

# User Constraint Editor

## 用户手册

(Version 1.7)

深圳市紫光同创电子有限公司

版权所有 侵权必究

## 文档版本修订记录

版本号	发布日期	修订记录
V1.0	2022.7.12	初始版本
V1.1	2022.8.5	增加 properties 窗口功能描述
V1.2	2022.8.5	增加查看 Schematic 功能描述
V1.3	2022.8.19	增加 select leaf cells 功能描述
V1.4	2022.8.19	增加 show connectivity 功能描述
V1.5	2022.8.19	增加 Bundle nets 功能描述
V1.6	2022.8.19	增加 Selection 窗口功能描述
V1.7	2022.10.26	增加 Region 资源过滤功能描述

## 目录

<b>UCE 界面功能介绍.....</b>	<b>9</b>
1.1 启动 UCE.....	9
1.2 不同类型工程的 UCE.....	10
1.3 UCE 主界面功能.....	11
1.3.1 右键菜单.....	12
1.3.2 UCE 的搜索工具.....	13
1.3.3 UCE 的按钮.....	15
1.3.4 Timing Constraints 界面.....	16
1.3.5 Attribute 表格.....	37
1.3.6 Compile Points 界面.....	41
1.3.7 Device 界面.....	42
1.3.8 Schematic 界面.....	62
1.3.9 Hierarchy 界面.....	63
1.3.10 Selection 界面.....	63
1.3.11 Properties 界面.....	64
1.4 解析差异造成的 FDC 修改.....	66
<b>2.UCE 约束命令介绍.....</b>	<b>68</b>
2.1 对象描述规则.....	68
2.2 时序约束.....	72
2.2.1 Clocks 命令.....	72
2.2.2 Inputs/outputs 命令.....	79
2.2.3 Assertio 命令.....	81
2.2.4 Exception 命令.....	82
2.2.5 Other 约束命令.....	87
2.3 逻辑约束.....	88
2.3.1 define_attribute 命令.....	88
2.3.2 define_global_attribute 命令.....	89
2.3.3 define_compile_points 命令.....	90
2.3.4 UCE 中属性的设置.....	90

---

2.3.5 Device 中的 I/O Table 相关配置.....	98
2.3.6 IP FDC 支持指定子 module.....	103
2.3.7 Fdc 约束文件逻辑约束支持通配符.....	103
2.3.8 UCE 属性使用实例.....	103
2.4 查询命令.....	105
2.4.1 get_cells.....	105
2.4.2 get_nets.....	106
2.4.3 get_pins.....	106
2.4.4 get_clocks.....	107
2.4.5 get_ports.....	107
2.5 常见约束错误.....	107
2.6 通用 IO 属性默认值.....	108
2.6.1 Compact(PGC2KG).....	108
2.6.2 Logos(PGL50H).....	109
2.6.3 Logos2(PG2L100H).....	112
2.6.4 Titan(PGT180H).....	115
2.6.5 Titan2(PG2T390H).....	117

## 图目录

图 1-1 UCE 主界面.....	9
图 1-2 工具栏 User Constraint Editor(Timing and Logic)图标.....	9
图 1-3 Tools 下的 User Constraint Editor(Timing and Logic).....	10
图 1-4 UCE 全量配置界面.....	12
图 1-5 UCE 右键菜单.....	13
图 1-6 IO table 右键.....	13
图 1-7 UCE 的 search 工具.....	13
图 1-8 UCE 的网表搜索工具.....	14
图 1-9 Pre Synthesize UCE 的按钮.....	15
图 1-10 Design Browser 的结构.....	15
图 1-11 Timing Constraints 表格.....	16
图 1-12 Create Clock 示例.....	17
图 1-13 Create Clock 窗口.....	18
图 1-14 Ceate Clock 表格.....	18
图 1-15 Create Generated Clock 示例.....	19
图 1-16 Ceate Generated Clock 窗口.....	19
图 1-17 Create Generated Clock 表格.....	20
图 1-18 Create Generated Clock 参数.....	20
图 1-19 Set Clock Latency 窗口.....	22
图 1-20 Set Clock Latency 表格.....	22
图 1-21 Set Clock Uncertainty 窗口.....	23
图 1-22 Set Clock Uncertainty 表格.....	23
图 1-23 Set Clock Groups 示例.....	24
图 1-24 Set Clock Groups 窗口.....	25
图 1-25 Set Clock Groups 表格.....	25
图 1-26 set_external_delay 窗口.....	26
图 1-27 set_external_delay 表格.....	26
图 1-28 Set Input Delay 示例.....	27

---

图 1-29 Set Input Delay 窗口.....	28
图 1-30 Set Input Delay 表格.....	28
图 1-31 Set Max Skew 窗口.....	29
图 1-32 Set Max Skew 表格.....	30
图 1-33 Set Max Delay 窗口.....	31
图 1-34 Set Max Delay 表格.....	31
图 1-35 Set Min Delay 窗口.....	32
图 1-36 Set Min Delay 表格.....	32
图 1-37 Set Multicycle Path 窗口.....	33
图 1-38 Set Multicycle Path 表格.....	34
图 1-39 Set False Path 窗口.....	35
图 1-40 Set False Path 表格.....	35
图 1-41 Set_disable_timing 窗口.....	36
图 1-42 set_disable_timing 表格.....	36
图 1-43 Attribute 表格.....	37
图 1-44 Attributes Table 窗口.....	37
图 1-45 Specify Object 窗口.....	38
图 1-46 Attributes table 中的 attribute 行.....	39
图 1-47 Attributes table 中的 Value 行.....	39
图 1-48 Attributes table 中的 command 栏.....	39
图 1-49 保存的逻辑约束.....	40
图 1-50 新增空白行.....	40
图 1-51 Attribute 表格的 Object 列.....	40
图 1-52 Attribute 表格的 Attribute 列.....	41
图 1-53 Attribute 表格的 value 相关列.....	41
图 1-54 Compile Points 表格.....	42
图 1-55 Device 界面.....	42
图 1-56 Device 界面工具栏.....	43
图 1-57 Device Broswer 界面 Design 窗口.....	44

---

图 1-58 Edit Color 功能.....	45
图 1-59 IO Table 表格.....	45
图 1-60 搜索框.....	46
图 1-61 Device 查看实例.....	46
图 1-62 IO 约束.....	47
图 1-63 区域约束.....	49
图 1-64 新建 region.....	49
图 1-65 区域约束资源弹窗.....	50
图 1-66 区域约束取消勾选结果.....	51
图 1-67 查看实例区域约束.....	52
图 1-68 region 中 General 栏属性.....	52
图 1-69 region 中 Statistics 栏属性.....	53
图 1-70 region 中 Instances 栏属性.....	53
图 1-71 preserve 约束.....	54
图 1-72 exclusive region 约束.....	54
图 1-73 clock region 显示.....	55
图 1-74 选中对象.....	56
图 1-75 第一次 show connectivity.....	56
图 1-76 扇出效果.....	56
图 1-77 show cell connections 窗口.....	56
图 1-78 bundle nets 显示.....	57
图 1-79 bundle net 选中.....	57
图 1-80 bundle net 设置.....	57
图 1-81 Select Leaf Cells.....	58
图 1-82 HighLight Selection.....	59
图 1-83 Show Highlight List.....	59
图 1-84 选中 device 右键.....	60
图 1-85 mark 效果.....	60
图 1-86 mark browser.....	61

---

图 1-87 mark 浏览器界面.....	61
图 1-88 Select 功能.....	62
图 1-89 schametic 窗口.....	62
图 1-90 Schematic 选项.....	62
图 1-91 Show Hierarchy.....	63
图 1-92 Hierchay 界面.....	63
图 1-93 selection 按钮.....	64
图 1-94 selection 窗口显示.....	64
图 1-95 Selection 窗口右键功能.....	64
图 1-96 Properties 按钮.....	65
图 1-97 Properties 界面显示.....	65
图 2-1 Attribute 表格.....	88
图 2-2 I\O Table 表格.....	98
图 2-3 具体操作.....	101

## 表目录

表 1-1 Properties 属性显示.....	66
表 2-1 属性.....	99
表 2-2 PAP_IO_PREEMP、PAP_IO_OUTPUT_IMPEDANCE 作用.....	101
表 2-3 PGT180H 的 IOB 驱动器 P 管/N 管 finger 数量的电流对应表.....	103
表 2-4 Compact(PGC2KG).....	109
表 2-5 Logos(PGL50H).....	112
表 2-6 Logos2(PG2L100H).....	115
表 2-7 Titan(PGT180H).....	117
表 2-8 Titan2(PG2T390H) hrlio.....	120
表 2-9 Titan2(PG2T390H) hpio.....	126

## 1.UCE 界面功能介绍

该文档详细描述了 User Constraint Editor (以后简称 UCE) 的使用方法和各项功能。UCE 能够对实例和 IO 进行时序约束和逻辑约束。UCE 主界面, 如下图所示。

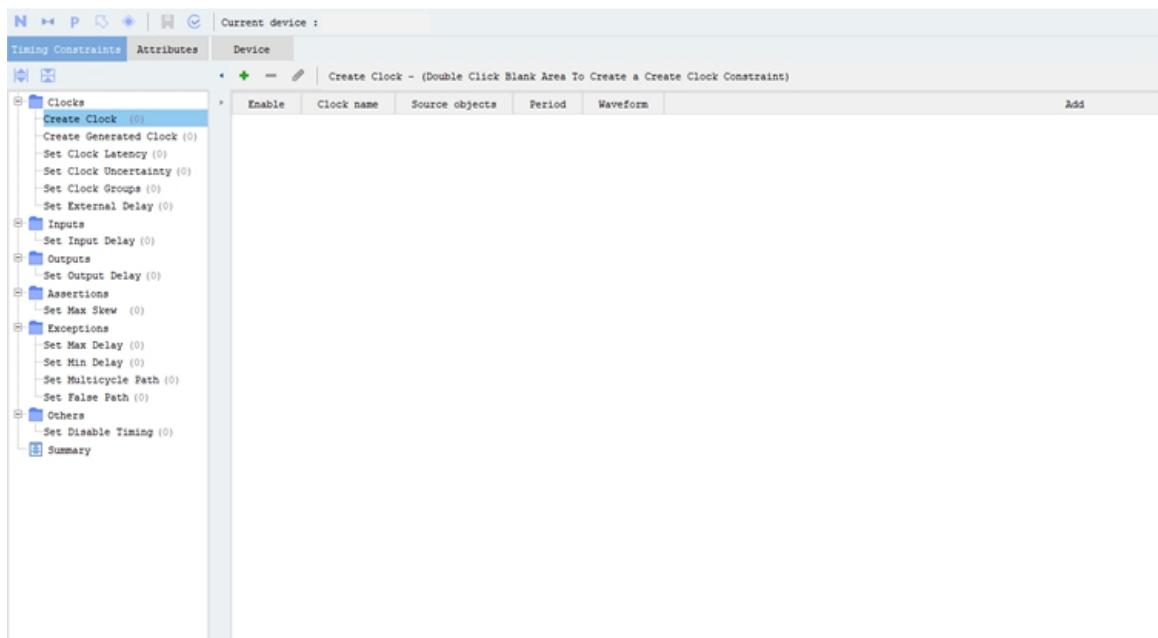


图 1-1 UCE 主界面

### 1.1 启动 UCE

打开软件, 新建工程, Synthesis Tool 选择 ADS, 选择添加 add.v; 或者不使用综合工具, 选择添加 add.vm, 然后在新建的工程里, 可采用以下三种方式启动 UCE:

1、点击工具栏图标 User Constraint Editor(Timing and Logic), 如下图所示。点击按钮后二级菜单会有两个选项: Pre Synthesize UCE(综合之前打开约束编辑器)、Post Synthesize UCE(综合之后打开约束编辑器)。根据选择进行打开。

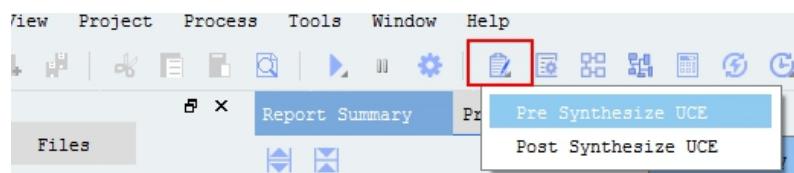


图 1-2 工具栏 User Constraint Editor(Timing and Logic)图标

2、点击菜单栏“Tools”下的“User Constraint Editor (Timing and Logic)”，如图所示。

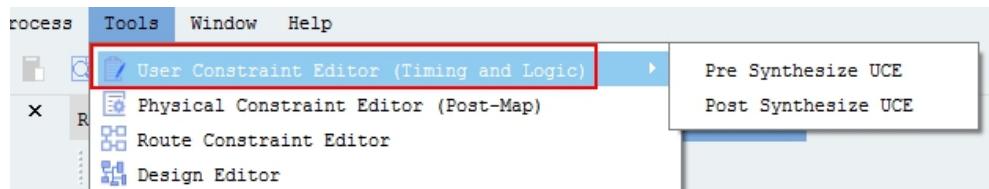


图 1-3 Tools 下的 User Constraint Editor(Timing and Logic)

3、当已经在工程中添加了 fdc 约束文件时，文件会显示在 PDS 主界面的“Navigator”窗口的“Sources”标签页下面的“Constraints”目录下面，双击 fdc 约束文件，则可以以文本编辑器的方式打开约束文件。

目前该目录下，支持添加多个 fdc 文件，均显示在 Constraints 目录下方。右键功能 Set As Target File, target file 将会保存 UCE 中新增的约束，切换 target file，则切换保存新增约束的约束文件。未修改的约束，将保存在原约束文件中。保存时，如果没有指定 target file，则会弹窗提醒用户选择 target file。

目前只支持初步简单的多约束文件，即多个约束文件的内容彼此之间独立（如时序约束、IO 约束等等放在不同的约束文件中）。若多约束文件中的约束信息有关联，有可能会造成先使用后定义的情况出现，导致约束检查报错或者约束被覆盖没有生效等等。目前 Constraints 层级下，多约束文件按照显示的顺序进行读取，即顺序读取。若有约束文件中的约束是其他约束文件中约束信息的前置条件，请将该约束文件显示顺序调至靠前。

## 1.2 不同类型工程的 UCE

根据是否有综合工具，以及综合工具的类型，对应的 UCE 及后续约束的流程也有所不同，具体的区别如下：

1、综合工具为 ADS 时： Pre Synthesize UCE 读写和编辑的是综合前的约束，即读取和保存的约束文件为 fdc (FPGA Design Constraint) 文件。保存的 fdc 约束文件会在综合时使用。Post Synthesize UCE 读写和编辑的是综合后的约束，即读取和保存的约束文件为 lcf (Logical Constraint File) 文件和 scf (Synthesis Constraint File) 文件。保存的 lcf 和 scf 约束文件会在 MAP 时使用。

2、综合工具为 OEM 时：Pre Synthesize UCE 读写和编辑的是综合前的约束，即读取和保存的约束文件为 fdc 文件。保存的 fdc 约束文件会在综合时使用。综合后 OEM 会生成两个约束文件 lcf 和 scf，作为 MAP 一步的输入文件。(目前不支持综合工具为 OEM 时，打开 Post Synthesize UCE)。

3、没有使用综合工具时：Post Synthesize UCE 读写的约束文件为 sdc (即 Synthesis Design Constraint，里面包含 timing 约束信息)，和 lcf (逻辑约束信息，目前为 attribute 信息)；

### 1.3 UCE 主界面功能

打开 UCE(针对 Pre Synthesize UCE 和 Post Synthesize UCE, 后续统称为 UCE)后，其主界面为一系列表格，这些表格分为不同的 Tab 项，这些表格分为 Timing 约束和 Logical 约束两类，其中 Timing Constraints Tab 为 Timing 约束相关，Attribute 和 Device 为 Logical 约束相关，Logical 约束在约束文件里皆以 attribute 命令的形式体现；Compile Points 表格用来设置 Complie Points(OEM 场景特有)。

UCE 的输入文件为用户设计文件、约束文件和器件信息。打开 UCE 时会经过约束检查后将正确的约束内容显示在 UCE 中。错误的约束将被过滤，不显示在 UCE 界面上。

在 UCE 界面上进行修改删除等操作后，可以通过【save constraints】按钮保存约束文件。

当前 UCE 界面约束保存 fdc 处理规则：

- (1) 新增约束以增量形式添加到 fdc 文件中。
- (2) 修改约束在原有行的基础上修改约束命令。
- (3) 删除约束将客户约束直接注释掉。
- (4) 原有约束文件中的注释需要保留。

在 PDS 主界面中 Edit 下拉选项的 Preferences 中，可以在 User Constraint Editor 选项中设置进入 Pre Synthesize UCE (综合工具为 OEM) 时，是全量解析网表还是只解析顶层端口，配置界面如下：

Generate all design netlist objects: 全量解析，解析整个网表，检查网表中所有约束的合法性；

Generate design ports only: 非全量解析，只解析顶层网表中的端口。由于没有解析网表，无法根据端口的类型来进行检查，只根据端口的方向和电平标准文件进行检查。

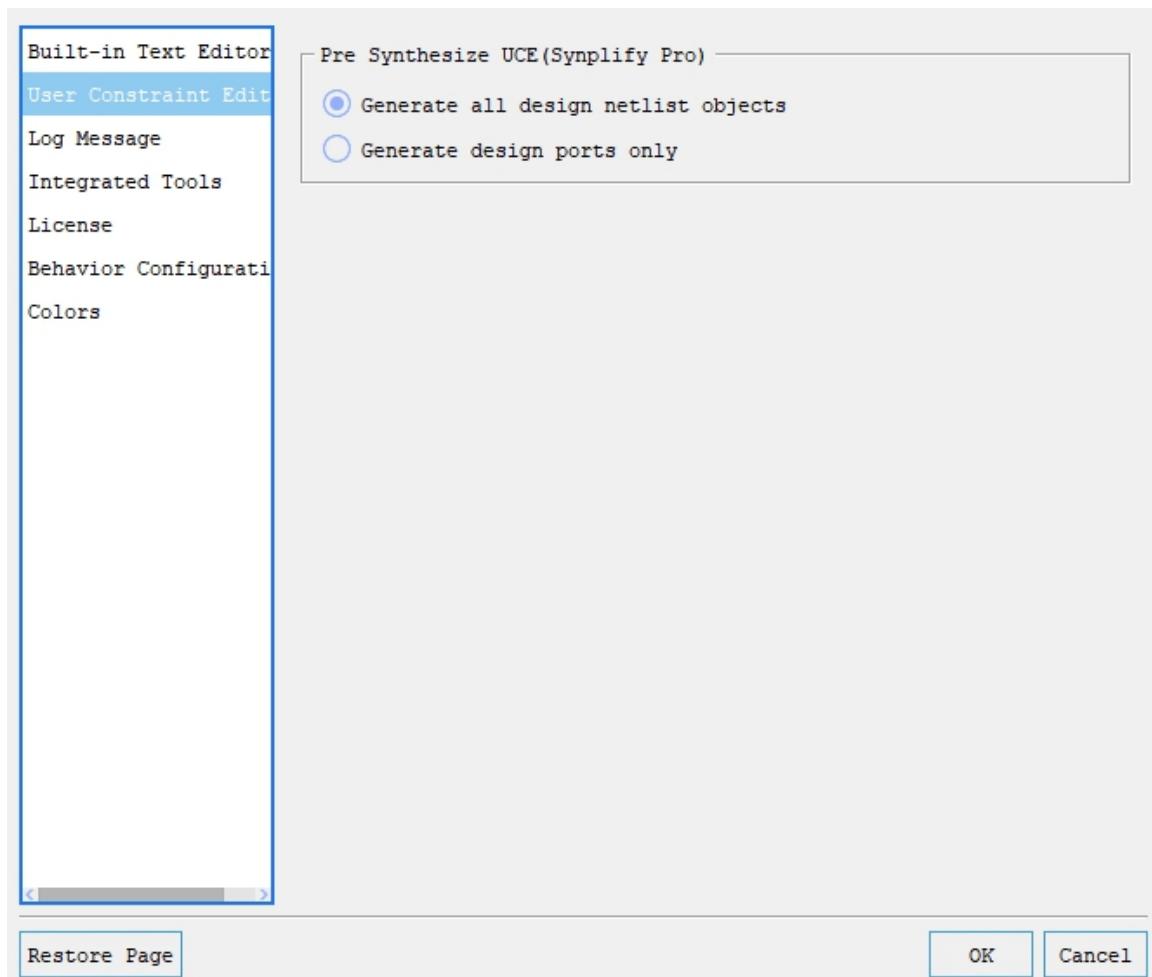


图 1-4 UCE 全量配置界面

### 1.3.1 右键菜单

在 UCE attribute 界面右键会弹出来一个菜单框，如下图所示，菜单内容分别有：Insert Row、Remove Row、Clear Constraint、Clear All Constraint，对应的是操作 UCE 表格的方法。

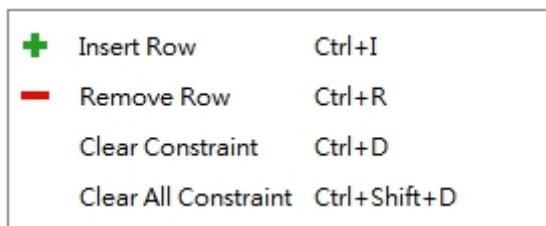


图 1-5 UCE 右键菜单

【Insert Row】：在该表中插入一个空白行；

【Remove Row】：删除光标所在的行；

【Clear Constraint】：清空光标所在行；

【Clear All Constraint】：清空该表的所有约束。

UCE 的 I/O table 界面右键弹出来通用操作的菜单框，如下图所示：

Clear Constraint	Ctrl+D
Clear All Constraint	Ctrl+Shift+D
Copy	Ctrl+C
Paste	Ctrl+V
Mark	▶
UnMark	

图 1-6 IO table 右键

【Clear Constraint】：清空光标所在行；

【Clear All Constraint】：清空该表的所有约束；

【Copy】：复制选择框内容；

【Paste】：粘贴；

【Mark】：对选中对象进行标记；

【Unmark】：取消标记；

【Schematic】：打开原理图；

### 1.3.2 UCE 的搜索工具

#### 1.3.2.1 UCE 界面 search 工具

UCE 中为 Attribute 表格和 Compile Points 表格提供了 search 工具来帮助搜索当前表格中的信息，使用”Ctrl + F”可以打开，位置处于 UCE 界面下方，如下图所示：

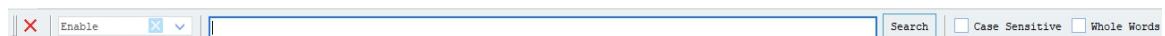


图 1-7 UCE 的 search 工具

其中第一个下拉列表用来选择将要搜索哪一个表格，第二个指定搜索的列，后边紧接着输入要搜索的内容，点击 search 就可以在 UCE 中定位到对应的行了。

### 1.3.2.2 UCE 的网表搜索工具

UCE 中为 Attribute 表格和 Timing Constraints 界面提供了网表搜索工具来帮助搜索当前网表中的内容，该搜索工具的入口统一表现为窗口或者表格中的按钮，在需要输入网表内容的地方点击即可。

该搜索工具的显示窗口，如下图所示：

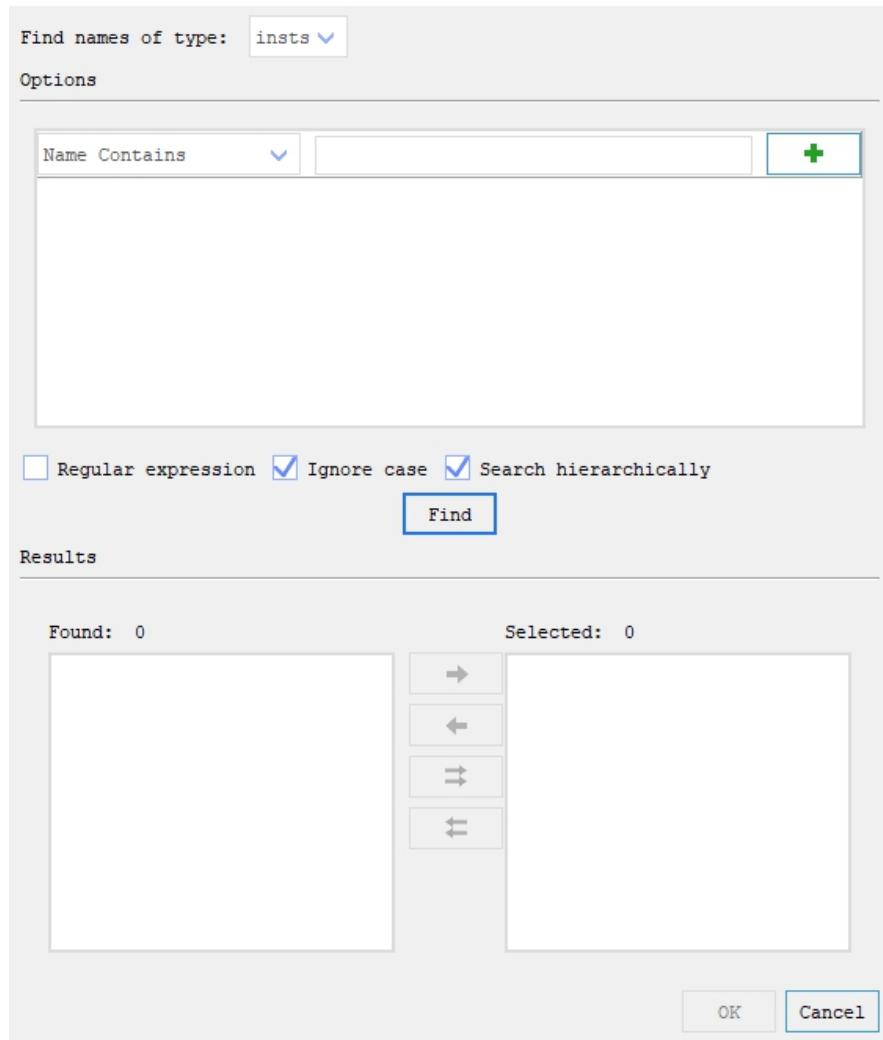


图 1-8 UCE 的网表搜索工具

该搜索工具分上下两部分，上半部分为搜索条件，下半部分为搜索结果。搜索条件基于该搜索工具入口处支持的搜索类型，并提供正则匹配/忽略大小写/搜索 hierarchy 结构等搜索方式；用户也可以增加对不同类型的不同匹配方式，比如：搜索 ports 时可以指定 direction 的类型为 in/out/inout 中的某种。下半部分中左侧为搜索结果的显示，右侧为用户挑选的搜索结果。

该搜索工具可以在网表中搜索 insts/ports/nets/pins/clocks 五种类型的信息。在 Attribute 和 Timing Constraints 界面中，均只显示入口处支持的 Object 类型。

### 1.3.3 UCE 的按钮

Pre Synthesize UCE 界面的上方有 5 个按钮，如下图所示：



图 1-9 Pre Synthesize UCE 的按钮

依次为【Design Browser】【Region】【Properties】【Selection】【Mark Browser】  
【Save】【Check Constraints】。

【Design Browser】按钮可以打开 UCE 的网表查看工具，通过该工具可以查看该阶段设计网表的结构和内容。网表的内容大致分为三类：Ports、Leaf Cells 和 Nets，由于 UCE 阶段会有 hierarchical 结构的，所以若有该种结构也会显示。

点击按钮以后会在 UCE 左侧显示树型结构，如下图所示。

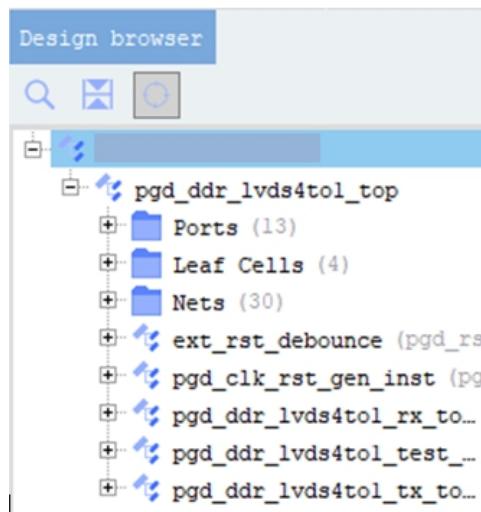


图 1-10 Design Browser 的结构

【Region】按钮可以打开 Region 窗口来进行修改、创建、修改 region。

【Properties】按钮可以打开 Properties 窗口来查看选中对象的属性。

【Selection】按钮用来打开 Selection 界面，可查看选中对象信息。

【Mark Browser】按钮用来打开 Mark Brower 列表，其中能够定位某个已 mark 的资源位置，并且能够将不需要 mark 的资源移除 mark browser。

【Save】按钮用来将当前 UCE 中填写的所有约束保存到约束文件 target file 中，不会过滤掉检查报告中的错误约束。所以，若想生成的约束文件全部正确，请先查看检查报告。

【Check Constraints】按钮用来检查 UCE 中所填写的约束的正确性，约束检

查报告 ccr 以文本框的形式，并列在 UCE Tab 栏。

**【Current device】** UCE 界面显示当前使用的器件信息。

接下来几个小节介绍 UCE 的各个窗口，UCE 表格里所填写的内容都对应于各自的约束命令的各个 option，下面在介绍时会指明其对应的命令的 option，为了方便描述表格内容与约束命令的对应关系，我们先将约束命令简要说明一下，对于约束命令的详细描述，在第二章的约束命令介绍里会有说明。

#### 1.3.4 Timing Constraints 界面

Timing Constraints 界面是处理 Timing 时序命令的界面，如下图所示。



The screenshot shows the 'Timing Constraints' interface. On the left is a tree view of timing commands under 'Clocks', 'Inputs', 'Outputs', 'Assertions', 'Exceptions', and 'Others'. Under 'Clocks', 'Create Clock (0)' is selected. On the right, a table for 'Create Clock' is displayed with columns: Enable, Clock name, Source objects, Period, and Waveform. A button 'Add' is at the bottom right of the table.

