Gowin タイミング制約

ユーザーガイド

SUG940-1.8.2J, 2024-10-25

免責事項

当社は、GOWINSEMI Terms and Conditions of Sale(GOWINSEMI取引条件)に規定されている内容を除き、(明示的か又は黙示的かに拘わらず)いかなる保証もせず、また、知的財産権や材料の使用によりあなたのハードウェア、ソフトウェア、データ、又は財産が被った損害についても責任を負いません。当社は、事前の通知なく、いつでも本文書の内容を変更することができます。本文書を参照する何れの団体及び個人も、最新の文書やエラッタ(不具合情報)については、当社に問い合わせる必要があります。

バージョン履歴

日付	バージョン	説明
2020/06/09	1.0J	初版。
2020/09/01	1.1J	● 温度グレード(オートモーティブ：-grade a)を追加。 ● 基本クロックと派生クロックの連動機能を追加。
2021/06/16	1.2J	● ワイルドカードの説明を追加。 ● 図面を更新。
2021/11/02	1.3J	● ソフトウェアのバージョンを更新。 ● 図面およびその説明を更新。 ● 付録 A.タイミング制約構文仕様を更新。
2022/05/20	1.3.1J	一部の説明を更新。
2022/07/28	1.4J	● Create Clock に「-Add」オプションの説明を追加。 ● 遅延データモデルの温度-電圧の説明を追加。 ● tUnc と tSu の説明を追加。 ● ワイルドカードの「階層間で一致」機能の説明を追加。
2022/12/16	1.4.1J	● セクション「4.7.2 I/O 遅延制約」のオプション Add delay の説明を更新。
2023/03/31	1.4.2J	● セクション「4.7.2 I/O 遅延制約」の set_input_delay と set_output_delay の説明を更新。 ● セクション「5.1.4 Total Negative Slack Summary」の説明を更新。
2023/04/20	1.4.3J	図 4-8 New ウィンドウと図 4-9 Open ウィンドウ、およびその説明を追加。
2023/05/25	1.5J	● 例外制約の説明を更新。 ● ダミークロック DEFAULT_CLK の説明を更新。
2023/08/18	1.5.1J	● set_operation_conditions を set_operating_conditions を変更。 ● 最大周波数報告制約の説明を更新。
2023/11/30	1.6J	● クロックサイクルと周波数変換の説明を削除。 ● Process ウィンドウのスクリーンショットとその説明を削除。 ● セクション「A.2.2 set_output_delay」の説明を更新。
2024/02/02	1.7J	derive_clocks 制約を追加。
2024/06/28	1.8J	● 一部の説明を更新。 ● create_clock 構文における-add オプションの使用例を追加。

日付	バージョン	説明
2024/08/09	1.8.1J	「図 5-1 静的タイミング解析レポート」と「図 5-2 Timing Summaries」を更新。
2024/10/25	1.8.2J	「5.1.2 Clock Summary」における Clock Name のデフォルト表示ルールの説明を改善。

目次

図一覧

表一覧

1 本マニュアルについて

1.1 マニュアルの内容

このマニュアルでは、主に、タイミング制約エディタ(Timing Constraint Editor)の使用、制約の構文、静的タイミング解析レポート(以下、タイミングレポート)など、タイミング制約について説明します。

1.2 関連ドキュメント

GOWIN セミコンダクターの公式サイト www.gowinsemi.com/ja から、以下の関連ドキュメントがダウンロード、参考できます：Gowin ソフトウェア クイックスタートガイド(SUG918)。

1.3 用語、略語

表 1-1 に、本マニュアルで使用される用語、略語、及びその意味を示します。

表 1-1 用語、略語

用語、略語	正式名称	意味
GUI	Graphical User Interface	グラフィカル・ユーザー・インターフェース
MPW	Minimum Pulse Width	最小パルス幅
OSC	Oscillator	オシレータ
REG	Register	レジスタ
SDC	Synopsys Design Constraint	Synopsys 設計制約
STA	Static Timing Analysis	静的タイミング解析

1.4 テクニカル・サポートとフィードバック

GOWIN セミコンダクターは、包括的な技術サポートをご提供しています。使用に関するご質問、ご意見については、直接弊社までお問い合わせください。

ホームページ：www.gowinsemi.com/ja

E-mail：support@gowinsemi.com

2 概要

このマニュアルには、STA の概要、タイミング制約エディタ、およびタイミングレポートなどが含まれています。

STA の概要では、静的タイミング解析の基本について説明されます。Timing Constraint Editor は、SDC ファイルを作成および変更できるグラフィカル・ユーザー・インターフェース(GUI)ツールです。配置配線が成功すると、ユーザーのタイミング制約に従って STA レポートが生成されます。

主な機能

- 基本クロックと派生クロックの制約、ソース遅延と不確定値の制約、およびグループ制約などのクロック制約をサポートします。
- データポートの入出力遅延の制約をサポートします。
- マルチサイクル制約、最大および最小遅延制約、フォルスパス制約などの例外制約をサポートします。
- module 最大周波数制約、Grid 密集レベル制約などのタイミングレポート制約をサポートします。
- 効率的なネットリストユニット検索機能を提供し、正規表現一致をサポートします。
- タイミング制約エディタはフラットデザインを採用し、使いやすいです。

主な特徴

- レポートの内容は、標準の W3C XHTML 1.0 仕様に厳密に従います。
- レポートは、ブラウザーを使用して閲覧することができます。
- TXT テキスト形式のタイミングレポートをサポートします。
- ナビゲーションバー機能により、コンテンツをすばやく見つけることができます。

- タイミング制約エディタによって生成されたすべての制約文を完全に報告します。
- レポートの内容は明確で、読みやすいです。

3 STA の概要

3.1 概要

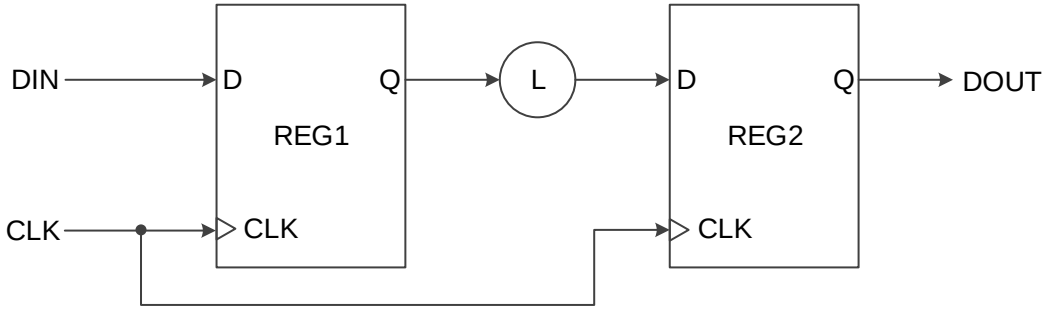
静的タイミング解析は、回路ネットリストのタイミングモデルに全面的な解析を行い、回路のタイミングパスの遅延を自動的に計算することにより、それがタイミング要件を満たしているかどうかを判断するものです。Gowin ソフトウェアは、設計のタイミングモデル回路を自動的に解析することができます。また、設計者が制約を追加し、その計算・解析を Gowin ソフトウェアで自動的に行うことも可能です。

以下は、解析プロセスに関する基本的なモデル、用語、コンセプトの紹介です。

3.2 タイミング解析の基本モデル

静的タイミング解析は、シーケンシャル・エレメントの開始から終了までのモデルのタイミング解析です。その基本モデルの参照図は図 3-1 に示すとおりです。レジスタ REG1 の D 端子のデータは、有効クロックエッジで Q 端子に同期され、論理回路を経由してレジスタ REG2 に到着します。レジスタ REG2 は、クロックの有効エッジでレジスタ REG1 が送信したデータをサンプリング(収集)します。静的タイミング解析は、REG2 が REG1 からのデータを正確にサンプリングしているか確認するものです。

図 3-1 タイミング解析の基本モデル



REG1 の有効クロックエッジは開始エッジ(launch edge)、REG2 の有効クロックエッジはラッチエッジ(latch edge)と呼ばれます。

3.3 タイミング解析の用語

以下は、タイミング解析モデルの基本的なシーケンシャル・セルの構成です。

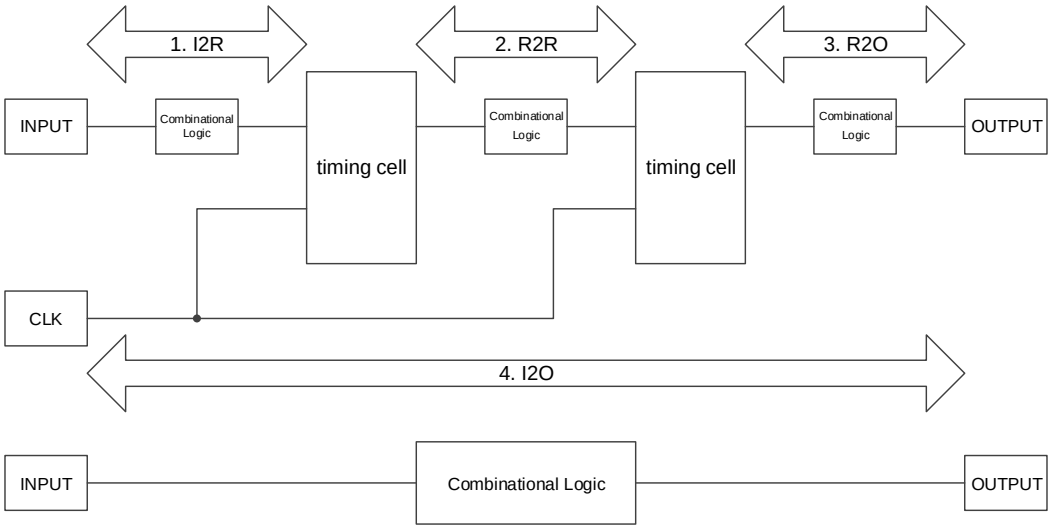
- Cells：LUT、DFF、MUX などの基本論理ユニット。
- Pins：Cells の入出力ポート。
- Ports：トップモジュールの入出力ポート。
- Nets：ネット。

