Arora V 设计物理约束

用户指南

SUG1018-1.0, 2023-04-20

版权所有 © 2023 广东高云半导体科技股份有限公司

GOWIN高云、 、Gowin、晨熙、云源以及高云均为广东高云半导体科技股份有限公司注册商标，本手册中提到的其他任何商标，其所有权利属其拥有者所有。未经本公司书面许可，任何单位和个人都不得擅自摘抄、复制、翻译本文档内容的部分或全部，并不得以任何形式传播。

免责声明

本文档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。除高云半导体在其产品的销售条款和条件中声明的责任之外，高云半导体概不承担任何法律或非法律责任。高云半导体对高云半导体产品的销售和／或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。高云半导体对文档中包含的文字、图片及其它内容的准确性和完整性不承担任何法律或非法律责任，高云半导体保留修改文档中任何内容的权利，恕不另行通知。高云半导体不承诺对这些文档进行适时的更新。

版本信息

日期	版本	说明
2023/04/20	1.0	初始版本。

目录

目录 i
图目录 iii
表目录 vi
1 关于本手册 1
1.1 手册内容 1
1.2 相关文档 1
1.3 术语、缩略语 2
1.4 技术支持与反馈 2
2 物理约束语法规范 3
2.1 I/O 位置约束 3
2.2 I/O 属性约束 4
2.3 原语位置约束 5
2.4 组约束 8
2.4.1 原语组约束 8
2.4.2 相对组约束 9
2.5 资源预留约束 10
2.6 参考电压约束 11
2.7 全局时钟分配约束 12
2.8 全局时钟原语约束 13
2.9 高速时钟原语约束 14
2.10 其他约束 14
2.10.1 ADC 输入电压源约束 14
3 FloorPlanner 16
3.1 简介 16
3.2 启动 17
3.3 界面 18
3.3.1 菜单栏 19
3.3.2 Summary 和 Netlist 窗口 29
3.3.3 Package View 窗口 34

3.3.4 Chip Array 窗口 38
3.3.5 Constraint 编辑窗口 45
3.3.6 Message 窗口 50
4 FloorPlanner 使用 51
4.1 新建约束文件 51
4.2 编辑约束文件 53
4.2.1 编辑约束示例 53
4.2.2 编辑 I/O 约束 55
4.2.3 编辑原语约束 56
4.2.4 编辑组约束 57
4.2.5 编辑资源预留约束 61
4.2.6 编辑全局时钟分配约束 62
4.2.7 编辑全局时钟约束 63
4.2.8 编辑高速时钟约束 64
4.2.9 编辑参考电压约束 65
5 时序调整 69

图目录

图 3-1 菜单栏启动 FloorPlanner 17
图 3-2 Process 窗口启动 17
图 3-3 Start Page 窗口启动 18
图 3-4 FloorPlanner 界面 19
图 3-5 File 菜单 20
图 3-6 Open Physical Constraints 20
图 3-7 Tools 菜单 20
图 3-8 Back-annonate Physical Constraints 对话框 21
图 3-9 反标 Port 布局信息 21
图 3-10 Constraints 菜单 22
图 3-11 原语查找对话框 22
图 3-12 新建原语组对话框 23
图 3-13 正确原语组对话框 24
图 3-14 无效位置 24
图 3-15 无效位置 24
图 3-16 创建相对位置组对话框 25
图 3-17 正确的相对组对话框 25
图 3-18 预留约束 26
图 3-19 时钟约束 26
图 3-20 创建全局时钟约束（GW5AT-138） 27
图 3-21 创建高速时钟约束（GW5AT-138） 28
图 3-22 参考电压约束 28
图 3-23 View 菜单 29
图 3-24 Windows 菜单 29
图 3-25 Summary 窗口 30
图 3-26 Netlist 窗口 31
图 3-27 BUS 和非 BUS 结合显示 32
图 3-28 层级显示 32
图 3-29 时序路径显示 33
图 3-30 Netlist 右键功能 34

图 3-31 GW5AT-138-FPBGA676A Package view 窗口 35
图 3-32 Package View 右键功能 36
图 3-33 差分对显示 36
图 3-34 Top View 37
图 3-35 Bottom View 37
图 3-36 Chip Array 窗口 38
图 3-37 网格模式约束 39
图 3-38 宏单元模式约束 39
图 3-39 原语模式约束 40
图 3-40 Chip Array 右键功能 42
图 3-41 Show Place View 显示 42
图 3-42 鼠标悬浮显示 43
图 3-43 右键高亮 44
图 3-44 时序路径高亮显示 45
图 3-45 I/O 约束窗口 46
图 3-46 原语约束窗口 47
图 3-47 组约束窗口 47
图 3-48 预留约束窗口 48
图 3-49 时钟约束窗口 48
图 3-50 全局时钟约束窗口 49
图 3-51 高速时钟约束窗口 49
图 3-52 Vref 约束窗口 50
图 3-53 Message 窗口 50
图 4-1 新建约束文件 51
图 4-2 选择器件 52
图 4-3 保存输出文件 53
图 4-4 拖拽到 Chip Array 创建 I/O Constraints 55
图 4-5 拖至 Package View 创建 I/O Constraints 56
图 4-6 拖拽到 Chip Array 创建 Primitive Constraints 57
图 4-7 Group Constraints 编辑器右键菜单 57
图 4-8 创建 Primitive Group Constraints 58
图 4-9 Primitive Group Constraints 59
图 4-10 Relative Group Constraints 创建 60
图 4-11 Relative Group Constraints 61
图 4-12 创建 Resource Reservation 约束 61
图 4-13 Resource Reservation 62
图 4-14 Clock Net Constraints 约束创建 63
图 4-15 Clock Net Constraints 约束 63

图 4-16 GCLK Primitive Constraints 创建 64
图 4-17 GCLK Primitive Constraints 64
图 4-18 HCLK Primitive Constraints 创建 65
图 4-19 HCLK Primitive Constraints..... 65
图 4-20 Vref Constraints 创建..... 65
图 4-21 拖拽至 Chip Array 窗口生成 Vref Constraints Location 信息 66
图 4-22 拖拽至 Package View 窗口生成 Vref Constraints location 信息 67
图 4-23 Vref Constraints 名字重复 68
图 5-1 读取时序路径文件..... 70
图 5-2 高亮关键路径操作..... 71
图 5-3 关键路径信号流向示意图..... 71
图 5-4 调整后的位置信息..... 72

表目录

表 1-1 术语、缩略语 2
表 2-1 DCS/DCE 可约束 Position 13

1 关于本手册

1.1 手册内容

本手册主要描述高云®半导体 FloorPlanner 的界面和使用以及 Arora V FPGA 产品物理约束语法规范，旨在帮助用户快速实现物理约束。因软件版本更新，部分信息可能会略有差异，具体以用户软件版本信息为准。

1.2 相关文档

通过登录高云半导体网站 www.gowinsemi.com.cn 可下载、查看以下相关文档：

- DS981，GW5AT 系列 FPGA 产品数据手册
- DS1103，GW5A 系列 FPGA 产品数据手册
- DS1104，GW5AST 系列 FPGA 产品数据手册
- SUG100，Gowin 云源软件用户指南
- UG982，GW5AT-138 器件 Pinout 手册
- UG985，GW5A-25 器件 Pinout 手册
- UG986，GW5AST-138 器件 Pinout 手册
- UG306，Arora V 时钟资源(Clock)用户指南
- UG704，Arora V FPGA 产品编程配置手册

1.3 术语、缩略语

本手册中的相关术语、缩略语及相关释义请参见表 1-1。

表 1-1 术语、缩略语

术语、缩略语	全称	含义
BSRAM	Block SRAM	块状静态随机存储器
CFU	Configurable Function Unit	可配置功能单元
CLKDIV	Clock Divider	时钟分频器
CLS	Configurable Logic Section	可配置逻辑块
DCS	Dynamic Clock Selector	动态时钟选择器
DDRDLL	Double Data Rate Delay-locked Loop	双倍数据速率延时锁相环
DLLDLY	DLL Delay	DLL 延迟
DQS	Bidirectional Data Strobe Circuit for DDR Memory	DDR 存储器双向数据选通电路
FloorPlanner	FloorPlanner	物理约束编辑器
FPGA	Field Programmable Gate Array	现场可编程门阵列
GCLK	Global Clock	全局时钟
I/O	Input/Output	输入/输出
IDE	Integrated Development Environment	集成开发环境
LUT	Look-up Table	查找表
LW	Long Wire	时钟长线资源
PLL	Phase-locked Loop	锁相环
SSRAM	Shadow SRAM	分布式静态随机存储器
VREF	Voltage Reference	参考电压

1.4 技术支持与反馈

高云半导体提供全方位技术支持，在使用过程中如有任何疑问或建议，可直接与公司联系：

网址：www.gowinsemi.com.cn

E-mail：support@gowinsemi.com

Tel：+86 755 8262 0391

2 物理约束语法规范

2.1 I/O 位置约束

I/O 位置约束可将 port 约束到指定 IO 位置处。具体 IO 位置可参考对应器件的 Pinout 手册。

- UG982，GW5AT-138 器件 Pinout 手册
- UG985，GW5A-25 器件 Pinout 手册
- UG986，GW5AST-138 器件 Pinout 手册

语法

IO_LOC "obj_name" obj_location [exclusive];

约束元素

obj_name

obj_name 取 port 的 name 作为 obj_name。

obj_location

obj_location 为 IO 位置，如 “A11”、“B12”等，若指定多个位置，位置之间需要用英文逗号分隔，如“A11,B2”。

exclusive

exclusive 为可选项，在约束位置之后，表明该约束语句中的 obj_location 仅可以放置 obj_name 指定的原语。

注！

当 obj_name 为 escaped name 格式（以反斜线开头，空格结尾）时，obj_name 两边需加上英文引号。

应用举例

示例 1

IO_LOC "io_1" A1;

// 对象 io_1 被约束在封装 A1 的位置。

示例 2

IO_LOC "io_1" A1, B14, A15;

// 对象 io_1 被约束在封装 A1、B14、A15 的位置，布局时将取三个位置中第一个合理位置进行布局。

示例 3

IO_LOC "io_2" A1 exclusive;

// 对象 io_2 被约束在封装 A1 处，且 A1 位置仅可以被 io_2 所占用。

示例 4

IO_LOC "io_2" A1, B14, A15 exclusive;

// 对象 io_2 被约束在封装 A1、B14、A15 处，且 A1、B14、A15 三个位置仅可以被 io_2 占用。

2.2 I/O 属性约束

I/O 属性约束，用于设定 I/O 的各种属性值。如 port 的电平标准 IO_TYPE，上拉/下拉模式 PULL_MODE，驱动能力 DRIVE 等，详细属性设置标准请参考 *DS981，GW5AT 系列 FPGA 产品数据手册*。

语法

IO_PORT " obj_name " attribute = attribute_value;

一个约束语句中可设定多个属性，各个属性之间使用空格分隔。

约束元素

obj_name

obj_name 取 port 的 name 作为 obj_name。

attribute 和 attribute value

需要属性约束的 port 的属性和属性值。具体可约束属性以及属性值可参考文档 *UG304，Arora V 可编程通用管脚(GPIO)用户指南*。

应用举例

示例 1

IO_PORT "port_1" IO_TYPE = LVTTL33;

// 设置 port_1 的 IO_TYPE 为 LVTTL33。

示例 2

IO_PORT "port_2" IO_TYPE = LVTTL33 PULL_MODE =KEEPER;

// 设置 port_2 的 IO_TYPE 为 LVTTL33， PULL_MODE 属性值为 KEEPER。

2.3 原语位置约束

Primitive Constraints 用于将原语布局到指定的 GRID 位置，可以通过 Primitive Constraints 对 LUT、BSRAM、SSRAM、DSP、PLL、DQS 等进行约束。

语法

INS_LOC "obj_name" obj_location [exclusive];

约束元素

obj_name

约束原语的 name。

obj_location

obj_location 包含如下几类：

1. LUT 约束位置
   - 单一位置信息，指定到 LUT，如：RxCy[0-3][A-B]；
   - 位置信息为一个范围，指定多行或多列。
     - 包含多个 CLS 或 LUT：“RxCy”、“RxCy[0-3]”；
     - 指定多行：“R[x:y]Cm”、“R[x:y]Cm[0-3]”、“R[x:y]Cm[0-3][A-B]”；
     - 指定多列：“RxC[m:n]”、“RxC[m:n][0-3]”、“RxC[m:n][0-3][A-B]”；
     - 指定多行多列：“R[x:y]C[m:n]”、“R[x:y]C[m:n][0-3]”、“R[x:y]C[m:n][0-3][A-B]”。

注!

- LUT 位置信息中的 x、m 指的是 GRID 的行信息；
- LUT 位置信息中的 y、n 指的是 GRID 的列信息；
- LUT 位置信息中的 R 代表 GRID 的行，C 代表 GRID 的列；
- LUT 位置信息中的 0-3 表示一个 GRID 某个 CLS 的编号；
- LUT 位置信息中的 A-B 表示一个 CLS 位置中的具体 LUT 位置的编号。

2. PLL 约束位置

对于 PLL 约束位置信息书写格式为“PLL_L”或“PLL_R”，若左边可放置多个 PLL，可设为“PLL_L[0]”、“PLL_L[1]”…，若右边可放置多个 PLL，可设为“PLL_R[0]”、“PLL_R[1]”…

3. BSRAM 约束位置

BSRAM 约束位置信息为“BSRAM_R10[0]”（第 10 行第一个 BSRAM），“BSRAM_R10[1]”…

4. DSP 约束位置

DSP 约束位置格式为“DSP_R37[0]”（第 37 行第一个 DSP Block），“DSP_R37[1]”… 若需指定 MULT12X12 时，可标记为：DSP_R37[0][A]或 DSP_R37[0][B]。

注！

一个 DSP Block 位置包括两个 macro，一个 macro 指的是一个 MULT12X12 可放置的位置。

5. DDRDLL 约束位置

DDRDLL 约束位置信息为“DDRDLLM_TL”，“DDRDLLS0_TL”…

注！

在一条约束语句中，可包含多个 obj_location，使用“,”分隔。

exclusive

关键字“exclusive”为可选项，在约束位置之后，表明该约束语句中的 obj_location 仅可以放置 obj_name 指定的原语。

应用举例

示例 1

```
INS_LOC "lut_1" R5C10[0][A];
```

// lut_1 被约束在 R5C10 的第 1 个 CLS 的第 1 个 LUT 的位置。

示例 2

```
INS_LOC "ins_2 " R5C6[2] exclusive;
```

// ins_2 被约束在 R5C6 的第 3 个 CLS 的位置，且该位置仅可以放置该原语。

示例 3

```
INS_LOC "ins_3" R[2:6]C1;
```

// ins_3 被约束在行坐标第二行到第六行，列坐标第一列的区域位置。

示例 4

INS_LOC “ins_4” R[1:4]C[2:6] exclusive;

// ins_4 被约束在行坐标为第二行到第五行，列坐标为第三列到第七列之间的区域位置，且该区域位置仅能被该原语所占用。

示例 5

INS_LOC "ins_5" R[1:4]C[2:6][1];

// ins_5 被约束在行坐标第二行到第五行，列坐标第三列到第七列之间的区域位置的任意一个 GRID 位置的第 2 个 CLS 中。

示例 6

INS_LOC "reg_name" B14;

// reg_name 被约束至 IO 的封装位置 B14。

示例 7

INS_LOC "pll_name" PLL_L[0];

// 通过对 PLL 的 INS_LOC 约束，约束 pll_name 至左边 PLL 的第一个位置。

示例 8

INS_LOC "bsram_name" BSRAM_R10[2];

// 通过对 BSRAM 的 INS_LOC 约束，约束 bsram_name 至第 10 行的第 3 个BSRAM 位置处。.

示例 9

INS_LOC "dsp_name" DSP_R19[2];

// 通过对 DSP 的 INS_LOC 约束，约束 dsp_name 位置第 19 行第 3 个 DSP Block。

2.4 组约束

Group Constraints 包括 Primitive Group Constraints 和 Relative Group Constraints，如下所述。

2.4.1 原语组约束

Primitive Group 约束用于定义一个组约束，组是包含各类 Instance 对象的集合。通过 Primitive Group 约束，可将普通 Instance 如 LUT、DFF 等，或 Buffer、IOLOGIC 等添加到一个组中，并可通过约束该组的位置实现对该组中所有的对象的位置约束。

语法

GROUP 的定义：

GROUP group_name = { "obj_names" } [exclusive];

添加 Instance 到组中：

GROUP group_name += { "obj_names" } [exclusive];

约束组的位置：

GRP_LOC group_name group_location[exclusive];

注！

当 group_name 为 escaped name 格式（以反斜线开头，空格结尾）时， group_name 两边需加上引号。

约束元素

group_name

定义一个 name 作为该组的 name。

obj_name

obj_name 用于将指定的 Instance 对象添加到组中。

group_location

指定该 group 的约束位置，group_location 可取 IOB 和 GRID 的位置。

exclusive

关键字“exclusive”为可选项，在组定义语句或位置约束语句之后；

一个对象可以被多个组包含，但在组定义语句后添加“exclusive”关键字，表示该组内的对象仅可被该组所包含；

在位置约束语句之后使用“exclusive”，表示该约束位置仅可被该组内的对象所占用。

应用举例

示例 1

GROUP group_1 = { "ins_1" "ins_2" "ins_3” "ins_4" };

// 创建一个名为 group_1 的组，添加对象 ins_1、ins_2、ins_3、ins_4 到该组中。

示例 2

GROUP group_2 = { "ins_5" "ins_6" "ins_7" } exclusive;

// 创建一个名为 group_2 的组，对象 ins_5、ins_6、ins_7 属于且仅可属于该组。

示例 3

GROUP group_1 += { "io_1" "io_2"};

// 添加 io_1、io_2 到组 group_1 中。

示例 4

GRP_LOC group_1 R3C4, A14, B4;

// 组 group_1 中的对象可布局在 R3C4、A14、B4 位置处。

示例 5

GRP_LOC group_2 R[1:3]C[1:4] exclusive;

// 组 group_2 中的 Instance 对象可布局在区域 R[1:3]C[1:4]的范围内，且该范围仅可布局 group_2 中的 Instance 对象。

2.4.2 相对组约束

通过 Relative Group Constraints，可实现对 instance 对象的相对位置约束。

语法

定义 Relative 约束的组：

REL_GROUP group_name = { "obj_names" };

添加 instance 对象到已定义的组中：

REL_GROUP group_name += { "obj_names"};

对组中的 instance 进行相对位置约束：

INS_RLOC “obj_name” relative_location;

约束元素

obj_name

约束对象的名称。

relative_location

行列相对位置信息描述。

应用举例

REL_GROUP grp_1 = { "ins_1" "ins_2" "ins_3" "ins_4" };

INS_RLOC "ins_1" R0C0;

INS_RLOC "ins_2" R2C3;

INS_RLOC "ins_3" R3C5;

// 定义一个名为 grp_1 的组约束，并添加 ins_1、ins_2、ins_3、ins_4 到 grp_1 中。以 ins_1 为相对位置原点 R0C0，ins_2 约束到相对 ins_1 的 R2C3 处，ins_3 约束到相对 ins_1 的 R3C5 处。

2.5 资源预留约束

通过 Resource Reservation 约束，可保留指定的位置或区域以避免在布局中使用。

语法

LOC_RESERVE location [res_obj];

应用举例

示例 1

LOC_RESERVE R2C3[0][A] -LUT;

LOC_RESERVE R2C3[0][A] -REG;

示例 2

LOC_RESERVE IOR3, IOR6, R2C3, R3C4;

示例 3

LOC_RESERVE R[2:5]C[3:6], R3C[8:9];

// 以上示例中约束的位置信息将会在布局阶段被保留。

2.6 参考电压约束

芯片支持外部参考电压输入，对整个 BANK 有效。Vref Constraints 约束可用于对外部参考电压的输入管脚的名称和位置进行约束。

注！

- 可设置外部参考电压的输入管脚位置必须有 IOLOGIC 资源；
- Vref Constraints 和 Port 属性约束联合使用才有效。当 input 或 inout 类型的单端 Port，IO Type 为 SSTL/HSTL 时 Vref 属性可设置为创建的 Vref Constraints，表示该 Port 的参考电压使用 Vref Constraints 位置输入的外部参考电压。

语法

USE_VREF_DRIVER vref_name [location];

约束元素

vref_name

自定义的 VREF pin name

location

GRID 中任意含 IOLOGIC 资源的 I/O 位置可作为 VREF pin 约束的 location。

应用举例

示例 1

```
USE_VREF_DRIVER vref_pin;
IO_PORT "port_1" IO_TYPE = SSTL25_I VREF=vref_pin;
IO_PORT "port_2" IO_TYPE = SSTL25_I VREF=vref_pin;
```

// 定义一个名为“vref_pin”的 VREF pin，设置 port_1 与 port_2 的 VREF 属性为 vref_pin。

示例 2

```
USE_VREF_DRIVER vref_pin C7;
IO_PORT "port_1" IO_TYPE = SSTL25_I VREF=vref_pin;
IO_PORT "port_2" IO_TYPE = SSTL25_I VREF=vref_pin;
```

// 定义一个名为“vref_pin”的 VREF pin，将其约束到 C7，设置 port_1 与 port_2 的 VREF 属性为 vref_pin，port_1 与 port_2 将布局到 C7 所在的 bank 上。

2.7 全局时钟分配约束

Clock Net Constraints 是对于设计中特定 net 走全局时钟线或不走时钟线的约束。

- BUFG[0-15]表示约束 net 走全局时钟线；
- LOCAL_CLOCK 表示约束 net 不走时钟线。

CLK 为连接时钟引脚的信号，CE 为连接时钟使能引脚的信号，SR 为连接 SET、RESET、CLEAR、PRESET 引脚的信号，LOGIC 为连接逻辑输入引脚的信号。

语法

CLOCK_LOC "net_name" global_clocks = signal_type;

约束元素

net_name

net 的名字。

global_clocks

BUFG[0-15]：走具体一条全局时钟线；

BUFG：走全局时钟线

LOCAL_CLOCK：不走时钟线。

signal_type

CLK：signal_type 为时钟引脚的 net；

CE：signal_type 为时钟使能引脚的 net；

SR：signal_type 为 SET、RESET、CLEAR、PRESET 引脚的 net；

LOGIC：signal_type 为以上 fanout 之外的 net。

指定多个 signal_type，可使用“|”符号进行分隔。

注！

若 global_clocks 选择的是 LOCAL_CLOCK，则 signal_type 不可选。

应用举例

示例 1

CLOCK_LOC "net" BUFG = CLK|CE;