图 1-11 Timing Constraints 表格

Timing Constraints 界面分为左右两部分，左边的树型结构显示命令及命令的数量。树型结构可以通过树型结构上面的按钮来实现折叠或者展开。右侧表格显示每种命令的内容。每条命令对应着一个表格。

树型结构显示支持的时序命令，共六类：Clocks、Inputs、Outputs、Assertions、Exceptions、Others，一共 14 条时序命令。树型结构中的 Summary，是预览命令的界面。单击命令则切换表格，双击命令、表格空白处双击，是弹出创建命令窗口界面。双击已存在命令，打开修改该条命令的窗口。点击 Summary，查看 UCE 中已设置的全部 Timing 命令。Summary 上部分为从文件中读取进来的已保存的命令。unsaved\_constraint 标签中为在 UCE 中新创建的命令。保存以后命令状态则改变为已保存。

每条命令均有独立的创建命令窗口界面，界面中对命令的每个 option 均提供输入操作。点击 OK，若窗口界面的命令错误，则会弹出错误信息。只有命令正确才能通过窗口界面传递到表格中。

### 1.3.4.1 Create Clock 窗口及表格

Create Clock 窗口和对应的 `create_clock` 命令一般用于创建基准时钟，基准时钟一般是通过输入 port 输入用户设计的板上时钟；因为会被其他的时序约束命令所涉及，基准时钟需要首先被定义，且只能通过 `create_clock` 命令设置。

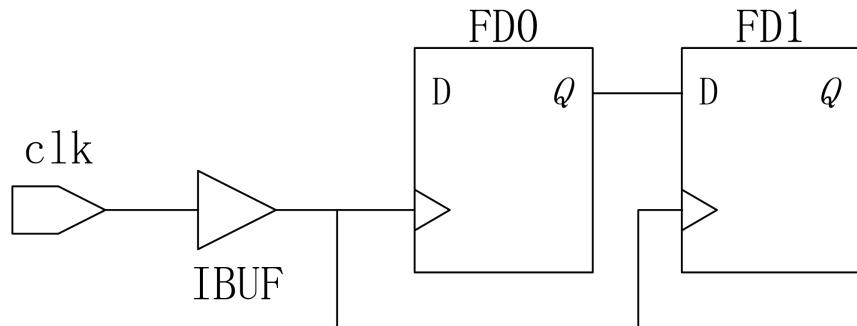


图 1-12 Create Clock 示例

如上图所示，板上时钟通过 `clk` port 进入器件，`clk` 时钟信号驱动 `FD0` 和 `FD1` 寄存器。为了保证从 `FD0/C` 到 `FD1/D` 路径的时序分析准确，需要在输入 port `clk` 处创建基准时钟。

Create Clock 窗口可创建 `create_clock` 命令，窗口上部为该条命令的简介；中间为创建该条命令的操作，`command` 一行可以显示创建的命令的形式以及参数；下部中点击 `Reset` 按钮则重置该界面。在输入 port `clk` 处创建名字为 `clk` 的基准时钟，对应窗口界面如下：

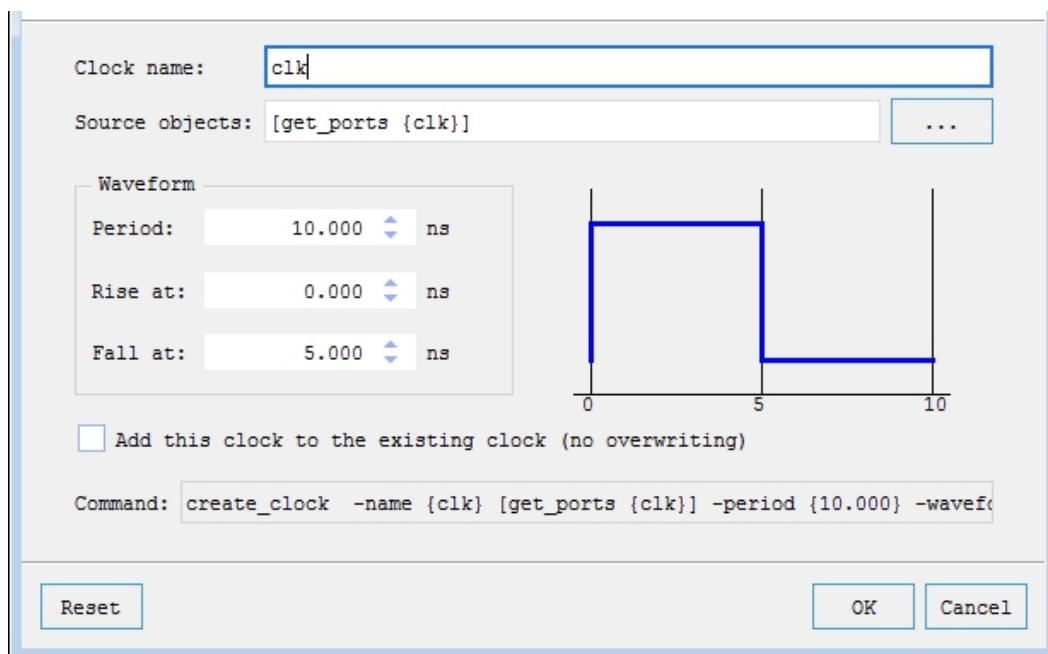


图 1-13 Create Clock 窗口

Create Clock 表格用来显示该条命令，表格如下：

Create Clock - (Double Click Blank Area To Create a Create Clock Constraint)					
	Enable	Clock name	Source objects	Period	Waveform
1	<input checked="" type="checkbox"/>	clk	[get_ports {clk}]	10.000	0.000 5.000

图 1-14 Create Clock 表格

其各列介绍如下：

**Enable 列：**该列是一个复选框，默认勾选，如果不勾选，则在保存约束文件时会在该条约最后加上-disable，除了 IO Table，其它表格都有 Enable，其用法是一样的，不再赘述；

**Clock name 列：**创建的时钟的名字，它对应于 create\_clock 命令的-name <clock\_name>参数；

**Source Objects 列：**对应 create\_clock 命令创建时钟的源，如 port, pin 等，必须通过 get\_pins、get\_ports 等命令获得，当其为空时，表示创建一个虚拟时钟；它对应于 create\_clock 命令的<source\_objects>参数；

**Period 列：**对应 create\_clock 命令所创建时钟的周期；它对应于 create\_clock 命令的-period <period\_value>参数；

**Waveform 列：**对应 create\_clock 命令所创建时钟上升沿和下降沿的位置；它对应于 create\_clock 命令的-waveform <edge\_list>参数；

**Add 列：**是否允许在同一个 source object 上定义多个时钟；它对应于 create\_clock 命令的-add 参数；

该表格所填写的值在保存约束文件时会体现为 create\_clock 命令。

#### 1.3.4.2 Create Generated Clock 窗口及表格

Create Generated Clock 窗口和对应的 create\_generated\_clock 命令用于创建生成时钟，生成时钟在设计内部被 pll 或某些用户逻辑所驱动。生成时钟和主时钟相关联，主时钟可以是基准时钟或者是其他的生成时钟。

相较于直接指定生成时钟的周期或者波形属性，生成时钟的属性由主时钟变换而来，只需要描述相对于主时钟的变换关系。主时钟和生成时钟之间的关系可以是下列这些情况分频、倍频、相位移动、波形反转和占空比转换以及他们的任意组合。

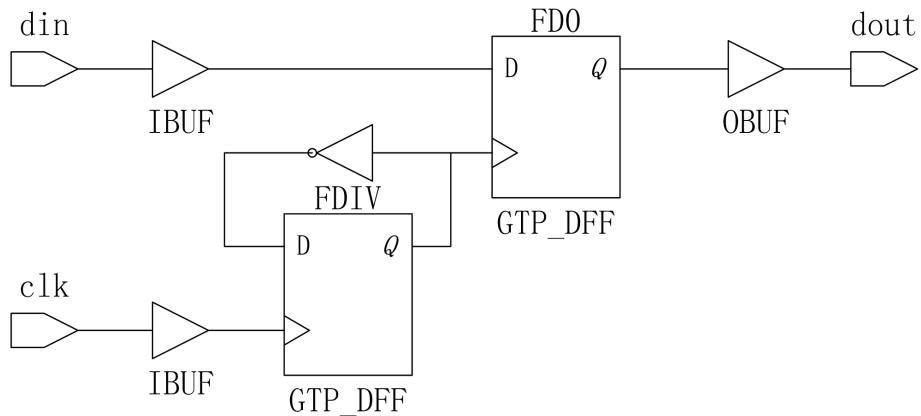


图 1-15 Create Generated Clock 示例

如上图所示，经输入 port clk 输入的基准时钟 clk 周期为 10ns，经过 FDIV 二分频驱动 FD0，对应的生成时钟为 clkdiv2。

Create Generated Clock 窗口创建 create\_generated\_clock 命令，窗口上部为该条命令的简介；中间为创建该条命令的操作，command 一行可以显示创建的命令的形式以及参数；下部中点击 Reset 按钮则重置该界面。窗口界面如下：

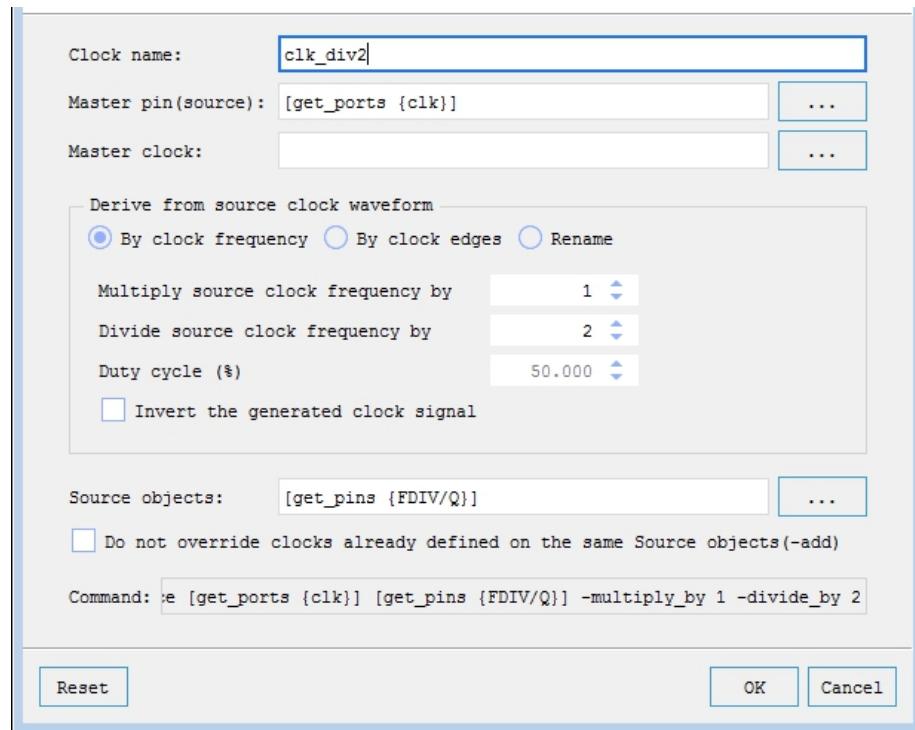


图 1-16 Ceate Generated Clock 窗口

Create Generated Clock 表格用来显示该条命令，表格如下：

Create Generated Clock								
	Enable	Clock name	Master pin(source)	Source objects	Master clock	Multiply By	Divide By	Duty Cycle(%)
1	<input checked="" type="checkbox"/>							

图 1-17 Create Generated Clock 表格

Create Generated Clock 表格指定的内容分两类：指定 source clock，指定如何进行变换，其各列介绍如下：

**Enable 列：**同上；

**Clock name 列：**对应表格在 source clock 的基础上产生的 generated clock 的名字；它对应于 create\_generated\_clock 命令的 -name <clock\_name> 参数；

**Master Pin (Source) 列：**用于描述产生该 generated clock 的 source clock 所在的 source；它对应于 create\_generated\_clock 命令的 -source <master\_pin> 参数；

**Source Objects 列：**用于描述产生该 generated clock 所作用的 source object；它对应于 create\_generated\_clock 命令的 <source\_objects> 参数；

**Master Clock 列：**用于描述产生该 generated clock 的 source clock，如果上面的 Source 列填写的 source 上只有一个 clock，本处可以不写，反之，如果 source 上有多个 clock 的话，Master Clock 必须填写，否则会报错，至于报错的内容在第三章约束检查部分会有说明；它对应于 create\_generated\_clock 命令的 -master\_clock <clock> 参数；

上面三个参数都是描述该 generated clock 的源头和作用对象的；下面的参数用于描述如何在 source clock 的基础上通过变换来产生 generated clock，如下图所示：

Master clock	Multiply By	Divide By	Duty Cycle(%)	Edges	Edge Shift	Invert	Add
						<input type="checkbox"/>	<input type="checkbox"/>

图 1-18 Create Generated Clock 参数

**Multiply By 列：**multiply\_by 参数：也是用来变换 source clock 的频率值，描述倍频系数，source clock 的频率值乘以 multiply\_by 参数值得到 generated clock 的频率值；它对应于 create\_generated\_clock 命令的 -multiply\_by <factor> 参数；

**Divide By 列：**顾名思义，divide\_by 是“被除以”之意，这里指的是 source clock 的频率“被除以”，所以本参数描述分频系数，source clock 的频率值除以 divide\_by 参数值得到 generated clock 的频率值；它对应于 create\_generated\_clock 命令的 -divide\_by <factor> 参数；

**Duty Cycle 列:** 用于描述高电平在整个周期中所占比例，需要配合 multiply\_by 一起使用；它对应于 create\_generated\_clock 命令的 -duty\_cycle <percent>参数；当 multiply\_by 参数未指定时，duty\_cycle 参数如果指定了会告警并忽略过去；

**Edges 列:** 表示从 source clock 引用的边沿序号，数目必须为 3，它对应于 create\_generated\_clock 命令的 -edges <edge\_list>参数；

**Edge Shift 列:** 配合 edges 使用，用于描述每条边的偏移量，单位是时间；它对应于 create\_generated\_clock 命令的 -edge\_shift <shift\_list>参数；当 edges 参数未指定时，edge\_shift 参数指定了也会告警并忽略过去；

**Invert 列:** 用于将 source clock 的信号取反，它对应于 create\_generated\_clock 命令的 -invert 参数；

**Add 列:** 用于描述当有多个 clock 扇入到某 source object 时如何处理；当指定 -add 参数时，当某 source object 已经存在 clock 时，则还会加到上面，不过需要指定不同的名字；否则就先去掉 source object 上面的 clock；它对应于 create\_generated\_clock 命令的 -add 参数；

#### 1.3.4.3 Set Clock Latency 窗口及表格

Set Clock Latency 窗口创建 set\_clock\_latency 命令，窗口上部为该条命令的简介；中间为创建该条命令的操作，command 一行可以显示创建的命令的形式以及参数；下部中点击 Reset 按钮则重置该界面。窗口界面如下：

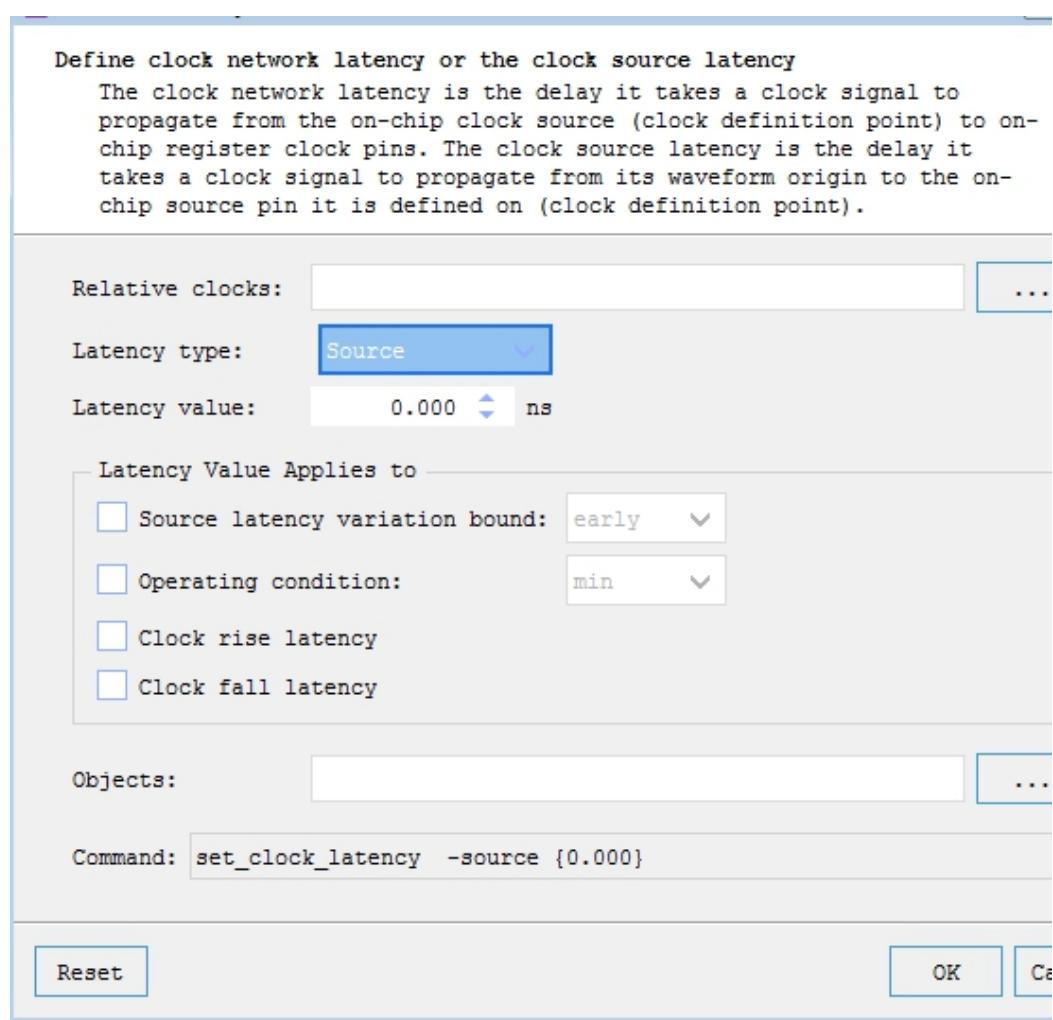


图 1-19 Set Clock Latency 窗口

Set Clock Latency 表格内容如下图所示：

Set Clock Latency								
	Enable	Relative clocks	Clock Transition	Min/Max Condition	Source Latency	Early/Late Condition	Latency value	Objects
1	<input checked="" type="checkbox"/>				<input type="checkbox"/>			

图 1-20 Set Clock Latency 表格

其各列介绍如下：

**Enable 列：** 同上；

**Relative Clocks 列：** 指定 Clock，对应命令的-clock clock\_name 选项；

**Clock Transition 列：** 指定是否为 clock rise latency、clock fall latency，对应命令的-rise/-fall 选项；

**Min/Max Condition 列：** 指定 max、min 参数，对应命令的-max/-min 选项；

**Source Latency 列：** 指定 latency 的类型，为 source latency，对应命令的-latency\_type 选项；

**Early/Late Condition 列:** 指定 source 类型的 latency 的参数 early、late，对应命令中的-early/-late 选项，该 option 只有在 latency 类型为 source 才能指定；

**Latency Value 列:** 指定 latency 的 value，对应命令的 latency\_value 选项；

**Objects 列:** 指定 objects，类型为 clock/port/pin，对应命令中的 source\_objects 选项；

#### 1.3.4.4 Set Clock Uncertainty 窗口及表格

Set Clock Uncertainty 窗口创建 set\_clock\_uncertainty 命令，窗口上部为该条命令的简介；中间为创建该条命令的操作，command 一行可以显示创建的命令的形式以及参数；下部中点击 Reset 按钮则重置该界面。窗口界面如下：

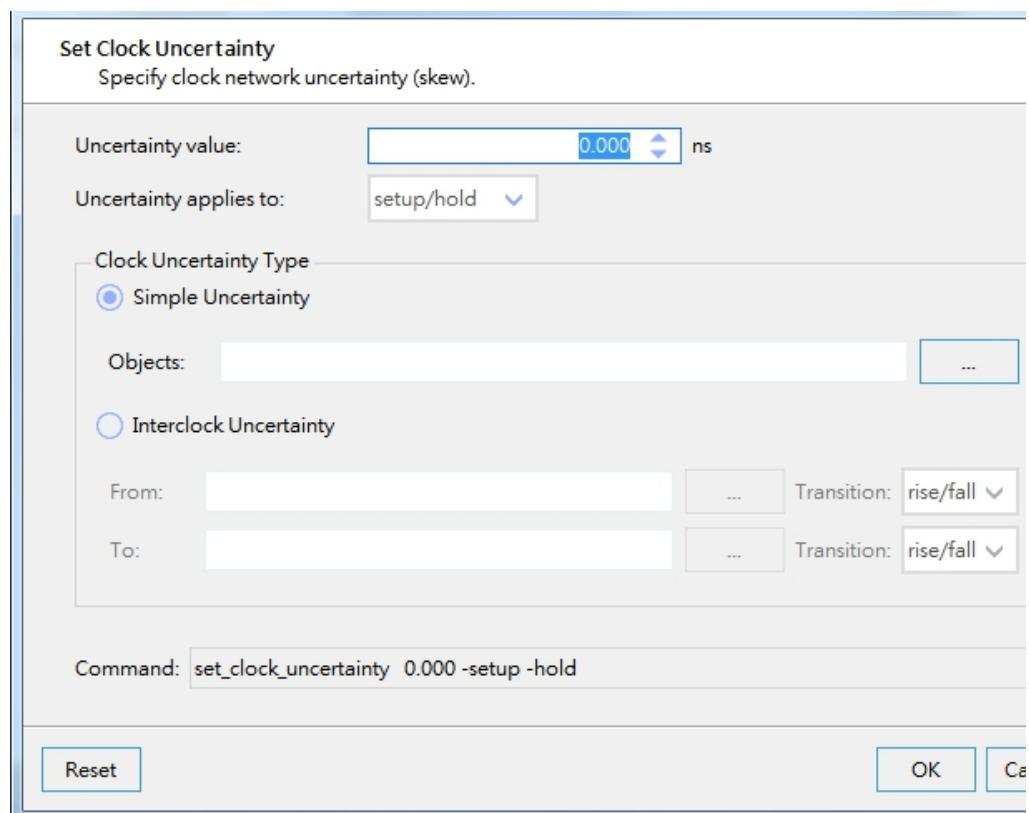


图 1-21 Set Clock Uncertainty 窗口

Set Clock Uncertainty 表格内容如下图所示：

	Enable	Uncertainty value	Setup/Hold	From Transition	From Clock	To Transition	To Clock	Objects
1	<input checked="" type="checkbox"/>							

图 1-22 Set Clock Uncertainty 表格

其各列介绍如下：

**Enable 列:** 同上；

**Uncertainty Value 列:** 指定 value 值, 对应命令中的 uncertainty\_value 参数;

**Setup/Hold 列:** 表示进行的是 setup/hold 分析, 对应命令中的 -setup/-hold;

**From Transition 列:** 指定 from 的 transition, -rise/-fall, 对应命令中的 -rise\_from/-fall\_from;

**From Clock 列:** 指定 clock, 对应命令中的 <from\_list>/ <rise\_from\_list>/ <fall\_from\_list>;

**To Transition 列:** 指定 to 的 transition, -rise/-fall, 对应命令中的 -rise\_to/-fall\_to;

**To Clock 列:** 指定 clock, 对应命令中的 <to\_list>/ <rise\_to\_list>/ <fall\_to\_list>;

**Objects 列:** 指定 objects, 类型为 clock/port/pin, 对应命令中的 <source\_objects>参数;

#### 1.3.4.5 Set Clock Groups 窗口及表格

Set Clock Groups 窗口和对应的 set\_clock\_groups 命令用于设置时钟组, 设置时钟组可以不进行不同时钟组之间的时序分析, 同时对时钟组内的分析没有影响。不同于 set\_false\_path 约束忽略单向的时序分析, 设置时钟组后会忽略双向的时序分析。

如下图所示, clkA 驱动触发器 FD0, clkB 驱动触发器 FD1A 和 FD1B; clkA 和 clkB 从不同的输入端口输入, 因此他们是异步的。为了保证从 FD0 到 FD1A 的路径被正确分析, 需要设置 clkA 和 clkB 之间的时钟组。

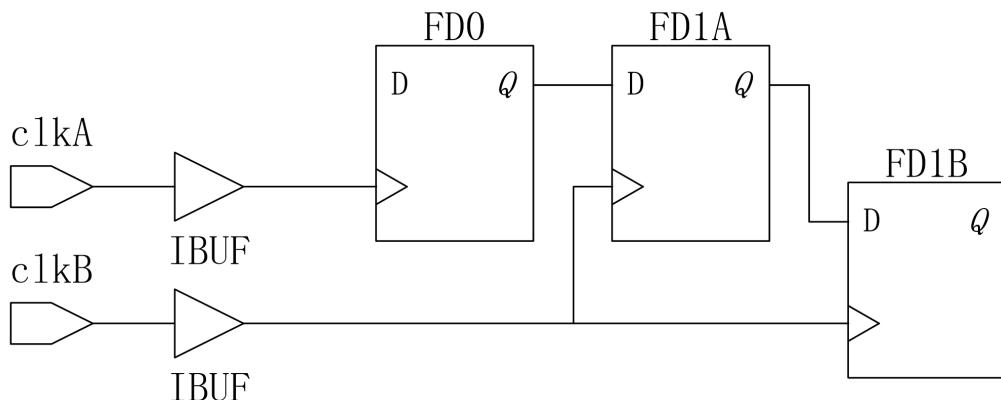


图 1-23 Set Clock Groups 示例

Set Clock Groups 窗口创建 set\_clock\_groups 命令, 窗口上部为该条命令的简

介；中间为创建该条命令的操作，command 一行可以显示创建的命令的形式以及参数；下部中点击 Reset 按钮则重置该界面。窗口界面如下：

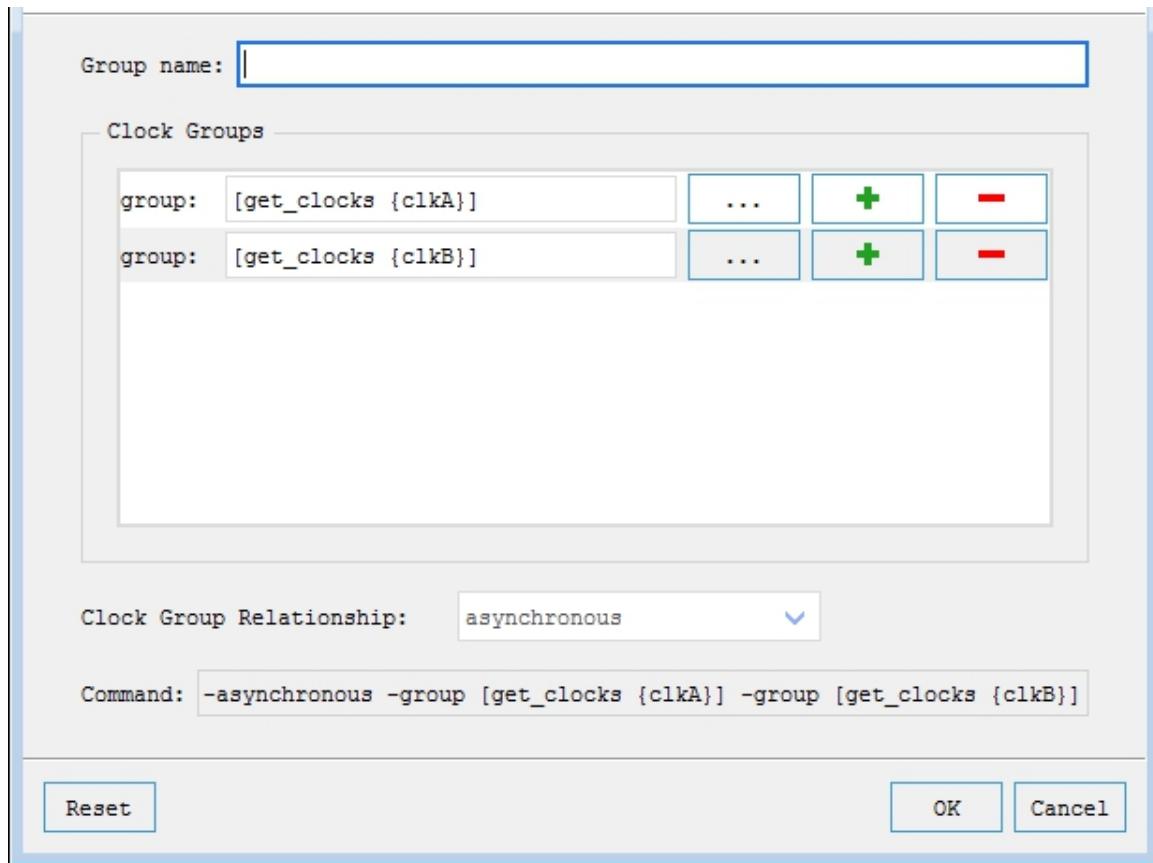


图 1-24 Set Clock Groups 窗口

Set Clock Groups 表格内容如下图所示：

Set Clock Groups - (Double Click Blank Area To Create a Set Clock Groups Constraint)				
	Enable	Group name	Clock Group Relationship	Clock Groups
1	<input checked="" type="checkbox"/>		asynchronous	-group [get_clocks {clkA}];-group [get_clocks {clkB}]

图 1-25 Set Clock Groups 表格

其各列介绍如下：

**Enable 列：** 同上；

**Group name 列：** 时钟组 clock group 的名字，它对应于 set\_clock\_groups 命令的-name group\_name 参数；

**Clock Group Relationship 列：** 时钟组 clock group 的类型，有 asynchronous、-logically\_exclusive 和-physically\_exclusive 三种；

**Clock Groups 列：** 指定的时钟组，它对应于 set\_clock\_groups 命令的-group <group\_list>参数；

### 1.3.4.6 Set\_External\_Delay 窗口及表格

Set External Delay 窗口创建 set\_external\_delay 命令，窗口上部为该条命令的简介；中间为创建该条命令的操作，command 一行可以显示创建的命令的形式以及参数；下部中点击 Reset 按钮则重置该界面。窗口界面如下：

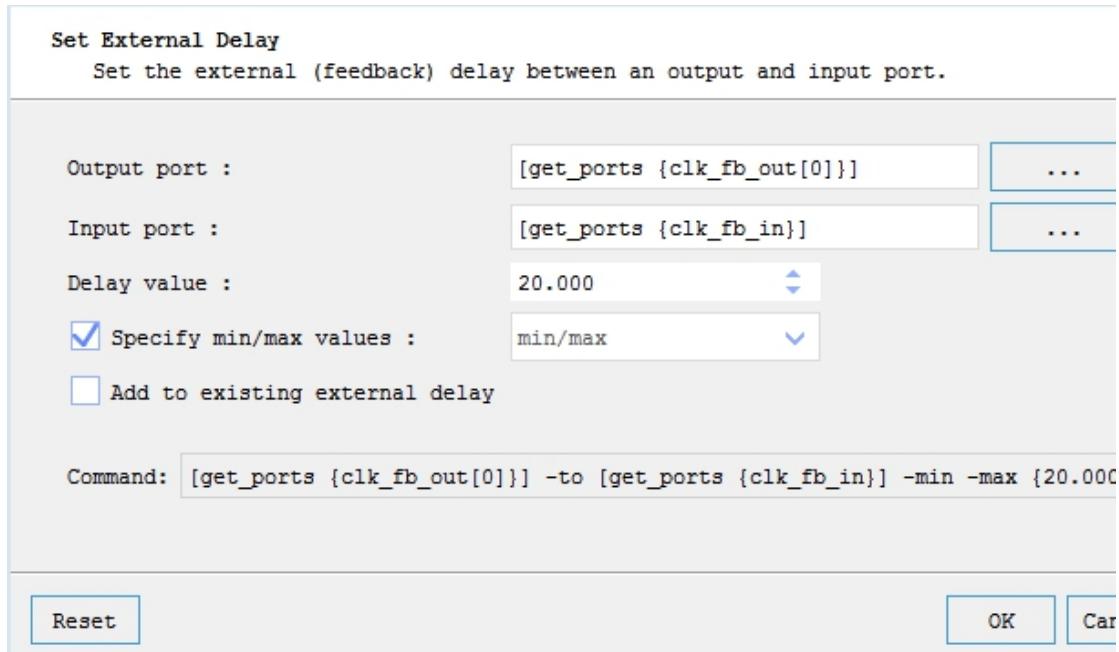


图 1-26 set\_external\_delay 窗口

Set\_external\_delay 表格内容如下所示：

Set External Delay - (Double Click Blank Area To Create a Set External Delay Constraint)					
Enable	Output port	Input port	Min/Max Condition	Delay value	Add
1 <input checked="" type="checkbox"/>	[get_ports {clk_fb_out[0]}]	[get_ports {clk_fb_in}]	min/max	20	<input type="checkbox"/>

图 1-27 set\_external\_delay 表格

其各列介绍如下：

**Enable 列：** 同上；

**Output port 列：** 指明该 delay 作用的外反馈路径的输出端口，对应命令的 -from <Output port>项；

**Input port 列：** 指明该 delay 作用的外反馈路径的输入端口，它对应命令的 -to <Input port>项；

**Min/Max Condition 列：** 指明这个 delay 是最小还是最大值，对应命令的 -min/-max 项；

**Delay value 列：** 指定外反馈路径的延时值，对应命令的 -delay\_value 项；

**Add 列：** 是否允许在同一条外反馈路径上定义多个延时值；它对应于 set\_external\_delay 命令的-add 参数；

#### 1.3.4.7 Set Input/Output Delay 窗口及表格

Set Input Delay 和 Set Output Delay 命令的 option 完全一样的，创建命令的窗口也完全一样，只有两条命令的名字不一样，所以以下只对 Set Input Delay 来做说明。

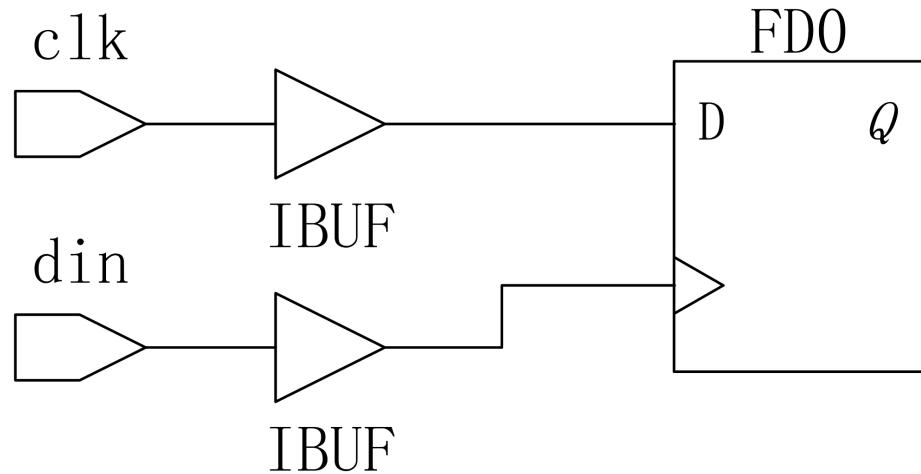


图 1-28 Set Input Delay 示例

Set Input Delay 窗口创建 set\_input\_delay 命令，窗口上部为该条命令的简介；中间为创建该条命令的操作， command 一行可以显示创建的命令的形式以及参数；下部中点击 Reset 按钮则重置该界面。窗口界面如下：