3.4 タイミング解析のパス

通常、静的タイミング解析は 4 種類のパスを解析します。これらのパスは、始点と終点によって分類されています(図 3-2)。

- I2R：入力ポートからシーケンシャル・エレメント。
- R2R：シーケンシャル・エレメントからシーケンシャル・エレメント。
- R2O：シーケンシャル・エレメントから出力ポート。
- I2O：入力ポートから出力ポート。

図 3-2 STA の 4 種のタイミングパス



静的タイミング解析(STA)は、この 4 種類のパスを通じてデータ到着

時間(data arrival time)とデータ要求時間(data required time)を計算します。

データ到着時間(data arrival time)は、データパスの始点から終点までの時間を指します。データ要求時間(data required time)は、タイミングパス内のクロックの始点から終点までの時間を指します。データ到着時間(data arrival time)を計算する時、クロックパスにはクロックスキュー(clock skew)があります。クロックスキュー(clock skew)は、クロックが各シーケンシャル・エレメントのクロックポートに到着する時間差を指します。

3.5 一般的なタイミングチェック

静的タイミング解析では、通常以下の 3 つの項目がチェックされ、配置配線プロセスに関するアドバイスが提供されます。これにより、ユーザーのタイミング要件をよりよく満たすことができるようになります。

3.5.1 セットアップ時間(setup time)とホールド時間(hold time)のチェック

- セットアップ時間：有効クロックエッジの前にデータが保持されていなければならない最小時間。この時間が満たされていない場合、データを正しくサンプリングできません。
- ホールド時間：有効クロックエッジの後にデータが保持されていなければならない最小時間。この時間が満たされていない場合、データを正しくサンプリングできません。

3.5.2 リカバリ時間(recovery time)とリムーバル時間(removal time)のチェック

- リカバリ時間：有効クロックエッジの前に非同期クリア/セット/リセット信号が安定していなければならない最小時間。この時間が満たされていなければ、フリップフロップは正常な動作状態に入ることができません。
- リムーバル時間：有効クロックエッジの後に非同期クリア/セット/リセット信号が安定していなければならない最小時間。この時間が満たされていなければ、フリップフロップは正常な動作状態に入ることができません。

3.5.3 最小クロックパルス(MPW)のチェック

最小クロックパルス(MPW)：デバイス内部のフリップフロップ(例えば、DFF)が認識できる High 及び Low レベルの最小幅。幅がこの幅よりも小さい場合、クロックを正常に認識できなくなります。

4 タイミング制約エディタ

4.1 概要

タイミング制約エディタは、複数種類のタイミングコマンドをサポートします。これにはクロック、入出力、パスなどの制約と、クロックレポートなどのコマンドが含まれます。ユーザーはその GUI でタイミング制約を追加できます。タイミング制約エディタの使用例については、『Gowin ソフトウェア クイックスタートガイド(SUG918)』を参照してください。

4.2 タイミング制約エディタの起動

タイミング制約エディタは、単独で使用するか、またはプロジェクト合成後に使用することができます。

単独で使用する場合は、「Tools> Timing Constraints Editor」をクリックして起動します。合成後に使用する場合は、Gowin ソフトウェアの Process ウィンドウで Synthesize を正常に実行した後、「Process > Timing Constraints Editor」をダブルクリックして起動します。ネットリストファイルとタイミング制約ファイルは、タイミング制約エディタに自動的にロードされます。プロジェクトにタイミング制約ファイルがない場合は、自動的に作成されます。

4.3 制約ファイルの新規作成、オープン、および追加

4.3.1 制約ファイルの新規作成

以下は、制約ファイルの新規作成手順です。

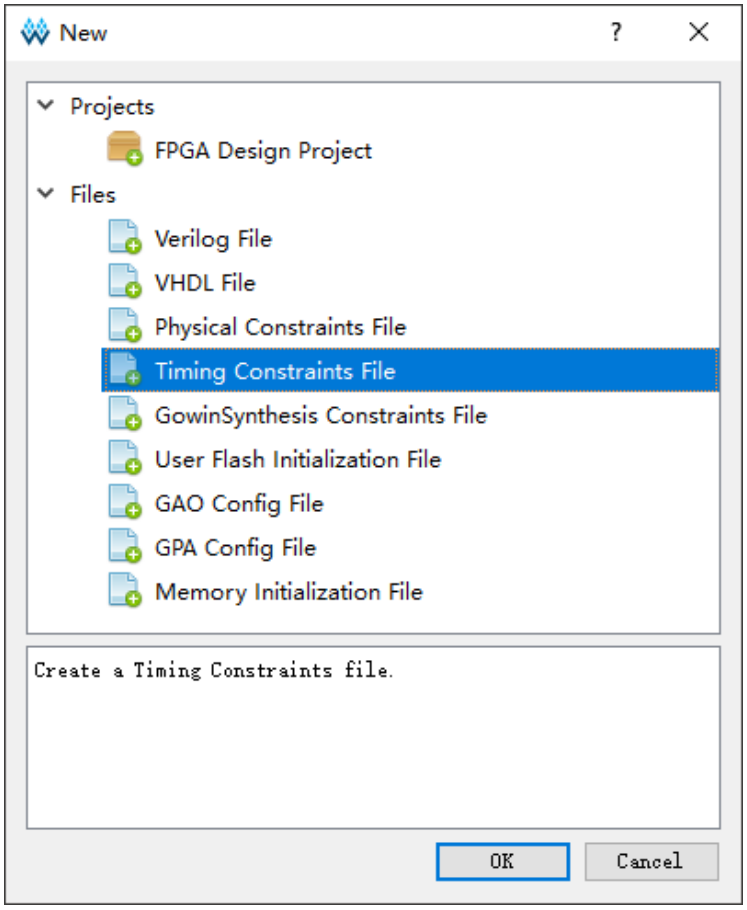
1. 「File>New」メニューをクリックすると、ファイル新規作成のダイアログボックスが開きます。
2. 「Timing Constraints File」オプションを選択します(図 4-1)。

注記：

または、以下の方法で制約ファイル新規作成ダイアログボックスを開きます。

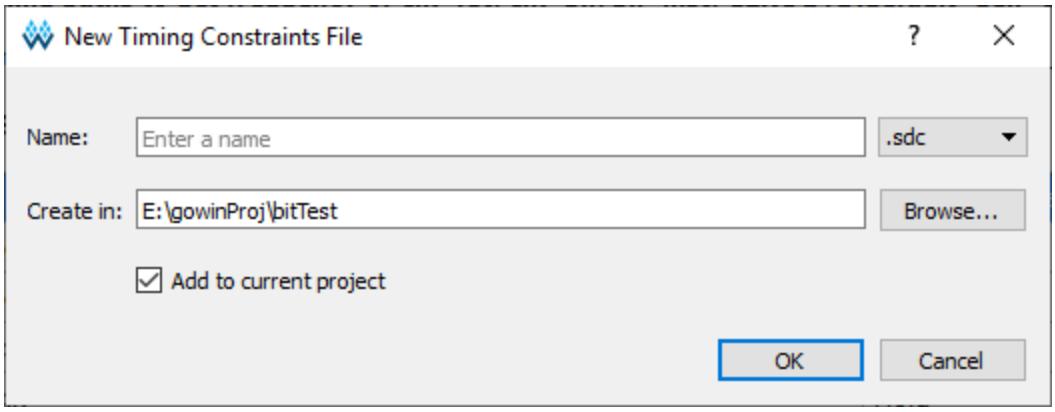
- ツールバーの「New」アイコンをクリックします。
- ショートカットキーCtrl + N を使用します。

図 4-1 制約ファイル新規作成ダイアログボックスを開く



3. 「OK」をクリックすると、タイミング制約ファイル新規作成ダイアログボックスがポップアップします(図 4-2)。

図 4-2 タイミング制約ファイルの新規作成



4. ファイル名を入力して作成ディレクトリを選択し、「OK」をクリック

すると、タイミング制約ファイルは作成されてプロジェクトに自動的にロードされます。

- Name：新規作成されたタイミング制約ファイルの名前。.sdc ファイルがサポートされます。
- Create in：「Browse」ボタンをクリックして新しい制約ファイルの保存場所を選択します。デフォルトのパスはプロジェクトディレクトリの src フォルダです。
- Add to current project：このオプションを選択すると、制約ファイルはプロジェクトに自動追加されます。デフォルトではチェックされています。

