**静态时序分析圣经翻译计划——第三章：标准单元库 （上）**

[**赵俊军**](https://www.zhihu.com/people/zhao-jun-jun-19)

[**​**](https://www.zhihu.com/question/48510028)

上海交通大学 电子与通信工程硕士在读

31 人赞同了该文章

本章节介绍库（library）里单元描述中所提供的时序信息。单元可以是标准单元、IO缓冲器或者是如USB内核这样的复杂IP。

●除时序信息外，库单元描述中还包含一些其它属性，例如单元面积和功能，这些属性与时序无关，但在RTL综合（synthesis）过程中会用到。在本章节中，我们仅关注与时序和功耗计算有关的那些属性。

●可以使用各种标准格式来描述库单元，各种格式的内容基本相似，本书中使用**Liberty**语法描述库单元。

●本章节的前面部分介绍了线性和非线性时序模型，随后介绍了用于纳米技术的高级时序模型，这些将在3.7节中进行介绍。

**3.1 引脚电容**

单元的每个输入和输出都可以在引脚（pin）上指定电容。在大多数情况下，仅为单元输入引脚指定电容，而不为输出引脚指定电容，即大多数单元库中的输出引脚电容为0。



上面的示例展示了输入INP1引脚电容值的一般规格（specification）。在最基本的格式中，引脚电容被指定为单个值（在上面的示例中为0.5个单位）。电容单位通常为皮法拉（pF），一般在库文件的开头指定。单元描述中还可以为**rise\_capacitance**（0.5个单位）和**fall\_capacitance**（0.45个单位）分别指定值，这些值是指引脚INP1上发生电平上升和下降跳变时的值。也可以将**rise\_capacitance**和**fall\_capacitance**的值指定为范围，并在描述中指定下限值和上限值。

**3.2 时序模型**

逻辑单元的时序模型（timing model）旨在为设计中的各种单元实例（instance）提供准确的时序信息。通常会从单元的详细电路仿真中获得时序模型，用以对单元工作时的实际情况进行建模，且需要为逻辑单元的每个时序弧都建立一个时序模型。

图3-1

让我们首先考虑图3-1中所示的反向器（inverter）的时序弧。 由于这是一个反相器，因此输入端的电平上升（下降）跳变会导致输出端的电平下降（上升）跳变。表征这个反相器的两种延迟是：

●Tr：输出上升沿延迟

●Tf：输出下降沿延迟

注意，延迟是根据单元库中定义的阈值（threshold）点（详见2.4节）测量的，通常为50％Vdd。 因此，延迟值是指从输入经过其阈值点到输出经过其阈值点的延迟。

通过反相器的时序弧的延迟取决于两个因素：

1.输出负载，即反相器输出引脚上的电容负载

2.输入信号的过渡（transition）时间

延迟值与负载电容有直接关系：负载电容越大，延迟越大。在大多数情况下，延迟会随着输入信号过渡时间的增加而增加。而在某些情况下，输入信号阈值点（用于测量延迟）与单元的内部开关点（switching point）明显不同。在这种情况下，通过单元的延迟可能相对于输入信号过渡时间表现出非单调行为：较大的输入信号过渡时间可能会产生较小的延迟，尤其是在输出负载较小的情况下。

逻辑单元输出引脚的压摆（slew）主要取决于输出引脚电容：输出信号过渡时间会随着输出负载的增加而增加。 因此，在输入端压摆较大（输入信号过渡时间较长）的情况下，选择合适的单元类型及输出负载，可以改善输出端的压摆。图3-2展示了通过调节逻辑单元的输出负载，可以改善或恶化单元输出信号过渡时间的情况。

图3-2

**3.2.1 线性时序模型**

一个简单的时序模型是线性延迟模型（linear delay model），该模型使用含两个参数的线性函数表示逻辑单元的延迟和输出过渡时间，这两个参数是：输入过渡时间（input transition time）和输出负载电容（output load capacitance）。通过逻辑单元的延迟（D）使用线性模型的一般形式如下所示：