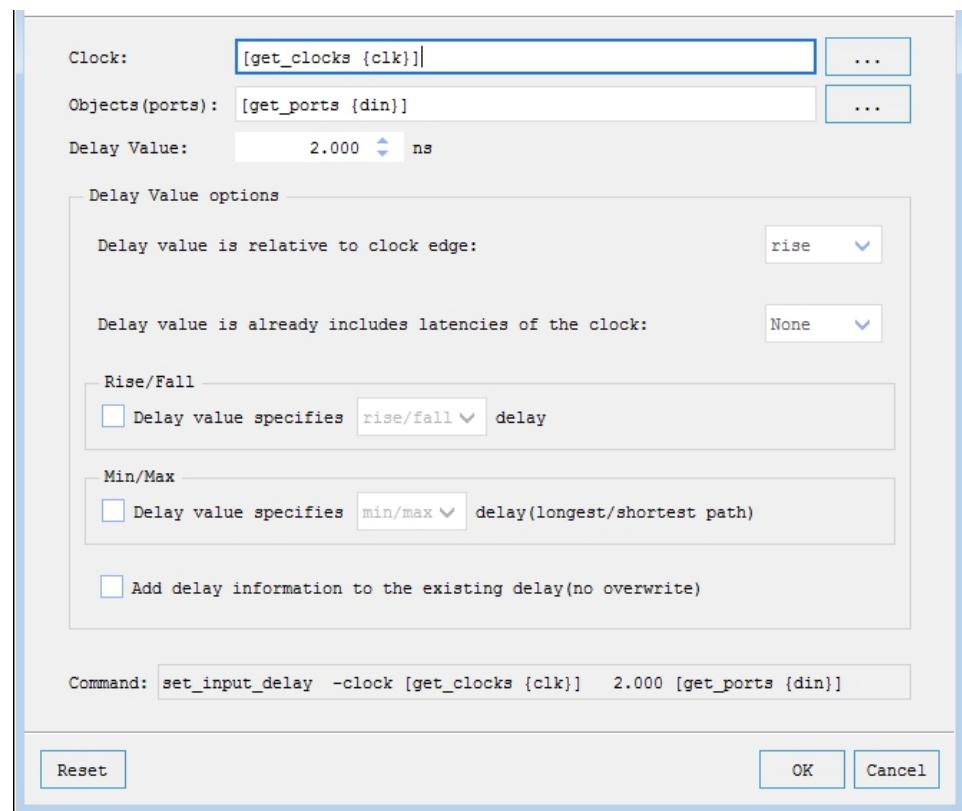


图 1-29 Set Input Delay 窗口

Set Input Delay 表格内容如下图所示：

Set Input Delay - (Double Click Blank Area To Create a Set Input Delay Constraint)								
Enable	Clock	Clock Edge	Delay Transition	Max/Min Delay Path	Add Delay	Latencies Included	Delay Value	Objects
1 <input checked="" type="checkbox"/>	[get_clocks {clk}]				<input type="checkbox"/>		2	[get_ports {din}]

图 1-30 Set Input Delay 表格

其各列介绍如下：

**Enable 列：** 同上；

**Clock 列：** 指明该 delay 作用的 clock 名，对应命令的-clock <clock\_name> 项；

**Clock Edge 列：** 指明该 delay 作用于 clock 的下降沿，它对应命令的-clock\_fall 项；

**Delay Transition 列：** 指定该 delay 只在 rising/falling transition 时有效，对应命令的-rise/-fall 项；

**Max/Min Delay Path 列：** 指明这个 delay 作用于最长路径/最短路径，对应命令的-max/-min 项；

**Add Delay 列：** 指定是否将当前 delay 覆盖已有的 delay，对应命令的-

add\_delay 项；

**Latencies Included 列：**该 option 表示时延值包含的时间，该值的含义为：

-source\_latency\_included 表示该延时值（数据路径的延迟）已经包含了时序路径的时钟源延迟时间；对应命令中的-source\_latency\_included

**Delay Value 列：**指定 delay 值，对应命令的<delay\_value>项；

**Objects 列：**指明该 delay 作用的 port，对应于命令的<objects>参数。

#### 1.3.4.8 Set Max Skew 窗口及表格

Set Max Skew 窗口创建 set\_max\_skew 命令，窗口上部为该条命令的简介；中间为创建该条命令的操作，command 一行可以显示创建的命令的形式以及参数；下部中点击 Reset 按钮则重置该界面。窗口界面如下：

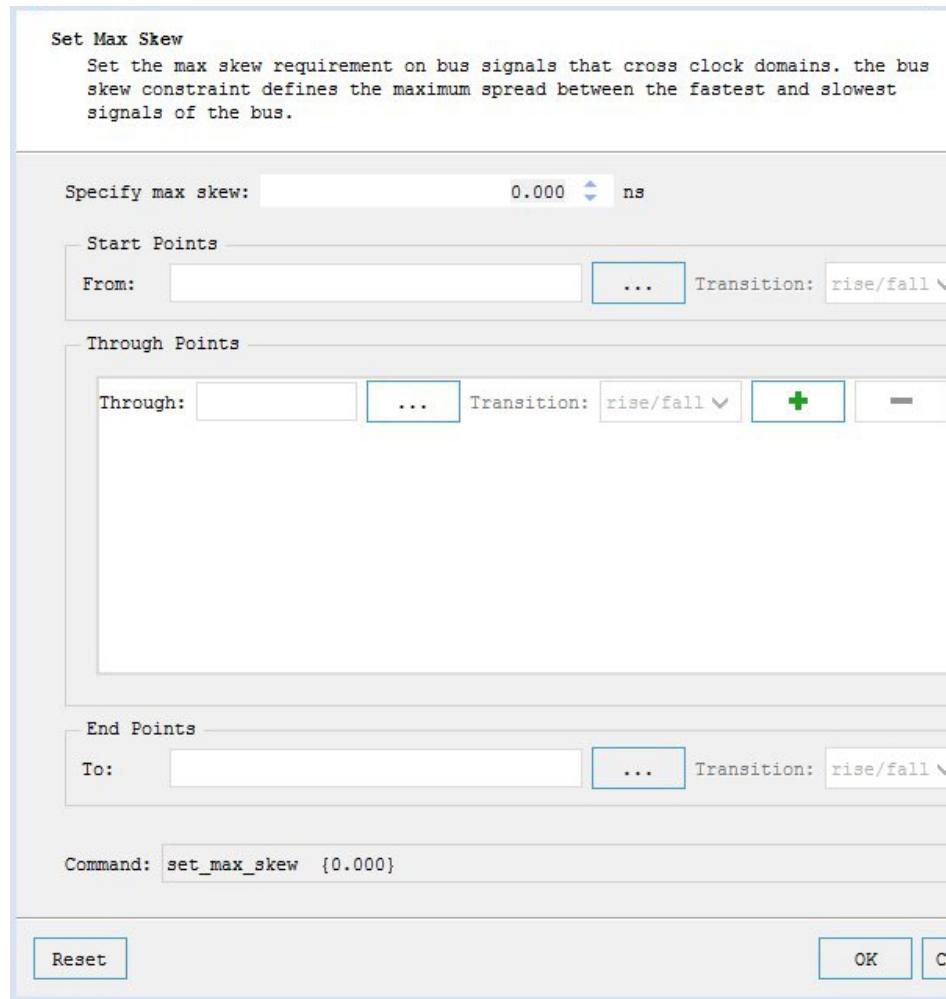


图 1-31 Set Max Skew 窗口

Set Max Skew 表格内容如下图所示：

Enable	Specify max skew	Transition(Start)	Start Points	Through Points And Transition	Transition(End)	End Points

图 1-32 Set Max Skew 表格

其各列介绍如下：

**Enable 列：** 同上；

**Specify max skew 列：** 设置 skew 值，对应命令中的 skew\_value；

**Transiton(Start) 列：** 表示 path 的 transition，对应命令中的 -from 参数；

**Start Points 列：** 表示 path 的 startpoint，对应命令中的 <from\_list> 参数；

**Through Points And Transition 列：** 表示 path 必须经过的中间点，  
对应命令中的 -through <through\_list> 参数；

**Transiton(End) 列：** 表示 path 的 transition，对应命令中的 -to 参数；

**End Points 列：** 表示 path 的 endpoint，对应命令中的 <to\_list> 参数；

#### 1.3.4.9 Set Max Delay 窗口及表格

Set Max Delay 窗口创建 set\_max\_delay 命令，窗口上部为该条命令的简介；  
中间为创建该条命令的操作， command 一行可以显示创建的命令的形式以及参数；  
下部中点击 Reset 按钮则重置该界面。窗口界面如下：

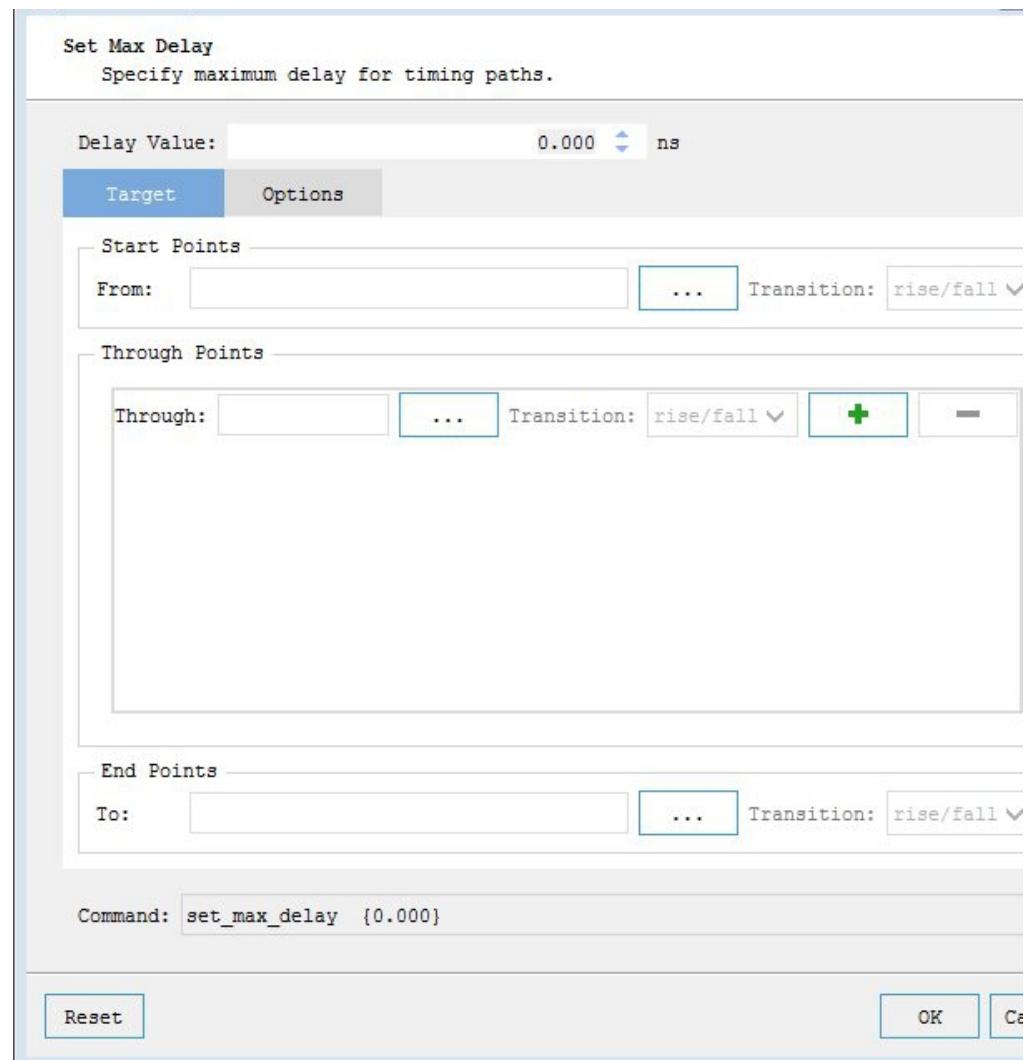


图 1-33 Set Max Delay 窗口

Set Max Delay 表格内容如下图所示：

Set Max Delay - (Double Click Blank Area To Create a Set Max Delay Constraint)								
	Enable	Data Path Only	Delay Value	Transition(Start)	Start Points	Through Points And Transition	Transition(End)	End Points
1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0	rise/fall	[get_cells {N3}]			

图 1-34 Set Max Delay 表格

其各列介绍如下：

**Enable 列：** 同上；

**Data Path Only:** 设置 datapath, 对应命令 -datapath\_only

**Delay Value 列：** 设置 delay 值，对应命令中的 <delay\_value>；

**Transiton(Start) 列：** 表示 path 的 transition，对应命令中的 -from 参数；

**Start Points 列：** 表示 path 的 startpoint，对应命令中的 <from\_list> 参数；

**Through Points And Transition 列：** 表示 path 必须经过的中间点，

对应命令中的-through <through\_list>参数；

**Transiton(End) 列：**表示 path 的 transition，对应命令中的-to 参数；

**End Points 列：**表示 path 的 endpoint，对应命令中的<to\_list>参数；

#### 1.3.4.10 Set Min Delay 窗口及表格

Set Min Delay 窗口创建 set\_min\_delay 命令，窗口上部为该条命令的简介；中间为创建该条命令的操作，command 一行可以显示创建的命令的形式以及参数；下部中点击 Reset 按钮则重置该界面。窗口界面如下：

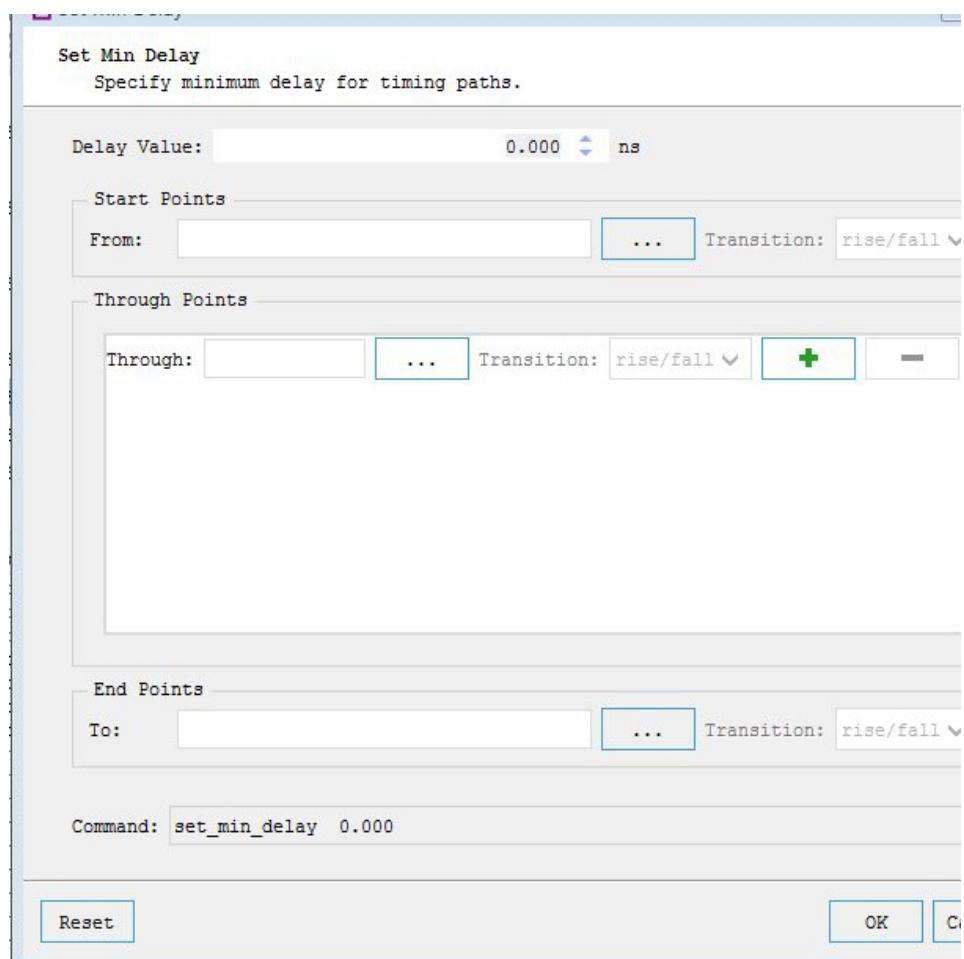


图 1-35 Set Min Delay 窗口

Set Min Delay 表格内容如下图所示：

	Enable	Delay Value	Transition(Start)	Start Points	Through Points And Transition	Transition(End)	End Points
1	<input checked="" type="checkbox"/>	1	rise/fall	[get_cells (tmp[7:0])]			

图 1-36 Set Min Delay 表格

其各列介绍如下：

**Enable 列:** 同上;

**Delay Value 列:** 设置 delay 值, 对应命令中的<delay\_value>;

**Transiton(Start) 列:** 表示 path 的 transition, 对应命令中的-from 参数;

**Start Points 列:** 表示 path 的 startpoint, 对应命令中的<from\_list>参数;

**Through Points And Transition 列:** 表示 path 必须经过的中间点,

对应命令中的-through <through\_list>参数;

**Transiton(End) 列:** 表示 path 的 transition, 对应命令中的-to 参数;

**End Points 列:** 表示 path 的 endpoint, 对应命令中的<to\_list>参数;

#### 1.3.4.11 Set Multicycle Path 窗口及表格

Set Multicycle Path 窗口创建 set\_multicycle\_path 命令, 窗口上部为该条命令的简介; 中间为创建该条命令的操作, command 一行可以显示创建的命令的形式以及参数; 下部中点击 Reset 按钮则重置该界面。窗口界面如下:

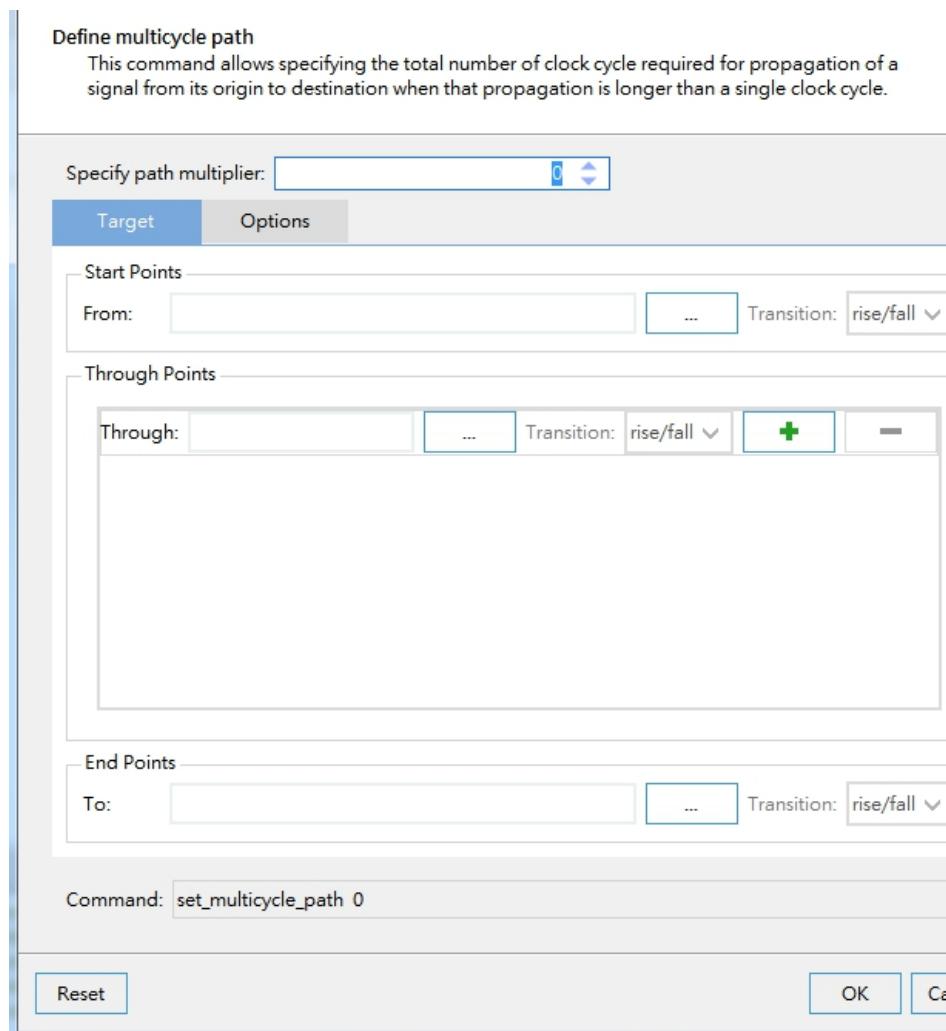


图 1-37 Set Multicycle Path 窗口

Set Multicycle Path 表格内容如下图所示：

Set Multicycle Path								
	Enable	Path Multiplier	Setup/Hold	Start/End	Transition(Start)	Start Points	Through Points And Transition	Transition(End)
1	<input checked="" type="checkbox"/>							

图 1-38 Set Multicycle Path 表格

其各列介绍如下：

**Enable 列：** 同上；

**Path Multiplier 列：** 设置 path multiplier 的值，对应命令的<path\_multiplier>；

**Setup/Hold 列：** 表示进行的是 setup/hold 分析，对应命令中的-setup/-hold；

**Start/End 列：** Start 表示 launch clock 右移的 clock 周期数目，End 表示 capture clock 右移的 clock 周期数目，对应命令中的-start/end；

**Transiton(Start) 列：** 表示 path 的 transition，对应命令中的-from 参数；

**Start Points 列：** 表示 path 的 startpoint，对应命令中的<from\_list>参数；

**Through Points And Transition 列：** 表示 path 必须经过的中间点，对应命令中的-through <through\_list>参数；

**Transiton(End) 列：** 表示 path 的 transition，对应命令中的-to 参数；

**End Points 列：** 表示 path 的 endpoint，对应命令中的<to\_list>参数；

#### 1.3.4.12 Set False Path 窗口及表格

Set False Path 窗口创建 set\_false\_path 命令，窗口上部为该条命令的简介；中间为创建该条命令的操作，command 一行可以显示创建的命令的形式以及参数；下部中点击 Reset 按钮则重置该界面。窗口界面如下：

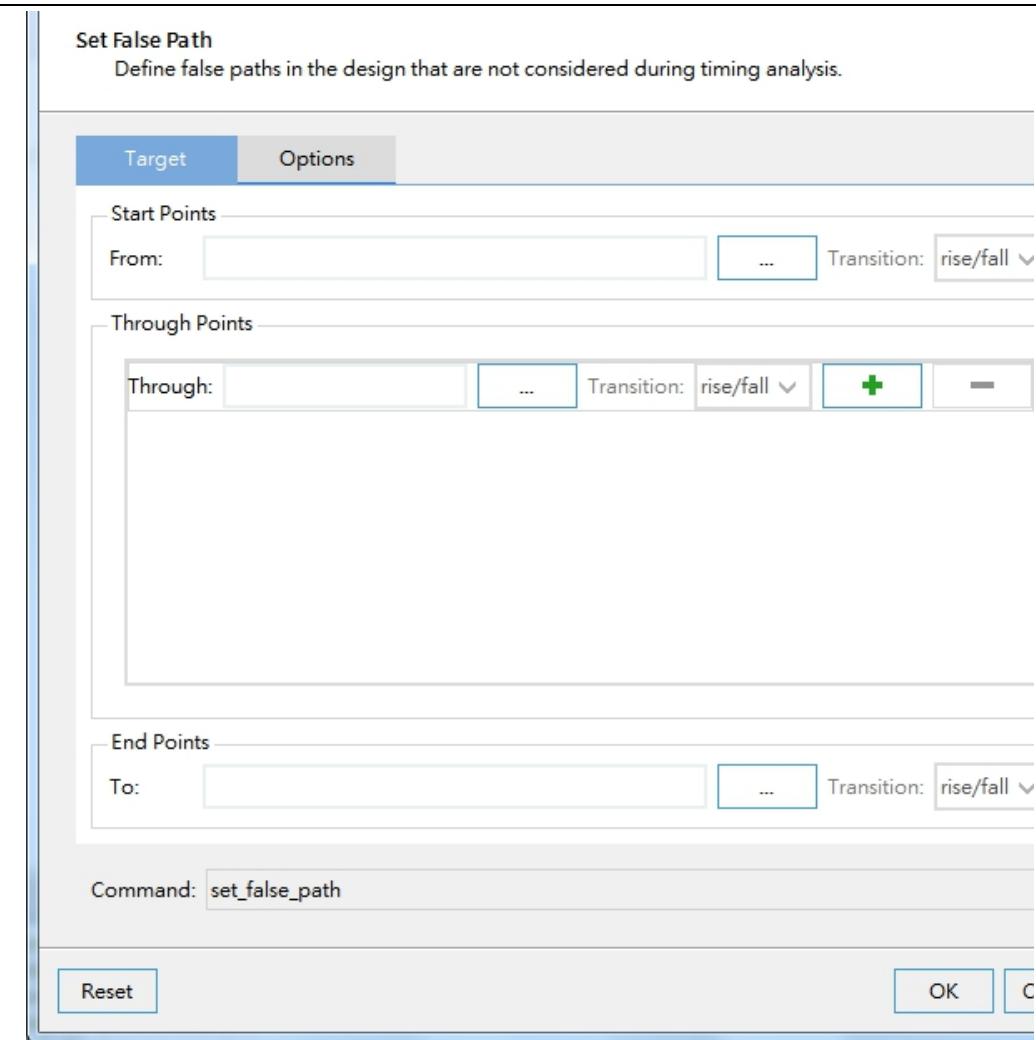


图 1-39 Set False Path 窗口

Set False Path 表格内容如下图所示：

Set False Path		Enable	Setup/Hold	Transition(Start)	Start Points	Through Points And Transition	Transition(End)	End Points
+	-	1	<input checked="" type="checkbox"/>					

图 1-40 Set False Path 表格

其各列介绍如下：

**Enable 列：** 同上；

**Setup/Hold 列：** 表示进行的是 setup/hold 分析，对应命令中的-setup/-hold；

**Transiton(Start) 列：** 表示 path 的 transition，对应命令中的-from 参数；

**Start Points 列：** 表示 path 的 startpoint，对应命令中的<from\_list>参数；

**Through Points And Transition 列：** 表示 path 必须经过的中间点，对应命令中的-through <through\_list>参数；

**Transiton(End) 列：** 表示 path 的 transition，对应命令中的-to 参数；

**End Points 列:** 表示 path 的 endpoint, 对应命令中的<to\_list>参数;

#### 1.3.4.13 Set\_Disable\_Timing 窗口及表格

Set Disable Timing 窗口创建 set\_disable\_timing 命令, 窗口上部为该条命令的简介; 中间为创建该条命令的操作, command 一行可以显示创建的命令的形式以及参数; 下部中点击 Reset 按钮则重置该界面。窗口界面如下:

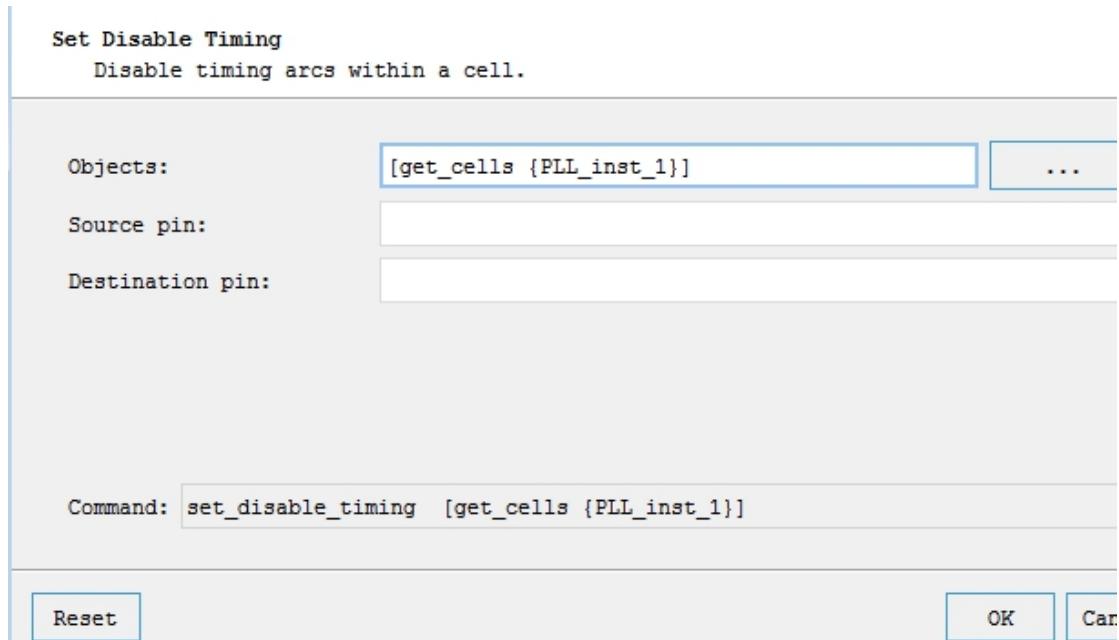


图 1-41 Set\_disable\_timing 窗口

Set Disable Timing 表格内容如下图所示:

Set Disable Timing - (Double Click Blank Area To Create a Set Disable Timing Constraint)				Objects
Enable	Source pin	Destination pin		
1 <input checked="" type="checkbox"/>			[get_cells {PLL_inst_1}]	

图 1-42 set\_disable\_timing 表格

其各列介绍如下:

**Enable 列:** 同上;

**Source pin 列:** 指明该 timing arc 的起点, 对应命令中的 -from <from\_pin\_name>参数;

**Destination pin 列:** 指明该 timing arc 的终点, 对应命令中的 -to <to\_pin\_name>参数;

**Objects 列:** , 对应命令中的<object\_list>参数;

### 1.3.5 Attribute 表格

Attribute 表格内容如下图所示：

Timing Constraints	Attributes	Device
+	Enable	Object
		Attribute
		Value
		Value Type
		Description

图 1-43 Attribute 表格

Attribute 表格用来设置 UCE 功能中的逻辑约束，目前逻辑约束是以 attribute 形式体现在约束文件里的，下面分别介绍该 Attribute 表格各功能。

点击左上角或右键的 Insert Row 按钮，弹出 Attributes Table 窗口，如下图所示：

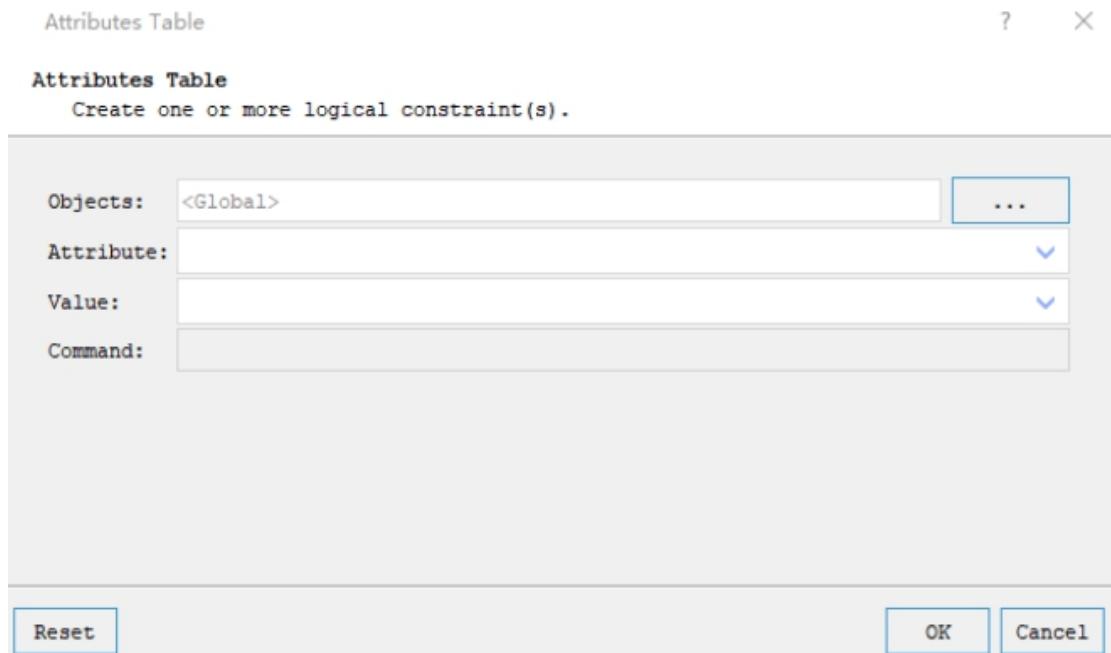


图 1-44 Attributes Table 窗口

**Object 栏：**表示该 attribute 标注到的 object，attribute 可以标注到 port/pin/instance/net 上，可以手动输入也可以选中单个或多个对象，当鼠标点击 Object 栏右侧选择按钮会弹出 Specify Object 窗口，下图所示：