NET_LOC "net" BUFG = CLK|CE;

// 约束 CLOCK 对象“net”的 signal_type 为时钟引脚或时钟使能引脚的 net 走全局时钟线。

示例 2

CLOCK_LOC "net" LOCAL_CLOCK;

// 约束 CLOCK 对象“net”不走时钟线。

2.8 全局时钟原语约束

GCLK Primitive Constraints 用于将 DCS、DCE 需要进行布局的全局时钟对象约束到指定的位置。

语法

INS_LOC "obj_name" position;

约束元素

obj_name

约束对象的名称。

position

表 2-1 DCS/DCE 可约束 Position

器件	位置	
	DCE	DCS
GW5AT-138	PTR0、PTR1、PTR2 PTR3、PTR0[0~5]、PTR1[0~5]、PTR20~5]、PTR3[0~5]、PBR0、PBR1、PBR2、PBR3、PBR0[0~5]、PBR1[0~5]、PBR2[0~5]、PBR3[0~5]、PG、SG、PG[0~11]、SG[0~15]	PTR0、PTR1、PTR2 PTR3、PTR0[0~1]、PTR1[0~1]、PTR2[0~1]、PTR3[0~1]、PBR0、PBR1、PBR2、PBR3、PBR0[0~1]、PBR1[0~1]、PBR2[0~1]、PBR3[0~1]、PG、PG[0~3]
GW5A-25	TOPLEFT、TOPRIGHT、BOTTOMLEFT、BOTTOMRIGHT、STOP、SBOTTOM	TOPLEFT、TOPRIGHT、BOTTOMLEFT、BOTTOMRIGHT

注！

DCS/DCE 可约束 Position 的具体位置可参考文档：*UG306，Arora V 时钟资源(Clock)用户指南*。

应用举例

INS_LOC "dcs_name" PTR0[1];

// 约束 DCS 对象 dcs_name 到 PTR0[1]位置。

2.9 高速时钟原语约束

通过 HCLK Primitive Constraints 约束，可将 CLKDIV、DLLDLY 约束到相关高速时钟位置。CLKDIV、DLLDLY 约束位置与普通 instance 对象约束位置不同，使用“BOTTOMSIDE”，“LEFTSIDE”，“RIGHTSIDE”表示约束位置的边。

语法

INS_LOC "obj_name" position;

约束元素

obj_name

取 CLKDIV、DLLDLY 的 instance name 作为 obj_name。

position

CLKDIV 可约束 Position (GW5AT-138)：BOTTOMSIDE[0~7]、LEFTSIDE[0~7]、RIGHTSIDE[0~7]

CLKDIV 可约束 Position (GW5A-25)：BOTTOMSIDE[0~3]、LEFTSIDE[0~3]

DLLDLY 可约束 Position (GW5AT-138)：BOTTOMSIDE[0~3]、LEFTSIDE[0~3]、RIGHTSIDE[0~3]

DLLDLY 可约束 Position (GW5A-25)：BOTTOMSIDE[0~1]、LEFTSIDE[0~1]、RIGHTSIDE[0~1]、TOPSIDE[0~1]

应用举例

INS_LOC "clkdiv_name" LEFTSIDE[0];

// 将 clkdiv_name 布局到 LEFTSIDE[0]处。

2.10 其他约束

2.10.1 ADC 输入电压源约束

ADC 输入电压源可以来自外部 IO，其中 bank0/2/3/4/5/6/7 中的 IO 位置可以通过 ADC 输入电压源约束来指定 ADC 提供输入电压源的入口。

ADC 输入电压约束有两个约束语法，分别对应 bus0 和 bus1。其中 bus0 对应 bank0/6/7 中的 IO 位置，bus1 对应 bank2/3/4/5 中的 IO 位置。

注!

ADC 输入电压约束支持器件：GW5A-25。

语法

USE_ADC_SRC bus0 location

USE_ADC_SRC bus1 location

约束元素

location

location 为 IO 位置。

注！

- IOL25/IOL31/IOB45/IOR31/IOR5/IOT43/IOT3 这些 IO 位置不能使用；
- location 可以使用 pin 管脚名称形式也可以使用 pad 名称形式

应用举例

```
USE_ADC_SRC bus0 IOT48
//将 IO 位置 IOT48 作为 ADC 输入电压源的外部输入位置
USE_ADC_SRC bus1 IOR24
//将 IO 位置 IOR24 作为 ADC 输入电压源的外部输入位置
```

3 FloorPlanner

3.1 简介

FloorPlanner 是高云半导体面向市场自主研发的物理约束编辑器，支持对 I/O、Primitive（原语）、Group 等属性及位置信息的读取与编辑，同时可根据用户的配置生成新的布局与约束文件，文件中规定了 I/O 的属性信息，原语、模块的位置信息等。FloorPlanner 提供了简单快捷的布局与约束编辑功能，提高编写物理约束文件的效率，同时可以根据布局信息和时序路径进行时序优化。

FloorPlanner 功能特点：

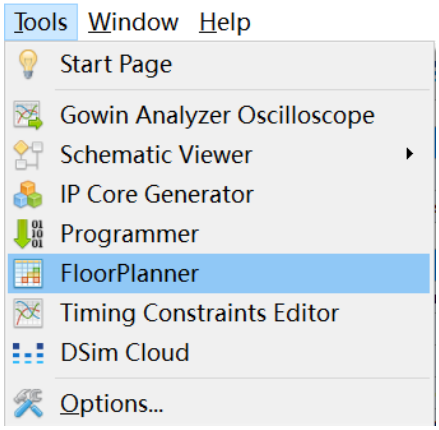
- 支持用户设计文件、约束文件的读入，约束文件的编辑，以及约束文件的输出
- 支持对用户设计文件中 IO Port、Primitive、Group 约束信息等的显示
- 支持用户新建、编辑、修改约束信息
- 支持 Chip Array 的网格模式、宏单元模式以及原语模式显示
- 支持依据 Package 信息的 Package View 显示
- 支持 Chip Array 和 Package View 的同步显示
- 支持约束位置信息的实时显示及区别显示
- 支持拖拽设置位置信息的功能
- 支持 IO Port 的属性配置功能，支持批量配置
- 支持 Clock Assignment 的显示、编辑功能
- 支持约束信息合法性检查的功能
- 支持 Back-annotate Physical Constraints 功能
- 支持手动调整时序

3.2 启动

可通过以下三种方式启动 FloorPlanner：

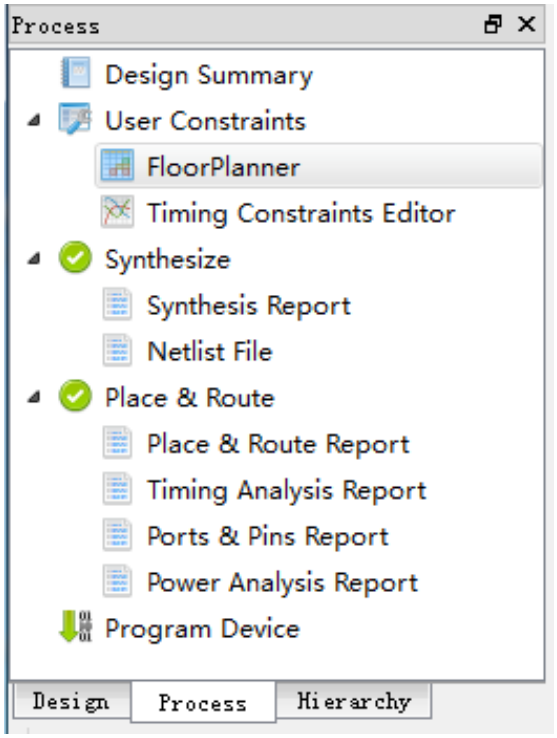
1. 单击“IDE > Tools”，打开“FloorPlanner”，如图 3-1 所示；

图 3-1 菜单栏启动 FloorPlanner



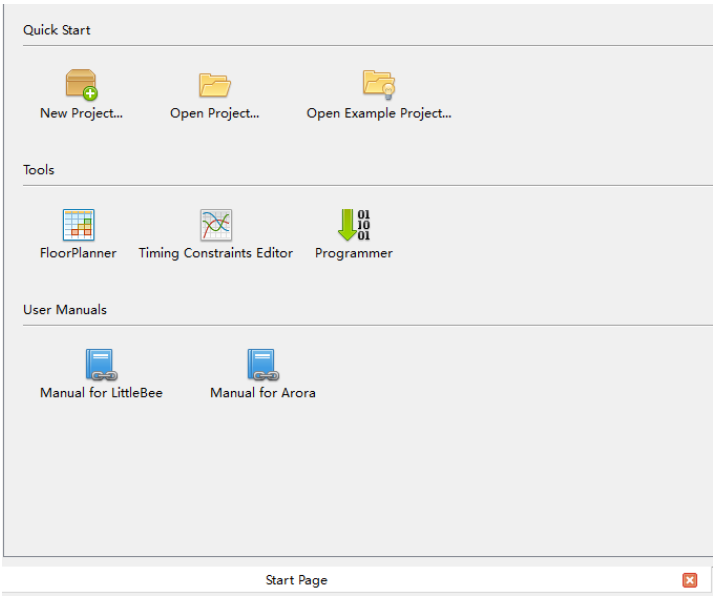
2. 建立工程在 Process 窗口运行 Synthesize 成功后，双击“FloorPlanner”，如图 3-2 所示。

图 3-2 Process 窗口启动



3. 单击“IDE > Start Page>Tools> FloorPlanner”，打开“FloorPlanner”，如图 3-3 所示。

图 3-3 Start Page 窗口启动



注!

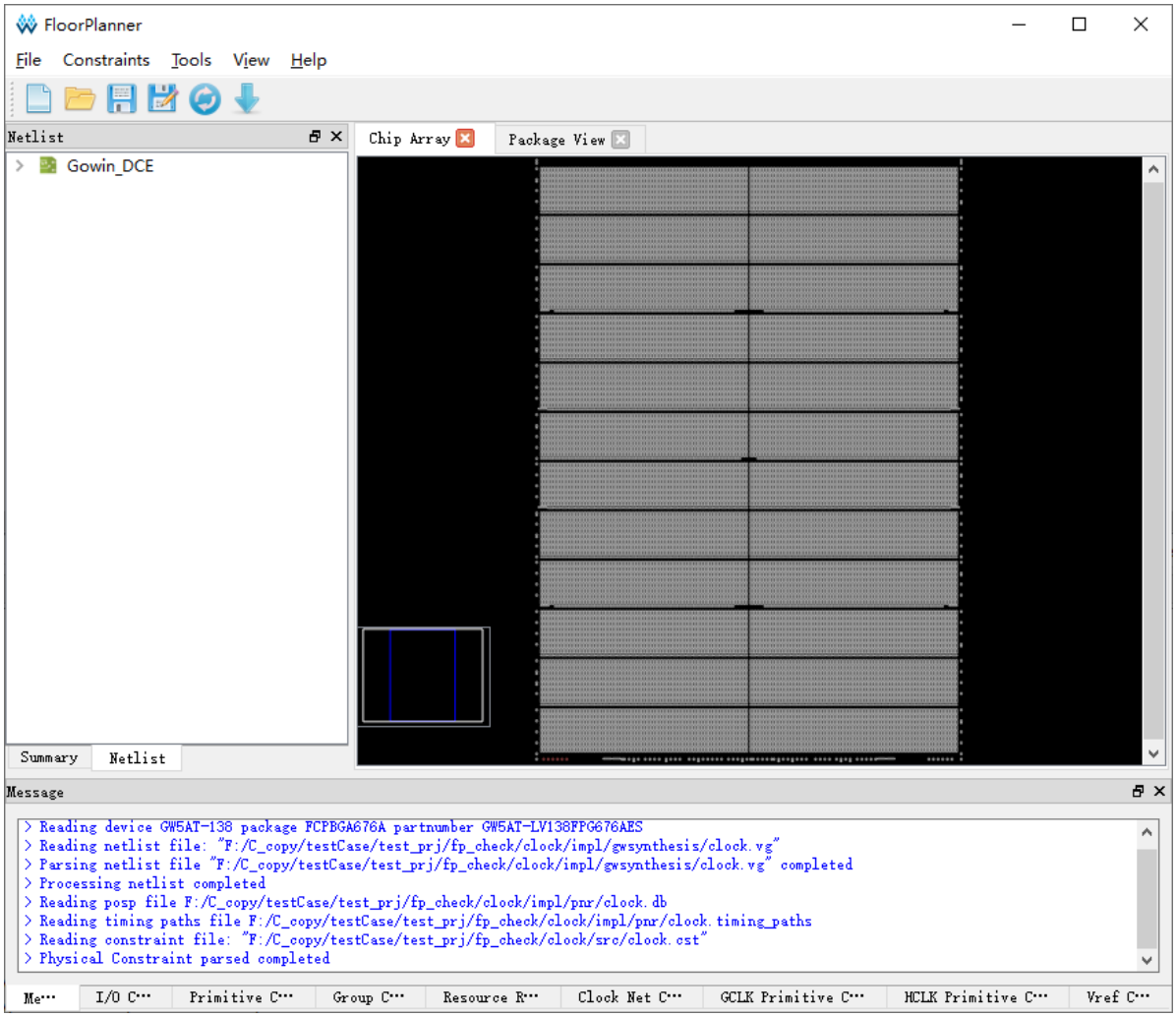
- 如需 FloorPlanner 进行约束，应先加载网表文件；
- 通过第一种和第二种方式启动 FloorPlanner 时，当前工程中的网表文件会自动加载；
- 通过第三种方式启动 FloorPlanner 时，需要通过“File > New”加载网表文件。

3.3 界面

新建或打开 FloorPlanner 界面（包含网表文件），如图 3-4 所示。

界面包括菜单栏、工具栏、Netlist 窗口、Project 窗口、Chip Array 窗口、Package View 窗口、Message 窗口以及各类约束编辑窗口等。

图 3-4 FloorPlanner 界面



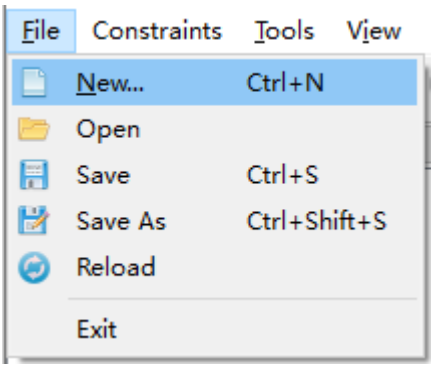
3.3.1 菜单栏

FloorPlanner 的菜单栏分为“File”、“Constraints”“Tools”、“View”及“Help”5 个子菜单。

File 菜单

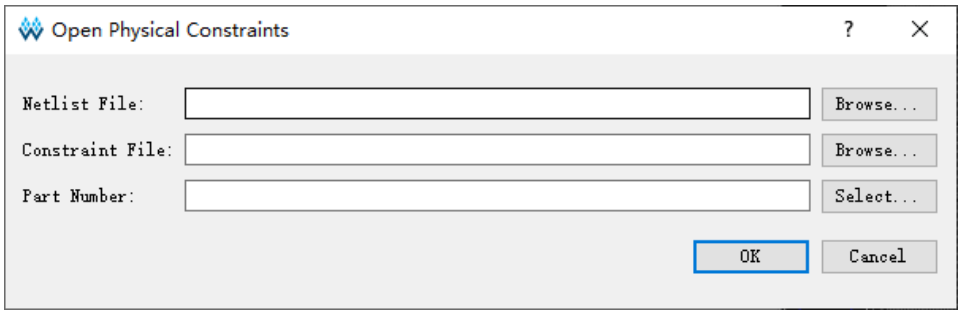
File 菜单如图 3-5 所示。

图 3-5 File 菜单



- New：新建约束，添加用户设计，设置器件信息；
- Open：打开约束，添加用户约束，设置器件信息如图 3-6；
- Reload：当在磁盘或工程中对物理约束文件、布局文件、时序路径文件进行修改后，可以重新加载；
- Save：当前约束信息的修改信息覆盖原约束文件；
- Save As：将当前约束信息的修改信息输出到用户指定的文件中，默认采用网表文件名作为约束文件名称，用户可修改；
- Exit：退出 FloorPlanner。

图 3-6 Open Physical Constraints

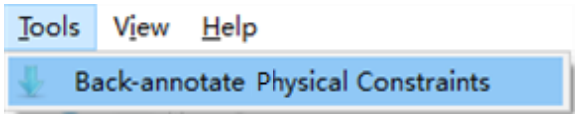


Tools 菜单

Tools 菜单如图 3-7 所示。

Back-annotate Physical Constraints：将各 Primitive 和 IO Port 布局信息反标至物理约束文件中。

图 3-7 Tools 菜单



1. 单击“Tools > Back-annotate Physical Constraints”弹出对象选择对话框如图 3-8 所示。Back-Annotate Physical Constraints 功能只有工程中运行 Place & Route 成功后，通过工程启动 FloorPlanner 才有效；
2. Back-annonate Physical Constraints 对话框中可以选择一个或者多个对象，单击“OK”按钮弹出“Save as”对话框，打印其布局信息至物理约束文件中；
3. 如图 3-9 所示，在 Back-annonate Physical Constraints 对话框中选择 Port 和 Port Attribute 后生成的物理约束文件。

图 3-8 Back-annonate Physical Constraints 对话框

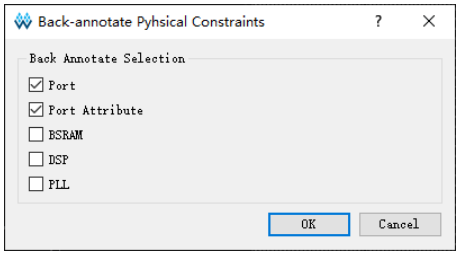
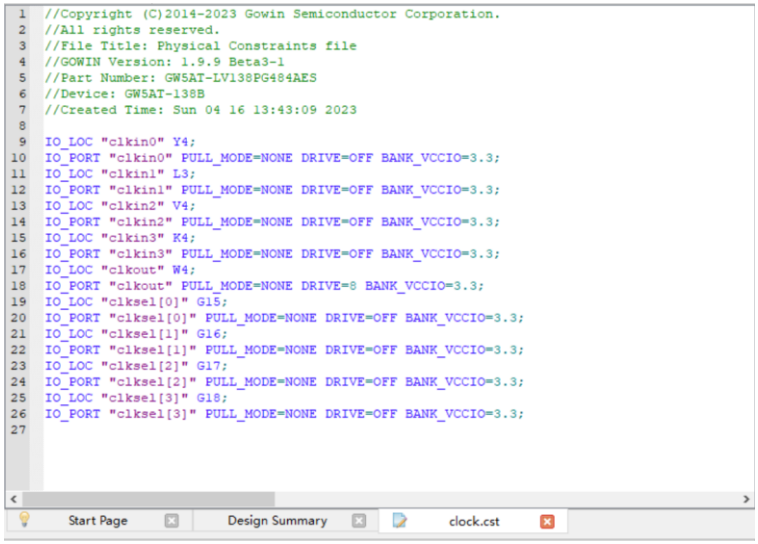


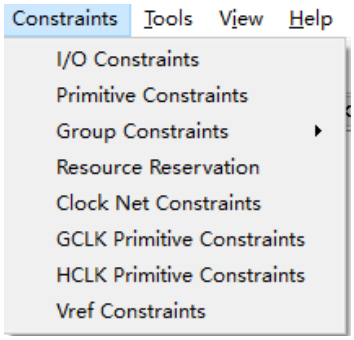
图 3-9 反标 Port 布局信息



Constraints 菜单

Constraints 菜单栏如图 3-10 所示。

图 3-10 Constraints 菜单



Primitive Constraints

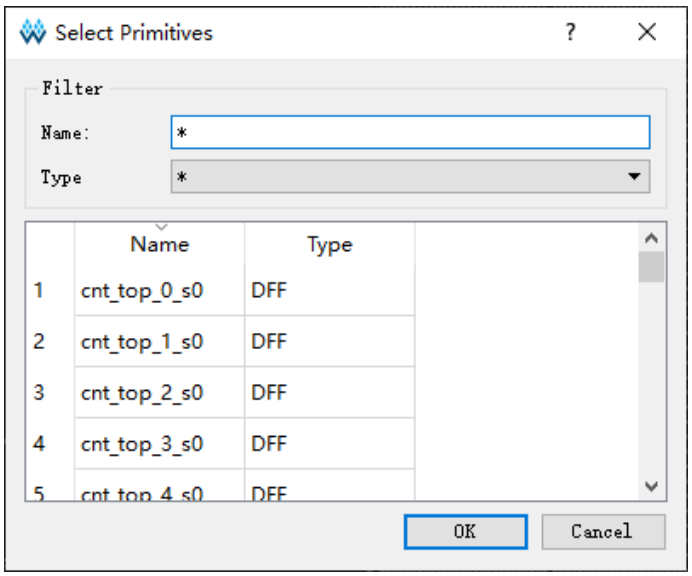
创建 Primitive 约束，选择 Primitive 创建对应的约束，右击选择 Select Primitive，可弹出如图 3-11 所示对话框。

1. 可通过 Primitives 名称或类型进行查找，选择对应的 Primitive；
2. 单击“OK”，产生约束信息，约束信息显示在主界面底部的“Primitive Constraints”约束编辑窗口中；
3. 用户可在编辑窗口中通过输入或拖拽的方式设置位置信息。

注！

所约束的位置在 Chip Array 窗口中呈浅蓝色高亮显示。

图 3-11 原语查找对话框



Group Constraints

Group Constraints，包括 New Primitive Group 和 New Relative Group，各功能介绍如下：

创建 Primitive Group。

1. 创建 Primitive Group 约束，右击选择 New Primitive Group，弹出如图 3-12 所示对话框；
2. 用户可设置 Group 的名称、包含的 Primitive 位置信息，以及 Group 的 Exclusive 信息；通过“ ”和“ ”两个按钮实现 Primitive 的添加和删除，正确创建的 Primitive Group 如图 3-13 所示；

注！

- Group 的名称、包含的 Primitive、Group 的位置为必填项；
- 可通过以下方式输入 Group 的位置信息：
  - 通过手动方式输入；
  - 建立 Group 约束前，在“Chip Array”窗口中，复制位置，粘贴到“New Primitive Group > Locations”中。