4.3.2 制約ファイルを開く

以下は、制約ファイルを開く手順です。

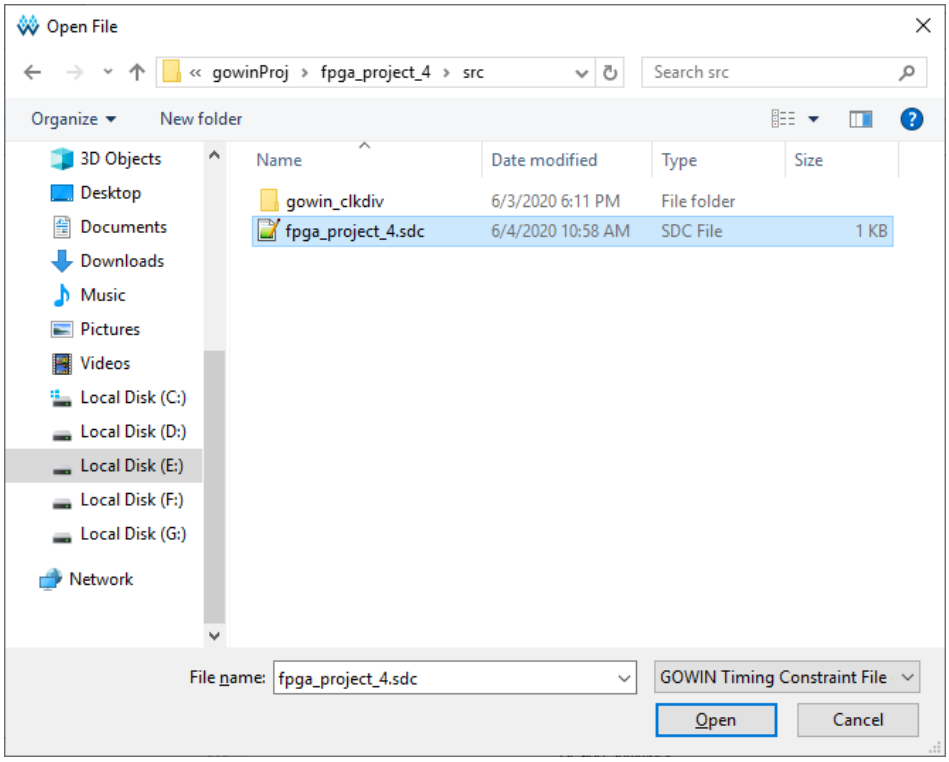
1. IDE で「File>Open」をクリックします。
2. 「Open File」ダイアログボックスを開きます(図 4-3)。

注記：

または、以下の方法を使用します。

- ツールバーの「Open」アイコンをクリックします。
- ショートカットキーCtrl + O を使用します。

図 4-3 タイミング制約ファイルを開く



3. タイミング制約ファイルが所在するディレクトリでファイルを選択して「Open」をクリックします。sdc ファイルがサポートされます。

注記：

ファイルを開くことにより、ファイルがプロジェクトに自動的にロードされることはありません。

4.4.3 制約ファイルの追加

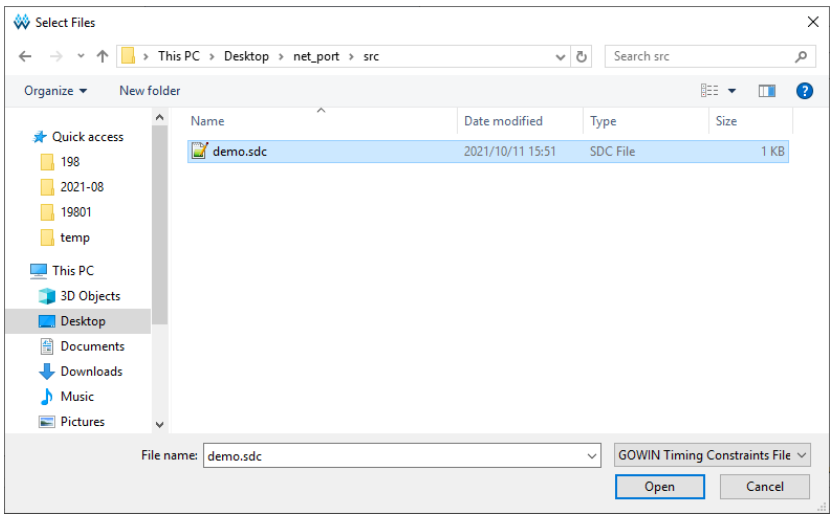
以下は、制約ファイルを追加する手順です。

1. IDE の Design ウィンドウで右クリックして、「Add Files」を選択します。
2. ポップアップする「Select Files」ダイアログボックスから「.sdc」ファイルタイプを選択します(図 4-4)。
3. 制約ファイルを選択したら、「Open」をクリックしてプロジェクトに追加します。

注記：

複数のファイルを追加した場合、1 つだけ有効です。

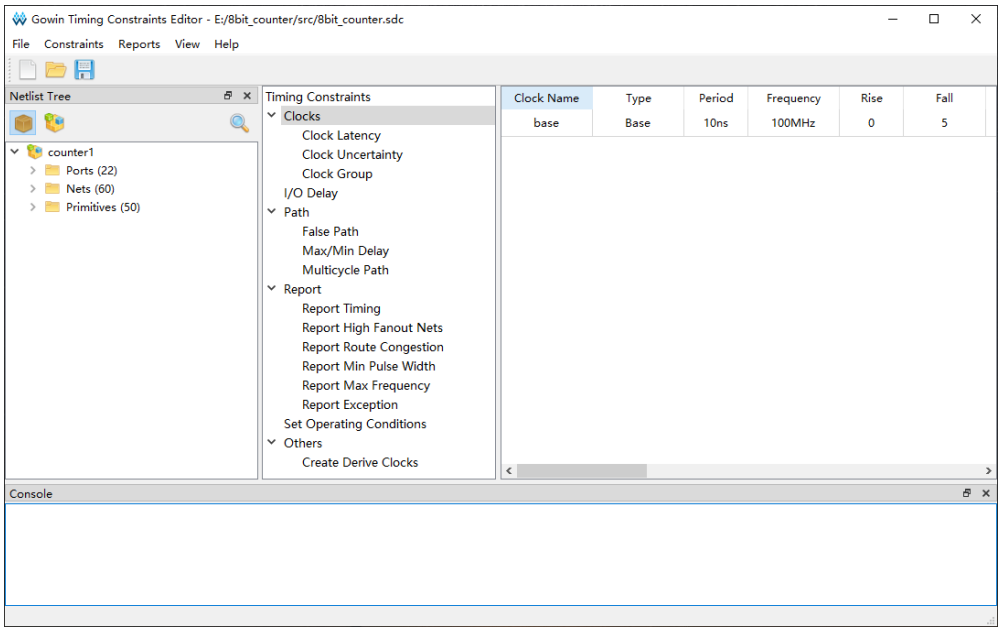
図 4-4 タイミング制約ファイルの追加



4.4 タイミング制約エディタの GUI

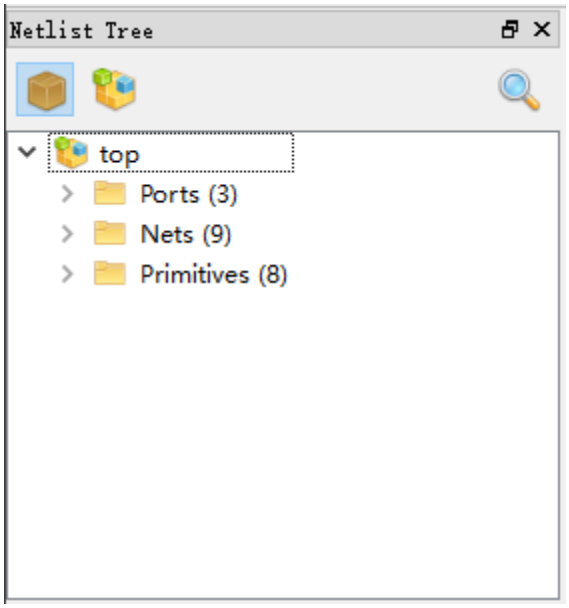
図 4-5 は、制約ファイルを開いた後のタイミング制約エディタの GUI です。