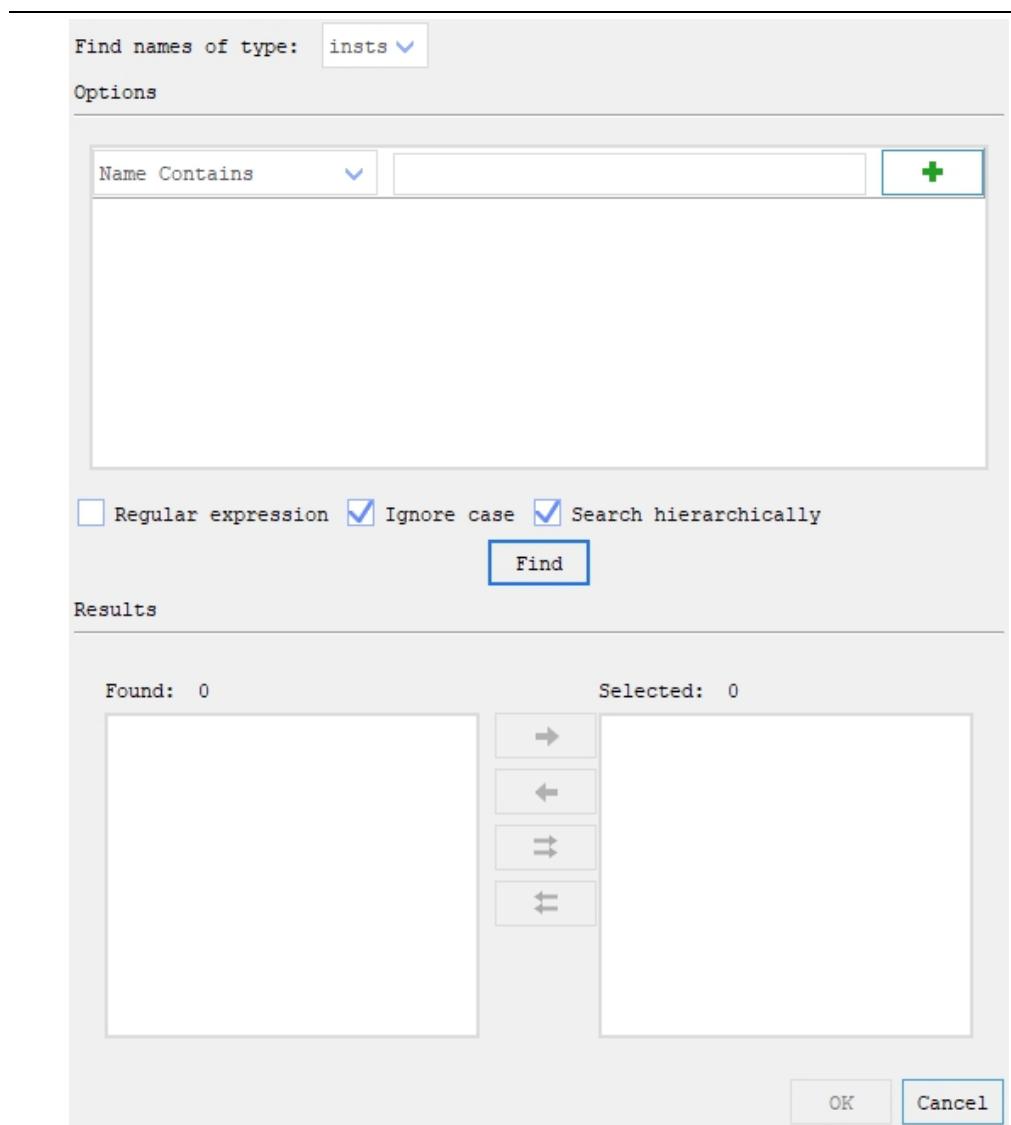


图 1-45 Specify Object 窗口

**Attribute 栏:** 表示所约束的 attribute 类型，可下拉进行选择 attribute，如下图所示：

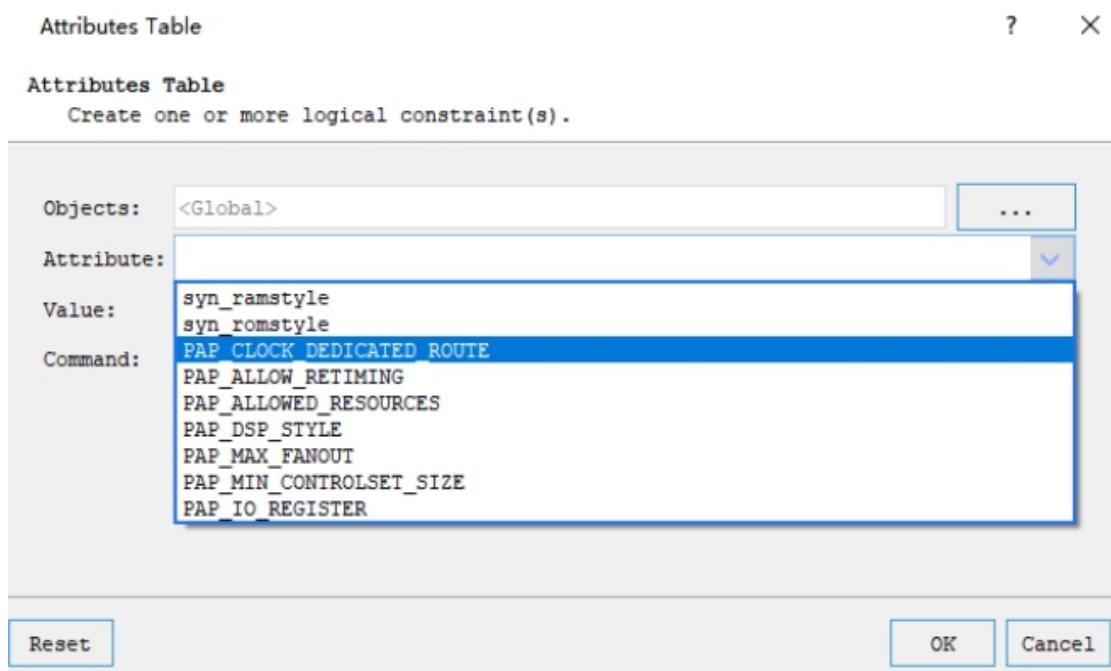


图 1-46 Attributes table 中的 attribute 行

**Value 栏:** 表示所约束的 attribute 类型的属性值，可下拉进行选择或者手动输入，如下图所示：

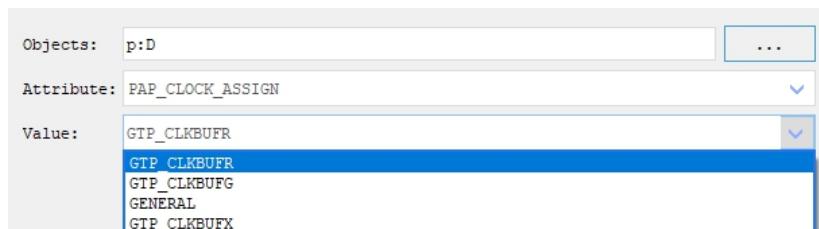


图 1-47 Attributes table 中的 Value 行

**command 栏:** 显示所设置的约束命令，如下图所示：

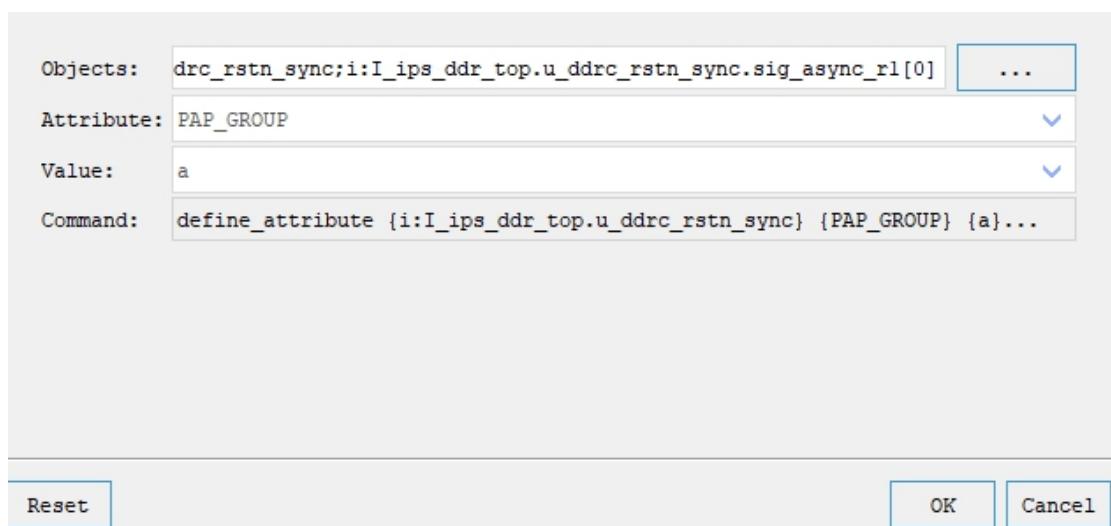


图 1-48 Attributes table 中的 command 样

此窗口中还存在 reset 按钮用于清空设置，点击 ok 后会进行约束检查，若存在错误，会出现弹窗提示，并且约束无法保存到 attribute 表格中。若在窗口选择多个对象做逻辑约束并点击 ok 后约束检查通过，多个对象的逻辑约束会逐条显示在 attribute 表格中，如下图所示：

	Enable	Object	Attribute	Value
— 1	<input checked="" type="checkbox"/>	i:genblk1[12].t2	PAP_GROUP	a
2	<input checked="" type="checkbox"/>	i:genblk1[13].t2	PAP_GROUP	a
3	<input checked="" type="checkbox"/>	i:genblk1[14].t2	PAP_GROUP	a
4	<input checked="" type="checkbox"/>	i:genblk1[15].t2	PAP_GROUP	a
5	<input checked="" type="checkbox"/>	i:genblk1[16].t2	PAP_GROUP	a
6	<input checked="" type="checkbox"/>	i:genblk1[17].t2	PAP_GROUP	a

图 1-49 保存的逻辑约束

若在 Attributes table 中不进行操作，直接点击 ok，attribute 表格中会增加空自行（如下图所示），并对可以对空自行进行操作。

	Timing Constraints		Attributes	Device
	Enable	Object	Attribute	Value
.v	— 1	<input checked="" type="checkbox"/>		

图 1-50 新增空白行

**Enable 列：**同上；

**Object 列：**表示该 attribute 标注到的 object，attribute 可以标注到 port/pin/instance/net 上，当鼠标点击该列，会出现下图所示：

注：在 Attribute 表格中的 Object 栏中只能选择或输入一个对象，若输入多个对象此命令无效。

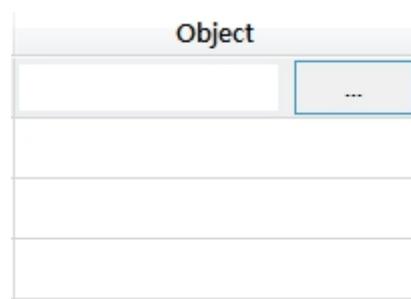


图 1-51 Attribute 表格的 Object 列

点击上图中的按钮，会弹出 UCE 的网表搜索工具。通过网表搜索工具来搜索相应的内容。

Object 列的 object 在表格中会加上一个前缀来表示其类型，其中， instance 类型前缀为“i:” , net 类型为“n:” , port 类型为“p:” , pin 类型为“t:”。

**Attribute 列：**表示所约束的 attribute 类型，如下图所示：

Timing Constraints								Attributes	Compile Points	Device
	Enable	Object	Attribute	Value	Value Type	Description	Comment			
-	1 <input checked="" type="checkbox"/>	i:N0	PAP_CLOCK_ASSIGN	GTP_CLKBUFR						
	2 <input checked="" type="checkbox"/>		PAP_CLOCK_ASSIGN							
	3 <input checked="" type="checkbox"/>		PAP_IO_REGISTER							
	4 <input checked="" type="checkbox"/>		PAP_GROUP							
	5 <input checked="" type="checkbox"/>		PAP_PLL_FEED_BACK_DELAY							
	6 <input checked="" type="checkbox"/>		syn_enum_encoding							
	7 <input checked="" type="checkbox"/>		syn_force_seq_prim							
			syn_rename_module							
			syn_safe_case							
			syn_sharing							
			syn_tristate							

图 1-52 Attribute 表格的 Attribute 列

这里的 attribute 类型会根据“Object”列选择的不同而提供不同的可选值，如上图所示，当“Object”类型选择 instance 时，“Attribute”列可选值为 PAP\_GROUP/PAP\_IO\_REGISTER 等属性。

剩下的列如下图所示：

Value	Val Type	Description	Comment
GTP_CLKBUFR			
TRUE			
0	boolean	Preferred clo...	

图 1-53 Attribute 表格的 value 相关列

**Value 列：**用于描述该 attribute 的值，其格式和值本身的要求跟具体 attribute 有关系，不符合的会在约束检查中报错或者警告，在第二章会介绍各个属性的值的要求；

**Value Type/Description/Comment 列：**用于进一步描述该 attribute value 相关的特性，目前忽略；

### 1.3.6 Compile Points 界面

Compile Points 界面内容如下图所示：

Timing Constraints				Attributes	Compile Points	Device
	Enable	View	Type			Comment
-	1 <input checked="" type="checkbox"/>					

图 1-54 Compile Points 表格

Compile Points 表格用来在 UCE 设置 Compile Points，目的是为增量式编译服务（综合工具为 OEM 是才存在 Compile Points 界面）。

**Enable 列：**同上；

**View 列：**Module name，该选项下拉列表中为用户设计文件中的所有 Module 名称。

**Type 列：**Compile Points 类型，共分为 black-box/hard/locked/soft 四种类型；

**Comment 列：**用于进一步描述该 Compile Points 相关的特性，目前忽略；

### 1.3.7 Device 界面

Device 界面内容如下图所示：

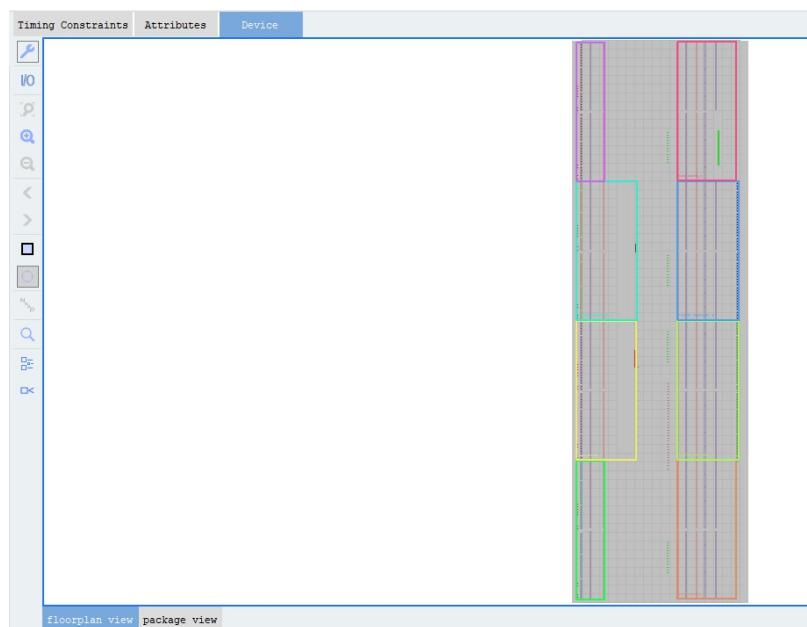


图 1-55 Device 界面

Device 界面主要用来处理以下约束：

- (1) 对用户 design 的顶层 port 进行约束，比如其位置、电平标准等；
- (2) 对区域进行约束，包括某些 design 约束到区域中；
- (3) 对预留的位置进行标记，并传递到后续步骤中。

#### 1.3.7.1 Device 界面工具栏

Device 界面工具栏如下图所示：



图 1-56 Device 界面工具栏

从左到右依次为：

Device Browser: 显示 Device Browser。

I/O Table: 显示 I/O Table 表格。

Zoom: View All、Zoom In、Zoom Out 三个按钮，点击分别有显示资源整体布局、放大、缩小的功能。

History: 显示 Previous View 和 Next View, 用于切换 floorplan view 或 package view 历史操作的视图。

Mode: Region Mode 按钮，点击切换 region 模式。

Auto-Fit Selection: 自适应选中功能。

Show Differential IO: 差分按钮，点击在 package view 中显示差分信息。

Search Inst: 搜索按钮，点击以后通过搜索框在 Device 界面搜索 Device 信息。

View All Widget: 该窗口可以实时显示 device view 界面所在芯片的位置以及选中的 design net，该窗口通过点击滚动条之间的按钮，在 device view 界面右下角显示。

Show Cell Connections: 该窗口可以设置 show connectivity 功能的显示规则。

### 1.3.7.2 Device 界面窗口

Device 界面窗口根据布局分为 floorplan view 窗口、package view 窗口。

floorplan view 窗口用来显示芯片的资源位置，四周为 IO 资源，中间为逻辑资源。用户可以通过拖放、取消约束等进行界面约束操作。拖放进 floorplan view 窗口的实例(primitives)会高亮显示。package view 窗口用来显示芯片的管脚封装位置，拖放约束的 IO 会高亮显示。

floorplan view 窗口和 package view 窗口可以通过 Device 界面下方的按钮进行切换，可以通过鼠标滑轮或者工具栏中的 Zoom 按钮进行缩放操作。

另外，Device Broswer、IO Table、搜索窗口分别通过工具栏相应的按钮控制显示或者隐藏，其界面显示如下所示：



图 1-57 Device Broswer 界面 Design 窗口

Design 窗口的下半部分为 Select save mode 设置项，包含三个选项分别为：

Use the default mode、Use the multi-line mode、Use the single-line mode，默认选中 Use the default mode，可对 ADS 流程下 UCE 的文件的 IO 约束命令进行转换，规则为：Use the default mode-如果当前 fdc/lcf 文件中的命令为全部是单行多属性形式，则新增约束按单行多属性命令形式增加。如果当前 fdc/lcf 文件中的命令为全部是单行多属性形式，则新增约束按单行多属性命令形式增加。如果当前 fdc/lcf 文件中的命令为单行多属性形式和多行单属性形式混合，则新增约束按多行单属性命令形式增加；Use the multi-line mode-将当前的 fdc/lcf 文件中的 IO 约束命令转换为多行单属性形式；Use the single-line mode-将当前的 fdc/lcf 文件中的 IO 约束命令转换为单行多属性形式；

Design 窗口的中部为 General 窗口存在配置项-Add comment to error cmds，默认不勾选，当勾选该框时，激活 PCE 保存时，pcf 中会将错误、无效的约束命令注释并在 console 打印报告信息；未勾选时，激活 PCE 保存时，pcf 中不会将错误、无效的约束命令注释并在 console 打印报告信息

Device 窗口上部展示硬件上的信息，主要为 Design、Device、I/O Banks。三种类型下分别为约束的 object、Bundle nets、硬件上的 device、硬件上的 I/O。若相应类型的前面状态为被选中，则表示该种类型在 floorplan view/package view 中显示。若没有被选中，则表示该种类型在 floorplan view/package view 中不会显示。

在 Device Browser 中选中 Bundle Nets 及其下子项右键存在 Edit Colors 功能。

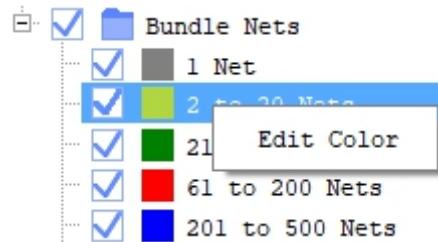


图 1-58 Edit Color 功能

点击 Edit Color 后可以打开 Preferences – Bundle Nets 页进行 Bundle nets 颜色、范围和线宽的编辑和修改工作，保存编辑后重新打开 UCE，更改的配置生效。

另外，Device 窗口中有 Clock Pad、Ref Pin、Share Pin、PLL Pad 四类 pin 脚的可选项，代表四种不同类型的管脚（打开界面时，四种类型均默认为不勾选状态）。勾选某种类型，则在 Package View 中画出对应标识的管脚（Package View 画图中，C 表示 Clock Pad，R 表示 Ref Pin，S 表示 Share Pin，P 表示 PLL Pad）。

Tool Tabs							
	I/O NAME	I/O DIRECTION	LOC	BANK	VCCIO	IOSTANDARD	DRIVE
	INVIN	Input					
	INVIN_B	Input					
	INVOUT	Output					
	INVOUTCO	Output					
	INVOUTCOB	Output					
	INVOUTDS	Output					
	INVOUTDSB	Output					

图 1-59 IO Table 表格

IO Table 表格可以对 I/O 进行设置，其约束效果也是实时反馈到 package view 和 Design Browser 中。Device 界面工具栏中点击按钮 “I/O Table” ，可以显示 IO Table 。再次点击，则隐藏 IO Table 。

IO Table 工具栏依次功能为：

搜索功能：对 port 名称进行过滤显示；

折叠功能：对树型结构显示的 IO Table 进行折叠；

展开功能：对树型结构的 IO Table 进行展开；

切换视图功能：对 IO Table 显示效果进行切换，表格视图和树型结构视图两者切换显示。两个视图显示的数据均同步。

另外：

点击 IO Table 表头，可以进行行排序；

移动 IO Table 表头，可以改变列的位置；

树型结构中支持对 portbus 类型进行批量操作。



图 1-60 搜索框

搜索框提供在芯片资源界面搜索具体 Device Instance 的功能，搜索结果会跳转到 floorplan view 窗口具体位置上。

### 1.3.7.3 Device 界面操作

#### (一) 约束实例

点击按钮【Design Browser】，点击某 primitives 实例名，例如 T\_GTP\_LUT2，按住鼠标左键，拖至 Floorplan view 上某位置，若约束成功，则该位置资源高亮，实例名前图标消失。该操作对应 PAP\_LOC 属性。

若某实例已约束，则名称前无图标（未约束的显示绿色图标）。点击该名称后，右侧高亮并显示该实例，如下图所示。

同时，可以在实例名处右键选择 undo constraint，或者鼠标在 Floorplan view 上拖出已经约束的 instance，进行取消约束。

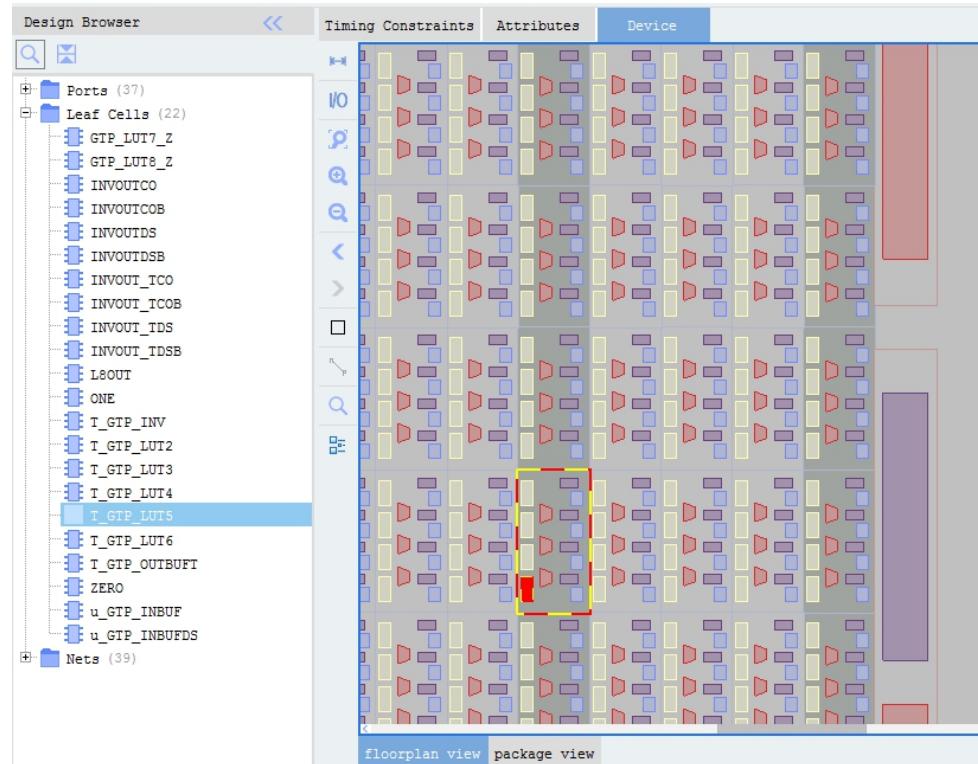


图 1-61 Device 查看实例

## (二) IO 约束

通过 IO Table 表格对 IO 进行编辑约束、取消约束。也可以通过 Package View 对 Design Broswer 的 IO 拖放操作进行约束。约束的 IO 信息会在 Design Broswer 和 Package view 中实时显示，如下图所示：

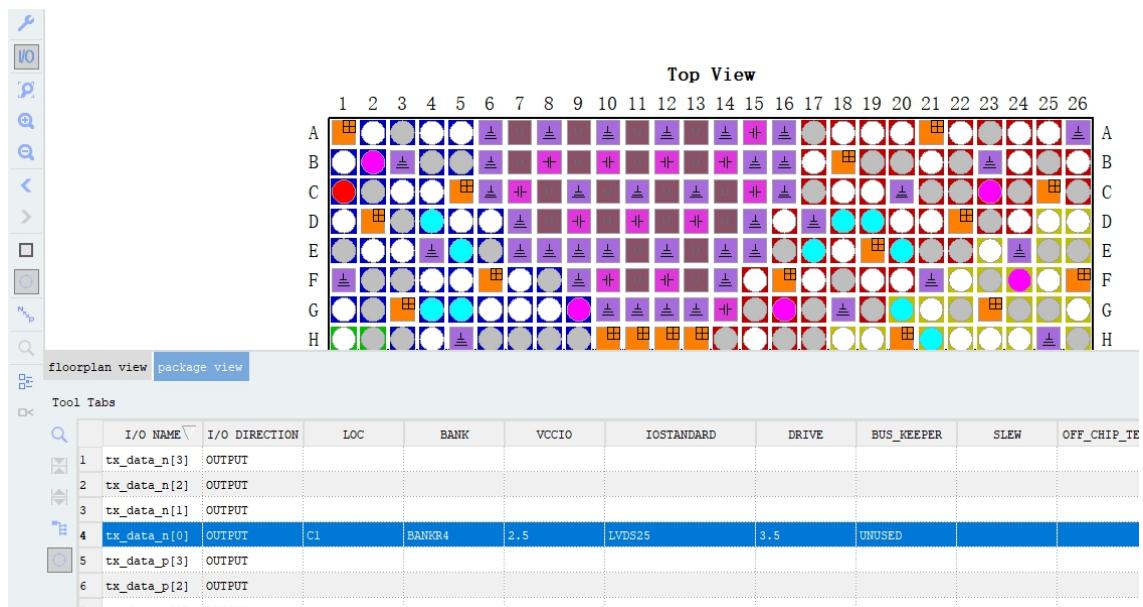


图 1-62 IO 约束

IO Table 在约束文件里也是以 attribute 的形式表示，其各列介绍如下：

I/O NAME 列：用于描述该 port 的名字；

I/O DIRECTION 列：显示 I/O 类型 input、output、inout；

LOC 列：约束的管脚；

BANK 列：该管脚所在的 bank；

VCCIO 列：IO 输入输出电压；

IOSTANDARD 列：IO 电平标准；

DISABLE\_DRIVE 列：DRIVE 使能（仅 TITAN 系列支持，该列用于使能 DRIVE，勾选则表示可以设置 PMOS/NMOS，不勾选则表示可以设置 DRIVE，界面默认为不勾选状态，该参数不会写入约束文件中）；

DRIVE 列：驱动电流；

PMOS\_FINGER：单端驱动器的 PMOS 驱动能力(仅 TITAN 系列支持)；

NMOS\_FINGER：单端驱动器的 NMOS 驱动能力(仅 TITAN 系列支持)；

BUS\_KEEPER 列：设置上拉下拉电阻；

SLEW 列：转换速率；  
OFF\_CHIP\_TERMINATION：外接电阻；  
HYS\_DRIVE\_MODE 列：输入迟滞模式；  
VREF\_MODE 列：参考电压 VREF 模式；  
VREF\_MODE\_VALUE 列：内部产生 VREF 和专用 VREF 输入管脚模式设置；  
DDR\_TERM\_MODE 列：DDR memory 应用输入匹配电阻设置；  
DIFF\_IN\_TERM\_MODE 列：差分输入的输入匹配电阻设置；  
OPEN\_DRAIN 列：驱动模式选择；  
IN\_DELAY 列：输入延迟；  
OUT\_DELAY 列：输出延迟；  
CP\_IPT 列：修正 poly 电流的 default 值；  
CAL\_MODE 列：DDR 模式输入内置端接电阻校准功能控制；  
DDR\_RES 列：使用校准功能使能时的内置电阻值  
差分端口：差分对设置属性时，其 IOSTANDARD/VCCIO 等配置应该一致。

且 P 端和 N 端的 LOC 约束应该也与 device 的 P 端和 N 端一致，如果约束的 LOC 相反，运行流程时会在 ADS 流程的 Synthesize 报错，OEM 流程的 Device Map 报错。

IO Table 表格右键中，有菜单帮助编辑 IO 约束。其中对已有的单行端口约束，可右键选择多个属性值进行复制，然后选择其他端口右键粘贴。达到复制多个的效果。对于点击的单行端口，可以右键清楚约束，或者清楚所有端口的约束。

### （三）区域约束

目前支持做区域约束的资源有 APM / DRM / CLMA / CLMS，其他资源推荐做 PAP\_LOC 约束。该操作对应 PAP\_REGION 属性，具体操作方法如下：

1) 进入打开 region 窗口，右键可选择 New Region、Delete Region，下方 General 显示区域支持 device，Statistics（已有区域约束）可显示选中区域的中含有资源，Design Instance 中可以看到已经约束到 region 中的 inst。如下图所示。



图 1-63 区域约束

2) 也可在 floorplan view 窗口中进行区域约束，点击工具栏中的 region mode 进入 region 模式，在 floorplan view 窗口右键选择 Draw Region，弹框 New Region 窗口，该窗口支持编辑区域支持的 device 和显示区域支持的 device 的数量和类型，鼠标变换为十字行后画出区域约束的范围即可。

对于在 floorplan 中画出的区域，可以根据弹窗中的操作对区域的范围进行调整。比如通过鼠标画出一个区域。在绘制的过程中鼠标出可以试试显示目前有的资源，供用户进行判断。然后会弹出新建 region 框来进一步显示 region 中资源状况

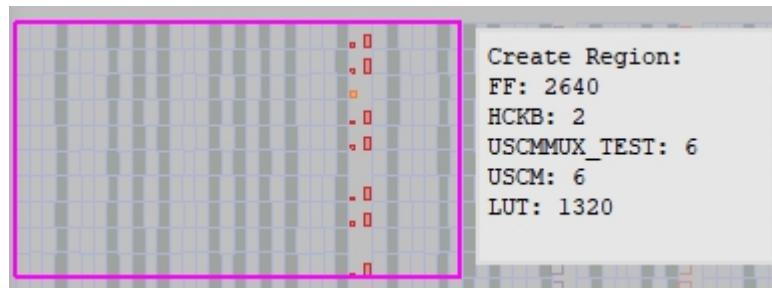


图 1-64 新建 region

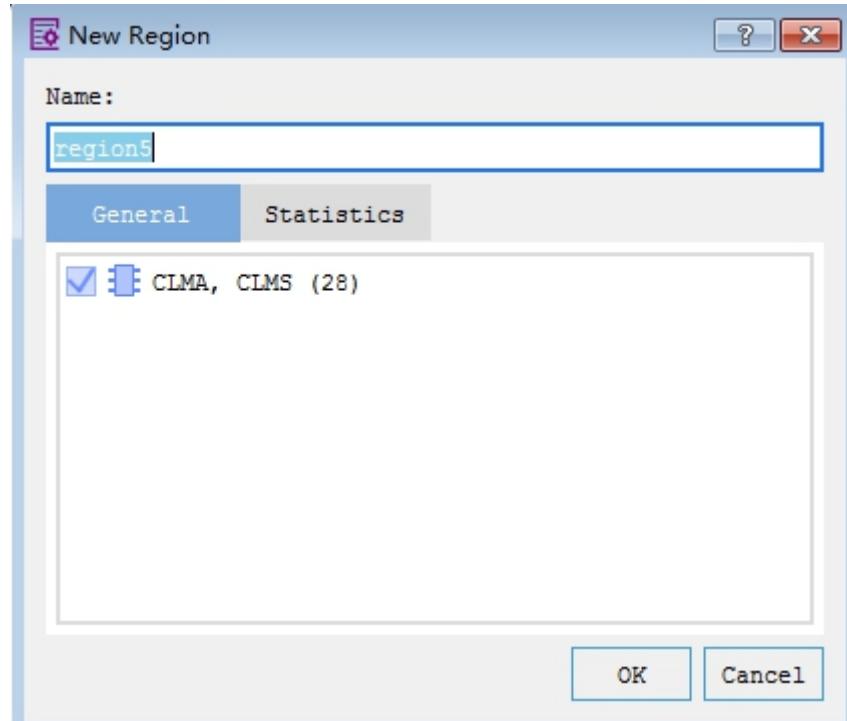


图 1-65 区域约束资源弹窗

下图是右键 draw region 画出了相同的区域范围，region1 是根据 LUT、FF、DTM 画出的范围，region2 是根据 DRM(在弹窗中取消勾选了 LUT、FF)画出的范围。Region 范围内画图样式为 45 度网格线阴影类型，范围内由红色线条覆盖，下方黑体显示该 Region 的名称。