3. Primitive Group 创建配置完成后，单击“OK”，工具此时会对 Group 的位置信息进行语法检查。
   - 若位置信息不合理或不合法，会弹出如图 3-14 和如图 3-15 所示的提示框，用户需重新修改位置信息；
   - 若无错误提示，单击“OK”，在 Chip Array 中会显示可用的位置。
4. 新产生的组约束显示在主界面底部的“Group Constraints”约束编辑窗口中，在“Group Constraints”约束编辑窗口中，双击 Primitive Group 约束可打开如图 3-13 所示的对话框，重新进行编辑修改。

图 3-12 新建原语组对话框

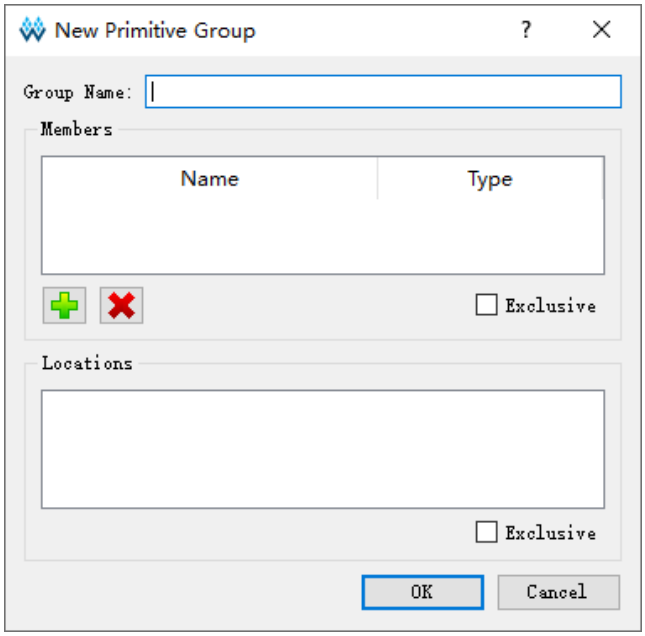


图 3-13 正确原语组对话框

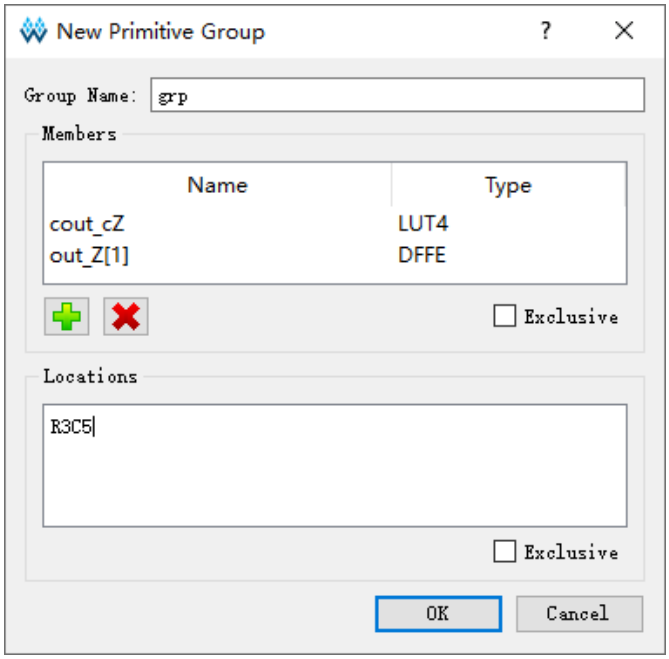


图 3-14 无效位置

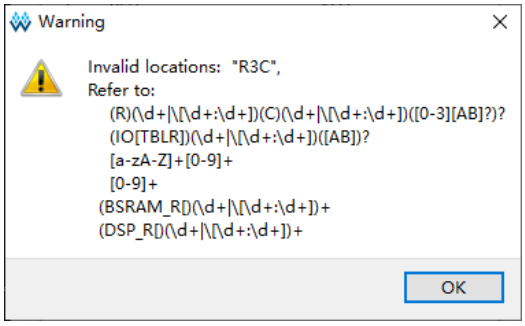
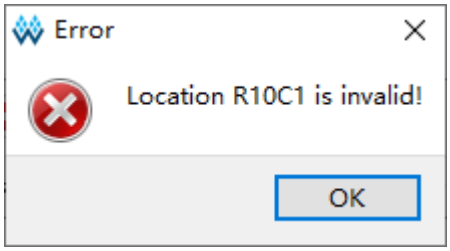


图 3-15 无效位置



创建 Relative Group

1. 创建 Relative Group 约束，右击选择“New Relative Group”，弹出如 图 3-16 所示对话框；
2. 用户可设置 Group 的名称、包含的 Primitive 以及各 Primitive 对应的相

对位置信息；可通过“ ”和“ ”实现 Primitive 的添加和删除，创建成功的 Relative Group 约束如图 3-17 所示；

注！

- Group 的名称、包含的 Primitive 及 Relative Location 为必填项；
- 可通过以下方式输入 Group 的位置信息：
  - 通过手动方式输入；
  - 在建立 Group 约束前，在“Chip Array”窗口中，复制位置，粘贴到“New Relative Group > Relative Location”中。

3. 配置完成后单击“OK”，产生约束信息。
4. 产生的约束信息显示在主界面底部的“Group Constraints”约束编辑窗口中；在编辑窗口中，双击约束，重新打开如图 3-17 所示的对话框，可进行编辑修改。

图 3-16 创建相对位置组对话框

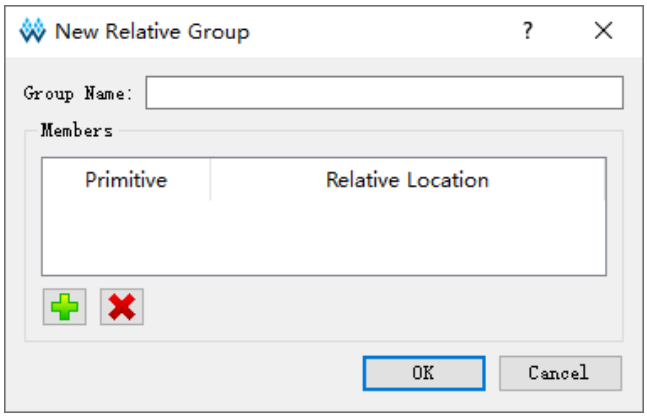
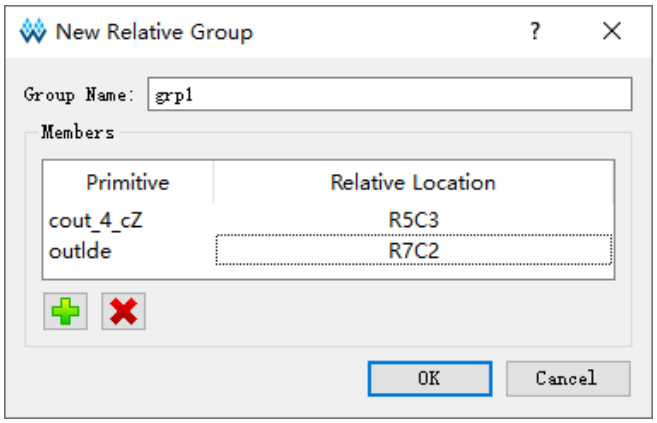


图 3-17 正确的相对组对话框



Resource Reservation

1. 创建 Resource Reservation 约束，在主界面底部的“Resource Reservation”约束编辑窗口右击选择 Reserve Resources，新建一条约

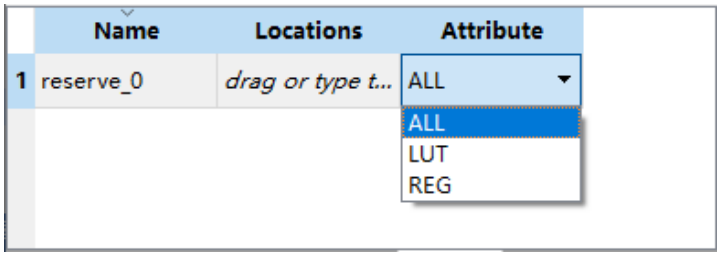
束；

2. 通过输入或拖拽的方式设置位置信息；
3. 双击“Attribute”栏或单击“Attribute”栏下拉框可设置预留位置的属性，如图 3-18 所示。

注！

Name 属性用于区分不同的预留约束，不可修改该名称。

图 3-18 预留约束



Clock Net Constraints

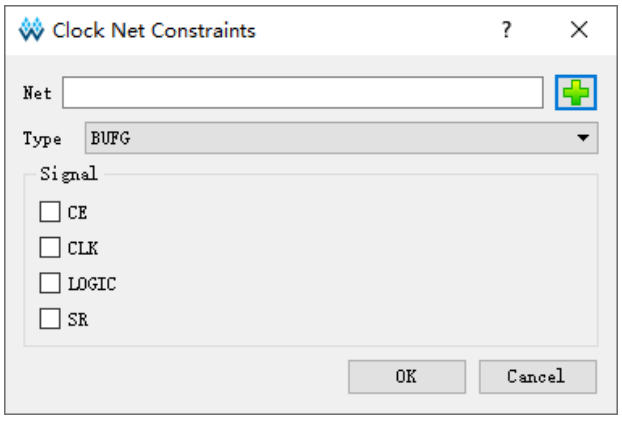
创建全局时钟分配约束，该约束的数目有限制，会对约束合法性时进行相应检查。右击选择“Clock Net Constraints”，可弹出如图 3-19 所示对话框，进行如下操作：

1. 单击“ ”按钮，选择对应 Net；
2. 通过“Type”下拉列表，选择“BUFG”、“BUFG[0]~[15]”、“LOCAL_CLOCK”；
3. 通过“CE”、“CLK”等复选框配置 Signal 类型，配置完成后，单击“OK”，产生约束信息，显示在主界面底部的“Clock Net Constraints”约束编辑窗口中，在编辑窗口中，双击，重新打开约束对话框，进行编辑。

注！

当 Type 选择 LOCAL_CLOCK 时 Signal 复选框为置灰不可配置状态。

图 3-19 时钟约束

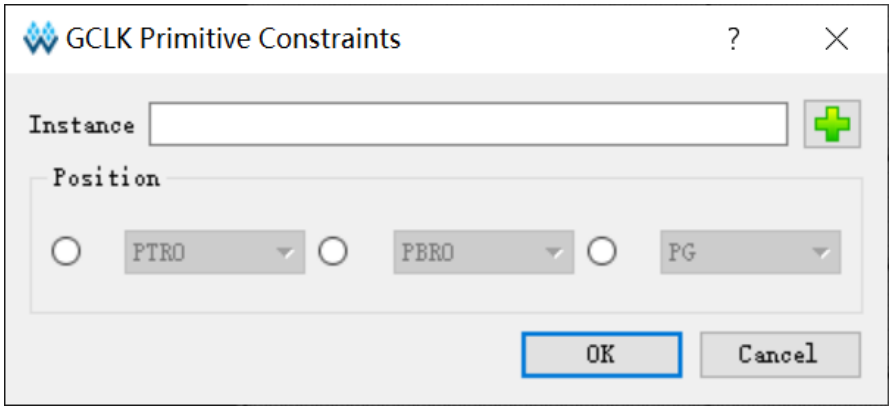


GCLK Primitive Constraints

用于创建针对全局时钟原语创建全局时钟约束。根据器件的全局时钟分布约束指定的 Instance 到具体的全局时钟，在主界面底部的“GCLK Primitive Constraints”约束编辑窗口右击选择“Select GCLK Primitive”，可弹出如图 3-20 所示对话框。相关操作如下所示：

1. 通过单击“ ”按钮，选择相应的 GCLK 原语，若设计无 GCLK 原语则无法添加；
2. 通过 Position 下的单选框以及对应的下拉列表配置全局时钟位置；
3. 单击“OK”，产生约束信息，显示在主界面底部的“GCLK Primitive Constraints”约束编辑窗口中，在编辑窗口中，双击，重新打开约束对话框，可进行约束编辑修改。

图 3-20 创建全局时钟约束（GW5AT-138）



注！

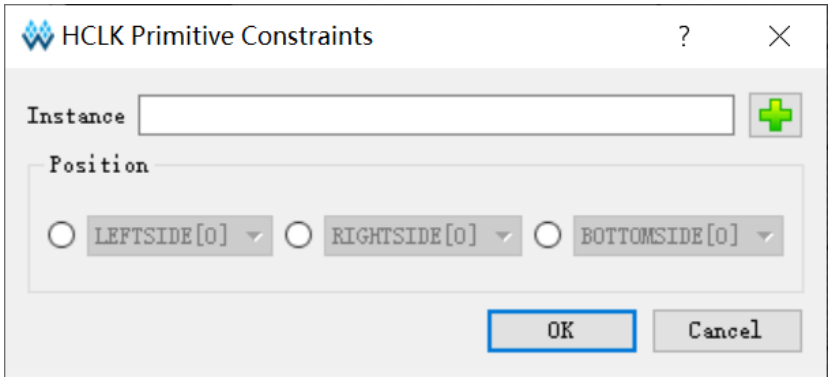
Position 根据器件不同可用 Position 不同，不同全局时钟原语可用 Position 也会不同。

HCLK Primitive Constraints

创建针对 HCLK 相关原语进行的约束，指定其约束在器件的高速时钟位置上。在主界面底部的“HCLK Primitive Constraints”约束编辑窗口右击选择 Select HCLK Primitive，可弹出如所图 3-21 示对话框。相关操作如下所示：

1. 用户可通过单击“ ”按钮选择相应的 HCLK 原语，若设计中无符合的原语则无法添加；
2. 通过 Position 下的单选框以及对应的下拉列表配置高速时钟位置；
3. 单击“OK”，产生约束信息，显示在主界面底部的“HCLK Primitive Constraints”约束编辑窗口中，在编辑窗口中，双击，重新打开约束对话框，可进行约束编辑修改。

图 3-21 创建高速时钟约束（GW5AT-138）



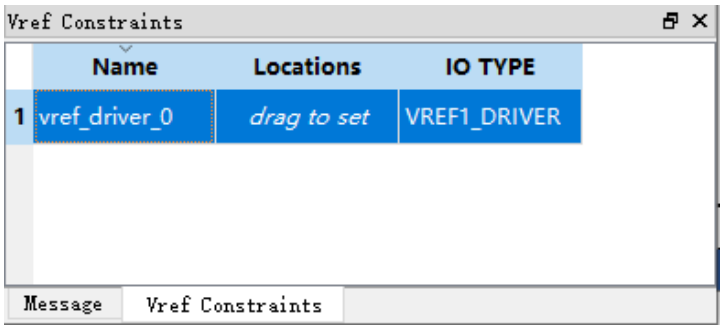
注！

Position 根据器件不同可用 Position 不同，不同高速时钟原语可用 Position 也会不同。

Vref Constraints

创建新的 Vref Driver，用于配置 IO Port 的参考电压，在主界面底部的“Vref Constraints”约束编辑窗口中右击选择 Define Vref Driver，新建一条约束，如图 3-22 所示。

图 3-22 参考电压约束



注！

- 可通过拖拽方式指定 Vref 约束位置；
- 可通过双击修改 Vref 的名称。

View 菜单

View 菜单如图 3-23 所示，主要用于控制工具条、窗口的显示以及 Chip Array 和 Package View 两个视图的放大、缩小等。各子菜单介绍如下：

- Toolbars：用于控制工具栏快捷按钮的显示；
- Windows：用于控制各个窗口的显示，如图 3-24 所示；
- Zoom In：用于放大 Chip Array 视图或 Package View 视图；
- Zoom Out：用于缩小 Chip Array 视图或 Package View 视图；
- Zoom Fit：按照窗口大小缩放 Chip Array 视图或 Package View 视图。

图 3-23 View 菜单

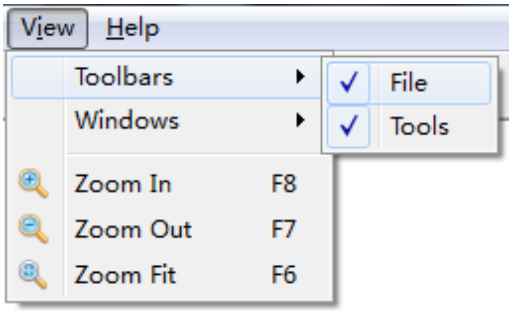
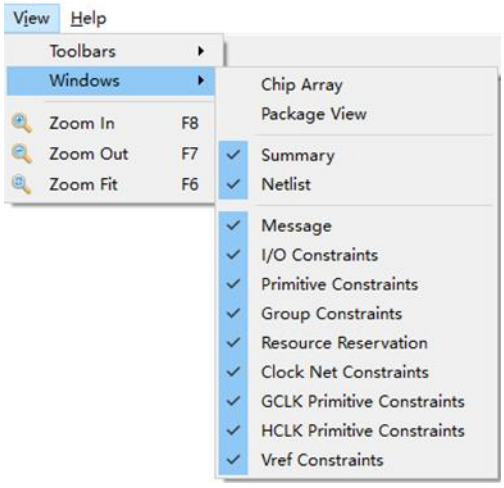


图 3-24 Windows 菜单



Help 菜单

Help 菜单用于提示软件的版本号及版权信息。

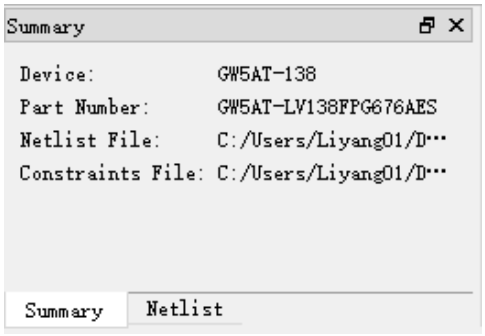
3.3.2 Summary 和 Netlist 窗口

Summary 和 Netlist 两个窗口可显示当前工程的器件信息、芯片型号信息、用户设计及约束文件的路径信息、Netlist 信息等。

Summary 窗口

Summary 窗口如图 3-25 所示，用于显示当前工程中所用的器件信息，包括 Device 和 Part Number，以及用户输入的设计文件和约束文件。

图 3-25 Summary 窗口



Netlist 窗口

Netlist 窗口如图 3-26 所示，以树形结构显示用户设计中的 Ports、Primitives、Nets、Module 和 Timing paths 以及对应的数量信息。

注！

- Port、Primitive 等名称采用全路径方式进行显示，默认按字母升序排序；
- Port 和 Net 的显示采用 Bus 和非 Bus 相结合的显示方式，如图 3-27 所示；
- Module 采用层级的方式显示，各 Module 后可显示各 Module 中各类型的 Instance 数目，如图 3-28 所示；
- 时序路径则按照 Slack 时间从小到大的顺序显示，如图 3-29 所示。

图 3-26 Netlist 窗口

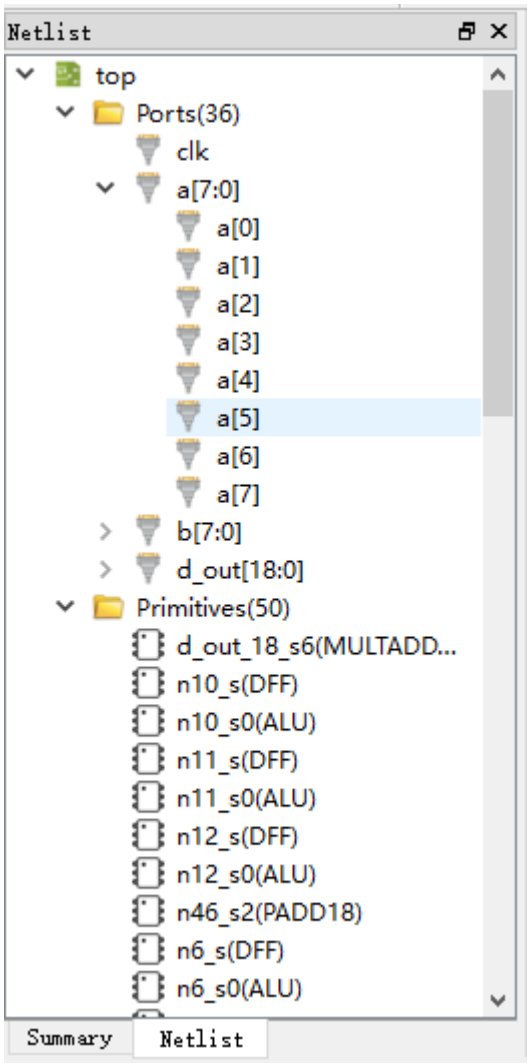


图 3-27 BUS 和非 BUS 结合显示

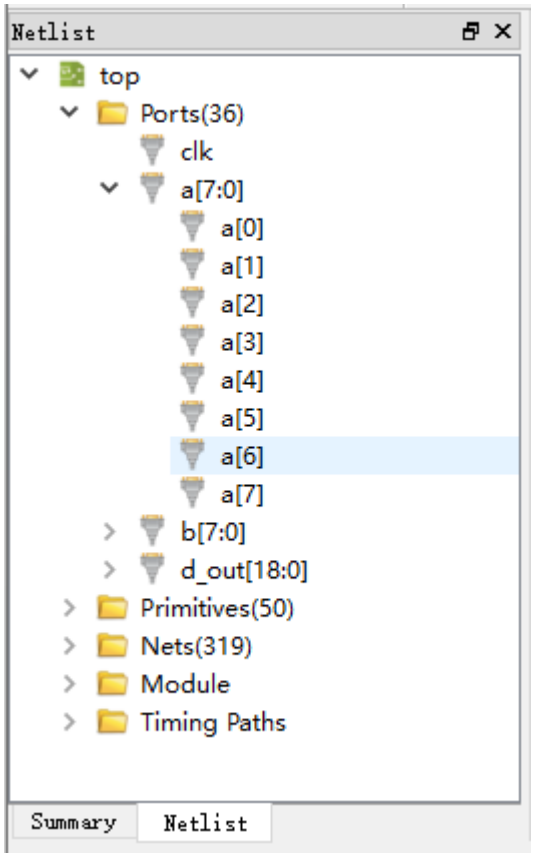


图 3-28 层级显示

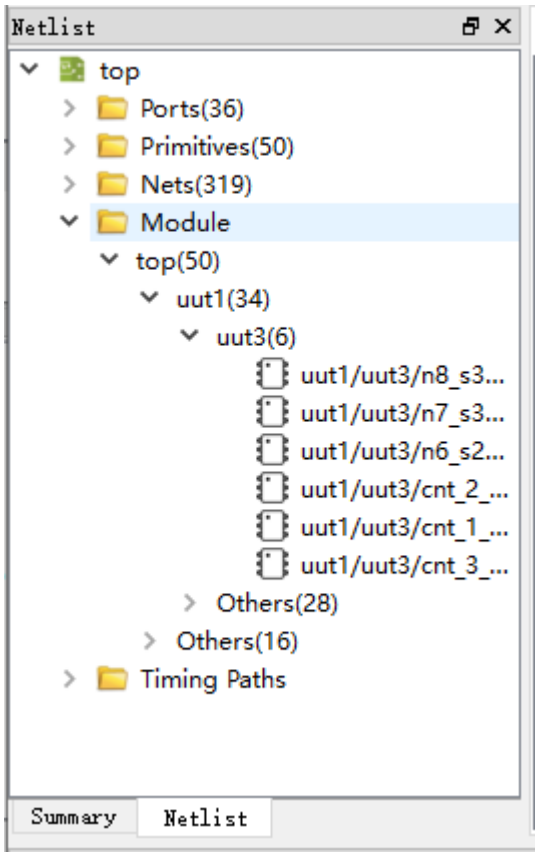
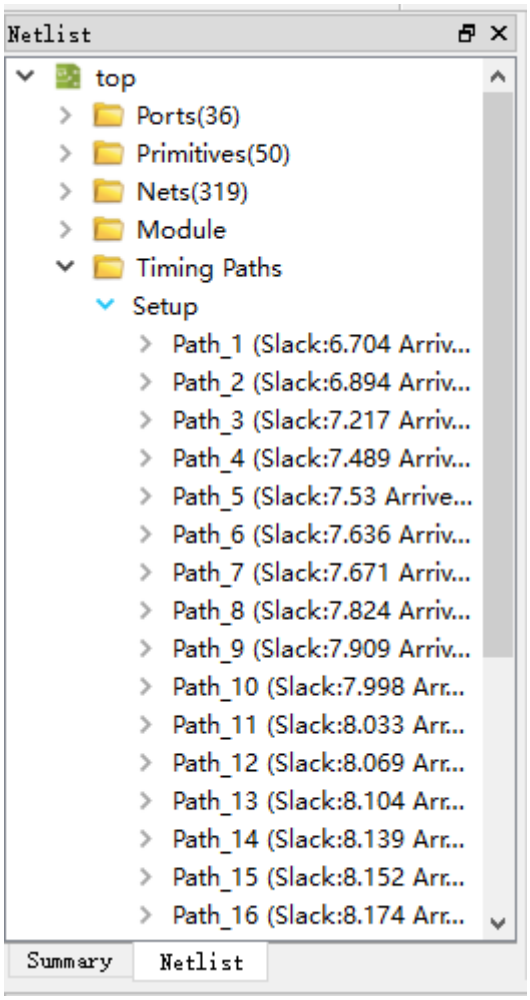