図 4-5 タイミング制約エディタの GUI



メインウィンドウの左上隅は、Netlist Tree ウィンドウです。(図 4-6)。

図 4-6 Netlist Tree ウィンドウ



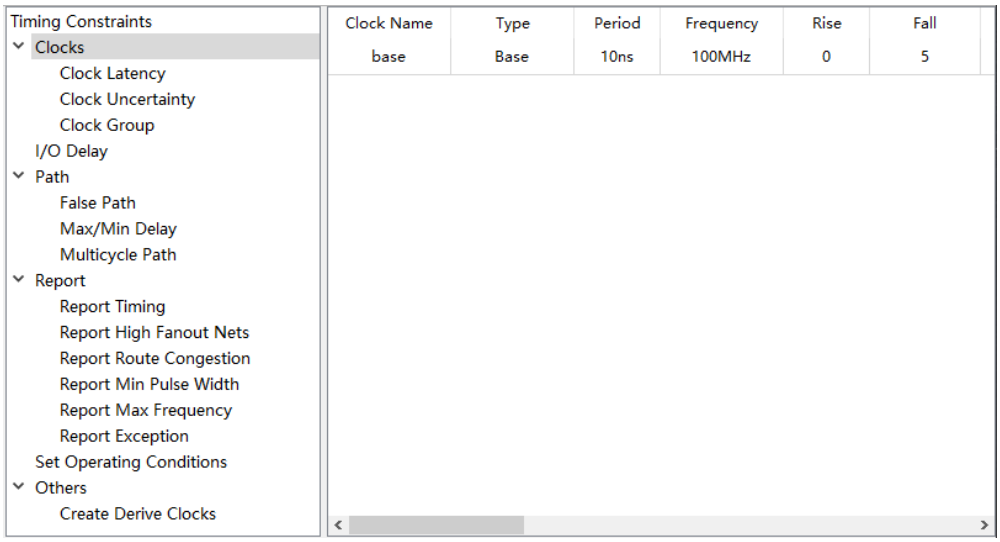
Netlist Tree ウィンドウには、現在のネットリストファイルの Top Module、I/O Ports、Nets、Primitives が含まれています。

- 「  」: flatten リストを確認します。
- 「  」: hierarchy リストを確認します。

メインウィンドウの中央及び右側のエリアは、制約編集エリアです

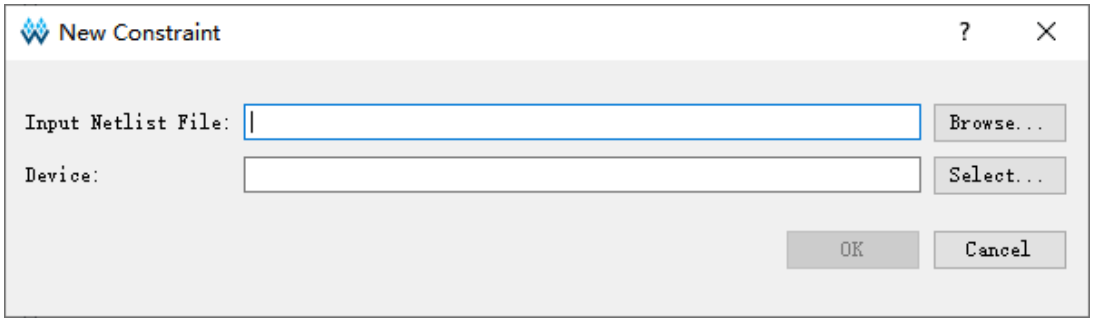
(図 4-7)。そのうち、左側のリストはタイミング制約タイプのディレクトリで、右側は表の編集エリアです。タイプのディレクトリで任意の制約タイプをクリックすると、表の編集エリアで設定済みの制約編集リストが表示されます。

図 4-7 制約編集ウィンドウ



メインウィンドウの上部のツールバーには、「New」、「Open」、「Save」というボタンがあります。New ウィンドウでは、ネットリストファイル「Input Netlist File」とデバイス情報「Device」を選択することができます。

図 4-8 New ウィンドウ



Open ウィンドウでは、ネットリストファイル「Input Netlist File」、制約ファイル「Constraint File」、およびデバイス情報「Device」を選択することができます。

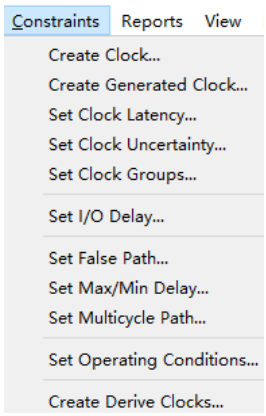
図 4-9 Open ウィンドウ

4.5 タイミング制約ウィンドウを開く

タイミング制約ウィンドウを開く方法は 2 つあります。

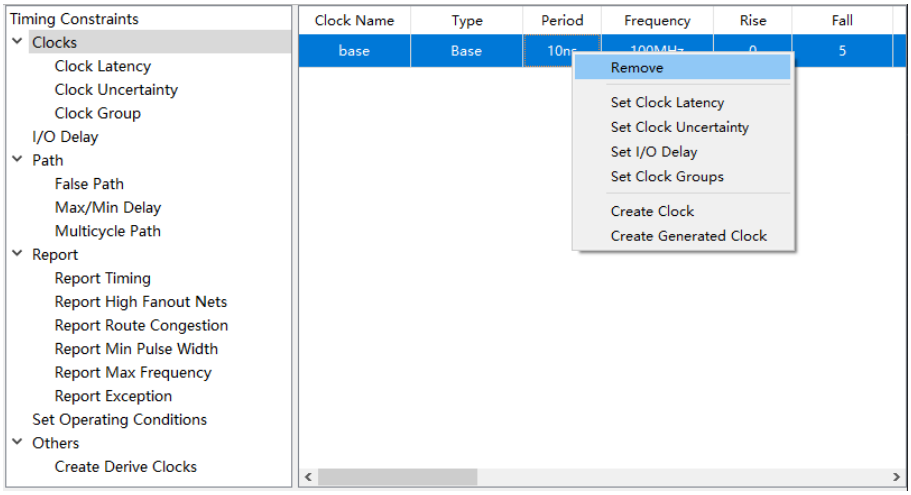
1. メニューバーで「Constraints」をクリックし、プルダウンリストからタイミング制約コマンドを選択して対応するタイミング制約ウィンドウを開きます(図 4-10)。

図 4-10 メニューバーからタイミング制約ウィンドウを開く



2. タイミング制約エディタの右側にある表で右クリックしてメニューからさまざまなタイミング制約コマンドを選択して対応するタイミング制約ウィンドウを開きます(図 4-11)。

図 4-11 右クリックしてタイミング制約ウィンドウを開く



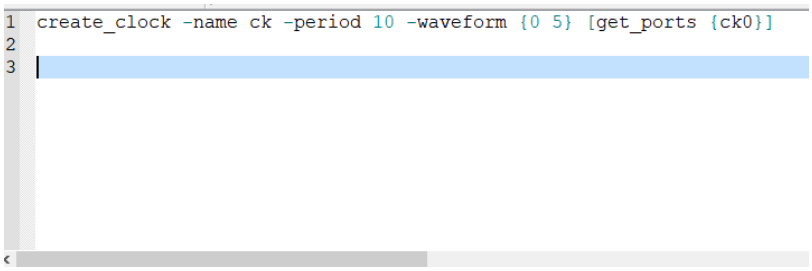
4.6 SDC ファイルの編集

Gowin ソフトウェアでは、プロジェクトの SDC ファイルを読み出し、テキストエディタで制約を簡単に手動変更することができます(図 4-12)。

SDC ファイルの解析はワイルドカードをサポートします。現在「*」と「?」の 2 つのワイルドカードがサポートされています。「*」は 0 文字以上の一致を実現し、「?」は 1 文字の一致を実現します。

SDC ファイルは、単一行コメントと複数行コメントをサポートします。単一行コメントの場合は「//」または「#」を使用し、複数行コメントの場合は「/* */」を使用します。

図 4-12 SDC ファイルの編集



4.7 タイミング制約の作成

このセクションでは、タイミング制約エディタを使用してタイミング制約を作成する方法を解説します。作成されたタイミング制約は、プロジェクトの SDC ファイルに書き込まれます。タイミング制約の構文の詳細については、付録 A を参照してください。

4.7.1 クロック制約

Create Clock

- クロックの名前、周期、周波数、立ち上がりエッジ、立ち下がりエッジ、およびクロックのオブジェクトなどのパラメータを指定できます。
- Gowin ソフトウェアは、複数のクロックドメインの確立をサポートし、クロスクロックドメイン解析をサポートます。

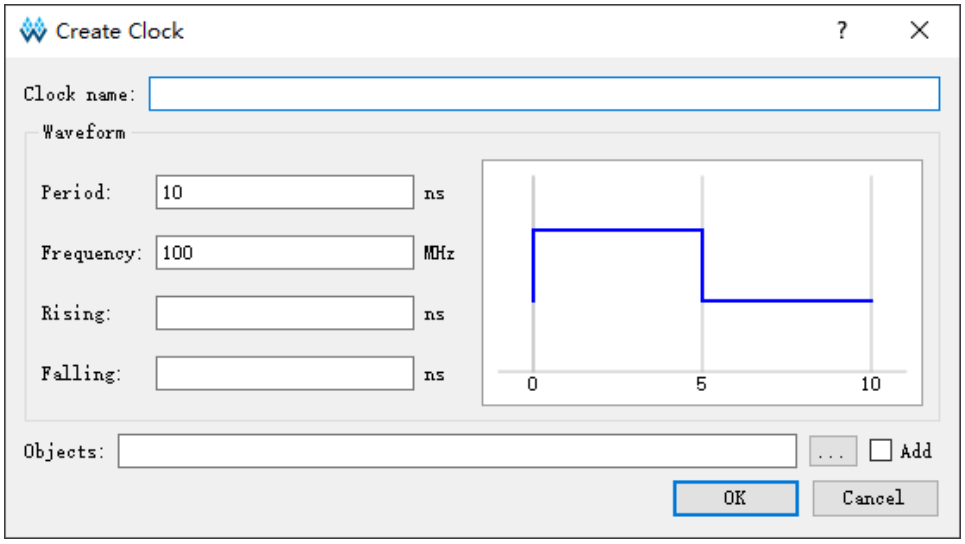
create_clock により、ユーザーデザインのための基本クロックを作成できます。

