

## 输入/输出延迟单元 (IODELAY) 简介

每个 I/O 模块包含一个可编程绝对延迟单元，称为 IODELAY。IODELAY 可以连接到 ILOGIC/ISERDES 或 OLOGIC/OSERDES 模块，也可同时连接到这两个模块。IODELAY 是具有 64 个 tap 的环绕延迟单元，具有标定的 tap 分辨率（见附图 1）。IODELAY 可用于组合输入通路、寄存输入通路、组合输出通路或寄存输出通路，还可以在内部资源中直接使用。IODELAY 允许各输入信号有独立的延迟。通过在《Virtex-5 数据手册》中规定的范围内选择 IDELAYCTRL 参考时钟，可以改变 tap 延迟分辨率。IODELAY 资源可用作 IDELAY、ODELAY 或组合延迟。

1、当用作 IDELAY 时，数据从 IBUF 或内部资源输入，然后输出到 ILOGIC/ISERDES。有三种可用操作模式：

- **零保持时间延迟模式** (IDELAY\_TYPE = DEFAULT)

这种操作模式允许向后兼容，以使用 Virtex-II、Virtex-II Pro 和 Virtex-4 器件中的零保持时间延迟功能的设计。当在没有 DCM 的情况下使用全局时钟采集数据（引脚到引脚参数）时，用这种延迟单元提供“非正保持时间”。在这种模式下使用时，不需要例化 IDELAYCTRL 基元。有关更多详情，请见“IDELAYCTRL 用法及设计指导原则”。

- **固定延迟模式** (IDELAY\_TYPE = FIXED)

在固定延迟模式下，配置时将延迟值预设置成由属性 IDELAY\_VALUE 确定的 tap 数。此值配置后不可更改。在这种模式下使用时，必须例化 IDELAYCTRL 基元。有关更多详情，请见“IDELAYCTRL 用法及设计指导原则”。

- **可变延迟模式** (IDELAY\_TYPE = VARIABLE)

在可变延迟模式下，可以在配置后通过操控控制信号 CE 和 INC 来改变延迟值。在这种模式下使用时，必须例化 IDELAYCTRL 基元。有关更多详情，请见“IDELAYCTRL 用法及设计指导原则”。

2、当用作 ODELAY 时，数据从 OLOGIC/OSERDES 输入，然后输出到 OBUF。有一种可用操作模式：

- **固定延迟输出模式**

在固定延迟输出模式下，配置时将延迟值预设置成由属性 ODELAY\_VALUE 确定的 tap 数。此值配置后不可更改。在这种模式下使用时，必须例化 IDELAYCTRL 基元。有关更多详情，请见“IDELAYCTRL 用法及设计指导原则”。

3、当用作双向延迟时，将 IOB 配置成双向模式。IODELAY 交替延迟输入通路和输出通路上的数据。有两种可用操作模式：

- **固定 IDELAY (IDELAY\_TYPE = FIXED) 和固定 ODELAY 模式**

在这种模式下，IDELAY 和 ODELAY 的值都是在配置时预设置，分别由 IDELAY\_VALUE 和 ODELAY\_VALUE 属性确定。此值配置后不可更改。在这种模式下使用时，必须例化 IDELAYCTRL 基元。有关更多详情，请见“IDELAYCTRL 用法及设计指导原则”。

计指导原则”。

• 可变 IDELAY (IDELAY\_TYPE = VARIABLE) 和固定 ODELAY 模式

在这种模式下，只有 IDELAY 值可以在配置后通过操控控制信号 CE 和 INC 来动态更改。IDELAY 基元中 T 引脚的逻辑级别动态确定模块是 IDELAY 模式还是 ODELAY 模式。在这种模式下使用时，必须例化 IDELAYCTRL 基元。有关更多详情，请见“IDELAYCTRL 用法及设计指导原则”。

表 1-1 列出了 IDELAY 支持的配置。

表 1-1: IDELAY 支持的配置

IDELAY 模式	IDELAY 的方向	IDELAY 单元中使用的输入引脚	信号源	目标	支持的延迟模式
IDELAY	I	IDATAIN	IBUF	ILOGIC/SERDES/ 内部资源	默认 / 固定 / 可变
		DATAIN	内部资源		固定 / 可变
ODELAY	O	ODATAIN	OLOGIC/OSERDES	OBUF	固定
双向延迟	I (T = 1 时)	IDATAIN	IBUF	ILOGIC/SERDES/ 内部资源	固定 / 可变
	O (T = 0 时)	ODATAIN	OLOGIC/OSERDES		固定

IDELAY 基元

图 1-1 所示为 IDELAY 基元。

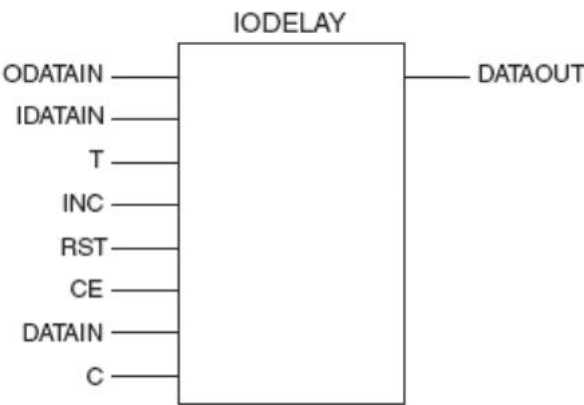


图 1-1: IDELAY 基元

表 1-2 列出了 IDELAY 基元中的可用端口。所有端口均为 1 位宽。

表 1-2: IDELAY 基元端口

端口名称	方向	功能
DATAOUT	输出	来自三个数据输入端口（IDATAIN、ODATAIN、DATAIN）之一的延迟数据
IDATAIN	输入	来自 IOB 的 IODELAY 的数据输入
ODATAIN	输入	来自 OSERDES/OLOGIC 的 IODELAY 的数据输入
DATAIN	输入	来自 FPGA 内部资源的 IODELAY 的数据输入
T	输入	三态输入控制端口。此端口动态确定 IODELAY 是用作 IDELAY 还是用作 ODELAY
CE	输入	启用递增 / 递减功能
INC	输入	Tap 延迟的递增 / 递减数
RST	输入	将 IODELAY 单元复位到预设值
C	输入	可变模式下使用的时钟输入

