

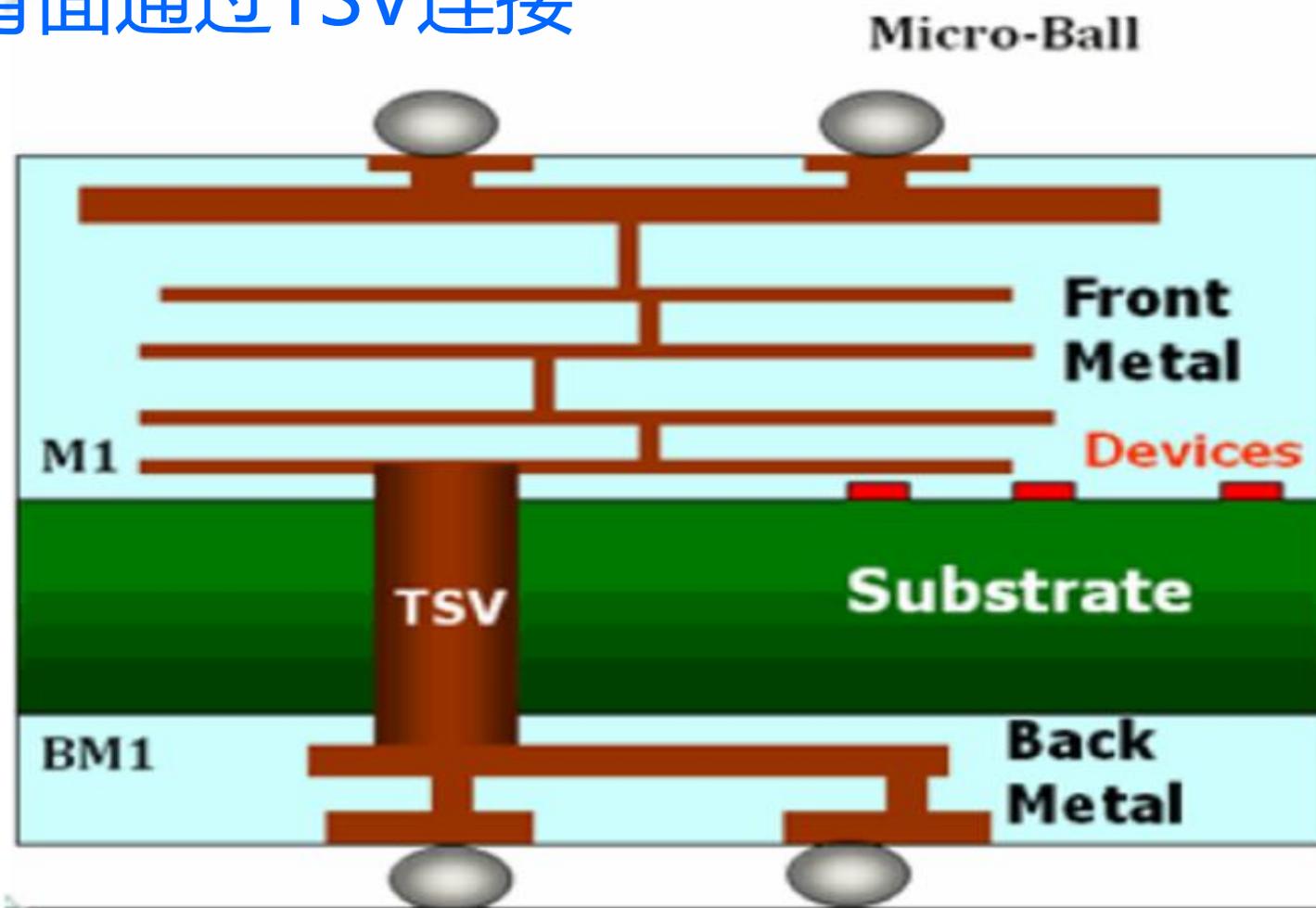
# 3D IC的TSV寄生参数提取流程

# 提纲

- 3D IC的TSV寄生参数提取背景
- 3D IC的TSV寄生参数提取难点
- 衬底内单个TSV提取
- 衬底内多个TSV耦合电容提取
- 衬底之外的TSV图形寄生参数提取
- 多个芯片提取后的网表合并
- 提取流程总结
- 附录： SuperCap工具简介

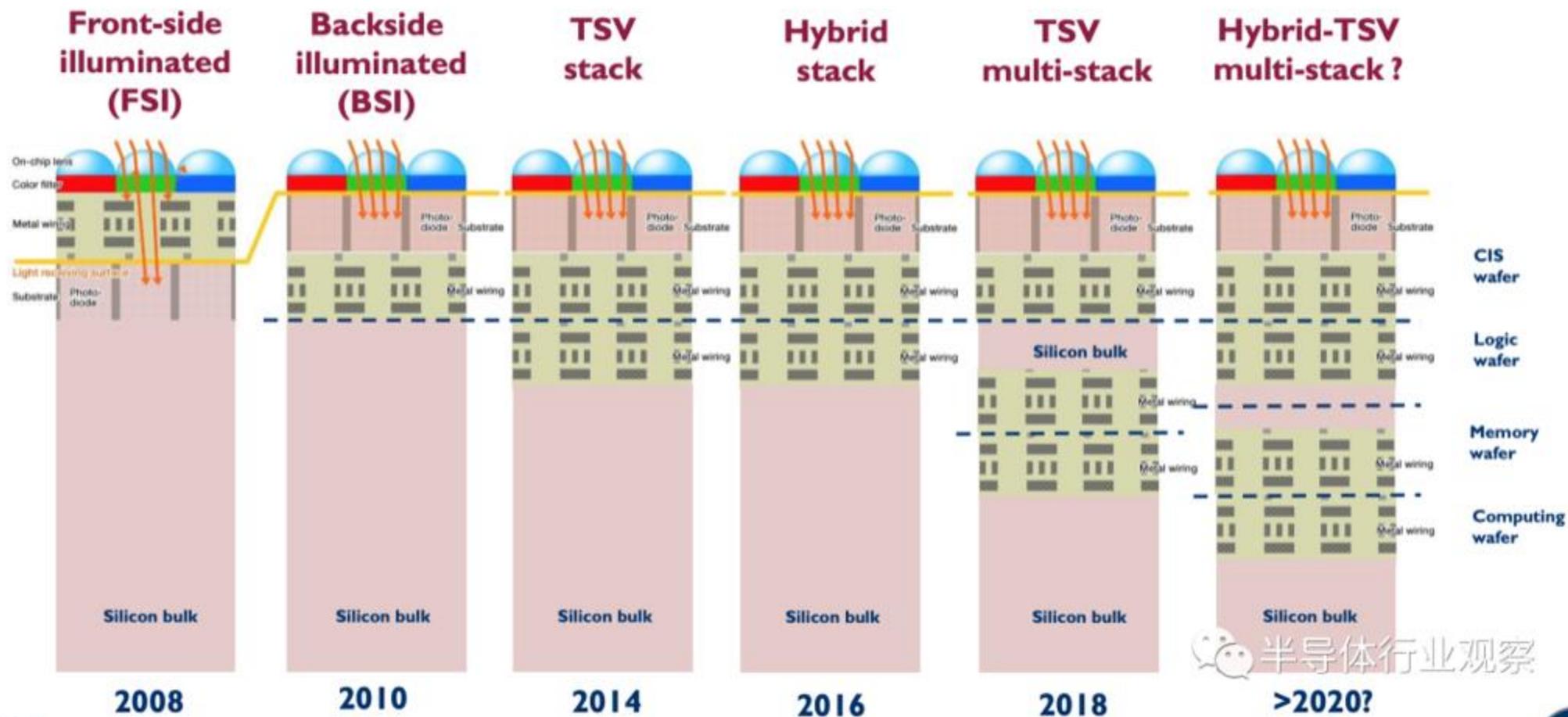
- CIS单个芯片正面和背面通过TSV连接

CIS制造工艺从前照式 (FSI) 到背照式 (BSI)、从背照式 (BSI) 到堆叠式 (Stacked BSI、Triple Stacked BSI) 的两次技术变革, 使得CIS的成像质量实现飞跃式提升。

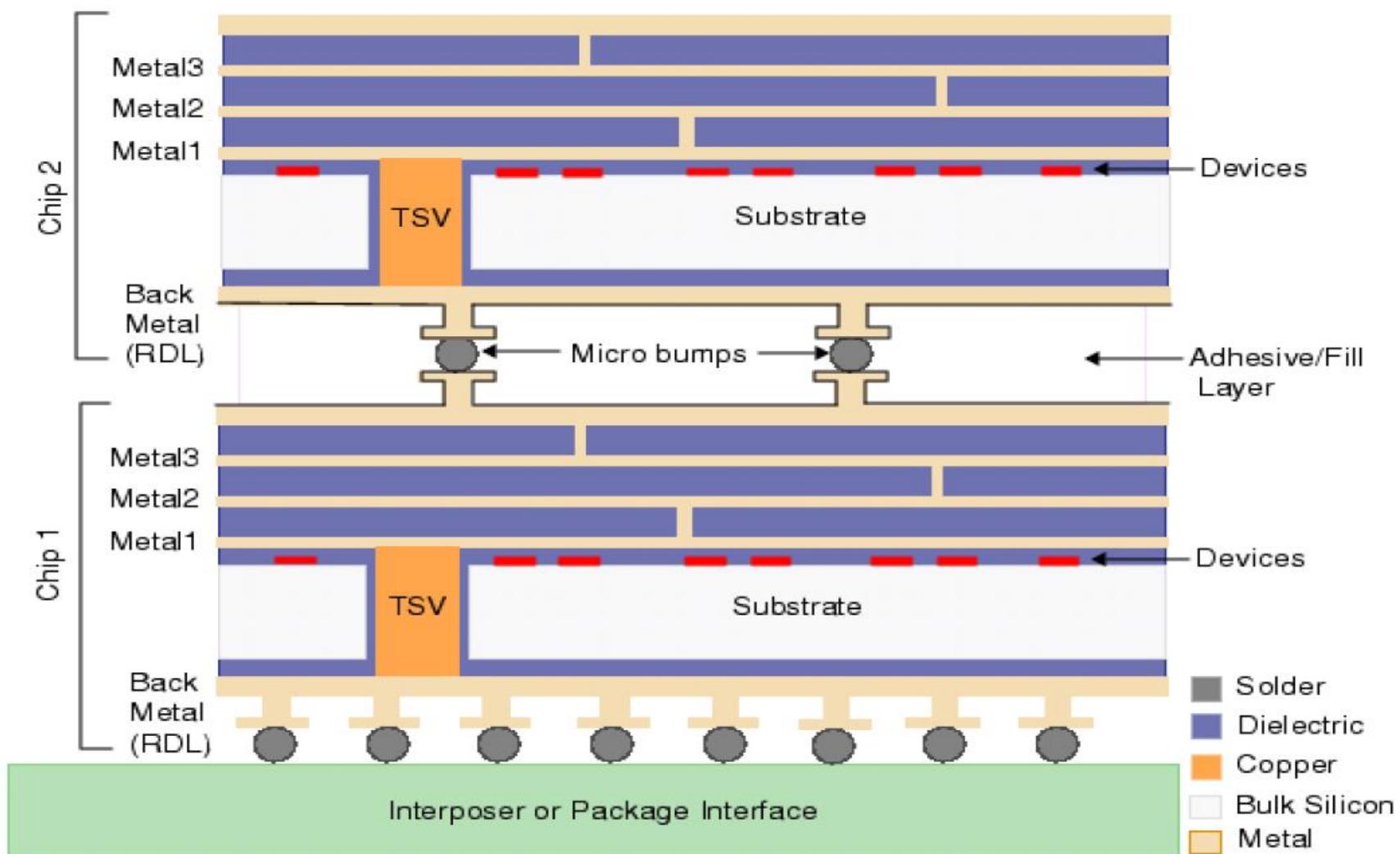


## • CIS工艺演进

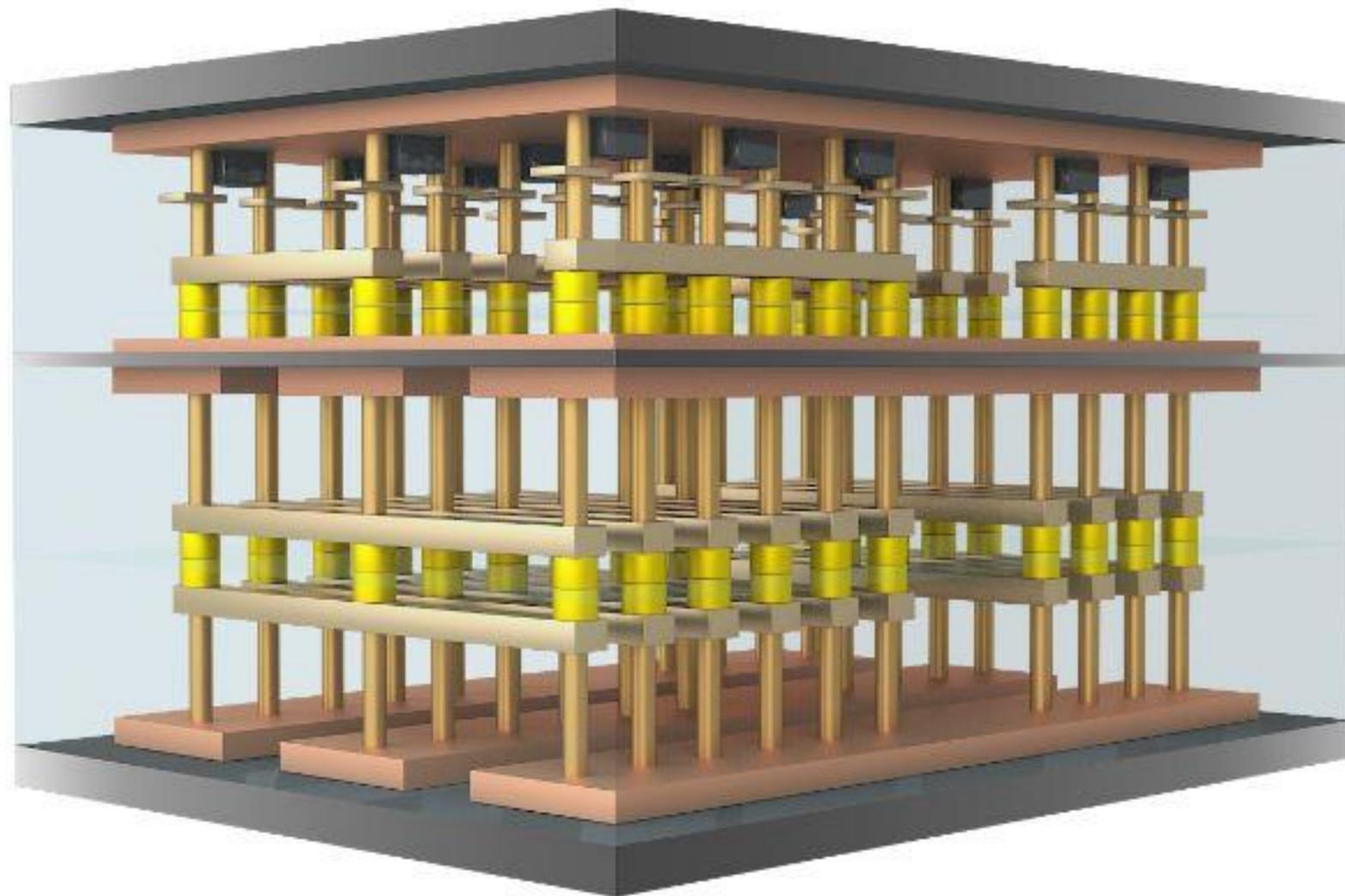
BSI opened the way to 3D semiconductor



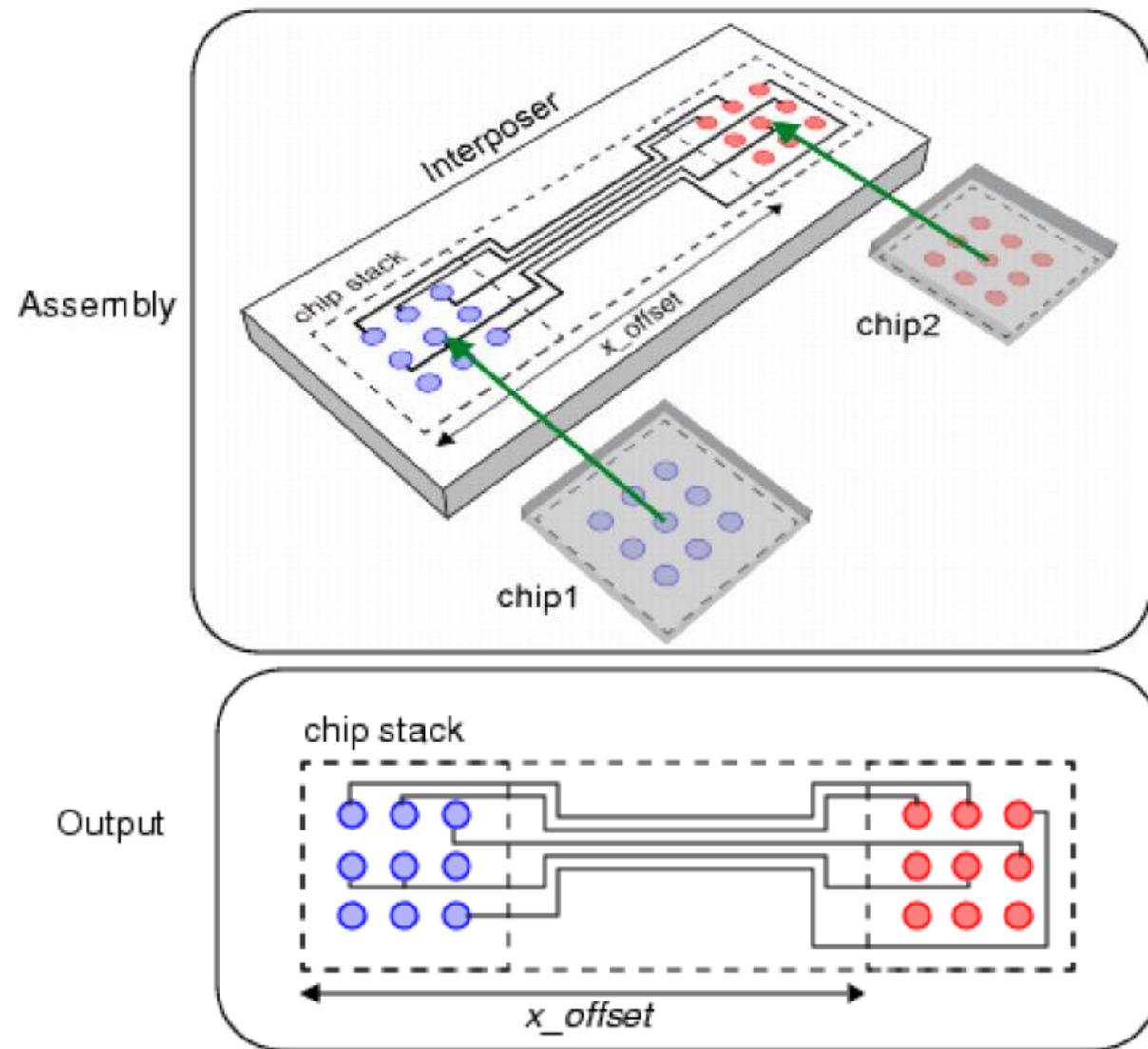
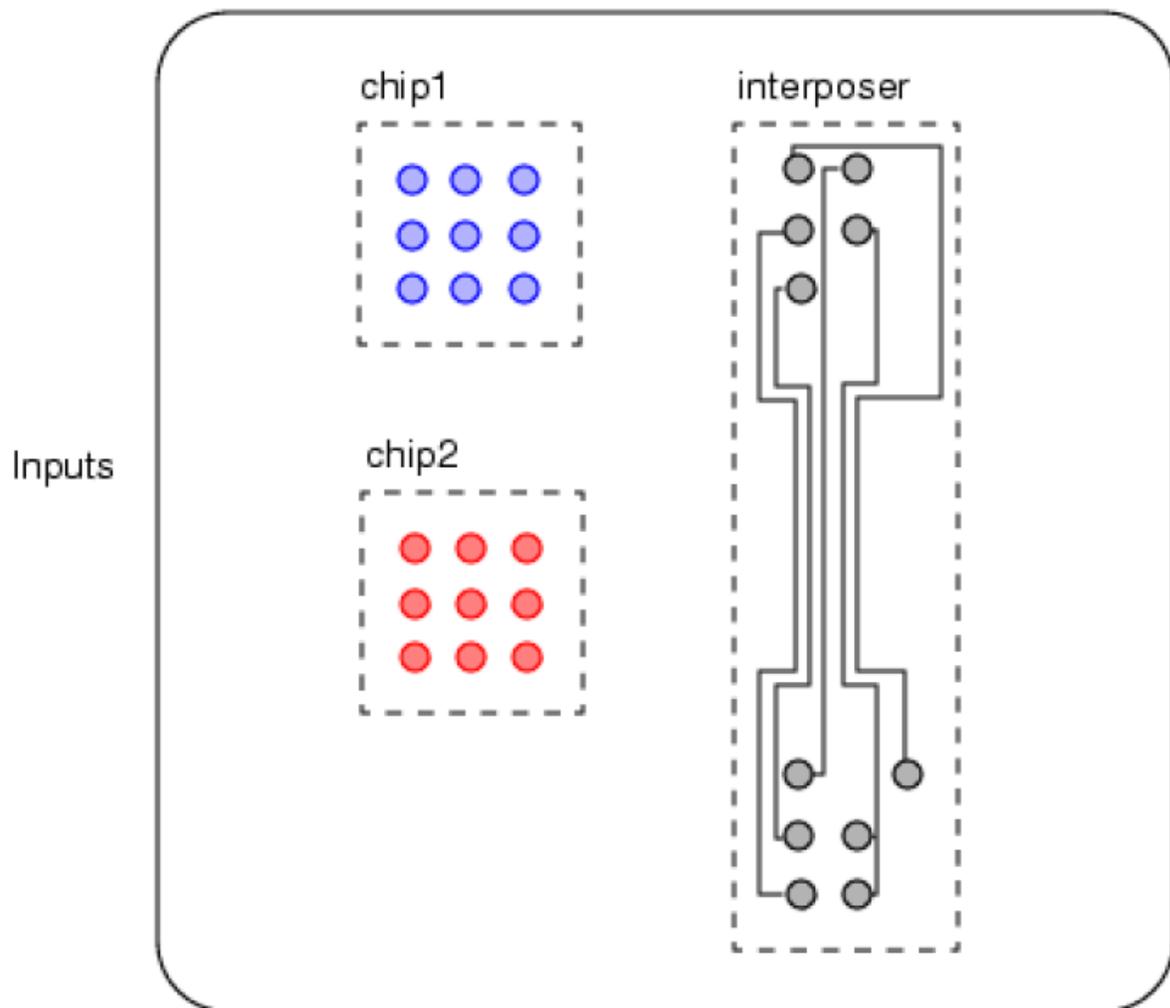
## • 晶圆级堆叠