Clock 制約を追加するには、2 つの方法があります。

1. 「Constraints」メニューで Clock 制約を追加します。

   a). 「Constraints > Create Clock…」を選択すると「Create Clock」ダイアログボックスがポップアップします(図 4-13)。

図 4-13 基本クロックの作成



   b). 「Clock Name」、「Waveform」、「Objects」などのクロック情報を入力します。

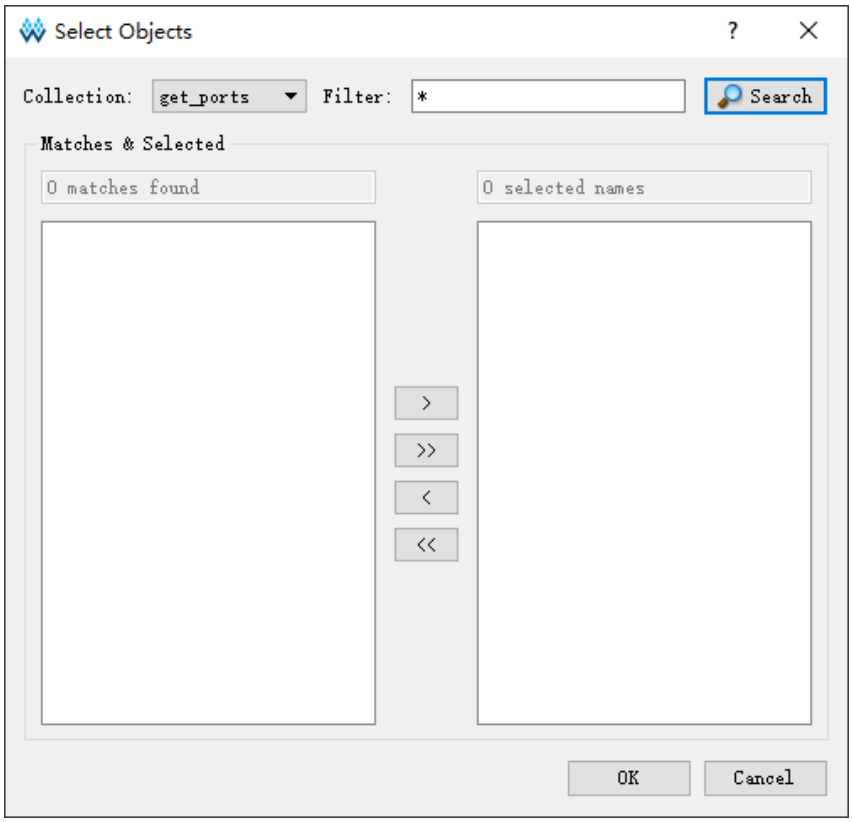
   - Clock Name：クロック名。文字またはアンダースコアで始まることをサポートします。
   - Period：周期、デフォルトは 10。0 より大きい浮動小数点数型、小数点以下 3 桁、単位は ns。
   - Frequency：周波数、デフォルトは 100。0 より大きい浮動小数点数型、小数点以下 3 桁、単位は MHz。
   - Rising：立ち上がりエッジ時点。0 より大きい浮動小数点数

型、小数点以下 3 桁、単位は ns。

- Falling：立ち下がりエッジ時点。0 より大きい浮動小数点数型、小数点以下 3 桁、単位は ns。
- Objects：オブジェクトを指定します。「...」をクリックして選択します。
- Add：同じソースに複数のクロックを追加する場合は、チェックする必要があります。

c). Objects 右側の「...」ボタンをクリックすると、「Select Objects」ダイアログボックスがポップアップします(図 4-14)。

図 4-14 オブジェクトの選択



d). 図 4-14 では、「Collection」は検索されるコレクションタイプを指定します。「Filter」はフィルターです。「Search」をクリックすると、すべての該当する結果トは左側に表示されます。「>」ボタンをクリックすると、選択されたアイテムは左側のリストから右側のリストに追加されます。「>>」をクリックすると、すべてのアイテムは右側のリストに追加されます。「<」をクリックすると、右側の選択されたアイテムが削除されます。「<<」をクリックすると、右側のすべてのアイテムが削除されます。

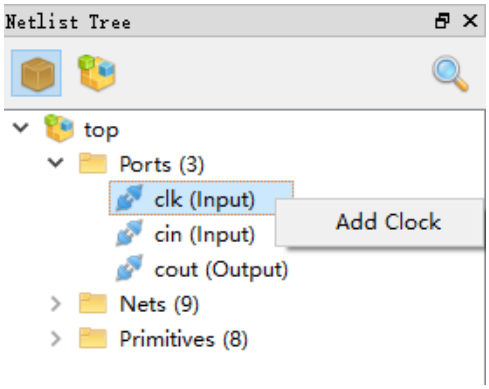
e). 「OK」をクリックして Objects の追加を完了します。

2. Netlist Tree で Clock 制約を追加します。

a). Netlist Tree で I/O Port または Net を選択します。

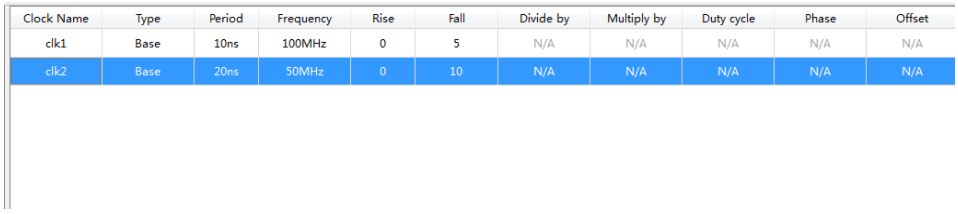
b). 右クリックして「Add Clock」を選択し、クロックを追加します(図 4-15)。

図 4-15 クロックの追加



クロックの作成完了後、Clock リストに対応する制約が追加されます(図 4-16)。

図 4-16 クロックリスト

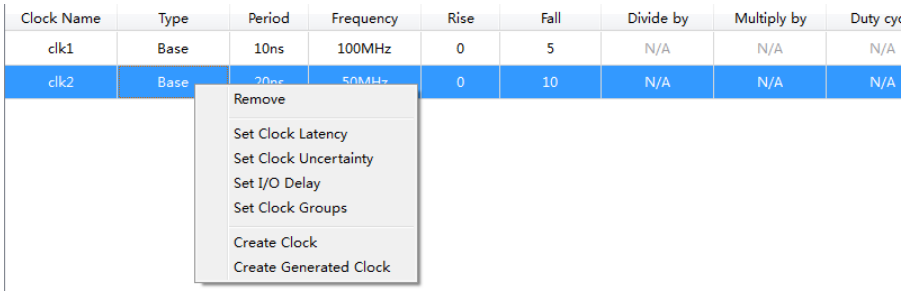


Clock Name	Type	Period	Frequency	Rise	Fall	Divide by	Multiply by	Duty cycle	Phase	Offset
clk1	Base	10ns	100MHz	0	5	N/A	N/A	N/A	N/A	N/A
clk2	Base	20ns	50MHz	0	10	N/A	N/A	N/A	N/A	N/A

このリストでは、以下の操作を行うことができます。

- Clock の編集。「Clocks」リストの対応する制約をダブルクリックすると Clock の編集ダイアログボックスが開くので、ダイアログで Clock 情報を編集できます。
- Clock の削除。リストでその Clock を右クリックして「Remove」を選択します。
- Clock を右クリックしてその Clock の Clock Latency、Clock Uncertainty、I/O Delay 情報を素早く設定できます(図 4-17)。

図 4-17 クロックリストの右クリック項目



注記：

- 制約が PLL の構成と矛盾する場合、Create Clock によって作成された制約が優先されます。配置配線中に警告メッセージが表示されます。
- Create Clock は、ダミークロックの作成をサポートしていません。