## IODELAY 端口

### 来自 IOB 的数据输入-IDATAIN

IDATAIN 输入由相应的 IOB 驱动。在 IDELAY 模式下，可以将数据驱动到 ILOGIC/ISERDES 模块或直接驱动到 FPGA 内部资源中，也可以通过 DATAOUT 端口按照由 IDELAY\_VALUE 设置的延迟将数据同时驱动到这两者。

### 来自 FPGA 内部资源的数据输入-ODATAIN

ODATAIN 输入由 OLOGIC/OSERDES 驱动。在 ODELAY 模式下，ODATAIN 按照由 ODELAY\_VALUE 设置的延迟驱动连接到 IOB 的 DATAOUT 端口。

### 来自 FPGA 内部资源的 IODELAY 的数据输入-DATAIN

DATAIN 输入直接由 FPGA 内部资源驱动，以提供一条可访问内部资源逻辑的延迟线。数据按照由 IDELAY\_VALUE 设置的延迟通过 DATAOUT 端口返回到内部资源中。DATAIN 可以在芯片内反转。数据不能驱动到 IOB。

## 数据输出-DATAOUT

来自三个数据输入端口的延迟数据。DATAOUT 连接到内部资源（IDELAY 模式）或 IOB（ODELAY 模式），或同时连接到这两者（双向延迟模式）。如果在双向延迟模式下使用，则 T 端口在 IDATAIN 和 ODATAIN 通路之间动态切换，根据来自 OLOGIC 模块的三态信号 T 所指示的方向交替提供输入/输出延迟。

## 三态输入-T

这是三态输入控制端口。对于双向操作，T 引脚信号还控制 OBUFT 的 T 引脚。

## 时钟输入-C

对 IODELAY 基元的所有控制输入（RST、CE 和 INC）均与时钟输入（C）同步。当 IODELAY 配置成可变模式时，必须将一个时钟连接到此端口。C 可以在芯片内反转。

## 模块复位-RST

IODELAY 复位信号 RST 将延迟单元复位到由 IDELAY\_VALUE 或 ODELAY\_VALUE 属性设置的值。如果未指定这些属性，则假定该值为零。RST 信号是高有效复位，与输入时钟信号（C）同步。

表 1-3 概述了控制引脚的功能。

表 1-3：控制引脚描述

引脚	类型	值	描述
INC	输入	1	Tap 延迟的递增 / 递减数
CE	输入	1	启用递增 / 递减功能
RST	输入	1	将延迟单元复位到预设定的值。如果没有预设定值则复位到 0

递增/递减信号– CE、INC

递增/递减由使能信号（CE）控制。当 IDELAY\_TYPE = VARIABLE 时，此接口只可用于 IDELAY 模式。

只要 CE 保持为 High，IDELAY 就会每时钟（C）周期按 IDELAYRESOLUTION 递增或递减。INC 的状态确定 IDELAY 是递增还是递减：INC = 1 为递增，INC = 0 为递减，与时钟（C）同步。如果 CE 为 Low，则通过 IDELAY 的延迟不变，与 INC 的状态无关。

当 CE 转为 High 时，在下一个时钟上升沿上开始递增/递减操作。当 CE 转为 Low 时，在下一个时钟上升沿上停止递增/递减操作。

IODELAY 是环绕可编程延迟单元。当达到延迟单元的末尾（tap 63）时，随后的递增功能就会返回到 tap0。同样的规则也适用于递减功能，即低于零的递减会转到 tap 63。表 1-4 概述了递增/递减操作。

表 1-4：递增/递减操作

操作	RST	CE	INC
复位到 IDELAY_VALUE	1	x	x
递增 tap 数	0	1	1
递减 tap 数	0	1	0
无变化	0	0	x

注：

- 1. RST 优先于 CE 和 INC。

IODELAY 属性

表 1-5 列出了 IDELAY 的属性。

表 1-5：IODELAY 属性概述



属性	值	默认值	描述
IDELAY_TYPE	字符串： DEFAULT、 FIXED 或 VARIABLE	DEFAULT	设置 tap 延迟线的类型。默认延迟用于保证零保持时间，固定延迟用于设置静态延迟值，可变延迟用于动态调整延迟值。
IDELAY_VALUE	整数：0 到 63	0	指定固定模式下的固定延迟 tap 数或可变模式下的初始 tap 数（输入通路）。
ODELAY_VALUE	整数：0 到 63	0	指定固定延迟 tap 数（输出通路）。
HIGH_PERFORMANCE_MODE	布尔值： FALSE、TRUE	FALSE	当设置为 TRUE 时，此属性减少输出抖动。
SIGNAL_PATTERN	字符串：DATA、 CLOCK	DATA	时钟和数据信号具有不同的电气特征曲线，因此在 IODELAY 链中累计的抖动量不同。通过设置此属性，用户可以令时序分析器在计算时序时计入相应的抖动。时钟信号具有周期性，没有 1 或 0 的连续长序列；而数据具有随机性，可以有 1 和 0 的长短序列。