图 3-29 时序路径显示



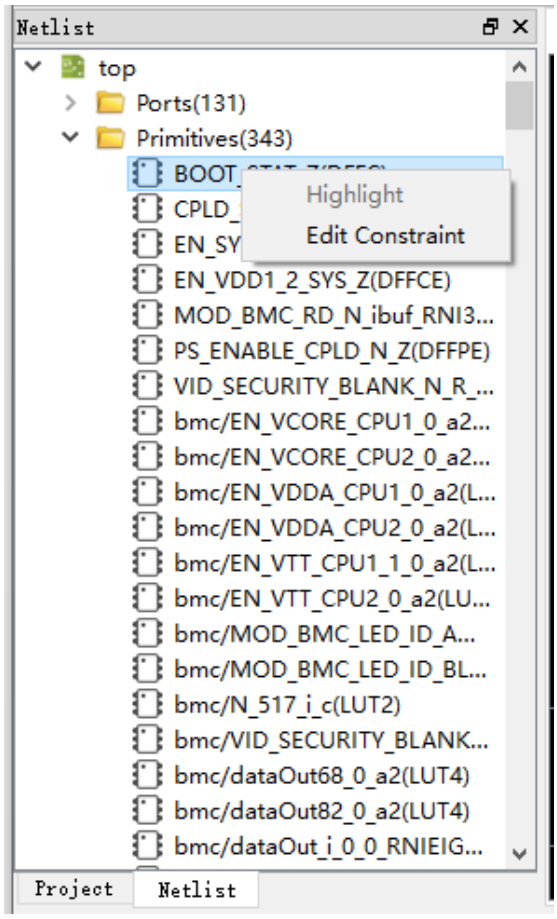
Netlist 窗口提供右键菜单功能，具有如下功能：

- Highlight：可实现在 Chip Array 中高亮显示对应的约束位置；
- Edit Constraint：编辑对应约束信息的功能。

注！

如当前 Primitive 或 Port 无位置约束，则高亮显示功能不可用，如图 3-30 所示。

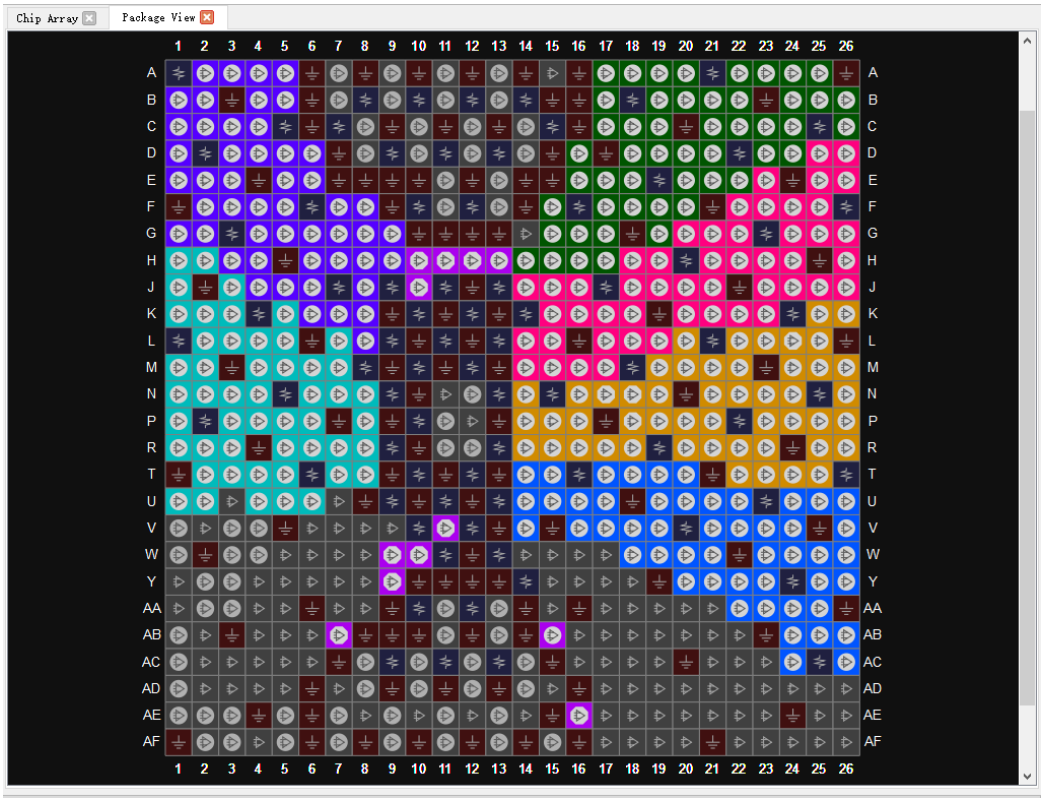
图 3-30 Netlist 右键功能



3.3.3 Package View 窗口

以 GW5AT-138-FCPBGA676A 为例，Package View 窗口如图 3-31 所示，该窗口以器件 package 信息为基础显示器件的封装信息，显示用户 I/O、电源、地等管脚。将鼠标放置某个位置上时，会悬浮显示该位置的 I/O 信息，包括 I/O 的 Type、Bank 以及 LVDS 信息等。

图 3-31 GW5AT-138-FPBGA676A Package view 窗口



用户 I/O、电源、地管脚使用不同的符号和颜色来区分。不同 BNAK 的 IO 引脚的颜色不同，图中管脚符号定义如下所示：

- “ ”表示用户 I/O；
- “ ”表示 VCCIO；
- “ ”表示 VSS。

Package View 支持右键菜单如图 3-32 所示，相关功能如下：

- Zoom In：放大 Package View 视图；
- Zoom Out：缩小 Package View 视图；
- Zoom Fit：按照窗口大小缩放 Package View 视图；
- Show Differential IO Pairs：显示差分对，如图 3-33 所示，红线相连的为一对差分对；
- Top View：Package View 以顶部视图进行显示，默认以顶部视图进行显示。如图 3-34 所示为 GW5AT-138-FCPBGA676A 封装的顶部视图，坐标原点在左上角；

- Bottom View：Package View 以底部视图进行显示。如图 3-35 所示为 GW5AT-138-FCPBGA676 封装的底部视图，坐标原点在右上角。

图 3-32 Package View 右键功能

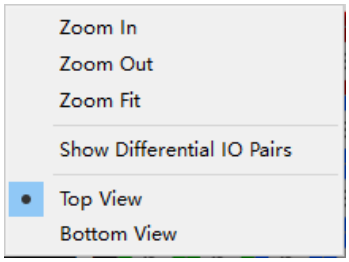
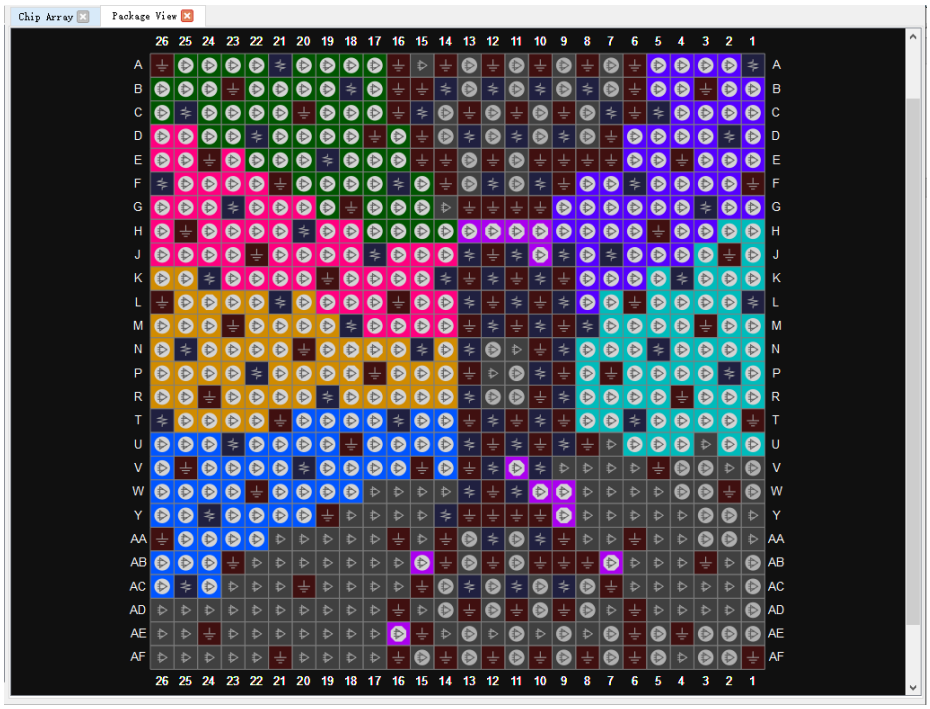


图 3-33 差分对显示

图 3-34 Top View

图 3-35 Bottom View



Package View 支持 IO Port 约束位置的显示，可以通过从 Netlist 窗口或底部 I/O Constraints 窗口中将 IO Port 拖拽到 Package View 窗口来约束 IO Port 的位置。拖拽时鼠标处会显示拖拽 Port 的名称，并且不可约束管脚呈现置灰不可拖拽状态。

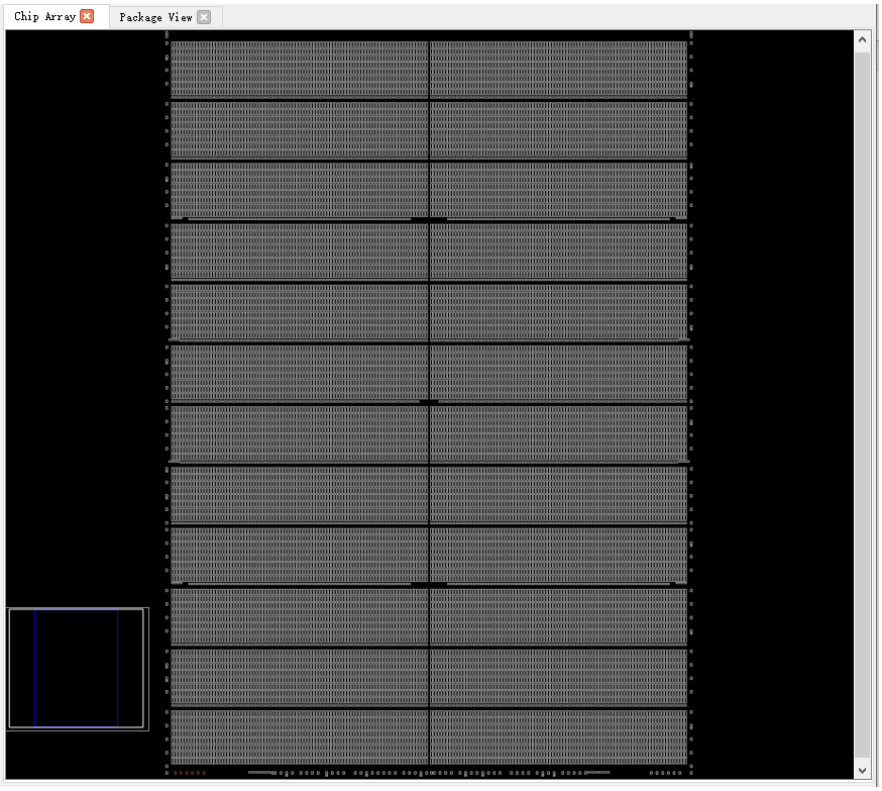
3.3.4 Chip Array 窗口

FloorPlanner 的 Chip Array 窗口如图 3-36 所示，Chip Array 窗口根据芯片的行列信息显示芯片的 I/O、CFU、DSP、PLL、BSRAM、DQS 等资源的分布，实现对所有约束位置的实时显示，支持放大、缩小、复制位置、悬停显示、拖拽等功能。

其中，I/O 是器件裸片的所有 I/O 位置，且会以不同颜色区分 I/O：

- 白色：该封装中封装出的 I/O 位置；
- 红色：该封装中未封装的 I/O 位置。

图 3-36 Chip Array 窗口



Chip Array 分为网格模式、宏单元模式、原语模式三种显示模式。

- 网格模式：以 GRID 为单位宏观显示约束位置，如图 3-37 所示；
- 宏单元模式：以 CLS、IOBLK 等为单位显示约束位置，如图 3-38 所示；
- 原语模式：以 REG、LUT 等为单位显示约束位置，如图 3-39 所示。

图 3-37 网格模式约束

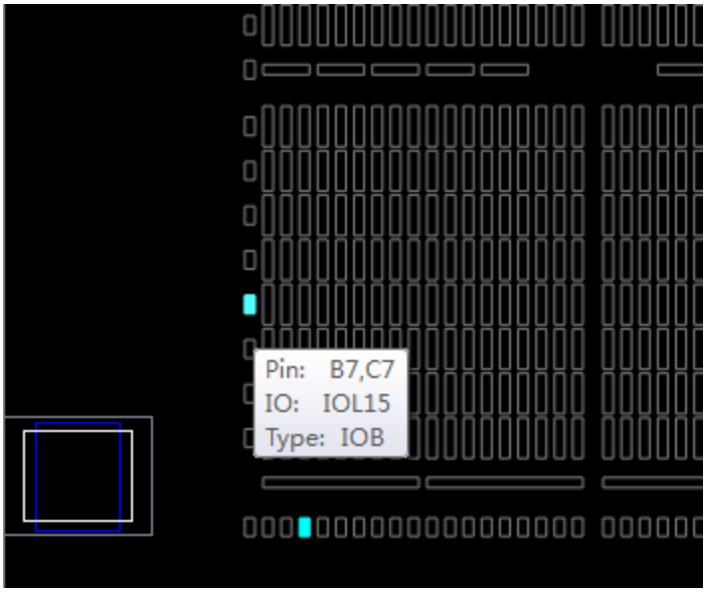


图 3-38 宏单元模式约束

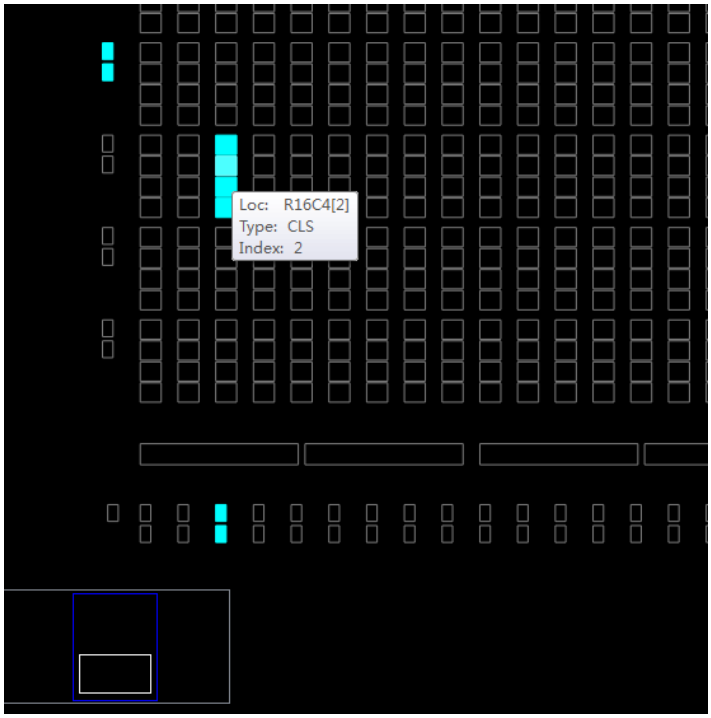
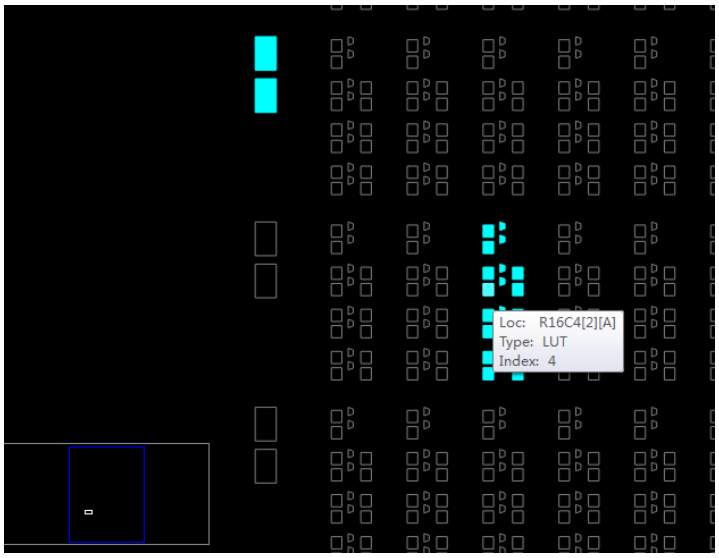


图 3-39 原语模式约束



Chip Array 支持以下拖拽功能：

- 从 Netlist 窗口拖拽到 Array 窗口，用于产生约束、指定约束位置；
- 从约束编辑窗口拖拽到 Array 窗口，用于指定约束位置。

Chip Array 窗口内置 chip 子窗口，用于实时显示当前视窗相对于整个器件的位置，拖动 chip 子窗口中的白色框，Chip Array 的视窗将跟随移动。同时，Chip Array 窗口采用不同颜色区分约束类型、显示约束位置，各颜色的含义分别介绍如下：

- 白色：用于显示处于选择状态或正在高亮显示的约束位置；
- 深蓝色：用于显示预留约束的位置信息，表示该位置不能再被占用；
- 浅蓝色：用于显示 I/O 和 Primitive 约束在某个 GRID 或 range 内，界面上显示为浅蓝色。

Chip Array 窗口支持右键菜单，具有如下功能：

- Zoom In：放大 Chip Array 视图；
- Zoom Out：缩小 Chip Array 视图；
- Zoom Fit：按照窗口大小缩放 Chip Array 视图；
- Show Constraints View：显示 Chip Array 的 instance 约束视图；
- Show Place View：显示 Chip Array 的 instance 布局视图，只有在工程运行完 Place & Route 后启动 FloorPlanner 才有效，否则置灰；
- Show Multi-View：同时显示 Chip Array 的 instance 约束和布局的复合视图，只有在工程运行完 Place & Route 后启动 FloorPlanner 才有效，否则置灰；

- Show In-Out Connection：在 Place View 中显示和选中 instance 输入输出连接的 instance 位置，只有在 Chip Array 窗口是 Show Place View > All Instance 视图中选中某个 instance 才能够使用，否则置灰；
- Show In Connection：在 Place View 中显示和选中 instance 输入连接的 instance 位置，只有在 Chip Array 窗口是 Show Place View > All Instance 视图中选中某个 instance 才能够使用，否则置灰；
- Show Out Connection：在 Place View 中显示和选中 instance 输出连接的 instance 位置，只有在 Chip Array 窗口是 Show Place View > All Instance 视图中选中某个 instance 才能够使用，否则置灰；
- Unhighlight All：对于选中位置或者区域消除高亮；
- Copy Location：复制选中的位置或者区域，若 Chip Array 窗口中有 GRID、Block 等处于选中状态，则右键菜单中的“Copy Location”功能可用，否则功能不可用，如图 3-40 所示。

在 Show Place View 中，可对 Lut、Reg 的密度进行显示，如图 3-41 所示，详情如下：

- ALL Instance：显示所有 Instance 的 place 情况，5 个以内呈淡绿色、6-10 个呈绿色、10 个以上呈深绿色；
- Only Lut：只显示所有 Lut 的 place 情况，2 个以内呈淡绿色、3-4 个呈绿色、4 个以上呈深绿色；
- Only Dff：只显示所有 Reg 的 place 情况，2 个以内呈淡绿色、3-4 个呈绿色、4 个以上呈深绿色。

