cell delay diff 有 20ps，而 cht 与 pt spice 的 cell delay diff 有 28ps，双方都相差比较大。且双方与 spice 相比的乐观悲观方向不一致。

		delay max rise (ns)		
		cell input trans	cell out trans	cell delay
PT SPICE	value	0.430	0.278	0.369
PT	value	0.430	0.215	0.389
	diff (PT - PT_SPICE)	0.000	-0.063	0.020
	diff ratio (diff/PT_SPICE)	0.0%	-22.7%	5.4%
CHT	value	0.430	0.198	0.341
	diff (CHT - PT_SPICE)	0.000	-0.080	-0.028
	diff ratio (diff/PT_SPICE)	0.0%	-37.2%	-7.2%

大部分 outlier 属于第一种类型，即大部分是 CHT 自身算法不够完善导致的问题，但一般 CHT 相较 PT 会更加悲观一些，所以从悲观度来说 CHT 覆盖率更广。而从第二、三种类型来看，PT waveform propagation 也存在算不准的情况。

Si 部分

Si delay（串扰延迟）是指 net 间的耦合电容引起的延迟。在集成电路设计中，由于空间限制，net 被紧密排列，它们之间不可避免地存在寄生电容。当两条相邻 net 中的一条 net（称为 aggressor net，简称 aggr）发生电平变化时，通过耦合电容，aggressor net 会向另一条 net（称为 victim net，简称 vic）注入电流。这种电流注入会导致 victim net 上信号变化提前完成或延迟完成，即产生所谓的 Si delay。



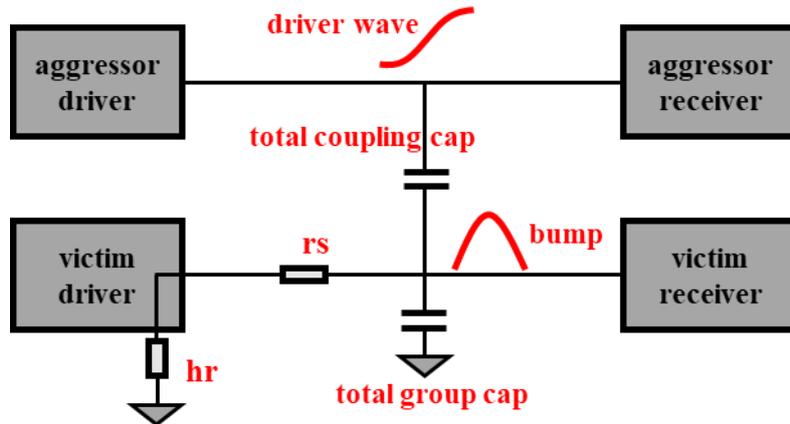
1 Aggressor 的选取

由于 victim net 的 aggressor net 有可能很多，且很多 aggressor net 对 victim net 的影响并不大，为了提高时序分析速度，CHT 只选取部分影响较大的 aggressor net 进行分析。如果参数 `xtk_virtual_attacker_mode` 被设置成 `statistical`，剩下被过滤的部分中影响较大的 aggressor net 将合并组成一个 Virtual aggressor，这样在提高分析速度的同时，也能保证一定的精度。CHT 也考虑了 victim net 与 aggressor net 之间的 arrival window 重叠关系，并根据 window 是否重叠对 aggressor net 进行过滤。这些 aggressor net 过滤的策略与对标工具都是一致的。

1.1 Small coupling 过滤

当 victim net 与 aggressor net 之间的 coupling cap 值小于一定的值（默认为 0.0001 pf）时，aggressor net 将被直接过滤。

