



Gowin 器件 设计优化与分析手册

SUG113-1.1,2017-09-11

版权所有©2017 广东高云半导体科技股份有限公司

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、翻译本档内容的部分或全部，并不得以任何形式传播。

免责声明

本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档进行适时的更新。

版本信息

日期	版本	说明
2016/12/20	1.0	初始版本。
2017/09/11	1.1	1. 更新 Gowin 器件对 DSP 模块的支持； 2. 修改时序报告内容描述。

目录

目录	i
图目录	iv
1 关于本手册	1
1.1 手册内容	1
1.2 适用产品	1
1.3 相关文档	1
1.4 术语、缩略语	2
1.5 技术支持与反馈	2
2 HDL 编码风格要求	3
2.1 HDL 编码总体要求	3
2.1.1 分层次编码综合要求	3
2.1.2 流水线编码设计要求	3
2.1.3 并行条件和优先级条件比较	4
2.1.4 避免生成锁存器	4
2.1.5 全局复位和局部复位	4
2.1.6 时钟使能	4
2.1.7 选择器	4
2.1.8 双向缓存	4
2.1.9 跨时钟域处理	5
2.1.10 块存储器和分布式存储器编码风格	5
2.1.11 DSP 编码风格	5
2.1.12 资源共享	6
2.2 FSM 状态机编码要求	6
2.2.1 状态机总体描述	6
2.2.2 状态机状态编码方式	6
2.2.3 安全状态机的初始状态值和默认状态值	6
2.2.4 完整条件和并行条件	7
2.3 器件硬件特性	7
2.3.1 IO 逻辑	7

2.3.2 DSP	7
2.3.3 块存储器和分布式存储器	7
2.4 低功耗编码	7
2.5 避免综合和仿真不一致的编码要求	7
2.5.1 敏感列表	8
2.5.2 阻塞赋值和非阻塞赋值	8
2.5.3 信号扇出	8
3 设计规划	9
3.1 Gowin 云源软件设计规划流程	9
3.1.1 Gowin 云源软件约束流程	10
3.1.2 Gowin 云源软件设计规划工具	12
3.2 管脚分配	13
3.2.1 管脚分配约束	13
3.2.2 管脚分配 Package View	14
3.2.3 管脚分配规则	14
3.2.4 管脚移植	14
3.2.5 管脚 DRC 检查	15
3.3 时钟分配	15
3.3.1 时钟资源分类	15
3.3.2 时钟资源分配规则	15
3.3.3 Gowin 云源软件时钟资源分配流程	15
3.4 平面布局	16
3.4.1 平面布局定义	16
3.4.2 平面布局约束	16
3.4.3 设计性能优化策略	16
3.4.4 平面布局特殊考虑	17
4 时序收敛	18
4.1 时序约束	19
4.2 时序问题的潜在原因	20
4.2.1 综合报告分析	20
4.2.2 时序报告分析	21
4.2.3 Total Negative Slack	22
4.2.4 Minimum Pulse Width	22
4.2.5 High Fanout Nets	22
4.2.6 Route Congestions	23
4.2.7 Timing Constraints Report	23
4.2.8 Clock Latency	23

4.2.9 IO Delay	23
4.3 时序收敛步骤	23
4.3.1 前端时序收敛策略	23
4.3.2 后端时序收敛策略	24
4.4 Gowin 器件架构对时序影响	24
4.4.1 块存储器的布线差异.....	24
4.4.2 DSP 的布线差异	25
4.4.3 查找表不同输入管脚延时差异	25
4.4.4 CFU 时钟使能管脚延时差异	25

图目录

图 2-1 Gowin 云源软件 IP Core Generator 存储器.....	5
图 2-2 Gowin 云源软件 IP Core Generator DSP	6
图 3-1 设计 RTL View	10
图 3-2 设计 Technology View	10
图 3-3 Physical Constraints Editor 工具界面.....	11
图 3-4 Package View 界面	14
图 3-5 Global Clock Constraints 界面.....	16
图 4-1 Timing Constraints Editor 界面	19
图 4-2 时序报告	21

1 关于本手册

1.1 手册内容

FPGA 设计优化主要分为编码风格、设计规划和时序收敛三大部分，这些因素直接决定了 FPGA 设计的成败。

编码风格直接影响 FPGA 设计的实现并最终影响设计的性能。尽管现在主流的综合工具集成了一系列优化算法，但是用户仍有必要遵循一定的编码风格去引导综合工具在特定 FPGA 架构上达到最优结果。

设计规划用于指导用户把设计更好地适配到所选用的 FPGA 上并合理地平衡面积和速度的要求，目的在于把 Gowin 器件支持的所有功能和特性完美地呈现出来。

时序收敛可以保证用户设计满足某个特定的时序需求，这部分主要描述时序需求、时序约束以及时序优化的方法。

1.2 适用产品

本手册中描述的信息适用于以下产品：

1. GW1N 系列 FPGA 产品：GW1N-1, GW1N-2, GW1N-4, GW1N-6, GW1N-9
2. GW1NR 系列 FPGA 产品：GW1NR-4, GW1NR-9
3. GW2A 系列 FPGA 产品：GW2A-18, GW2A-55
4. GW2AR 系列 FPGA 产品：GW2AR-18

1.3 相关文档

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

Gowin 云源软件用户指南

1.4 术语、缩略语

下表中列出了本手册中出现的相关术语、缩略语及相关释义。

术语、缩略语	全称	含义
FPGA	Field Programmable Gate Array	现场可编程门阵列
HDL	Hardware Description Language	硬件描述语言
FSM	Finite State Machine	有限状态机
DSP	Digital Signal Processor	数字信号处理器
DRC	Design Rule Check	设计规则检查
RTL	Register Transfer Level	寄存器传输级

1.5 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：<http://www.gowinsemi.com.cn/>

E-mail: support@gowinsemi.com

Tel: +86 755 8262 0391

2 HDL 编码风格要求

2.1 HDL 编码总体要求

2.1.1 分层次编码综合要求

对于复杂的系统级设计，分层次编码是很有必要的。分层次编码可以一次综合所有的模块，也可以按层次结构分别综合各个模块。当一次综合所有模块时，设计可以被综合成一个无层次结构的模块或者多个带层次结构的模块。

这两个策略各有优缺点，在复杂的系统级设计中分层次编码更具优势，原因在于分层次编码更容易定位问题，同时提高了模块的重用性并缩短了开发周期。当然分层次编码导致优化只能局限在模块内部而不能跨模块优化，这会导致更高的资源利用率和更低的设计性能。但是用户可以采用合理的分层结构克服这些缺陷。下面是创建分层次编码结构的一些建议。