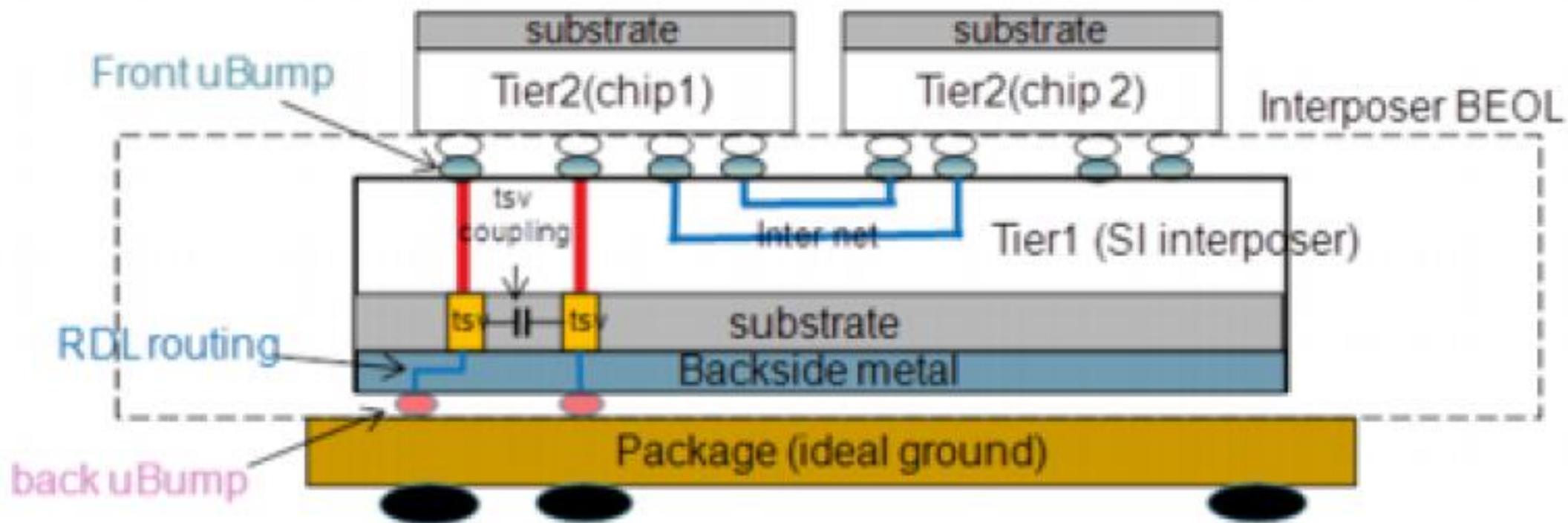
- 存储器晶圆级堆叠



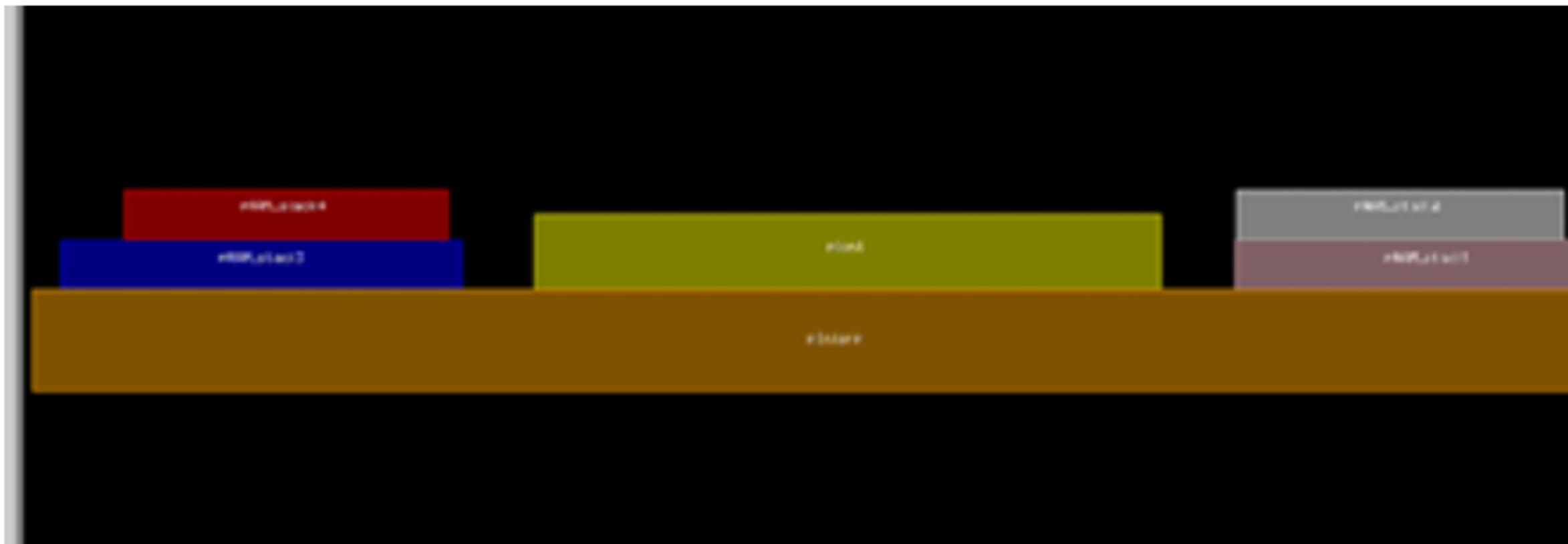
- Interposer的2.5D连接



- Interposer的连接



- 3D + 2.5D的组合连接



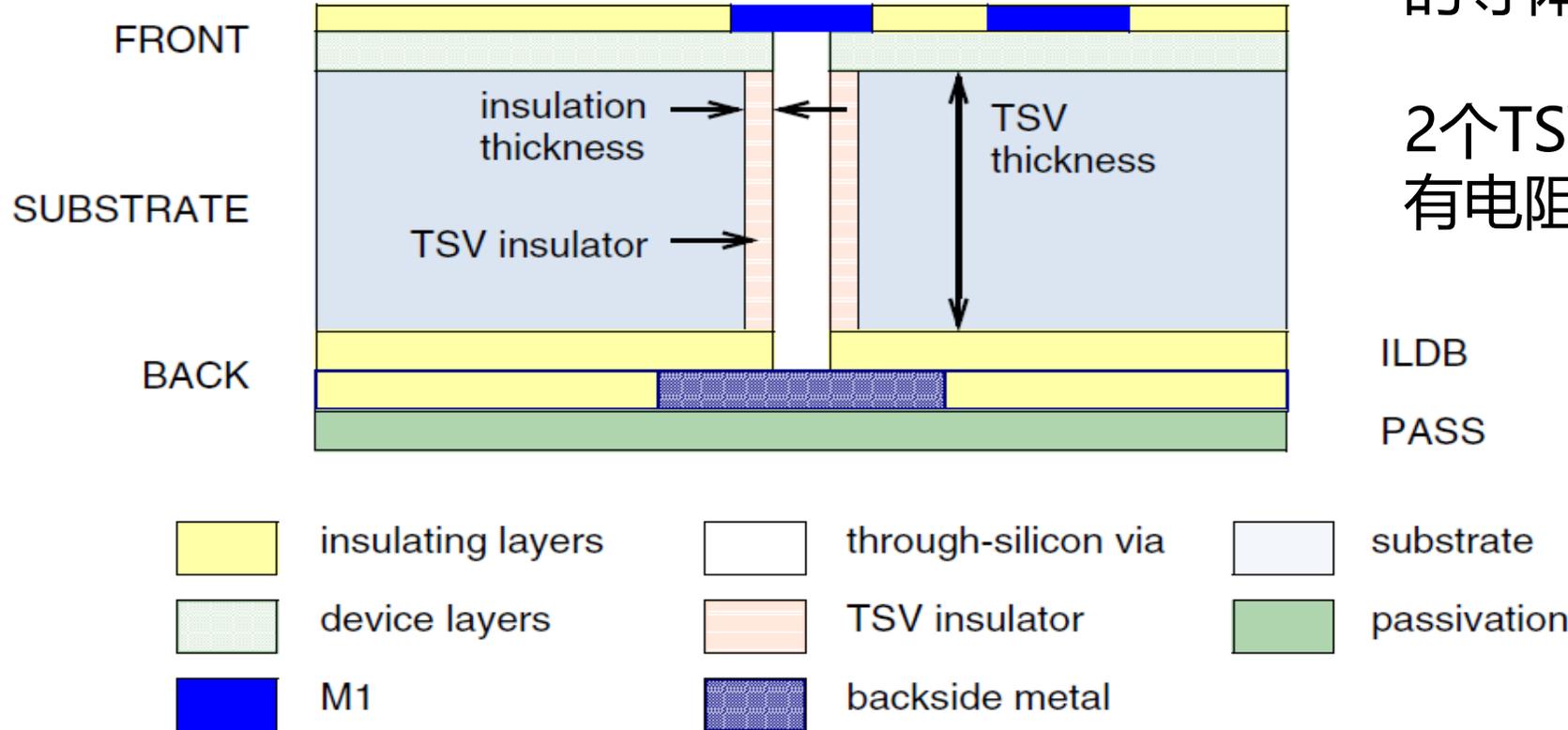
# 3D IC 提取难点

- 针对3D IC TSV结构的寄生参数提取难点

- 在衬底内部，介质是半导体，不是普通的绝缘层，TSV的寄生电容如何计算得到？
- 在衬底外部，TSV图形和其它导体的耦合电容如何计算得到？
- 如何把衬底外部和衬底内部分别提取的电容合并得到完整的网表？不同网表的节点如何对应？

# 3D IC 提取难点

- TSV结构的 Cross View



衬底内部是半导体区域，既不是绝缘介质，也不是导电的导体。

2个TSV之间有电容效应，也有电阻效应

ILDB

PASS

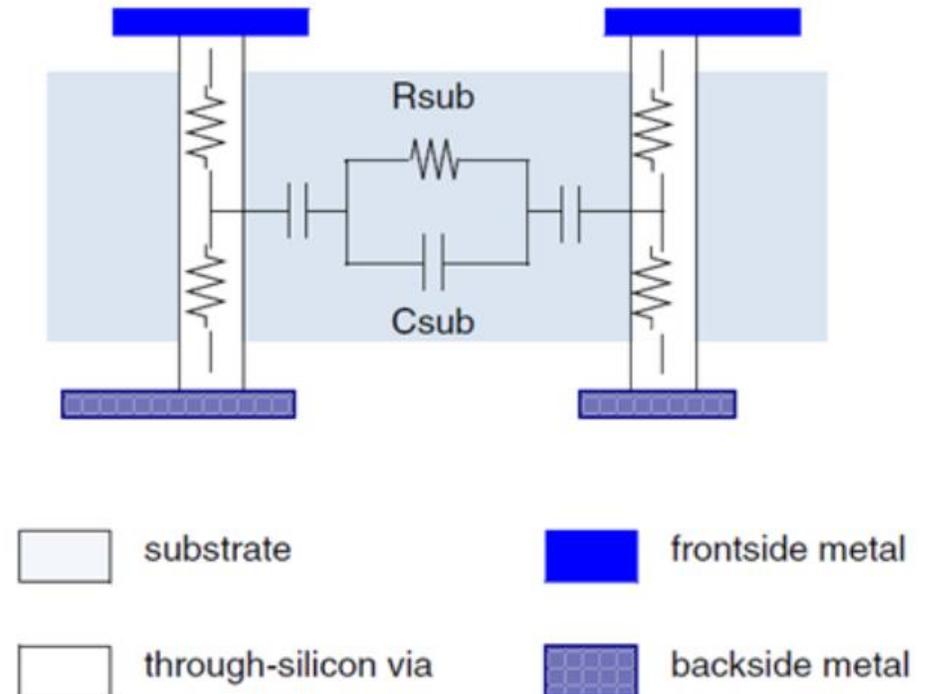
# 主流工具不足

- 业界主流工具的不足 (1)

- 针对衬底内部的TSV之间的耦合电容提取，需要用户准备好查表法的数据。但是如何得到这些查表得数据，没有给出具体的方法。查表法仅适合于均匀array结构。

## Syntax

```
TSV tsv_name {  
  FROM = layer1  
  TO = layer2  
  RHO = rho_value  
  AREA = area_value  
  THICKNESS = thickness_value  
  INSULATION_THICKNESS = ins_thickness_value  
  INSULATION_ER = er_value  
  [CRT1 = lin coeff] [CRT2 = quad coeff] [T0 = nominal_  
  CSUB_VS_SPACING { (s1,c1) (s2, c2) ... (sn,cn) }  
  RSUB_VS_SPACING { (s1,r1) (s2, r2) ... (sn,rn) }  
  LEFT_VS_FREQUENCY_AND_SPACING {  
  SPACINGS { s1 s2 s3 ... }  
  FREQUENCY { f1 f2 f3 ... }  
  VALUES { v(s1 f1) v(s2 f1) v(s3 f1) ...  
            v(s2 f1) v(s2 f2) v(s2 f3) ... } }  
}
```

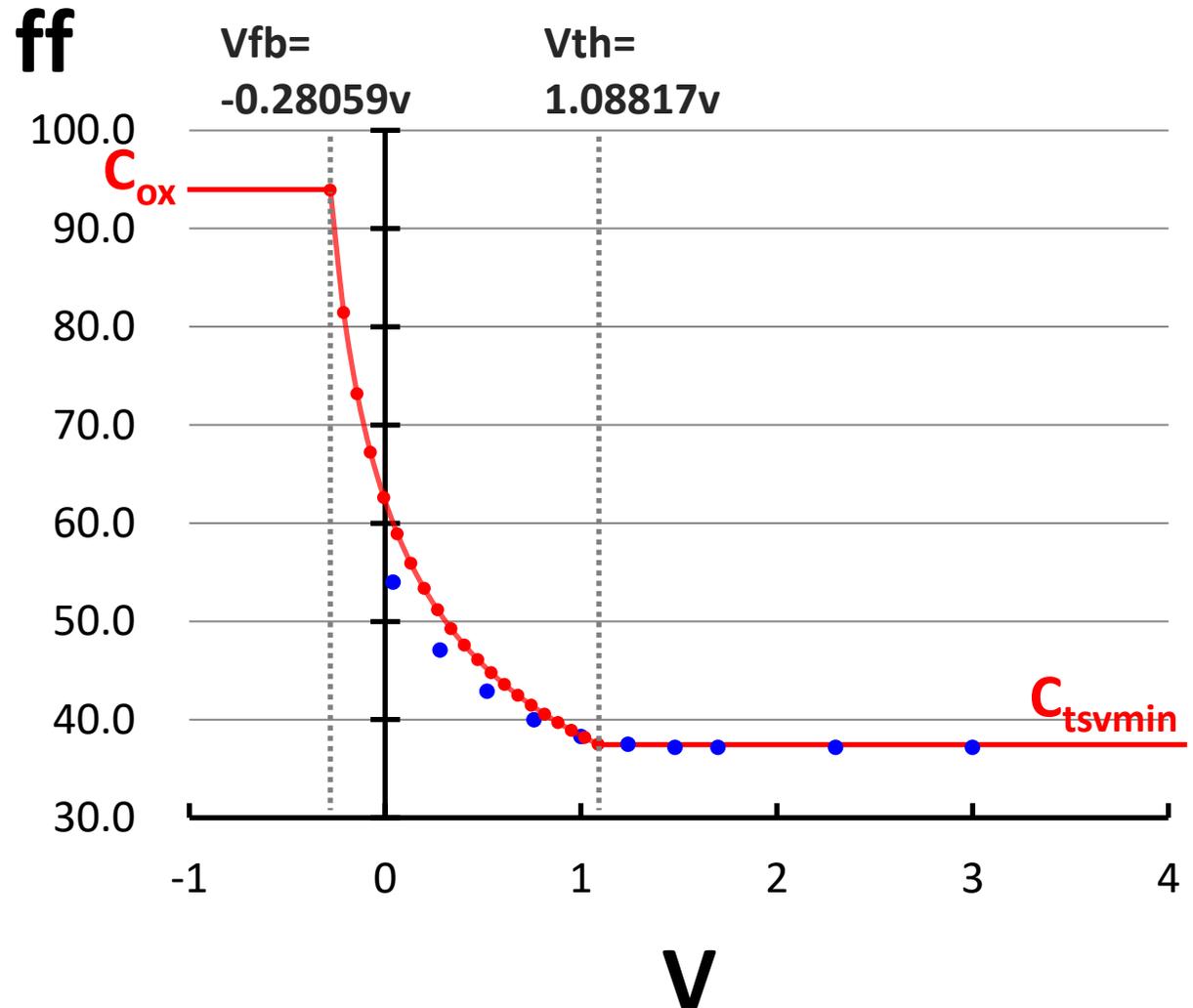


# 主流工具不足

- 业界主流工具的不足 (2)

- 针对TSV的自电容计算，没有考虑TSV的绝缘介质外的耗尽层电容影响。计算结果比实际电容会偏大。

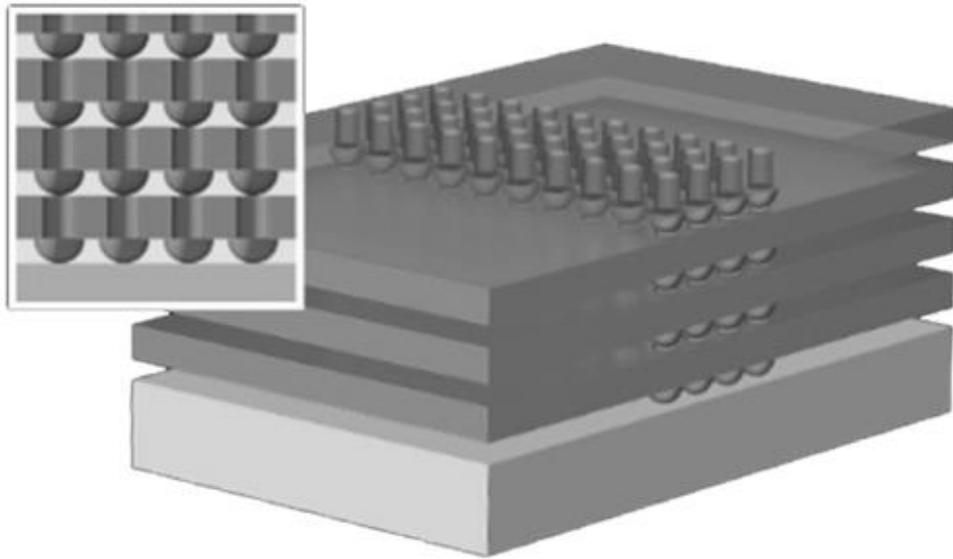
- 不考虑耗尽层为94ff，考虑后为37ff，差距较大



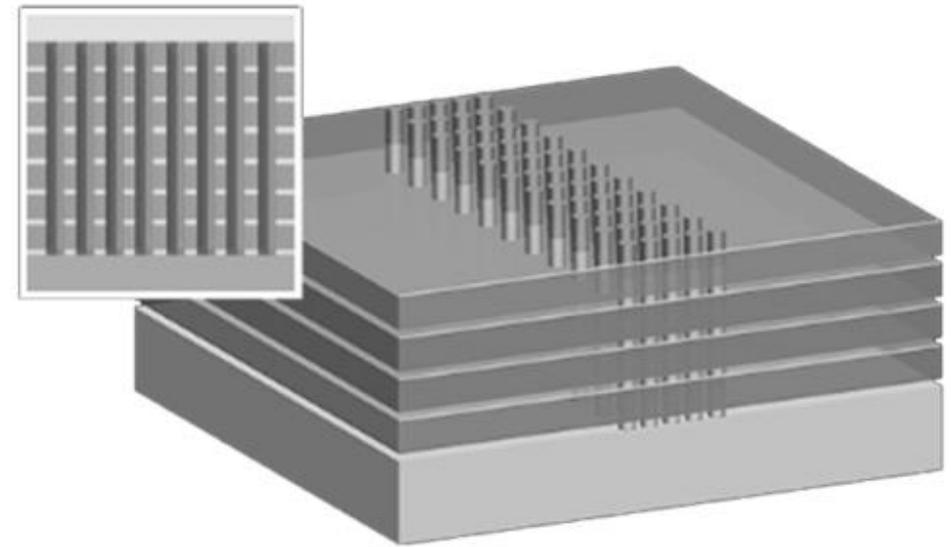
# 主流工具不足

- 业界主流工具的不足 (3)