1.2 快速 Bump 过滤

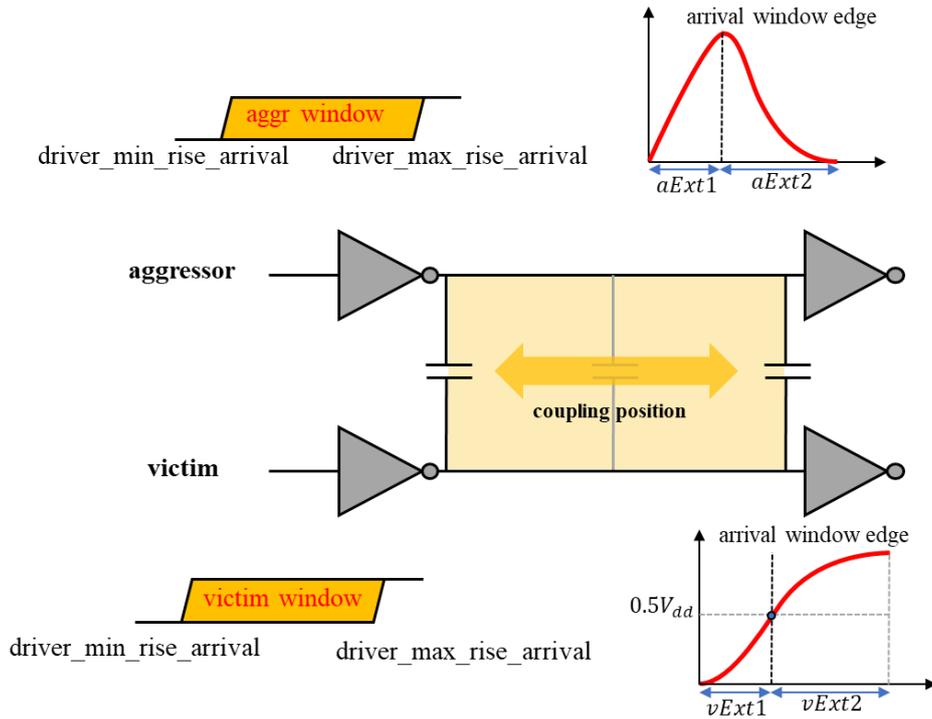


根据上图对 aggressor net 和 victim net 进行建模，快速估算每条 aggressor net 对 victim net 的 bump。其中 hr 为 victim driver cell 的内阻（也称为 holding resistor）。rs 为 aggressor 所有 coupling cap 到 victim driver 的加权平均电阻，由每条 aggressor 的 coupling cap 到 victim driver 的电阻经过 coupling cap 的加权平均得到。大量对比数据证明，该模型估算的 bump 与对标工具的值一致性很好。

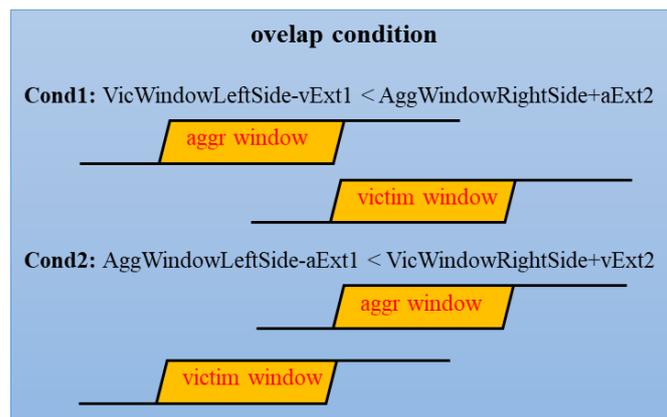
所有 aggressor net 的 bump 估算完毕后，对 bump 由小到大进行排序。对于 bump/vdd 小于参数 `xtk_single_bump_filter_threshold` (默认为 0.01) 的 aggressor net 将被预过滤。将预过滤的 aggressor net 的 bump/vdd 进行从小到大依次累加。累加值小于参数 `xtk_accum_bump_filter_threshold` (默认为 0.03) 的 aggressor net 将被过滤，剩余被预过滤的 aggressor net 将被保留。

1.3 Window overlap 过滤

由于 aggressor net 和 victim net 的不同 sink pin 的 arrival time 都有所差异，如果考虑 sink pin 的 window 将带来非常大的悲观度，以及复杂度。CHT 采用 aggressor net 和 victim net driver pin 的 arrival time 计算它们的 window。如下图以 rise min 的 si delay 计算为例。Aggressor 的 window 范围为[aggressor driver pin rise min arrival time, aggressor driver pin rise max arrival time]。而 victim 的 window 范围为[victim driver pin rise min arrival time, victim driver pin rise max arrival time]。为了保证一定的悲观性，CHT 对 window 进行了一定的扩展。Aggressor window 根据计算的 bump 进行扩展，左边扩展值为 aExt1，右边扩展值为 aExt2。Victim window 根据 transition 进行扩展，左边扩展值为 vExt1，右边扩展值为 vExt2。