Create Generated Clock

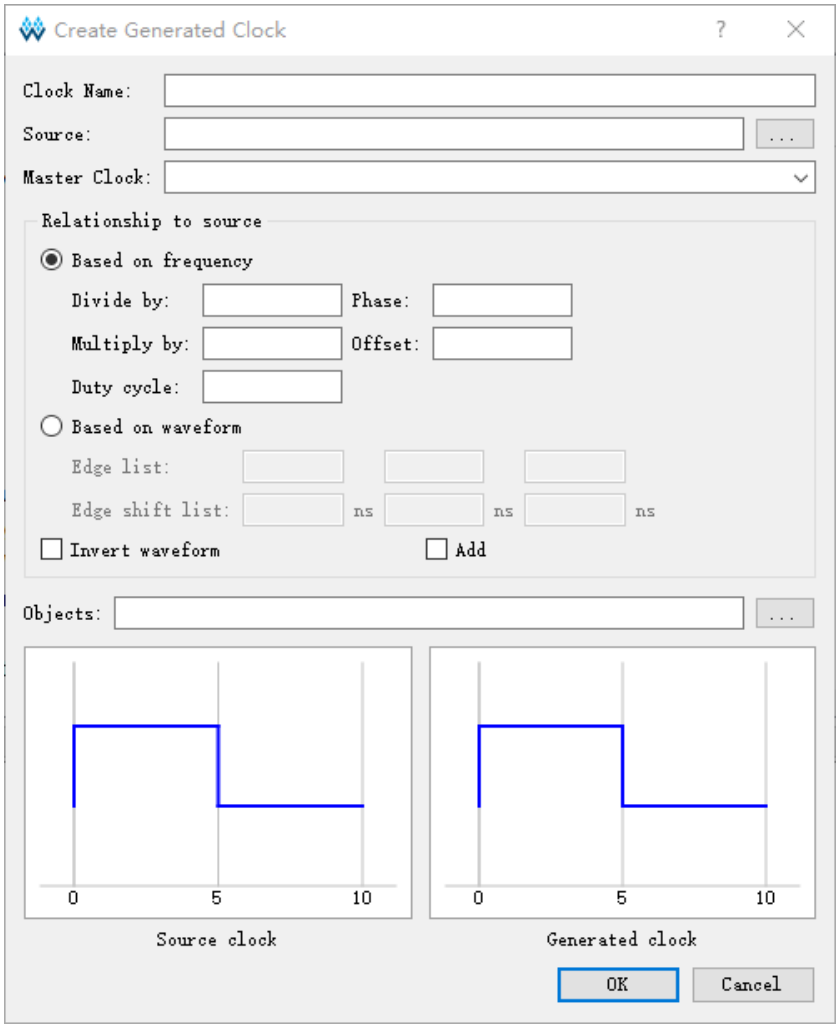
- 基本クロックに基づいた派生クロックを作成します。
- この制約により、基本クロックに基づいて周波数分割、周波数逓倍、位相シフト、デューティサイクル調整などの操作を実行することで派生クロックを作成することができます。

派生クロックは、基本クロックに基づいて作成する必要があります。ユーザーデザインの任意のノードで作成できます。実際のアプリケーションでは、通常、PLL や CLKDIV などのハードコアの出力ポートに作用します。ユーザーがデザインで PLL を使用する場合、基本クロックを作成した後、Objects が PLL.CLKOUT、Source が基本クロックである派生クロックを作成できます。作成された派生クロックは自動的に基本クロックと連動します。基本クロックの属性が変更されると、派生クロックもそれに応じて自動的に変更されます。

以下の 2 つの方法で派生クロックを新規作成できます。

1. 「Constraints」メニューで作成します。
   a). 「Constraints」メニューで「Create Generated Clock」を選択すると、「CreateGenerated Clock」ダイアログボックスがポップアップします(図 4-18)。
   - Clock Name：クロック名。文字またはアンダースコアで始まることをサポートします。
   - Source：派生クロックのソース。右側の「…」をクリックして選択します。
   - Master Clock：ソースに作用するクロック。右側の「⌄」をクリックして選択できます。

図 4-18 派生クロック制約の作成



b). Source の右側にある「...」をクリックして、クロックのソースを選択します。「Source」に関連付けられたクロックは、「Master Clock」リストに自動的に追加されます。次に「Master Clock」を選択します。「Master Clock」に複数のクロックがある場合、1 つのみを選択できます。

c). 「Relationship to source」では、」Base on frequency」の場合は、現在作成されている派生クロックに対して周波数分割、逓倍、オフセット、デューティサイクル、および位相の調整などを実行できます。」Base on waveform」の場合は、Edge list および Edge shift list を併用することにより、派生クロックのエッジ調整を実現できます。

- Divide by：分周値。正の整数。
- Phase：位相。浮動小数点数型、小数点以下 3 桁。負の数の場合は左にシフトされ、正の数の場合は右にシフトされます。

- Multiply by：逓倍値。正の整数。
- Offset：オフセット。浮動小数点数型、小数点以下 3 桁。負の数の場合は左にシフトされ、正の数の場合は右にシフトされます。
- Duty cycle：デューティサイクル。浮動小数点数型、小数点以下 3 桁。100 以下。
- Edge list：順番に増加する正の整数。
- Edge shift list：浮動小数点数型、小数点以下 3 桁。

d). 「Invert waveform」はクロックの反転を実現します。「Add」はさらなる追加を実現できます。これは STA の際にも有効です。

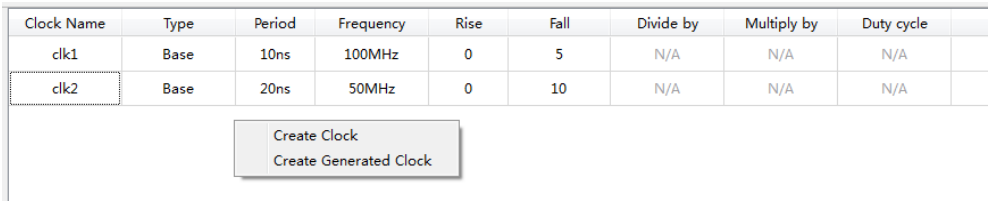
e). Objects の右側にある「  」ボタンをクリックし、「Select Objects」ダイアログボックスがポップアップします。オブジェクトを選択します。

注記：

- Clock がない Source を選択した場合、Source を再選択する必要がありません。
- 制約が PLL の構成と矛盾する場合、Create Generated Clock によって作成された制約が優先され、配置配線中にも警告メッセージが表示されます。

2. Clocks リストで Generated Clock を作成します。Clocks リストの空白で右クリックし、「Create Generated Clock」を選択して Generated Clock を作成します(図 4-19)。

図 4-19 Create Generated Clock を選択



追加後、テーブルの編集エリアに新規作成された制約が追加されます。

このリストでは、以下の操作を行うことができます。

- Generated Clock 制約の編集。「Clocks」リストの対応する制約をダブルクリックすると Generated Clock の編集ダイアログボックスが開くので、ダイアログで Generated Clock 情報を編集できます。
- Generated Clock の削除。表編集エリアでこの C l ock を選択し、右クリックで「Remove」を選択します。

Set Clock Latency

- クロック信号がデバイスのクロックポートに到着する前の遅延を設定

するために使用され、パラメータを選択することで、クロックの立ち上がりエッジ/立ち下がりエッジがアクセスポイントに到着する最大/最小遅延をそれぞれ正確に設定できます。

- クロック遅延にはネットワーク(network)遅延とソース(source)遅延の 2 種類があります。
  - ネットワーク(network)遅延はデバイス内部のクロックパスの遅延です。
  - ソース(source)遅延はデバイス外部のクロックパスの遅延です。
- クロックのネットワーク(network)遅延は Gowin ソフトウェアにより自動的に計算されるため、ユーザーはソース(source)遅延を設定するのみです。

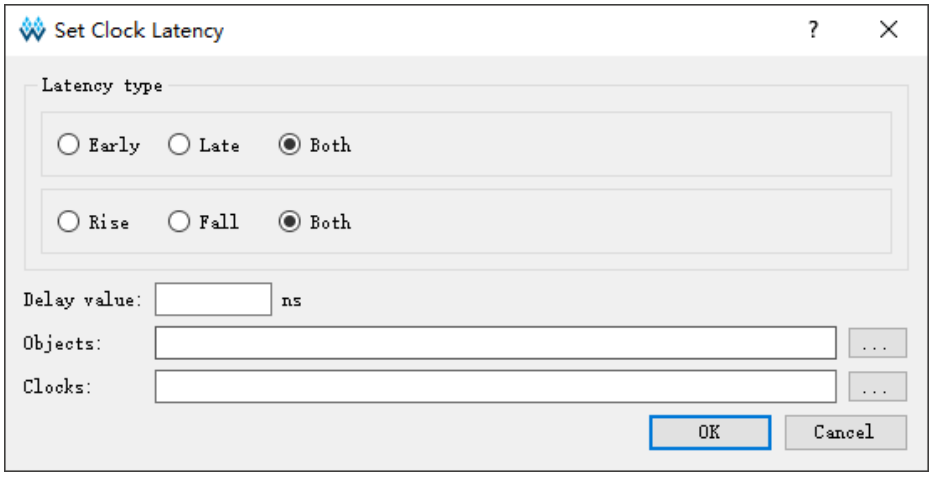
クロックソース(外部水晶発振器など)からデバイスのクロックポートまでのクロック信号の遅延は、クロックのソース遅延と呼ばれます。この遅延値は、Gowin ソフトウェアでは自動的に取得できず、デフォルトは 0ns です。ユーザーがソース遅延が 2ns であることを知っている場合、Delay Value を 2ns に構成することができます。Gowin ソフトウェアは、タイミング解析を実行するときにこの 2ns を自動的に計算に入れ、タイミングレポートでの Setup、Hold レポートの Tcl データに反映します。