- 顶层模块只用于调用子模块，控制逻辑尽量在各个子模块实现。
- 管脚输入输出缓存需要在顶层例化。
- 所有的输入、输出和双向管脚需要在顶层例化。
- 只允许在顶层使用双向管脚的三态声明。
- 尽可能保证模块所有输出信号全部使用寄存器，这样的好处在于：
 - 组合逻辑和寄存器在一个模块里，克服了综合不能跨模块优化的不足。
 - 把相关逻辑放在一个模块里，可以保证资源共享并优化关键路径。
 - 分割不相关逻辑到不同模块，则可以采用不同的优化策略，比如速度优先或者面积优先。

2.1.2 流水线编码设计要求

流水线设计通过重构多逻辑级数的数据路径并增加时钟周期分割逻辑级数达到提高设计性能的目的。流水线结构是提高数据路径速度的有效方法，但需要注意的是它会增加数据路径的传输时延。

2.1.3 并行条件和优先级条件比较

`if-then-else` 结构生成优先级条件逻辑，而 `case` 结构生成并行条件逻辑。`if-then-else` 结构可以包含多个不同的条件表达式，但是 `case` 结构只能包含一个表达式。在条件互斥的情况下，可以认为 `if-then-else` 结构和 `case` 结构是等效的。

2.1.4 避免生成锁存器

用户在 `FPGA` 设计中应该避免使用锁存器，综合工具通过组合逻辑环路实现锁存器，不可避免地造成资源浪费和性能下降，同时组合逻辑环路给静态时序分析带来了很大困难。

综合工具在组合逻辑条件表达式不完整情况下会生成锁存器，比如 `if-then-else` 结构中没有 `else` 或者 `case` 结构中没有 `default`。为了避免不必要的锁存器，在条件语句中需要把所有条件都遍历到。

2.1.5 全局复位和局部复位

高云器件包含一个专用的全局复置位网络，直接连接到器件的内部逻辑，可用作异步/同步复位或异步/同步置位，`CFU` 和 `I/O` 中的寄存器均可以独立配置。全局复位资源并不占用普通绕线资源，并且遍布芯片各个角落。在设计中主要有两种方法调用全局复位资源，即使用全局复位去控制 `FPGA` 中的所有模块或者选用扇出最大的复位作为全局复位以节省绕线资源。局部复位一般扇出较小，可以考虑使用普通绕线作为复位信号。

2.1.6 时钟使能

`FPGA` 设计中应该避免使用门控时钟，门控时钟会导致很多不可控的时序问题，比如不可预知的时钟飘移。`Gowin` 器件 `CLS` 结构中包含专用时钟使能信号，用户可以使用时钟使能达到门控时钟同样的功能而不必担心时序问题。以下是使用 `Gowin` 器件时钟使能的一些注意事项。

- 只有寄存器支持时钟使能，锁存器不支持。
- 同一个 `CFU` 里面的各个 `CLS` 共用一个时钟使能信号。
- 所有寄存器时钟使能高有效。
- 时钟使能优先级高于同步复位。

2.1.7 选择器

`CLS` 里面的查找表可以灵活配置以实现二选一选择器、三选一选择器、四选一选择器或五选一选择器等等，用户还可以通过调用多个四输入查找表级联实现更大的选择器。

2.1.8 双向缓存

使用双向缓存可以节约管脚，控制输出使能达到节省功耗的目的。用户

可以关闭综合工具自动管脚输入输出缓存插入选项，为特定的管脚例化特定的管脚输入输出缓存。

2.1.9 跨时钟域处理

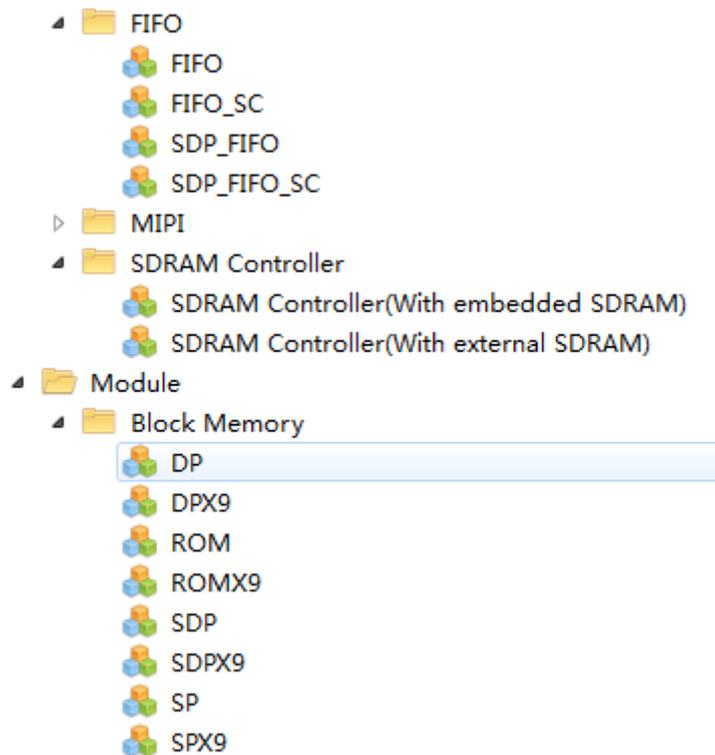
当数据路径从一个时钟域跨越到另一个时钟域时，必须确保生成的亚稳态不会影响数据的建立时间和保持时间。对于单比特信号，建议使用两级寄存器结构消除亚稳态；对于多比特信号，建议使用异步 FIFO。

2.1.10 块存储器和分布式存储器编码风格

虽然随机存储器的 RTL 行为级描述可定制且非常直观，但不同的编码风格可能会产生不同的综合结果。

对于 Gowin 器件而言，建议使用 Gowin 云源软件 IP Core Generator 工具产生块存储器和分布式存储器。Gowin 器件支持多种存储结构，包括双端口 RAM、单端口 RAM、伪双端口 RAM、ROM、同步 FIFO 以及异步 FIFO，如图 2-1 所示。