在 Show Place View > ALL Instance 可以查看设计中所有 instance 的布局情况：

- 在 Chip Array 窗口中鼠标悬浮至 instance 布局位置，可显示该位置具体布局的 instance 名称，如图 3-42 所示；
- 在 Netlist 窗口中选中某个具体的 instance 右键选择 Highlight，该 instance 的布局位置会在 Chip Array 窗口中高亮显示，如图 3-43 所示。

注！

通过“Ctrl”键+鼠标左键拖动，可选取区域，右击选择 Copy Location，可复制所选区域的位置信息，复制的位置可直接粘贴到任意约束编辑窗口中。

图 3-40 Chip Array 右键功能

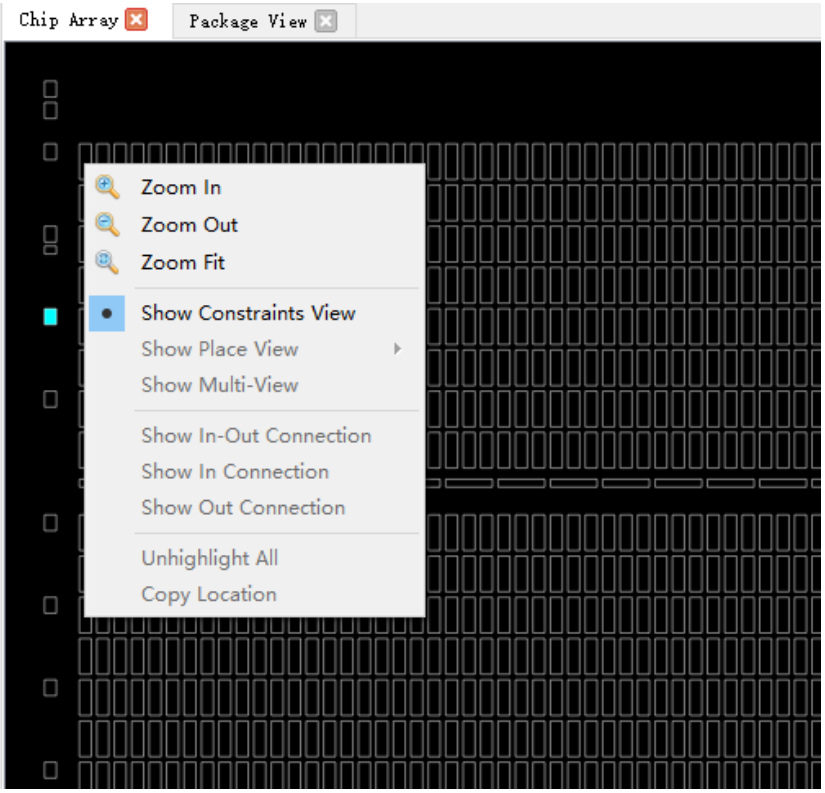


图 3-41 Show Place View 显示

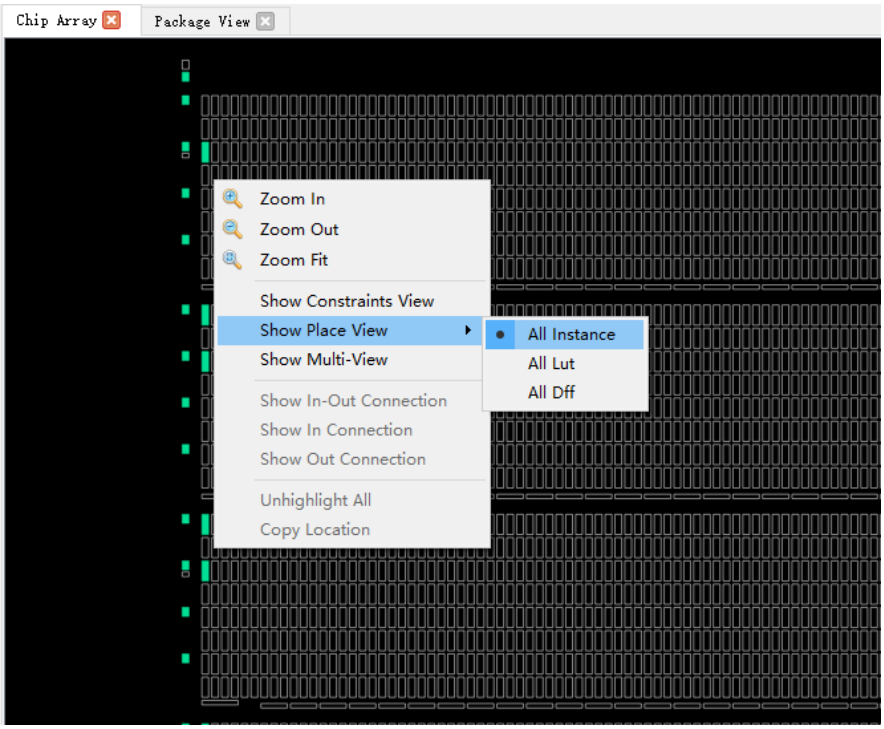


图 3-42 鼠标悬浮显示

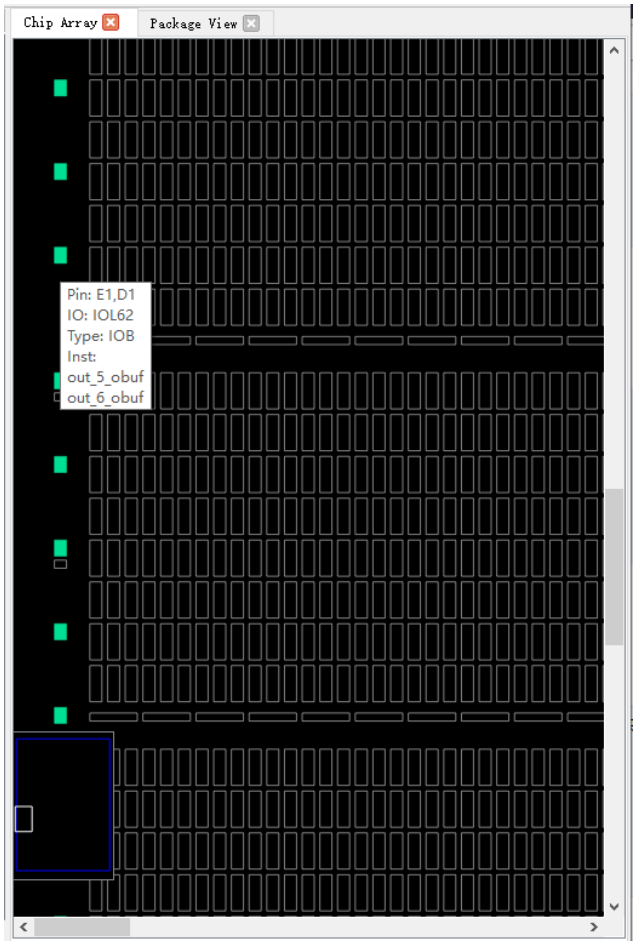
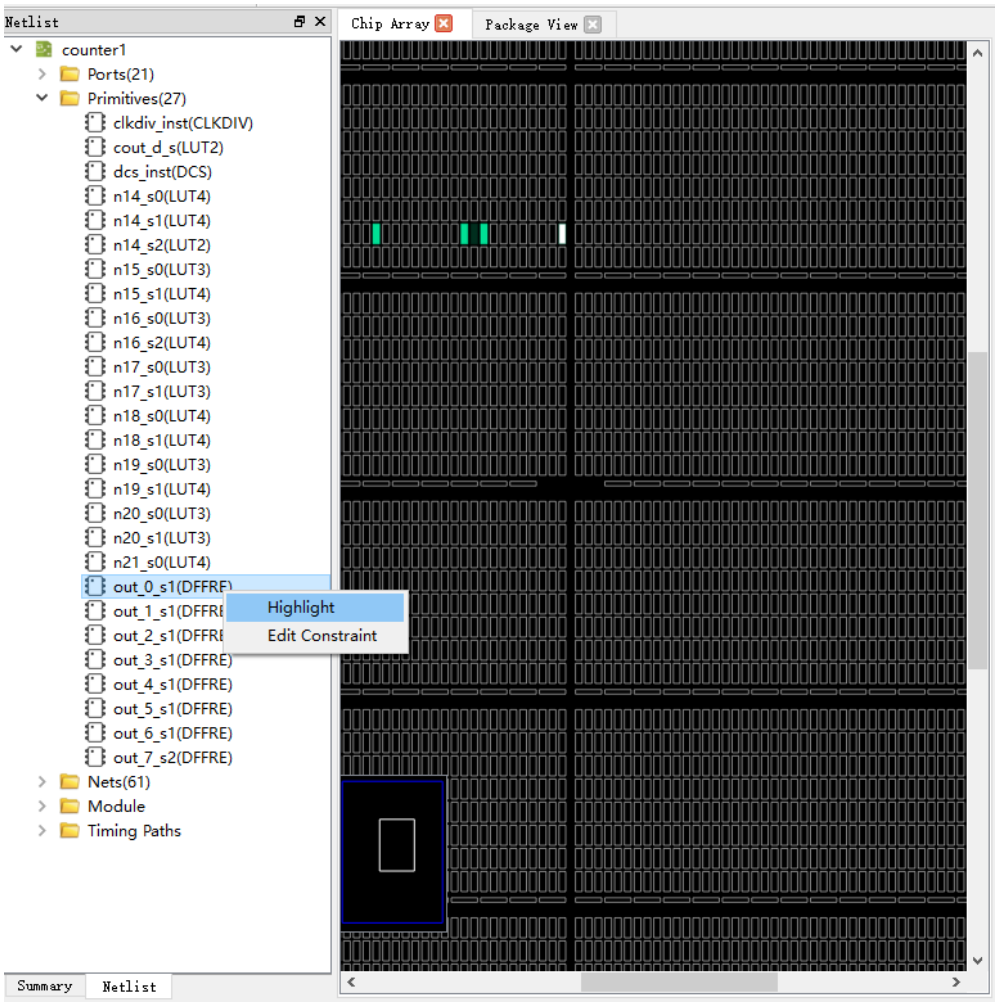
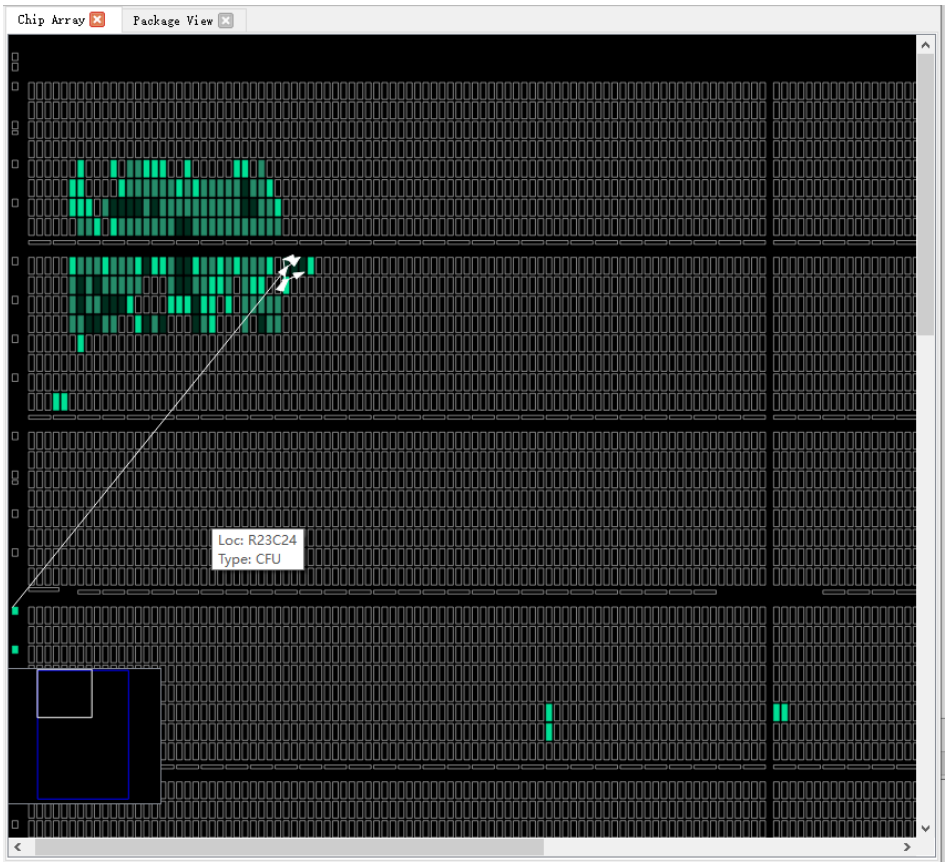

Chip Array
Package View
Pin: E1,D1
IO: IOL62
Type: IOB
Inst:
out_5_obuf
out_6_obuf

图 3-43 右键高亮



另外，Chip Array 窗口还可对时序路径进行高亮显示，如图 3-44 所示。

图 3-44 时序路径高亮显示



3.3.5 Constraint 编辑窗口

Constraint 编辑窗口包含“I/O Constraints”、“Primitive Constraints”、“Group Constraints”等 8 个编辑窗口，用于显示各约束的详细信息，并提供约束编辑功能和位置拖拽功能，各窗口分别介绍如下：

I/O Constraints

I/O Constraints 是对设计的 port 进行管脚约束。I/O 约束窗口如图 3-45 所示，各功能如下：

- 显示用户设计中所有 IO Port 的属性及约束信息，如 Port 的 Direction、Bank、IO Type、Pull Mode 等；
- 提供约束位置、属性等编辑功能；
- 可通过拖拽、双击等方式改变约束信息。

注！

- I/O 的位置可以通过拖拽的方式进行设置，也可以双击输入；
- 在拖拽 IO 过程中会显示所拖拽的 IO 名称；
- 将 IO 拖拽至 Chip Array 窗口中时，可放置的位置变亮，不可放置的位置颜色亮度不变；

- 将 IO 拖拽至 Package View 窗口中时，可放置位置亮度不变，不可放置位置变暗；
- 设置完成后，在 Chip Array 窗口中约束的位置变为浅蓝色高亮，在 Package View 窗口中约束的位置变为橙色高亮。

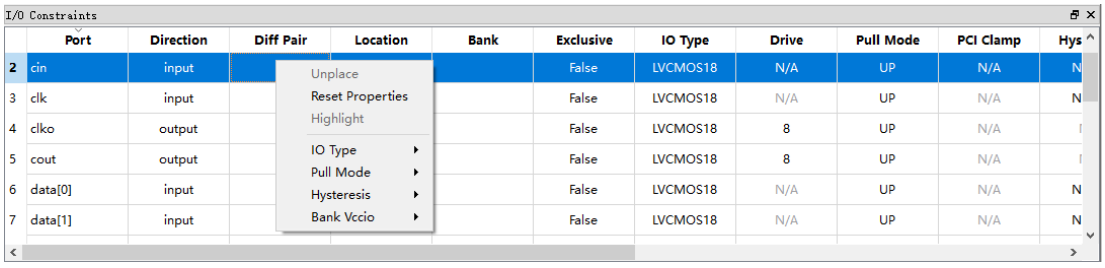
窗口提供右键菜单功能，详情如下：

- Unplace：取消放置；
- Reset Properities：复位 Port 属性设置；
- Highlight：高亮显示约束位置；
- IO Type：设置电平标准；
- Drive：设置驱动电流；
- Pull Mode：设置上拉模式；
- PCI Clamp：设置 PCI 协议的开关；
- Hysteresis：设置迟滞量；
- Open Drain：设置开漏电路的开关；
- Vref：设置外部参考电压；
- Single Resistor：设置单端电阻的开关；
- Diff Resistor：设置差分电阻的开关；
- Bank Vccio：设置 BANK 电压。

注！

右键菜单支持用户批量修改 Port 属性的功能，用户可选择多个 Port，若多个 Port 有相同的属性值可配置，则通过右键菜单可统一进行配置；详细属性设置标准请参考 *DS981，GW5AT 系列 FPGA 产品数据手册*。

图 3-45 I/O 约束窗口



Primitive Constraints

Primitive Constraints 是约束设计中原语的位置，原语约束窗口如图 3-46 所示，功能如下：

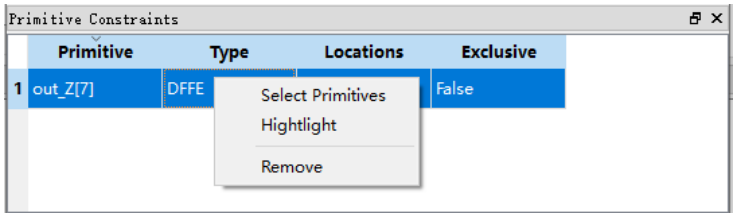
- 用于显示当前所有 Primitive 约束的名称、类型、位置以及 Exclusive 信息；

- 提供编辑功能，该窗口提供右键菜单功能，用于提供高亮显示约束位置、删除和添加约束的功能。

注！

- 可通过拖拽的方式或双击输入的方式修改位置信息；
- 可通过双击设置 Exclusive 属性；
- 在 Primitive 约束位置进行手动输入时，会对位置进行语法检查及合法性检查，错误提示对话框如图 3-14 和图 3-15 所示。

图 3-46 原语约束窗口

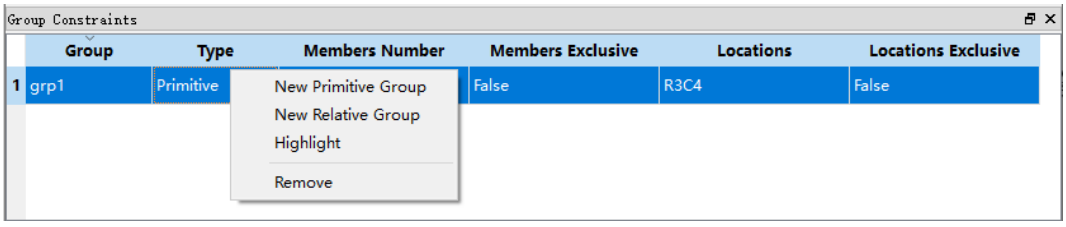


Group Constraints

Group Constraints 是对设计中的 I/O 和部分原语进行组约束，组约束窗口如图 3-47 所示，功能如下：

- 用于显示当前所有组约束的名称、类型、包含的 Primitive 个数、位置以及 Exclusive 信息，包含 Primitive 和 Relative 两种 Group 的显示；如图 3-13 和图 3-17 所示，双击对应的 Group 条目，打开对话框，可实现约束信息的编辑修改功能；
- 该窗口提供右键菜单功能，用于提供高亮显示约束位置、删除和添加约束的功能。

图 3-47 组约束窗口



Resource Reservation

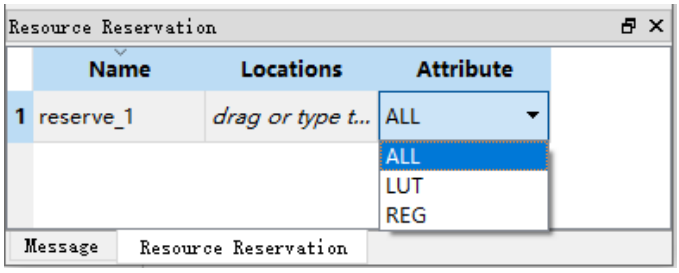
Resource Reservation 是对当前封装中的可用资源进行预留约束，预留约束窗口如图 3-48 所示，功能如下：

- 用于显示当前所有预留约束的位置信息；
- 该窗口提供右键菜单功能，用于提供高亮显示约束位置、删除、添加约束的功能；
- Name 属性用于区分各资源预留约束，用户不能进行修改。

注！

可通过拖拽或双击输入修改位置信息。

图 3-48 预留约束窗口



Clock Net Constraints

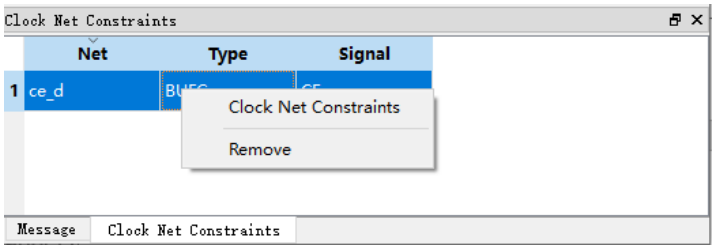
Clock Net Constraints 是对设计中的 net 进行全局时钟分配约束，时钟分配约束窗口如图 3-49 所示，功能如下：

- 用于显示当前所有时钟分配约束的相关信息；
- 该窗口提供右键菜单功能，用于提供添加、删除时钟分配约束的功能。

注！

- 双击约束可进行编辑修改；
- CLOCK Net 约束无位置信息，不支持拖拽功能；
- 新建全局时钟约束的窗口如图 3-19 所示。

图 3-49 时钟约束窗口



GCLK Primitive Constraints

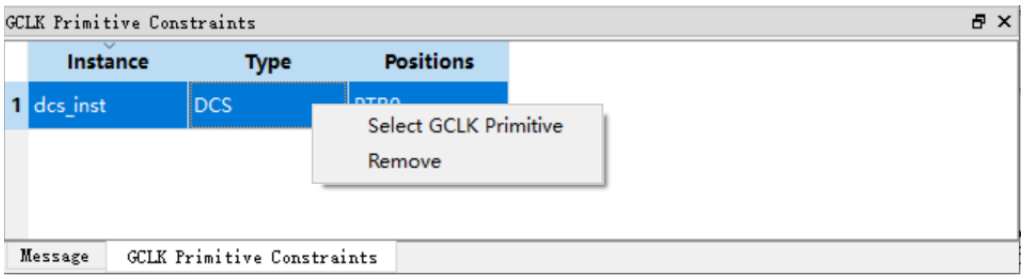
GCLK Primitive Constraints 是对全局时钟原语进行约束，全局时钟约束窗口如图 3-50 所示，功能如下：

- 用于显示所有的全局时钟约束，包括 Instance 名称、类型以及象限位置；
- 窗口支持右键菜单功能，用于添加新的全局时钟约束和删除已有约束。

注！

新建全局时钟约束的窗口如图 3-20 所示。

图 3-50 全局时钟约束窗口



HCLK Primitive Constraints

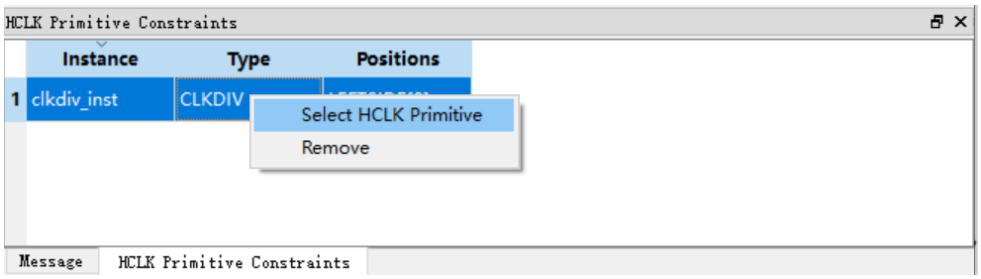
HCLK Primitive Constraints 是对设计中的高速时钟原语进行高速时钟约束，高速时钟约束窗口如图 3-51 所示，功能如下：