创建 region 后，在 region mode 下，在 floorplan 上可以通过拖动的 region 的位置的方式来更改 region 位置，也可以通过拖拽 region 边框的方式更改 region 的大小范围。

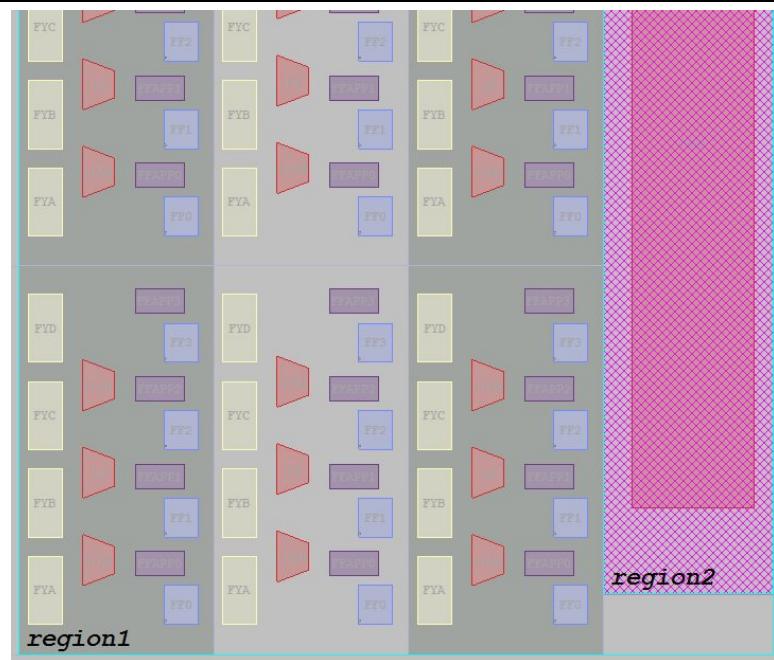


图 1-66 区域约束取消勾选结果

创建出区域约束之后，可将 instance 或 module 约束到区域当中，具体操作方法如下：

- 1) 点击按钮【Design Browser】，点击某 primitives 实例名或 module 名，按住鼠标左键，拖至 Floorplan view 上的区域中，若约束成功，则该被约束的 instance 或 module 高亮，实例名或 module 名前图标消失。存在两种冲突场景处理：子 module 存在约束时，父 module 约束区域会有弹框提示子 module 会被移除；父 module 已存在约束时，约束子 module 存在弹框提示子 module 会被忽略约束。
- 2) 也可选中的 instance 或 module 右键 Assign to Region/Unassign from Region，将选中的 instance 或 module 约束到指定的区域中或取消区域约束。

Floorplan View、Schematic 与 Hierarchy 界面中选中 instance 或 module 也支持 Assign to Region/Unassign from Region 功能。

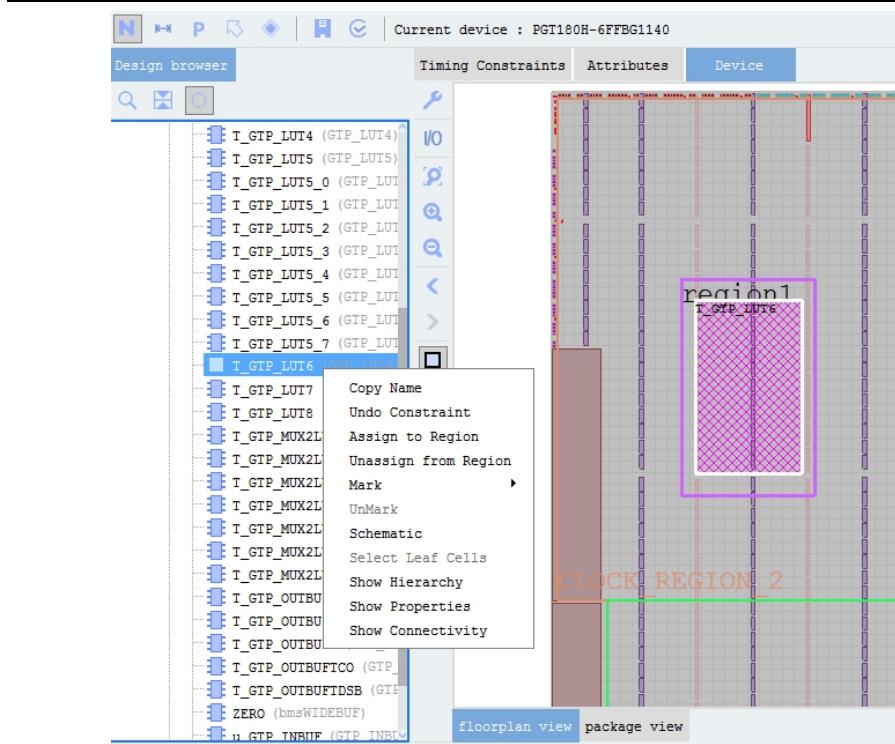


图 1-67 查看实例区域约束

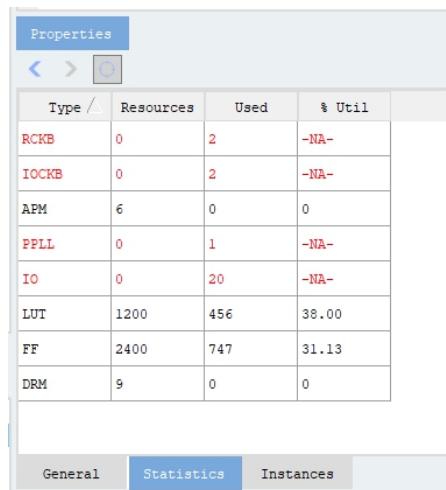
将 design inst 约束到 region 中，在打开 Properties 窗口，在 Properties 窗口下可以查看到 region 的各种属性。

在 General 栏中可以查看该 region 中可使用资源类型，并通过勾选对 region 中可使用资源进行过滤。



图 1-68 region 中 General 栏属性

在 Statistics 也可以查看各类型资源的使用数量以及使用率，在 Post syn uce 中 region 中资源使用率若超限，对应项会标红显示。

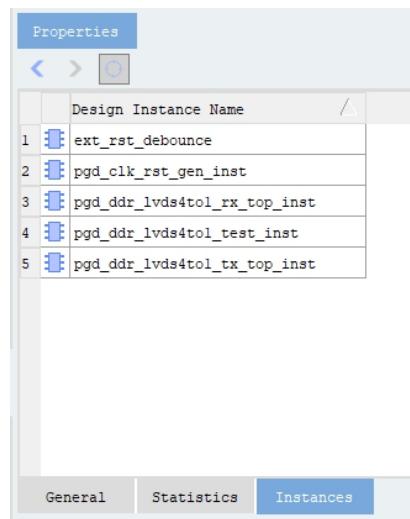


Type /	Resources	Used	% Util
RCKB	0	2	-NA-
IOCKB	0	2	-NA-
APM	6	0	0
PPLL	0	1	-NA-
IO	0	20	-NA-
LUT	1200	456	38.00
FF	2400	747	31.13
DRM	9	0	0

General    Statistics    Instances

图 1-69 region 中 Statistics 样属性

在 Instances 样中可以查看到该 region 中的约束对象。



	Design Instance Name
1	ext_RST_debounce
2	pgd_clk_RST_gen_inst
3	pgd_ddr_lvds4tol_rx_top_inst
4	pgd_ddr_lvds4tol_test_inst
5	pgd_ddr_lvds4tol_tx_top_inst

General    Statistics    Instances

图 1-70 region 中 Instances 样属性

#### (四) preserve 约束

在 floorplan view 窗口中，选择需要预留的位置（某个 Device Instance），右键选择 Set Preserve，此 Instance 即变为预留的状态。在此 Device Instance 位置右键选择 unset preserve，则是取消预留位置。该操作对应 PAP\_SITE\_PRESERVE 属性。

下图所示为预留位置以后的 device 图形，右键菜单分别为设置 preserve 和取消 preserve。预留以后该 device 框样式为 45 度网格线阴影类型，范围内由白色线条覆盖。下方黑体显示该 device 的名称。

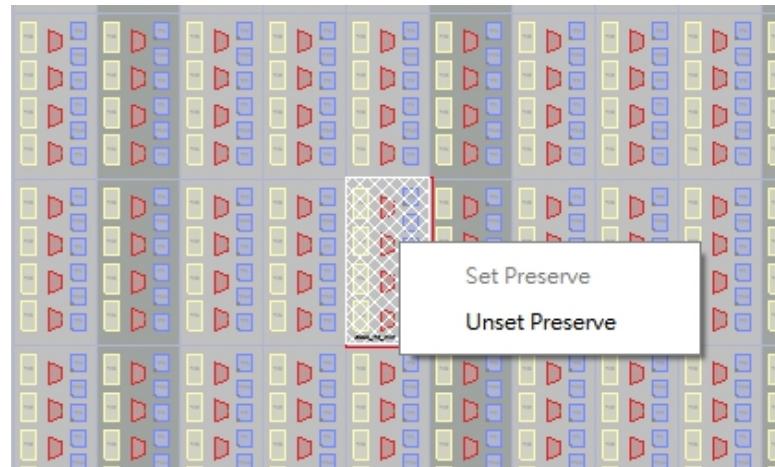


图 1-71 preserve 约束

#### (五) exclusive region 约束

在 floorplan view 窗口中，选择某个 region，右键选择 Set Exclusive Region 和 Unset Exclusive Region，对应为将 region 设置为 Exclusive region 和取消 Exclusive region。该操作对应 PAP\_EXCLUSIVE\_REGION 属性。

下图为设置该属性后的 Region 图形，设置该属性的 Region 样式为密集实粒阴影类型，范围内由白色粒子高度密集覆盖。下方黑体显示该 Region 的名称，Region 前有标示“RM:”表明该属性用于局部动态重配。



图 1-72 exclusive region 约束

#### (六) clock region 显示

在 floorplan view 视图上，根据时钟区域进行划分，标出 clock region 的范围，并实时绘制在 floorplan view 视图上。方便确认位置、编辑区域约束等操作。

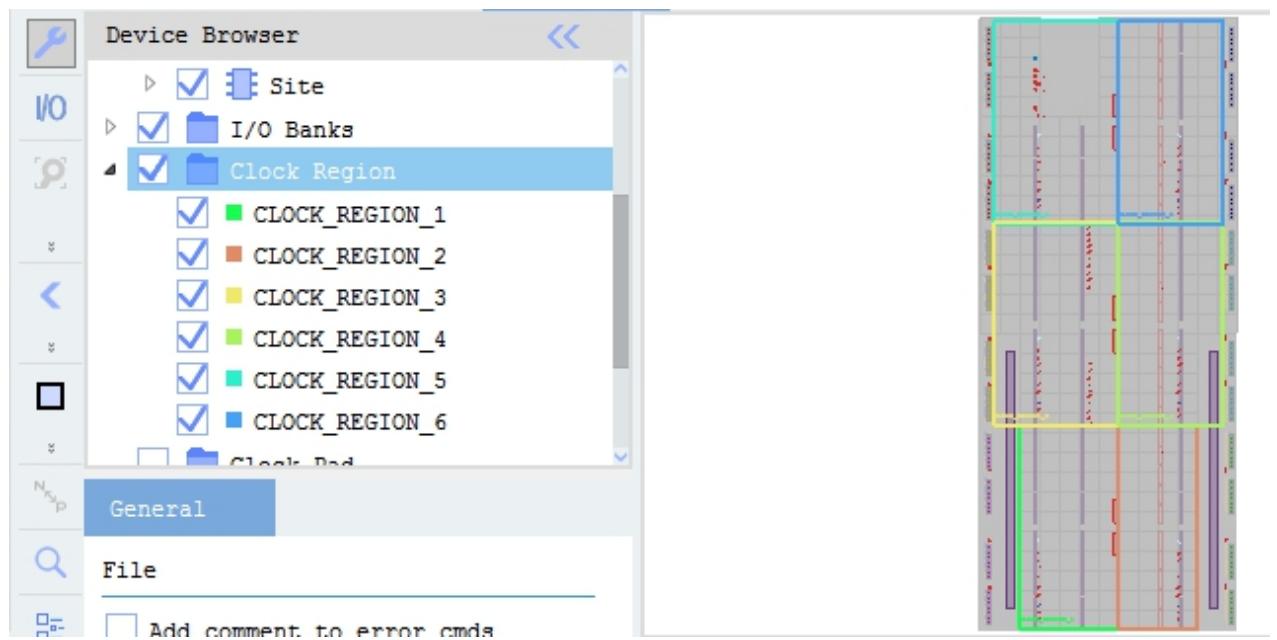


图 1-73 clock region 显示

Clock Region 画图显示如上图右侧所示。不同 Clock Region 有颜色上的区分，边框的颜色不一样。每个 Clock Region 左下角，会标记出 Clock Region 的名字。上图左侧的 Device Browser 对应有 Clock Region 项，用于过滤右侧 floorplan view 中的实时绘制。默认状态下是全部勾选，表示实时绘制所有的 Clock Region。可通过单个或者全部取消勾选的方式，来控制是否实时绘制某个 Clock Region 或者全部的 Clock Region。

为避免区域重叠等显示问题，画图时只按照 CLMA/CLMS 资源类型来进行划分，所以 Clock Region 的范围有一定的调整。

#### (七) Show Connectivity 功能

在 floorplan 上选中位置约束对象后，该对象高亮，且 floorplan 上若存在与该对象有连接关系的位置约束对象，会存在 connectivity 显示，如下图：



图 1-74 选中对象

箭头方向为输出方向。

选中该对象右键点击 show connectivity, floorplan 上与该对象存在连接关系的对象也被高亮选中, 如下图:

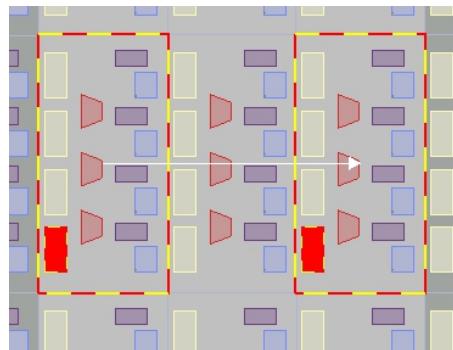


图 1-75 第一次 show connectivity

多次点击 show connectivity, 存在扇出效果, 如下图:

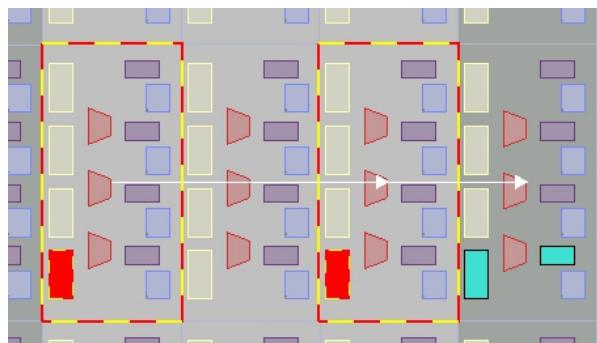


图 1-76 扇出效果

并且 show connectivity 的显示规则, 可以 device 界面工具栏中的 Show Cell Connections 来进行设置 (注: 重新打开组件后, 该项会恢复默认设置)

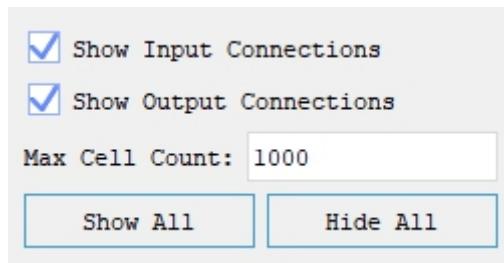


图 1-77 show cell connections 窗口

【show Input Connections】控制是否显示输入连接关系

【show output Connections】控制是否显示输出连接关系

【Max Cell Count】设置最大计数个数

【Show All】显示所有连接关系

【Hide All】隐藏所有连接关系

### (八) Bundle Nets 功能

floorplan 上的区域约束之间会显示 bundle nets，以此来显示区域之间的紧密程度，如下图：

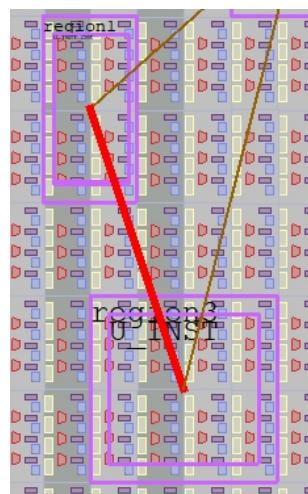


图 1-78 bundle nets 显示

选中显示的 bundle nets，打开 selection 窗口中，可以看到 Name 栏显示 bundle nets(\*)，其中\*代表 bundle net 数量，如下图：

Selection		
Id	Name	Type
1	Bundle Net(67)	Bundle Net

图 1-79 bundle net 选中

可以在 Device Browser 中的 bundle nets 项来控制各个等级的 bundle nets 是否显示。

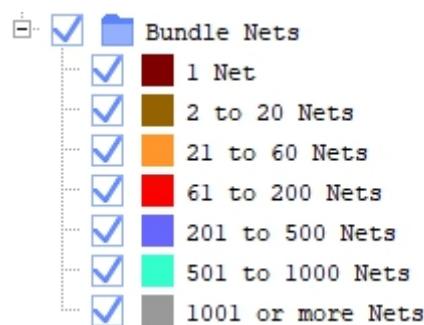


图 1-80 bundle net 设置

### (九) select leaf cells 功能

在 design browser 或者 floorplan 上选中对象，若对象存在 leaf cell，可以查看右键 select leaf cells 高亮可选

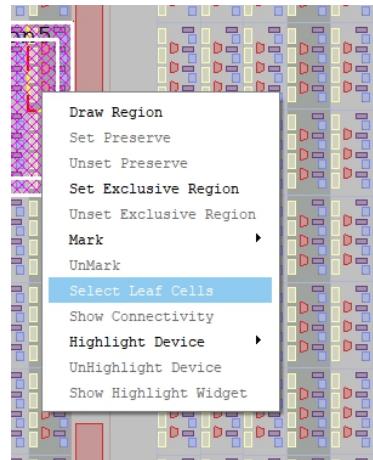


图 1-81 Select Leaf Cells

点击此项和可以选中该对象下所有的 leaf cell，并在 selection 窗口中刷新显示。

#### (十) Highlight 功能

Floorplan view 支持用户设置自定义 device 高亮颜色，右键新增菜单选项如下设置：

1. HighLight Selection(当有devcie选中时高亮，设置颜色可设置一个或多个选中的device)

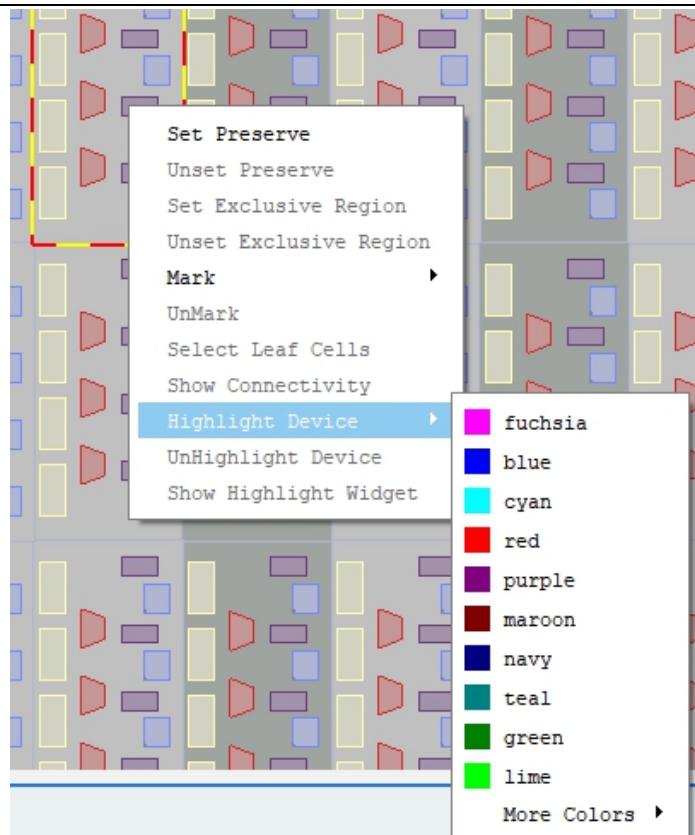


图 1-82 HighLight Selection

2. UnHighlight (存在设置高亮的device时高亮, 可以取消一个或多个设置高亮的device的高亮颜色)
3. Show Highlight List (存在显示高亮配置界面上有设置高亮的device时, 该选项高亮显示)

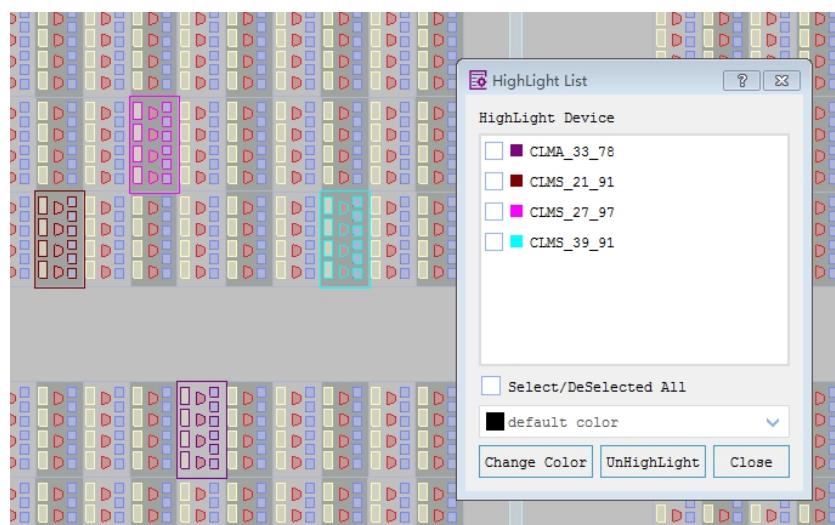


图 1-83 Show Highlight List

## (十一) Mark 功能

在 floorplan 中，支持对 instance 资源进行 mark，mark 后会有“菱形”标识。与此同时，在 mark browser 会同步 mark 信息，并以颜色区分。在 mark browser 能够定位某个以 mark 的资源位置，并且能够将不需要 mark 的资源移除 mark browser。

具体操作如下所示：

1.对于选中资源，右键选择 Mark/Unmark

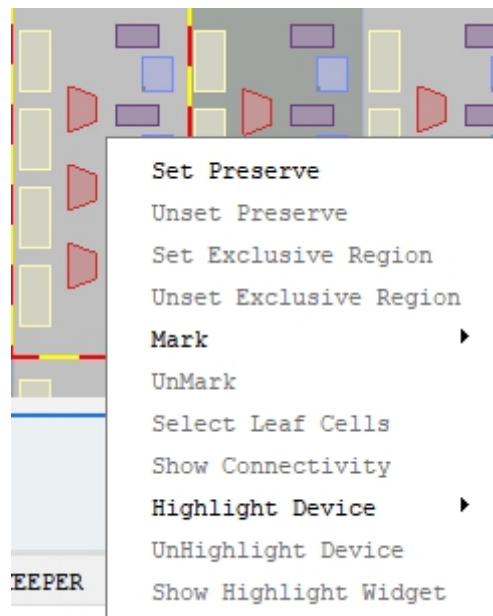


图 1-84 选中 device 右键

2. Mark 资源效果如下

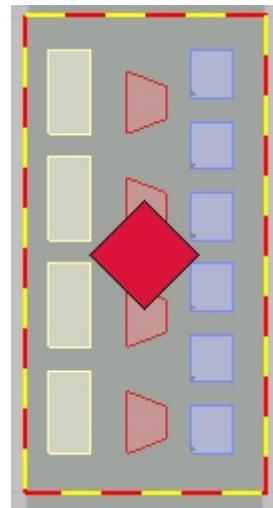


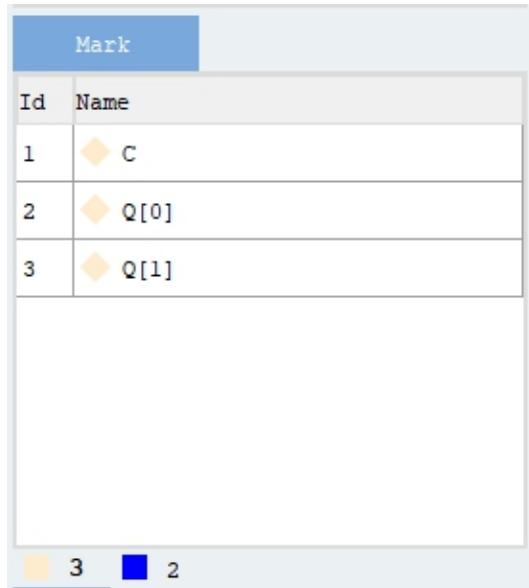
图 1-85 mark 效果

3. 在工具栏有按钮菱形控制是否显示 Mark browser 窗口



图 1-86 mark browser

4. 显示 Mark browser 如下



ID	Name
1	C
2	Q[0]
3	Q[1]

图 1-87 mark 浏览器界面

5. 在 Mark browser 能够移除掉不需要的资源，具体做法是选中资源右键菜单单选中 Unmark 功能，或者点击 Unmark all 清除全部的 mark 标记。

支持 mark 操作的视图还有 design browser、package view、IO table，且 mark 具有通信功能。即在 design browser 中选中 inst 进行 mark，mark browser 中会显示改 inst 名，再将该 inst 进行 LOC 约束，floorplan 界面中给 inst 上存在相同的 mark 标记，且在其中一处禽畜 mark 标记，其他相关通信界面该 inst 的 mark 标记同样清除。（IO table、IO tree、package view、mark browser 的 mark 通信同理）

#### (十二) Select 功能

在 floorplan 上可以通过点击右键 select 功能来选中切换鼠标处位置的选中对象。

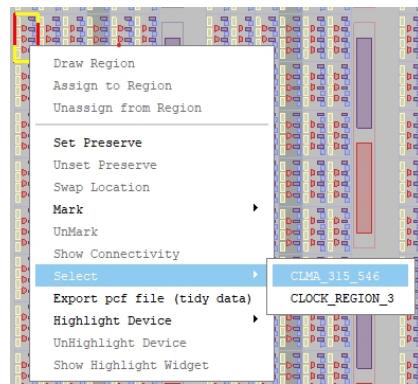


图 1-88 Select 功能

### 1.3.8 Schematic 界面

Schematic 界面主要用来查看对象的原理图，如下图

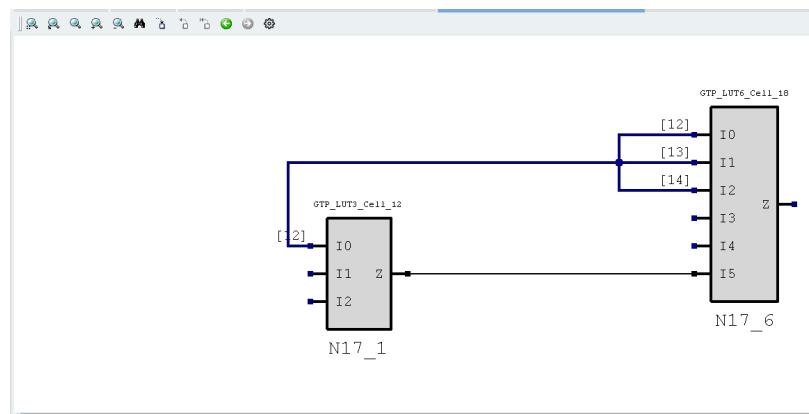


图 1-89 schametic 窗口

在 design browser、IO table、floorplan 中选中单个或者多个同层级 design inst, 右键菜单中存在 schematic 选项，点击后可以在 UCE 中打开选中对象的原理图。

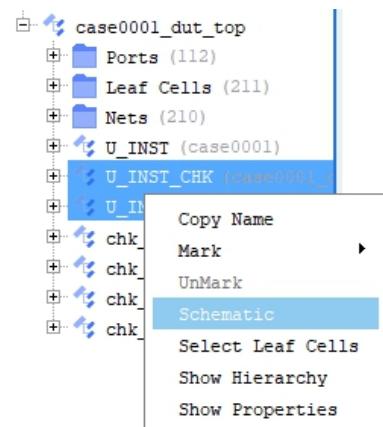


图 1-90 Schematic 选项

Schematic 页面的功能与综合网表中功能相同，可以查阅：

《ADS\_Synthesics\_User\_Guide》手册中的第 3 章，ADS 综合网表分析

### 1.3.9 Hierarchy 界面

UCE 中新增 show hierarchy 功能，该功能会添加新的 tab 页到 UCE 中，并显示 hierarchy 内容。具体显示 module inst 内的 instance 数量及 hierarchy 层级关系。如下图所示，Design browser 中右键，新增右键菜单 show hierarchy。

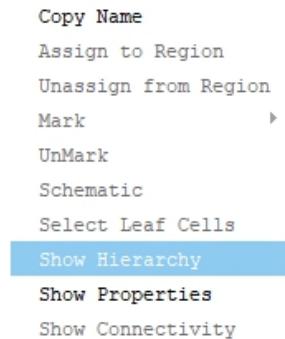


图 1-91 Show Hierarchy

点击右键 show hierarchy，则在 UCE 中新增 tab 页。该界面显示选中的 module inst 或者顶层内的所有 hierarchy 结构。该界面右边可根据 design 资源进行过滤显示。该界面上面有功能栏，可切换显示模式等。



图 1-92 Hierchay 界面

### 1.3.10 Selection 界面

UCE 中提供了 Selection 界面，供用户查询，筛选选中的对象。可以在 UCE

的工具栏中点击 selection 按钮打开 Selection 界面，见下图：



图 1-93 selection 按钮

打开 selection 窗口后在 design browser, Devcie, IO table 等界面选中单个或者多个对象，选中对象的类型以及名称会以表格形式显示在 selection 窗口中，若存在 Name 太长无法显示完全鼠标悬浮于该栏可出现冒泡窗口显示全名，见下图：

Selection		
ID	Name	Type
1	pgd_ddr_lvds4tol_test_inst/...	Instance
2	pgd_ddr_lvds4tol_test_inst/...	Instance
3	pgd_ddr_lvds4tol_test_inst/...	Instance
4	pgd_ddr_lvds4tol_1_pgddr_lvds4tol1_test_inst/prbs_txdata_gen/N8	

图 1-94 selection 窗口显示

在 Selection 窗口中点击其中一个选中对象，点击右键存在以下功能：

Selection		
ID	Name	Type
1	pgd_ddr_lvds4tol_test_inst/	Instance
2	pgd_ddr_lvds4tol_t	
3	pgd_ddr_lvds4tol_t	
4	pgd_ddr_lvds4tol_t	