图 2-1 Gowin 云源软件 IP Core Generator 存储器

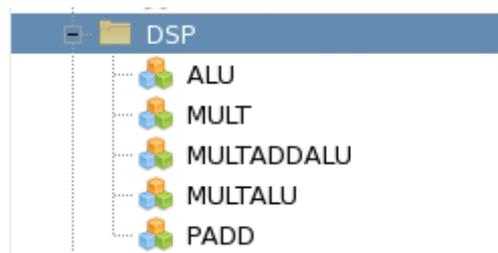


2.1.11 DSP 编码风格

虽然 DSP 的 RTL 行为级描述可定制且非常直观，但不同的编码风格可能会产生不同的综合结果。

对于 Gowin 器件而言，建议使用 Gowin 云源软件 IP Core Generator 工具产生 DSP。Gowin 器件支持多种 DSP 结构，包括 MULT、MULTALU、MULTADDALU 以及 PADD。

图 2-2 Gowin 云源软件 IP Core Generator DSP



2.1.12 资源共享

很多综合工具默认都支持资源共享，这样的好处是可以节约资源，但是资源共享会增加绕线资源并造成局部拥塞。当出现时序收敛问题时，建议把全局资源共享设置关闭，并对特定的模块使用综合属性`/*synthesis syn_sharing = "on" */`。

2.2 FSM 状态机编码要求

有限状态机在时钟边沿完成当前状态到下一个状态的跳转，下面主要讨论状态机编码的方法和策略。

2.2.1 状态机总体描述

状态机实现主要有两种方式，一种是在一个进程中同时处理状态跳转和状态输出；另一种则在两个独立的进程中分别处理状态跳转和状态输出。建议用户使用第二种方式，它不仅便于阅读和修改，而且对于状态无寄存直接输出情况，第二种方式可以保证不会引起额外的时延。

2.2.2 状态机状态编码方式

目前主要有二进制编码、独热编码以及格雷编码三种状态机编码方式。二进制编码及格雷编码使用较少的触发器，但使用较多的组合逻辑，而独热编码恰好相反。

独热编码的最大优势在于状态比较时仅仅需要比较一个比特位，从而一定程度上简化了译码逻辑。虽然在表示同等数量的状态时，独热编码占用较多的比特位，即使用较多的触发器，但这些触发器占用的面积可以和译码电路省下来的面积相抵消。

因此，对于状态数小于 5 的状态机，建议使用二进制编码和格雷编码；对于状态数大于 5 的状态机，建议使用独热编码。

2.2.3 安全状态机的初始状态值和默认状态值

安全状态机在上电后必须初始化进入一个初始有效状态，用户可以通过上电复位或全局复位操作进入一个初始有效状态。同时状态机必须有一个默认状态值来保证状态机不会进入非法状态，当代码中有状态组合没有被遍历时，非法状态就会产生。

2.2.4 完整条件和并行条件

RTL 语言有专门的属性来定义默认状态，完整条件可以确保不会出现非法状态；并行条件则确保所有状态都是互斥的，在同一时间只有一个条件满足。

2.3 器件硬件特性

2.3.1 IO 逻辑

输入输出逻辑集成了串并转换、并串转换、延迟控制以及字节对齐等功能，主要用于高速数据传输场合。输入输出逻辑支持普通模式、单倍速率(SDR)模式、双倍速率(DDR)、四倍速率、八倍速率以及十倍速率等多种模式。用户可以根据具体的设计需求调用 Gowin 器件的输入输出逻辑。

2.3.2 DSP

Gowin 器件的 DSP 支持 9bit、18bit 两种宽度的有符号数及无符号数乘法器、乘法、累加器、54bit 的算术/逻辑运算单元、桶形移位器，同时 DSP 支持输入输出寄存器的流水线和旁路功能。

2.3.3 块存储器和分布式存储器

Gowin 器件提供了丰富的块状静态随机存储器资源，简称块存储器 (B-SRAM)。Gowin 器件中的每个 B-SRAM 可配置为最高 18Kbits，数据位宽和深度都可以配置。每块 B-SRAM 有两个端口，两个端口彼此独立，有独立的时钟、地址端口、数据端口和控制端口，但是两个端口共享存储空间。每个 B-SRAM 可以配置 4 种操作模式：单端口模式、双端口模式、伪双端口模式和 ROM 模式。

分布式存储器可配置成深度为 16，位宽为 1/2/4bit 的单端口随机存储器，也可以配置成深度为 16，位宽为 1/2/4bit 的伪双端口随机存储器，还可以配置 16bit x 1 的只读 ROM16。

2.4 低功耗编码

以面积为优化目标可以降低逻辑和布线资源使用率，进而降低功耗，建议使用 Gowin 云源软件 IP Core Generator 去调用 Gowin 器件内部基本单元，以得到低功耗的实现，消除已知的毛刺以降低电源功耗，降低信号翻转率并在适当时刻进入休眠状态可降低系统功耗。

2.5 避免综合和仿真不一致的编码要求

特定编码风格会导致综合和仿真的结果不一致，原因在于仿真工具会忽略一些错误信息而这些错误信息会被综合工具检测出来，这些错误一般可以通过 BKM 检测出来，因此编码风格建议采用 Gowin 编码风格。

2.5.1 敏感列表

组合逻辑的信号敏感列表必须包含所有输入输出信号以确保综合和仿真结果一致。

2.5.2 阻塞赋值和非阻塞赋值

组合逻辑一般建议使用阻塞赋值；时序逻辑一般建议使用非阻塞赋值。

2.5.3 信号扇出

信号扇出控制用于保证综合后的扇出在一个合适的范围，综合工具通过复制电路来减少信号扇出。对特定信号使用综合属性 `syn_maxfan` 可以得到更好的时序收敛效果。Gowin 器件架构可以使用专用时钟网络处理大扇出的时钟信号，并使用专用全局复位网络处理大扇出的复位信号。但是综合工具倾向于对大扇出的逻辑进行电路复制，复制电路会带来一系列副作用，用户需要根据实际情况灵活使用综合属性 `syn_maxfan` 以避免生成过多的复制电路。

3 设计规划

FPGA 设计规划主要包括两个阶段：

1. 在设计前需要定义设计的功能和架构，挑选合适的 FPGA 器件并进行 RTL 编码。
2. 在设计过程中需要将设计以最优方式适配到所选的器件中，并充分利用 FPGA 器件中的资源。

这两个阶段是相互影响的，本章主要侧重介绍第二阶段，目的在于让用户使用 Gowin 云源软件将 Gowin 器件的功能和特性得到最大程度的发挥。

3.1 Gowin 云源软件设计规划流程

并不是所有设计都需要涉及设计规划，但是设计规划对大多数设计是有益无害的，尤其对一些资源使用率高，时序要求严格的大型设计。对于这些大型设计，设计规划可以减少潜在的布局布线问题和时序收敛问题，并提高设计的重用性和可移植性。