- 用于显示针对高速时钟相关的 Instance 的位置约束，包括 Instance 名称、类型以及高速时钟位置；
- 窗口支持右键菜单功能，用于添加新的高速时钟约束和删除已有约束。

注！

新建全局时钟约束的窗口如图 3-21 所示。

图 3-51 高速时钟约束窗口



Vref Constraints

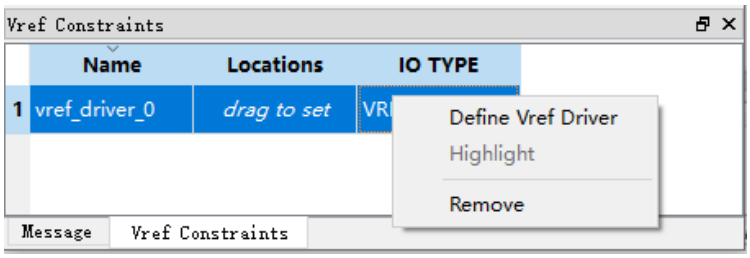
Vref Constrains 是约束所在 bank 的外部参考电压，Vref 约束窗口如图 3-52 所示，功能如下：

- 用于显示用户自定义的 Vref Driver 信息，用户可自定义 Vref 的名称、位置信息；
- 窗口支持右键菜单功能，用于高亮显示约束位置、添加、删除约束信息。

注！

位置信息只能通过拖拽的方式进行设置。

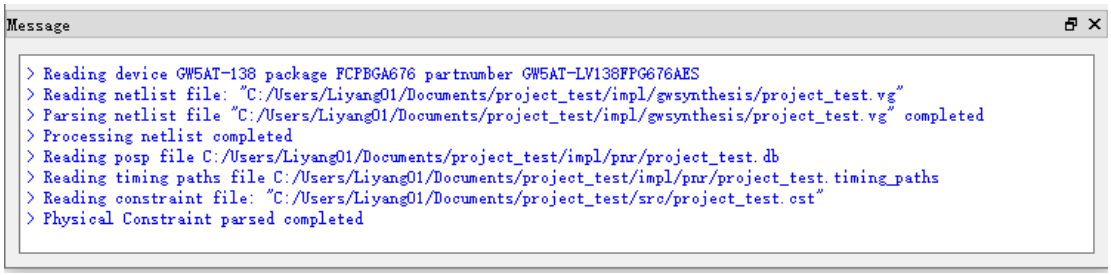
图 3-52 Vref 约束窗口



3.3.6 Message 窗口

Message 窗口如图 3-53 所示，窗口提供输出结果的显示。

图 3-53 Message 窗口



4 FloorPlanner 使用

FloorPlanner 可以新建和编辑约束，生成供布局布线流程使用的物理约束文件。

4.1 新建约束文件

FloorPlanner 可输出新建的物理约束文件，亦可输出修改后的物理约束文件，操作步骤如下所示：

1. 根据 3.2 启动所述，启动 FloorPlanner；
2. 单击“File > New”，打开“New”对话框，如图 4-1 所示；

   注！

   亦可通过以下两种方式打开“New”对话框：

   - 使用快捷键 Ctrl + N；
   - 单击工具栏上的“New”图标。
3. 输入工程的网表文件，选择器件类型，单击“OK”。

图 4-1 新建约束文件

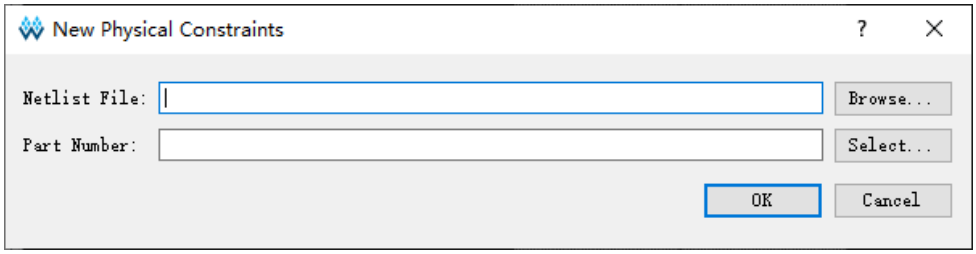
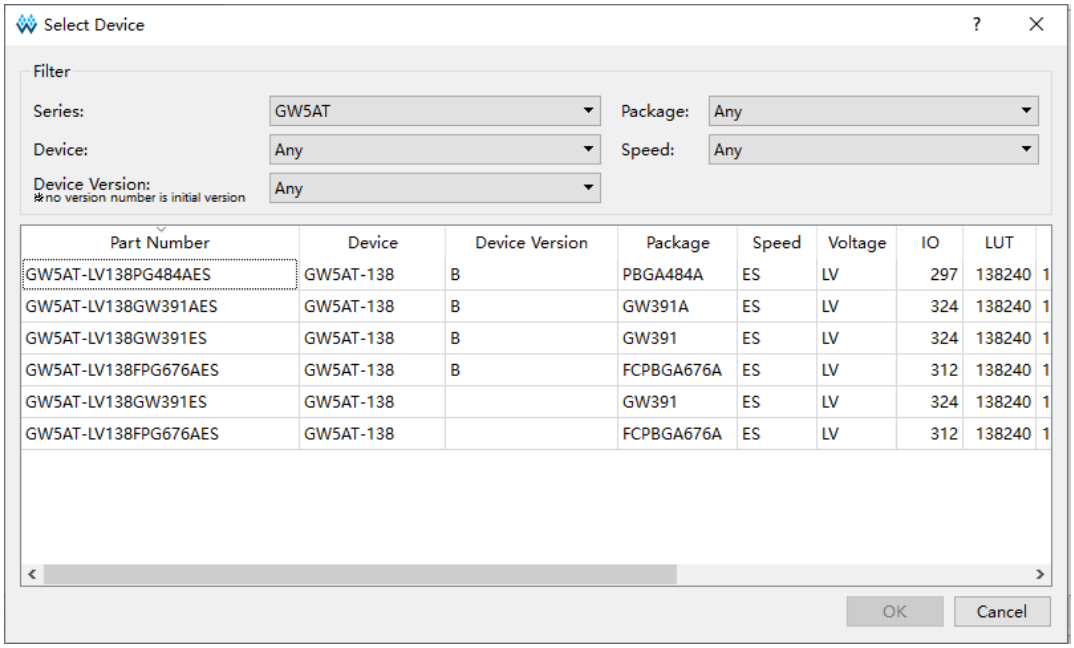


图 4-2 选择器件



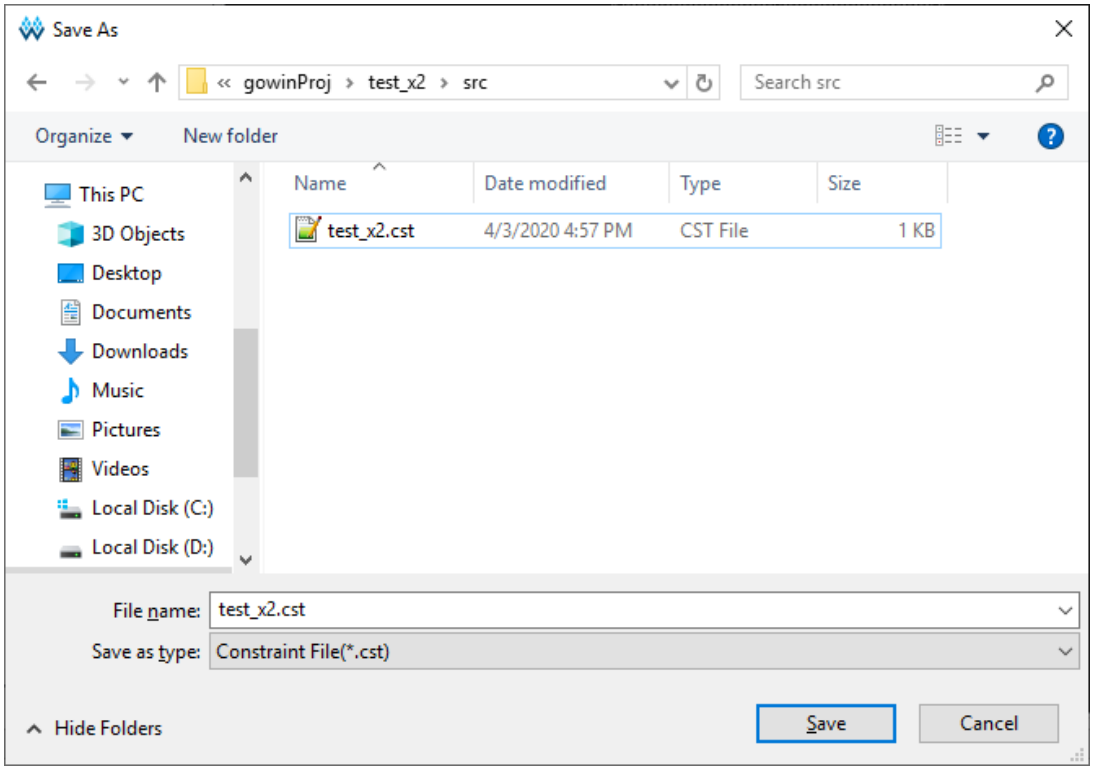
注！

- Select …按钮用于选取器件、封装信息，支持高云半导体所有的 FPGA 器件，如图 4-2 所示；
- 启动 FloorPlanner 采用 3.2 启动介绍中的第三种方式。

新建物理约束，在 FloorPlanner 主界面中可进行如下操作：

1. 用户通过拖拽等方式分配管脚位置；
2. 单击工具栏中的“Save”图标，即可输出约束文件；
3. 在弹出的“Save”对话框中，可修改文件名，如图 4-3 所示。

图 4-3 保存输出文件



4.2 编辑约束文件

FloorPlanner 支持 I/O 约束、原语约束、组约束、资源预留约束、全局时钟分配约束、参考电压约束等的创建。可通过 Constraints 菜单编辑生成约束，详情请参考 3.3.1 菜单栏。

注！

亦可通过其他方式创建约束，本节主要以拖拽的方式为例，介绍如何通过拖拽编辑生成约束。

4.2.1 编辑约束示例

以用户设计 counter.v 为例，演示如何编辑各类型约束：

```
module counter1(out, cout, data, load, cin, clk, clko);
output [7:0] out;
output cout;
output clko;
input [7:0] data;
input load, cin, clk;
```

```
reg [7:0] out;
always @(posedge clk)
begin
    if (load)
        out = data;
    else
        out = out + cin;
end
assign cout = &out & cin;
wire clkout;
CLKDIV clkdiv_inst (
    .CLKOUT(clkout),
    .HCLKIN(clk),
    .RESETN(1'b1),
    .CALIB(1'b0)
);
defparam clkdiv_inst.DIV_MODE = "2";

DCS dcs_inst (
    .CLKOUT(clko),
    .CLKSEL(4'b0000),
    .CLKIN0(clkout),
    .CLKIN1(clkout),
    .CLKIN2(clkout),
    .CLKIN3(clkout),
    .SELFORCE(1'b0)
);
```

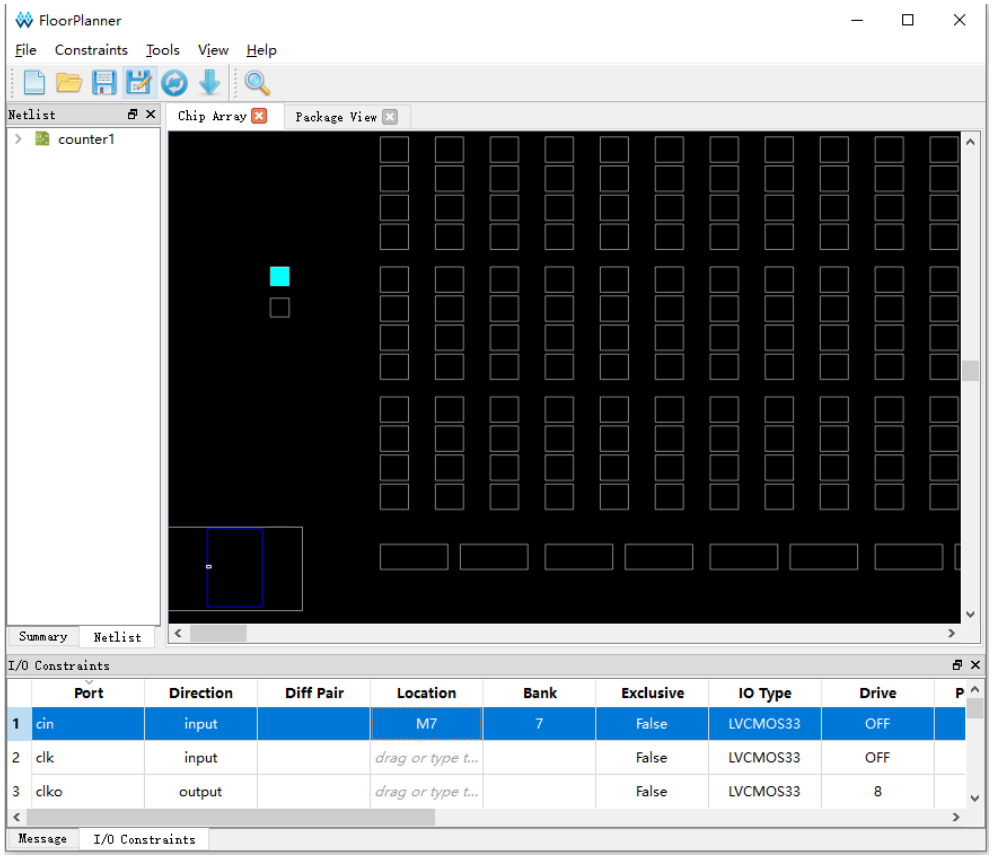
```
defparam dcs_inst.DCS_MODE = "RISING";

endmodule
```

4.2.2 编辑 I/O 约束

拖至 Chip Array 创建 I/O Constraints，步骤如下：

1. 单击 IO Constraints 编辑窗口，放大 Chip Array 视图至宏单元模式；
2. 选中 Port“cin”拖拽至 Chip Array 窗口中的 M7 位置，如图 4-4 所示；
3. Port“cin”的 Location 信息显示为 M7。

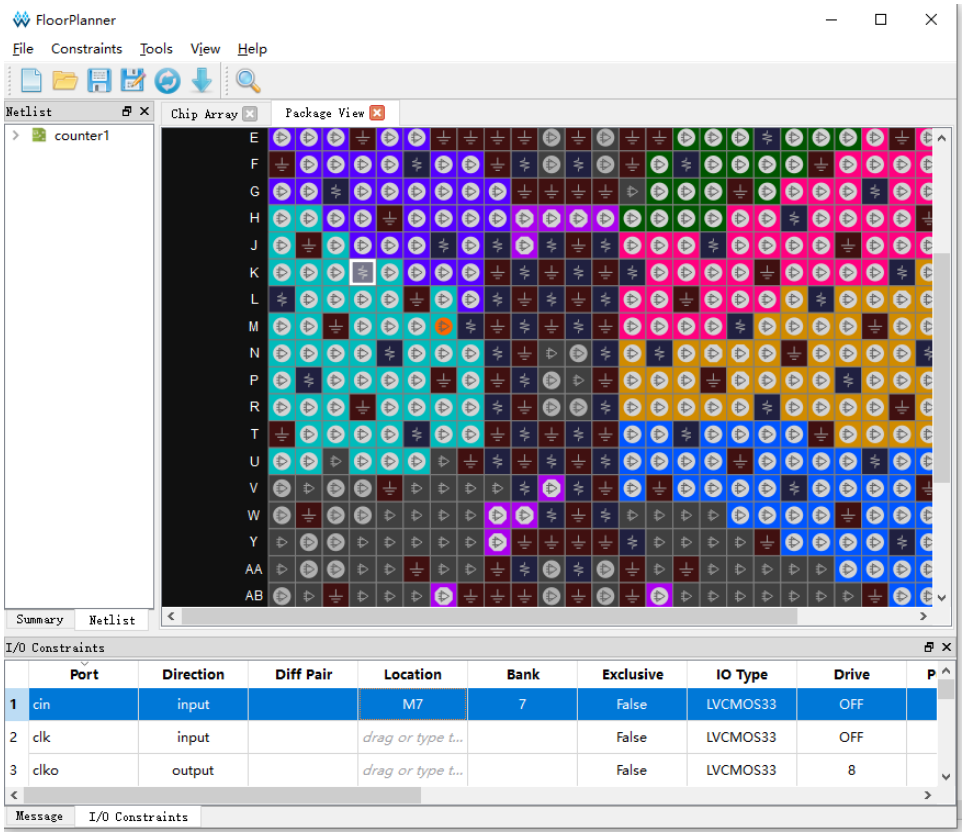
图 4-4 拖拽到 Chip Array 创建 I/O Constraints



拖至 Package View 创建 I/O Constraints，步骤如下：

1. 单击至 IO Constraints 编辑窗口；
2. 选中 Port“cin”拖拽至 Package View 窗口中的 M7 位置，如图 4-5 所示；
3. Port“cin”的 Location 信息显示为 M7。

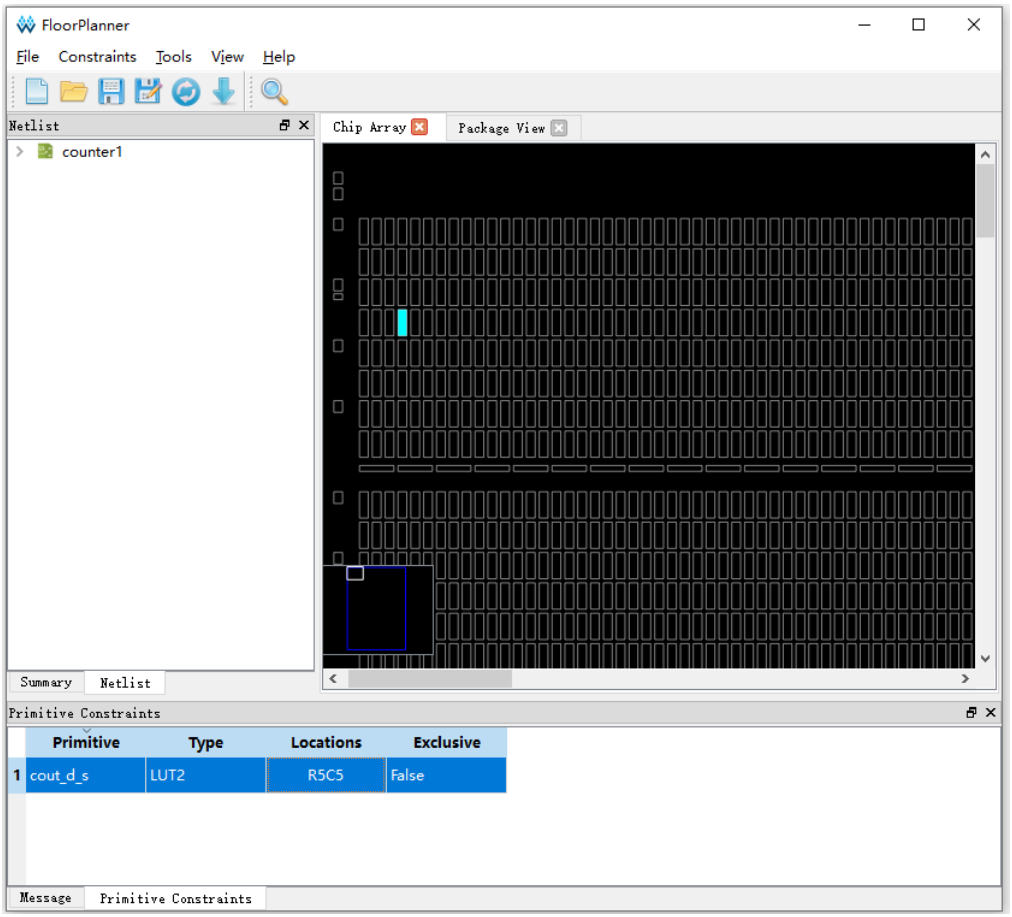
图 4-5 拖至 Package View 创建 I/O Constraints



4.2.3 编辑原语约束

1. 在 Primitive Constraints 编辑窗口，右击选择“Select Primitives”，弹出“Primitive Finder”对话框后，选择 Primitive“cout_d_s”，单击“OK”；
2. 选中新创建的 Primitive 约束，拖拽至 Chip Array 窗口中的“R5C5”位置，如图 4-6 所示；
3. Primitive“cout_d_s”的 Location 信息显示为 R5C5。

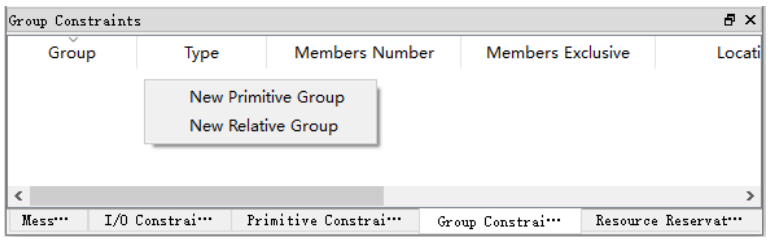
图 4-6 拖拽到 Chip Array 创建 Primitive Constraints



4.2.4 编辑组约束

如图 4-7 所示，在 Group Constraints 编辑器中，可通过右击选择菜单，创建 Primitive Group 和 Relative Group。

图 4-7 Group Constraints 编辑器右键菜单



编辑原语组约束

1. 在 Group Constraints 编辑窗口中，右击选择“New Primitive Group”，弹出“New Primitive Group”对话框；
2. 输入 Group Name“grp1”，单击“ ”弹出“Primitive Finder”对话框；
3. 选取所要设置的 Primitive“n17_s0”、“cout_d_s”，单击“OK”，加入 Members 列表；

4. 在 Locations 输入所要约束的位置“R5C10”，如图 4-8 所示；
5. 在New Primitive Group对话框中单击“OK”，可创建一条Primitive Group Constraints，如图 4-9 所示。