当 Aggressor 和 victim 的 window 都扩展之后，就可以通过下面 2 个条件判定 window 是否重叠。与 victim 不重叠的 aggressor net 将不会参与 si delay 计算，而重叠的 aggressor net 将会与 victim net 组成 net mesh 网络，通过搜索与 victim 不同的对齐位置，得到最差情况下的 si delay。如果 window 重叠是因为 window 扩展引起的（即 window 扩展之前不重叠，而 window 扩展之后才重叠），这种情况下，aggressor 产生的 bump 在 victim window 区间内不会全部叠加在 victim net 上，此时 aggressor 对 victim 有影响，但只是部分影响。CHT 将计算 window 重叠区域与 window 扩展长度之间的比例，并通过这个比例因子缩小 aggressor 对 victim 的影响。

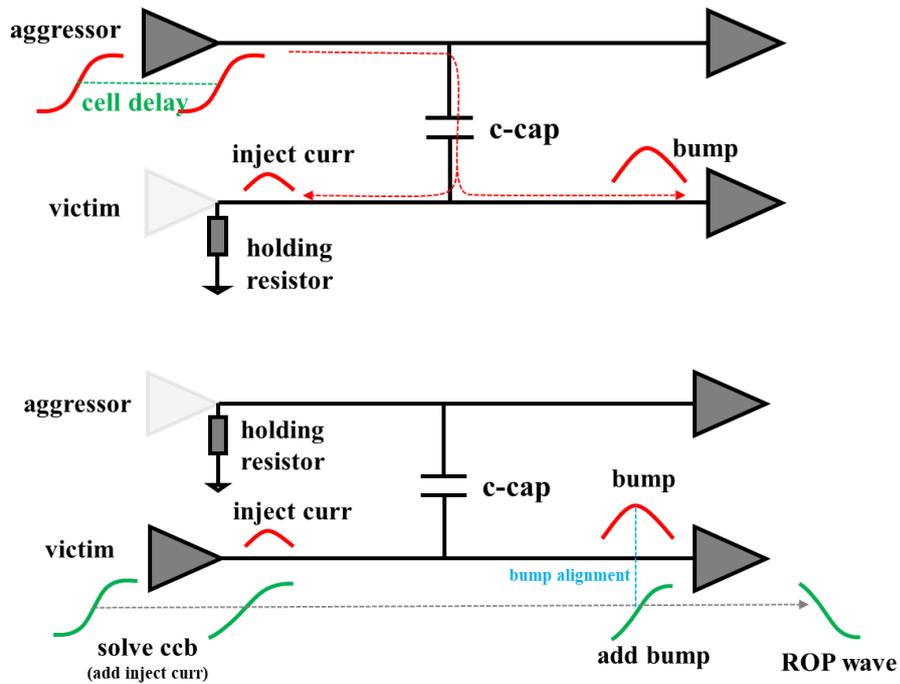


1.4 Virtual aggressor 构建

参数 `xtk_virtual_attacker_mode` 被设置成 `statistical` 时，被快速 bump 过滤的 aggressor net，如果 `bump/vdd` 大于一定的值(默认为 $0.7 * xtk_single_bump_filter_threshold$)，将被组合成一个 virtual aggressor，作为这些 aggressor nets 的等效。Virtual aggressor 的 coupling cap，coupling cap 到 victim driver 的电阻，以及等效 aggressor driver transition 都是通过统计学计算得到，并根据与快速估 bump 相同的方式得到其 bump，作为 Virtual aggressor 对 victim net 的影响。

2 Mesh 构建与降阶

将没有被过滤的 aggressor net 与 victim net 的 net RLC tree 拼接成 2 个 mesh，如下图所示。其中一个 mesh（记为 aggr mesh）在 victim 的 driver 端连接 victim driving cell 的内阻（也即 holding resistor），主要用来计算 aggressor 在 victim driver 端产生的注入电流，以及在 victim receiver 端产生的 bump。另一个 mesh（记为 vic mesh）在 aggressor 的 driver 端连接 aggressor driving cell 的内阻，主要用来计算 victim net 受 si 效应之后的 stage delay。两个 mesh 用 Arnoldi 算法进行 4 阶模型降阶，得到 4 阶传输函数。