在 Gowin 云源软件中，设计规划开始于综合完成后布局布线开始前。CST 文件包含所有的设计规划约束用于指导后端布局布线。一旦设计规划改变，就意味着 CST 文件的改变，整合设计流程回到综合完成后布局布线开始前这个阶段。CST 文件约束及使用方法请参见 [《Gowin 设计约束指南》](#)。

3.1.1 Gowin 云源软件约束流程

Gowin 云源软件约束流程允许客户对综合后的端口、网表、寄存器、实例进行约束。**CST** 文件是用户定义设计规划约束的一个主要入口。

用户设计在综合后生成一个 VM 文件，它包含了整个设计的逻辑信息，用户可以使用 Synplify Pro 的 RTL View 或 Technology View 工具查看综合前和综合后的逻辑单元，如图 3-1 和图 3-2 所示。

图 3-1 设计 RTL View

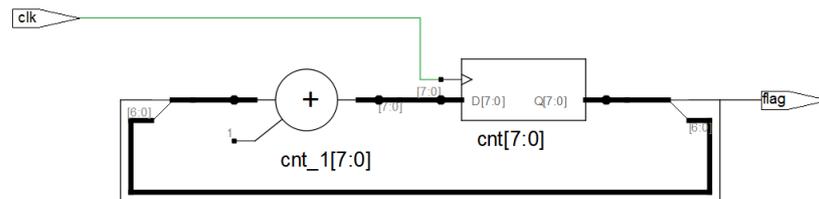
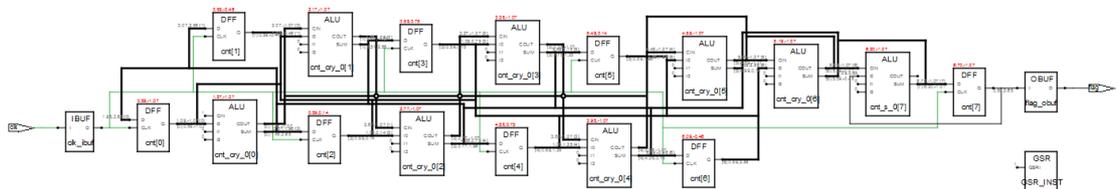
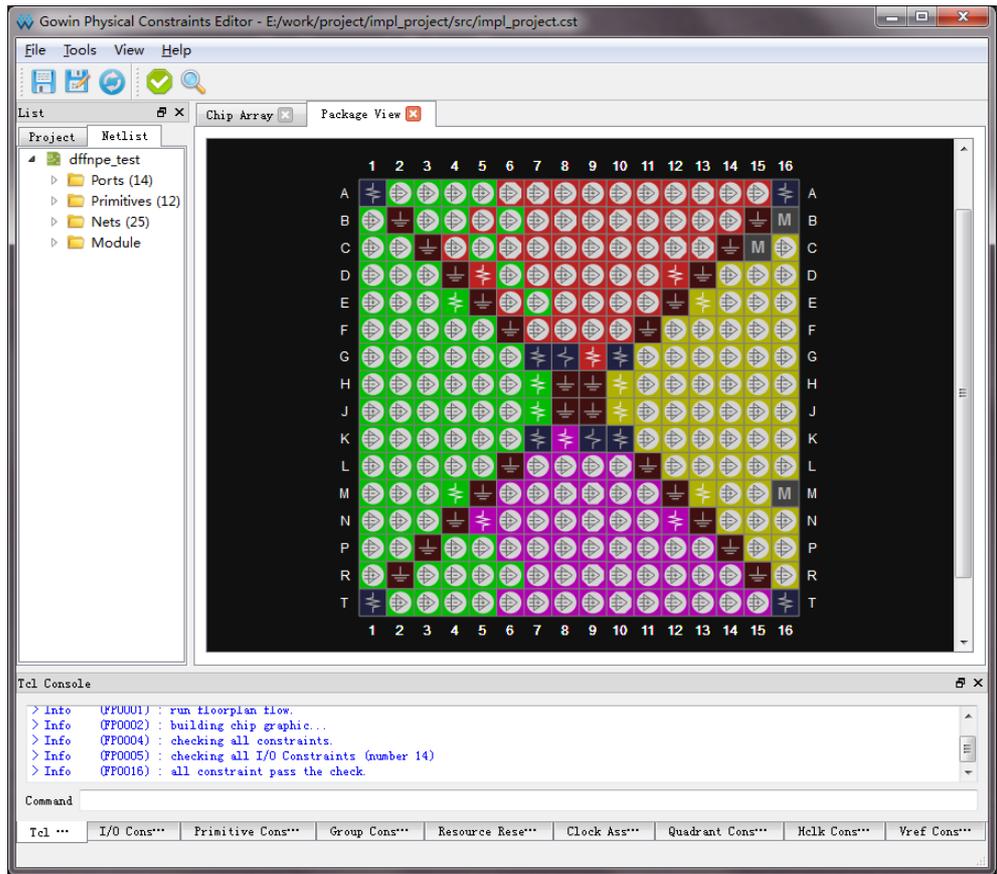


图 3-2 设计 Technology View



后端布局布线软件通过读取 **CST** 文件获取用户的约束信息。如果 **CST** 文件出现语法错误或非法约束错误，后端布局布线软件会直接退出。**CST** 文件可以文本编辑，也可以使用 **Physical Constraints Editor** 工具产生。文本编辑方式适用于高级用户；**Physical Constraints Editor** 工具有检查功能，建议用户尽量使用工具产生 **CST** 文件。如图 3-3 所示。

图 3-3 Physical Constraints Editor 工具界面



3.1.2 Gowin 云源软件设计规划工具

Gowin 云源软件提供 Physical Constraints Editor 工具进行设计规划, 并提供如下功能:

- 预览设计中用户可以处理的所有基本逻辑单元。
- 预览器件中支持的所有硬件资源。
- 分配特定的逻辑单元到特定的硬件资源。
- 使用 DRC 检查把一些潜在的约束错误提前报告给用户。
- 支持拖拽操作实现器件约束。

3.2 管脚分配

管脚分配规划是一个定义管脚标准和位置的过程，它基于用户的实际设计和所选的器件，管脚分配涉及如下步骤：

1. 分配设计中的端口到特定管脚位置。
2. 定义管脚电平标准和管脚特征参数。
3. 检查分配的合法性。
4. 如有必要，根据 PCB 和电路图更改管脚分配和配置。