- 针对多层晶圆堆叠，只能每个晶圆单独提取。但是没有自动化工具把多个晶圆提取出的网表自动合并



3D IC

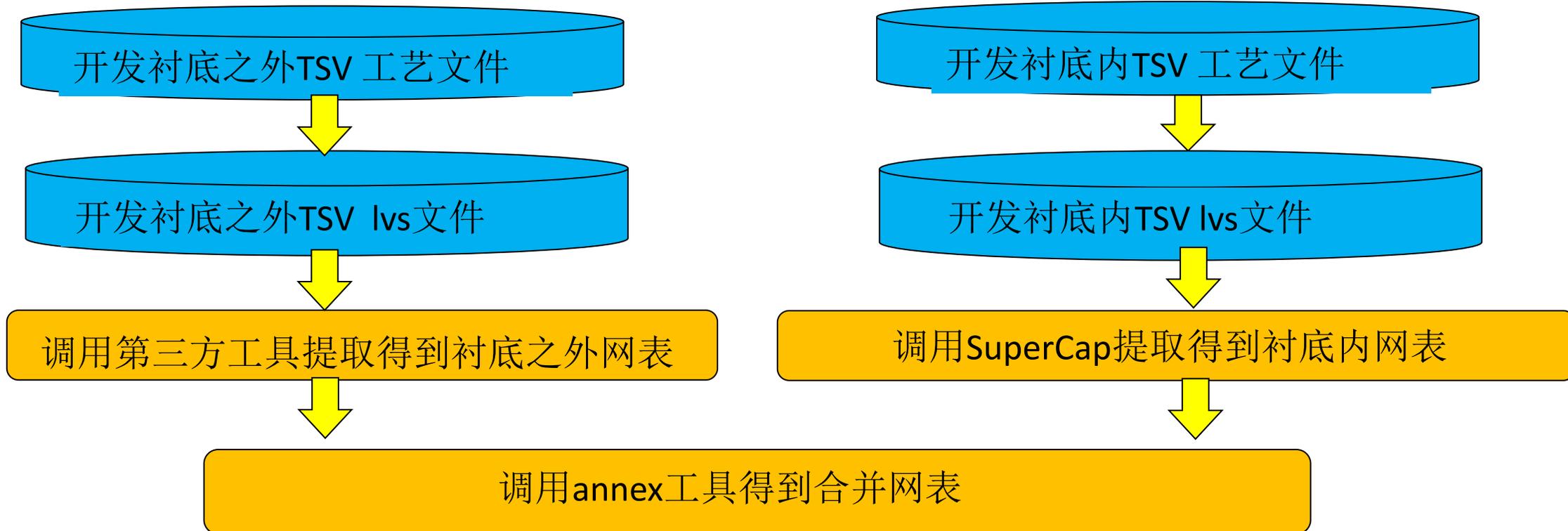


TSMC-SoIC

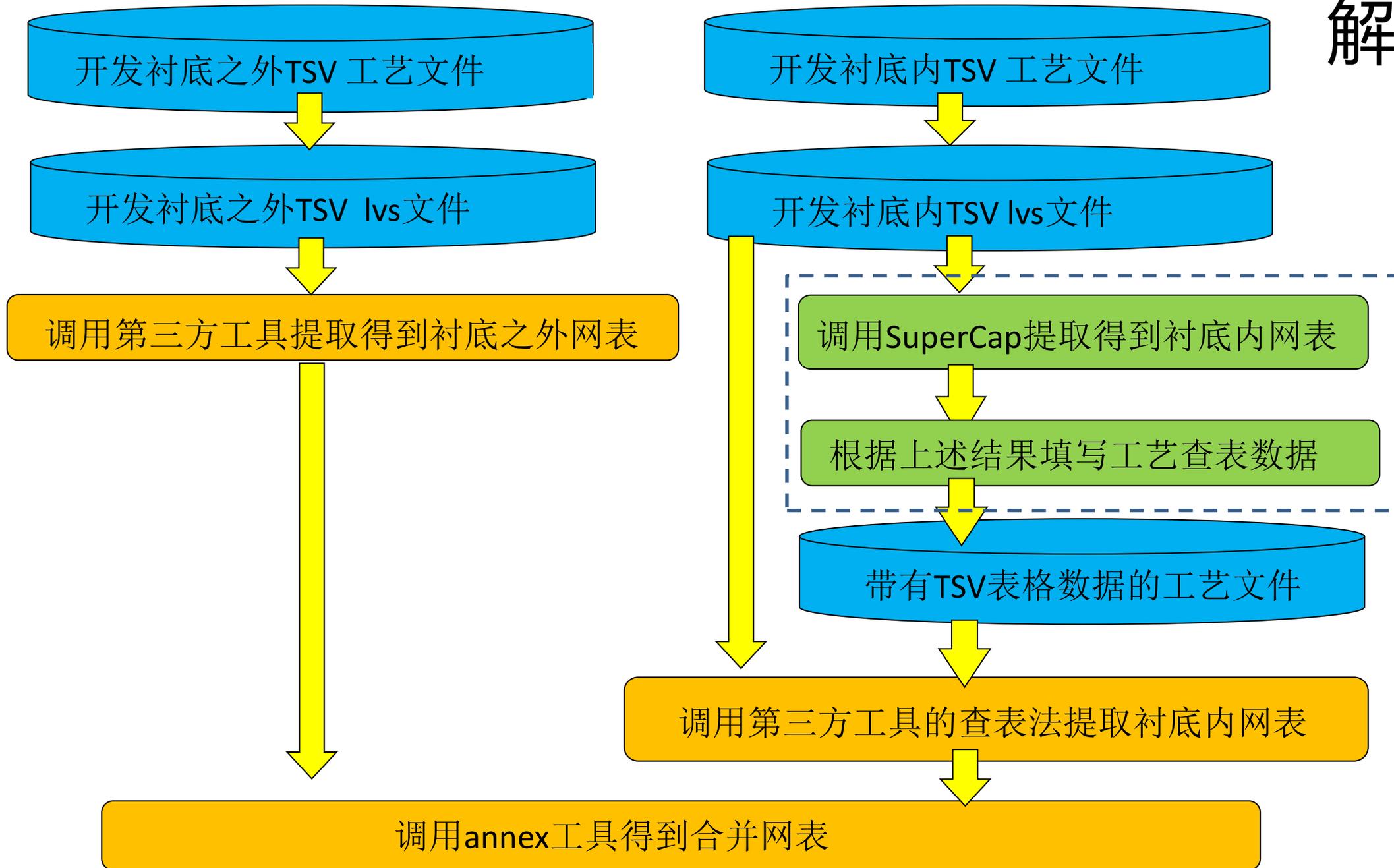
# 本文解决方案

- 针对3D IC TSV结构的寄生参数提取难点
  - 在衬底内部，TSV之间的耦合电容通过SuperCap三维提取工具得到准确结果。
  - TSV的自电容计算根据不同的电压会自动计算得到不同的电容值
  - 在衬底外部，通过在工艺文件和LVS文件中增加TSV layer的描述，调用主流工具提取单个芯片的寄生参数，包括了TSV图形对其它导体的耦合电容
  - 通过annex工具自动把多个堆叠芯片的提取网表进行合并

# 本文解决方案1



# 解决方案2



虚线部分内的绿色为Foundry内部调用工具。

最终用户需调用黄色的软件工具

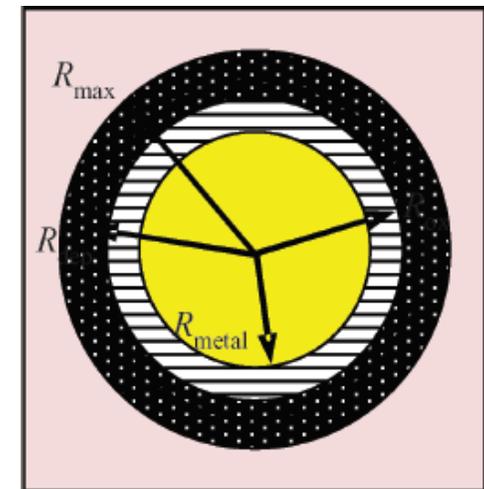
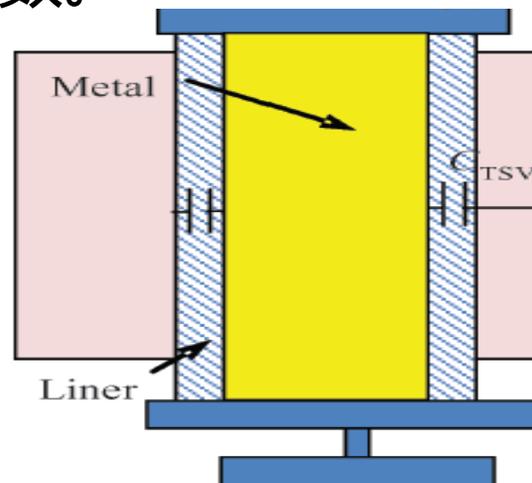
# 单个TSV提取原理

- 单个TSV的电容计算（没有考虑耗尽层）

- 同轴电缆解析公式：

$$C_{\text{ox}} = \frac{2\pi\epsilon_{\text{ox}}l_{\text{TSV}}}{\ln \frac{R_{\text{ox}}}{R_{\text{metal}}}}$$

- 其中 $R_{\text{metal}}$ 就是TSV的半径， $R_{\text{ox}}$ 就是从圆心到insulator外圈的半径。 $l_{\text{tsv}}$ 是TSV的厚度， $\epsilon_{\text{ox}}$ 是insulator的介电常数。



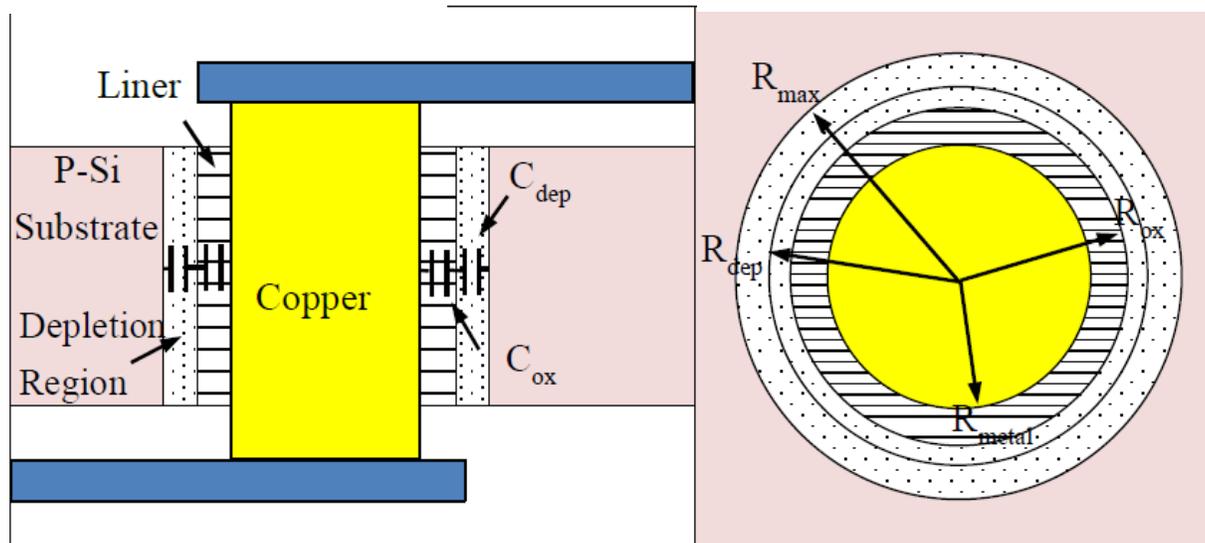
# 单个TSV提取原理

- 单个TSV的电容计算（考虑耗尽层）

- Insulator外部形成了一个比较薄的耗尽层， $C_{tsv}$ 的值更符合物理学原理，但是 $R_{dep}$ 计算比较复杂，目前主流工具一般都忽略了耗尽层的影响

$$C_{dep} = \frac{2\pi\epsilon_{si}l_{TSV}}{\ln\left(\frac{R_{dep}}{R_{ox}}\right)}$$

$$C_{TSV} = \frac{C_{ox}C_{dep}}{C_{ox}+C_{dep}}$$



# 单个TSV提取原理

- 单个TSV的耗尽层电容 $C_{dep}$ 计算所需输入数据

Na 掺杂浓度	客户给定 (测试 $2e15cm^{-3}$ )
Ni 本征硅载流子浓度	$1e10cm^{-3}$ (测试 $1.5e10cm^{-3}$ )
温度 (计算费米势)	300K
功函数 (材料参数相关)	<b>Cu</b> 4.53 - 5.10eV <b>Si</b> 4.60 - 4.85eV (测试 $\Phi_{MS}=0$ ) <small>(CRC Handbook of Chemistry and Physics version 2008 &amp; wikipedia)</small>
$Q_{ot}$ (表面态电荷, 材料制备相关)	客户给定 (测试 $0.75e10cm^{-2}$ )

# 单个TSV提取原理

测试用值:

$$N_a = 2e21m^{-3}$$

$$N_i = 1.5e16m^{-3}$$

$$T = 300K$$

$$\Phi_{MS} = 0v$$

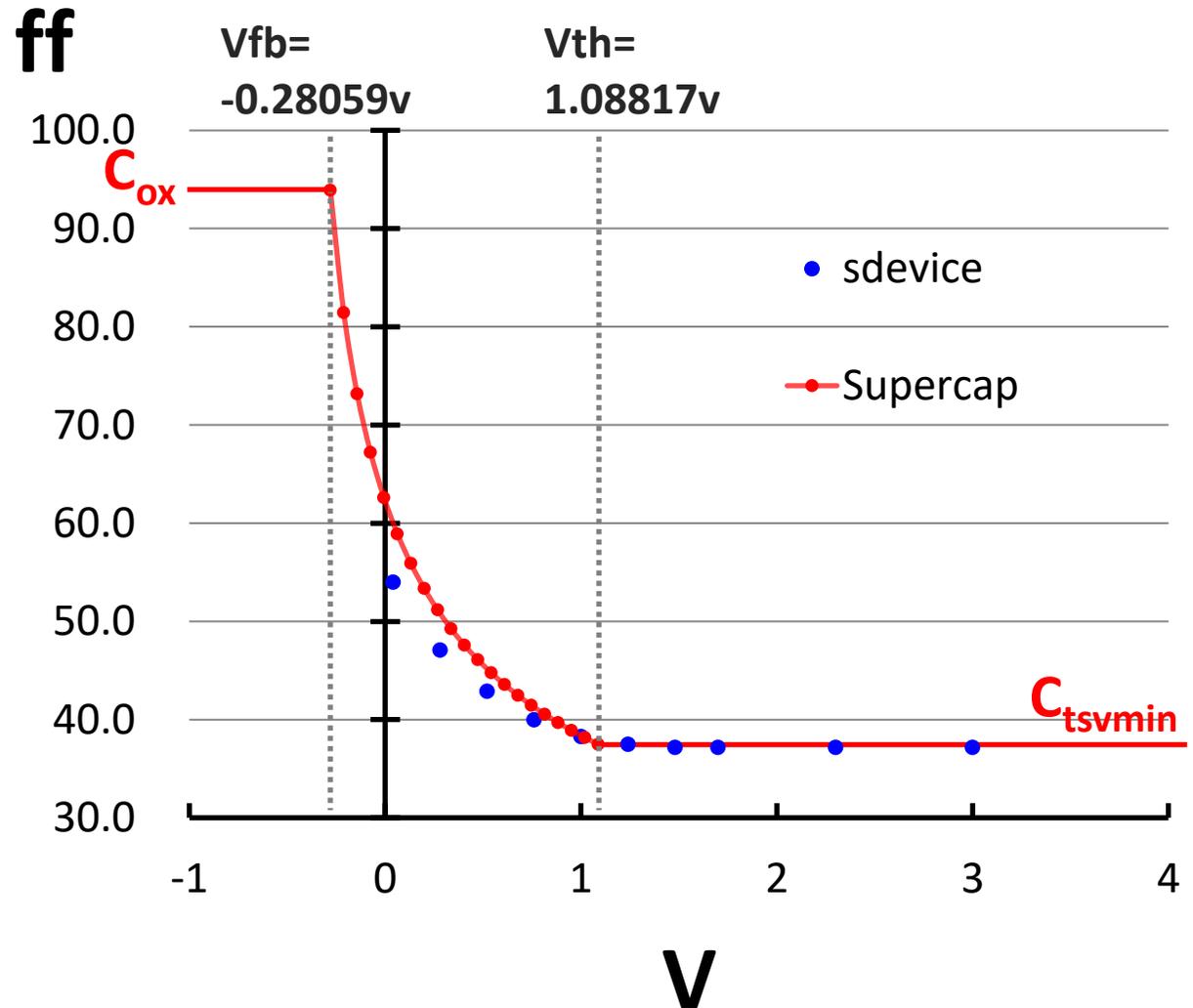
$$Q_{ot} = 0.5e15m^{-2}$$

Supercap结果:

$$C_{tsvmin} = 37.5063ff$$

$$V_{fb} = -0.28059v$$

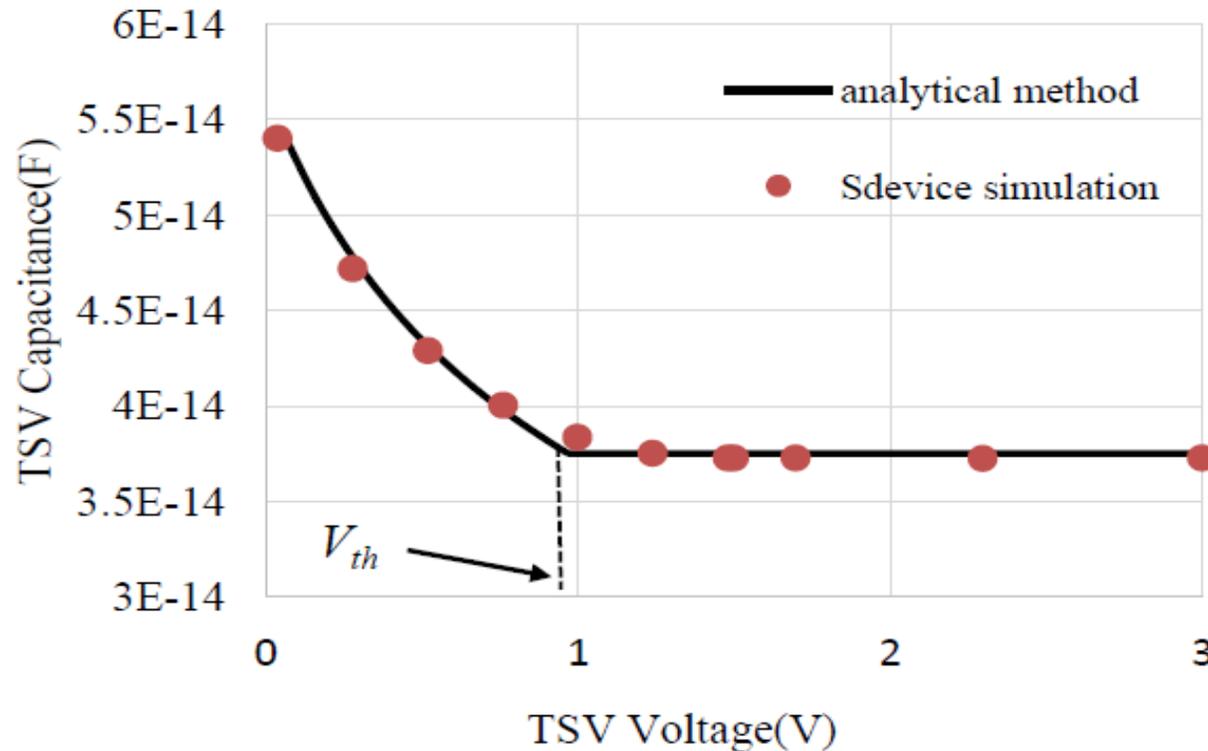
$$V_{th} = 1.08817$$



# 单个TSV提取原理

- 单个TSV的解析公式与TCAD仿真结果相比

- 误差最大为2%左右，说明解析公式是准确的



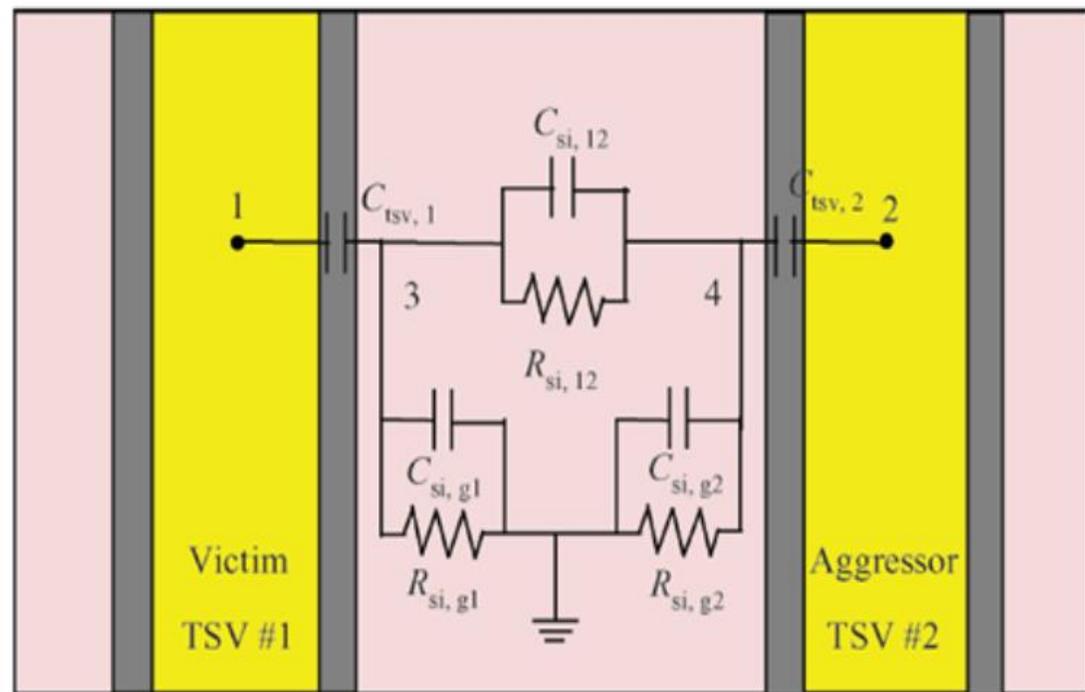
# 多个TSV提取原理

- 多个TSV衬底内耦合电容的3种计算模式
  - None模式, 不提取TSV耦合电容
  - RCsub模式, 把衬底内部的电阻、电容值用spice网表描述, 用于spice仿真工具
  - CEFF模式, 把衬底内部的电阻电容值等价于与频率相关的等效电容模型, 用于静态时序分析

