**静态时序分析圣经翻译计划——第三章：标准单元库 （中）**

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

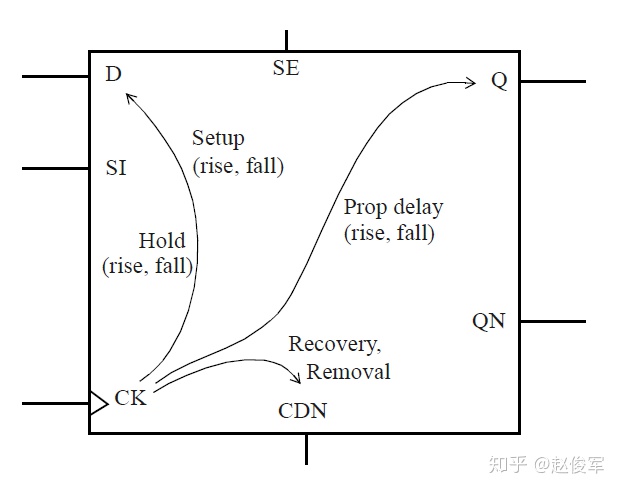
[**​**](https://www.zhihu.com/question/48510028)

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

**3.4 时序模型-时序逻辑单元**

考虑如下图3-8所示的时序逻辑单元的时序弧：

图3-8

对于同步（synchronous）输入，例如引脚D（或SI，SE），存在以下时序弧：

● 建立（Setup）时间检查时序弧（包括上升沿和下降沿）

● 保持（Hold）时间检查时序弧（包括上升沿和下降沿）

对于异步（asynchronous）输入，例如引脚CDN，存在以下时序弧：

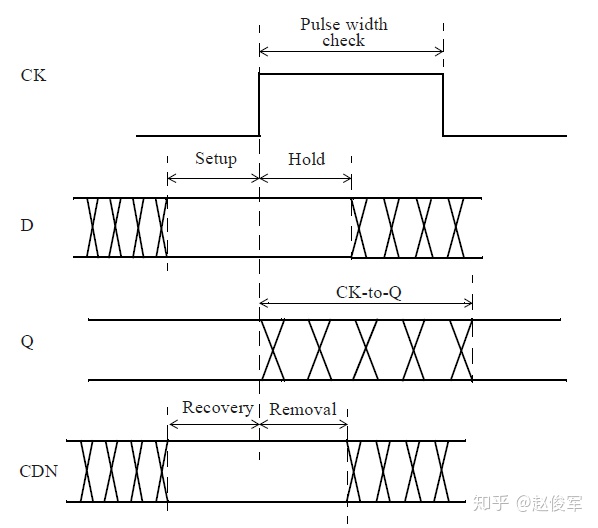
● 恢复（Recovery）时间检查时序弧

● 撤销（Removal）时间检查时序弧

对于触发器的同步输出，例如引脚Q或QN，存在以下时序弧：

● 时钟引脚（CK）到输出端口（Q或QN）的传播延迟时序弧（包括上升沿和下降沿）

所有同步时序弧均相对于时钟的有效沿（active edge），即时序逻辑单元捕获数据的时钟沿。此外，时钟引脚和异步引脚（如复位引脚）还需要进行脉冲宽度（pulse width）时序检查。各种时序检查如下图3-9所示：