3.2.1 管脚分配约束

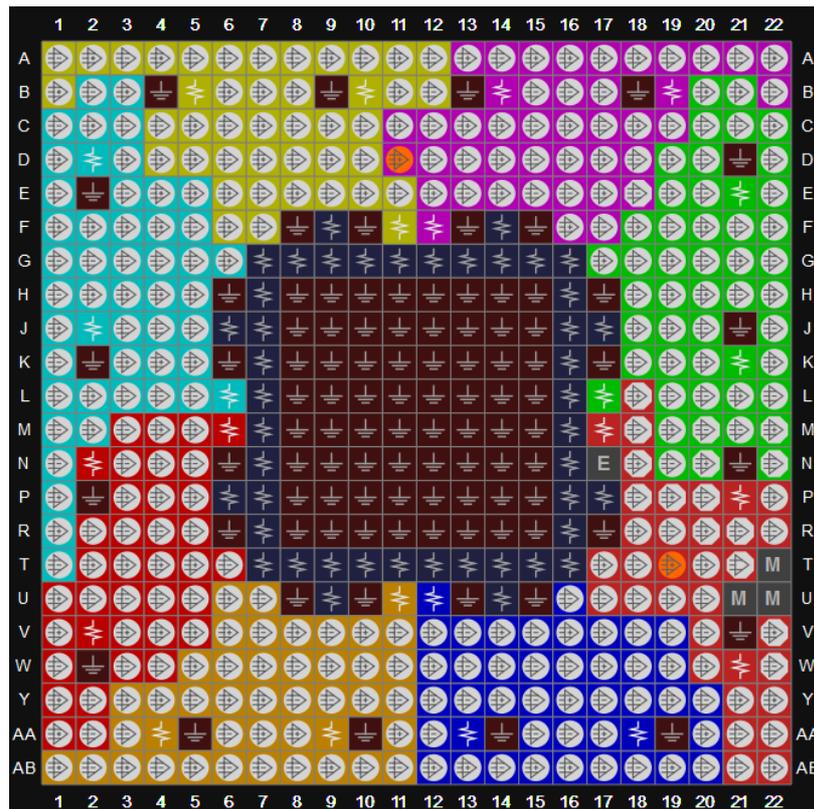
Gowin 云源软件的物理约束编辑器支持系统管脚属性设置，物理约束编辑器用于产生 CST 文件，下面对 CST 文件支持的物理约束作详细介绍。

- 对系统管脚进行物理位置锁定：
IO_LOC "xxx" H4 exclusive
- 为系统管脚设置 IO 类型：
IO_PORT "xxx" IO_TYPE=LVC MOS18D
- 为输出管脚或双向管脚设置驱动能力：
IO_PORT "xxx" DRIVE=12
- 设置上下拉模式，其中 UP：上拉；DOWN：下拉；KEEPER：总线保持；NONE：高阻：
IO_PORT "xxx" PULL_MODE=DOWN
- 为系统管脚设置参考电压，既可以来自外部管脚也可以来自内部参考电压生成器：
IO_PORT "xxx" VREF=VREF1_LOAD
- 为输入管脚或双向管脚设置迟滞量，从小到大依次是 NONE->H2L->L2H->HIGH：
IO_PORT "xxx" HYSTERESIS=L2H
- 为输出管脚或双向管脚打开或关闭漏记开路，提供 ON/OFF 选项：
IO_PORT "xxx" OPEN_DRAIN=ON
- 为输出管脚或双向管脚设置转换速率，SLOW：低噪声模式；FAST：高速模式：
IO_PORT "xxx" SLEW_RATE=SLOW
- 为单端信号设置终端匹配电阻，提供 OFF 和 ON 选项：
IO_PORT "xxx" SINGLE_RESISTOR=ON
- 为差分信号设置终端匹配电阻，提供 OFF 和 ON 选项：
IO_PORT "xxx" Diff_RESISTOR=ON

3.2.2 管脚分配 Package View

FloorPlanner 工具的主界面右侧为 Package view 界面，列出了所选用 package 的管脚位置信息，如图 3-4 所示。图中标注了当前 package 信息对应的可用管脚数目、位置信息以及复用管脚信息。将鼠标放置某个位置上时，会悬浮显示该位置的管脚信息，包括管脚的类型、bank 以及 LVDS 信息等。图中不同颜色代表不同 bank，不同标识代表不同含义，详情可使用悬浮信息查看。

图 3-4 Package View 界面



注!

在 package view 界面进行右键勾选 Enable 'MODE' Pins，灰色 M 专用管脚图标会变成普通管脚图标。

3.2.3 管脚分配规则

- 首先分配专用管脚，比如时钟输入、锁相环输入、DDR 等。
- 分配普通管脚到专门的 BANK 而不是特定的管脚，这样可以使后端布局布线工具计算得到一个最优的结果。
- 对于复用管脚，需要检查是否和器件编程模式相冲突。
- 运行 DRC 检查，及时发现并纠正潜在的错误。

3.2.4 管脚移植

对于封装相同的器件，用户有时候希望把设计从低密度器件移植到高密

度器件用于未来扩展功能, 或者从高密度器件移植到低密度器件以降低成本。但是管脚分配需要保持不变或作一些细微改动以避免 PCB 重新投板。Gowin 云源软件提供了管脚移植特性。在 **Physical Constraints Editor** 和 **FloorPlanner** 可以看到一些非适配的管脚, 可以使用 **LOC_RESERVE** 屏蔽这些管脚。

3.2.5 管脚 DRC 检查

Gowin 云源软件提供管脚设计规则检查, 用于检验管脚分配结果的合法性。DRC 检查主要有两种模式, 即实时检查和按需检查。DRC 检查可以在设计早期发现很多常见的错误, 防止错误的管脚用法, 最终的管脚分配结果以 **Place&Route** 报告为准。

3.3 时钟分配

3.3.1 时钟资源分类

Gowin 器件包括全局时钟资源、高速时钟资源、象限时钟资源。

- 专用的全局时钟资源贯穿整个芯片, 它是基于象限的。
- 高速时钟横跨器件的左、右、上或者下边, 主要应用于大吞吐量的高速接口。
- 一个全局时钟可以分割成 **LEFT** 和 **RIGHT** 共两个 (**GW1N**、**GW1NR** 系列) 或 **TOPLEFT**、**TOPRIGHT**、**BOTTOMLEFT**、**BOTTOMRIGHT** 共四个 (**GW2A**、**GW2AR** 系列) 象限时钟, 象限时钟提供时钟动态选择功能。