3 Bump 计算和注入电流计算

Aggressor driving cell 通过 ccsn 模型求解得到 aggressor driver 处的波形，再通过 aggr mesh 的 4 阶传输函数卷积计算，分别得到 victim driver 端的注入电流和 victim receiver 端的 bump。

4 Victim si wave propagation

Victim si wave propagation 即在考虑所有 aggressor net 对 victim net 的注入电流和 bump 的情况下，将 victim driver cell input 的波形传播至 victim receiver cell 的 input pin 或者 output pin 处，从而计算有 si 效应的 stage delay。

4.1 victim driver wave propagation

victim driving cell 通过 ccsn 模型求解，同时在求解时考虑所有 aggressor 在此 victim driver 处产生的注入电流，得到受 si 影响的 victim driver 波形。

4.2 victim receiver wave propagation

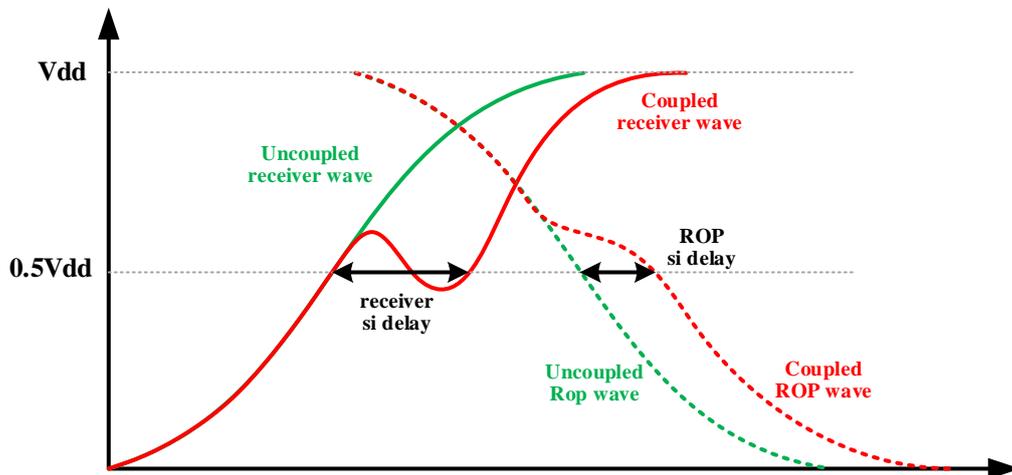
victim driver 波形通过 vic mesh 的 4 阶传输函数卷积计算，得到 victim receiver 处的波形。在此波形的基础上，将所有 aggressor 在此 victim receiver 处产生的 bump 进行线性叠加，得到最终受 si 影响的 victim receiver 波形。通常来说实际的情况并不是一个完全线性叠加的过程，但线性叠加是最简单且精度较高的做法

4.3 victim ROP wave propagation

通常由受 si 影响的 victim receiver 波形直接得出的 si delay 过于悲观，这是因为 victim receiver cell 通常具有低通滤波的特性，受 si 影响的 victim receiver 波形在经过 victim receiver cell 之后，将变得更加平滑。为了保证计算的 si delay 不至于过于悲观，提高精确度，CHT 将受 si 影响 victim receiver 波形进一步传播到 victim receiver cell output。传播方法是使用 ccsm 模型进行传播。因此，如果 victim receiver cell 不存在 ccsm 模型，那么 si delay 将直接在 victim receiver 处计算得出。