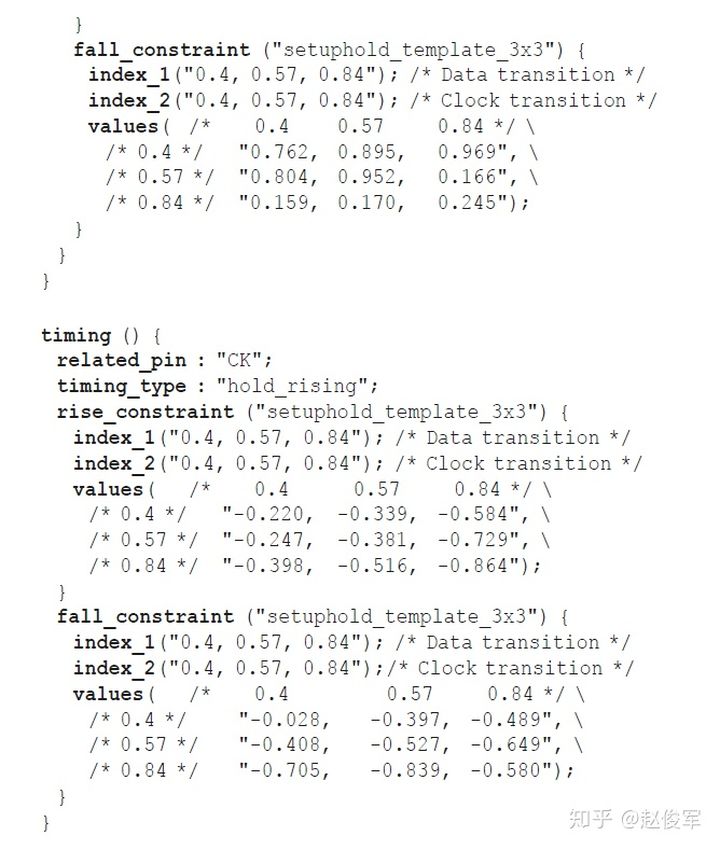
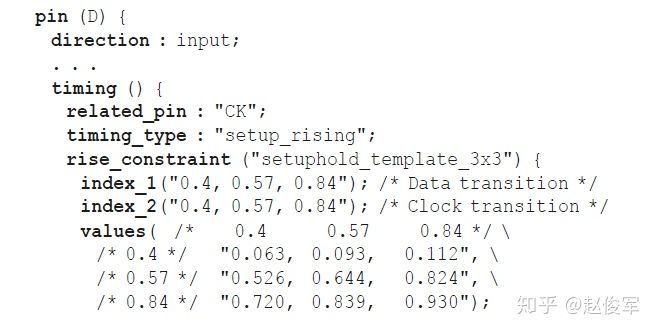
图3-9

**3.4.1 同步时序检查：建立时间与保持时间**

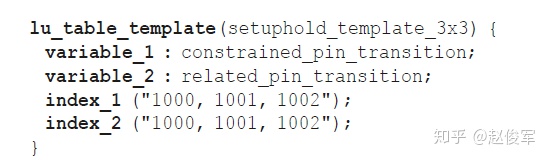
需要进行建立时间和保持时间的同步时序检查，才能确保数据能够正确通过时序单元传播。这些时序检查可验证输入的数据在时钟有效沿上是否为确定的逻辑状态，并且在有效沿上将正确的数据锁存下来。这些时序检查也可以验证输入的数据在有效时钟沿附近是否稳定：输入的数据必须在有效时钟沿之前保持稳定的最短时间称为建立时间（setup time）。这是根据最晚的（the latest）数据信号超过其阈值（通常为Vdd的50％）到有效时钟沿超过其阈值（通常为Vdd的50％）的时间间隔测量的。同样，保持时间（hold time）是在时钟有效沿之后输入的数据必须保持稳定的最短时间，这也是根据有效时钟沿超过其阈值到最早的（the earliest）数据信号超过其阈值的时间间隔来度量的。如前文所述，时序单元的时钟有效沿是指使时序单元捕获数据的上升沿或下降沿。

**建立时间与保持时间检查示例**

时序单元同步引脚的建立时间和保持时间约束通常用二维表格来描述，如下所示。下面的例子展示了触发器数据引脚的建立时间和保持时间时序信息：



上面的例子展示了相对于时序单元时钟引脚CK的上升沿，在输入引脚D上的建立时间和保持时间约束。二维表格模型是根据约束引脚**constrained\_pin**（D）和相关引脚**related\_pin**（CK）处的过渡时间确定的，二维表格的查找是基于库中描述的模板**setuphold\_template\_3x3**的。对于上述示例，查找表模板**setuphold\_template\_3x3**如下所示：