属性	值	默认值	描述
REFCLK_FREQUENCY	实数：190.0 到 210.0	200	IDELAYCTRL 参考时钟频率 (MHz)。
DELAY_SRC	字符串：I、O、 IO 或 DATAIN	DATAIN	I：IODELAY 链输入是 IDATAIN O：IODELAY 链输入是 ODATAIN IO：IODELAY 链输入是 IDATAIN 和 ODATAIN （由 T 控制） DATAIN：IODELAY 链输入是 DATAIN

### IDELAY\_TYPE 属性

IDELAY\_TYPE 属性设置所用延迟的类型。此属性的值是 DEFAULT、FIXED 和 VARIABLE。当设置成 DEFAULT 时，选择零保持时间延迟单元。当在没有 DCM 的情况下使用全局时钟采集数据（引脚到引脚参数）时，用这种延迟单元提供非正保持时间。

当设置成 FIXED 时，tap 延迟值固定为由 IDELAY\_VALUE 属性设置确定的 tap 数。此值是预设置的，配置后不可更改。

当设置成 VARIABLE 时，选择可变 tap 延迟单元。Tap 延迟可以通过设置 CE = 1 和 INC = 1 递增，或通过设置 CE = 1 和 INC = 0 递减。递增/递减操作与输入时钟信号 C 同步。

### IDELAY\_VALUE 属性

IDELAY\_VALUE 属性指定初始 tap 延迟数。可能的值是 0 到 63 之间的任意整数。默认值是零。当 tap 延迟复位时，tap 延迟值恢复到 IDELAY\_VALUE。在可变模式下，此属性确定延迟线的初始设置。

### ODELAY\_VALUE 属性

ODELAY\_VALUE 指定 tap 延迟数。可能的值是 0 到 63 之间的任意整数。默认值是零。当 tap 延迟复位时，tap 延迟值恢复到 ODELAY\_VALUE。

### HIGH\_PERFORMANCE\_MODE 属性

当设置为 TRUE 时，此属性减少输出抖动。输出抖动减少导致 IODELAY 单元的功率耗散略有上升。

**SIGNAL\_PATTERN 属性**

SIGNAL\_PATTERN 属性致使时序分析器计入数据或时钟通路中的相应延迟链抖动量。

**IODELAY 时序**

表 1-6 为 IODELAY 开关特性。

表 1-6: IODELAY 开关特性

符号	描述
$T_{IDELAYRESOLUTION}$	IDELAY 的 tap 分辨率
$T_{ICECK}/T_{ICKCE}$	相对于 C 的 CE 引脚建立 / 保持
$T_{IINCCK}/T_{ICKINC}$	相对于 C 的 INC 引脚建立 / 保持
$T_{IRSTCK}/T_{ICKRST}$	相对于 C 的 RST 引脚建立 / 保持

图 1-2 所示为 IDELAY 时序图。假设 IDELAY\_VALUE = 0。

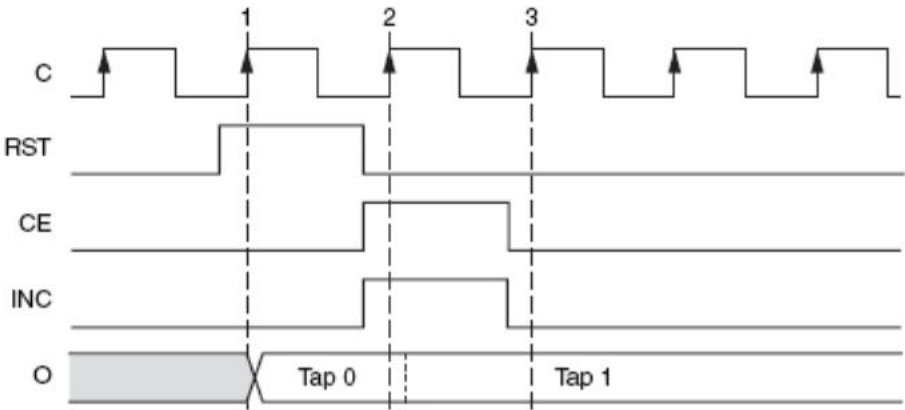


图 1-2: IDELAY 时序图

**时钟事件 1**

在 C 的上升沿上检测到复位，致使输出 O 选择 tap 0 为 64 tap 链的输出（假设 IDELAY\_VALUE = 0）。

**时钟事件 2**

在 C 的上升沿上检测到 CE 和 INC 的脉冲。这说明是递增操作。输出从 tap 0 到 tap 1 无毛刺变化。请见“递增/递减操作后的稳定性”。

**时钟事件 3**

至此，输出已经稳定在 tap 1，因而完成了递增操作。输出无限期地保持在 tap 1，直到 RST、CE 或 INC 引脚上有进一步活动。

**递增/递减操作后的稳定性**

图 1-2 显示了输出从一个 tap 向另一个变化时的不稳定时段。显然，当 tap 0 处的数据值与 tap 1 处的数据值不同时，输出必须改变状态。然而，如果 tap 0 和 tap 1 处的数据值相同（例如两者都是 0 或 1），则从 tap 0 到 tap 1 的过渡就不会在输出上造成毛刺或混乱。这一

概念可以通过设想 IODELAY 的 tap 链中的接收器数据信号来理解。如果 tap 0 和 tap1 都靠近接收器数据眼的中心，则在 tap 0 处采样的数据应与在 tap 1 处采样的数据没有区别。在这种情况下，从 tap 0 到 tap 1 的过渡不会引起输出变化。为了确保这种情况，IODELAY 的递增/递减操作设计成了无毛刺操作。

在活动用户数据通过 IODELAY 单元时，用户可以实时动态调整 IODELAY 的 tap 设置，不会扰乱活动用户数据。

当时钟信号通路中使用 IODELAY 单元时，也适用无毛刺行为。调整 tap 设置不会在输出上引起毛刺或混乱。可以调整时钟通路中 IODELAY 单元的 tap 设置，不会扰乱可能正在该时钟上运行的状态机。