3.3.2 时钟资源分配规则

在考虑时钟资源规划时, 时钟主频和时钟扇出是需要考量的两个重要的因素, 此外也需要考虑 Gowin 器件本身包含时钟资源的限制。

通常情况下, 专用的时钟资源可以得到更好的时序结果, 因为专用时钟资源的相对延时最小, 节省下来的布线资源可以更好地应用于拥塞度比较高的场合。只有在极少数情况下用户需要将时钟分配到普通绕线上, 普通绕线会导致时钟的相对延时很大, 因此只有低速小扇出的应用场景才会考虑使用普通绕线资源作为时钟。下面介绍时钟资源分配的一般规则:

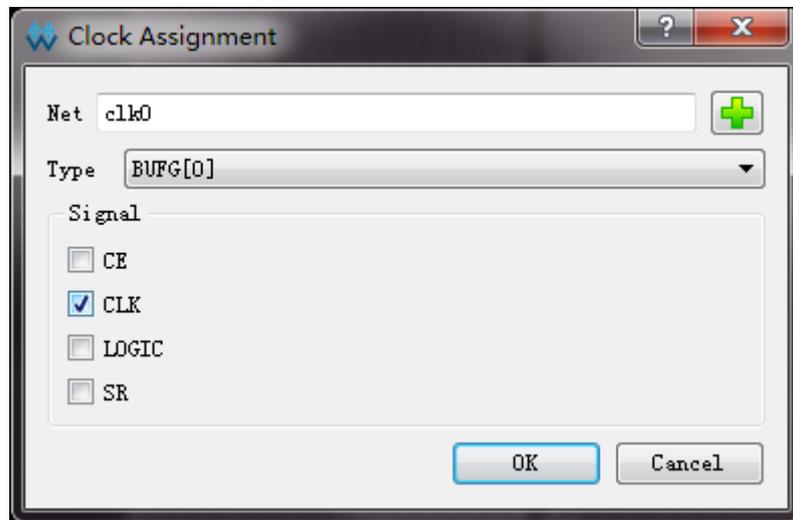
- 确定用户设计中时钟数目和每个时钟的扇出。
- 确定目标器件提供的时钟资源。
- 明确每个时钟的速度要求。
- 高速大扇出的时钟一般需要分配到全局时钟。
- 如果全局时钟数目小于设计中时钟数, 需要考虑使用象限时钟。
- 对大吞吐量的高速接口使用高速时钟资源。

3.3.3 Gowin 云源软件时钟资源分配流程

```
NET_LOC "xxx" BUFG0 = CLK | CE | SR | LOGIC
```

分配时钟信号到专门的全局时钟网络, BUFG0~BUFG7 分别代表 Gowin 器件支持的八个全局时钟。其中 CLK 表明约束对象是时钟; CE 表明约束对象是时钟使能; SR 表明约束对象是同步复位; LOGIC 表明约束对象是逻辑器件。如图 3-5 所示。

图 3-5 Global Clock Constraints 界面



3.4 平面布局

3.4.1 平面布局定义

平面布局从逻辑角度划分设计基本单元, 这会改变设计的物理布局 and 实现方式。平面布局主要通过锁定 FPGA 内部资源的位置来实现。

对于 Gowin 器件, 平面布局对于改善设计性能有很大帮助, 尤其有助于优化层次结构清晰的设计。平面布局提供了一种结合自动化和用户控制的编译方式, 提高了设计的模块化和重用性, 它是增量编译的基础。

3.4.2 平面布局约束

```
INS_LOC "cnt[5]" R2C2
```

约束特定的对象到特定的 CFU 位置。

```
GROUP hh = { "cnt_Z[1]" "cnt_Z[2]" "cnt_Z[3]" "cnt_Z[4]" "cnt_Z[5]"  
"cnt_Z[6]" "cnt_Z[7]" }
```

```
GRP_LOC hh R[3:6]C[4:6]
```

把特定对象分组并约束到特定的区域位置。

3.4.3 设计性能优化策略

- 逻辑模块划分应该基于层次结构。
- 逻辑模块划分应该基于关键路径。
- 逻辑模块划分应该基于大扇出的输入输出信号。
- 划分逻辑模块并对各个模块使用不同的优化策略。

3.4.4 平面布局特殊考虑

- 块存储器可以单独进行位置约束，不需要分组。
- 大逻辑模块进行位置约束需要指定起始位置和相对大小。
- 进位链和总线不需要分组。
- 互联逻辑不需要分组。

4 时序收敛

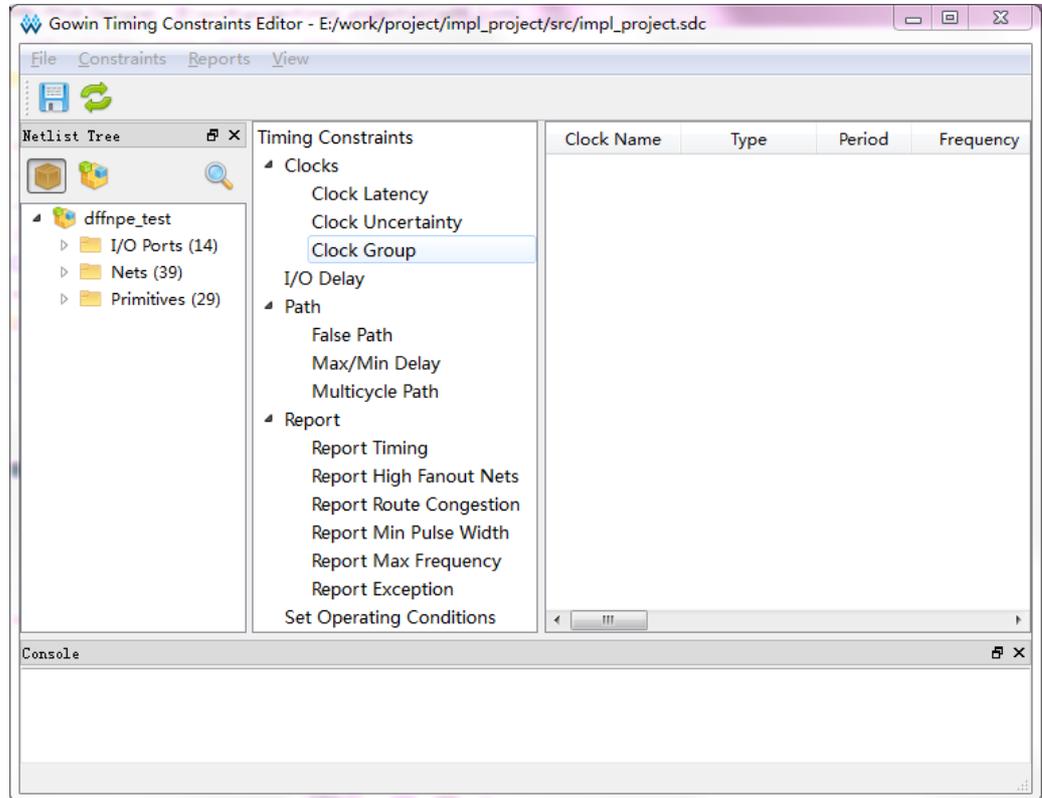
每个设计一般都需要运行在特定的速率上，FPGA 设计中主要涉及三类需求，即时序、带宽和时延。带宽和时延是互斥的，大带宽一般需要多级流水线，这会增加时延，反之时延小需要复杂的组合逻辑路径来实现，这会减少流水线级数，降低带宽。因此，在定义设计架构时首先需要把带宽和时延平衡好。