# 多个TSV提取原理

- 2个TSV的RCsub电路模型

- 黄色为TSV（导体），粉红色为衬底（半导体），灰色为Insulator（绝缘介质）
- 图中共有5个节点。其中1, 2号节点是TSV本身的节点，3, 4号节点是绝缘介质与衬底交界面的节点，还有一个接地的节点。



# 多个TSV提取原理

- 2个TSV的电阻与电容的反比模型

- 先通过SuperCap把电容计算得到，电阻：

$$R_{si} = \frac{\epsilon_{si}}{\sigma_{si} C_{si}},$$

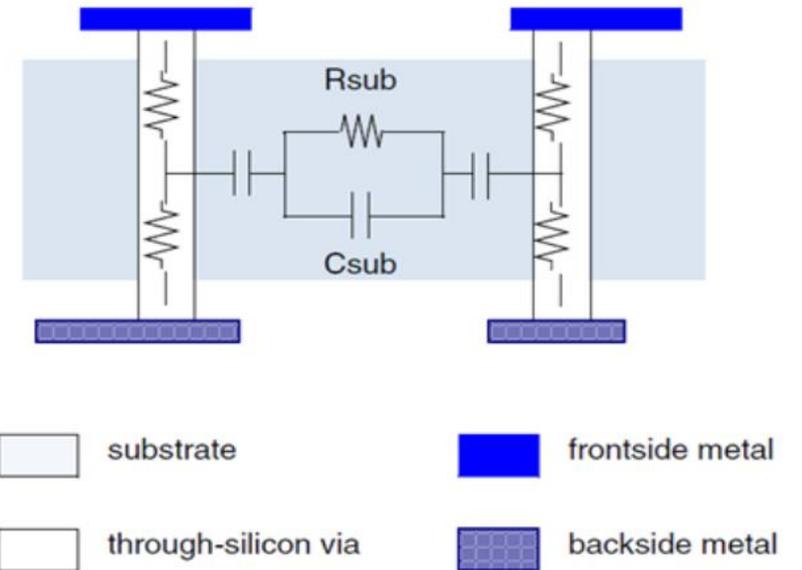
- $\epsilon_{si}$ 是衬底硅的介电常数 11.7,与掺杂浓度无关

- $\sigma_{si}$  是衬底硅的电导率 = 14.3 ，与掺杂浓度有关，用户填写

# 多个TSV提取原理

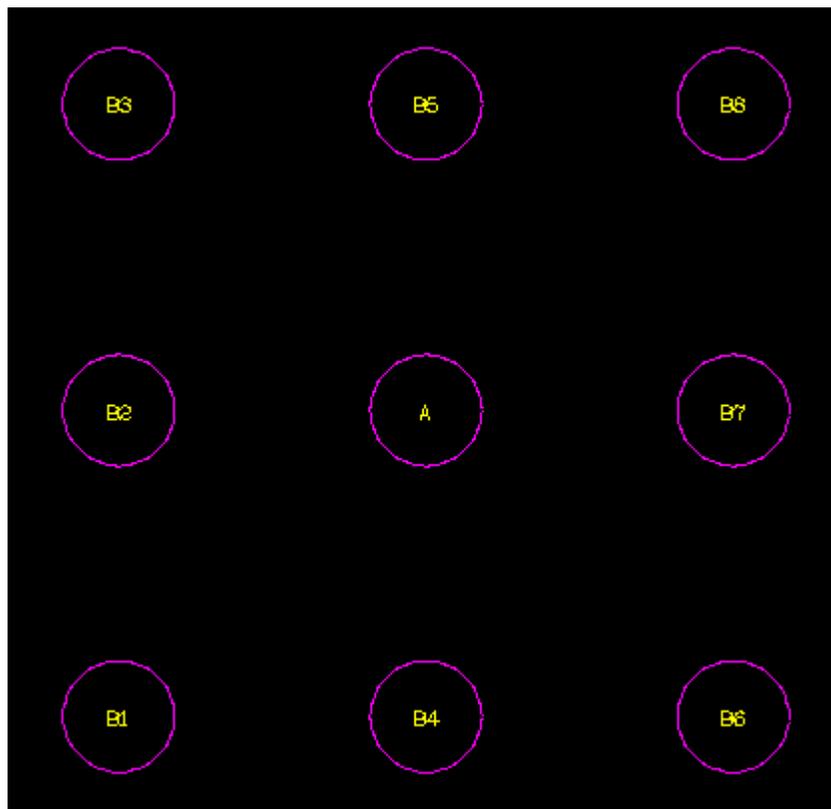
- 2个TSV的RCsub表格参数计算过程
  - 首先通过SuperCap工具计算出Csub, 然后通过反比模型计算出Rsub。
  - 构造不同间距的TSV pattern, 得到如下表格。

```
CSUB_VS_SPACING { (s1, c1) (s2, c2) ... (sn, cn) }  
RSUB_VS_SPACING { (s1, r1) (s2, r2) ... (sn, rn) }
```



# 多个TSV提取原理

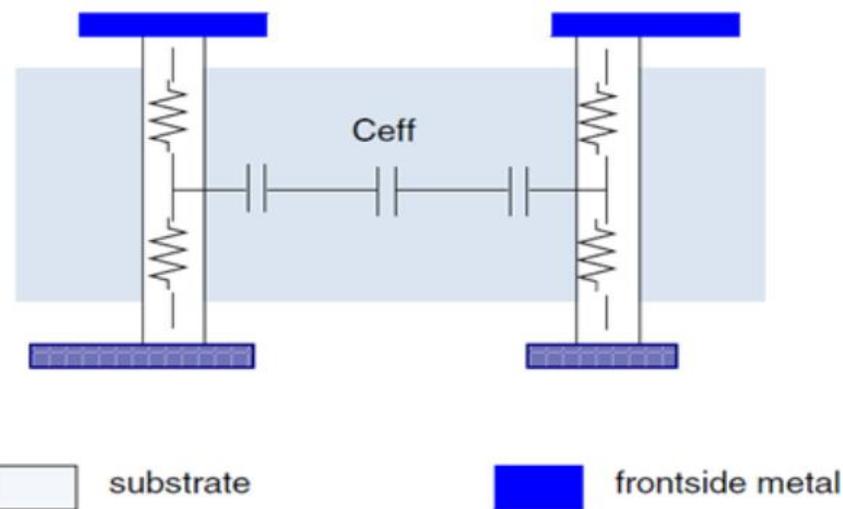
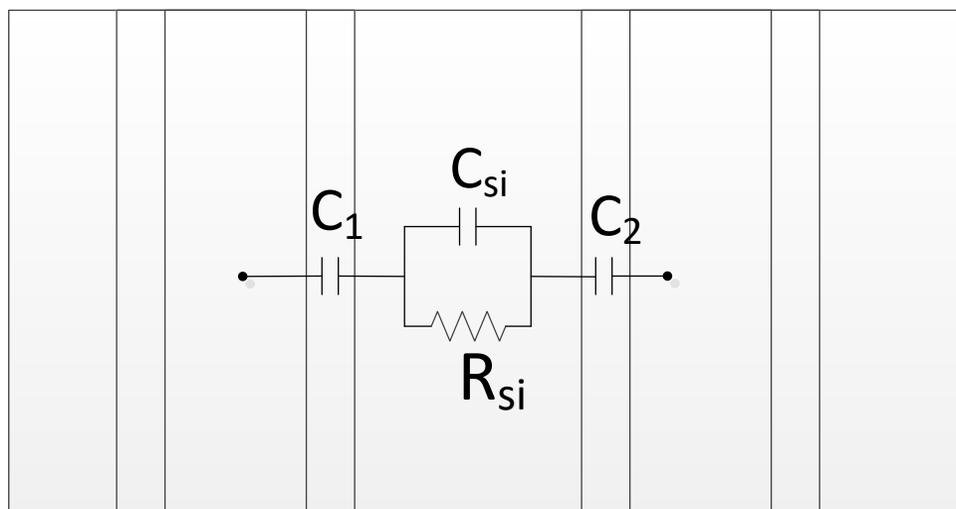
- 自动生成TSV Pattern



# 多个TSV提取原理

- 2个TSV的CEFF模型

- 把Rcsub模型的RC电路进行等效处理，得到与频率相关的等效电容Ceff
- 左下图为RCsub模型，右下图为CEFF模型



```
CEFF_VS_FREQUENCY_AND_SPACING {  
  SPACINGS { s1 s2 s3 ... }  
  FREQUENCY { f1 f2 f3 ... }  
  VALUES { v(s1 f1) v(s2 f1) v(s3 f1) ...  
            v(s2 f1) v(s2 f2) v(s2 f3) ... } }
```

# 多个TSV提取原理

- 2个TSV的CEFF模型

- $w$ 是频率

- $C_0 = C_1 + C_2$

- $$C(w) = \frac{[\sigma^2 C_0 C_{si}^2 + w^2 \varepsilon^2 C_0 C_{si} (C_0 + C_{si})]}{w^2 \varepsilon^2 (C_0 + C_{si})^2 + \sigma^2 C_{si}^2}$$

- 当 $w$ 比较小时,  $C(w)$ 等于 $C_0$ ,

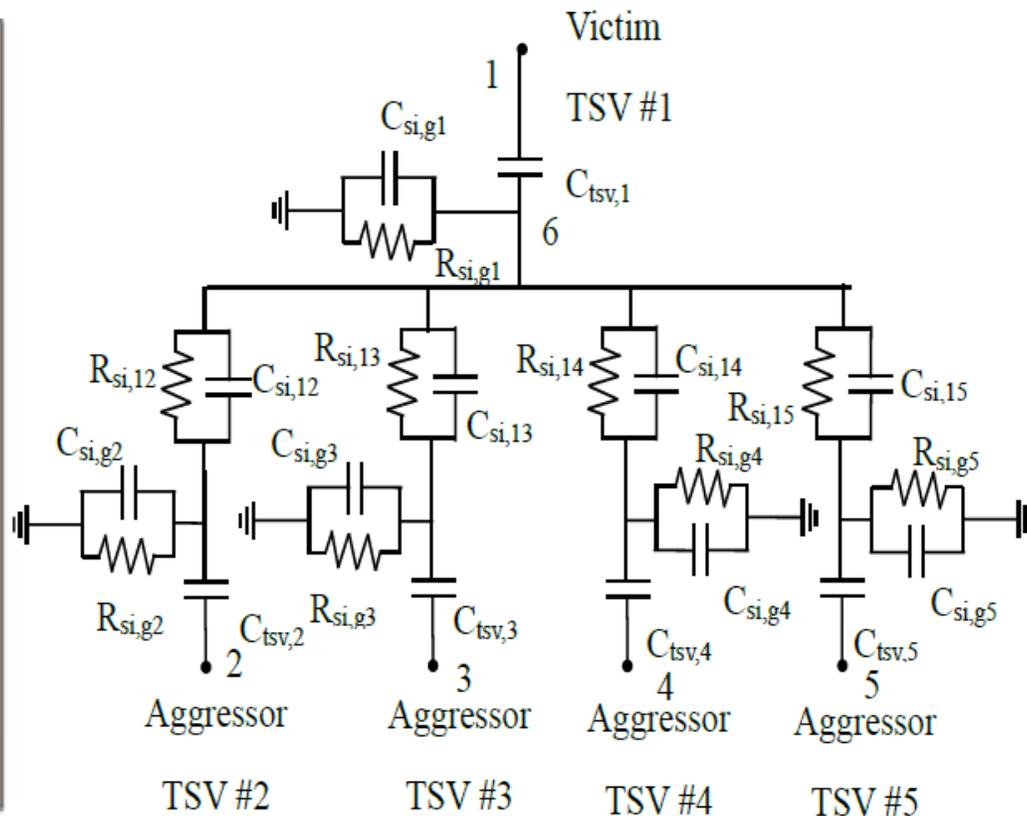
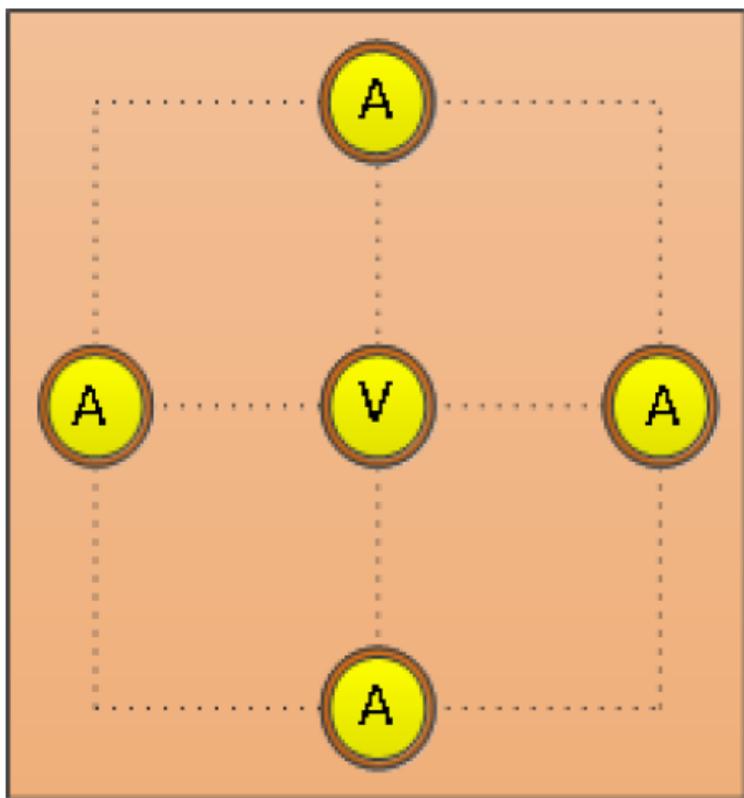
- 当 $w$ 比较大时,  $C(w) = \frac{[C_0 C_{si}]}{(C_0 + C_{si})^2}$ ,