与前面示例中一样，表中的值如嵌套循环一样被指定，其中第一个索引**index\_1**是外部循环变量，第二个索引**index\_2**是内部循环变量， 以此类推。因此，当D引脚的上升沿过渡时间为0.4ns，CK引脚的上升沿过渡时间为0.84ns时，D引脚上升沿的建立时间约束为0.112ns，该值是从**rise\_constraint**表中读取的。对于D引脚的下降沿，建立时间约束将检查建立时间表中的**fall\_constraint**表。在表中查找过渡时间不与索引值相对应的建立时间和保持时间约束时，可使用章节3.2中所描述的非线性模型查找的一般过程。

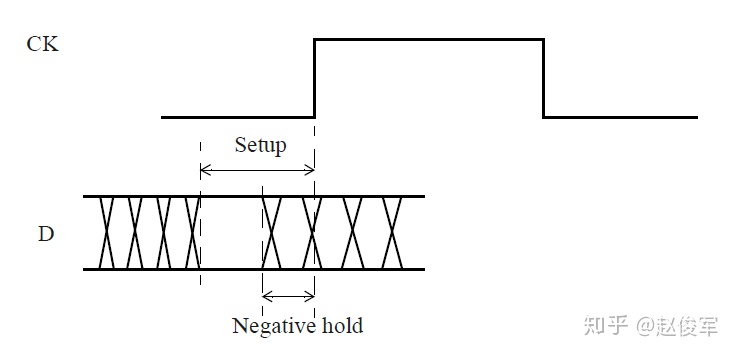
请注意，建立时间约束的**rise\_constraint**和**fall\_constraint**表是指**constrained\_pin**。所用的时钟过渡时间由**Timing\_type**决定，它会指定该时序逻辑单元是上升沿触发还是下降沿触发。

**建立时间和保持时间检查里的负值**

请注意，上面示例中的某些保持时间值是负数。这是可以接受的，这种情况通常发生在触发器的引脚到数据内部锁存点的路径长于相应的时钟路径时。因此，负的保持时间检查意味着触发器的数据引脚可以在时钟引脚之前改变，并且仍然满足保持时间的检查要求。

触发器的建立时间值也可以为负。这意味着触发器的数据引脚可以在时钟引脚之后改变，并且仍然满足建立时间的检查要求。

建立时间和保持时间可以都为负吗？不行！为使建立时间和保持时间检查相一致，建立时间和保持时间的数值总和应为正。因此，如果建立（或保持）时间检查包含负值，则相应的保持（或建立）时间检查应为正值，且足够大，以使建立时间加上保持时间之和是一个正数。 具有负的保持时间值的示例如图3-10所示，由于建立时间检查会先于保持时间检查，所以此时的建立时间加上保持时间显然是一个正数。建立时间加上保持时间即为要求数据信号稳定的区域的宽度。

图3-10

对于触发器，在扫描数据（scan data）输入引脚上设置一个负的保持时间会很有帮助，这样可以给予时钟偏斜（clock skew）一定的灵活性，并且可以消除当扫描模式（scan mode）下保持时间发生违例时插入缓冲器（buffer）的需要（扫描模式是将触发器串联在一起形成一条扫描链的模式，触发器的输出通常连接到串联的下一个触发器的扫描数据输入引脚，这些连接是用于芯片测试的）。