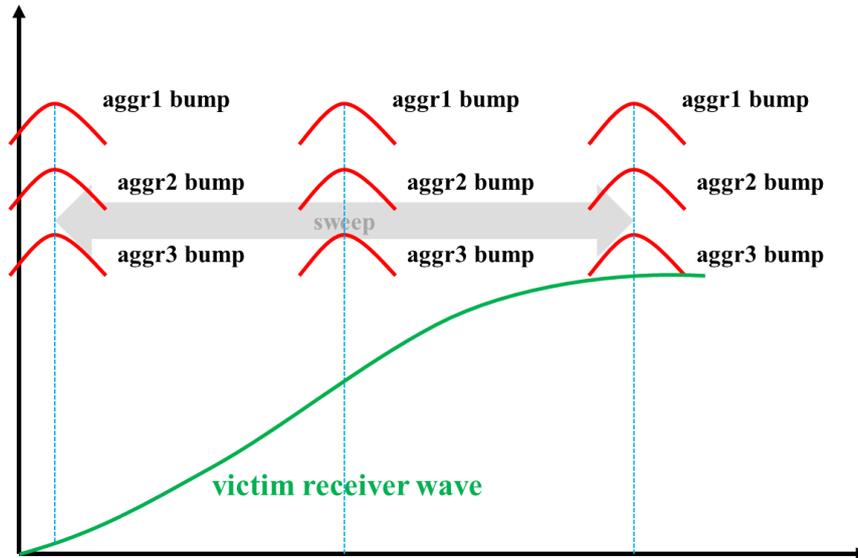
4.4 si delay 计算

受 si 影响 victim Rop 波形或者 victim receiver 波形相对于 victim driver cell input pin 处的 delay 减去 base delay 即可得到 si delay。但通常 ccsm 模型的计算值与 base ccsm 模型计算的值存在一定的差异，为了消除这些影响，CHT 将在 si delay 计算过程中，重新用相同 ccsm 模型以及不含 aggressor 的 net RLC tree 降阶模型计算 victim uncoupled ROP 波形或者 victim uncoupled receiver 波形。通过 victim coupled ROP 波形与 victim uncoupled ROP 波形的差值或者 victim coupled receiver 波形与 victim uncoupled receiver 波形的差值，得到更加精确的 si delay 值。



5 alignment

前面提到 aggressor net 和 victim net 都存在 window。为了保证计算的 si delay 是最悲观情况。需要在 aggressor net 和 victim net 的 window 内，进行多次对齐搜索。每次对齐操作，aggressor net 在 victim driver 处的注入电流和在 victim receiver 处的 bump 叠加在 victim 波形上的时刻都不一样。通过 CHT 特有的搜索算法，在保证速度的情况下，使得到最悲观的 si delay 值。以下是进行对齐搜索操作的 bump 叠加过程示意图。



6 Correlation

下面 2 张表分别展示了 CHT 和 PT 的 correlation，以及和 SPICE 的 correlation。从表中可以看出，CHT 的 si delay 计算精度同样非常高，±5%以内的 correlation 全部都在 99%以上。

表 6-1 CHT 和 PT 的 si slack correlation

index	design	corner	process	setup slack ±5%	hold slack ±3%	hold slack ±5%
1	ca53_cpu_wrap	ff	S28	99.40%	97.33%	99.62%
2	ca53_cpu_wrap	ss	S28	99.25%	98.63%	99.76%
3	GPU	ff	S28	99.87%	100.00%	100.00%
4	GPU	ss	S28	99.93%	100.00%	100.00%
5	MEDIA	ff	S28	99.55%	99.68%	99.91%
6	MEDIA	ss	S28	99.87%	99.77%	99.97%
7	VDH	ff	S28	99.97%	99.98%	99.99%
8	VDH	ss	S28	99.94%	100.00%	100.00%
9	ca53_cpu	ff	T28	99.98%	99.84%	99.99%
10	ca53_cpu	ss	T28	99.20%	99.92%	99.99%
11	DPD	ff	T28	99.70%	100.00%	100.00%
12	DPD	ss	T28	99.99%	100.00%	100.00%
13	ETH	ff	T28	99.99%	99.30%	99.89%
14	ETH	ss	T28	99.99%	99.62%	99.99%
15	mali	ff	T28	99.99%	99.99%	100.00%
16	mali	ss	T28	100.00%	100.00%	100.00%
17	PCIE	ff	T28	98.10%	99.95%	99.99%
18	PCIE	ss	T28	100.00%	99.99%	100.00%
19	PERI	ff	T28	99.97%	99.91%	99.99%
20	PERI	ss	T28	100.00%	100.00%	100.00%
21	R52	ss	T28	100.00%	100.00%	100.00%
22	SA	ff	T28	99.98%	95.71%	100.00%
23	SA	ss	T28	99.99%	100.00%	100.00%
24	SOC	ff	T28	99.84%	99.97%	100.00%