通常，时序收敛是决定设计能否满足特定速率需求的关键因素，这往往需要花费大量时间运用各种技术去达到时序收敛。本章主要讨论时序收敛，并讨论如何使设计达到时序收敛。

4.1 时序约束

Gowin 云源软件 Timing Constraints Editor 支持常用的时序约束，如图 4-1 所示。

图 4-1 Timing Constraints Editor 界面



注！

关于时序约束语法规范请参见《Gowin 设计约束指南》>附录 B 时序约束语法规范。

4.2 时序问题的潜在原因

4.2.1 综合报告分析

综合工具运行在时序优化模式下，它会使用时序约束；当综合工具运行在面积优化模式下，时序约束就会被忽略掉。用户可以根据需要设置 Syplify Pro 的属性“Frequency”。在时序模式下，用户需要设置“Frequency”属性，这是设计主频的一个全局属性，默认值是 AUTO，当用户使用时序约束对设计中的时钟进行约束时，默认值就会被忽略掉。

用户也可以使用其他的一些综合属性去改善设计的性能。比如资源共享、流水线和重定时等。

使用合适的综合策略以及时序约束后，用户可以运行综合过程。综合过程结束后会产生报告，报告中包含时序信息，可以搜索关键词 **START OF TIMING REPORT/END OF TIMING REPORT**。有几个地方用户需要关注下：

- 起始处指示了综合用到的时序约束文件 SDC 以及默认的时钟主频。
- Performance Summary 处，列出了设计中各个时钟的 worst slack。
- Clock relationship 处，列出了寄存器和寄存器之间的 slack，如果设计中有多个时钟，这些时钟关系会罗列在这里。
- Interface information 处，列出了外部管脚时序信息，包括管脚的 input setup 以及 clock to output 的时序和 slack。

综合时序报告是基于综合结果的，它并不包括布局和布线信息。因此布局布线后的时序报告才能准确地反映时序分析结果。

4.2.2 时序报告分析

Gowin 云源软件在执行完时序分析后会生成时序报告，其格式如图 4-2 所示。

图 4-2 时序报告

Timing Analysis Report

Setup Analysis:

Path1

Path Summary:

Slack:	8.355
Data Arrival Time:	3.363
Data Required Time:	11.718
From:	reg11_Z
To:	reg12_Z
Launch Clk:	sysclk[R]
Latch Clk:	sysclk[R]

Data Arrival Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysclk(clock)
0.000	0.000	tCL	RR	1	IOL11[A]	clk1_ibuf#i
1.024	1.024	tINS	RR	2	IOL11[A]	clk1_ibuf/O
2.375	1.351	tNET	RR	1	R2C2[1][B]	reg11_Z/CLK
2.999	0.624	tC2Q	RR	1	R2C2[1][B]	reg11_Z/Q
3.363	0.364	tNET	RR	1	R2C2[1][A]	reg12_Z/D

Data Required Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
10.000	10.000					active clock edge time
10.000	0.000					sysclk(clock)
10.000	0.000	tCL	RR	1	IOL11[A]	clk1_ibuf#i
11.013	1.013	tINS	RR	2	IOL11[A]	clk1_ibuf/O
12.143	1.130	tNET	RR	1	R2C2[1][A]	reg12_Z/CLK
11.943	-0.200	tUnc				reg12_Z
11.718	-0.225	tSu		1	R2C2[1][A]	reg12_Z

Path Statistics:

Clock Skew:	-0.232
Setup Relationship:	10.000
Logic Level:	0
Arrival Clock Path Delay:	cell: 1.024, 43.112%; route: 1.351, 56.888%
Arrival Data Path Delay:	cell: 0.000, 0.000%; route: 0.364, 36.874%; tC2Q: 0.624, 63.126%
Required Clock Path Delay:	cell: 1.013, 47.279%; route: 1.130, 52.721%

时序报告第一部分总结了关键路径的一系列时序信息。

- **前级寄存器**：即一个时序路径上发送数据的寄存器。
- **后级寄存器**：即一个时序路径上接收数据的寄存器。
- **launch 边沿**：即前级寄存器发送数据的那个时钟边沿。这个边沿也是时序分析计算时间的起点。
- **latch 边沿**：即后级寄存器接收数据的时钟边沿。该边沿也是时序分析计算时间的终点。
- **Slack (裕量)**：数据允许最迟到达时间减去数据实际到达时间。在报告中是 **Data Required Time** 减去 **Data Arrival Time**。需要注意，时序分析工具精确到小数点后三位，这可能导致时序分析报告中显示的数据小数点后第三位不一定准确。
- **Data Arrival Time**：是指前级寄存器在时钟的 **launch** 边沿发送的数据到达后级寄存器的时间。
- **Data Required Time**：其本质是指后级寄存器要接收前级寄存器发来的数

据, 这个数据到达后级寄存器允许的最晚时间。在计算方法上是, 从 latch 边沿算起, latch 边沿到达后级寄存器的时刻。

- **From:** 指的是时序分析中时序分析模型的起点。
- **To:** 指的是时序分析中时序分析模块的终点。
- **Launch Clock:** 指的是 launch 边沿是由哪个时钟提供的。
- **Latch Clock:** 指的是 latch 边沿是由哪个时钟提供的。

经过计算 Data Arrival Time 和 Data Required Time 之后, 两者之差就是时序收敛裕量。正的表示时序收敛, 负的表示时序不收敛, 需要修改设计以提高时序性能, 或是改变约束条件, 降低频率。

时钟报告第二部分和第三部分分别对 Data Arrival Path 和 Data Required Path 进行了详细的分析, 并罗列了延迟的类型、扇出、位置等信息。