与同步数据输入引脚上的建立时间或保持时间检查类似，对异步引脚进行的时序约束检查将在下个小节中介绍。

**3.4.2 异步时序检查**

**恢复时间和撤销时间检查**

诸如异步复位（clear）或异步置位（set）之类的异步引脚会覆盖单元的任何同步行为。当异步引脚处于有效（active）状态时，输出将由异步引脚控制，而不是由时钟锁存到的输入数据控制。但是，当异步引脚变为无效（inactive）状态时，时钟的有效边沿将开始锁存输入的数据。异步恢复（recovery）时间和撤销（removal）时间约束检查将验证异步引脚在下一个时钟有效沿处是否已明确返回到了无效状态。

恢复时间是指异步输入被置为无效（de-asserted）后在下一个时钟有效沿之前需要保持稳定的最短时间。

同样，撤销时间是指在一个有效的时钟沿之后，异步引脚必须保持有效状态的最小时间，即在有效时钟沿之后必须经过一段撤销时间才能够将异步引脚置为无效状态。

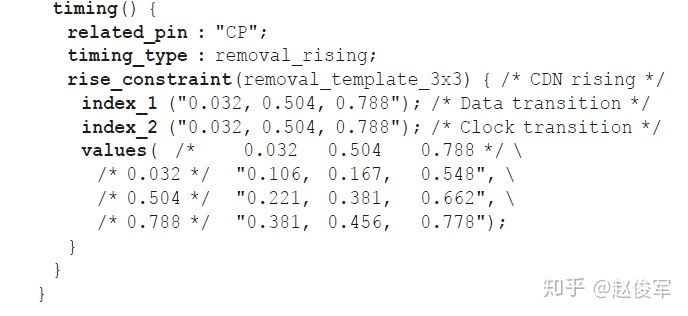
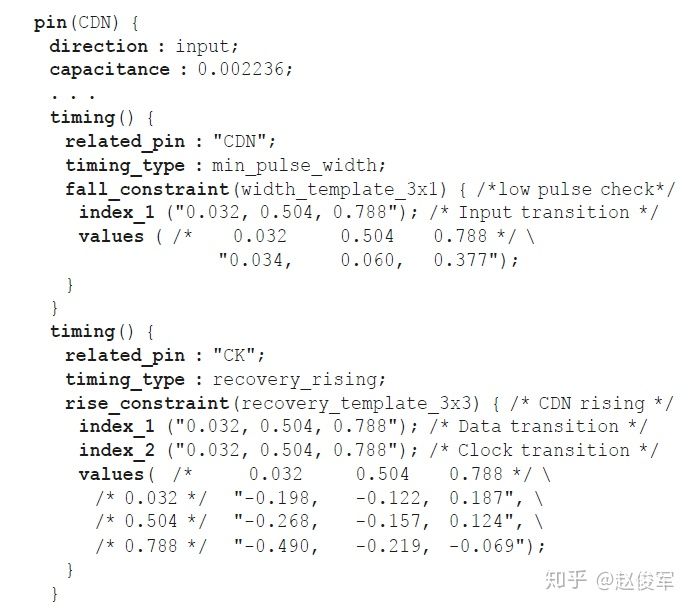
异步恢复时间和撤销时间检查将分别在8.6节和8.7节中介绍。

**脉冲宽度检查**

除了同步和异步时序检查外，还有一项检查去确保逻辑单元输入引脚上的脉冲宽度满足最低要求。例如，如果时钟引脚上的脉冲宽度小于指定的最小值，则时钟可能无法正确锁存数据。也可以为相关的同步和异步引脚指定脉冲宽度检查（pulse width check），并且可以为高脉冲和低脉冲分别指定最小脉冲宽度检查。