当 $C_0$ 比 $C_{si}$ 大很多时,  $C(w)$ 基本可以等于 $C_{si}$

# 多个TSV提取原理

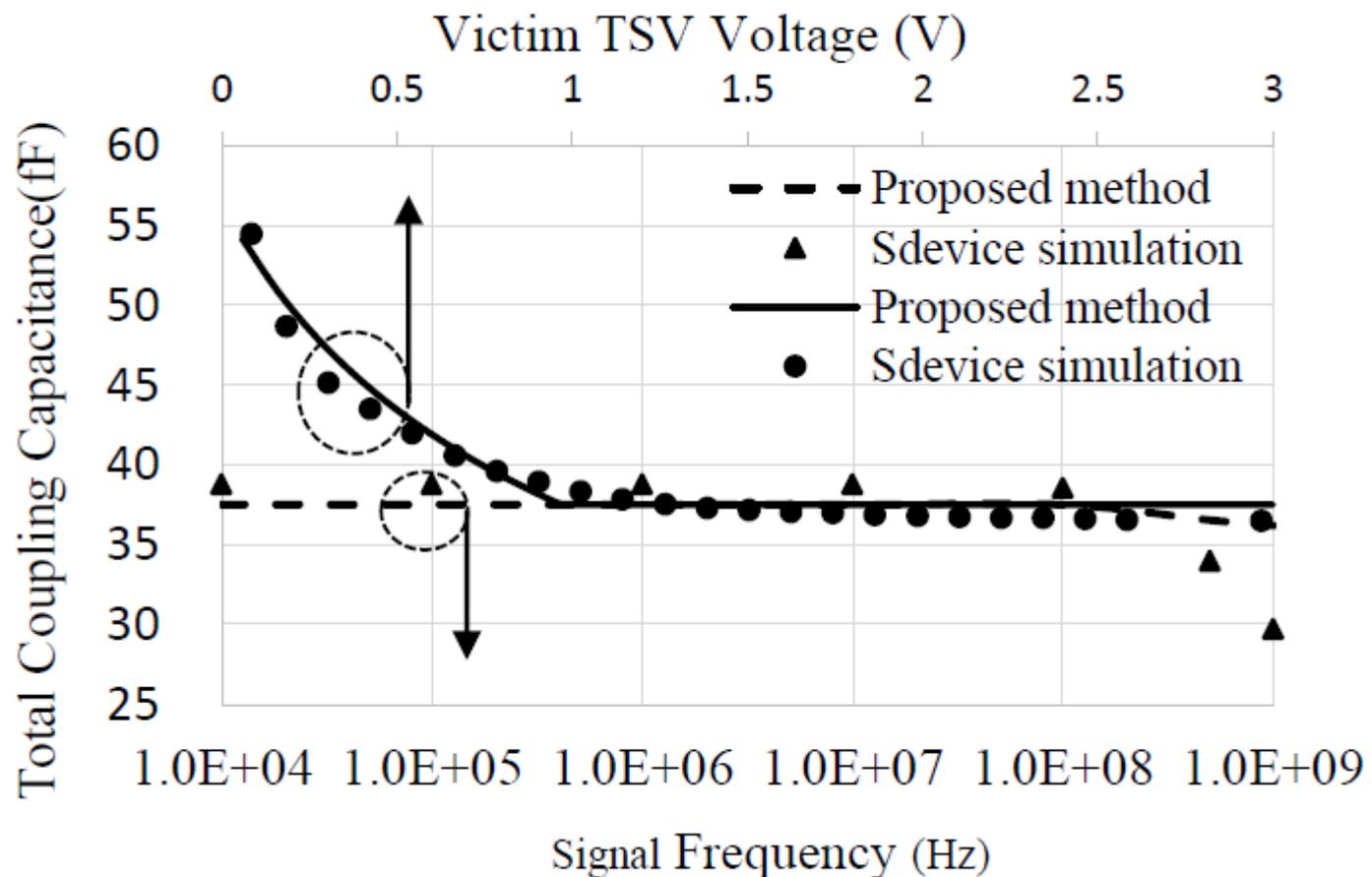
- 多个TSV的CEFF模型

- 5个TSV，中间的Victim导体的Ceff模型如下



# 多个TSV提取原理

- 2个TSV的Ceff结果精度对比

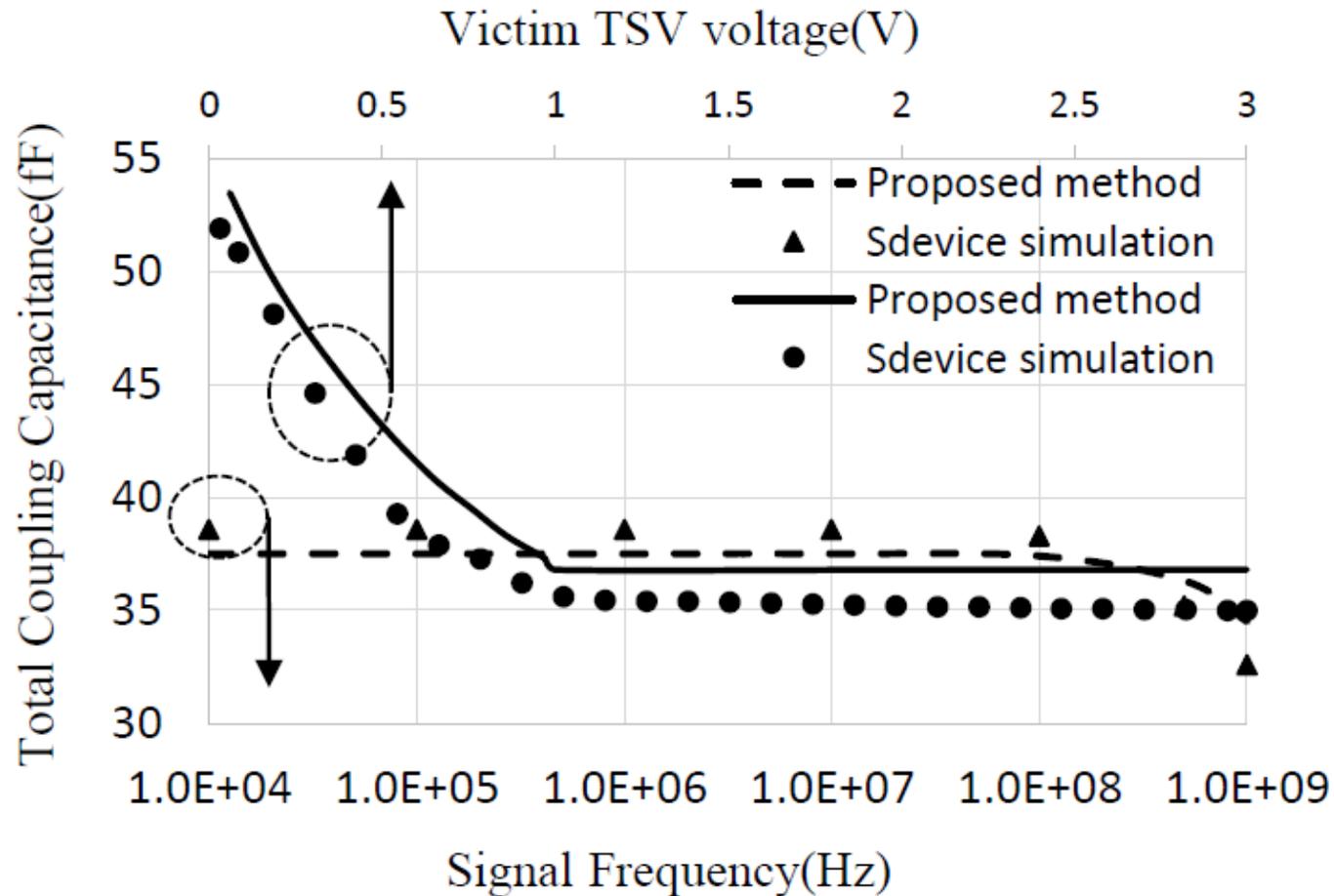


虚线数据：当TSV的主导体电压为1V时，随着频率的变化，Ceff的SuperCap计算结果

三角数据：当TSV的主导体电压为1V时，随着频率的变化，Ceff的TCAD计算结果

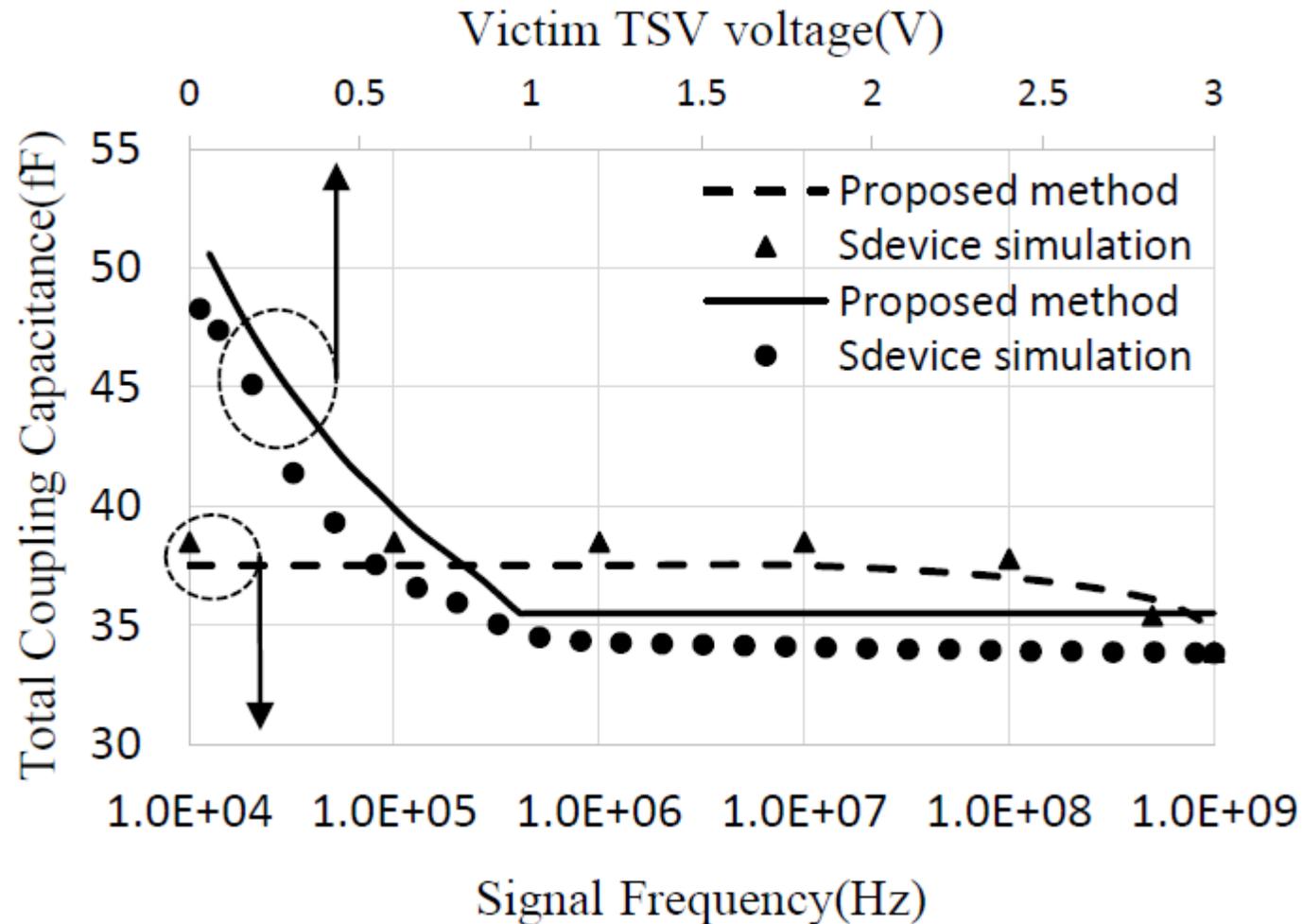
# 多个TSV提取原理

- 5个TSV的Ceff结果精度对比



# 多个TSV提取原理

- 9个TSV的Ceff结果精度对比



# 多个TSV提取原理

- 多个TSV的Ceff计算效率对比
  - 9个TSV时, SuperCap计算速度是TCAD工具的29.5倍

Case	TCAD工具		本文策略	
	Grid	Time(s)	Time(s)	加速比
2-TSV	3476	52.5	6.8	7.7
5-TSV	8464	147.8	7.1	20.8
9-TSV	14562	236.3	8.0	29.5

# 衬底之外TSV图形提取

- 位于衬底外TSV，需要考虑其对其它导体的电容
  - 在工艺文件描述中，把TSV看作一个特殊的Via
  - 在LVS文件中，增加TSV的连接语句
  - 在map file中，增加TSV的映射

# 衬底之外TSV图形提取

- 位于衬底外TSV，需要考虑其对其它导体的电容

```
VIA TSV {  
  FROM = SUBSTRATE  
  TO = RDL  
  AREA = 49  
  CRT1 = 0.0050  
  CRT2 = 0.0  
}
```

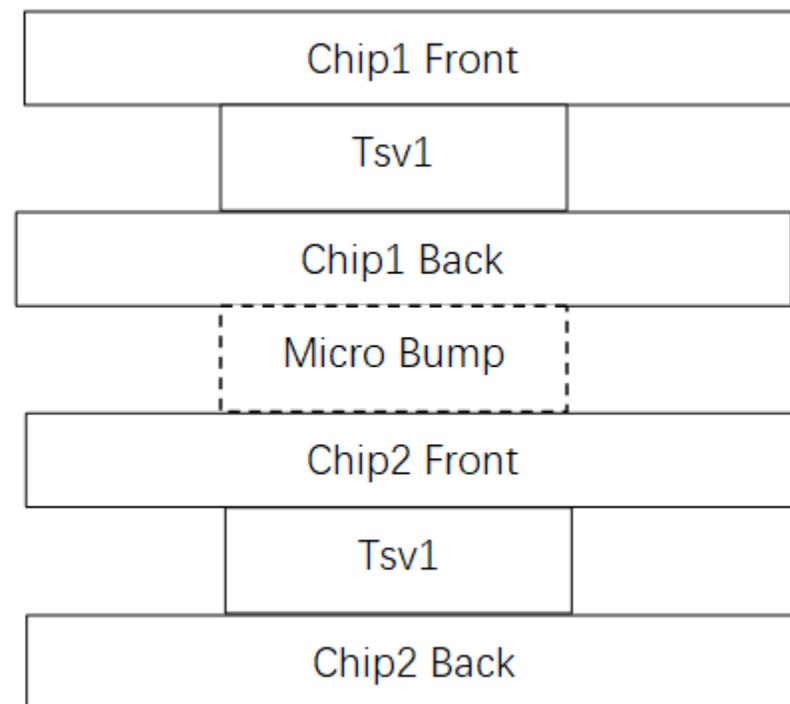
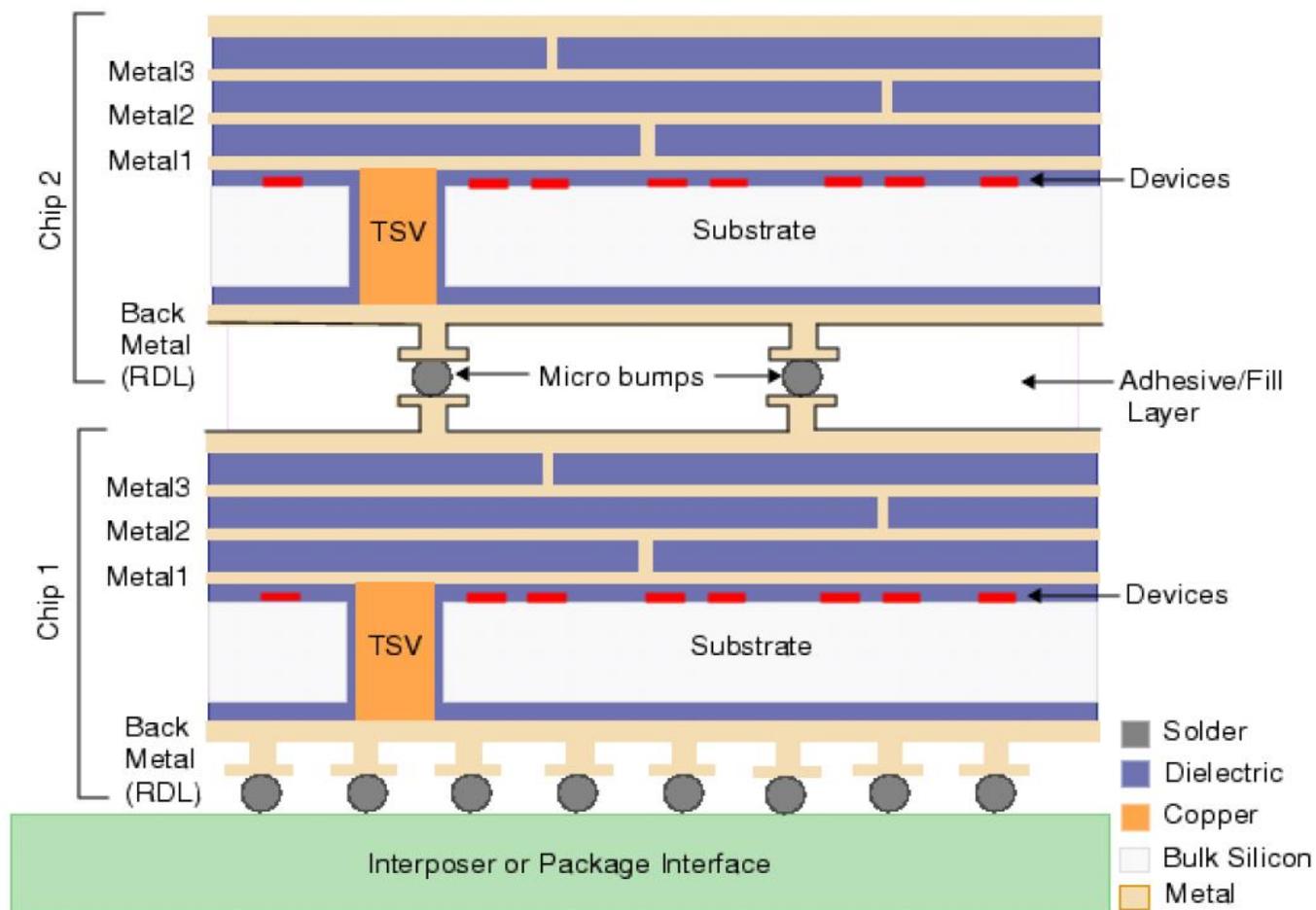
还需要描述micro bump与RDL的连接的信息，如下：

```
VIA topuBump {  
  FROM = RDL  
  TO = virtual_top_layer  
  AREA = 49  
  CRT1 = 0.0050  
  CRT2 = 0.0  
}
```

其中virtual\_top\_layer是一个很薄的虚拟金属层，位于正面RDL的上方。

# 网表合并

## • 电容典型网表合并示意图



# 网表合并

- 网表合并需求

- 多个芯片堆叠，每个芯片的网表单独提取，需要把上下相邻芯片的节点连接
- 位于衬底内部的TSV网表也是单独提取的，它需要与上下芯片的节点连接

- 网表合并3种模式

- 基于线网名字相同的节点合并
- 基于transistor level提取CCI数据的坐标一致的节点合并
- 基于cell level提取LEF/DEF数据的坐标一致的节点合并

# 网表合并

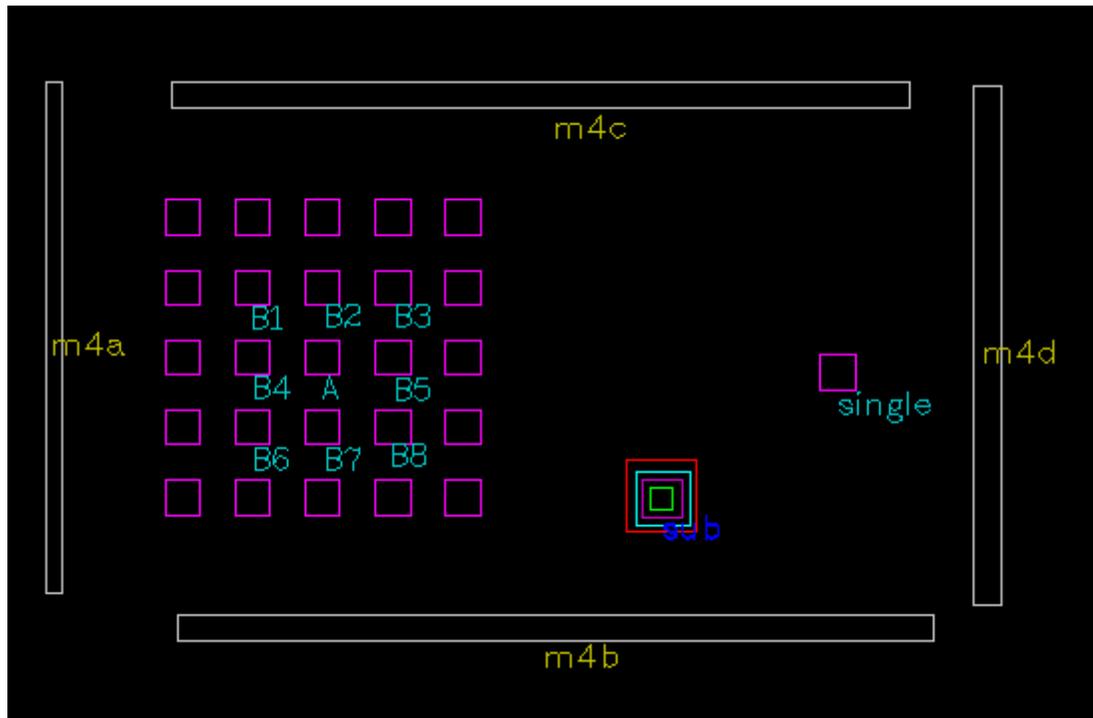
- 基于线网名字的合并

- 条件：TSV 用于对外连接的pin都标识有线网名，并且与其它芯片对外连接的线网名字是相同的。
- 当TSV个数很多时，要求用户针对每个TSV都要标识线网名，用户工作量大。
- 合并实现较为简单，不依赖于gds等版图信息，仅依赖于网表。

# 网表合并

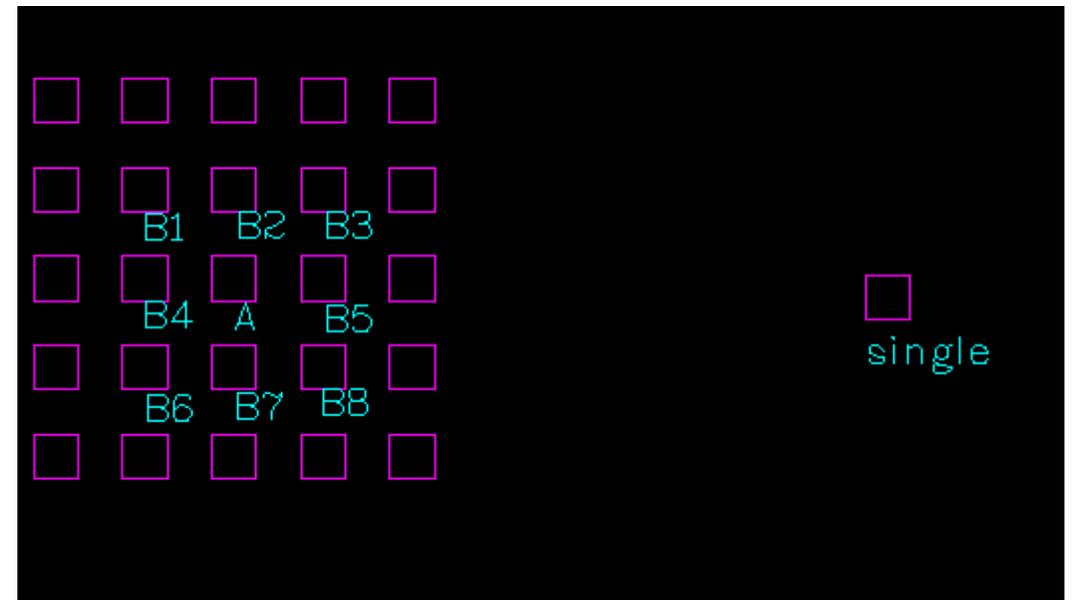
- Circuit1(full chip)

26个tsv, 只有10个有线网名



- Circuit2(TSV)

26个tsv, 只有10个有线网名



# 网表合并

- 基于线网名的网表合并 配置文件

Top\_to\_Bottom: Circuit1 Circuit2 Circuit3

+-----+

| Circuit1 |

Circuit1 TopCell: top1

+-----+

Circuit1 netlist: tsv\_fullchip/RC.spf

| Circuit2 |

Circuit2 TopCell: top\_tsv

+-----+

Circuit2 netlist: tsv\_only/RC.spf

| Circuit3 |

Circuit3 TopCell: top1

+-----+

Circuit3 netlist: tsv\_fullchip/RC.spf

# 网表合并

- 基于线网名的网表合并结果

注意26个图形上下重叠，只连接了10个图形，其余16个未连接

Circuit1-Circuit2 connect pin

B6 - B6

B4 - B4

B1 - B1

B7 - B7

A - A

B2 - B2

B8 - B8

B5 - B5

B3 - B3

single - single

Circuit2-Circuit3 connect pin

B6 - B6

B4 - B4

B1 - B1

B7 - B7

A - A

B2 - B2

B8 - B8

B5 - B5

B3 - B3

single - single

# 网表合并

- 基于transistor level的版图坐标一致的网表合并
  - 条件： TSV 用于对外连接的pin没有标识线网名，需要通过坐标对比的方式来找到与其它网表连接在一起的名字。
  - 要求寄生参数提取工具采用CCI的数据进行运行，工具通过读取CCI的坐标数据来判断哪些节点需要连接。
  - 合并实现较为复杂，需要在配置文件中定义自顶向下的网表顺序，还需要定义每一个芯片的对外连接的最上层，最下层layer，以便工具判断连接。