### IDELAYCTRL 概述

如果用设置为 FIXED 或 VARIABLE 的 IOBDELAY\_TYPE 属性例化 IODELAY 或 ISERDES 基元，则必须在代码中例化 IDELAYCTRL 模块。IDELAYCTRL 模块在其区域内连续标定各延迟单元 (IODELAY) (见图 1-5)，以减少随工艺、电压和温度变化的影响。IDELAYCTRL 模块使用用户提供的 REFCLK 标定 IODELAY。

### IDELAYCTRL 基元

图 1-3 所示为 IDELAYCTRL 基元。

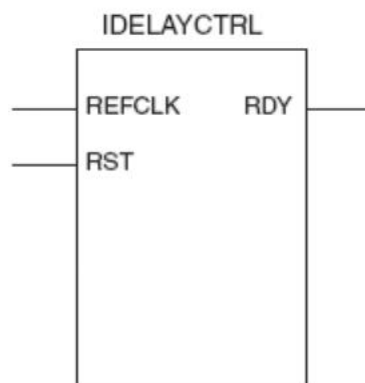


图 1-3: IDELAYCTRL 基元

### IDELAYCTRL 端口

#### RST-复位

复位输入引脚 (RST) 是 High 有效异步复位。IDELAYCTRL 在配置 (以及 REFCLK 信号稳定) 之后必须复位，以确保 IODELAY 操作正常。要求复位脉冲宽度为 TIDELAYCTRL\_RPW (详见附图 1)。IDELAYCTRL 在配置后必须复位。

#### REFCLK-参考时钟

参考时钟 (REFCLK) 提供对 IDELAYCTRL 的时间参考，以标定同区域中的所有 IODELAY 模块。此时钟必须由全局时钟缓冲器 (BUFGCTRL) 驱动。REFCLK 必须是 FIDELAYCTRL\_REF ± 指定的 ppm 容差 (IDELAYCTRL\_REF\_PRECISION)，以保证指定的 IODELAY 分辨率 (TIDELAYRESOLUTION)。REFCLK 可以由用户提供的信号源、PLL

或 DCM 直接提供，必须在全局时钟缓冲器上传输。

**RDY-就绪**

就绪（RDY）信号指示特定区域内的 IDELAY 模块标定完毕。如果 REFCLK 在一个或几个时钟周期内保持为 High 或 Low，则 RDY 信号置为无效。如果 RDY 置为 Low 无效，则 IDELAYCTRL 模块必须复位。实现工具允许不连接/忽略 RDY。图 1-4 所示为 RDY 与 RST 之间的时序关系。

**IDELAYCTRL 时序**

表 1-7 所示为 IDELAYCTRL 开关特性。

表 1-7：IDELAYCTRL 开关特性

符号	描述
$F_{IDELAYCTRL\_REF}$	REFCLK 频率
IDELAYCTRL_REF_PRECISION	REFCLK 精度
$T_{IDELAYCTRLCO\_RDY}$	IDELAYCTRL 从复位 / 启动到就绪的时间

如下图 1-4 所示，Virtex-5 RST 是边沿触发信号。

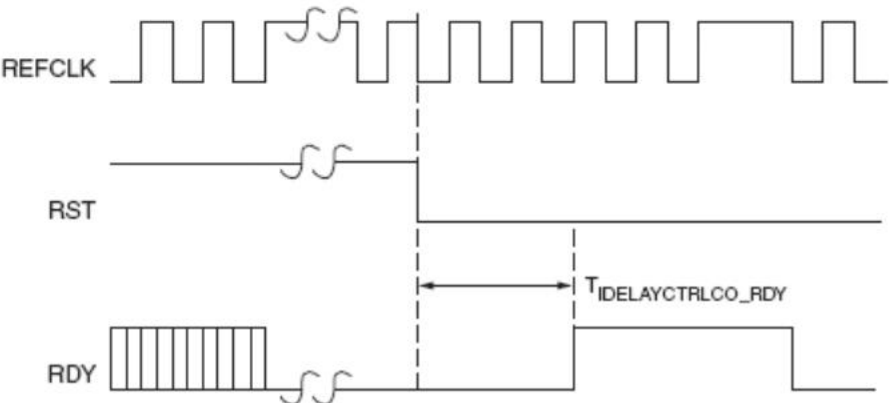


图 1-4：RDY 与 RST 之间的时序关系

**IDELAYCTRL 位置**

每个时钟区域的每个 I/O 列中都有 IDELAYCTRL 模块。一个 IDELAYCTRL 模块标定其时钟区域内的所有 IDELAY 模块。图 1-5 所示为各 IDELAYCTRL 模块的相对位置。