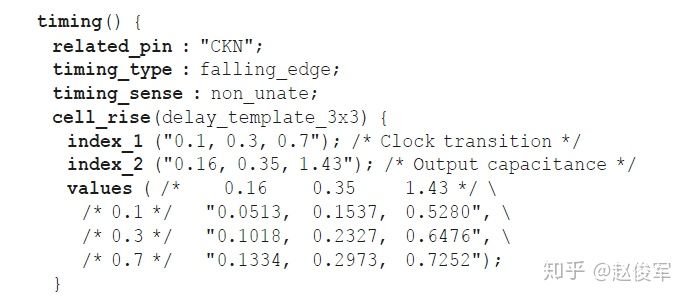
**恢复时间、撤销时间和脉冲宽度检查示例**

下面给出了触发器的异步清零引脚CDN的恢复时间、撤销时间和脉冲宽度检查的示例，其中恢复时间和撤销时间检查还与时钟引脚CK有关。由于恢复时间和撤销时间检查是针对异步引脚被置为无效的时刻，因此在以下示例中仅存在上升约束。CDN引脚的最小脉冲宽度检查用于低脉冲，这是由于CDN引脚为低电平有效，因此该引脚上的高脉冲宽度没有限制，因此未指定。

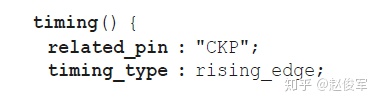


**3.4.3 传播延迟**

时序单元的传播延迟（propagation delay）是指从时钟的有效沿到输出的上升沿或下降沿。以下是一个时钟下降沿触发的触发器从时钟引脚CKN到输出Q的传播延迟弧的示例。这是一个非单边类型（non-unate）的时序弧，因为时钟的有效沿会导致输出引脚Q的电平上升或下降。



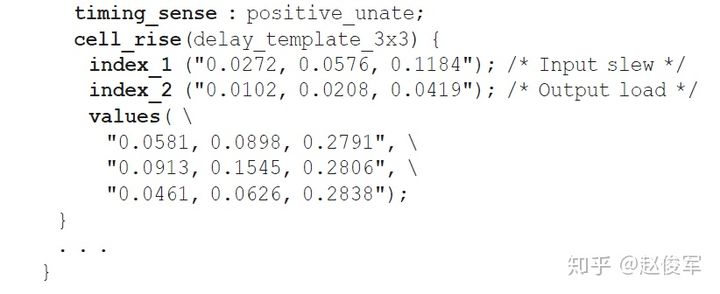
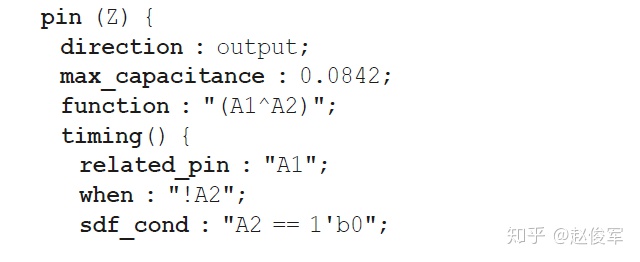
与前一章中的示例一样，输出的传播延迟会根据输入过渡时间和输出负载电容以二维表格的形式呈现。但是在此示例中，由于触发器是时钟下降沿触发的，因此要使用的输入过渡时间是CKN引脚的下降过渡时间，这在上面的示例中可以从**Timing\_type**字段中看出，上升沿触发的触发器将把**rising\_edge**指定为其**timing\_type**。