Show Properties  
 Unselect  
 Unselect All  
 Unselect All Except

图 1-95 Selection 窗口右键功能

**【Show Properties】** 显示该选中对象的 Properties

**【Unselect】** 取消该对象的选中

**【Unselect All】** 取消全部对象的选中

**【Unselect All Except】** 取消除了该选择对象外其他对象的选中

### 1.3.11 Properties 界面

UCE 中提供了 Properties 界面，供用户查看选中对象的基本属性值。可以在

该窗口在 UCE 的工具栏中点击 Properties 按钮打开 Properties 界面，见下图：



图 1-96 Properties 按钮

Properties 界面会显示选中对象的基本属性，如下图：

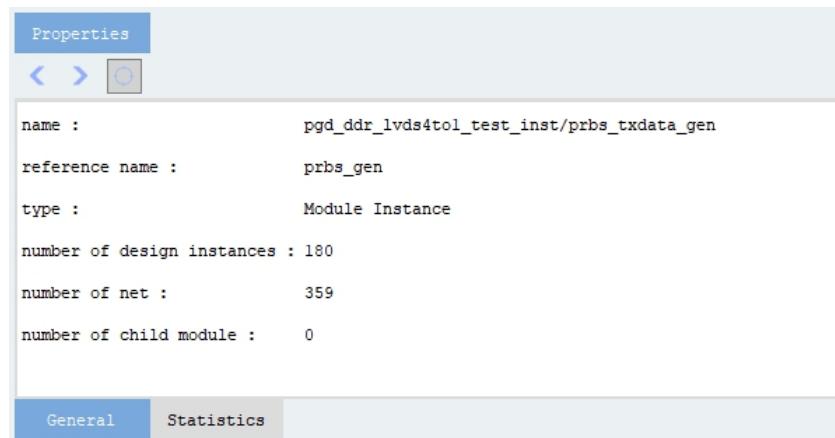


图 1-97 Properties 界面显示

且 Properties 的工具栏提供了一些操作工具共用户保持，切换查看 Properties，这些工具从左到右依次为 previous object, next object, Automatically update。

**【previous object】** 切换显示上一个对象的 Properties

**【next object】** 切换显示下一个对象的 Properties

**【Automatically update】** 自动更新成选中对象的 Properties，默认为开启状态

不同的对象 Properties 页面中显示的属性不同，具体划分见下表：

选中对象	显示页面	子页面显示内容
region	General	显示 region 中资源类型（可选择）
	Statstics	显示 region 中各类型资源数量
	Instances	显示约束在 region 中的 design inst
preserve region	Statstics	显示 region 中各类型资源数量
Exclusive region	Statstics	显示 region 中各类型资源数量
gate devcie inst	General	显示 name, unit, cell, view, site(x, y), (x, y) 像素坐标
prim device inst	General	显示 name, parent,unit,cell
top view	General	显示 name, reference name, type, design inst number, net numbr, child module number
	Statstics	显示该设计所用基本资源数量

module inst	General	显示 name, reference name, type, design inst number, net number, child module number
	Statstics	显示该 module 所用基本资源数量
inst	General	显示 Cell Paramarters, name, unit, cell, view, file, connected, pin out 信息
	nets	显示该 module 的 pin 以及连接的 nets
port	General	显示 name, file, direction
port bus	General	显示 name, file, direction, left_index, right_index, size
net	General	显示 name, file, pin port count
	Connectivity	显示该 net 连接的 pin 或 port
net bus	General	显示 name, file, left_index, right_index, size
pin pad	General	显示 name, unit, cell, view, site(x,y), (x,y)
bank	General	显示 name, (ref pin size), pin size, (ref pin name)
	Site	显示 bank 中各 Pin 的 pin name, Instance name, Pad name
Bundle nets	General	显示 name
Clock Region	General	显示 name, unit, cell, view, site(x,y), (x,y)

表 1-1 Properties 属性显示

## 1.4 解析差异造成的 FDC 修改

该节用于说明解析的差异造成的必要的 FDC 文件的修改，以及对该部分修改的一致指导意见。

综合工具为 OEM 时，UCE 读写和编辑的是综合之前的约束，此时打开 UCE 是使用 ADS 对网表进行解析。由于当前 ADS 和 Synplify Pro 均作为独立的综合工具，其解析结果存在差异。存在的差异可能会导致以下场景的发生。

1. 在 UCE 中用搜索工具进行搜索对象并确保将搜索到的对象保存到 FDC 文件中。流程中的 Synthesize 会报错不识别该对象或该对象不存在，导致约束失败。该种情况下，需要先确认是否是解析差异造成的。可以在综合后的 vm 文件中进行搜索，确认是否有约束的对象。若没有该约束对象，则认为是解析差异导致，可根据 vm 文件或者 RTL view 手动调整约束的内容。

另外，synthesize/synplify\_impl/synplify\_cck.rpt 文件是 OEM 给出的约束检查报告文件，该文件中可以查看约束检查的报错信息，便于定位问题。修改后通过该文件确认约束是否修改正确。

例如：形式如宽位寄存器等对象，在 UCE 中搜索的结果是 rdt\_len[6:0]，无

---

法搜索出具体的单个对象。此时约束 rdt\_len[6:0]在流程中会报对象不存在。所以需要查看综合后的 vm 文件，找出具体的单 bit 名称。通常名称的前面部分都是一样的，可以通过搜索 rdt\_len 来进行查找。一般 Synplify Pro 对宽位 instance 展开后名称并不一定是 rdt\_len[1]，可能形式会类似于 rdt\_len\_1[6]。找到然后替换掉约束文件中错误的对象名称即可。

2.流程中能跑通，没有 ERROR。但是打开 UCE，会报对象不识别或者对象不存在。

该种情况下，在 UCE 的搜索工具中对报错的对象进行搜索，如果没有搜索到，说明是差异造成，此时可以忽略打开 UCE 时的报错。

## 2.UCE 约束命令介绍

本章所描述的都是保存在约束文件里的约束命令，这些命令的各个 option 与界面表格里填写内容的对应关系在第一章 UCE 界面功能介绍里有说明。

### 2.1 对象描述规则

1、集成版： 在 fdc 或 UCE 中构建约束对象时，instance 名字与子 module 实例的名字之间用分隔符“.”来区分， instance 名字和 pin 脚的名字之间用分隔符“.”或者“/”来区分。

例：

```
create_generated_clock -name {CLKA} -source [get_ports {clka_in}] [get_pins {pll8_u1.I_GTP_PLL/CLKOUT0}] -master_clock {CLKA_IN} -multiply_by {16}
```

2、后端版： 在 sdc 或 lcf 中构建约束对象时，instance 名字与子 module 实例的名字之间用分隔符“/”来区分， instance 名字和 pin 脚的名字之间用分隔符“:”来区分。

例：

```
create_generated_clock -name {CLKA} -source [get_ports {clka_in}] [get_pins {pll8_u1/I_GTP_PLL:CLKOUT0}] -master_clock {CLKA_IN} -multiply_by {16}
```

3、如果对象名字有重复，如 net 名与 instance 名一样，为了区分指定对象，可以加前缀来区分：

“p:” 表示 port

“n:” 表示 net

“t:” 表示 pin

“i:” 表示 instance

4、如果约束对象位于 generate 循环过程中，对于 begin 后面有 ": label" 的情况， external name 就是 label[idx] 的形式；

对于 begin 后面没有 label 的情况，规定默认前缀是 genblk，后面加上当前

## 2.UCE 约束命令介绍

---

是第几个 generate block, 如 genblk1, genblk2, 对于 generate for, 则再加上 [idx], 合起来就是类似 genblk2[1]。例如:

```
genvar i;  
for(i = 0; i<PORTNUM; i = i+1)  
begin : mod_loop  
  
    always @ (posedge clk_bmu)  
        if(cs == 1'b0 && addr[15:12] == i)  
            mod_cs[i] <= #1 1'b0;  
        else  
            mod_cs[i] <= #1 1'b1;  
  
        ge_mod_pango #(  
            .JUMBO_MODE (JUMBO_MODE[i]),  
            .ETH_MODE (ETH_MODE[i]),  
            .MANAGEMENT_INTERFACE ("TRUE"),  
            .AUTO_NEGOTIATION ("TRUE"),  
            .CLOCKING_LOGIC ("TRUE"),  
            .CLOCKEN ("TRUE"),  
            .SIM_MODE (SIM_MODE),  
            .WR_SLOT (WR_SLOT),  
            .RD_SLOT (RD_SLOT),  
            .TA_SLOT (TA_SLOT),  
            .TX_POLARITY (TX_POLARITY),  
            .RX_POLARITY (RX_POLARITY)  
        ) u_ge_mod_pango (  
            .clk_bmu(clk_bmu),  
            .cs(mod_cs[i]),
```

2.UCE 约束命令介绍

---

```
.we(we),  
.oe(oe),  
.data_in(data_in),  
.addr(addr[11:0]),  
.data_out(data_out_mod[16*i+15:16*i]),  
  
.ta(mod_ta[i]),  
.debug_rxin_out ( ), // O  
.debug_rxout_out ( ), // O  
.pcs_clk_bufg (pcs_clk_bufg[i]),  
.external_rstn (external_rstn ), // I  
.tx_rstn (tx_rstn[i]), // I  
  
.status_vector (status_vector[7*i+6:7*i] ),  
  
.phy_link (phy_link ), // I  
.phy_duplex (phy_duplex ), // I  
.pin_cfg_en (pin_cfg_en ), // I  
  
.sgmii_clk (sgmii_clk[i]),  
.sgmii_rx_clk (sgmii_rx_clk[i]),  
  
.tx_tdata (tx_tdata[8*i+7:8*i] ),  
.tx_tfirst (tx_tfirst[i]),  
.tx_tvalid (tx_tvalid[i]),  
.tx_tlast (tx_tlast[i]),  
.tx_areset (tx_areset[i]),  
.tx_tready (tx_tready), // O  
.rx_tdata (rx_tdata[8*i+7:8*i] ),
```

2.UCE 约束命令介绍

---

```
.rx_tfirst (rx_tfirst[i]),  
.rx_tvalid (rx_tvalid[i]),  
.rx_tlast (rx_tlast[i]),  
.rx_areset (rx_areset[i]),  
.rx_tready (rx_tready[i]),  
  
//pcs rx  
.pcs_rxk (pcs_rxk[i]),  
.pcs_rdispdec_er (pcs_rdispdec_er[i]),  
.pcs_rxd (pcs_rxd[8*i+7:8*i]),  
.lsm_synced (lsm_synced[i]),  
  
.pcs_rx_clk_bufg (pcs_rx_clk_bufg[i]),  
  
//pcs tx  
.pcs_txk (pcs_txk[i]),  
.pcs_tx_dispctrl (pcs_tx_dispctrl[i]),  
.pcs_tx_dispsel (pcs_tx_dispsel[i]),  
.pcs_txd (pcs_txd[8*i+7:8*i]),  
.tx_clken (tx_clken[i]),  
.rx_clken (rx_clken[i]),  
  
.p_cfg_clk (p_cfg_clk),  
.paddr (5'd0),  
.pwrite (1'b0),  
.psel (1'b0),  
.penable (1'b0),  
.pwdata (16'd0),  
.prdata (),  
.hsst_ch_ready (hsst_ch_ready[4*i+3:4*i])
```

## 2.UCE 约束命令介绍

---

);

end // mod\_loop

上面例子中，如果想对 for 循环中的 u\_ge\_mod\_pango 对象做 PAP\_REGION 约束，对象预构建写成上面红色部分的字样加上 index 坐标，即：

mod\_loop[index]. u\_ge\_mod\_pango 其中 index 是循环遍历的下标(0~PORTNUM)

如果没有写第三行 begin 后面的 mod\_loop，则构建标准为：

genblk1[index]. u\_ge\_mod\_pango

其中 genblk 是默认前缀，1 表示第 1 个循环，index 表示循环内的第几个对象的坐标(0~PORTNUM)。

## 5. 对象支持 tcl 内置命令、支持变量替换

```
for {set a 0} {$a < 10} {incr a} \
{
    define_attribute "i:T_GTP_LUT5_$a" {PAP_GROUP} "group$a"
}
```

上面的命令会生成 10 条命令，\$a 会被依次替换为 0~9

此命令在 UCE 界面若有修改或删除一条属性，回写到 fdc，for 循环展开显示命令，展开时做变量子命令替换（时序命令子命令会保留子命令的形式）。

## 2.2 时序约束

### 2.2.1 Clocks 命令

在树型结构 Clocks 中有 create\_clock、create\_generated\_clock、set\_clock\_latency、set\_clock\_uncertainty 和 set\_clock\_groups 五条命令：

## 2.UCE 约束命令介绍

## 2.2.1.1 create\_clock

在 Clocks 窗口中可以设置 create\_clock 命令的参数，这个命令的作用是为时序分析创建一个时钟。需要注意的是用户在使用其他命令有引用 clock 的时候需要先定义该 clock。命令格式如下：

```
create_clock
    -period <period_value>
    [-name <clock_name>]
    [-waveform <edge_list>]
    [-add]
    [<source_objects>]
    [-disable]
```

命令中各个参数的含义说明如下：方括号表示该参数为可选值，尖括号表示参数值或参数列表。

**-period** : 定义创建时钟的周期，为浮点数，单位是 ns。

**[-name]** : 创建的时钟的名字，如果未定义，则一定要指定 source\_objects，同时名字和 source\_objects 中第一个参数的名字相同；如果用户未定义 source objects，则用户必须定义-name，此时创建的 clcok 是一个虚拟时钟。

**[-waveform]** : 定义时钟上升沿和下降沿的位置，其中上升沿在前，下降沿在后，成对存在，格式为：[-waveform] {time\_rise time\_fall}。

**[-add]** : 在同一个引脚上定义多个 clock，如果不加这个参数，那么后面定义的 clock 会将已定义的 clock 覆盖掉，同时软件将会给出警告。

**[<source\_objects>]** : 创建的时钟的源，可以是 pin 或 port(必须通过 get\_pins、get\_ports 得到)，当其为空时，创建虚拟时钟。

**[-disable]** : 使本条约束不生效。

## 2.2.1.2 create\_generated\_clock

在 Generated Clocks 窗口中可以设置 create\_generated\_clock 命令的参数，该命令的格式如下：

```
create_generated_clock
    [-name <clock_name>]
```

## 2.UCE 约束命令介绍

---

```
-source <master_pin>
[-master_clock <clock>]
[-multiply_by <factor>] [-divide_by <factor>]
[-duty_cycle <percent>]
[-edges <edge_list>] [-edge_shift <shift_list>]
[-invert]
[-add]
<source_objects>
[-disable]
```

命令中各个参数的含义说明如下：方括号表示该参数为可选值，尖括号表示参数值或参数列表。

**[-name]** : 创建的时钟的名字。

**-source**: 指定 source clock 的 pin 或者 port, 驱动该 generated clock。

**[-master\_clock]** : 指定 source clock, 在 master pin 上面有多个 source clock, 而 generated clock 只来自于其中一个 clock 时使用；当 master pin 上面只有一个 source clock 的时候，可以不指定该参数。

**[-multiply\_by]** : 倍频系数，倍频系数是一个大于0的整数，将时钟周期缩小指定倍数，倍频不止引起周期的变化，波形也将等比例缩小。

**[-divide\_by]** : 分频系数，分频系数是一个大于0的整数，将时钟周期扩大指定倍数。分频不止引起周期的变化，还会导致波形的变化；在PDS里，当分频系数为2以及2的指数倍时会将占空比设置为50%，其他系数值则保持原占空比不变。

**-divide\_by** 和 **-multiply\_by** 可以同时使用以做到非整数倍的频率变换，即小数分频。

**[-duty\_cycle]** : 单位是 percent (%) , 描述占空比，即高电平占整个周期的比例；占空比只能在使用**-multiply\_by** 的时候使用。

**[-edges]** : 表示从 source clock 引用的边沿的列表，每一个 edge 不能小于 1 且不能小于上个 edge, 和 create\_clock 中 waveform 的格式一致；不同的是最后还要加上第二个周期开始的位置，PDS 要求 edge\_list 中参数个数一定为 3 个，

## 2.UCE 约束命令介绍

里面不仅包含了第一个周期的上升沿和下降沿，还包含第二个周期的上升沿，软件能从中计算出 generated clock 的周期；例如-edges {2 4 6}，其中“2”表示 source clock 的第二个 edge，“4”表示 source 的第四个 edge.....依次类推；在 generated clock 里，“2”对应第一个上升沿位置，“4”表示第一个下降沿的位置，“6”表示下一个上升沿的位置。

**[-edge\_shift]**：指定了一系列浮点数，来表示每条边的偏移量，单位是 ns，必须和-edges 一起使用；如果 generated clock 需要实现相移功能，那么就需要用到“-edge\_shift”，“-edge\_shift”里面的列表元素个数必须和“-edges”里面的个数一致。

**[-invert]**：表示相对于 source clock，generated clock 的 clock 信号被取反。

**[-add]**：在同一个引脚上定义多个 clock，如果不加这个参数，那么后面定义的 clock 就会将前面的覆盖掉。

**<source\_objects>**：被创建 generated clock 的 pin，port 或者是 net。

**[-disable]**：使本条约束不生效。

注：

1. -edges/-edge\_shift 不能和-multiply\_by、-divide\_by、-duty\_cycle 同时使用。
2. 对于用户在 PLL 上创建的 create\_generated\_clock 约束，如果 PLL 配置了相移，那么约束命令就需要通过-edges/-edge\_shift 选项将 PLL 的相移信息体现在 generated clock 的波形上；软件自动在 PLL 上 infer 的时钟也通过上述方法体现 PLL 的相移延时。另外，用户需要注意时序分析路径上的时钟边沿对，有时可能需要通过 set\_multicycle\_path 约束进行调整。
3. create\_generated\_clock 命令支持 rename 功能，即用户只需要指定 create\_generated\_clock 中的-name，source\_objects，软件会自动 infer 出该时钟的周期和波形，同样也可以指定-source | -master\_clock 来筛选 source 时钟，该功能只支持 PLL 和 CLKDIV 的时钟输出端口。如果用户想要使用此功能，则不能指定如下选项：[-edges edge\_list] [-edge\_shift shift\_list] [-divide\_by factor] [-multiply\_by factor] [-duty\_cycle percent] [-invert]。用户可使用该功能重命名软件 infer 的 generated clock，并基于指定的名字进行后续的时序约束。命令示例如下：

```
create_generated_clock -name pll_clk [get_pins {pll.clkout1}]
```

## 2.UCE 约束命令介绍

## 1.2.1.3 set\_clock\_latency

在 Set Clock Latency 窗口中可以设置 set\_clock\_latency 命令的参数，该命令的格式如下：

set\_clock\_latency

[-clock <clock\_list>]

-source

<latency\_value>

[-early] [-late]

[-max] [-min]

[-rise] [-fall]

<object\_list>

[-disable]

命令中各个参数的含义说明如下：方括号表示该参数为可选值，尖括号表示参数值或参数列表。

**[-clock]** : 该条命令作用的clock。

**-source:** 指定latency的类型，目前只支持source latency，必须定义-source参数。

**<latency\_value>** : 延时值，为浮点数，单位是ns。

**[-min]** : 在bc\_wc的环境下，-min表示fast corner；-min和-max不能同时指定；不指定-min和-max，默认fast corner和slow corner命令都会生效。

**[-max]** : 在bc\_wc的环境下，-max表示slow corner；-min和-max不能同时指定；不指定-min和-max，默认fast corner和slow corner命令都会生效。

**[-early]** : 表示该latency是clock的左偏移量；-early和-late不能同时指定；不指定-early和-late，默认clock的左右偏移量都会生效。

**[-late]** : 表示该latency是clock的右偏移量，-early和-late不能同时指定；不指定-early和-late，默认clock的左右偏移量都会生效。

**[-rise]** : 表示该latency只应用在触发时钟边沿为上升沿的情况。

**[-fall]** : 表示该latency只应用在触发时钟边沿为下降沿的情况。

**<object\_list>** : 时钟定义点。可以通过get\_pins、get\_ports等方式获取对应

## 2.UCE 约束命令介绍

---

pin/port上定义的时钟，也可以直接将latency定义在clock上。

[-disable] : 使本条约束不生效。

#### 2.2.1.4 set\_clock\_uncertainty

在 Set Clock Uncertainty 窗口中可以设置 set\_clock\_uncertainty 命令的参数，该命令的格式如下：

```
set_clock_uncertainty
[<uncertainty_value>]
[-setup] [-hold]
[<object_list> |
 -from <from_clock>
   | -rise_from <rise_from_clock>
   | -fall_from <fall_from_clock>
   -to <to_clock>
     | -rise_to <rise_to_clock>
     | -fall_to <fall_to_clock>]
[-disable]
```

命令中各个参数的含义说明如下，方括号表示该参数为可选值，尖括号表示参数值或参数列表，| 表示或符号，同级和它的下一级中的几个选项只能使用一个：

[<uncertainty\_value>] : uncertainty 值，为浮点数，单位是 ns。

[-setup] : 表示 uncertainty value 只用在 setup 分析中。

[-hold] : 表示 uncertainty value 只用在 hold 分析中。

[<object\_list>] : 如果触发器时钟端的 pin 或者 port 上面定义了多个时钟，而用户只想为其中的一个时钟定义 uncertainty，那么 object 可以是用户定义的一个 clock；如果用户想为触发器时钟端的 pin 或者 port 上面的所有 clock 定义同一个 uncertainty，那么 object 可以是一个 pin 或者 port。

[-from] : 表示 launch clock，不考虑 clock 的极性。

[-rise\_from] : 表示只考虑 launch clock 为上升沿的情况。

## 2.UCE 约束命令介绍

[-fall\_from] : 表示只考虑 launch clock 为下降沿的情况。

[-to] : 表示 capture clock, 不考虑 clock 的极性。

[-rise\_to] : 表示只考虑 capture clock 为上升沿的情况。

[-fall\_to] : 表示只考虑 capture clock 为下降沿的情况。

[-disable] : 使本条约束不生效。

## 2.2.1.5 set\_clock\_groups

在 Set Clock Groups 窗口中可以设置 set\_clock\_groups 命令的参数, 该命令的格式如下:

```
set_clock_groups
[-name group_name]
[-asynchronous | -logically_exclusive | -physically_exclusive]
[-group <clock_list>]*
[-disable]
```

## 2.2.1.6 set\_external\_delay

Set\_external\_delay 命令用于设置一个 Output port 和一个 Input port 之间的外反馈路径延时，这个延时被用于 PLL 的外反馈路径延时计算。可以指定最大或最小值，默认情况下这个延时可以同时适用于最小（hold）和最大（setup）分析。命令格式如下：

```
set_external_delay
- from <Output port>
- to <Input port>
[-min][-max]
[-add]
<delay_value>
[-disable]
```

命令中各参数含义说明如下，方括号表示该参数为可选值，尖括号表示该参数为必选值。

-from: PLL 芯片外外反馈下的 Output port name。

## 2.UCE 约束命令介绍

-to: PLL 芯片外外反馈下的 Input port name。

[-min]: 表示设置的延时值是最小值。

[-max]: 表示设置的延时值是最大值。

[-add]: 表示可以在同一个外反馈路径上设置多个延时值，如果之前该路径上面已经定义了一个延时值，并且用户未加-add 参数，那么后面的定义的延时值就会将前面的覆盖掉。

<delay\_value>: 延时值，为浮点数，单位是 ns。

[-disable]: 使本条约束不生效。

## 2.2.2 Inputs/outputs 命令

在 Set Input Delay 和 Set Output Delay 的创建命令窗口和表格中，均可以设置“set\_input\_delay”和“set\_output\_delay”命令的参数。

## 2.2.2.1 set\_input\_delay

在 Set Input Delay 窗口中可以设置 set\_input\_delay 命令的参数，该命令的格式如下：

```
set_input_delay
  -clock <clock_name>
    [-clock_fall]
    [-rise] [-fall]
    [-max] [-min]
    [-add_delay]
    [-source_latency_included]
    <delay_value>
    <port_list>
    [-disable]
```

命令中各个参数的含义说明如下，方括号表示该参数为可选值，尖括号表示参数值或参数列表：

-clock : 表示其关联的 capture 时钟的名字。

## 2.UCE 约束命令介绍

[**-clock\_fall**] : 表示只考虑触发时钟为下降沿的情况。

[**-rise**] : 即只考虑 input 处的输入信号为上升沿的情况。

[**-fall**] : 即只考虑 input 处的输入信号为下降沿的情况。

注意：如果用户二者都没有定义，那么软件将同时考虑 input 处的输入信号的上升沿和下降沿情况。

[**-max**] : 表示用户定义的延时值是最大值。

[**-min**] : 表示用户定义的延时值是最小值。

注意：如果用户二者都没有定义，那么软件将用户定义的延时值同时在最大和最小延时分析时使用。

[**-add\_delay**] : 如果用户之前已经对该 input port 做了 set\_input\_delay 的约束，那么加这个参数表示软件将保留之前的设定的约束值。

[**-source\_latency\_included**] : 表示 clock source latency 已经算在 value 之内了，不用再计算。

<delay\_value> : 延时值，为浮点数，单位是 ns。

<port\_list> : input port，必须通过 get\_ports 得到。

[**-disable**] : 使本条约束不生效。

## 2.2.2.2 set\_output\_delay

在 Set Output Delay 窗口中可以设置 set\_output\_delay 命令的参数，该命令的格式如下：

```
set_output_delay
    -clock <clock_name>
    [-clock_fall]
    [-rise] [-fall]
    [-max] [-min]
    [-add_delay]
    [-source_latency_included]
    <delay_value>
    <port_list>
    [-disable]
```

命令中各个参数的含义说明如下，方括号表示该参数为可选值，尖括号表示参数值或参数列表：

-clock : 表示其关联的 launch 时钟的名字。

[-clock\_fall] : 表示只考虑触发时钟为下降沿的情况。

[-rise] : 只考虑 output 处的输入信号为上升沿的情况。

[-fall] : 只考虑 output 处的输入信号为下降沿的情况。

注意：如果用户二者都没有定义，那么软件将同时考虑 output 处的输入信号的上升沿和下降沿情况。

[-max] : 表示用户定义的延时值是最大值。

[-min] : 表示用户定义的延时值是最小值。

注意：如果用户二者都没有定义，那么软件将用户定义的延时值同时在最大和最小延时分析时使用。

[-add\_delay] : 如果用户之前已经对该 output port 做了 set\_output\_delay 的约束，那么加这个参数表示软件将保留之前设定的约束值。

[-source\_latency\_included] : 表示 clock source latency 已经算在 value 之内了，不用再计算。

<delay\_value> : 延时值，为浮点数，单位是 ns。

<port\_list> : output port，必须通过 get\_ports 得到。

[-disable] : 使本条约束不生效。

## 2.2.3 Assertio 命令

### 2.2.3.1 set\_max\_skew

在 Set Max Skew 窗口中可以设置 set\_max\_skew 命令的参数，该命令的格式如下：

```
set_max_skew
[-from <from_list> | -rise_from <rise_from_list> | -fall_from <fall_from_list>]
[-through <through_list>]*
[-to <to_list> | -rise_to <rise_to_list> | -fall_to <fall_to_list>]
<skew_value>
```

## 2.UCE 约束命令介绍

## [-disable]

命令中各个参数的含义说明如下，方括号表示该参数为可选值，尖括号表示参数值或参数列表，\*表示该参数可以重复，| 表示或符号，同级的多个选项只能使用一个：

[-from] : <from\_list>表示 path 的 startpoint，可以通过 get\_ports、get\_cells、get\_pins、get\_clocks 得到。

[-rise\_from] : 该参数只能指定为时钟 clocks，表示该时序路径起点被时钟上升沿触发，此处的时钟上升沿指的是时钟定义点的上升沿。

[-fall\_from] : 该参数只能指定为时钟 clocks，表示该时序路径起点被时钟下降沿触发，此处的时钟下降沿指的是时钟定义点的下降沿。

[-to] : <to\_list>表示 path 的 endpoint，可以通过 get\_ports、get\_cells、get\_pins、get\_clocks 得到。

[-rise\_to] : 该参数只能指定为时钟 clocks，表示该时序路径终点被时钟上升沿触发，此处的时钟上升沿指的是时钟定义点的上升沿。

[-fall\_to] : 该参数只能指定为时钟 clocks，表示该时序路径终点被时钟下降沿触发，此处的时钟下降沿指的是时钟定义点的下降沿。

[-through] : <through\_list>表示 path 经过的中间点，可以通过 get\_cells、get\_pins、get\_nets 得到。

<skew\_value> : 指定这条命令约束的一组 path 的最大 skew 值，为浮点数，单位是 ns。

[-disable] : 使本条约束不生效。

## 2.2.4 Exception 命令

在树型结构 Exception 中有 set\_max\_delay、set\_min\_delay、set\_multicycle\_path 和 set\_false\_path 四条命令：

## 2.2.4.1 set\_max\_delay

在 Set Max Delay 窗口中可以设置 set\_max\_delay 命令的参数，该命令的格式如下：

set\_max\_delay

## 2.UCE 约束命令介绍

---

```
[ -from <from_list> | -rise_from <rise_from_list> | -fall_from <fall_from_list> ]
[ -through <through_list> ]*
[ -to <to_list> | -rise_to <rise_to_list> | -fall_to <fall_to_list> ]
[ -datapath_only ]
<delay_value>
[ -disable ]
```

命令中各个参数的含义说明如下，方括号表示该参数为可选值，尖括号表示参数值或参数列表，\*表示该参数可以重复，| 表示或符号，同级的多个选项只能使用一个：

**[-from]** from\_list : from\_list 表示 path 的 startpoint，可以通过 get\_ports、get\_cells、get\_pins、get\_clocks 得到。

**[-rise\_from]** : 该参数只能指定为时钟 clocks，表示该时序路径起点被时钟上升沿触发，此处的时钟上升沿指的是时钟定义点的上升沿。

**[-fall\_from]** : 该参数只能指定为时钟 clocks，表示该时序路径起点被时钟下降沿触发，此处的时钟下降沿指的是时钟定义点的下降沿。

**[-through]** : <through\_list> 表示 path 经过的中间点，可以通过 get\_cells、get\_pins、get\_nets 得到。

**[-to]** : <to\_list> 表示 path 的 endpoint，endpoint 可以通过 get\_ports、get\_cells、get\_pins、get\_clocks 得到。

**[-rise\_to]** : 该参数只能指定为时钟 clocks，表示该时序路径终点被时钟上升沿触发，此处的时钟上升沿指的是时钟定义点的上升沿。

**[-fall\_to]** : 该参数只能指定为时钟 clocks，表示该时序路径终点被时钟下降沿触发，此处的时钟下降沿指的是时钟定义点的下降沿。

**[-datapath\_only]** : 表示指定的 max delay path 不关心时钟的 skew 和 jitter，只考虑 data path，包括第一个触发器的 CLK->Q 延时，第二个触发器的 setup time，以及两个触发器之间的组合路径延时。

**<delay\_value>** : 延时值，为浮点数，单位是 ns。

**[ -disable ]** : 使本条约束不生效。

## 2.UCE 约束命令介绍

## 2.2.4.2 set\_min\_delay

在 Set Min Delay 窗口中可以设置 set\_min\_delay 命令的参数，该命令的格式如下：

```
set_min_delay
[-from <from_list> | -rise_from <rise_from_list> | -fall_from <fall_from_list>]
[-through <through_list>]*
[-to <to_list> | -rise_to <rise_to_list> | -fall_to <fall_to_list>]
<delay_value>
[-disable]
```

命令中各个参数的含义说明如下，方括号表示该参数为可选值，尖括号表示参数值或参数列表，\*表示该参数可以重复，| 表示或符号，同级的多个选项只能使用一个：

**[-from] from\_list** : from\_list 表示 path 的 startpoint，可以通过 get\_ports、get\_cells、get\_pins、get\_clocks 得到。

**[-rise\_from]** : 该参数只能指定为时钟 clocks，表示该时序路径起点被时钟上升沿触发，此处的时钟上升沿指的是时钟定义点的上升沿。

**[-fall\_from]** : 该参数只能指定为时钟 clocks，表示该时序路径起点被时钟下降沿触发，此处的时钟下降沿指的是时钟定义点的下降沿。

**[-through]** : <through\_list> 表示 path 经过的中间点，可以通过 get\_cells、get\_pins、get\_nets 得到。

**[-to]** : <to\_list> 表示 path 的 endpoint，endpoint 可以通过 get\_ports、get\_cells、get\_pins、get\_clocks 得到。

**[-rise\_to]** : 该参数只能指定为时钟 clocks，表示该时序路径终点被时钟上升沿触发，此处的时钟上升沿指的是时钟定义点的上升沿。

**[-fall\_to]** : 该参数只能指定为时钟 clocks，表示该时序路径终点被时钟下降沿触发，此处的时钟下降沿指的是时钟定义点的下降沿。

**<delay\_value>** : 延时值，为浮点数，单位是 ns。

**[-disable]** : 使本条约束不生效。

---

#### 2.2.4.3 set\_multicycle\_path

在 Set Multicycle Path 窗口中可以设置 set\_multicycle\_path 命令的参数，该命令的格式如下：

```
set_multicycle_path
[-setup] [-hold]
[-start] [-end]
[-from <from_list> | -rise_from <rise_from_list> | -fall_from <fall_from_list>]
[-through <through_list>]*
[-to <to_list> | -rise_to <rise_to_list> | -fall_to <fall_to_list>]
<path_multiplier>
[-disable]
```