以下の 2 つの方法で Clock Latency 制約を新規作成できます。

1. 「Constraints」メニューで Clock Latency 制約を新規作成します。
   「Constraints」メニューで「Set Clock Latency」を選択すると、「Set Clock Latency」ダイアログボックスがポップアップします(図 4-20)。Latency 情報を入力し、「OK」をクリックして制約を保存します。

- Early と Late：それぞれ最小遅延と最大遅延を示します。
- Rise と Fall：それぞれ立ち上がりエッジ有効と立ち下がり有効を示します。Both は両者有効を示します。
- Delay value：クロックの遅延値。浮動小数点数型、小数点以下 3 桁、単位は ns。
- Objects：右側の「…」をクリックして選択します。クロックの入力ポートまたはクロックを指定します。
- Clocks：右側の「…」をクリックして選択します。対象クロックを示します。

図 4-20 クロック遅延の設定



2. Clocks リストから Clock Latency 制約を新規作成します。

Clocks リストで Clock を右クリックして Set Clock Latency を選択し、この Clock の Latency 情報を設定します。Objects は自動的にこのクロックとして指定されます。

Set Clock Uncertainty

- クロック伝送の解析に用いる、クロックのばらつきまたはオフセットを設定します。
- setup と hold に対してそれぞればらつきを設定できます。また、クロックの立ち上がりエッジと立ち下がりエッジの伝送に対してそれぞればらつきを設定することもできます。
- ユーザーは、この制約を通じてクロックジッタ(jitter)、ペシミズム(pessimism)などを Gowin ソフトウェアに通知することでタイミング計算に影響を与えることができます。

時間とともにクロックのばらつきが生じないことが理想的ですが、通常、クロックのばらつきは避けられません。Gowin ソフトウェアは、デフォルトでばらつき値を計算に入れます。ユーザーは、実際のハードウェアの使用環境に応じて、より適切なばらつき値を設定することもできます。例えば、デバイスが強い磁気環境で動作し、ユーザーがばらつき値が 0.2 ns であることを知っている場合、Uncertainty を 0.2 ns に設定します。その結果については、Setup、Hold レポートの tUnc を参照してください。

以下は、Clock Uncertainty の新規作成の手順です。

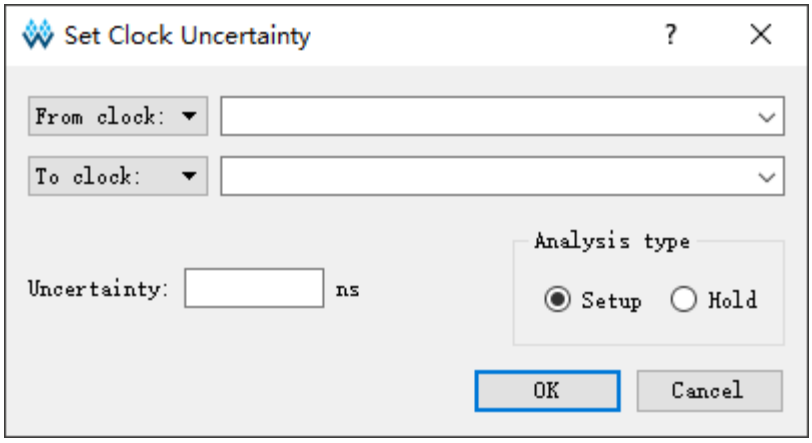
1. 「Constraints」メニューで「Set Clock Uncertainty」を選択すると、「Set Clock Uncertainty」ダイアログボックスがポップアップします(図 4-21)。

- From clock：送信側のクロックを示します。右側の「˅」をクリック

して選択します。

- To clock：受信側のクロックを示します。右側の「⌄」をクリックして選択します。
- Uncertainty：浮動小数点数型、小数点以下 3 桁、単位は ns。クロックのばらつき値を設定します。
- Analysis type：解析のタイプを示します。

図 4-21 ばらつきの設定



2. 左側のプルダウンリストで From のタイプ(From clock、Rise from、Fall from)と To のタイプ(To clock、Rise to、Fall to)を選択し、右側のプルダウンリストで現在のすべての作成済み Clock からオブジェクトの Clock を選択します。
3. 情報の入力が完了後、「OK」をクリックして制約を保存すると、Uncertainty の追加が完了します。

Set Clock Group

- 各クロック間の関係を指定します。
- デフォルトではグループメンバー間は関係しており、各グループ間は関係していません。
- デフォルトでは、Gowin ソフトウェアは、デザイン内のすべてのクロックが同じグループに属し、すべて関連していると想定しています。

この制約は通常、相互に排他的または非同期のクロックの制約に使用されます。たとえば、デザインには異なる周波数の 2 つのクロック CLK1 と CLK2 があります。この 2 つのクロックは 2:1 マルチプレクサを介してシーケンシャルロジックを駆動するので、相互に排他的になります。この場合、ユーザーはこの制約を利用して、CLK1 および CLK2 を 2 つの異なるグループに制約して無関係なタイミング解析を実行できます。

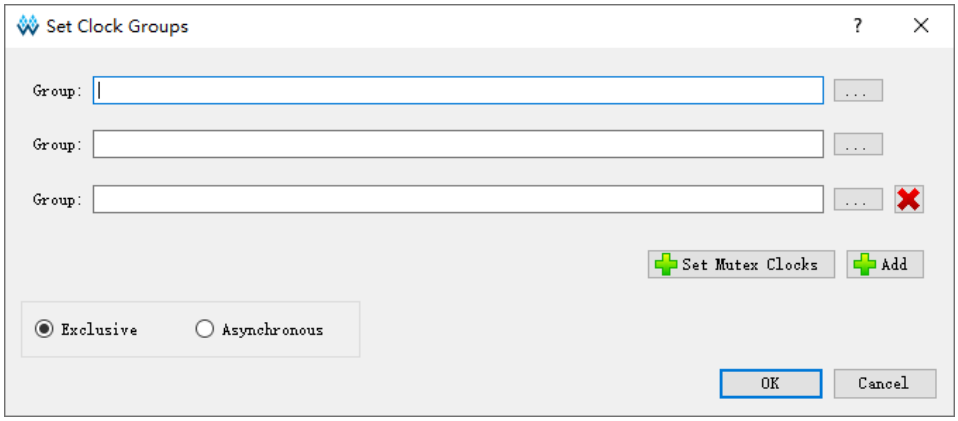
この制約文を使用して、クロック間の関係を厳密に指定し、非同期または相互に排他的なクロックに対してクロックグループ制約をかけることをお勧めします。

新しい Clock Group を作成する方法は次のとおりです。

1. 「Constraints」メニューで「Set Clock Groups」を選択すると「Set Clock Groups」ダイアログボックスがポップアップします(図 4-22)。

- Group：右側の「...」をクリックして選択します。グループ内のクロックを示し、少なくとも 1 つのクロックを指定する必要があります。
- Set Mutex Clocks：複数のクロックグループを一度に設定するために使用されます。
- Add：Group 項目をもう 1 つ追加します。
- Exclusive は、クロックが相互に排他的であり、クロックが同時に有効ではないことを示します。たとえば、Clock0 と Clock1 は MUX2 を経由したあと、1 つのクロックのみが出力されます。
- Asynchronous は、クロックが非同期であり、クロックのクロックソースが異なることを示します。

図 4-22 クロックグループの設定



2. 「...」ボタンをクリックし、Group の Clock を選択します。追加した Group を削除したい場合、対応する項目の右にある「✖」ボタンをクリックします。
3. 「OK」をクリックして制約を保存します。

注記：

オプション「Exclusive」と「Asynchronous」は、同じ効果を持ちます。

4.7.2 I/O 遅延制約

set_input_delay

データ入力の遅延値を設定し、データ到着とクロック到着の間の時間関係を分析します。ユーザーは適切な入力遅延値を設定する必要があります。ソフトウェアはこの設定された遅延値に従ってスラックを分析しま

す。

注記：

Gowin ソフトウェアによって生成されたタイミングレポートでは、入力遅延タイプは「tIn」です。

set_output_delay

データ出力の遅延値を設定し、データ出力とクロック出力の間の時間関係を分析します。ユーザーは適切な出力遅延値を設定する必要があります。ソフトウェアはこの設定された遅延値に従ってスラックを分析します。