时钟报告第四部分对关键路径进行了分析并做了相应统计。

- **Clock Skew:** 指时钟倾斜, 时钟到达前级寄存器的时刻与后级寄存器的时刻之差。
- **Setup Relationship:** 即为前级寄存器发送数据, 后级寄存器接收数据的时间关系。
- **Logic Level:** 是指 Data Arrival Path 中两个寄存器之间数据通路中的逻辑器件数量, 0 代表直接相连。
- **Arrival Clock Path Delay:** 统计了 Data Arrival Path 上时钟延时的情况。
- **Arrival Data Path Delay:** 统计了 Data Arrival Path 上数据的延时情况。
- **Required Clock Path Delay:** 统计了 Data Required Path 上时钟的延时情况。
- **Hold Check** 计算方法完全同 Setup Check 相似, 涉及到的概念完全一致, 不再赘述。

4.2.3 Total Negative Slack

Total Negative Slack 统计时序分析中所有 Setup Check 时 Slack 为负的值。

Number Of Negative Slack Paths 为时序不收敛的路径总条数, Total Negative Slack 为 Setup Check 时 Slack 负值的累加和。

4.2.4 Minimum Pulse Width

Minimum Pulse Width 分析所有参与时序分析的路径上时序元件的最小脉冲跨度, 包括高电平最小脉冲和低电平最小脉冲两种。

4.2.5 High Fanout Nets

High Fanout Nets 分析所有参与时序分析的路径中网表的扇出情况, 同时还会分析该网表的最差 Slack 和最大延时。

4.2.6 Route Congestions

Route Congestions 分析每个 Grid 上绕线的拥塞度。计算方法为已用绕线资源/可用绕线资源，包括长线、固定连线。

4.2.7 Timing Constraints Report

Timing Constraints Report 报告时序约束中所有时钟，包括衍生的时钟。

4.2.8 Clock Latency

Clock Latency 设置时钟延时，这将直接影响建立时间和保持时间的计算。

Clock Latency 包含 Clock Source Latency 及 Clock Network Latency。Clock Source Latency 是指时钟定义点之前时钟源到时钟定义点的传输时间。Clock Network Latency 是指时钟定义点之后时钟到达寄存器的时间。

4.2.9 IO Delay

IO Delay 约束用来完成 FPGA 端口延时的说明。

4.3 时序收敛步骤

当使用 Gowin 云源软件时，需要遵从一些总体的准则以更好地达到时序收敛的目的。

- 遵循 Gowin 的 RTL 编码规范。
- 运用合适的约束去指导综合和布局布线。
- 使用 Gowin 云源软件编译整个设计，如果综合时序出错，一般是 RTL 代码问题，用户需要优化 RTL 代码。
- 检查综合报告、布线布局报告、时序报告以及 Post PnR 报告，得到设计完整的时序信息。
- 尽量使用专用时钟资源，比如 GCLK 和 HCLK。
- 分析大扇出路径、关键路径以及大延迟路径。
- 如有必要运用设计规划约束，重复如上步骤直到满足时序。

4.3.1 前端时序收敛策略

- 使用时序或者面积作为优化目标。前者一般可以得到更好的性能，但是会增加资源使用率。如果用户的设计资源使用率很高，时序优化增加的额外资源开销会影响性能，比如局部拥塞和长走线，进而会带来时序问题。对于低速设计，综合一般应该以面积作为优化目标。
- 添加完整且合理的时序约束。时钟主频约束是最基本也是最重要的约束，如果没有任何主频约束，系统就会使用 200MHz 的默认值。当设计中包含有多个时钟时，需要添加 maxdeley 或者 multicycle 处理跨时钟域信号；对于管脚输入输出需要添加 set_input_delay 或 set_output_delay 约束。
- 仔细阅读综合时序报告。由于综合时序报告没有包含布局布线信息，因

此综合时序报告的结果往往比实际的结果要好，一般来说实际结果要差 1/3 到 1/2。

- 检查是否有过约束(Over Constraints)。有些时候用户会过约束以期得到更好的性能以及更大的裕度。但是这个方法不一定有效，原因在于过约束会增加资源开销，这些额外的资源开销会影响时序。
- 使用专用的 GSR 资源。如果设计中含有大扇出的复位或者置位信号，我们建议使用专用的 GSR 资源，这会降低布线拥塞，并提高布线效率；
- 使用输入输出管脚寄存器改善管脚时序。使用输入管脚寄存器可以改善输入建立时间；使用输出管脚寄存器可以改善时钟到输出时间。
- 使用内嵌可编程延迟单元改善输入保持时间。使用输入管脚寄存器可能带来保持时间问题，原因在于数据通道的时延较短。使用 HCLK 可以调整数据和时钟之间的关系以同时满足建立时间和保持时间。另外，Gowin 器件有内嵌可编程延时单元，通过在数据路径增加一个可控的延时用于补偿输入保持时间。
- 减少扇出以提高主频。对于关键路径选择性地使用 syn_maxfan 属性以减少扇出，减少扇出的代价是复制寄存器。
- 时钟使能尽量使用 GCLK 资源。时钟使能通常是一组驱动大量寄存器的大扇出信号，如果时钟使能使用普通绕线资源，有些时候会引入大延时，建议用户把时钟使能放在 GCLK 资源上。
- 使用独热状态机编码。对于高速设计，建议用户使用独热状态机编码，但是会增加资源利用率和功耗。
- 资源复用。资源复用会增加逻辑级数，并产生大延迟的路径。通常综合工具会对非关键路径进行资源复用，但是也有例外，用户需要检查关键路径以确保不会引入时序问题。
- 流水线和重定时。这个选项允许综合工具调整寄存器的前后位置以改善时序。

4.3.2 后端时序收敛策略

- 仔细阅读布局布线后的时序报告。特别需要关注工具给出的一些告警和错误信息、跨时钟域的分析、时钟主频以及逻辑级数；
- 使用 multicycle 或者 maxdelay 放松时序；
- 进行设计规划；
- 避免欠约束，比如没有时钟约束；没有管脚约束；管脚路径被屏蔽掉；内部产生的时钟；异步逻辑等等；
- 避免过约束，比如时钟主频高于实际值；不可识别的 multicycle 路径；冗余约束。

4.4 Gowin 器件架构对时序影响

了解 Gowin 器件的架构有助于帮助客户充分利用硬件性能，避免不必要的时序问题。

4.4.1 块存储器的布线差异

块存储器输出到顶部 CLS 延时和到底部 CLS 延时有差异。

4.4.2 DSP 的布线差异

TBD

4.4.3 查找表不同输入管脚延时差异

查找表 C、D 端输入延时小于 A、B 端输入延时。

4.4.4 CFU 时钟使能管脚延时差异

时钟使能端输入延时大于数据输入端。