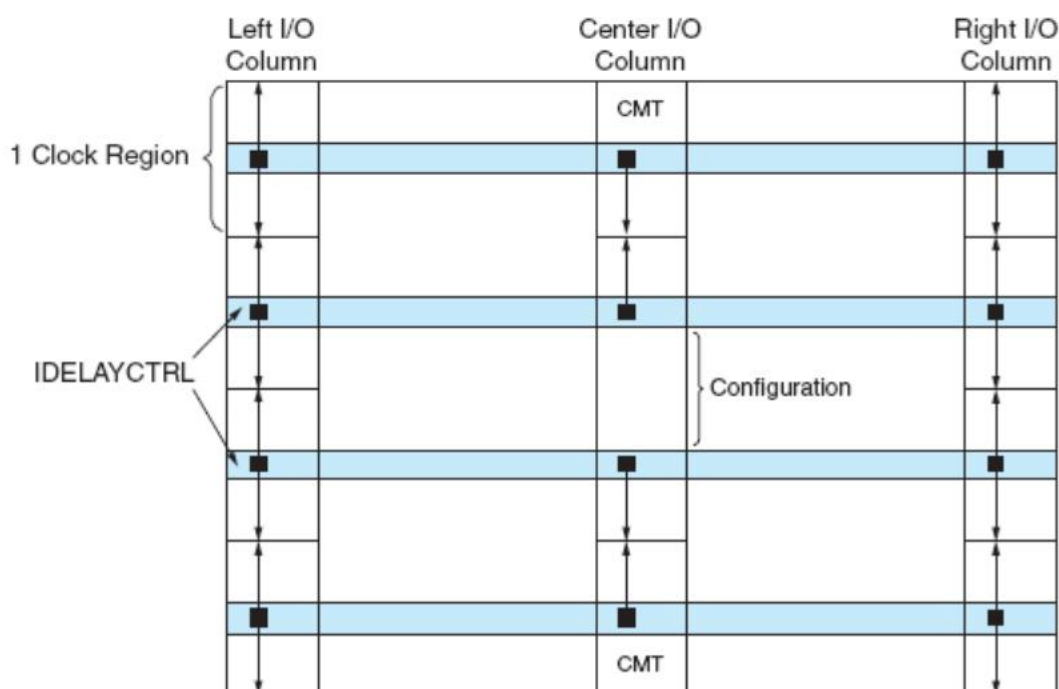


图 1-5: 各 IDELAYCTRL 模块相对位置

## IDELAYCTRL 用法及设计指导原则

本部分讲述 Virtex-5 IDELAYCTRL 模块的使用、设计指导原则和推荐用法。

### 例化无 LOC 约束的 IDELAYCTRL

当例化无 LOC 约束的 IDELAYCTRL 时，用户在 HDL 设计代码中必须只例化一个 IDELAYCTRL 实例。实现工具将 IDELAYCTRL 实例自动复制到整个器件，甚至复制到未使用延迟单元的时钟区域中。这样做资源占用率较高，在每个时钟区域内都要使用一个全局时钟资源，并且使用布线资源较多，因此功耗较大。（注：在简单工程中对 IDELAYCTRL 基元是否制定 LOC 约束，通过用 XPOWER 工具对功耗进行估算对比发现，是否制定 LOC 对功耗影响不大或者几乎无影响，但在大规模工程中，会由于 LOC 约束的指定带来功耗的额外增加）已例化 IDELAYCTRL 实例的 RST 和 REFCLK 输入端口和所复制 IDELAYCTRL 实例的对应输入端口相连接。

有两个特例：

- 1、当忽略 RDY 端口时，所有被复制的 IDELAYCTRL 实例的 RDY 信号均不连接。

对于例化无 LOC 约束的 IDELAYCTRL 基元并且不连接 RDY 输出端口的情况，《库指南》中提供了 VHDL 和 Verilog 使用模型。

图 1-6 所示为例化 IDELAYCTRL 组件后形成的电路。

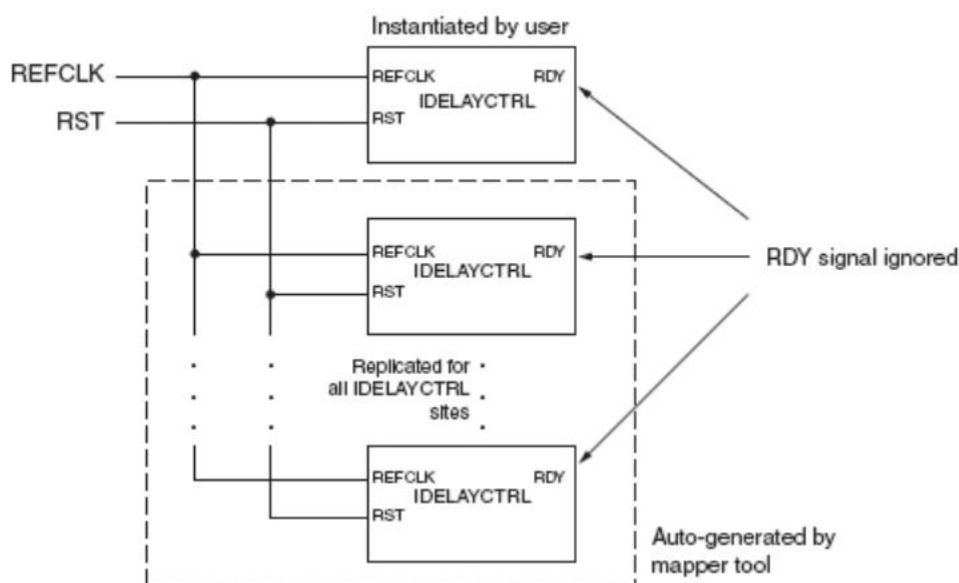


图 1-6: 例化无 LOC 约束的 IDELAYCTRL-不连接 RDY

2、当连接 RDY 端口时，例化一个宽度等于时钟区域数的与门，并且将例化和复制的 IDELAYCTRL 实例的 RDY 输出端口连接到与门的输入。工具将连接到已例化 IDELAYCTRL 实例的 RDY 端口的信号名称赋予与门的输出。

对于例化无 LOC 约束的 IDELAYCTRL 基元并且连接 RDY 端口的情况，《库指南》中提供了 VHDL 和 Verilog 使用模型。图 1-7 所示为例化 IDELAYCTRL 组件后形成的电路。