​ D = D0 + D1 \* S + D2 \* C

其中D0，D1，D2是常数，S是输入过渡时间，C是输出负载电容。 对于亚微米（submicron）技术，线性延迟模型在输入过渡时间和输出负载电容的范围内并不准确，因此，目前大多数单元库都使用更复杂的模型，例如非线性延迟模型（non-linear delay model）。

**3.2.2 非线性延迟模型**

大多数单元库都包括表格模型（table model），用于为单元的各种时序弧指定延迟并进行时序检查。一些用于纳米技术的较新的时序库还提供了基于电流源的高级时序模型（例如CCS，ECSM等），本章稍后将对此进行介绍。 这些表格模型被称为**NLDM**（**N**on-**L**inear **D**elay **M**odel），可用于延迟、输出压摆计算或其他时序检查。表格模型中提供了：在单元输入引脚处输入过渡时间和输出引脚处输出负载电容的各种组合下通过单元的延迟。

NLDM模型以二维形式表示，两个独立变量是输入过渡时间和输出负载电容，表中的数值表示延迟。以下是一个典型反相器单元的表格模型示例：



上面的示例描述了输出引脚OUT的延迟信息。单元描述里的这个部分包含了从引脚INP1到引脚OUT时序弧的上升和下降延迟模型，以及引脚OUT的**max\_transition**允许时间。上升和下降延迟有单独的模型（用于输出引脚），分别标记为**cell\_rise**和**cell\_fall**。索引（index）变量的类型和顺序在以下查找表模板**delay\_template\_3x3**中进行了描述：



该查找表模板指定了表中的第一个变量是输入过渡时间，第二个变量是输出负载电容。表中数值的给定类似于嵌套循环，其中第一个索引（index\_1）为外部循环变量，第二个索引（index\_2）为内部循环变量，依此类推。在上面的示例中每个变量都有三个可能值（条目），因此它对应于一个3×3的表。在大多数情况下，表的条目也可按照表的格式设置，然后可以将第一个索引（index\_1）视为行索引，而第二个索引（index\_2）则等同于列索引。索引值（例如1000）是伪占位符，它们会被**cell\_fall**和**cell\_rise**延迟表中的实际索引值覆盖。指定索引值的另一种方法是在模板定义中指定索引值，而不是在**cell\_rise**和**cell\_fall**表中指定索引值，例如以下示例：



根据延迟表，输入下降过渡时间为0.3ns且输出负载为0.16pf时，反相器的上升延迟为0.1018ns。 由于输入的下降沿跳变导致反相器输出的上升沿跳变，因此当输入引脚发生下降沿跳变时，应该去查询**cell\_rise**延迟表。

将表格中的延迟值表示为输入过渡时间和输出负载电容这两个变量的函数的这种形式称为非线性延迟模型，因为在这些表格中可以看到延迟随输入过渡时间和输出负载电容的非线性变化。

注意，表格模型也可以是3维的，例如一个具有互补输出Q和QN的触发器，这将在3.8节中进行介绍。

NLDM模型不仅可以用于计算延迟，而且还可用于计算逻辑单元输出引脚的过渡时间，该时间同样由输入过渡时间和输出负载电容来表征。因此，还存在着另外一张独立的用于计算单元的输出上升和下降过渡时间的二维表格。



输出过渡时间也同样有两张表：**rise\_transition**和**fall\_transition**。如第2章所述，过渡时间是根据特定的压摆阈值点（通常为电源的10％-90％）测量的。

综上所述，使用NLDM模型的反相器单元具有下列表格模型：

​ ● 上升延迟时间（Rise delay）

​ ● 下降延迟时间（Fall delay）

​ ● 上升过渡时间（Rise transition）

​ ●下降过渡时间（Fall transition）

假设有这么一个反相器单元，其输入过渡时间和输出负载电容如图3-3所示。对于15ps的输入（下降）过渡时间和10fF负载，可从**cell\_rise**表中获得输出上升延迟时间，而从**cell\_fall**表中可获得20ps输入（上升）过渡时间和10fF负载下的输出下降延迟时间 。