# 网表合并

- 基于transistor level的版图坐标一致的网表合并 配置文件

Top\_to\_Bottom: Circuit1 Circuit2 Circuit3

Circuit1 TopCell: aa

Circuit1 netlist: /home/tsv\_fullchip/RC.spf

Circuit1 CCI Path: /home/tsv\_fullchip

Circuit1 CCI query: query\_cmd

Circuit1 CCI top layer: top\_layer

Circuit1 CCI bottom layer: bot\_layer

Circuit1 CCI top device: tsv\_top

Circuit1 CCI bottom device: tsv\_bot

Circuit2 TopCell: bb

Circuit2 netlist: /home/tsv\_only/RC.spf

Circuit2 CCI Path: /home/tsv\_only

Circuit2 CCI query: query\_cmd

Circuit2 CCI top layer: tsv\_layer

Circuit2 CCI bottom layer: tsv\_layer

Circuit2 CCI top device: tsv

Circuit2 CCI bottom device: tsv

# 网表合并

- 基于transistor level的版图坐标一致网表合并结果

注意26个图形上下重叠，连接了26个pin

Circuit1-Circuit2 connect pin

26:A - 1

27:A - 2

28:A - 3

29:A - 4

30:A - 5

31:A - 6

B6 - B6

B4 - B4

B1 - B1

35:A - 10

36:A - 11

B7 - B7

A - A

B2 - B2

40:A - 15

41:A - 16

B8 - B8

B5 - B5

B3 - B3

45:A - 20

46:A - 21

47:A - 22

48:A - 23

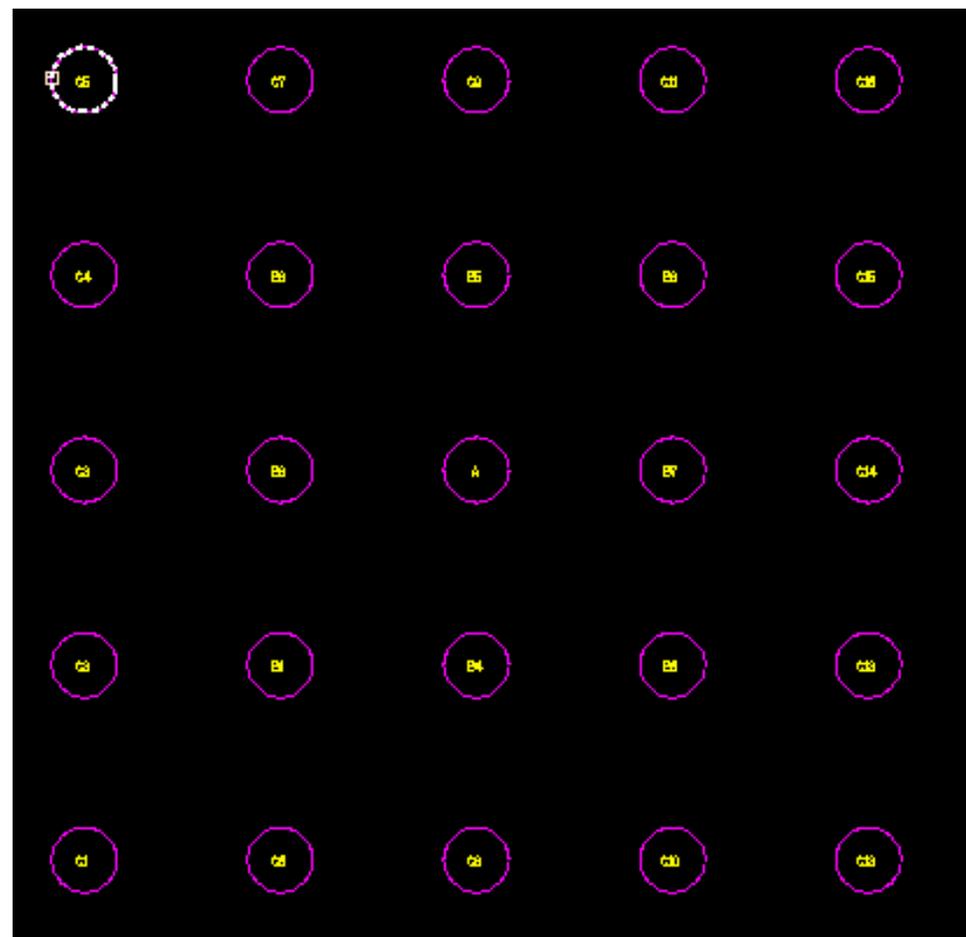
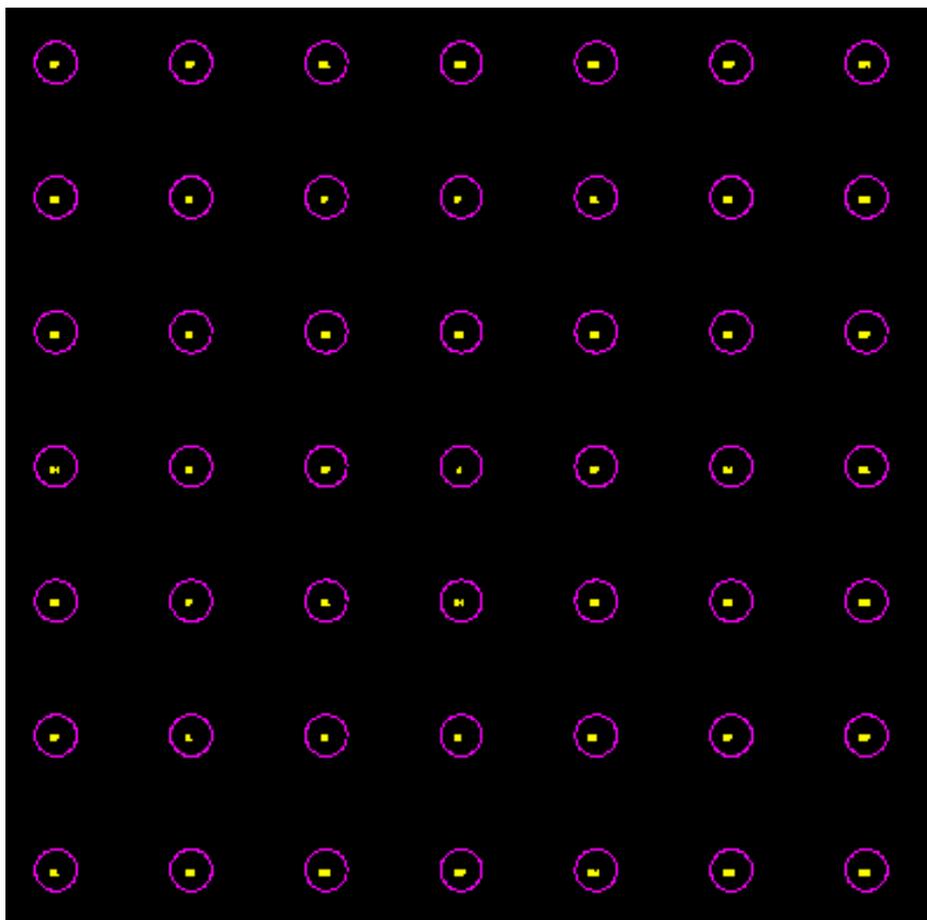
49:A - 24

50:A - 25

single - single

# 3D IC TSV提取流程

- 自动生成TSV Pattern( 可生成3x3, 5x5, 7x7等组合)



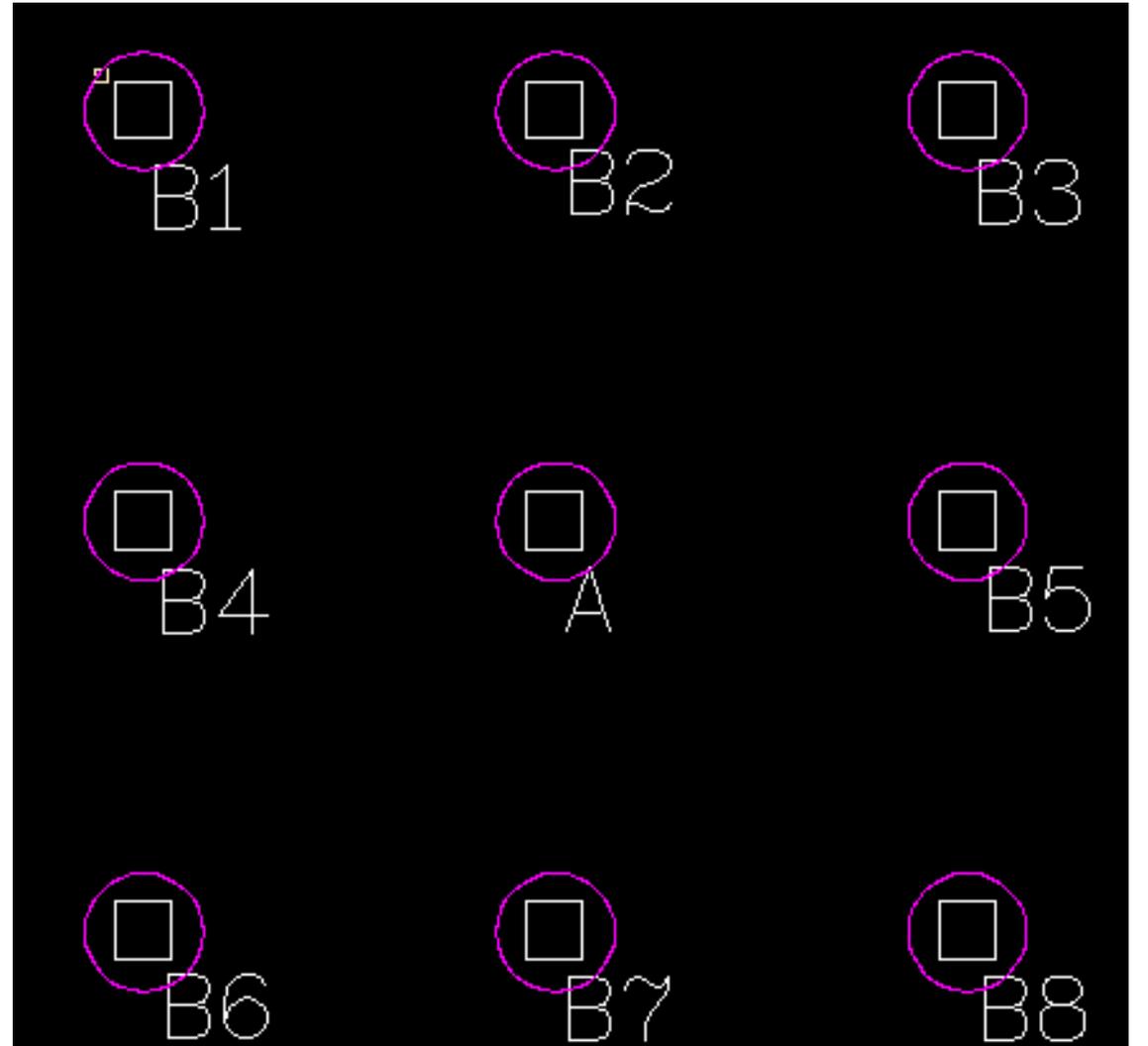
# 3D IC TSV提取流程简化

- 流程简化思路

- 电路频率比较低时，可以忽略TSV之间的耦合电容，这时，流程可以可以去掉TSV 的耦合电容计算，仅计算TSV自电容

# 实测用例1

- TSV版图为3x3的array
  - TSV半径 2.5um
  - TSV中心点距离 20um
  - TSV高度 20um
  - TSV介质厚度 0.1182um
  - 中心TSV名字为A, 电压1.0V
  - B2,B4,B5,B7为紧邻A的TSV, 电压0
  - B1,B3,B6,B8为对角相邻A的TSV, 电压0



# 实测用例1

- 最中心TSV的RCSUB结果

Ctsv电容	中心TSV到紧邻TSV耦合电容(Csi)	中心TSV到对角TSV耦合电容(Csi)	中心TSV对地电容
1V电压: 37.41ff 0V电压: 55.79ff	1.14ff	0.395ff	0.063ff

	中心TSV到紧邻TSV电阻(Rsi)	中心TSV到对角TSV电阻(Rsi)	中心TSV对地电阻
	6.40k	18.30k	115k

# 实测用例1

- 最中心TSV的RCSUB结果

TSV pair	Csi	Rsi
A_0      B1_0	3.952204e-16	1.829853e+04
A_0      B3_0	4.022308e-16	1.797961e+04
A_0      B6_0	4.021936e-16	1.798127e+04
A_0      B8_0	3.929505e-16	1.840423e+04
A_0      B2_0	1.119998e-15	6.457110e+03
A_0      B4_0	1.130552e-15	6.396833e+03
A_0      B5_0	1.126785e-15	6.418217e+03
A_0      B7_0	1.127511e-15	6.414085e+03
A_0      GROUND	6.288000e-17	1.150120e+05

A 对所有其它TSV的耦合电容之和 Csi = 6.159e-15

# 实测用例1

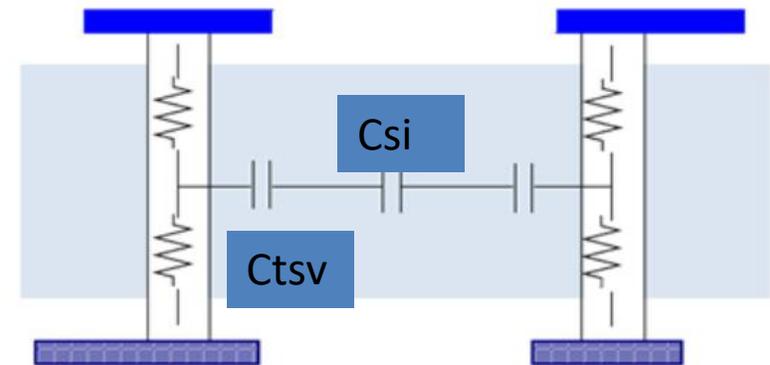
- 最中心TSV的CEFF有效总电容

频率	100K	1M	10M	100M	1G	10G	100G
中心TSV的等效总电容CEFF	37.41ff	37.41ff	37.39ff	36.25ff	31.54ff	8.15ff	5.24ff

- 低频下等效电容趋近于  $C_{tsv}$  37.41ff

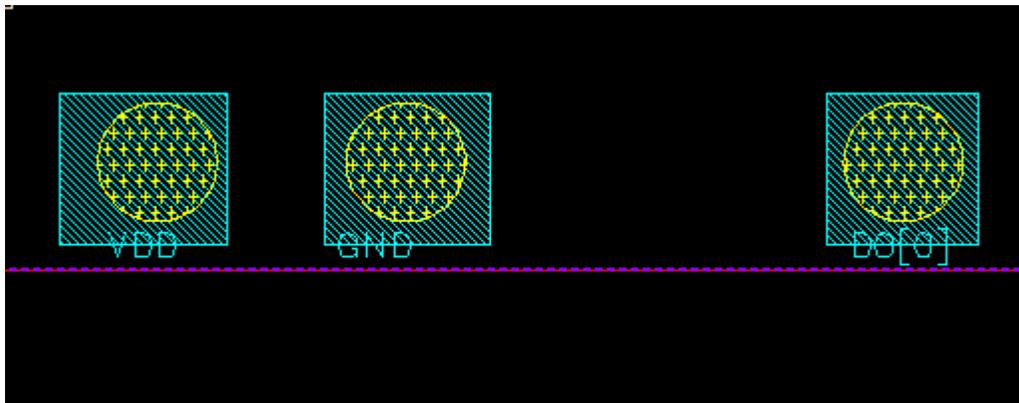
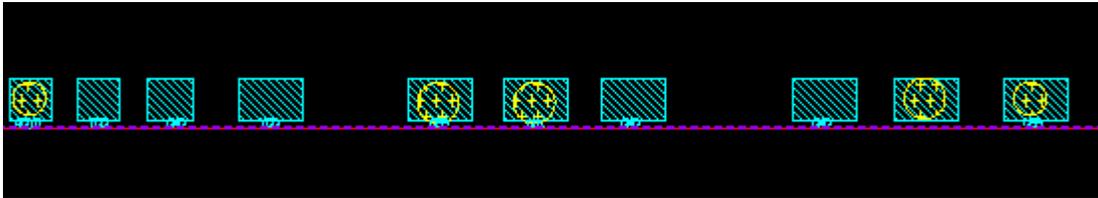
- 高频下等效电容趋近于

$$C_{tsv} * C_{si} / (C_{tsv} + C_{si}) = 5.29ff$$



# 实测用例2

- SRAM版图堆叠3次，通过TSV连接
  - TSV为圆形，
  - 14个pin 通过TSV连接



# 实测用例2

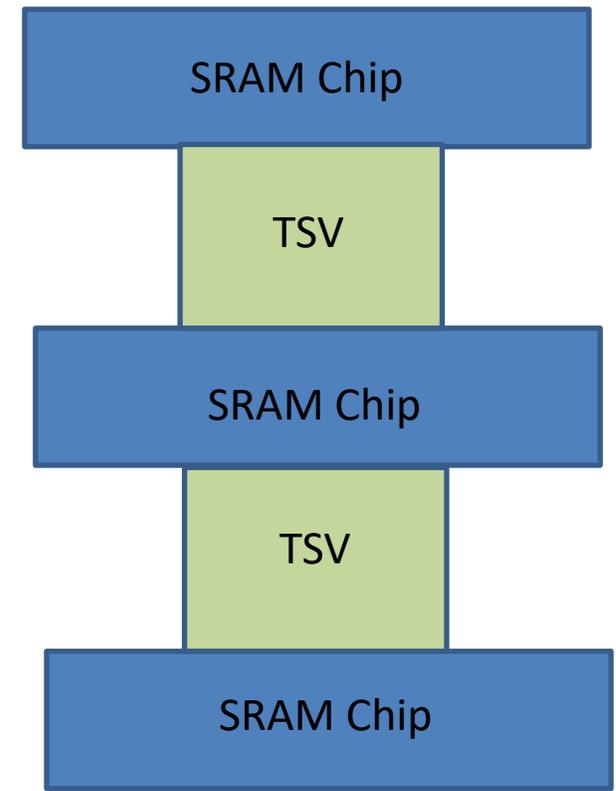
- 运行步骤1: 准备适用于堆叠的TSV lvs文件

tsv\_layer = copy TSV

top\_layer = size tsv\_layer by 0.001

bot\_layer = size tsv\_layer by 0.001

CONNECT top\_layer bot\_layer by tsv\_layer



# 实测用例2

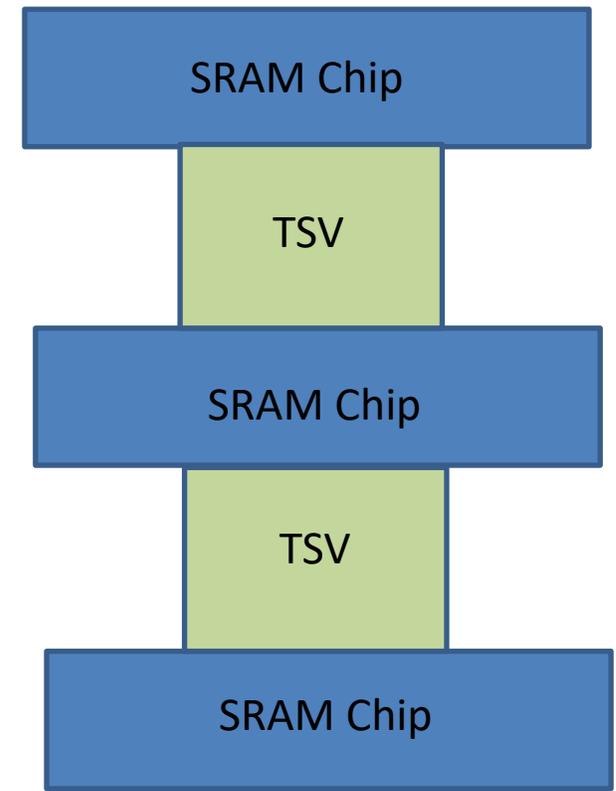
- 运行步骤2: 提取TSV和SRAM寄生参数
  - 调用第三方工具提取SRAM寄生电阻和电容
  - 调用SuperCap提取TSV寄生电容

OPERATING\_FREQUENCY 1G

WAFER\_CONDUCTIVITY 14.3

TSV\_ONLY\_MODE YES

TSV\_COUPLING CEFF



# 实测用例2

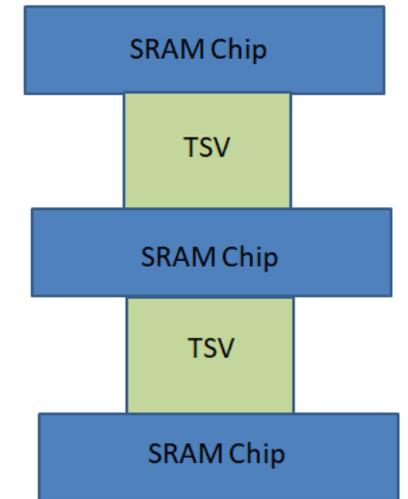
- 运行步骤3: 调用annex合并5个网表

- annex merge\_cci.cfg 其中 merge\_cci.cfg如下:

Top\_to\_Bottom: chip3 tsv chip3 tsv chip3

chip3 TopCell:	topcell_char2
chip3 netlist:	chip3/RC.spf
chip3 CCI Path:	chip3
chip3 CCI query:	query_cmd
chip3 CCI top layer:	top_layer
chip3 CCI bottom layer:	bot_layer
chip3 CCI top device:	tsv_top
chip3 CCI bottom device:	tsv_bot

tsv TopCell:	tsv
tsv netlist:	tsv/nameinput.dspf
tsv CCI Path:	tsv
tsv CCI query:	query_cmd
tsv CCI top layer:	tsv_layer
tsv CCI bottom layer:	tsv_layer