命令中各个参数的含义说明如下，方括号表示该参数为可选值，尖括号表示参数值或参数列表，\*表示该参数可以重复，| 表示或符号，同级的多个选项只能使用一个：

**[-setup]**：表示 setup 分析时，时钟周期使用指定的倍数。Setup 指定之后，hold 也会受到影响。默认情况下，是使用 setup。

**[-hold]**：表示 hold 分析时，时钟周期使用指定的倍数。

**[-start]**：表示这个多周期的设置是和 launch clock 相关的。

**[-end]**：表示这个多周期的设置是和 capture clock 相关的。

**[-from] from\_list**：from\_list 表示 path 的 startpoint，可以通过 get\_ports、get\_cells、get\_pins、get\_clocks 得到。

**[-rise\_from]**：该参数只能指定为时钟 clocks，表示该时序路径起点被时钟上升沿触发，此处的时钟上升沿指的是时钟定义点的上升沿。

**[-fall\_from]**：该参数只能指定为时钟 clocks，表示该时序路径起点被时钟下降沿触发，此处的时钟下降沿指的是时钟定义点的下降沿。

**[-through] <through\_list>**：<through\_list> 表示 path 经过的中间点，可以通过 get\_cells、get\_pins、get\_nets 得到。

**[-to] <to\_list>**：<to\_list> 表示 path 的 endpoint，endpoint 可以通过 get\_ports、get\_cells、

## 2.UCE 约束命令介绍

get\_pins、get\_clocks 得到。

**[-rise\_to]** : 该参数只能指定为时钟 clocks, 表示该时序路径终点被时钟上升沿触发, 此处的时钟上升沿指的是时钟定义点的上升沿。

**[-fall\_to]** : 该参数只能指定为时钟 clocks, 表示该时序路径终点被时钟下降沿触发, 此处的时钟下降沿指的是时钟定义点的下降沿

**<path\_multiplier>** : 表示设置的周期数目, 是一个大于等于 0 的整数。当设置为 0 时, 表示指定的 path 为 zero cycle path。

**[-disable]** : 使本条约束不生效。

#### 2.2.4.4 set\_false\_path

在 Set False Path 窗口中可以设置 set\_false\_path 命令的参数, 该命令的格式如下:

```
set_false_path
  [-setup] [-hold]
  [-from <from_list> | -rise_from <rise_from_list> | -fall_from <fall_from_list>]
  [-through <through_list>]*
  [-to <to_list> | -rise_to <rise_to_list> | -fall_to <fall_to_list>]
  [-disable]
```

命令中各个参数的含义说明如下, 方括号表示该参数为可选值, 尖括号表示参数值或参数列表, \*表示该参数可以重复, | 表示或符号, 同级的多个选项只能使用一个:

**[-setup]** : 表示进行的是 setup 分析

**[-hold]** : 表示进行的是 hold 分析

**[-from]** : <from\_list> 表示 path 的 startpoint, 可以通过 get\_ports、get\_cells、get\_pins、get\_clocks 得到。

**[-rise\_from]** : 该参数只能指定为时钟 clocks, 表示该时序路径起点被时钟上升沿触发, 此处的时钟上升沿指的是时钟定义点的上升沿。

**[-fall\_from]** : 该参数只能指定为时钟 clocks, 表示该时序路径起点被时钟下降沿触发, 此处的时钟下降沿指的是时钟定义点的下降沿。

**[-through]** : <through\_list> 表示 path 经过的中间点, 可以通过 get\_cells、

## 2.UCE 约束命令介绍

get\_pins、get\_nets 得到。

[**-to**] : <to\_list> 表示 path 的 endpoint, 可以通过 get\_ports、get\_cells、get\_pins、get\_clocks 得到。

[**-rise\_to**] : 该参数只能指定为时钟 clocks, 表示该时序路径终点被时钟上升沿触发, 此处的时钟上升沿指的是时钟定义点的上升沿。

[**-fall\_to**] : 该参数只能指定为时钟 clocks, 表示该时序路径终点被时钟下降沿触发, 此处的时钟下降沿指的是时钟定义点的下降沿。

[**-disable**] : 使本条约束不生效。

## 2.2.5 Other 约束命令

## 2.2.5.1 set\_disable\_timing

Set\_disable\_timing 用于 disable 某段时序弧。命令格式如下:

```
set_disable_timing
[-from <from_pin_name>]
[-to <to_pin_name>]
<object_list>
[-disable]
```

命令中各参数含义说明如下, 方括号表示该参数为可选值, 尖括号表示该参数为必选值:

**-from**: 表示时序弧的起点, 只有 object\_list 指定的对象为 lib\_cell 和 Inst 时可用, 在指定 cell 中只 disable 起点为此 from\_pin 的 timing arc, from\_pin\_name 应对应 object\_list 中 lib\_cell 或 Inst 的 pin name。

**-to**: 表示时序弧的终点, 只有 object\_list 指定的对象为 lib\_cell 和 Inst 时可用, 在指定 cell 中只 disable 终点为此 to\_pin 的 timing arc, to\_pin\_name 应对应 object\_list 中 lib\_cell 或 Inst 的 pin name。

**<object\_list>**: 表示指定对象的 list, 对象可以为 lib\_cell、inst、pin、port, 目前不支持 hierarchical inst。

**[**-disable**]**: 使本条约束不生效。

### 2.3 逻辑约束

在 UCE 表格中 Attributes 和 Device 界面中可以编辑生成 define\_attribute 命令和 define\_global\_attribute 命令。如下图所示：

	Timing Constraints	Attributes	Device
	Enable	Object	
+			
-	1	<input checked="" type="checkbox"/>	

图 2-1 Attribute 表格

#### 2.3.1 define\_attribute 命令

define\_attribute 命令可以为单个 design 对象(port/instance/pin/net)设置属性。其命令格式如下：

define\_attribute : [-disable]

port/inst/pin/net name

attribute

attribute value

其中中括号中的值为可选值。

[-disable]：表示这条命令将不会被执行

port/inst/pin/net name：指定需要约束属性的对象，根据属性的不同，对象可以分别是 port name、pin name、net name 或 design inst name

attribute：指定所需要指定的属性，例如 PAP\_REGION

attribute value：指定对应属性的值

实例：

define\_attribute {i:T\_GTP\_LUT5\_4} {PAP\_REGION} {reg}

给 T\_GTP\_LUT5\_4 这个 design instance 指定一个 PAP\_REGION 属性，规定其必须放置区域 reg 内，该 reg 需要先由 define\_global\_attribute 声明才能使用。

define\_attribute {i:T\_GTP\_LUT5\_4} {PAP\_IO\_LOC} {CLMA\_63\_0}

给 T\_GTP\_LUT5\_4 这个 instance 指定一个 PAP\_IO\_LOC 属性，规定其必须约束在 floorplan 上的坐标 CLMA\_63\_0 上。

### 2.3.2 define\_global\_attribute 命令

define\_global\_attribute 命令可以设置全局属性。其命令格式如下：

define\_global\_attribute : [-disable]

attribute\_name

attribute\_value

其中中括号中的值为可选值。

[-disable]：表示这条命令将不会被执行

attribute\_name：属性名,目前只允许为 PAP\_REGION/PAP\_SITE\_PRESERVE /PAP\_CLOCK\_DEDICATED\_ROUTE 其中一种,且每种属性在保存文件中最多只允许添加一条命令。

attribute\_value：指定约束属性的多个对象，对象之间用分号隔开。若属性为 PAP\_REGION，则属性值为一个或多个 region，且属性采用先声明后使用的方式，如：先通过 define\_global\_attribute {PAP\_REGION}

{reg1(1,2,3,4);reg2(2,6,2,6)}[CLMA,CLMS]定义了两个区域 reg1,reg2，且标记支持资源类型为 CLMA, CLMS（该资源类型可以缺省），则后续可以使用 define\_attribute { i:T\_GTP\_LUT5\_4}

{PAP\_REGION} {reg1}来使用已经通过 define\_global\_attribute 声明的区域 reg1。若属性为 PAP\_SITE\_PRESERVE，则属性值为一个或多个 device instance，如 {CLMA\_6\_4;CLMA\_6\_7}。若属性为 PAP\_CLOCK\_DEDICATED\_ROUTE，则属性值为 TRUE/FALSE。

实例：

define\_global\_attribute {PAP\_REGION} {reg1(1,2,3,4);reg2(2,6,2,6)}

[CLMS, CLMA]

通过两个全局的区域约束属性定义了两个区域，分别是 reg1 和 reg2；

define\_global\_attribute {PAP\_SITE\_PRESERVE} {CLMA\_6\_4;CLMA\_6\_7}

定义两个 device instance 的全局约束属性，将两个 device instance 设置为不可用（预留），即约束和布局的时候都不会使用这两个 device instance。

### 2.3.3 define\_compile\_points 命令

define\_compile\_attribute 命令可以设置不需要重新综合的约束 Module。即需要重新综合整个 design 时，可以选择只重新综合与修改相关的 design，其他的模块不需要综合，以节省运行时间。其命令格式如下：

define\_compile\_points: [-disable]

[-common]

[modulename]

[-type ]

其中中括号中的值为可选值。

[-disable] : 表示这条命令将不会被执行

[-common] : 该 option 用于进一步描述该 Compile Points 相关的特性，目前忽略。

[modulename] : 表示不需要重新综合的 Module name。

[-type] : 表示 Compile Points 类型，共分为 black-box/hard/locked/soft 四种类型。

实例：

define\_compile\_point -comment {first time} {v:CLM\_LUT} -type {black\_box} 将用户 design 文件中的某个 CLM\_LUT module 设置为 compile points，该 module 将不会重新综合；

### 2.3.4 UCE 中属性的设置

UCE 支持对属性进行设置。属性保存文件根据是否使用综合工具、保存的文件类型不同，使用综合工具(集成版)时，属性会保存在约束文件(.fdc)中；不使用综合工具(后端版)时，属性会保存在逻辑约束文件(.lcf)中。属性的保存格式在两种情况下均使用 define\_attribute 命令格式。

设置 instance attribute 目前有两种用法：

1、在 UCE 的 Attribute 表格中设置属性

## 2.UCE 约束命令介绍

2、直接在.v 或.vm 里利用 verilog 语法进行设置

这两种方法设置的 attribute 都可以在 flatten 和 map 后通过 DB 文件传递下来。

## 2.3.4.1 PAP\_GROUP

**功能:** PAP\_GROUP 是作为 RTL 中自定义的 module 实例的 attribute，可以对逻辑进行分组，布局时会尽量将同组的逻辑放置在靠近的位置。

**对象:** 作用对象是 RTL 中自定义的一个 module 的实例。

**属性值:** 属性值的形式可以是符合命名标准的任意字符串。

**描述:** PAP\_GROUP 属性是通过 DB 文件向下传递的，作用对象是一个 module instance 实例，该 module instance 在 flatten 后的 top instance 以及 map 后转换的所有 gop instance 都会拥有该属性。

**使用说明:** 在 UCE Attribute 表格可以设置该属性。

**实例:**

```
define_attribute {u_top_module•sub_module} {PAP_GROUP} {group_01}
```

## 2.3.4.2 PAP\_CLOCK\_ASSIGN

**功能:** PAP\_CLOCK\_ASSIGN 属性的作用对象是 net、pin 或 port，在布局布线时会根据该属性的值来决定是否使用全局时钟（信号）、局部时钟、水平时钟等。

**对象:** 作用对象是 design net(未连接 port, 且存在 driver pin); port(非 output); design pin(必须是 output)。

**属性值:** 属性值的形式是 GTP\_CLKBUFR、GTP\_CLKBUFG、GTP\_CLKBUFX、GTP\_BUFGS 和 GENERAL。其中 GTP\_CLKBUFR (Compact 系列不支持) 表示走局部时钟，GTP\_CLKBUFG 表示走全局时钟，GTP\_CLKBUFX 表示走 HCKB (Logos2, Titan2 系列支持)，GTP\_BUFGS 表示走 USSMBUF (Compact 系列支持)，GENERAL 表示不走时钟路线。

**描述:** 用户可以将该 attribute 设置到 design net、port、design pin。根据设置的对象，最后都会 annotate 到对应的 net 和 instance(driver pin 所在的 instance)上。

## 2.UCE 约束命令介绍

**使用说明:** 在 UCE 的 Attribute 表格中可以设置该属性。

**表现形式:** DE 里可看到作用对象连接的是否与所设值一致。

**实例:**

```
define_attribute {n:net} {PAP_CLOCK_ASSIGN} {GTP_CLKBUFR}  
  
define_attribute {n:u_top_module•sub_module_net} {PAP_CLOCK_ASSIGN}  
{GTP_CLKBUFR}  
  
define_attribute {t:instance.pin} {PAP_CLOCK_ASSIGN} {GTP_CLKBUFG}  
  
define_attribute {p:port} {PAP_CLOCK_ASSIGN} {GENERAL}
```

## 2.3.4.3 PAP\_IO\_REGISTER

**功能:** PAP\_IO\_REGISTER 属性的作用对象是 register，在 map 时会根据该属性的值来判断是否将 register 与 IO pack 到一起。该属性不支持 option -disable 和-comment。

**对象:** 作用对象是可以与 IO 相连的 register。

**属性值:** 属性值的形式是 TRUE 和 FALSE。

**描述:** 用户可以将该 attribute 设置到 register 上，后台都全部设置为 port 连接的 buf 上。

**使用说明:** 在 UCE 中设置 PAP\_IO\_REGISTER 属性时，目前支持对属性对象和属性值进行检查。

**表现形式:** DE 里可看到作用对象 register 是否与 IO pack 到一起。

**实例:**

```
define_attribute {i:register} {PAP_IO_REGISTER} {TRUE}  
  
define_attribute {i:u_top_module•sub_module_inst} {PAP_IO_REGISTER}  
{TRUE}
```

## 2.3.4.4 PAP\_PLL\_FEEDBACK\_DELAY

**功能:** PAP\_PLL\_FEEDBACK\_DELAY 属性的作用对象是 PLL 实例，作用是配置 PLL 的反馈端的 delay。

## 2.UCE 约束命令介绍

**对象:** 作用对象是 PLL 实例。

**属性值:** 属性值的形式是数字加单位, 如: 120ns, 目前只支持 ns。

**描述:** 用户可以将该 attribute 设置到 PLL 实例上。

**使用说明:** 在 UCE 中设置 PAP\_PLL\_FEEDBACK\_DELAY 属性时, 目前仅 logos 系列支持。

**表现形式:** 无。

**实例:**

```
define_attribute {i:pll} {PAP_PLL_FEEDBACK_DELAY} {120ns}
```

## 2.3.4.5 PAP\_DONT\_TOUCH

**功能:** PAP\_DONT\_TOUCH 属性的作用对象是 GTP 类型的 inst 实例和 module 实例, 作用是保持该实例在 Device Map 一步不被改变。

**对象:** 作用对象是 GTP 类型的 inst 实例和 module 实例。

**属性值:** 属性值的形式是 TRUE 和 FALSE。

**描述:** 用户可以将该 attribute 设置到 inst 实例上。

**使用说明:** 在 UCE 中设置 PAP\_DONT\_TOUCH 属性时, 在 Device Map 一步该 inst 将不被优化不被改变。FF pack 的时候无视 DNOT\_TOUCH 属性。

**表现形式:** DE 里可看到该实例没有被优化掉。

**实例:**

```
define_attribute {i:GTP_LUT6} {PAP_DONT_TOUCH} {TRUE}
```

## 2.3.4.6 PAP\_LOC

**功能:** PAP\_LOC 是位置约束, map 时会转化为 pcf 中的位置约束命令 def\_inst\_site。

**对象:** 作用对象通常是一些可以 place 到特殊资源(如 APM)的一些 instance。

**属性值:** 属性值的形式是一个具体的 device instance 的名字。

**描述:** PAP\_LOC 属性是通过 DB 文件向下传递的, 作用对象通常是一些可以 place 到特殊资源的一些 instance, 在 map 的过程中, 这个属性会传递给这个 top 转化的 gop 对象。

## 2.UCE 约束命令介绍

**使用说明:** 在 UCE 中设置 PAP\_LOC 属性时，支持对属性对象和属性值进行检查。此命令只在约束文本中编写，界面操作可以在上述“Device 界面操作”章节查看。

**表现形式:** 在 UCE 的 Device 界面可以看到初始阶段的约束，最终属性的表现形式在 PCE 中可以看到，指定的约束对象会在属性值所指定的位置上被约束。

**实例:**

```
define_attribute {i:instance} {PAP_LOC} {APM_134_172}  
  
define_attribute {i:u_top_module.sub_module_inst} {PAP_LOC}  
{APM_134_172}  
  
define_attribute {i:instance} {PAP_LOC} {CLMA_5_4:FGA}
```

表示约束到某个具体的 LUT 上。

## 2.3.4.7 PAP\_REGION

**功能:** PAP\_REGION 是区域约束属性，dev\_map 步骤会将逻辑约束文件(.lcf)里的区域约束转化为物理约束文件 (.pcf) 中的区域约束命令 def\_region 和 def\_inst\_region。

**对象:** 约束对象是一个库单元 (GTP) 实例或者是 RTL 源码中定义的 module 的实例；如果仅仅定义一个区域，则不需要指定约束对象。

**属性值:** 属性值的形式是以 region 名字开头，然后加上(x1,x2,y1,y2)格式的坐标值，即：

region\_name(x1,x2,y1,y2)中间允许有空格，这些坐标值表示 floorplan 坐标系中位置 (x1,y1) 和位置 (x2,y2) 为对角线点的矩形区域，(x1,y1) 表示左下角的坐标，(x2,y2) 表示右上角的坐标，所以要求  $0 \leq x1 \leq x2$ ,  $0 \leq y1 \leq y2$ ，但 x1,x2,y1,y2 不能同时为 0，并且是合法的 floorplan 坐标

**描述:** PAP\_REGION 属性是通过 DB 文件沿着 flow 向下传递的，如果它的约束对象是一个 GTP instance，那么在 dev\_map 的过程中，这个属性会传递给这个由该 GTP 实例转化成的所有 GOP 实例；如果约束对象是一个自定义的 module 实例，那这个 module 实例 flatten 后的所有 GTP instance 以及这些 GTP instance

## 2.UCE 约束命令介绍

在 dev\_map 时转换成的所有 GOP instance 都会拥有该属性；全局的区域约束采用先声明后定义的使用方式。

**使用说明：**在 UCE 中设置 PAP\_REGION 属性时，支持对属性对象和属性值进行格式和内容方面的合法性检查。此命令只在约束文本中编写，界面操作可以在上述“Device 界面操作”章节查看。区域约束需要遵循，先声明后使用的原则，即：先定义区域后使用该区域。

**表现形式：**属性的约束效果在 UCE 的 Device 界面或者 PCE 中可以看到，指定的约束对象转换的 GOP instance 会在属性值所指定的区域被约束。

**实例：**

```
define_attribute {i:T_GTP_LUT5_4} {PAP_REGION} {region1}
define_global_attribute {PAP_REGION}
{region2(69,78,50,64)[CLMA,CLMS,APM,DRM];}

define_attribute {u_top_module.sub_module} {PAP_REGION}
{region(1,2,3,4)}
```

## 2.3.4.8 PAP\_SITE\_PRESERVE

**功能：**PAP\_SITE\_PRESERVE 会将指定的某个 device instance 设置为不可用（预留），即约束和 pnr 的时候都不会使用该 device。

**对象：**预留的 device instance。

**属性值：**多个具体的 device instance 的名字，形式为 {CLMA\_6\_4;CLMA\_6\_7}。

**描述：**PAP\_SITE\_PRESERVE 是将指定的 device instance 设置为不可用。

**使用说明：**在 UCE 中设置 PAP\_SITE\_PRESERVE 属性时，支持对属性对象和属性值进行检查。此命令只在约束文本中编写，界面操作可以在上述“Device 界面操作”章节查看。

**表现形式：**最终属性的表现形式是在 PCE 和 DE 中该指定的 device instance 都不能被使用。

**实例：**

## 2.UCE 约束命令介绍

---

```
define_global_attribute {PAP_SITE_PRESERVE} { CLMA_6_4;CLMA_6_7}
```

## 2.3.4.9 PAP\_CLOCK\_DEDICATED\_ROUTE

**功能:**

用于控制时钟走线。未设置该属性或该属性设置为 TRUE 的情况下不允许时钟走 SRB 之间的互连通道；如果希望时钟走 SRB 之间的互连通道，需要将 "PAP\_CLOCK\_DEDICATED\_ROUTE" 属性设置为 FALSE。

若时钟没有布局到专用时钟端口并最终上了时钟网络，PDS 流程运行到 place&route 会报错(报错编号为 E: Place-0084)，如果将该时钟相关的 net 或该 net 的驱动 pin 脚设置该属性为 FALSE，则 PDS 流程会在 place&route 中将报错降级为 Critical Warning (C: Place-2028)。

**对象:** output pin 或 net 或不指定对象(全局属性)。

**属性值:** TRUE/FALSE。

**描述:** PAP\_CLOCK\_DEDICATED\_ROUTE 随 DB 向下传递

在 place&route 阶段中的 place 阶段，当用户在做 IO 约束的过程中，如果将 clock 端口放置到了非专用时钟端口，则 place 阶段会给出报错提示从而停止 flow，如果用户希望 flow 继续运行，则可以将该时钟相关的 net 或该 net 的驱动 pin 脚设置该属性为 FALSE，或者直接将全局属性设置为 FALSE，之后流程就会将报错调整为警告，从而继续运行流程而不会停止。需要注意的是，在 Place 阶段中，对 net 或者 pin 脚设置的单条降级属性在优先级上大于全局属性的降级；

在 place&route 阶段中的 route 阶段，全局属性将不起作用，只有对单条 net、pin 设置降级属性，才会在详细布线阶段，检测到设置的 PAP\_CLOCK\_DEDICATED\_ROUTE 属性；

**使用说明:** 在 UCE 中设置 PAP\_CLOCK\_DEDICATED\_ROUTE 属性时，支持对属性对象和属性值进行检查。

**局部属性:** object 为与时钟 IO 相连 net，或者是 output 类型的 pin；attribute 为 PAP\_CLOCK\_DEDICATED\_ROUTE；value 值为 FALSE。

**全局属性:** object 为空，attribute 为 PAP\_CLOCK\_DEDICATED\_ROUTE；value 值为 FALSE(全局属性会有将应该报错的 error 降低级别到 critical warning，

## 2.UCE 约束命令介绍

---

所以需慎用）。

上述两种属性任意一种设置保存后，在 PNR 阶段就不会上报时钟端口没有从 clock pad 输入上时钟网络的 Error，只会报 Critical Warning（例如：C: Place-2028: GLOBAL\_CLOCK: the driver clk\_ibuf/opit\_1 fixed at IOLHR\_16\_1008 is unreasonable. Sub-optimal placement for a clock source and a clock buffer.）

**表现形式：**最终属性的表现形式为 PDS 流程在 place&route 阶段不会报错停止而是以报警告代替。布线可以选择是否走 SRB 之间的互连通道。

**实例：**

```
define_attribute {n:net} {PAP_CLOCK_DEDICATED_ROUTE} {FALSE}  
define_attribute {t:pin} {PAP_CLOCK_DEDICATED_ROUTE} {FALSE}  
define_global_attribute {PAP_CLOCK_DEDICATED_ROUTE} {FALSE}
```

#### 2.3.4.10 PAP\_KEEP\_CHAIN

**功能：**在 CarryChain 的 instance 上添加属性，控制 CarryChain 不折断。

**对象：**CarryChain 的 instance。

**属性值：**TRUE / FALSE。

**描述：**假设折断的条件是长度等于 200，默认行为是可能会折断：长度小于等于 200 不折断，大于 200 折断。参数设置 TRUE，所有长度都不折断。

**使用说明：**在 UCE 中设置 PAP\_KEEP\_CHAIN 属性时，支持对属性对象和属性值进行检查。

**实例：**

```
define_attribute {i:T_GTP_LUT2} {PAP_KEEP_CHAIN} {TRUE}
```

#### 2.3.4.11 PAP\_ALLOW\_RETIMING

具体描述见 ADS 用户手册 syn\_allow\_retimming 属性的用法。

2.UCE 约束命令介绍

#### 2.3.4.12 PAP\_ALLOWED\_RESOURCES

具体描述见 ADS 用户手册 syn\_allowed\_resources 的用法。

#### 2.3.4.13 PAP\_DSP\_STYLE

具体描述见 ADS 用户手册 syn\_dspstyle 的用法。

#### 2.3.4.14 PAP\_HIER

具体描述见 ADS 用户手册 syn\_hier 的用法。

#### 2.3.4.15 PAP\_MAX\_FANOUT

具体描述见 ADS 用户手册 syn\_maxfan 的用法。

#### 2.3.4.16 PAP\_MIN\_CONTROLSET\_SIZE

具体描述见 ADS 用户手册 syn\_reduce\_controlset\_size 的用法。

#### 2.3.4.17 PAP\_UNCONNECTED\_INPUTS

具体描述见 ADS 用户手册 syn\_uncounnnected\_inputs 的用法。

### 2.3.5 Device 中的 I\O Table 相关配置

Tool Tabs					
	I/O NAME	I/O DIRECTION	LOC	BANK	VCCIO
1	Q[0]	OUTPUT			
2	Q[1]	OUTPUT			
3	C	INPUT			
4	CE	INPUT			
5	CLK	INPUT			
6	D	INPUT			

图 2-2 I\O Table 表格

(1) UCE 的 IO Table 中能传递下去的属性

各个系列的器件的 UCE 的 IO Table 中能传递下去的属性包括：

## 2.UCE 约束命令介绍

属性名	TITAN	LOGOS	COMPACT	TITAN2	LOGOS2
PAP_IO_DIRECTION	Y	Y	Y	Y	Y
PAP_IO_LOC	Y	Y	Y	Y	Y
PAP_IO_VCCIO	Y	Y	Y	Y	Y
PAP_IO_STANDARD	Y	Y	Y	Y	Y
PAP_IO_DRIVE	Y	Y	Y	Y	Y
PAP_IO_PMOS_FINGER	Y	N	N	N	N
PAP_IO_NMOS_FINGER	Y	N	N	N	N
PAP_IO_SLEW	Y	Y	Y	Y	Y
PAP_IO_OPEN_DRAIN	Y	Y	Y	Y	Y
PAP_IO_IN_DELAY	Y	Y	Y	Y	Y
PAP_IO_OUT_DELAY	Y	Y	Y	Y	Y
PAP_IO_PULLUP	Y	Y	Y	Y	Y
PAP_IO_PULLDW	Y	Y	Y	Y	Y
PAP_IO_KEEPER	Y	Y	Y	Y	Y
PAP_IO_UNUSED	Y	Y	Y	Y	Y
PAP_IO_VREF_MODE	Y	Y	Y	Y	Y
PAP_IO_VREF_MODE_VALUE	Y	Y	Y	Y	Y
PAP_IO_DDR_TERM_MODE	Y	Y	Y	Y	Y
PAP_IO_DIFF_IN_TERM_MODE	Y	Y	Y	Y	Y
PAP_IO_REGISTER	Y	Y	Y	Y	Y
PAP_IS_VIRTUAL_IO	Y	Y	Y	Y	Y
PAP_IO_IPT	N	Y	Y	Y	Y
PAP_IO_CAL_MODE	N	Y	Y	N	Y
PAP_IO_DDR_RES	N	Y	Y	Y	Y
PAP_IO_DIFFOUT_DYN_EN	N	N	Y	N	N
PAP_IO_DIFFIN_DYN_EN	N	N	Y	N	N
PAP_IO_HYS_DRIVE_MODE	Y	Y	Y	N	Y
PAP_IO_OFF_CHIP_TERMINATION	Y	N	N	Y	Y
PAP_IO_CLAMP	N	N	Y	N	N

表 2-1 属性

这些属性分别将 IO Table 中的对应选项值以属性的形式记录在约束文件中或 design 文件中，并通过 DB 文件向下传递，在 map 后又会将这些属性转化为 pcf 文件中的 define\_port 命令。

实例：

```
define_attribute {p:GTP_LUT2_Z} {PAP_IO_LOC} {A17}
```

```
define_attribute {p:GTP_LUT2_Z} {PAP_IO_VCCIO} {3.3}
```

## 2.UCE 约束命令介绍

```
define_attribute {p:GTP_LUT2_Z} {PAP_IO_STANDARD} {LVCMOS33}
define_attribute {p:GTP_LUT2_Z} {PAP_IO_SLEW} {FAST}
define_attribute {p:GTP_LUT2_Z} {PAP_IO_DRIVE} {4}
define_attribute {p:GTP_LUT2_Z} {PAP_IO_PULLUP} {TRUE}
define_attribute {p:GTP_LUT2_Z} {PAP_IO_OPEN_DRAIN} {OFF}
define_attribute {p:GTP_LUT2_Z} {PAP_IO_REGISTER} {TRUE}
```

转化为 pcf 后：

```
def_port      GTP_LUT2_Z      LOC=A17      IOSTANDARD=LVCMOS33
VCCIO=3.3 PULLUP=TRUE  DRIVE=4  SLEW=FAST  OPEN_DRAIN=OFF
```

### (2) PAP\_IO\_REGISTER 属性说明

PAP\_IO\_REGISTER 属性并不会转化为 pcf 约束命令，该属性是用来设置该 port 是否使用 IO Register。

### (3) PAP\_IS\_VIRTUAL\_IO 属性说明

PAP\_IS\_VIRTUAL\_IO 属性即设置虚拟 IO，其相关的功能及使用方法如下：

- 1) 主要功能：在用户设计或移植工程初期阶段，用于评估 CLM 资源使用情况，此阶段不考虑 IO 数目和放置限制，故设置虚拟 IO。
- 2) 作用对象：只能为 IO。
- 3) 约束方式：

支持命令约束。在 fdc 中约束，如： define\_attribute {p:CLK} {PAP\_IS\_VIRTUAL\_IO} {TRUE}

支持界面约束。在 UCE 的 IO Table 中设置，如：勾选对应管脚的 VIRTUAL\_IO 项。

- 4) 属性特点：
  - a. 虚拟 IO 不会转化为 pcf 约束命令，且进入 PCE 界面时也不会加载该 IO。
  - b. 虚拟 IO 不会参与布局布线。
  - c. 禁止对虚拟 IO 做其他属性设置操作。
  - d. 只支持 PDS 的简单 PNR 流程，支持 DE 打开布局布线结果。

## 2.UCE 约束命令介绍

- e. 不支持 MPNR、timing 分析和报告以及位流生成。

(4) 在 IO Table 增加批量操作功能，该功能可以选中特定的列进行批量操作。多个 Port 进行批量操作时，可设值为多个 Port 对应值的交集，如果交集不存在，那么不能进行批量操作。其中 Clear Constraint 菜单的功能是清除高亮的约束，Clear All Constraint 菜单的功能是清除 IO Table 所有的约束，具体操作如下图所示

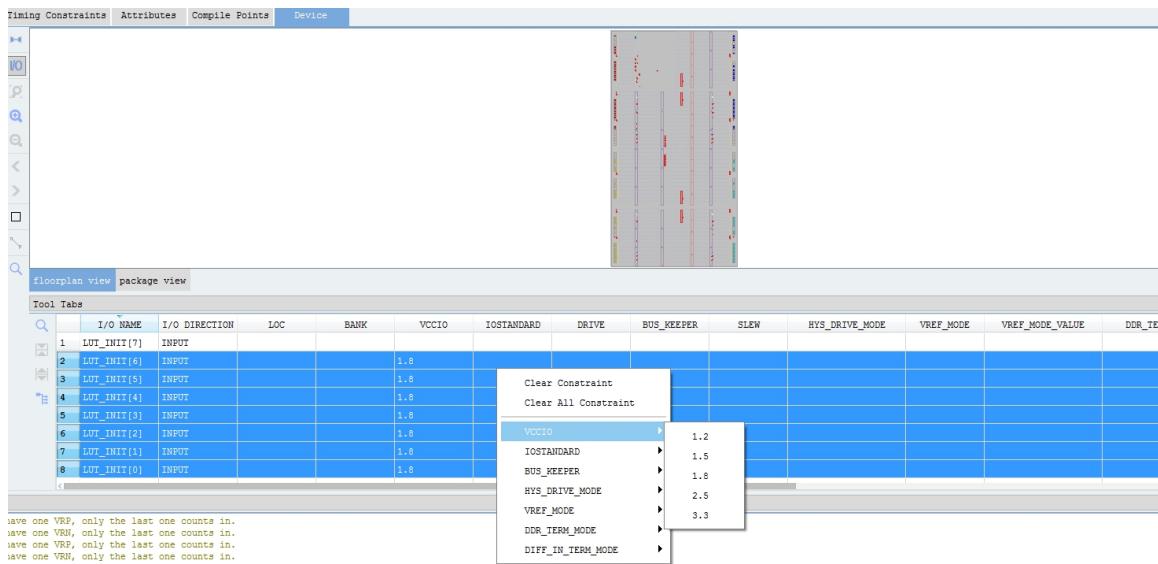


图 2-3 具体操作

(5) PG2T390H 新增了 3 个 IO 约束，分别是 PAP\_IO\_DIFFOUT\_CM、PAP\_IO\_PREEMP、PAP\_IO\_OUTPUT\_IMPEDANCE。具体作用如下图所示