图3-3

指定单元逻辑功能为“反相”（invert）的信息在哪里呢？该信息可以在时序弧里的**timing\_sense**字段中找到，但在某些情况下如果该字段中未指明，则应该也可从引脚功能（pin function）中推断出。

例如对于反相器单元，时序弧为**negative\_unate**，这表明输出引脚电平跳变方向与输入引脚电平跳变方向相反。因此，在**cell\_rise**表中进行查找时使用的参数是输入引脚的下降过渡时间。

**查找非线性延迟模型**

这个小节通过一个例子来说明表格模型的查找。如果输入过渡时间和输出电容正好与表格条目中的值相对应，则表格查找是很简单的，因为可以直接从表中对应的位置读出时序值。以下示例对应于一般情况，即输入过渡时间和输出电容无法与表格条目中的值对应的情况。在这种情况下，可利用二维插值的方法来得到最终的时序值，一般选择每个维度中两个最接近的表格条目中的值以进行表格插值。考虑输入过渡时间为0.15ns、输出电容为1.16pF的输出下降沿查找表（前文的示例）， 下面复制了与二维插值有关的输出下降过渡表的相应部分。



在下面的公式中，两个index\_1值分别表示为x1和x2；两个index\_2值分别表示为y1和y2，而对应的表值分别表示为T11，T12，T21和T22。

如果需要在表中查找（x0，y0）的值，则查找值T00只能通过插值获得，计算公式如下：

​ T00 = X20 \* Y20 \* T11 + X20 \* Y01 \* T12 + X01 \* Y20 \* T21 + X01 \* Y01 \* T22

其中

​ X01 = （x0 - x1）/（x2 - x1）

​ X20 = （x2 - x0） /（x2 - x1）

​ Y01 = （y0 - y1）/（y2 - y1）

​ Y20 = （y2 - y0）/（y2 - y1）

将0.15ns代入index\_1，将1.16pF代入index\_2，根据上述公式可计算出fall\_transition值为：

​ T00 = 0.75 \* 0.25 \* 0.1937 + 0.75 \* 0.75 \* 0.7280 + 0.25 \* 0.25 \* 0.2327 +0.25 \* 0.75 \* 0.7676 = **0.6043**

请注意，上面的等式对于内插和外推均有效，即当索引值（x0，y0）超出范围时依然成立。例如，当index\_1索引值为0.05且index\_2索引值为1.7时，fall\_transition值计算如下：

​ T00 = 1.25 \* （-0.25） \* 0.1937 + 1.25 \* 1.25 \* 0.7280 + （-0.25）\*（-0.25） \* 0.2327 + （-0.25） \* 1.25 \* 0.7676 = **0.8516**

**3.2.3 阈值规格和压摆降额**

压摆值（slew）基于的是在库中指定的测量阈值点，大多数上一代的库（0.25um或更旧的库）都使用10％和90％作为压摆（或称过渡时间）的测量阈值点。

压摆阈值点的选择对应的是波形的线性部分。随着技术的发展，实际波形最线性的部分通常在30％至70％之间。 因此，大多数新一代时序库都将压摆测量阈值点指定为Vdd的30％和70％。但是，由于之前测得的过渡时间在10％至90％之间，因此在填充库时，通常将测得的30％至70％的过渡时间加倍，这由压摆降额系数（slew derate factor）指定，通常指定为0.5。压摆测量阈值点为30％和70％且压摆降额系数为0.5，等效于测量阈值点为10％和90％。 阈值设置的示例如下：



上面的设置规定了要将表格中的过渡时间乘以0.5，以获得与阈值（30%-70%）设置相对应的过渡时间。这意味着表格中的值（以及相应的索引值）实际上是10%-90%阈值点的测量值。在标定过渡时间值时，首先在30%-70%处测量，然后再把测量值外推到10％至90％（（70-30）/（90-10）= 0.5）。