图 4-8 创建 Primitive Group Constraints

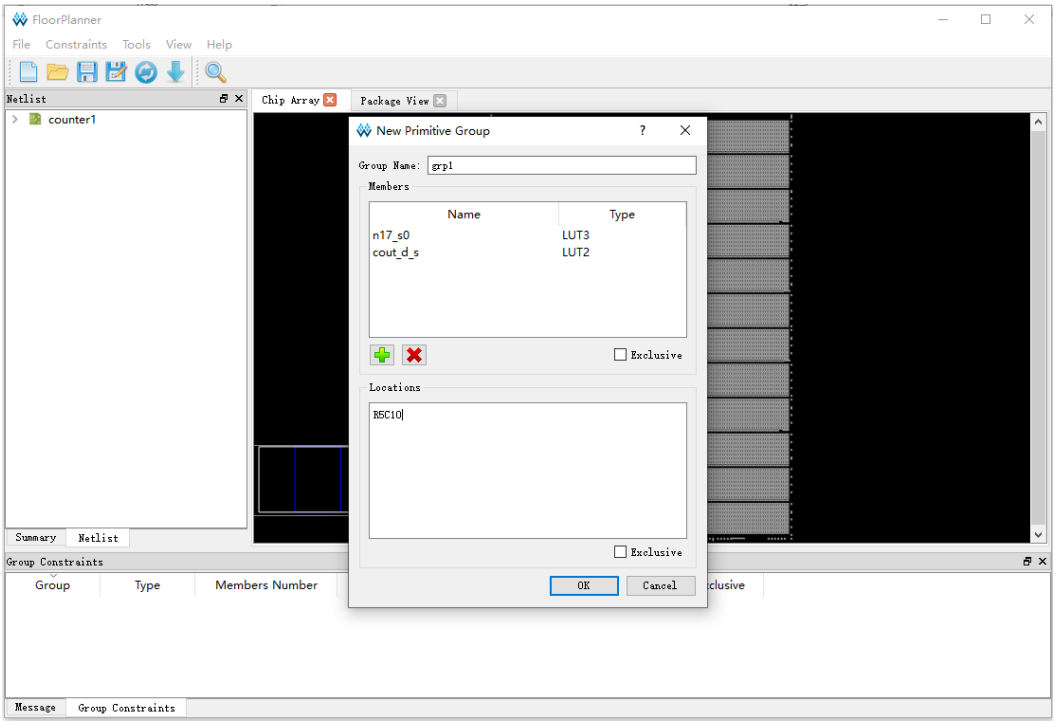
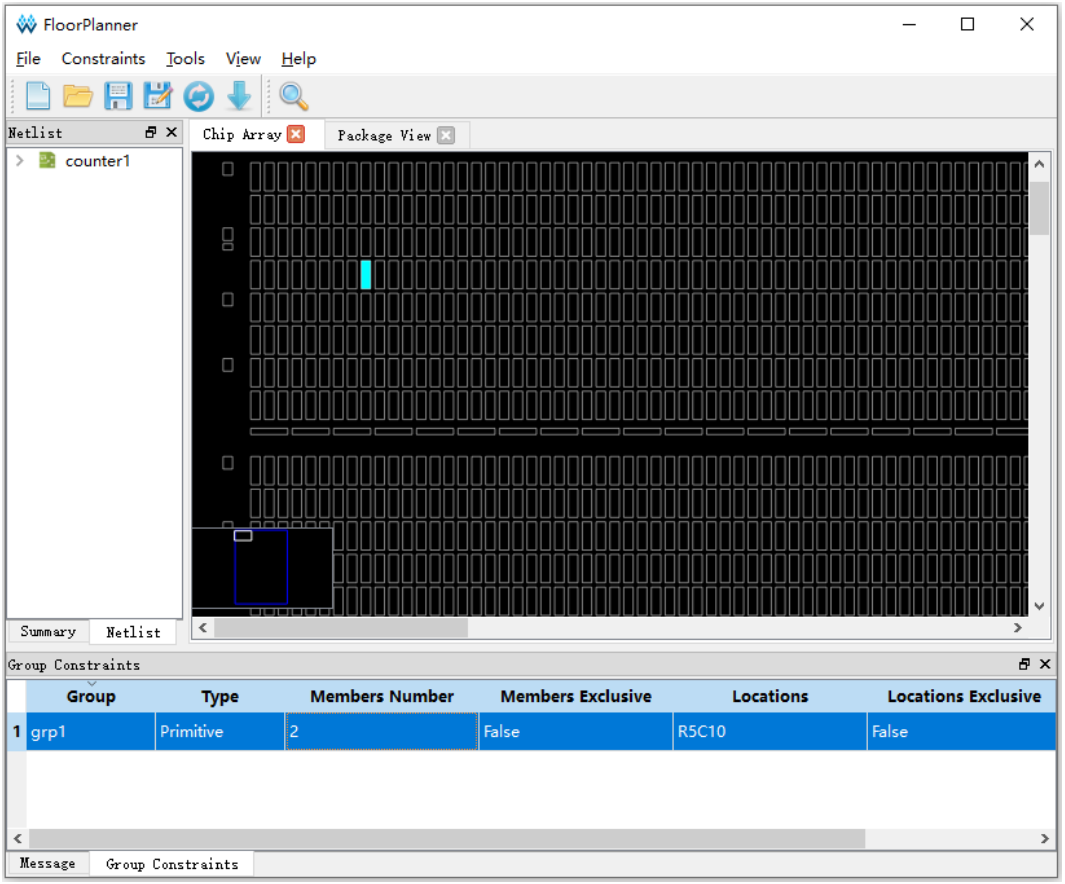


图 4-9 Primitive Group Constraints



注！

Primitive Group Constraints 的 Location 信息只能通过手动输入或者从 Chip Array 窗口中复制，不能通过拖拽实现。

编辑相对组约束

1. 在 Group Constraints 编辑窗口中，右击选择“New Relative Group”，弹出“New Relative Group”对话框；
2. 输入 Group 的名字“rel_grp”，单击“”，弹出 Primitive Finder 对话框；
3. 在 Primitive Finder 对话框中选择所要设置的 Primitives“cout_0_s1”、“cout_1_s1”，选择“OK”，添至“Member”列表中；
4. 为每个 Primitive 添加相对位置“R0C0”、“R4C5”，如图 4-10 所示；
5. 在 New Relative Group 对话框中，选择“OK”创建 Relative Group Constraints，如图 4-11 所示。

图 4-10 Relative Group Constraints 创建

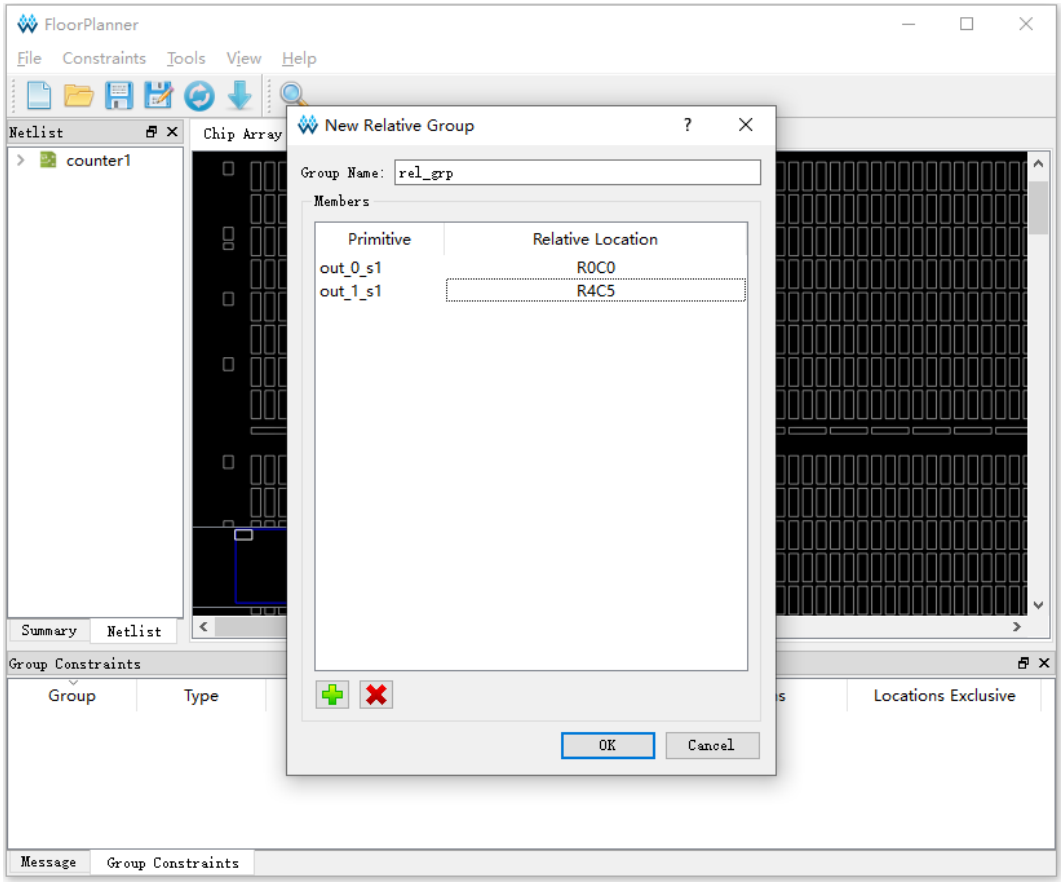
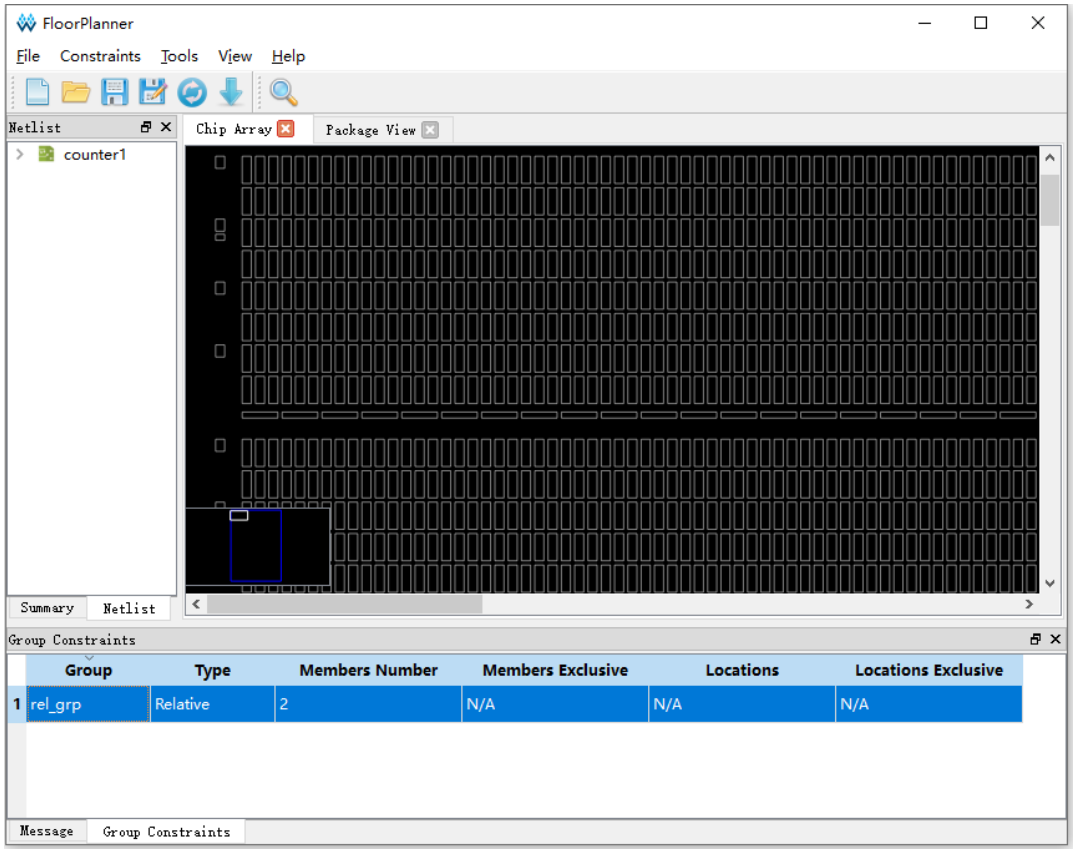


图 4-11 Relative Group Constraints



4.2.5 编辑资源预留约束

1. 在 Resource Reservation 编辑窗口中，进行右击选择“Reserve Resources”，在编辑器中添加 Resource Reservation 约束，如图 4-12 所示；
2. 选中新建的 Resource Reservation 约束拖拽到 Chip Array 窗口中想要进行预留约束的位置，图 4-13 所示拖拽至 BSRAM_R10[1]位置完成 Resource Reservation 约束。

图 4-12 创建 Resource Reservation 约束

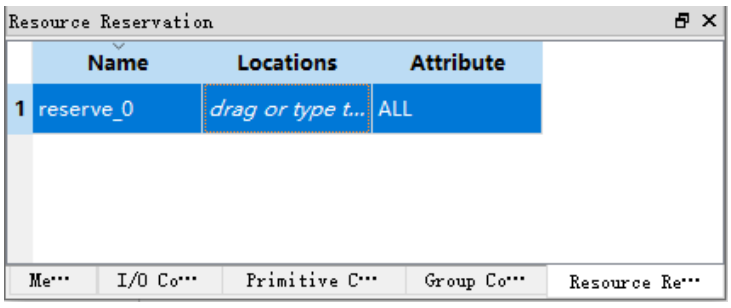
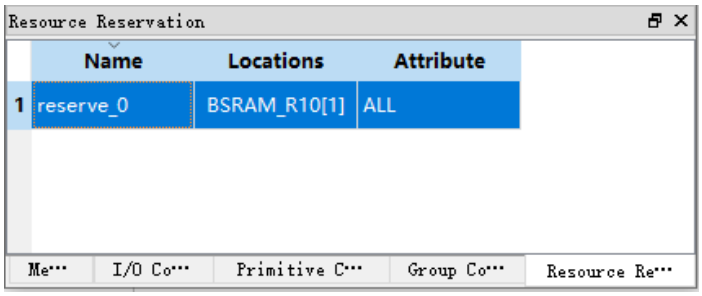


图 4-13 Resource Reservation



4.2.6 编辑全局时钟分配约束

1. 在 Clock Net Constraints 编辑窗口中，右击选择“Clock Net Constraints”，弹出“Clock Net Constraints”设置对话框；
2. 单击“ ”，弹出“Select Net”对话框，选择需要约束的 Net，在 Select Net 对话框中，单击“OK”添加 Net；
3. 设置时钟类型，在 Type 下拉列表中选择 Type 类型，设置 Signal 类型，如图 4-14 所示；
4. 设置完成后，单击“OK”，即会将该条约束添加至 Clock Net Constraints 编辑窗口中，如图 4-15 所示。

图 4-14 Clock Net Constraints 约束创建

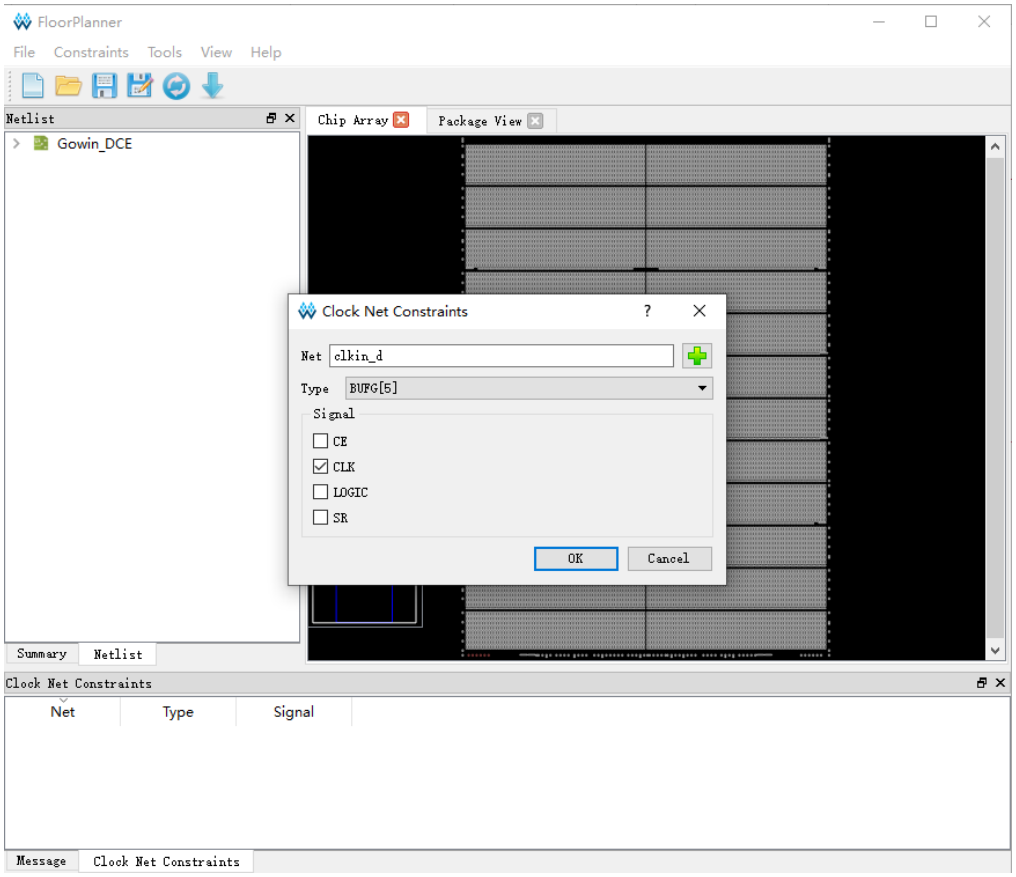
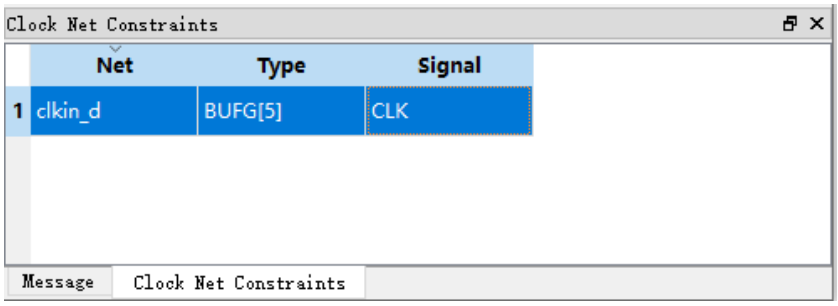


图 4-15 Clock Net Constraints 约束



4.2.7 编辑全局时钟约束

GCLK Primitive Constraints 仅支持对 DCS 和 DCE 的约束。

GCLK Primitive Constraints 的创建步骤如下：

1. 在 GCLK Primitive Constraints 编辑窗口中，进行右击选择“Select GCLK Primitive”操作，弹出“GCLK Primitive Constraints”对话框；

2. 单击“ ”，弹出 GCLK 选择对话框，选取 Instance，在 GCLK 选择对话框中，单击“OK”，Instance 完成设置；

3. 在 GCLK Primitive Constraints 对话框的“Position”下方选取所要约束的全局时钟位置，如图 4-16 所示；
4. 在 GCLK Primitive Constraints 对话框中单击“OK”，即会将该条约束添至 GCLK Primitive Constraints 编辑窗口，如图 4-17 所示。

图 4-16 GCLK Primitive Constraints 创建

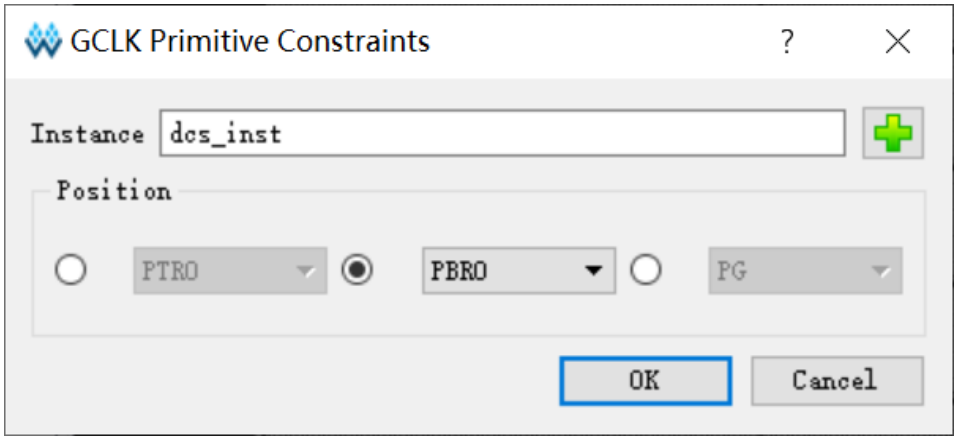


图 4-17 GCLK Primitive Constraints

4.2.8 编辑高速时钟约束

HCLK Primitive Constraints 仅对以下两种类型的 CLKDIV 和 DLLDLY Instance 进行约束。

HCLK Primitive Constraints 的创建步骤如下：

1. 在 HCLK Primitive Constraints 编辑窗口中，进行右击选择“Select HCLK Primitive”，弹出“HCLK Primitive Constraints”对话框；
2. 单击“ ”按钮，弹出 HCLK 对话框，选取 Instance，在 HCLK 对话框中，单击“OK”，设置 Instance；
3. 在 HCLK Primitive Constraints 对话框中“Position”下方选取所要约束的高速时钟位置，如图 4-18 所示；
4. 单击 HCLK Primitive Constraints 对话框的“OK”，即可将该约束添至 HCLK Primitive Constraints 编辑器中，如图 4-19 所示。