25	SOC	ss	T28	99.99%	99.98%	100.00%
26	VENC	ff	T28	100.00%	99.78%	100.00%
27	VENC	ss	T28	99.99%	100.00%	100.00%

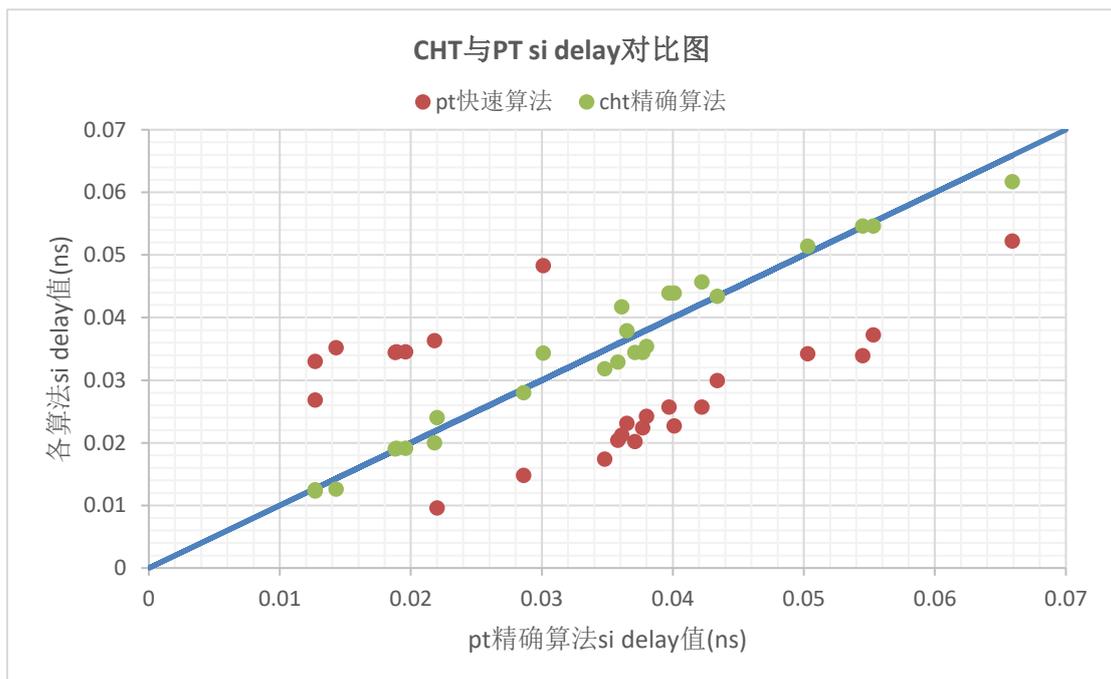
表 6-2 三星 7ns 工艺下 CHT 与 SPICE 的 si stage correlation

Corner	Type	Total Stages	Avg Diff	Criteria 1 ($\pm 5\%$) passing ratio	Criteria 2 ($\pm 7.5\%$) passing ratio
SSPG/0.71/125C	Setup	500	0.7%	99.2%	100.0%
	Hold	500	-0.4%	99.8%	100.0%
FFPG/0.79/125C	Setup	500	-0.2%	99.2%	100.0%
	Hold	500	0.3%	100.0%	100.0%
SSPG/0.61/-25C	Setup	500	1.0%	100.0%	100.0%
	Hold	498	-0.5%	99.6%	100.0%
FFPG/0.90/-25C	Setup	500	0.0%	99.6%	100.0%
	Hold	500	0.3%	100.0%	100.0%

7 Outlier

7.1 PT 快速算法产生的 outlier

PT 在 enable spice 之后其 si delay 值通常会发生非常大的变化。经过长时间的对比研究，推测 PT 为了加快 si delay 分析速度，会采用快速算法。而快速算法计算的 si delay 值不够准确，从而使 CHT 产生了虚假的 outlier。例如，对于 dth 某个 case，我们用脚本选出了 1000 条 si delay 相差最大的 net。在 PT enable spice 之后，其中 75% 的 net 的 si delay 值与 CHT 计算的值变得非常接近。因此这部分 outlier 是由于 PT 快速算法导致的差异，并不是 CHT 计算错误。下图挑选了其中 25 个数据点进行说明，其 aggr 全部是 infinite window 情况，因此避免了 window 的影响。可以看到，CHT 计算的 si delay 与 PT 快速算法有差异，但是跟 PT 精确算法的计算值非常接近。