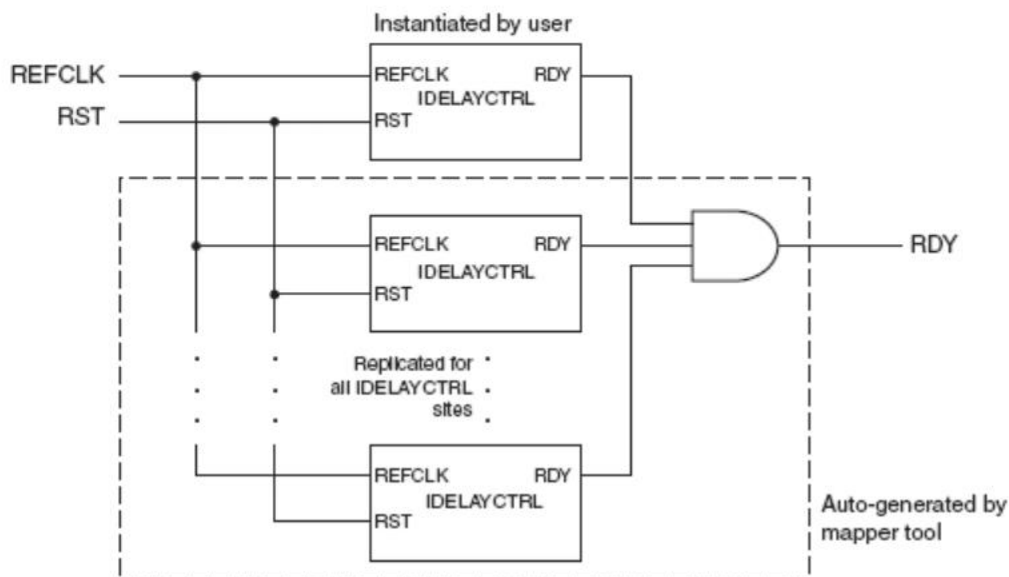


图 1-7: 例化无 LOC 约束的 IDELAYCTRL-连接 RDY

### 例化有位置（LOC）约束的 IDELAYCTRL

使用 IDELAYCTRL 模块的最有效方法是定义并锁定设计中所用每个 IDELAYCTRL 实例的位置。具体做法是例化有位置（LOC）约束的 IDELAYCTRL 实例。用户必须在使用延迟单元时定义并锁定所有 ISERDES 和 IDELAY 组件的位置。（IOBDelay\_Type 属性设置

为 FIXED 或 VARIABLE。)完成后, 可以选择 IDELAYCTRL 的位置并指定 LOC 约束。

**Xilinx 强烈建议使用有 LOC 约束的 IDELAYCTRL。**

(注: 如图 1-5 所示, 一个 IDELAYCTRL 基元可同时覆盖一个 clock region, 而一个 clock region 包含不止一个 IOB, 在约束 IDELAYCTRL 基元 LOC 时, 需要仔细定位该 IOB 属于哪个 LOC 的 IDELAYCTRL。)

### 位置约束

每个 IDELAYCTRL 模块都有 XY 位置坐标 (X: 行, Y: 列)。为了约束位置, IDELAYCTRL 实例可以附带 LOC 属性。IDELAYCTRL 位置坐标的命名规则与用来命名 CLB 位置的规则不同。这样做便于在各阵列之间传递 LOC 属性。

为 IDELAYCTRL 实例附加 LOC 属性有两种方法。

- 1、在 UCF 文件中插入 LOC 约束
- 2、在 HDL 设计文件中直接嵌入 LOC 约束

### 在 UCF 文件中插入 LOC 约束

下列语法用于在 UCF 文件中插入 LOC 约束。

```
INST "instance_name" LOC=IDELAYCTRL_X#Y#;
```

### 在 HDL 设计文件中直接嵌入 LOC 约束

下列语法用于在 HDL 设计文件中嵌入 LOC 约束。

```
// synthesis attribute loc of instance_name is "IDELAYCTRL_X#Y#";
```

在 VHDL 代码中, 用 VHDL 属性描述 LOC 约束。约束必须用下列语法声明后才能使用:

```
attribute loc : string;
```

声明之后, LOC 约束可以指定为:

```
attribute loc of instance_name:label is "IDELAYCTRL_X#Y0#";
```

《库指南》包括用于例化有 LOC 约束的 IDELAYCTRL 基元的 VHDL 和 Verilog 使用模型模板。

图 1-8 所示为例化 IDELAYCTRL 组件后形成的电路。

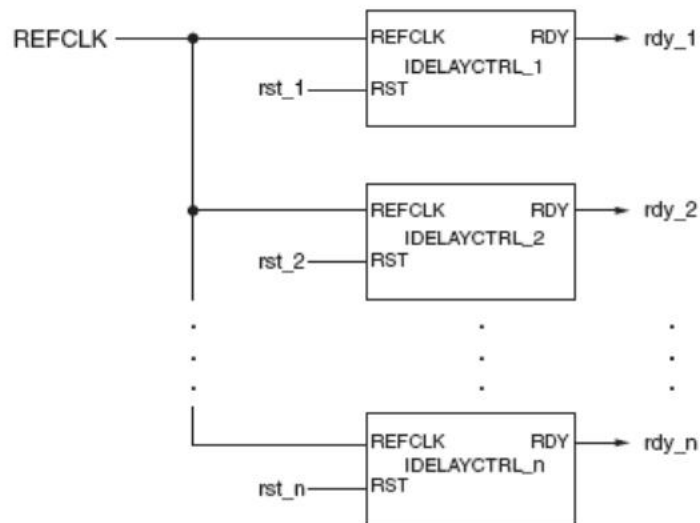


图 1-8: 例化有 LOC 约束的 IDELAYCTRL

### 例化有 LOC 约束和无 LOC 约束的 IDELAYCTRL

在有些情况下，用户例化有 LOC 约束的 IDELAYCTRL 模块，但还例化无 LOC 约束的 IDELAYCTRL 模块。如果用无位置约束的 IDELAYCTRL 模块例化 IP 核，但还需要为设计的另一部分例化无 LOC 约束的 IDELAYCTRL 模块，则实现工具会执行以下各项：