图 4-18 HCLK Primitive Constraints 创建

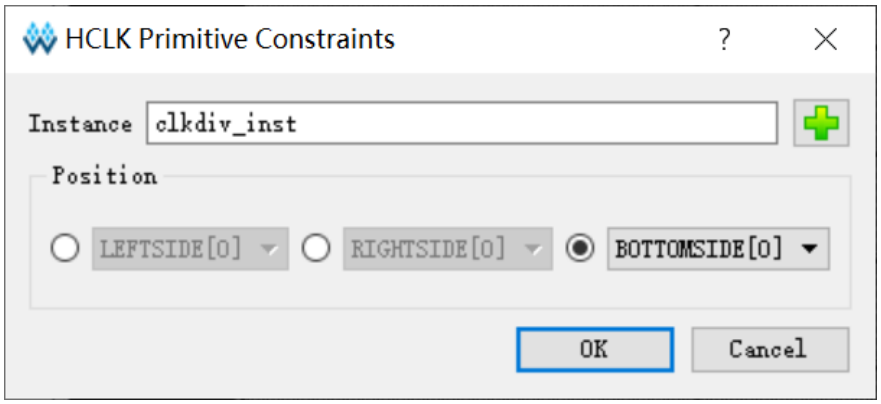
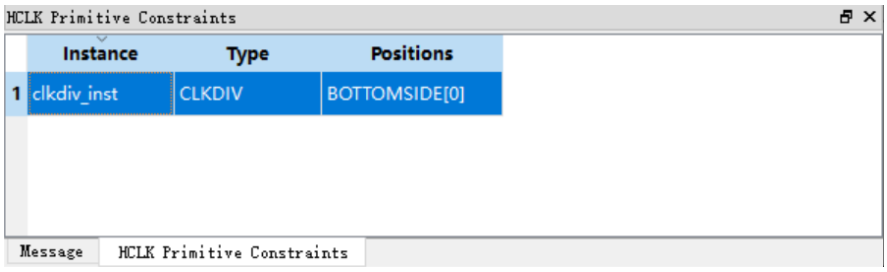


图 4-19 HCLK Primitive Constraints

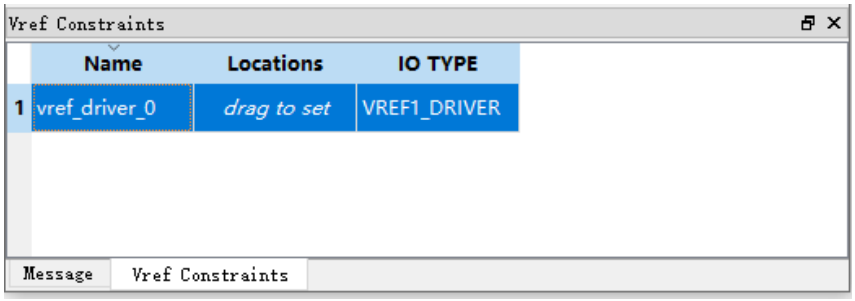


4.2.9 编辑参考电压约束

拖至 Chip Array 窗口创建 Vref Constraints，步骤如下：

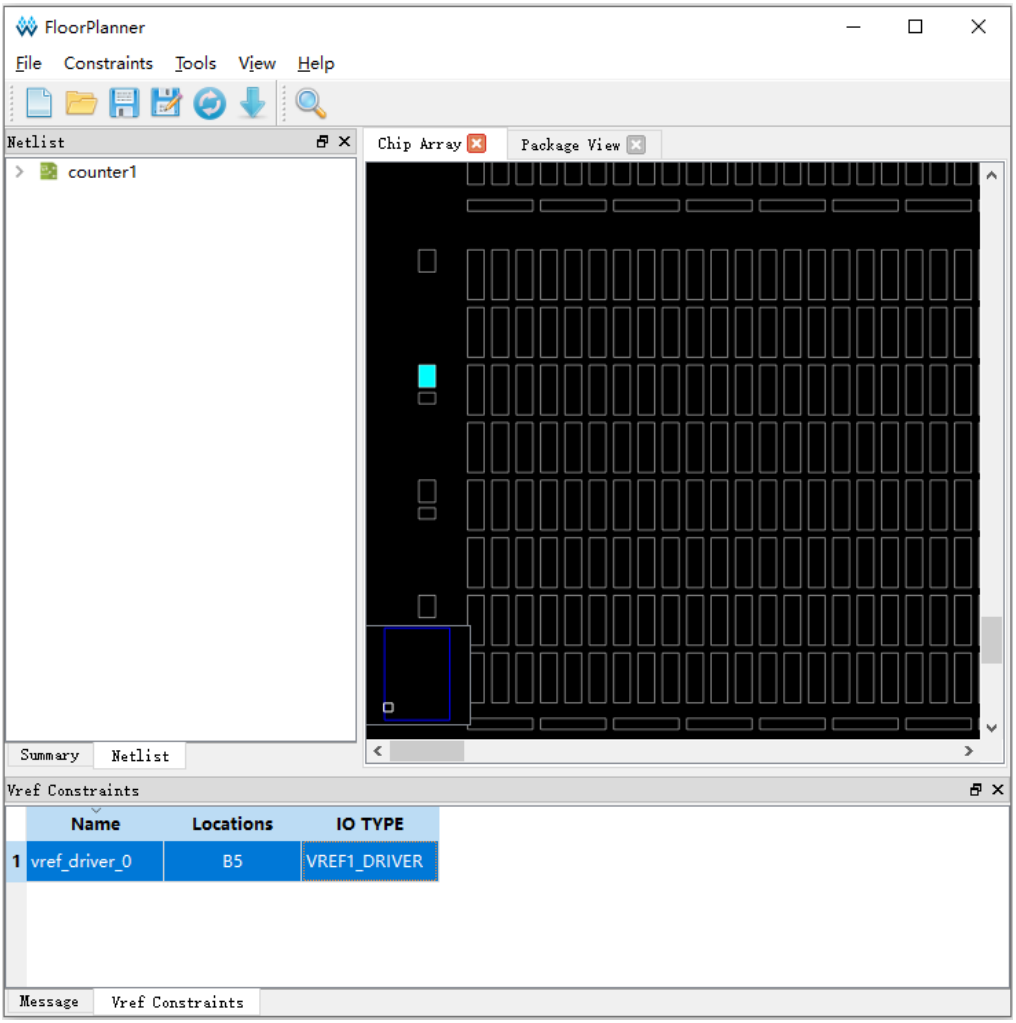
1. 在 Vref Constraints 编辑窗口中，进行右击选择“Define Vref Driver”，即可将该条 Vref Constraints 约束添至 Vref Constraints 编辑器中，如图 4-20 所示；
2. 放大 Chip Array 视图至宏单元模式，选中 Vref Constraints 编辑窗口中新创建的 Vref Constraints，拖拽至 Chip Array 窗口中的“B5”位置，Vref Constraints 的 Location 信息显示为“B5”，如图 4-21 所示。

图 4-20 Vref Constraints 创建



可自定义 Vref 约束名，Vref 名字不允许重名，设置期间，系统会进行检查，如名字重复，则提示用户，如图 4-23 所示。

图 4-21 拖拽至 Chip Array 窗口生成 Vref Constraints Location 信息



拖至 Package View 创建 Vref Constraints，步骤如下：

1. 在 Vref Constraints 编辑窗口中，进行右击选择“Define Vref Driver”，即可将该条 Vref Constraints 约束添至 Vref Constraints 编辑器中，如图 4-20 所示；
2. 选中 Vref Constraints 编辑窗口中新创建的 Vref Constraints，拖拽至 Package View 窗口中的“B5”位置，Vref Constraints 的 Location 信息显示为“B5”，如图 4-22 所示。

图 4-22 拖拽至 Package View 窗口生成 Vref Constraints location 信息

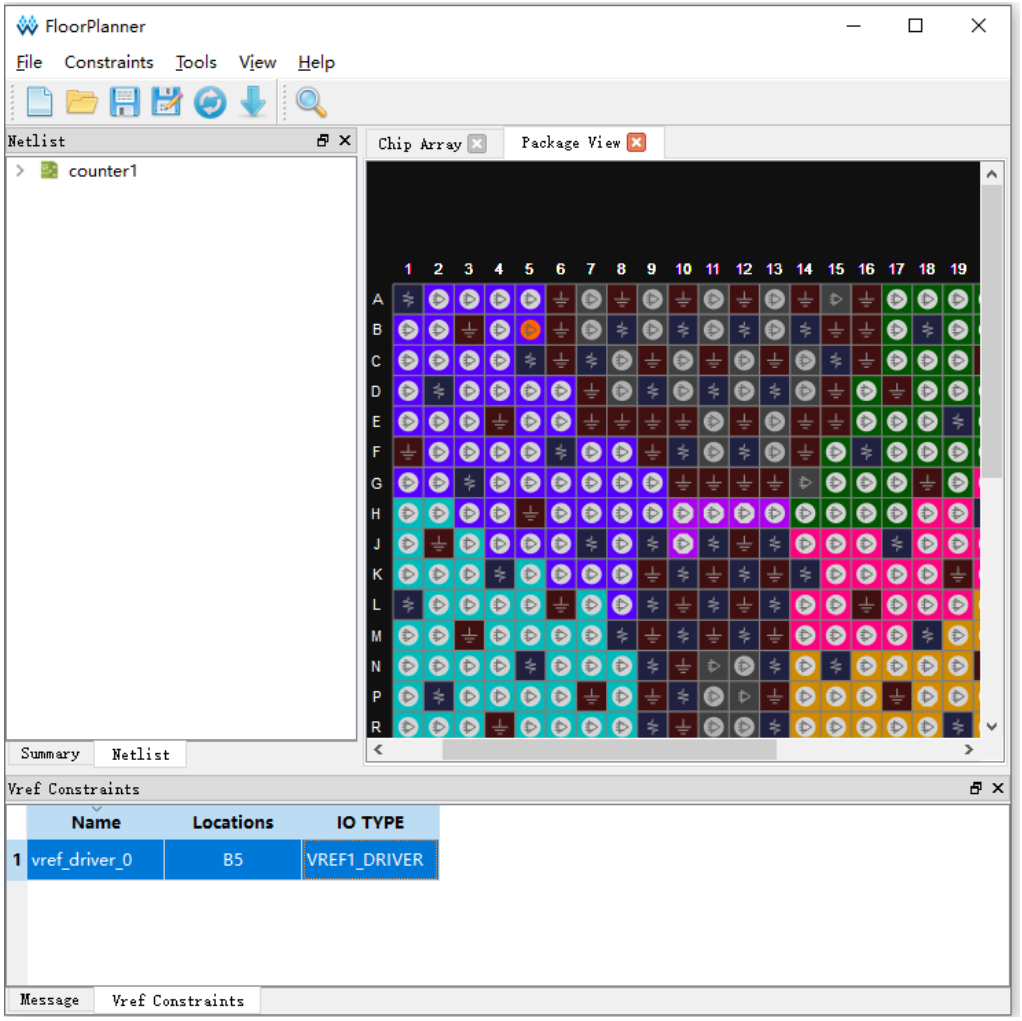
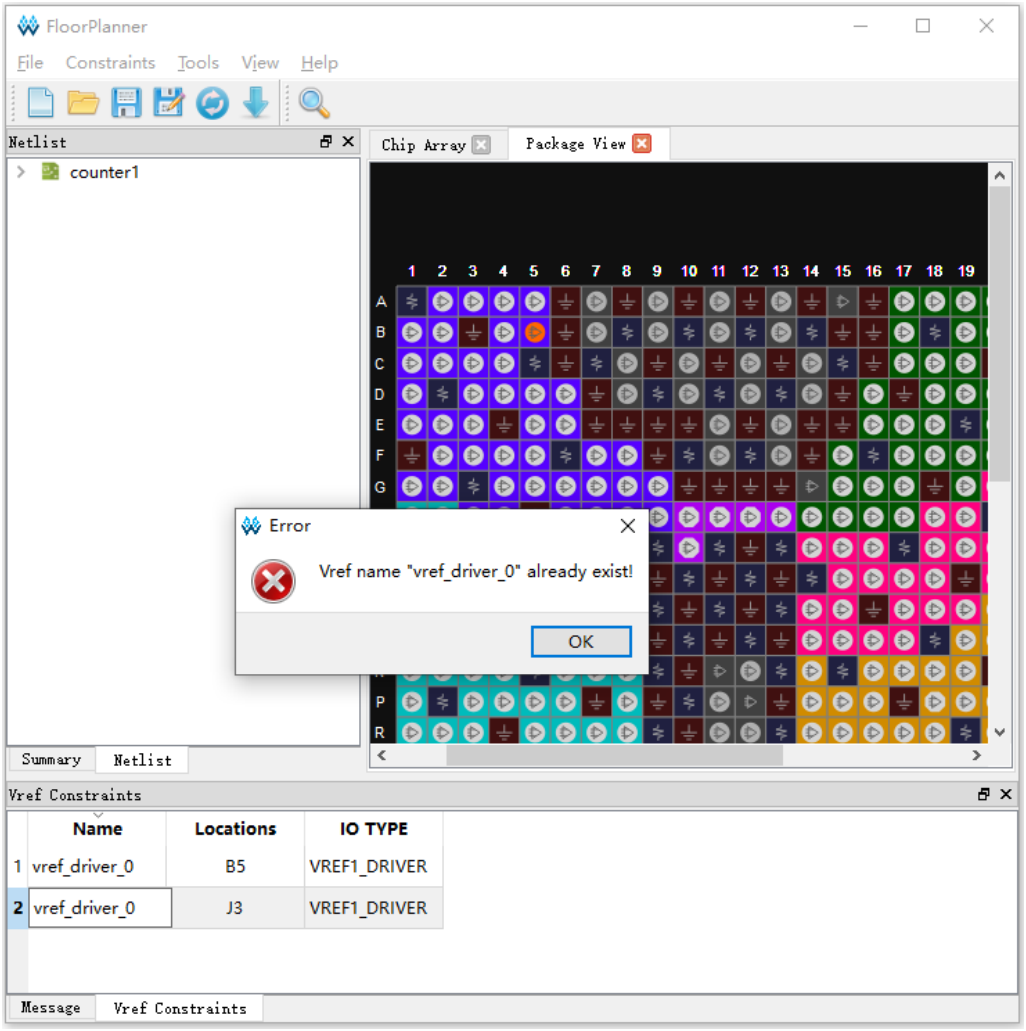


图 4-23 Vref Constraints 名字重复



5 时序调整

FloorPlanner 支持工程的时序调整，通过物理位置的约束或关键路径的修改等方式帮助用户实现时序收敛。

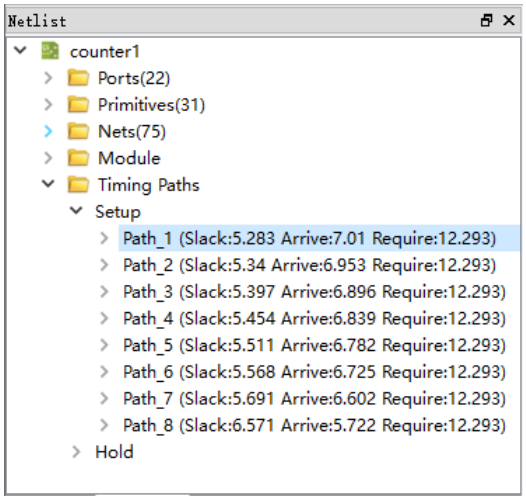
采用 FloorPlanner 进行时序优化的操作流程如下：

1. 新建工程；
2. 运行“Synthesize”实现综合，生成网表文件，扩展名为.vg；
3. 添加物理约束文件和时序约束文件。物理约束和时序约束非必需选项，但可帮助用户更好地实现功能，建议添加；
4. 运行“Place & Route”进行布局布线，生成布局信息文件和时序路径信息文件；
5. 查看时序报告，如最大频率满足设计需求可不再调试。如果时序不满足，需使用 FloorPlanner 产生多种组合的约束，多次迭代实现时序收敛；
6. 运行“Place & Route”后运行启动 FloorPlanner 工具，在“Netlist > Timing Paths”窗口中显示读入的关键路径信息，显示每条 Path 的 Slack、Arrive time、Require time 等信息，如图 5-1 所示；

注！

在反复调试过程中，不必重复启动 FloorPlanner，用户使用“Reload”重新加载即可重新加载布局信息文件和时序路径文件。

图 5-1 读取时序路径文件



7. 在时序收敛的调试过程中，需找出工程中的关键路径，通过修改设计或调整布局实现时序收敛。在 FloorPlanner 中，可通过调整位置信息实现时序收敛，步骤如下：

   a). 查看关键路径信号流向；
   在时序收敛的调试中，关键路径的信号流向是一个因素。在 Chip Array 窗口中右键> Place View > All Instance，调整到工程布局视图，在 FloorPlanner 的 Netlist 窗口中选择一条关键路径，右击选择 Highlight 如图 5-2 所示，在“Chip Array”窗口中可观察到该路径的信号流向，如图 5-3 所示。

   b). 调整不合理的位置信息。
   如图 5-3 所示，模块的分布相对集中，只有一个分布的位置相对较远。再观察关键路径的信号流向，路径比较曲折，跨度较大，是影响时序的一个因素。以拖拽的方式调整位置不合理的那个模块，减少信号的曲折路径，如图 5-4 所示。

8. 重新运行“Place & Route”，查看时序结果。如果频率满足用户需求可不再做调试，如果不满足重复上述 5、6、7 步骤。

图 5-2 高亮关键路径操作

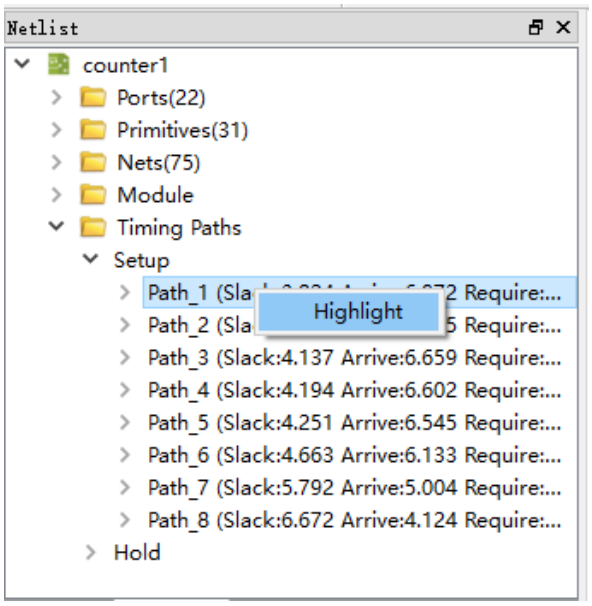


图 5-3 关键路径信号流向示意图

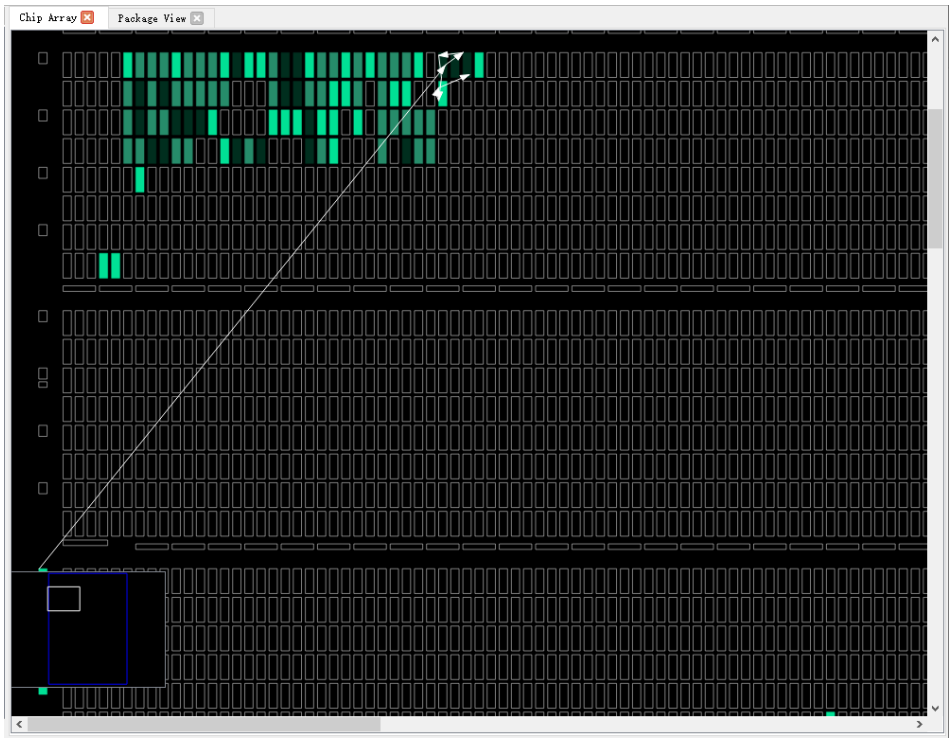
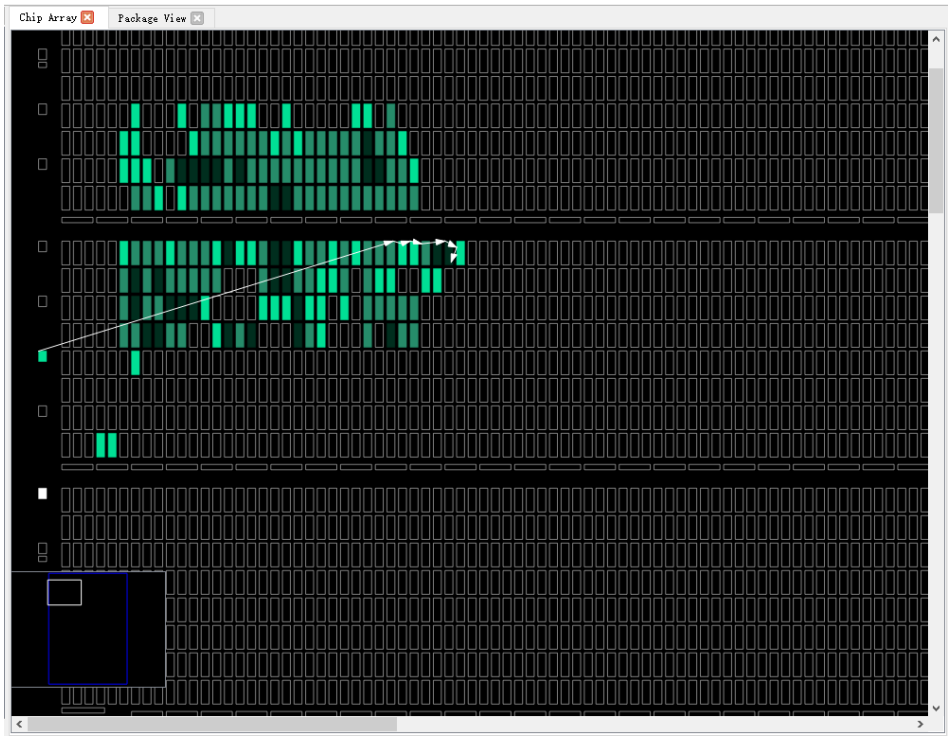


图 5-4 调整后的位置信息



GOWIN高云
智慧逻辑 定制未来