以下是压摆测量阈值点设置的另一个例子：

**slew\_lower\_threshold\_pct\_fall**：20.0；

**slew\_upper\_threshold\_pct\_fall**：80.0；

**slew\_lower\_threshold\_pct\_rise**：20.0；

**slew\_upper\_threshold\_pct\_rise**：80.0；

**slew\_derate\_from\_library** not specified

在上述20%-80%压摆阈值设置的例子中，未指定**slew\_derate\_from\_library**（默认值为1.0），这意味着库中的过渡时间不会降额（derate）。表格中的过渡时间值直接对应于20%-80%的阈值点，如图3-4所示：

图3-4

对上述单元库中压摆阈值设置稍作修改，如下所示：

**slew\_lower\_threshold\_pct\_fall**：20.0；

**slew\_upper\_threshold\_pct\_fall**：80.0；

**slew\_lower\_threshold\_pct\_rise**：20.0；

**slew\_upper\_threshold\_pct\_rise**：80.0；

**slew\_derate\_from\_library**：0.6；

在这种情况下，将**slew\_derate\_from\_library**设置为0.6，并将压摆测量阈值点指定为20％和80％，这意味着库中的过渡时间表里的数据对应于外推到0％至100％（（80-20）/（100-0）= 0.6）的值，如图3-5所示：

图3-5

指定了压摆降额后，计算延迟时使用的压摆值等于：

​ 库中过渡时间值 \* 压摆降额系数 （ **library\_transition\_time\_value** \* **slew\_derate**）

这就是延迟计算工具在内部使用的压摆值，是对应于指定的压摆阈值测量点的。

**3.3 时序模型-组合逻辑单元**

考虑一个两输入与门（and）的时序弧，由前面章节所述可知该单元的两个时序弧均为正单边类型（positive\_unate）， 因此，输入引脚的电平上升对应于输出引脚的电平上升，反之亦然。

对于这个两输入与门，共有以下4种延时：

● A -> Z ：输出上升沿延迟（Output rise）

● A -> Z ：输出下降沿延迟（Output fall）

● B -> Z ：输出上升沿延迟（Output rise）

● B -> Z ：输出下降沿延迟（Output fall）

图3-6

这意味着对于NLDM模型，将会有四个表格模型用于指定延迟。同样，也将有四个表格模型用于指定输出过渡时间（压摆）。

**3.3.1 延迟与压摆模型**

以下是一个用于分析三输入与非门（nand）单元的输入INP1到输出OUT的时序模型：



在上面的例子中，使用了两个单元延迟表**cell\_rise**和**cell\_fall**以及两个过渡时间表**rise\_transition**和**fall\_transition**来描述从INP1到OUT的时序弧的特性，上面的例子中还包括了输出的最大过渡时间（**max\_transition**）值。

如章节2.7中所述，与非门单元中的时序弧为负单边类型（negative unate），这表示输出引脚电平跳变方向与输入引脚电平跳变方向是相反的。因此，查找**cell\_rise**表对应于输入引脚上的下降过渡时间。同样，通过与门单元或者或门单元的时序弧是正单边类型（positive unate）的，因为输出引脚电平跳变方向与输入引脚电平跳变方向相同。

**3.3.2 通用组合逻辑块**

考虑以下这个具有三输入和两输出的通用组合逻辑块（General Combinational Block）：

图3-7

像这样的组合逻辑块可以具有多个时序弧。通常，从块的每个输入到每个输出都有一条时序弧。如果从输入到输出的逻辑路径是同相（non-inverting）或正单边类型的，则输出的极性（polarity）与输入的极性相同；如果逻辑路径是反相（inverting）或负单边类型，则输出的极性与输入相反，即当输入电平上升时，输出电平将下降。这些时序弧代表了通过这个组合逻辑块的传播延迟。

通过组合逻辑单元的某些时序弧可以既是正单边类型，也是负单边类型的。一个例子是通过两输入异或门（xor）单元的时序弧，根据该单元的另一个输入的逻辑状态，两输入异或门单元的输入引脚处的电平跳变会导致输出引脚处的电平沿相同或相反的方向跳变。这些时序弧可以描述为非单边类型（non-unate）的，也可以描述为两组状态相关（state-dependent）的正单边类型时序模型和负单边类型时序模型。这种与状态相关的表格模型将在3.5节中有更详细的描述。

第三章： 标准单元库 （未完待续......）