- 按“例化有位置（LOC）约束的 IDELAYCTRL”部分所述例化 LOC IDELAYCTRL 实例。
- 复制无位置约束的 IDELAYCTRL 实例，以使用一个 IDELAYCTRL 实例填充每个不具备有位置约束 IDELAYCTRL 实例的时钟区域。
- 将无位置约束 IDELAYCTRL 实例的 RST 和 REFCLK 输入端口和所复制 IDELAYCTRL 实例的对应输入端口相连接。
- 如果忽略无位置约束 IDELAYCTRL 实例的 RDY 端口，则同时忽略所复制 IDELAYCTRL 实例的所有 RDY 信号。
- 如果连接无位置约束 IDELAYCTRL 实例的 RDY 端口，则将无位置约束实例的 RDY 端口和所复制实例的 RDY 端口连接到一个自动生成的与门。实现工具将连接到无位置约束实例的 RDY 端口的信号之名称赋予与门的输出。
- 有位置约束实例的所有端口（RST、REFCLK 和 RDY）均相互独立并且独立于复制的实例。

图 1-9 所示为例化 IDELAYCTRL 件后形成的电路。

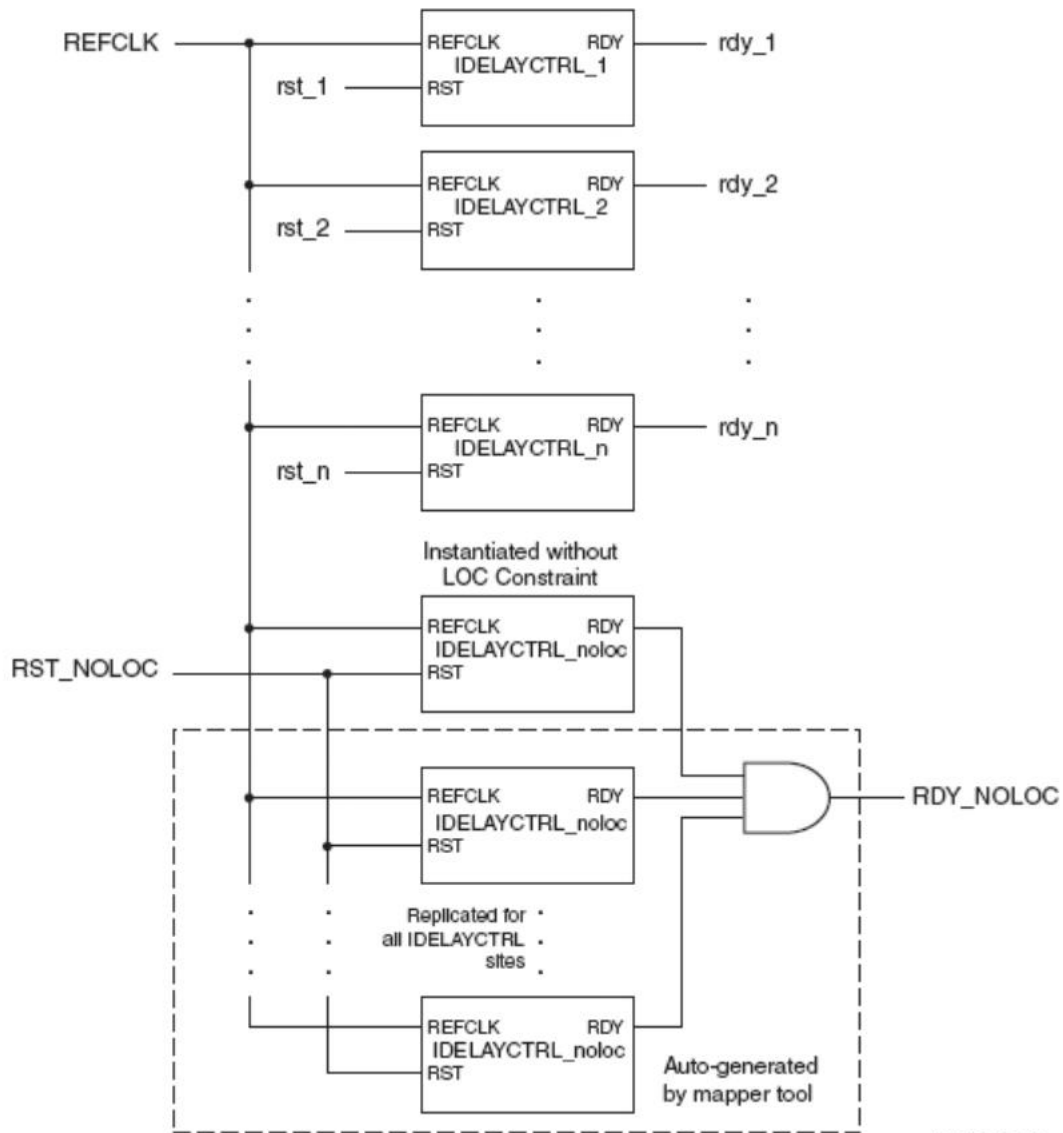


图 1-9: IDELAYCTRL 单元的混合例化例化

### 例化多个无 LOC 约束的 IDELAYCTRL

禁止例化多个无 LOC 属性的 IDELAYCTRL。如果发生这种情况，实现工具就会报错。

(注:若在一个工程中同时例化多个不同 DELAY\_SRC 属性,且无 LOC 约束的 IODELAY 时,只需例化一个 IDELAYCTRL 模块,否则 map 会报错。举例在同一个工程中需要例化一个固定 IDELAY\_VALUE 的 IODELAY\_INST0 的 DELAY\_SRC 属性设置为“1”,表示其输入为 IDATAIN,而同时需要例化另外一个固定 ODELAY\_VALUE 的 IODELAY\_INST1 的 DELAY\_SRC 属性设置为“0”,其输入为 ODATAIN,但是 IODELAY\_INST0 和 IODELAY\_INST1 使用时均无 LOC 约束,在例化 IDELAYCTRL 基元时,只需例化一个 IDELAYCTRL 基元,无需因为例化两个不同属性的 IODELAY 基元而分别例化两个 IDELAYCTRL 基元,否则 map 会报错。)