Parameter name	Description
PAP_IO_DIFFOUT_CM	差分输出驱动器共模电平控制，包括：0.2V, 0.9V, 1.2V, 1.25V
PAP_IO_PREEMP	TX 预加重使能配置，包括：OFF, ON
PAP_IO_OUTPUT_IMPEDANCE	驱动器输出阻抗，包括： $27\Omega$ （部分支持）， $34\Omega$ , $40\Omega$ , $48\Omega$ , $60\Omega$ ，当 IO 约束到 R5/R6/R7 BANK 上，且 IOSTANDARD 选择支持该属性的值时才可设置

表 2-2 PAP\_IO\_PREEMP、PAP\_IO\_OUTPUT\_IMPEDANCE 作用

(6) PG2T390H 相同 HPIO BANK 接口标准兼容规则

### 1) 输出标准兼容规则

具有相同 VCCIO 的输出标准可以约束在相同 BANK；

例如：LVCMOS15、SSTL15D\_I 可以约束在相同 BANK；

### 2) 输入标准兼容规则

- a. 具有相同 VCCIO；

## 2.UCE 约束命令介绍

---

- b. 输入参考电压相同或是无需参考电压;

如：SSTL15、HSTL\_II、LVCMOS15可以约束在相同BANK;

### 3) 输入和输出标准兼容规则

- a. 输入标准和输出标准具有相同VCCIO;
- b. 输入标准具有相同参考电压或是无需参考电压;

例如：LVCMOS18输出和HSTL18\_I、SSTL18\_I、LVSMOS18输入可以约束在相同BANK;

### 4) 双向IOB标准兼容规则

满足前面3条兼容规则的可以约束在相同BANK;

例如：LVCMOS12、SSTL12、HSTL12\_I、HSUL12D双向标准可以约束在相同BANK;

### 5) 其他兼容规则

a. HPIO BANK: LVDS\_18输入可以和其他任何输入、输出、双向标准约束在相同BANK;

b. HRIO BANK: LVDS25、RSDS、PPDS、MINI-LVDS输入可以和其他任何输入、输出、双向标准约束在相同BANK;

(7) PGT180H 的 PAP\_IO\_DRIVE 与 PAP\_IO\_PMOS\_FINGER、  
PAP\_IO\_NMOS\_FINGER 之间的对应说明

PGT180H 可以选择设置 PAP\_IO\_DRIVE，也可以选择设置  
PAP\_IO\_PMOS\_FINGER 和 PAP\_IO\_NMOS\_FINGER，它们之间存在的对应关系  
详见下表：

Finger	PGT180H的IOB驱动器P管/N管finger数量对应的电流(mA)				
	Vccio=3.3V	Vccio=2.5V	Vccio=1.8V	Vccio=1.5V	Vccio=1.2V
1	9	7.5	5.2	3.8	2.2
2	18	15	10.4	7.6	4.4
3	27	22.5	15.6	11.4	6.6

## 2.UCE 约束命令介绍

4	36	30	20.8	15.2	8.8
5	45	37.5	26	19	11.2
6	54	45	31.2	22.8	13.2

表 2-3 PGT180H 的 IOB 驱动器 P 管/N 管 finger 数量的电流对应表

说明：

- 1) 测试条件为 ttcorner。
- 2) 表 中 的 Finger 列 表 示 PAP\_IO\_PMOS\_FINGER 或 PAP\_IO\_NMOS\_FINGER 设置的数值。

### 2.3.6 IP FDC 支持指定子 module

仅支持手写 fdc，格式：define\_current\_design current\_design\_name

当约束文件中设置该命令，后续命令将针对 current\_design 做约束。重复该过程，可以切换 module。

说明：

- 1、首次使用需将该命令 define\_current\_design 放置 fdc 首行，否则报错。
- 2、不支持对子 module 的端口做约束

### 2.3.7 Fdc 约束文件逻辑约束支持通配符

格式：define\_attribute {p:men\_dq\*} {PAP\_IO\_DIRECTION} {INOUT}

Port 中\*匹配的对象支持该属性约束。

### 2.3.8 UCE 属性使用实例

属性的传递可以通过两种方式：

- a.通过逻辑约束文件传递
- b.在 RTL 代码中直接编写

#### 2.3.8.1 通过逻辑约束文件传递

在逻辑约束文件中编写命令可以采取两种方式：

通过 UCE 中的 Attribute 和 IO Table 中编辑生成属性命令。

目前 IO table 默认保存格式是多行单属性格式，格式如下：

```
define_attribute {p:port_name} {attribute1}{value1}
```

```
define_attribute {p:port_name} {attribute2}{value2}
```

## 2.UCE 约束命令介绍

.....

综合工具仅 ADS 同时支持单行多属性格式，可通过手动编辑，格式如下：

```
define_attribute -attributes {attribute1 value1 attribute2 value2 · · · · }
```

```
{p:port_name}。
```

保存规则需注意以下两点：

1、对于 IO table 读取 fdc 中单行多属性格式，若有新增、修改属性，按单行多属性格式保存，保持原用户格式不变。

2、同一 port 约束命令同时存在单行多属性格式及多行单属性格式，若有新增、修改属性，按默认格式多行单属性。

Fdc 中存在的单行多属性需要切换综合工具 OEM 使用时，可通过使用转换工具转行成多行单属性。（转换工具入口->选中约束文件右键-Convert Constraint Style）

b.直接手动编写约束文件

这两种方式生成的命令格式应该符合上面的属性介绍。

### 2.3.8.2 在 RTL 代码中直接编写

在 RTL 代码中写属性命令：

在.v 和 vm 中描述属性方法是一样的，如下为一段.v 文件部分示例：

```
module case_test(
    sclk,
    xrst,
    wptr,
    if_data,
    of_data
);
(* PAP_IO_STANDARD="LVCMOS33", PAP_IO_LOC="A1" *)input sclk;
input xrst;
input [7:0] wptr;
input [7:0] if_data;
```

## 2.UCE 约束命令介绍

```
output[7:0] of_data;  
wire [7:0] of_data;  
(* PAP_REGION="reg(1,2,3,4)" *)reg cnt2;
```

在上面的.v 文件中给 input sclk 设置了属性 PAP\_IO\_STANDARD 和 PAP\_IO\_LOC 属性，给 reg cnt2 设置了 PAP\_REGION 属性，这里等同于在 fdc 约束文件中写属性命令：

```
define_attribute {p:sclk} {PAP_IO_LOC} {A1}  
define_attribute {p:sclk} {PAP_IO_STANDARD} {LVCMOS33}  
define_attribute {cnt2} {PAP_REGION} {reg(1,2,3,4)}
```

在综合工具处理之后会将这些属性传递到约束文件 lcf 中，对应与其等同的 design 文件中描述部分如下：

```
(* PAP_IO_STANDARD="LVCMOS33", PAP_IO_LOC="A1" *)input sclk ;  
(* PAP_REGION="reg(1,2,3,4)" *) GTP_DFF_AC cnt2 (  
    .Q(cnt2_Z),  
    .D(cnt2_i_0),  
    .CLK(sclk_c),  
    .ACLR(xrst_c_i_0)  
);
```

属性最终会跟随 DB 文件传递下来，对于属性命名以 PAP\_IO\_(PAP\_IO\_DIRECTION 和 PAP\_IO\_REGISTER 除外) 开始的属性，均会转化存放在物理约束文件中。

## 2.4 查询命令

### 2.4.1 get\_cells

get\_cells 用于返回所有满足条件的 cell 的集合。命令格式如下：

```
get_cells [-hierarchical] [-hsc separator] [-regexp] [-nocase] <patterns>
```

-hierarchical：匹配完整层次名，例如 A\*C 可以匹配到完整层次名为 A/B/C 的对象。

## 2.UCE 约束命令介绍

-hsc separator : hierarchical 分隔符。如果该 cell 是 hierarchical，那么在书写名字的时候应该加上 module 的名字，并且用分隔符隔开，例如，假设分隔符为 ‘\_’ module A 中有 cell 名字为 unst，A 被例化为 a，那么 cell 的名字应该写为 a\_unst。

-regexp : 表示支持正则表达式而不是简单通配符。

-nocase : 表示不关心 pattern 的大小写。

patterns : instance 的名字。

## 2.4.2 get\_nets

get\_nets 用于返回所有满足条件的 net 的集合。命令格式如下：

get\_nets [-hierarchical] [-hsc separator] [-regexp] <patterns>

-hierarchical : 匹配完整层次名，例如 A\*C 可以匹配到完整层次名为 A/B/C 的对象。

-hsc separator : hierarchical 分隔符。如果该 net 是 hierarchical，那么在书写名字的时候应该加上 module 的名字，并且用分隔符隔开，例如，假设分隔符为 ‘\_’ module A 中有 net 名字为 reset，那么 net 的名字应该写为 A\_reset。

-regexp : 表示支持正则表达式而不是简单通配符。

-nocase : 表示不关心 pattern 的大小写。

patterns : net 的名字。

## 2.4.3 get\_pins

get\_pins 用于返回所有满足条件的 pin 的集合。命令格式如下：

get\_pins [-hierarchical] [-hsc separator] [-regexp] [-nocase] <patterns>

-hierarchical : 匹配完整层次名，例如 A\*C 可以匹配到完整层次名为 A/B/C 的对象。

-hsc separator : hierarchical 分隔符。如果该 pin 是 hierarchical，那么在书写名字的时候应该加上 module 的名字，并且用分隔符隔开，例如，假设分隔符为 ‘\_’ module A 中有 pin 名字为 reset，那么 pin 的名字应该写为 A\_reset

-regexp : 表示支持正则表达式而不是简单通配符。

-nocase : 表示不关心 pattern 的大小写。

## 2.UCE 约束命令介绍

---

patterns : pin 的名字。

#### 2.4.4 get\_clocks

get\_clocks 用于返回所有满足条件的 clock 的集合。命令格式如下：

get\_clocks [-regexp] [-nocase] [-include\_generated\_clocks] <patterns>

-regexp : 表示支持正则表达式而不是简单通配符。

-nocase : 表示不关心 pattern 的大小写。

-include\_generated\_clocks : 返回的时钟里面包括匹配到的 patterns 时钟所驱动的 generated clocks。

patterns : clock 的名字。

#### 2.4.5 get\_ports

get\_ports 用于返回所有满足条件的 clock 的集合。命令格式如下：

get\_ports [-regexp] [-nocase] <patterns>

-regexp : 表示支持正则表达式而不是简单通配符。

-nocase : 表示不关心 pattern 的大小写。

patterns : port 的名字。

### 2.5 常见约束错误

约束文件中，差分端口放置相反时，ADS 流程 synthesize 阶段约束检查报错

报错信息：E: ConstraintEditor-0063: Differential port is not properly placed.

该规则适用于 GTP\_INBUFDS、GTP\_OUTBUFDS、GTP\_IOBUFDS

修改建议，检查差分端口放置是否匹配

## 2.6 通用 IO 属性默认值

以下，根据不同器件，对通用的 IO 属性的默认值进行说明。当用户通过约束文件约束了 IO，但是没有对该属性进行设置，则后续会根据默认值来进行赋值。NA 表示没有默认值。

### 2.6.1 Compact(PGC2KG)

Direction	I/O Standard	默认值			
		DRIVE	SLEW	BUS_KEEPER	
Input	LVTTL33	NA	NA	PULLUP	
	PCI33				
	LVCMOS12				
	LVCMOS15				
	LVCMOS18				
	LVCMOS25				
	LVCMOS33				
	LVDS				
	LVPECL33				
	LVTTL33D				
	LVCMOS33D				
	LVCMOS25D				
	MLVDS25			NONE	
	BLVDS25				
	LVCMOS18D				
Output	GTL1P0_2V5	4	SLOW		
	VREF				
	GTL1P0_3V3				
	LVTTL33				
	PCI33				
	LVCMOS33				
	LVCMOS25				

## 2.UCE 约束命令介绍

	LVDS	3.5	NA	SLOW	
	LVPECL33				
	BLVDS25	8			
	MLVDS25				
	LVTTL33D				
	LVCMOS33D	4			
	LVCMOS25D				
	LVDS25E	8			
	LVCMOS18D	4			
Inout	LVTTL33			SLOW	
	PCI33				
	LVCMOS33	4			
	LVCMOS25				
	LVCMOS18				
	LVCMOS15				
	LVCMOS12	2			
	MIPI				
	LVDS	3.5	NA		
	LVPECL33	8		SLOW	NONE
	MLVDS25		12		
	BLVDS25				
	LVCMOS33D				
	LVCMOS25D	4			
	LVTTL33D				
	LVDS25E	8			
	LVCMOS18D				
	GTL1P0_2V5	4			
	GTL1P0_3V3				

表 2-4 Compact(PGC2KG)

## 2.6.2 Logos(PGL50H)

Direction	I/Ostandard	默认值		
		DRIVE	SLEW	BUS_KEEPER
Input	LVTTL33	NA	NA	NONE
	PCI33			
	LVCMOS12			
	LVCMOS15			
	LVCMOS18			
	LVCMOS25			
	LVCMOS33			

## 2.UCE 约束命令介绍

	SSTL25_I				
	SSTL25_II				
	SSTL18_I				
	SSTL18_II				
	SSTL15_I				
	SSTL15_II				
	SSTL15D_I				
	SSTL15D_II				
	HSTL18_I				
	HSTL18_II				
	HSTL18D_I				
	HSTL18D_II				
	HSTL15_I				
	LVDS25				
	LVDS33				
	MINI-LVDS				
	SUB-LVDS				
	SSTL25D_I				
	SSTL25D_II				
	SSTL18D_I				
	SSTL18D_II				
	HSTL15D_I				
	TMDS				
	BLVDS				
	PPDS				
	RSDS				
	LVPECL				
Output	LVCMOS33	8	SLOW		
	LVCMOS25				
	LVCMOS18				
	LVCMOS15				
	LVCMOS12	2			
	PCI33	1.5			
	LVTTL33	8			
	SSTL25_I	8.1			
	SSTL25_II	16.2			
	SSTL18_I	6.7			
	SSTL18_II	13.4			
	SSTL15_I	7.5			

2.UCE 约束命令介绍

	SSTL15_II	8.8	NA	SLOW		
	HSTL18_I	8				
	HSTL18_II	16				
	HSTL15_I	8				
	LVDS25	3.5				
	LVDS33					
	MINI-LVDS	3				
	SUB-LVDS	1.5				
	TMDS	9				
	BLVDS	16				
	PPDS					
	RSDS					
	LVPECL					
	SSTL25D_I	8.1				
	SSTL25D_II	16.2				
	SSTL18D_I	6.7				
	SSTL18D_II	13.4				
	SSTL15D_I	7.5				
	SSTL15D_II	8.8				
	HSTL18D_I	8				
	HSTL18D_II	16				
	HSTL15D_I	8				
Inout	LVCMOS33	8	SLOW			
	LVCMOS25					
	LVCMOS18					
	LVCMOS15					
	LVCMOS12					
	LVTTL33	8				
	PCI33	1.5				
	SSTL25_I	8.1				
	SSTL25_II	16.2				
	SSTL18_I	6.7				
	SSTL18_II	13.4				
	SSTL15_I	7.5				
	SSTL15_II	8.8				
	HSTL18_I	8				
	HSTL18_II	16				
	HSTL15_I	8				
	LVDS25	3.5	NA			

## 2.UCE 约束命令介绍

LVDS33				
MINI-LVDS	3			
SUB-LVDS	1.5			
SLVS	2			
TMDS	9			
BLVDS				
PPDS				
LVPECL		16		
RSDS				
SSTL25D_I	8.1			
SSTL25D_II	16.2			
SSTL18D_I	6.7			
SSTL18D_II	13.4			
SSTL15D_I	7.5			
SSTL15D_II	8.8			
HSTL18D_I	8			
HSTL18D_II	16			
HSTL15D_I	8			

表 2-5 Logos(PGL50H)

### 2.6.3 Logos2(PG2L100H)

Direction	I/Ostandard	默认值		
		DRIVE	SLEW	BUS_KEEPER
Input	LVTTL33			
	PCI33			
	LVCMOS12			
	LVCMOS15			NONE
	LVCMOS18			
	LVCMOS25			
	LVCMOS33			
	SSTL18D_I	NA	NA	
	SSTL18D_II			
	SSTL18_I			
	SSTL18_II			
	SSTL15_I			UNUSED
	SSTL15_II			
	HSTL15_II			
	HSTL15D_I			

## 2.UCE 约束命令介绍

	HSTL18_I			
	HSTL18_II			
	HSTL18D_I			
	HSTL18D_II			
	HSTL15_I			
	HSTL15D_II			
	SSTL135_I			
	SSTL135_II			
	SSTL135D_I			
	SSTL135D_II			
	HSUL12			
	HSUL12D			
	SSTL15D_I			
	SSTL15D_II			
	LPDDR			
	LPDDR			
	LVDS25			
	MINI-LVDS			
	BLVDS			
	PPDS			
	RSDS			
	TMDS			
	LVPECL			
	LVCMOS33			
	LVCMOS25			
	LVCMOS18	4		
	LVCMOS15			
	LVCMOS12			
	LVTTL33			
	PCI33	0.1		
	LPDDR	8		
	SSTL18_I	13.4		
	SSTL18_II	8.9		
	SSTL15_I			
	SSTL15_II	13		
	HSTL18_I	8		
	HSTL18_II	16		
	HSTL15_I	8		
	HSTL15_II	16		

2.UCE 约束命令介绍

	SSTL135_I	8.9	NA	
	SSTL135_II	13		
	HSUL12	0.1		
	LVCMOS12D	4		
	BLVDS	8		
	LVDS25	3.5		
	MINI-LVDS	4.5		
	TMDS	5		
	RSDS	3.5		
	PPDS	2.5		
	LPDDRD	0.1	FAST	NONE
	SSTL18D_I	8		
	SSTL18D_II	13.4		
	SSTL15D_I	8.9		
	SSTL15D_II	13		
	HSTL18D_I	8		
	HSTL18D_II	16		
	HSTL15D_I	8		
	HSTL15D_II	16		
	SSTL135D_I	8.9		
	SSTL135D_II	13	UNUSED	
	HSUL12D	0.1		
Inout	LVCMOS33	4	FAST	NONE
	LVCMOS25			
	LVCMOS18			
	LVCMOS15			
	LVCMOS12			
	LVTTL33			
	PCI33	4	UNUSED	
	LPDDR	0.1		
	SSTL18_I	8		
	SSTL18_II	13.4		
	SSTL15_I	8.9		
	SSTL15_II	13		
	HSTL18_I	8		
	HSTL18_II	16		
	HSTL15_I	8		
	HSTL15_II	16		
	SSTL135_I	8.9		

## 2.UCE 约束命令介绍

	SSTL135_II	13		
	HSUL12	0.1		
	LVDS25	3.5	NA	
	MINI-LVDS	4.5		
	TMDS	5	FAST	NONE
	BLVDS	8		
	RSDS	3.5	NA	
	PPDS	2.5		
	MIPI	4	NA	
	LPDDRD	0.1	FAST	UNUSED
	SSTL18D_I	8		
	SSTL18D_II	13.4		
	SSTL15D_I	8.9		
	SSTL15D_II	13		
	HSTL18D_I	8		
	HSTL18D_II	16		
	HSTL15D_I	8		
	HSTL15D_II	16		
	SSTL135D_I	8.9		
	SSTL135D_II	13		
	HSUL12D	0.1		
Inout	LVCMOS12D	4		NONE

表 2-6 Logos2(PG2L100H)

## 2.6.4 Titan(PGT180H)

Direction	I/Ostandard	默认值		
		DRIVE	SLEW	BUS_KEEPER
Input	LVTTL33	NA	NA	UNUSED
	PCI33			
	LVCMOS12			
	LVCMOS15			
	LVCMOS18			
	LVCMOS25			
	LVCMOS33			
	SSTL25_I			
	SSTL25_II			
	SSTL18_I			

## 2.UCE 约束命令介绍

Output	SSTL18_II						
	SSTL15_I						
	HSTL18_I						
	HSTL18_II						
	HSTL15_I						
	HSTL15_II						
	LVDS						
	SSTL25D						
	SSTL18D						
	SSTL15D						
	HSTL18D						
	HSTL15D						
	LVCMOS33		4	SLOW	UNUSED		
	LVCMOS25						
	LVCMOS18						
	LVCMOS15						
	LVCMOS12	2					
	PCI33	1.5					
	LVTTL33	4					
	SSTL25_I	8.1					
	SSTL25_II	16.2					
	SSTL18_I	6.7					
	SSTL18_II	13.4					
	SSTL15_I	7.5					
	SSTL15_II	8.8					
	HSTL18_I	8					
	HSTL18_II	16					
	HSTL15_I	8					
	LVDS	3.5	2	NA	SLOW		
	RSDS						
	PPDS						
	LVDS25E		4				
	RSDS25E						
	PPDS25E						
	LVPECL33E						
	SSTL25D_I	8.1					
	SSTL25D_II	16.2					
	SSTL18D_I	6.7					
	SSTL18D_II	13.4					

## 2.UCE 约束命令介绍

Inout	SSTL15D_I	7.5			
	SSTL15D_II	8.8			
	HSTL18D_I	8			
	HSTL18D_II	16			
	HSTL15D_I	8			
	LVCMOS33		4		
	LVCMOS25				
	LVCMOS18				
	LVCMOS15				
	LVCMOS12	2			
	LVTTL33	4			
	PCI33	1.5			
	SSTL25_I	8.1			
	SSTL25_II	16.2			
	SSTL18_I	6.7			
	SSTL18_II	13.4			
	SSTL15_I	7.5			
	SSTL15_II	8.8			
	HSTL18_I	8			
	HSTL18_II	16			
	HSTL15_I	8			
	LVDS	3.5	NA		
	RSDS				
	PPDS	2			
	SSTL25D_I	8.1	SLOW		
	SSTL25D_II	16.2			
	SSTL18D_I	6.7			
	SSTL18D_II	13.4			
	SSTL15D_I	7.5			
	SSTL15D_II	8.8			
	HSTL18D_I	8			
	HSTL18D_II	16			
	HSTL15D_I	8			

表 2-7 Titan(PGT180H)

### 2.6.5 Titan2(PG2T390H)

#hrio

2.UCE 约束命令介绍

Direction	I/O standard	默认值		
		DRIVE	SLEW	BUS _KEEPER
Input	LVTTL33	NA	NA	NONE
	PCI33			
	LVCMOS12			
	LVCMOS15			
	LVCMOS18			
	LVCMOS25			
	LVCMOS33			
	LPDDR			
	LPDDRD			
	SSTL18_I			
	SSTL18_II			
	SSTL15_I			
	SSTL15_II			
	SSTL18D_I			
	SSTL18D_II			
	HSTL18_I			
	HSTL18_II			
	HSTL18D_I			
	HSTL18D_II			
	HSTL15_I			UNUSED
	HSTL15_II			
	HSTL15D_I			
	HSTL15D_II			
	SSTL135_I			
	SSTL135_II	NA	NA	NONE
	SSTL135D_I			
	SSTL135D_II			
	HSUL12			
	HSUL12D			
	SSTL15D_I			
	SSTL15D_II			
	LVDS25			
	MINI-LVDS			
	BLVDS			
	PPDS			
	RSDS			

2.UCE 约束命令介绍

	TMDS			
Output	LVCMOS33	4	SLOW	NONE
	LVCMOS25			
	LVCMOS18			
	LVCMOS15			
	LVCMOS12			
	LVTTL33			
	PCI33	1.5		
	LPDDR	0.1		
	SSTL18_I	8		
	SSTL18_II	13.4		
	SSTL15_I	8.9		
	SSTL15_II	13		
	HSTL18_I	8		
	HSTL18_II	16		
	HSTL15_I	8		
	HSTL15_II	16		
	SSTL135_I	8.9		
	SSTL135_II	13		
	HSUL12	0.1		
Unused	LVDS25	3.5	NA	UNUSED
	MINI-LVDS	4.5		
	TMDS	5		
	BLVDS	8	SLOW	
	RSDS	4	NA	
Inout	PPDS	3		
	LPDDRD	0.1	SLOW	NONE
	SSTL18D_I	8		
	SSTL18D_II	13.4		
	SSTL15D_I	8.9		
	SSTL15D_II	13		
	HSTL18D_I	8		
	HSTL18D_II	16		
	HSTL15D_I	8		
	HSTL15D_II	16		
	SSTL135D_I	8.9		
	SSTL135D_II	13		
	HSUL12D	0.1		
	LVCMOS33	4		

2.UCE 约束命令介绍

LVCMOS25				
LVCMOS18				
LVCMOS15				
LVCMOS12				
LVTTL33				
PCI33	1.5			
LPDDR	0.1			
SSTL18_I	8			
SSTL18_II	13.4			
SSTL15_I	8.9			
SSTL15_II	13			
HSTL18_I	8			
HSTL18_II	16			
HSTL15_I	8			
HSTL15_II	16			
SSTL135_I	8.9			
SSTL135_II	13			
HSUL12	0.1			
LVDS25	3.5	NA	NONE	
MINI-LVDS	4.5			
TMDS	5	SLOW		
BLVDS	8			
RSDS	4			
PPDS	3			
MPI	4	NA	UNUSED	
LPDDRD	0.1	SLOW		
SSTL18D_I	8			
SSTL18D_II	13.4			
SSTL15D_I	8.9			
SSTL15D_II	13			
HSTL18D_I	8			
HSTL18D_II	16			
HSTL15D_I	8			
HSTL15D_II	16			
SSTL135D_I	8.9			
SSTL135D_II	13			
HSUL12D	0.1			

表 2-8 Titan2(PG2T390H) hrio

#hpio

Direction	I/Ostandard	默认值		
		DRIVE	SLEW	BUS_KEEPER
Input	LVCMOS12	NA	NA	NONE
	LVCMOS15			
	LVCMOS18			
	SSTL15_I			
	SSTL18_I			
	SSTL18D_I			
	HSTL18_I			
	SSTL18_II			
	SSTL15_II			
	SSTL18D_II			
	HSTL18_II			
	HSTL18D_I			
	HSTL18D_II			UNUSED
	HSTL15_I			
	HSTL15_II			
	HSTL15D_I			
	HSTL15D_II			
	SSTL135_I			
	SSTL135_II			
	SSTL135D_I			
	SSTL135D_II			
	HSUL12			
	HSUL12D			
	SSTL15D_I			
	SSTL15D_II			

2.UCE 约束命令介绍

SSTL12			
HSTL12_I			
SSTL12D			
HSTL12D_I			
POD12			
POD12D			
SSTL18_I_CAL			
SSTL18_II_CAL			
SSTL15_I_CAL			
SSTL15_II_CAL			
SSTL18D_I_CAL			
SSTL18D_II_CAL			
HSTL18_I_CAL			
HSTL18_II_CAL			
HSTL18D_I_CAL			
HSTL18D_II_CAL			
HSTL15_I_CAL			
HSTL15_II_CAL			
HSTL15D_I_CAL			
HSTL15D_II_CAL			
SSTL135_I_CAL			
SSTL135_II_CAL			
SSTL135D_I_CAL			
SSTL135D_II_CA			
HSUL12_CAL			
HSUL12D_CAL			
SSTL15D_I_CAL			
SSTL15D_II_CAL			
SSTL12_CAL			
HSTL12_I_CAL			
SSTL12D_CAL			
HSTL12D_I_CAL			
POD12_CAL			
POD12D_CAL			
LVCAL_15			
LVCAL_18			
HSLVCAL_15			
HSLVCAL_18			
LVDS_18			NONE

## 2.UCE 约束命令介绍

Output	LVC MOS18	12	SLOW	UNUSED
	LVC MOS15	2		
	LVC MOS12			
	SSTL18_I	8		
	SSTL18_II	13.4		
	SSTL15_I	8.9		
	SSTL15_II	13		
	HSTL18_I	8		
	HSTL18_II	16		
	HSTL15_I	8		
	HSTL15_II	16		
	SSTL135_I	8.9		
	SSTL135_II	13		
	HSUL12	0.1		
	HSTL12_I	6.3		
	SSTL12	14.25		
	POD12	NA		
	LVDS_18	3.5	NA	
Inout	SSTL18D_I	8	SLOW	UNUSED
	SSTL18D_II	13.4		
	SSTL15D_I	8.9		
	SSTL15D_II	13		
	HSTL18D_I	8		
	HSTL18D_II	16		
	HSTL15D_I	8		
	HSTL15D_II	16		
	SSTL135D_I	8.9		
	SSTL135D_II	13		
	HSUL12D	0.1		
	HSTL12D_I	6.3		
	SSTL12D	14.25		
	POD12D	NA		
Inout	LVC MOS18	12	NONE	UNUSED
	LVC MOS15	2		
	LVC MOS12			
	SSTL18_I	8		
	SSTL18_II	13.4		
	SSTL15_I	8.9		
	SSTL15_II	13		

## 2.UCE 约束命令介绍

	HSTL18_I	8	NA	NONE
	HSTL18_II	16		
	HSTL15_I	8		
	HSTL15_II	16		
	SSTL135_I	8.9		
	SSTL135_II	13		
	HSTL12_I	6.3		
	SSTL12	14.25		
	POD12	NA		
	HSUL12	0.1		
	MPI	2		
	LVDS_18	3.5		
	SSTL18D_I	8		
	SSTL18D_II	13.4		
	SSTL15D_I	8.9		
	SSTL15D_II	13		
	HSTL18D_I	8		
	HSTL18D_II	16		
	HSTL15D_I	8		
	HSTL15D_II	16		
	SSTL135D_I	8.9		
	SSTL135D_II	13		
	HSUL12D	0.1		
	HSTL12D_I	6.3		
	SSTL12D	14.25		
	POD12D			
Output	LVCAL_18		NA	NONE
	LVCAL_15			
	SSTL18_I_CAL			
	SSTL18_II_CAL			
	SSTL15_I_CAL			
	SSTL15_II_CAL			
	HSTL18_I_CAL			
	HSTL18_II_CAL			
	HSTL15_I_CAL			
	HSTL15_II_CAL			
	SSTL135_I_CAL			
	SSTL135_II_CAL			
	HSUL12_CAL			

## 2.UCE 约束命令介绍

	HSTL12_I_CAL	NONE	UNUSED
	SSTL12_CAL		
	POD12_CAL		
	HSLVCAL_18		
	HSLVCAL_15		
	SSTL18D_I_CAL		
	SSTL18D_II_CAL		
	SSTL15D_I_CAL		
	SSTL15D_II_CAL		
	HSTL18D_I_CAL		
	HSTL18D_II_CAL		
	HSTL15D_I_CAL		
	HSTL15D_II_CAL		
	SSTL135D_I_CAL		
	SSTL135D_II_CA		
	HSUL12D_CAL		
	HSTL12D_I_CAL		
	SSTL12D_CAL		
	POD12D_CAL		
Inout	LVCAL_18	NONE	UNUSED
	LVCAL_15		
	HSLVCAL_18		
	HSLVCAL_15		
	SSTL18_I_CAL		
	SSTL18_II_CAL		
	SSTL15_I_CAL		
	SSTL15_II_CAL		
	HSTL18_I_CAL		
	HSTL18_II_CAL		
	HSTL15_I_CAL		
	HSTL15_II_CAL		
	SSTL135_I_CAL		
	SSTL135_II_CAL		
	HSTL12_I_CAL		
	SSTL12_CAL		
	POD12_CAL		
	HSUL12_CAL		
	SSTL18D_I_CAL		
	SSTL18D_II_CAL		

2.UCE 约束命令介绍

SSTL15D_I_CAL			
SSTL15D_II_CAL			
HSTL18D_I_CAL			
HSTL18D_II_CAL			
HSTL15D_I_CAL			
HSTL15D_II_CAL			
SSTL135D_I_CAL			
SSTL135D_II_CA			
HSTL12D_I_CAL			
SSTL12D_CAL			
POD12D_CAL			
HSUL12D_CAL			

表 2-9 Titan2(PG2T390H) hpio