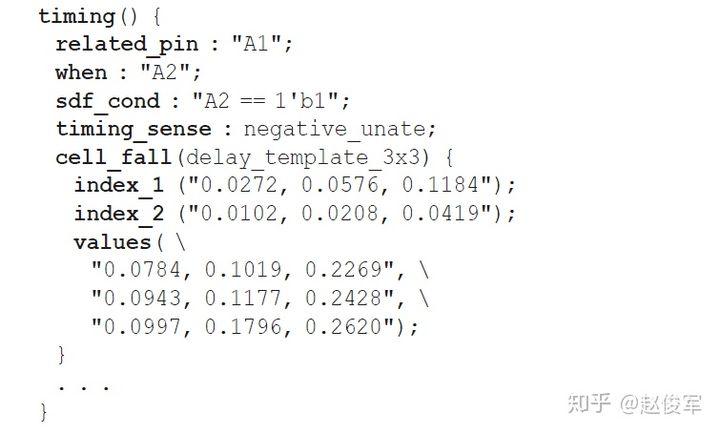
**3.5 状态相关的时序模型**

在许多组合逻辑模块中，输入和输出之间的时序弧取决于模块中其他引脚的逻辑状态。输入和输出引脚之间的这些时序弧可以是正单边类型时序弧、负单边类型时序弧也可以既是正单边类型又是负单边类型时序弧。异或门（xor）以及同或门（xnor）单元就是一个例子，其输入到输出的时序弧可以是正单边类型的也可以是负单边类型的。 在这种情况下，其时序行为可能会有所不同，具体取决于该模块其他输入的状态。通常，取决于引脚状态的多个时序模型被称为状态相关的时序模型（state-dependent models）。

考虑一个两输入异或门单元的例子。当另一个输入A2为逻辑0时，从输入A1到输出Z的时序路径为正单边类型；当输入A2为逻辑1时，从A1到Z的路径为负单边类型。这两个时序模型是使用状态相关的时序模型指定的。A2为逻辑0时从A1到Z的时序模型如下所示：

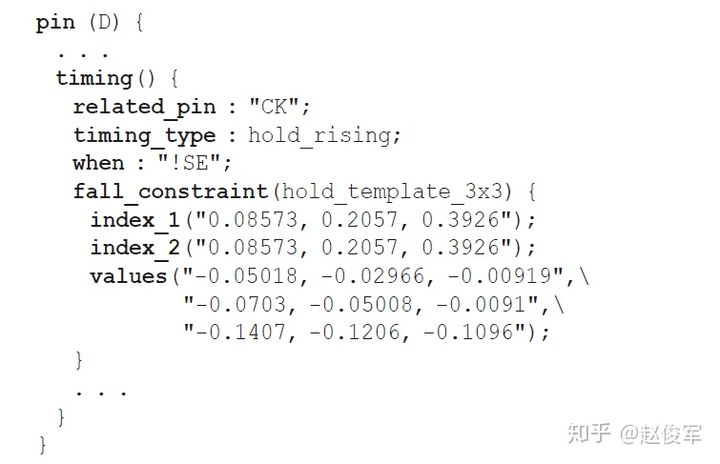


使用**when**来指定与状态有关的条件。虽然以上单元模型的片段仅说明了**cell\_rise**延迟，但其它时序模型（**cell\_fall**，**rise\_transition**和**fall\_transition**）也指定了相同的**when**条件。当**when**中条件改变，如改为A2为逻辑1这个条件时，会指定另一个独立的时序模型：



**sdf\_cond**用于指定生成**SDF**文件时要使用的时序弧的条件，详情参见后面3.9节中的示例。

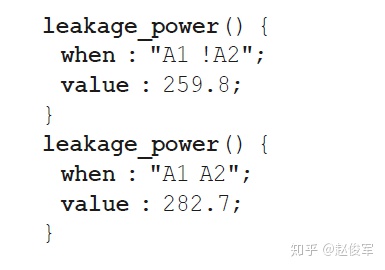
状态相关的时序模型可用于各种类型的时序弧，许多时序逻辑单元会使用状态相关的时序模型指定建立时间或保持时间的时序约束。以下是使用状态相关的时序模型进行保持时间约束的扫描触发器（scan flip-flop）的一个示例。在这种情况下，指定了两组时序模型：扫描使能引脚SE处于有效状态时的一组，以及扫描使能引脚处于无效状态时的另一组。