参考文献:

- 【1】： ug190-Virtex-5 用户指南
- 【2】： ds202- Virtex-5 Data Sheet: DC and Switching Characteristics
- 【3】： v51dl-Virtex-5 Libraries Guide for HDL Designs

Symbol	Description	Speed Grade			Units
		-3	-2	-1	
$T_{\text{IDELAYRESOLUTION}}$	IDELAY Chain Delay Resolution	$1/(64 \times F_{\text{REF}} \times 1e^6)$			ps
$T_{\text{IDELAYCTRLCO\_RDY}}$	Reset to Ready for IDELAYCTRL	3.00	3.00	3.00	$\mu$ s
$F_{\text{IDELAYCTRL\_REF}}$	REFCLK frequency	200.00	200.00	200.00	MHz
$\text{IDELAYCTRL\_REF\_PRECISION}$	REFCLK precision	$\pm 10$	$\pm 10$	$\pm 10$	MHz
$T_{\text{IDELAYCTRL\_RPW}}$	Minimum Reset pulse width	50.00	50.00	50.00	ns
$T_{\text{IDELAYRESOLUTION\_ERR}}$	Tap resolution error				%
$T_{\text{IODCK\_CE}}/T_{\text{IODCK\_CE}}$	CE pin Setup/Hold with respect to CK	0.29 -0.06	0.34 -0.06	0.42 -0.06	ns
$T_{\text{IODCK\_INC}}/T_{\text{IODCK\_INC}}$	INC pin Setup/Hold with respect to CK	0.18 0.02	0.20 0.04	0.24 0.06	ns
$T_{\text{IODCK\_RST}}/T_{\text{IODCK\_RST}}$	RST pin Setup/Hold with respect to CK	0.25 -0.12	0.28 -0.12	0.33 -0.12	ns

附图 1: tap 延迟精度及相关时序参数

## Verilog Instantiation Template

```
// IDELAY: Input and Output Fixed/variable Delay Element
// Virtex-5
// Xilinx HDL Libraries Guide, version 9.1i
IDELAY # (
    .DELAY_SRC          ( "I"          ), // Specify which input port to be used
                                   // "I"=IDATAIN, "O"=ODATAIN, "DATAIN"=DATAIN
    .IDELAY_TYPE        ( "DEFAULT" ), // "DEFAULT", "FIXED" or "VARIABLE"
    .IDELAY_VALUE       ( 0            ), // 0 to 63 tap values
    .ODELAY_VALUE       ( 0            ), // 0 to 63 tap values
    .REFCLK_FREQUENCY   ( 200.0        ) // Frequency used for IDELAYCTRL
                                   // 175.0 to 225.0
) IDELAY_INST (
    .DATAOUT ( DATAOUT ), // 1-bit delayed data output
    .C       ( C         ), // 1-bit clock input
    .CE      ( CE        ), // 1-bit clock enable input
    .DATAIN  ( DATAIN   ), // 1-bit internal data input
    .IDATAIN ( IDATAIN   ), // 1-bit input data input (connect to port)
    .INC     ( INC       ), // 1-bit increment/decrement input
    .ODATAIN ( ODATAIN   ), // 1-bit output data input
    .RST     ( RST       ), // 1-bit active high, synch reset input
```

```

        .T          ( T          ) // 1-bit 3-state control input
    );
// End of IODELAY_inst instantiation

```

工程实例:

```

IODELAY #(
    .DELAY_SRC          ( "0"          ), //IODELAY源是ODATAIN
    .IDELAY_TYPE        ( "FIXED" ), //静态延迟
    .ODELAY_VALUE       ( 16          ), //固定延迟数为16个tap
    .REFCLK_FREQUENCY   ( 200.0      ) //IDELAYCTRL参考时钟频率: 200MHz
) IODELAY_INST (
    .C          ( 1'b0          ), //可变模式下使用的时钟输入
    .CE          ( 1'b0          ), // 是否启动递增/递减功能
    .DATAIN      ( 1'b0          ), //
    .IDATAIN     ( 1'b0          ), //
    .INC          ( 1'b0          ), //tap延迟的递增/递减选择
    .ODATAIN     ( qLED          ), //来自OLOGIC/OSERDES的信号
    .RST          ( 1'b0          ), //将延迟单元次坏皆と瓊0 闹, 若没有预设值则复位到0
    .T            ( 1'b0          ), //用作ODELAY
    .DATAOUT     ( qLED_cld      )
);

IDELAYCTRL IDELAYCTRL_inst (
    .RDY          ( RDY          ), // 1-bit ready output
    .REFCLK        ( System_Clk ), // 1-bit reference clock input
    .RST           ( 1'b0          ) // 1-bit reset input
);

```

```

#INST "IDELAYCTRL_inst" LOC = IDELAYCTRL_X1Y2; (ucf 文件添加)

```

```

OBUF obuf_qLED (
    .I ( qLED_cld      ),
    .O ( qLED          )
);

```