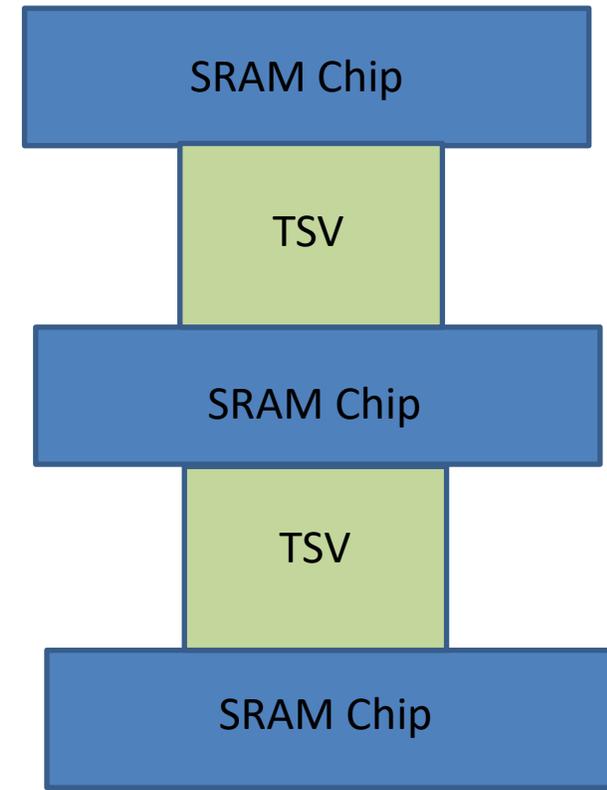


# 实测用例2

合并网表结果如下

Xchip3r1

```
+ chip3r1_DO[0]_tsvr1_DO[0]_chip3r2_DO[0]_tsvr2_DO[0]_chip3r3_DO[0]
+ chip3r1_DI[0]_tsvr1_DI[0]_chip3r2_DI[0]_tsvr2_DI[0]_chip3r3_DI[0]
+ chip3r1_AD[1]_tsvr1_AD[1]_chip3r2_AD[1]_tsvr2_AD[1]_chip3r3_AD[1]
+ chip3r1_AD[0]_tsvr1_AD[0]_chip3r2_AD[0]_tsvr2_AD[0]_chip3r3_AD[0]
+ chip3r1_CEB_tsvr1_CEB_chip3r2_CEB_tsvr2_CEB_chip3r3_CEB
+ chip3r1_CLK_tsvr1_CLK_chip3r2_CLK_tsvr2_CLK_chip3r3_CLK
+ chip3r1_OEB_tsvr1_OEB_chip3r2_OEB_tsvr2_OEB_chip3r3_OEB
+ chip3r1_VDD_tsvr1_VDD_chip3r2_VDD_tsvr2_VDD_chip3r3_VDD
+ chip3r1_GND_tsvr1_GND_chip3r2_GND_tsvr2_GND_chip3r3_GND
+ chip3r1_11:A
+ chip3r1_26:A_tsvr1_11_chip3r2_11:A
+ RS16X2_CM4_WM0_char2
```



# 实测用例2

## 合并网表结果 (续)

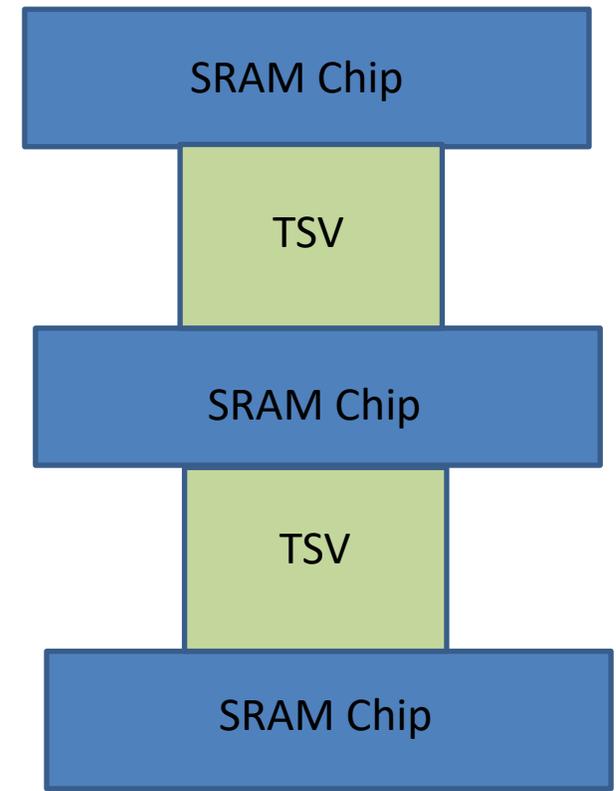
Xtsvr1

```
+ chip3r1_VDD_tsvr1_VDD_chip3r2_VDD_tsvr2_VDD_chip3r3_VDD
+ chip3r1_GND_tsvr1_GND_chip3r2_GND_tsvr2_GND_chip3r3_GND
+ chip3r1_DO[0]_tsvr1_DO[0]_chip3r2_DO[0]_tsvr2_DO[0]_chip3r3_DO[0]
+ chip3r1_DI[0]_tsvr1_DI[0]_chip3r2_DI[0]_tsvr2_DI[0]_chip3r3_DI[0]
+ chip3r1_AD[1]_tsvr1_AD[1]_chip3r2_AD[1]_tsvr2_AD[1]_chip3r3_AD[1]
+ chip3r1_AD[0]_tsvr1_AD[0]_chip3r2_AD[0]_tsvr2_AD[0]_chip3r3_AD[0]
+ chip3r1_CEB_tsvr1_CEB_chip3r2_CEB_tsvr2_CEB_chip3r3_CEB
+ chip3r1_CLK_tsvr1_CLK_chip3r2_CLK_tsvr2_CLK_chip3r3_CLK
+ chip3r1_OEB_tsvr1_OEB_chip3r2_OEB_tsvr2_OEB_chip3r3_OEB
+ chip3r1_26:A_tsvr1_11_chip3r2_11:A
+ tsv
```

# 实测用例2

- 合并结果分析

- 既可以自动连接带有线网名的节点，也可以自动连接带有数字的内部节点。
- 连接上下2个芯片的原则：通过坐标的一致性和layer的上下连接关系来判断，不依赖于节点命名
- 如果上下两个芯片的节点连接在一起，则该节点的命名取名为：每个芯片的名字 + 芯片内部节点名，上下2个芯片之间通过下划线来连接。
- 节点名虽然名字较长，但其连接含义清晰，更有利于用户使用



# 总结

- 3D IC TSV的寄生参数提取流程

- 衬底内部是半导体模型，与普通的绝缘介质提取方法不同
- 衬底之外的TSV需要考虑与其它导体的耦合电容
- 多个芯片堆叠后，每个芯片和TSV的网表都需要单独提取，需要有自动网表合并
- 采用该流程需要用到第三方寄生参数提取工具，也需要用到SuperCap, Annex等自主开发的特色工具，解决主流工具不能解决的问题
- 同时提供服务，帮助用户书写针对TSV提取的相关工艺文件/LVS文件

# 附录：三维寄生电容场求解器 SuperCap工具简介

# SuperCap产品定位

- SuperCap主要用于Foundry做RC Runset的golden校准工具
  - SuperCap是全三维的Field Solver，精度高。
  - SuperCap支持工业界的CCI接口，兼容性较高
  - SuperCap误差可以由用户自主设置，精度可调可控
  - SuperCap目前支持28nm以上工艺的DFM效应
- SuperCap也可用于计算TSV的衬底耦合电容

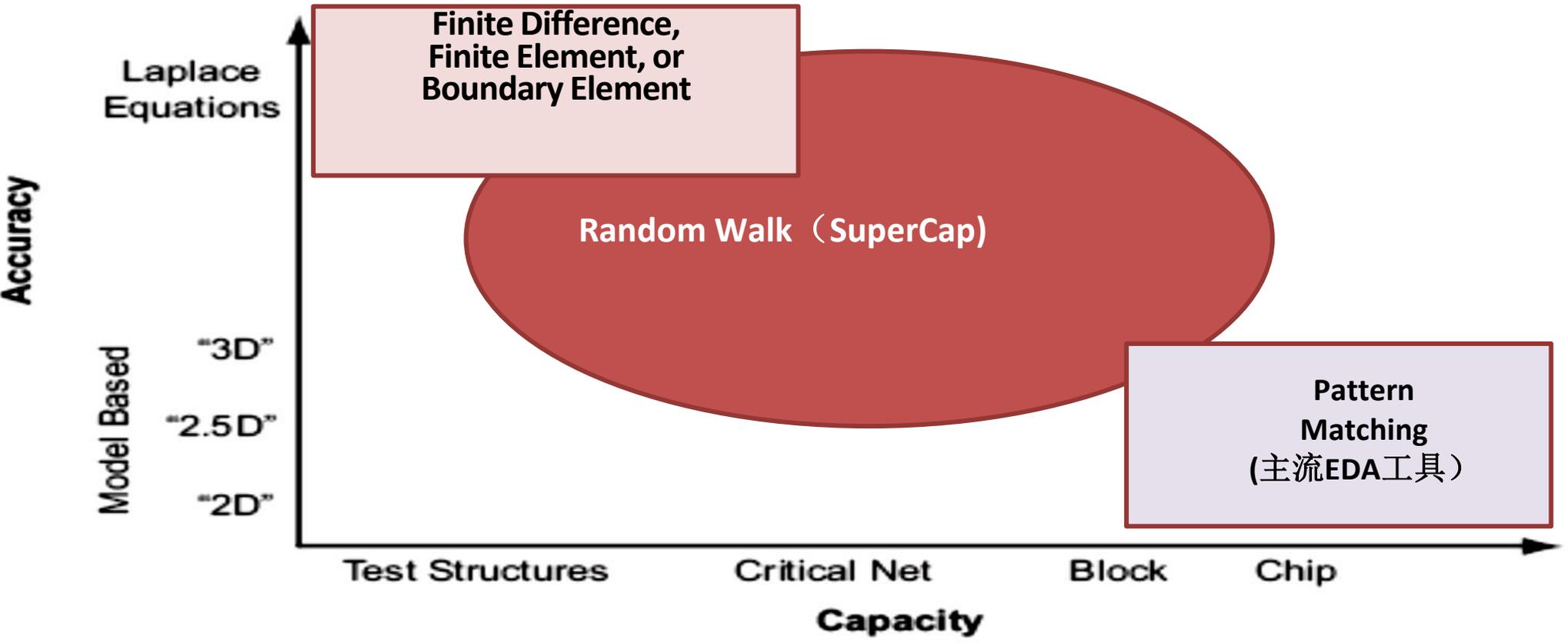
# SuperCap主要功能

- SuperCap基本功能
  - 全三维寄生电容提取器
  - 采用Random Walk（随机漫步）算法，比传统的有限差分法、边界元素法等确定性算法效率更高，精度可控
  - 输入数据与主流工具兼容，可在主流寄生参数工具环境下直接运行SuperCap
  - 输出数据与主流工具兼容，可直接进行后仿真
  - 支持多线程并行加速计算

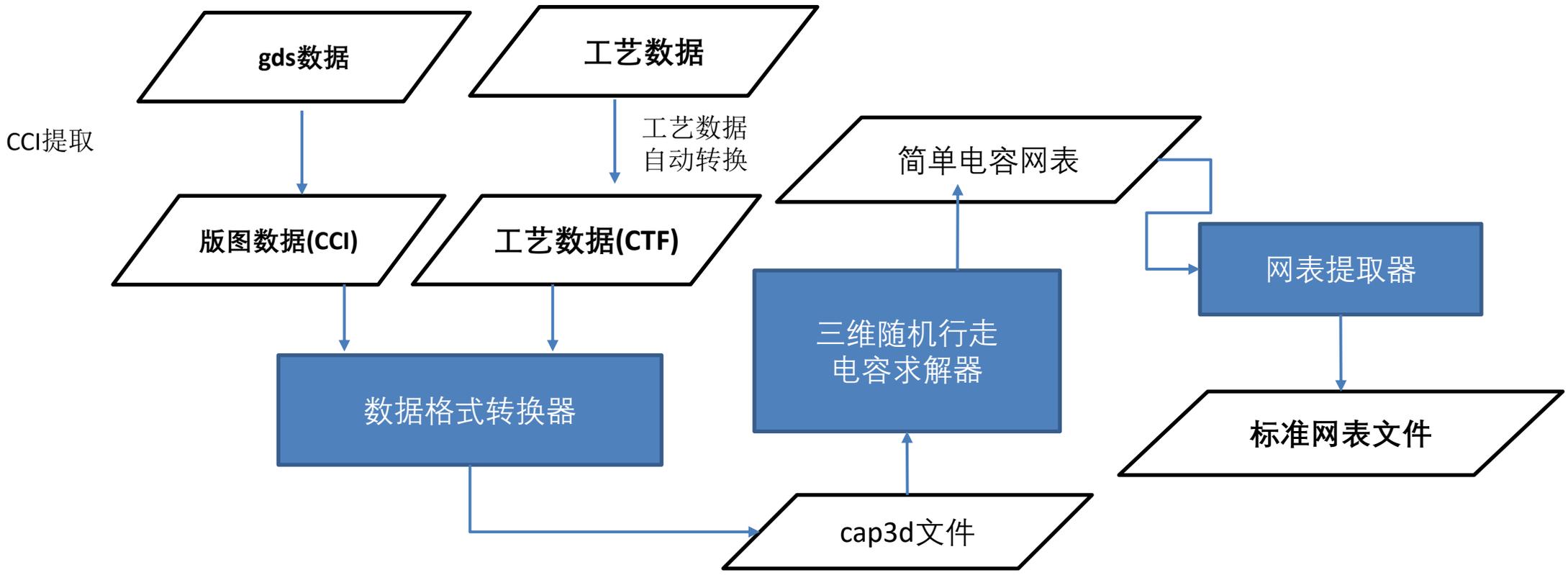
# SuperCap主要功能

- 传统寄生电容提取方法有2类
  - 准三维方法（Pattern Match方法）
  - 基于有限差分、有限元、边界元等确定性算法的三维场求解器（field solver）方法
  - 这2类方法速度差4个数量级，即1万倍
- SuperCap介于上述2类方法之间
  - 在达到三维场求解器精度的前提下，速度比Pattern Match慢2个数量级，比传统三维field solver快2个数量级

# SuperCap主要功能



# SuperCap主要功能



# SuperCap主要功能

- SuperCap不需要建立电容库
  - 主流EDA工具采用准三维方法（Pattern Match方法），运行工具前需要花很长时间建立电容库，即需要先生成nxtgrd/qrcTechFile/rules.C等文件才能运行电容提取
  - SuperCap采用全三维现场计算方法，不需要建立电容库，直接用类似于itf/ict/mipt格式的ctf文件，即可进行电容提取

# SuperCap主要功能

- SuperCap计算精度用户可以自主控制

- 在SuperCap的命令文件中，用户可设置

SELF_CAP_ERR	0.03	( 集总电容误差设置为3% )
COUPLING_CAP_ERR	0.03	( 耦合电容误差设置为3% )

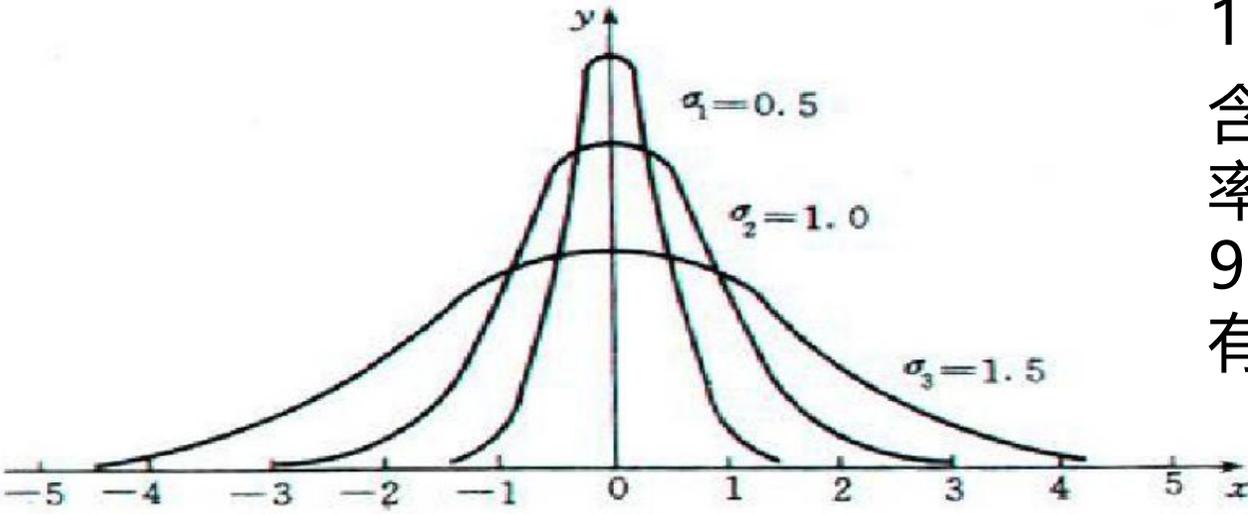
- Supercap根据用户的设置自动迭代收敛到预期精度
- 传统的确定性算法如有限差分、有限元、边界元需要通过网格数来控制精度，网格数与精度没有直接的对应关系,用户很难事先控制计算误差

# SuperCap主要功能

- SuperCap误差设置含义

- 误差的百分比就是正态分布的标准差西格玛 ( $\sigma$ )

- $\sigma$  不同 (标准差)



正态分布标准差  $\sigma$  的特性

`self_cap_err = 0.01` , 即  $\sigma = 1.0$  ,

含义: 集总电容有99.73%的概率与期望值误差在3%以内, 有95.45%的概率误差在2%以内, 有68.27%的概率误差在1%以内。

# SuperCap主要功能

- SuperCap支持多线程并行加速计算

- SuperCap支持多线程并行计算，命令文件设置方法如下：

- ```
THREADS 8
```

- 设置Threads = 8的含义是：并行按照最大8个线程同时运行，最大可提高效率接近8倍。

- 用户可根据CPU内部的核数来设置最大的threads数。

# SuperCap主要功能

- SuperCap是transistor level提取
  - SuperCap支持transistor level的提取，可以精确提取器件之间的电容，与物理实际模拟结果最接近
  - SuperCap暂不支持cell level的提取，即不支持LEF/DEF, OpenAccess格式的输入文件

# SuperCap主要功能

- SuperCap可以忽略某些layer之间的电容
  - 常见的mos器件、MIM器件、MOM器件、RF器件，其内部寄生参数已经包含在spice model中，因此寄生参数提取时要忽略掉这些内部电容
  - SuperCap通过设置忽略layer之间电容的方法可以有效忽略掉器件内部电容，保证与主流工具结果一致
- SuperCap是flatten模式提取，精度高
  - SuperCap作为一款高精度field solver，采用flatten模式的提取，精度比hierarchical的近似提取更高

# SuperCap主要功能

- SuperCap支持关键线网提取

- SuperCap支持对某个或某些线网的提取，不必提取全部线网
- 设置部分提取线网的语法如下：

```
EXTRACT_NETS I_test A* B1*
```

- 其中\*是通配符，如果设置上述语句，则工具只提取满足上述语法的线网，其余线网不作为主导体提取。

# SuperCap主要功能

- SuperCap支持 half-node的版图缩放
  - 某些工艺有半个工艺节点的概念，例如，0.13微米的半节点是0.11微米，65nm的工艺半节点是55nm等。
  - 一般情况下，半节点的寄生参数工艺文件与其母节点的工艺文件完全一样，不做修改，仅仅是版图数据做了一定比例的缩放。这样的好处是：一套工艺文件可以既可以用于母节点，也可以用于半节点。
  - SuperCap支持半节点的设置，书写如下  
MAGNIFICATION 0.9
  - 0.9是版图缩放比例，用户可任意设置该数字。

# SuperCap输入输出

- SuperCap输入文件准备

- 工艺文件准备

SuperCap采用CTF格式描述工艺文件，如果用户有其它格式的文件，可通过ctfx 的命令自动转换为CTF文件。

- CCI数据准备

SuperCap采用CCI接口读取版图数据，用户需要准备好lvs文件、query cmd文件、gds文件，通过运行cci流程得到CCI数据。

- 版图layer与工艺layer映射文件准备

SuperCap采用与主流工具相同格式的mapping file文件，不用修改。

- Supercap命令文件准备

SuperCap命令文件是一组简单的控制参数，用text文本编写。

# SuperCap输入输出

- SuperCap命令文件准备

- 命令文件是一个text文本文件，控制简洁，典型如下：

|                    |               |    |
|--------------------|---------------|----|
| CTF_FILE           | new.ctf       | 必写 |
| TECH_LAYER_MAP     | mappingfile   | 必写 |
| QUERY_CMD          | query_cmd     | 必写 |
| #####              |               |    |
| NETLIST_FILE       | nmos_lvt.dspf | 必写 |
| #IGNORE_MOS_DEVICE | ALL           | 可选 |
| IGNORE_SUBSTRATE   | ALL           | 必写 |
| MAGNIFICATION      | 0.9           | 可选 |
| SELF_CAP_ERR       | 0.015         | 可选 |
| #COUPLING_CAP_ERR  | 0.01          | 可选 |
| THREADS            | 8             | 可选 |
| POWER_NETS:        | vpwrh vrtnh   | 可选 |
| EXTRACT_NETS       | I_test        | 可选 |