当扫描使能引脚SE为逻辑0时使用以上时序模型，当**when**中条件为SE是高电平时也有类似的时序模型。

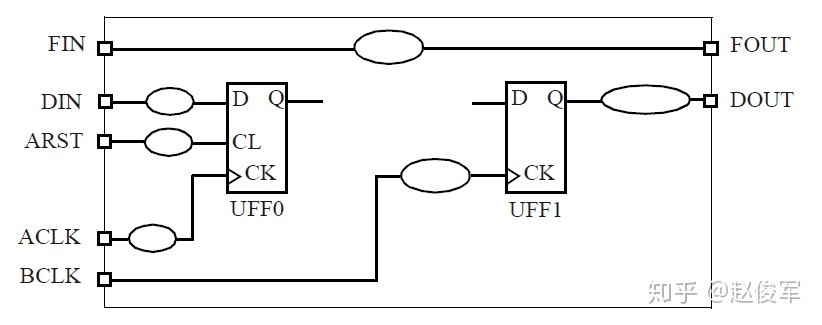
可以既使用状态相关的时序模型又使用非状态相关的时序模型来指定一些时序关系。在这种情况下，如果单元的状态是已知的并且这个状态包含在状态相关的时序模型之一中，则时序分析将使用状态相关的时序模型。而如果状态相关的时序模型未包含单元的这个状态，则使用来自非状态相关模型的时序。例如考虑一种情况，保持时间约束仅由当**when**条件为SE处于逻辑0时来指定，而没有为SE处于逻辑1时指定单独的状态相关模型。在这种情况下，如果SE被置为逻辑1，则将会使用非状态相关时序模型的保持时间约束。而如果没有用于保持时间约束的非状态相关时序模型，那么将没有任何有效的保持时间约束！

可以为时序库中的任何属性指定状态相关的模型，因此可以存在功率、漏电功率、过渡时间、上升和下降延迟、时序约束等状态相关的模型。下面给出了一个状态相关的漏电功率模型示例：



**3.6 黑盒的接口时序模型**

本节将介绍黑盒（任意模块或块）的IO接口时序弧，时序模型将捕获黑盒（black box）IO接口（interface）的时序。黑盒的接口时序模型可以具有组合逻辑弧，也可以有时序逻辑弧。通常，这些时序弧也可能取决于状态。

图3-11

对于如图3-11所示的例子，可以将时序弧分为以下几类：

● 输入到输出的组合逻辑弧：这对应于直接从输入到输出的组合逻辑路径，例如从输入端口FIN到输出端口FOUT。

● 输入时序逻辑弧：这对应于连接到触发器D引脚的输入端口的建立时间或保持时间。通常，在将模块的输入端口连接到触发器的D引脚之前，可以存在一些组合逻辑。这样的一个例子是在输入端口DIN上相对于时钟端口ACLK的建立时间检查。

● 输出时序逻辑弧：这类似于触发器时钟端到输出端Q的传播延迟。通常，触发器输出引脚与模块输出端口之间可以存在一些组合逻辑。一个示例是从时钟BCLK到触发器UFF1的输出引脚再到输出端口DOUT的路径。

● 异步输入时序弧：这类似于触发器异步输入引脚的恢复时间和撤销时间时序约束，例如输入端口ARST到触发器UFF0的异步清零引脚。

除上述时序弧外，在黑盒的外部时钟引脚上还可以进行脉冲宽度检查。还可以定义内部节点（node）并在这些内部节点上定义衍生时钟（generated clock），从而指定在这些节点之间的时序弧。总之，黑盒模型可以具有以下时序弧：

● 纯组合逻辑路径的输入到输出时序弧（Input to output timing arcs）

● 从同步输入端口到相关时钟端口的建立时间和保持时间时序弧（Setup and hold timing arcs）

● 从异步输入端口到相关时钟端口的恢复时间和撤销时间时序弧（Recovery and removal timing arcs）

● 从时钟端口到输出端口的输出传播延迟（Output propagation delay）

如上所述的接口时序模型并非旨在捕获黑盒的内部时序，而只是捕获其接口上的时序。

第三章： 标准单元库 （未完待续......）