7.2 CHT 自身产生的 outlier

另外有少部分的 outlier 不是由于 PT 快速算法引起的。这部分 outlier 的原因主要可以归类为下面几种类型：

第一种类型：ccsn 模型选取差异。在计算 victim driver cell 或者 ROP 的波形时，CHT 选取的 ccsn 模型与 PT 的 ccsn 模型来自不同的 arc，导致 si delay 存在差异。CHT 选取 arc 的标准是使得 transition 是最悲观的，而 PT 在某些情况下并不是这一标准。

如下表的例子，CHT 和 PT 在 enable spice 模式下都采用 infinite window 和高精度算法计算 si delay，即规避了算法问题和 window 问题。但是 CHT 和 PT 的 si delay 依旧相差 20ps，spice 仿真结果也验证下来二者相差 14ps（见表倒数第二列数据）。对比 spice 仿真文件，发现二者的 spice 仿真文件的所选 timing arc 不一致，当 CHT spice 仿真时选用相同 arc 时，spice 结果几乎完全一致（见表最后一列数据）。

	Tool (ns)			Spice (ns)	
	si delay (enable spice)	1.5 stage delay (no si)	1.5 stage delay (si)	1.5 stage spice delay (si)	1.5 stage spice delay (si, same arc)
CHT	-0.093	0.305	0.211	0.205	0.193
PT	-0.113	0.303	0.190	0.191	0.191
Diff (CHT - PT)	0.020	0.001	0.021	0.014	0.002

第二种类型：window 差异。CHT 的 window 是以 driver pin 的 arrival time 构建的，而 PT 的 window 可能并不是这种构建方式。且 arrival time 受 base delay 和 waveform propagation 的影响，CHT 和 PT 本身就存在差异，因此 window 可能存在一定的差异。Window 的差异将导致 aggressor 过滤的数目不同，从而导致 si delay 出现 outlier。

这种类型的 outlier 非常容易分辨，只需要将 CHT 和 PT 的 si delay 计算次数都设置成 1 次，CHT 和 PT 都使用 infinite window 进行计算，outlier 的差异将瞬间变得非常小。如下表中的 2 个例子，CHT 的 si delay 与 PT 相差十几皮秒，但是当用 1 次迭代计算的 si delay 值确非常的接近。

	stage1	stage2
PT si delay	0.0091	0.0285
CHT si delay	0.0282	0.0415
diff (CHT - PT)	0.0191	0.013
PT si delay (Iter1)	0.0377	0.0419
CHT si delay (Iter1)	0.0332	0.0415
diff (CHT - PT)	-0.0045	-0.0004

第三种类型：alignment 差异。Alignment 是搜索求解最悲观的 si delay 的关键步骤。由于 holding resistor 的数值差异，bump 的峰值以及峰值位置将出现较大差异，这将导致 alignment 位置可能出现偏差，从而导致 si delay 出现差异。这种问题只能降低差异，但不可避免。这种类型引起的 outlier 通常较小，且数量不多。

如下表的 2 个 stage 为例，默认情况 CHT 使用的快速的 alignment 算法，其计算的 si delay 与 pt 相差 10% 左右。但是如果 CHT 使用穷举法（即 exhaustive_align），进行细微的 align 对齐扫描，CHT 计算的 si delay 与 pt 相差非常小。由于穷举法耗时太长，为了兼顾计算速度，CHT 默认不用穷举法进行 alignment 操作。

		stag1	stage2
pt si delay	value	0.1043	0.0879
cht si delay	value	0.0939	0.0808
	diff (CHT - PT)	-0.0104	-0.0071
	diff ratio (diff/PT)	-11.1%	-8.8%
cht si delay (exhaustive_align)	value	0.1007	0.0862
	diff (CHT - PT)	-0.0036	-0.0017
	diff ratio (diff/PT)	-3.5%	-1.9%