# SuperCap输入输出

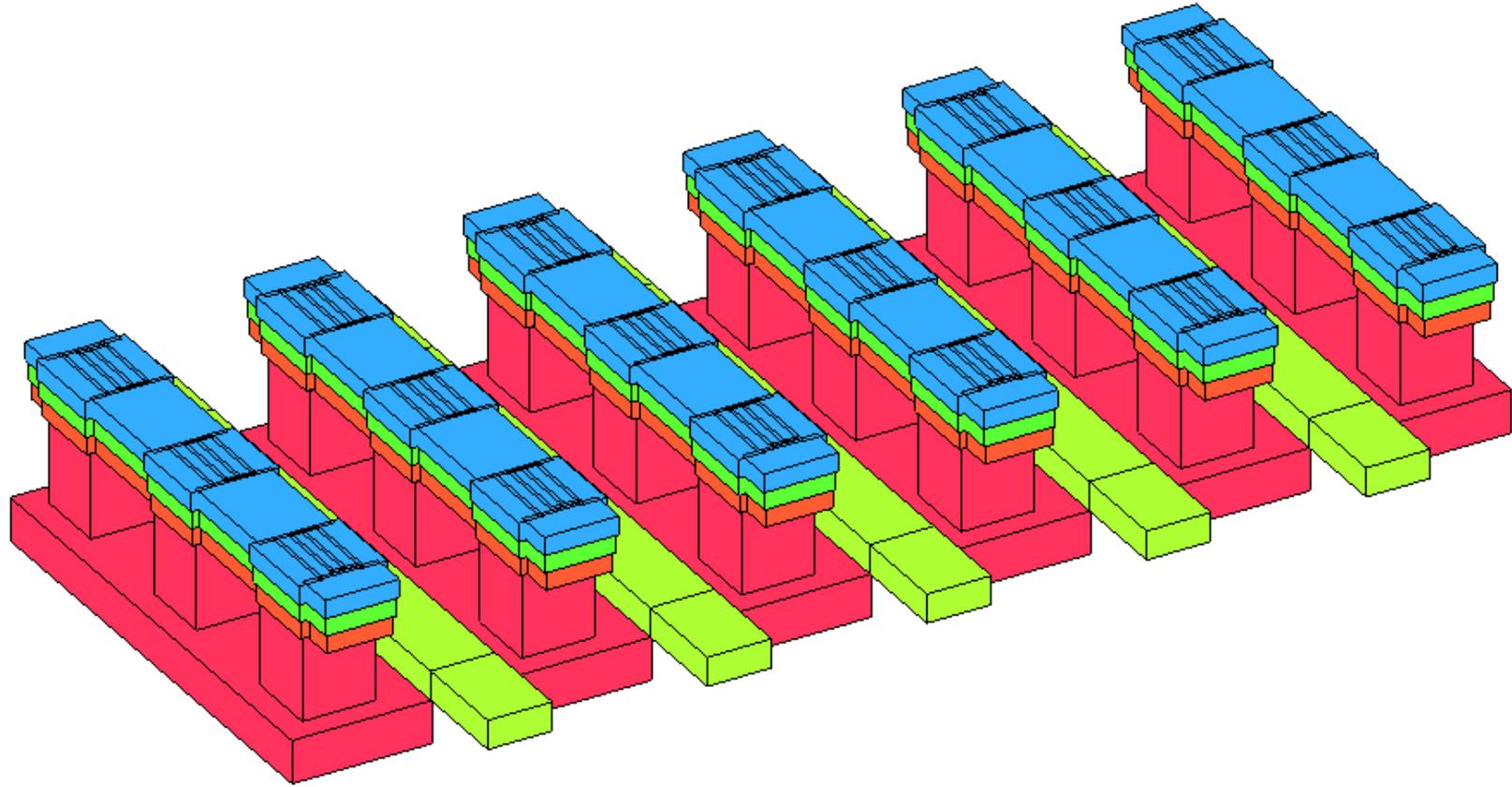
- SuperCap的输出文件

- Supercap的输出文件是标准的dspf格式，包含了器件信息，可以直接进行后仿真。典型如下：

```
*|GROUND_NET 0
*|NET LPW 3.77469e-15
C0 LPW 0 5.57972e-17
*|NET nmos_lvt_d 4.92675e-16
C1 nmos_lvt_d LPW 3.66298e-16
C2 nmos_lvt_d 0 6.58426e-18
*|NET nmos_lvt_g 3.02901e-16
C3 nmos_lvt_g nmos_lvt_d 5.40531e-17
C4 nmos_lvt_g LPW 1.89634e-16
C5 nmos_lvt_g 0 6.89019e-18
*|NET nmos_lvt_s 5.05946e-16
C6 nmos_lvt_s nmos_lvt_g 5.23233e-17
C7 nmos_lvt_s nmos_lvt_d 6.57399e-17
```

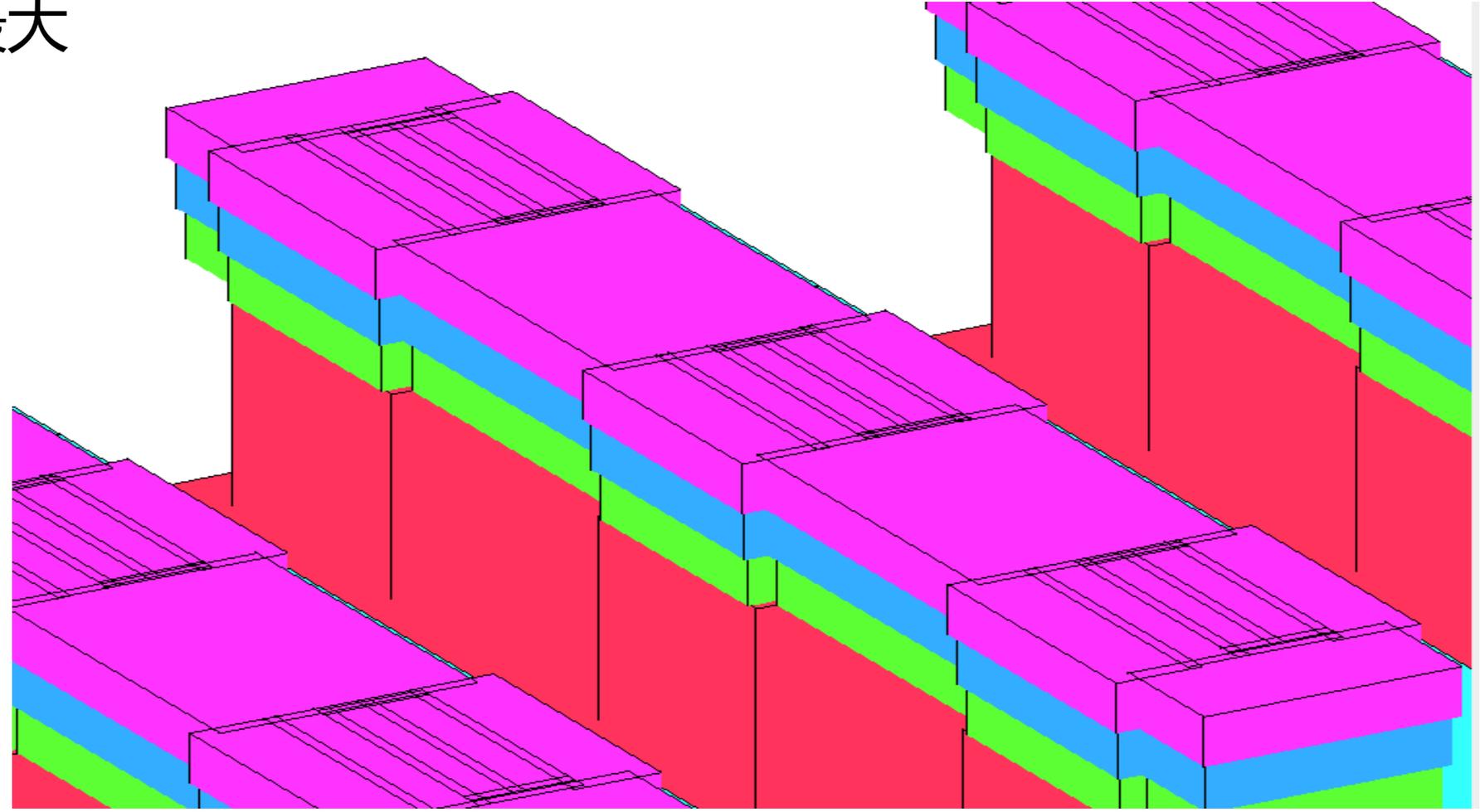
```
X0 nmos_lvt_d nmos_lvt_g nmos_lvt_s
LPW nlp2_lvt_iso_ckt ad=1.3074e-12
as=1.3074e-12 l=9e-08 m=1 nf=1
nrd=0.0688389 nrs=0.0688389 nsd=0
pd=9.316e-06 ps=9.316e-06 sa=3e-07
sb=3e-07 sca=0.784878 scb=3.9785e-08
scc=4.31278e-16 w=4.358e-06
.ENDS
```

# SuperCap显示三维图形



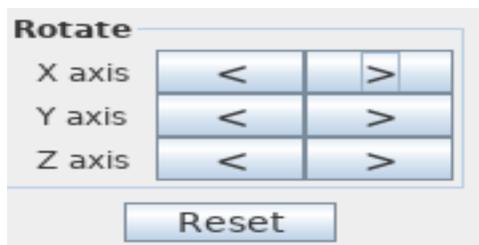
# 放大M1的三维显示

鼠标右键拖动，可把选中图形  
Zoom到最大

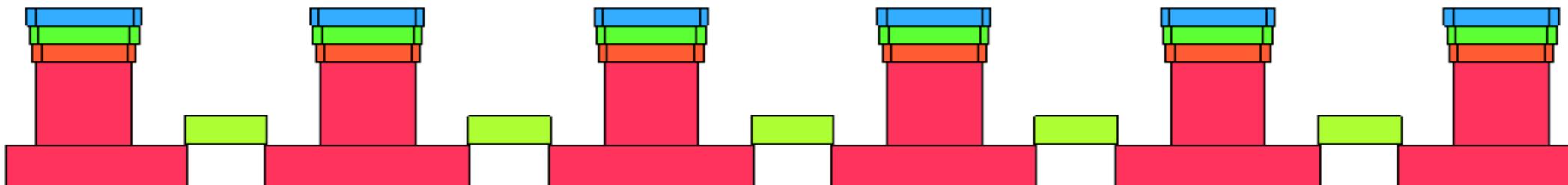


# 旋转到z平面

可以通过旋转看到不同截面的形状



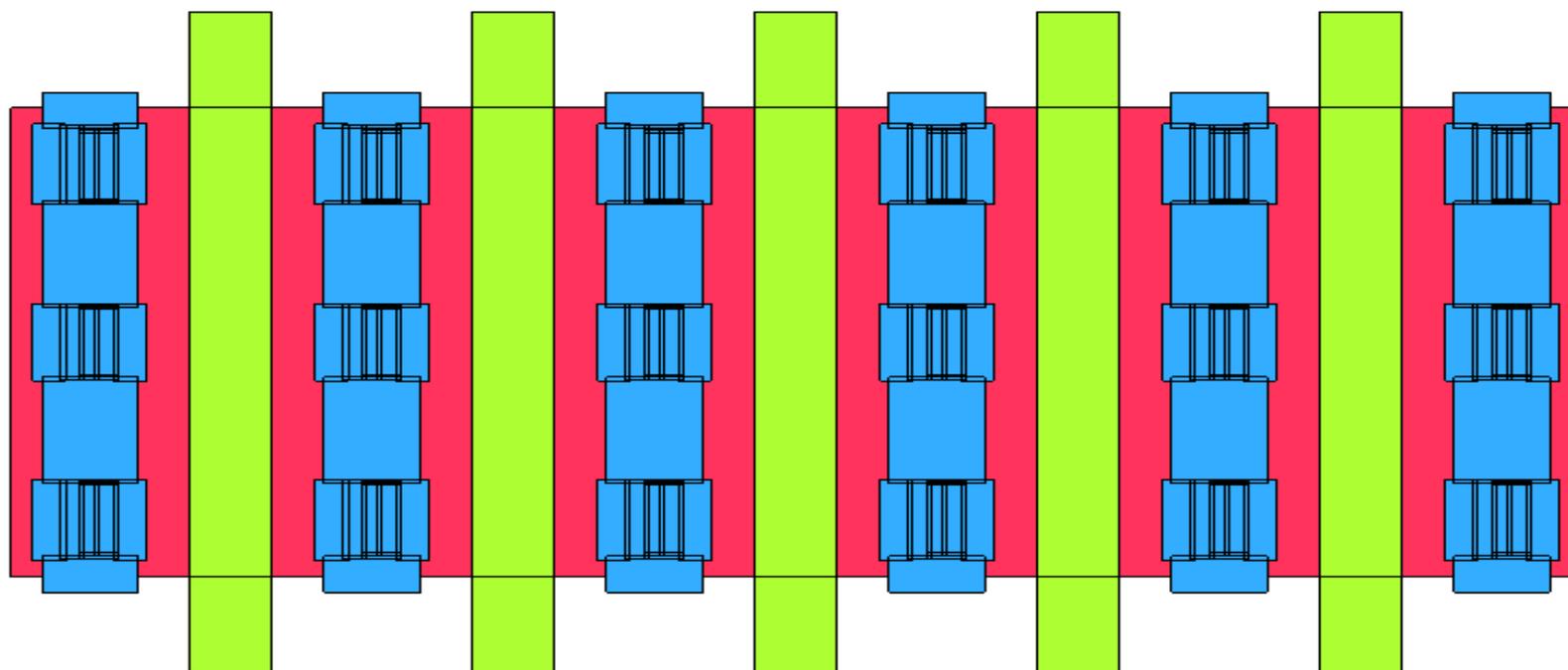
点击Z向右旋转1次，X向右旋转1次  
得到cross view如下



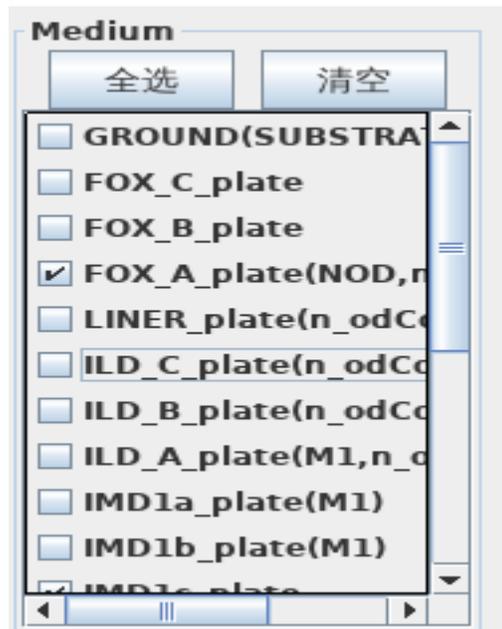
# XOY, YOZ, XOZ 截面

也可直接点击界面中的快捷键： XOY YOZ XOZ

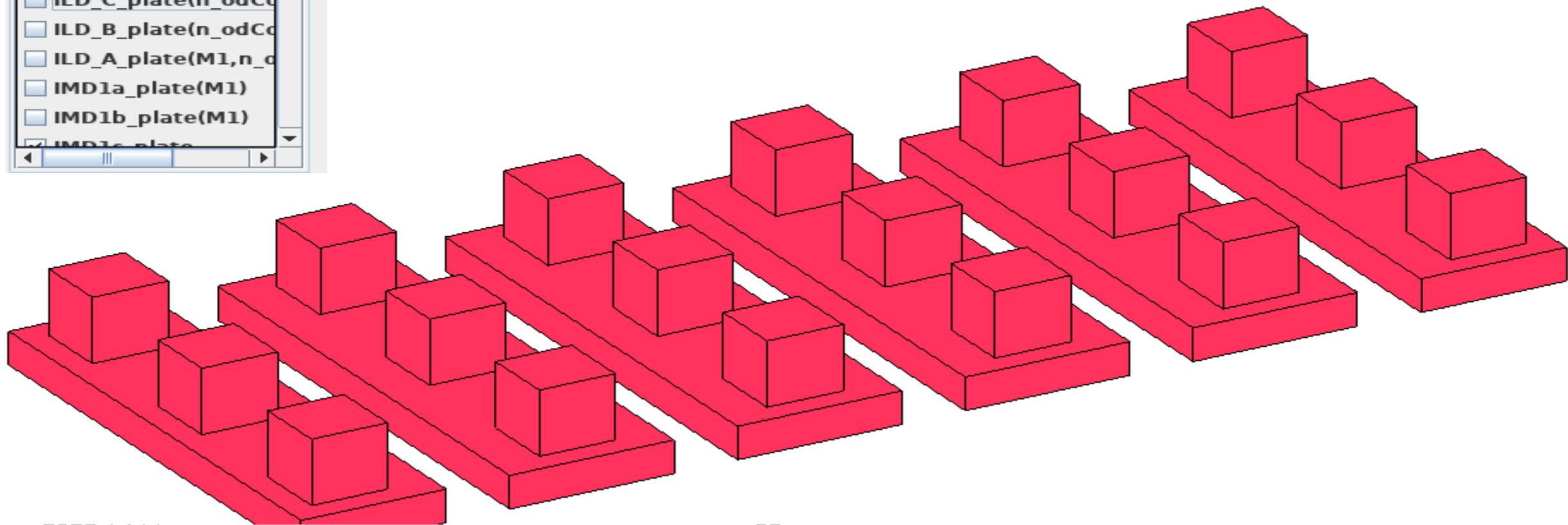
| Reset  |     |
|--------|-----|
| Visual | XOY |
| YOZ    | XOZ |



# 分层显示diffusion和cont

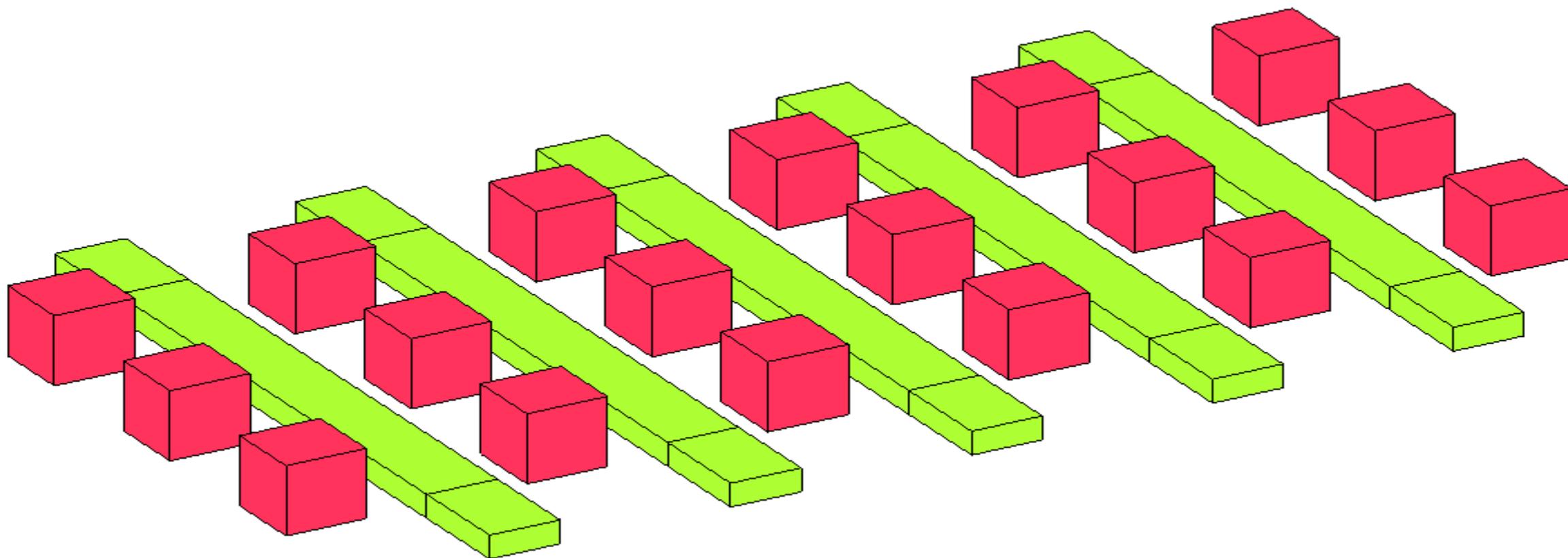


选中右侧的Layer, 只显示某层介质内的导体



# 分层显示gate和cont

可以看到，Gate到Contact距离近，Via电容不能忽略



# 总结

- SuperCap主要用于Foundry做RC Runset的golden校准工具
  - SuperCap是全三维的Field Solver，精度高。
  - SuperCap支持工业界的CCI接口，兼容性较高
  - SuperCap误差可以用户自主设置，精度可调可控
  - SuperCap目前支持工业界常用的DFM效应
- SuperCap也可用于计算TSV的衬底耦合电容