注記：

Gowin ソフトウェアのタイミングレポートで、出力遅延のタイプは「tOut」です。

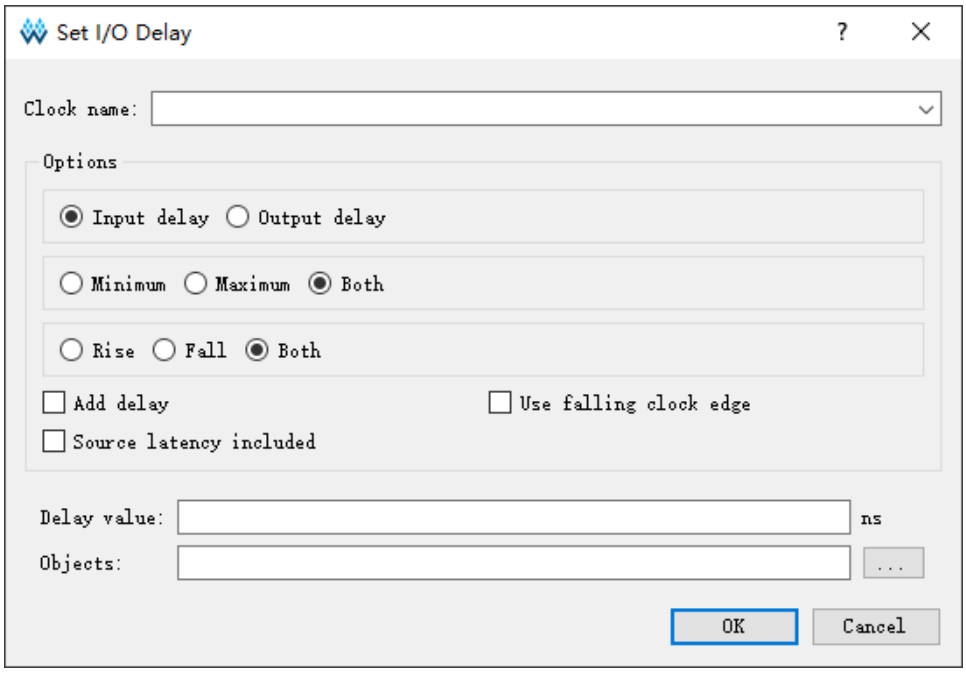
以下は、I/O Delay 制約の新規作成方法です。

1. 「Constraints」メニューで「Set I/O Delay」を選択すると「Set I/O Delay」ダイアログボックスがポップアップします(図 4-23)。

- Clock name は、I/O に関連付けられたクロックの名前を示します。これは既存のクロックである必要があります。右側の「」をクリックして選択します。
- Options では、遅延タイプ、最大および最小遅延、クロックエッジなどを構成することができます。
- Input delay、Output delay：それぞれ入力遅延および出力遅延で、相互に排他的です。
- Minimum、Maximum：I/O の最小遅延および最大遅延です。Both は、2 つの遅延値が同じであることを意味します
- Rise、Fall：それぞれ立ち上がりエッジ有効と立ち下がり有効を指定します。Both は両者有効を示します。
- Delay value：I/O の遅延値を設定します。浮動小数点数型、小数点以下 3 桁、単位は ns。負の場合は早期到着を意味し、正の場合は遅延到着を意味します。
- Objects：入出力ポートを指定します。右側の「」ボタンをクリックして選択してください。
- Add delay：同じポートに遅延値を追加するために使用されます。同じポートに複数の遅延値が存在する場合、ソフトウェアは Setup 解析に最大値を、Hold 解析に最小値を選択します。このオプションが指定されていない場合、同じポートの同じ制約が上書きされます。
- Use falling clock edge：チェックされている場合、関連クロックの立ち下がりエッジに関連していることを示します。デフォルトは立ち上がりエッジに関連しています。

- Source Latency include：チェックされている場合、設定された遅延値にクロックの遅延値がすでに含まれていることを意味します。チェックされていない場合、Gowin ソフトウェアは計算にクロックの遅延値を入れます。

図 4-23 I/O Delay 制約の作成



2. 構成が完了後、「OK」をクリックして制約を保存します。

4.7.3 タイミング例外制約

タイミング例外を使用することにより、ユーザーは特定のパスのデフォルトの静的タイミング解析ルールを変更できます。set_false_path、set_multicycle_path、set_max_delay、および set_min_delay の 4 つのタイミング例外制約コマンドがあります。

Set False Path

Gowin ソフトウェアはデフォルトですべてのタイミングパスを解析します。この制約文を使用して、解析する必要のないパス(つまり、非クリティカルパス)を指定できます。これはタイミング例外制約文です。これを使用して解析する必要のないパスを指定することをお勧めします。

一般に、解析不要なタイミングパスには 2 つがあります。

- テスト回路など、デザインの通常の動作に関係しないシーケンシャル回路。
- 非同期クロックドメイン・クロッシングのパス。例えば、フリップフロップ A とフリップフロップ B があり、A がデータを B に出力し、A

およびBがそれぞれ非同期クロックCLK1とCLK2によって駆動される場合、FromはCLK1、ToはCLK2として構成すると、Gowinソフトウェアは CLK1 launch から CLK2 latch のパスを解析しないようになります。

以下は、False Path制約の新規作成方法です。

1. 「Constraints > Set False Path」を選択すると、「Set False Path」ダイアログボックスがポップアップします(図 4-24)。

- Analysis type は、Setup または Hold に対するチェックを指定します。Both は両者に対するチェックを示します。
- From はパスの始点を指定します。
- To はパスの終点を指定します。
- Through は、パスの通過点またはネットを指定します。

注記：

From、To、および Through は、単独で使用するか、併用することができます。

図 4-24 False Path 制約の作成

2. 右側の「...」を右クリックし、From、To、および Through に対応する Object を選択します(図 4-14)。

Set Max/Min Delay

パス上の最大遅延値と最小遅延値を指定します。

通常、ピン間の遅延解析で使用されます。例えば、データが入力ポート A から組み合わせ回路を経由してポート B に出力される場合、デフォルトで Gowin ソフトウェアはポート A からポート B へのパスを解析および報告しません。ユーザーは、この制約を使用して、A から B までの適切な遅延値を指定することができます。Gowin ソフトウェアは、ユーザーが指定したパスを自動的に計算、解析、報告します。最大遅延が指定されている場合は、Setup 解析レポートで報告され、最小遅延が指定されている場合は、Hold 解析レポートで報告されます。

以下は、Max/Min Delay 制約の新規作成方法です。

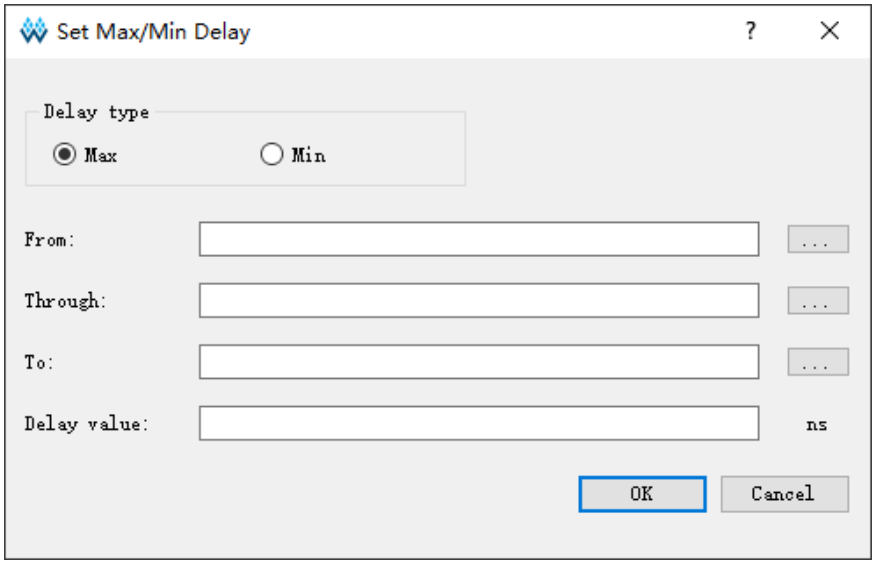
1. 「Constraints>Set Max/Min Delay」を選択すると、「Set Max/Min Delay」ダイアログボックスがポップアップします(図 4-25)。

- From：パスの始点を指定します。右側の「...」をクリックして選択します。
- To：パスの終点を指定します。右側の「...」をクリックして選択します。
- Through：パスの通過点またはネットを指定します。右側の「...」をクリックして選択します。
- Delay value：ユーザーが指定した遅延値。浮動小数点数型、小数点以下 3 桁、単位は ns。

注記：

From、To、および Through は、単独で使用するか、併用することができます。

図 4-25 Max/Min Delay 制約の作成



2. 上図に従って構成します。Delay 情報の入力後、「OK」をクリックして作成を完了します。

Set MultiCycle Path

デフォルトでは、Gowin ソフトウェアはシングルサイクルのクロック解析を実行します。つまり、セットアップ時間のチェックは、ソースクロックエッジの次のクロックサイクルの有効クロックエッジで行われます。この方法は特定のタイミングパスには適用できません。論理設計回路解析がその最も典型的な例です。論理回路は計算が複雑であるか、パスが長いです。この場合、データが安定した状態になるには、1 クロック以上の時間がかかります。

例えば、タイミングパス Path_A のデータが安定するまでに 2 サイク

ルが必要な場合、ユーザーは Value を 2 に設定する必要があります。その結果については、Setup、Hold レポートを参照してください。

注記：

- マルチサイクルパスのコマンド設定は、セットアップ時間(setup)とホールド時間(hold)に一定の影響を与えます。-setup または-hold オプションが指定されていない場合、Gowin ソフトウェアはデフォルトで-setup を選択します。setup 値が設定されている場合、hold 値はその影響を受けません。
- Gowin ソフトウェアはデフォルトで hold の自動修正機能を提供します。ユーザーが hold 値を指定した場合、Gowin ソフトウェアはユーザーが設定した制約を優先します。

以下は、Multicycle Path 制約の新規作成方法です。

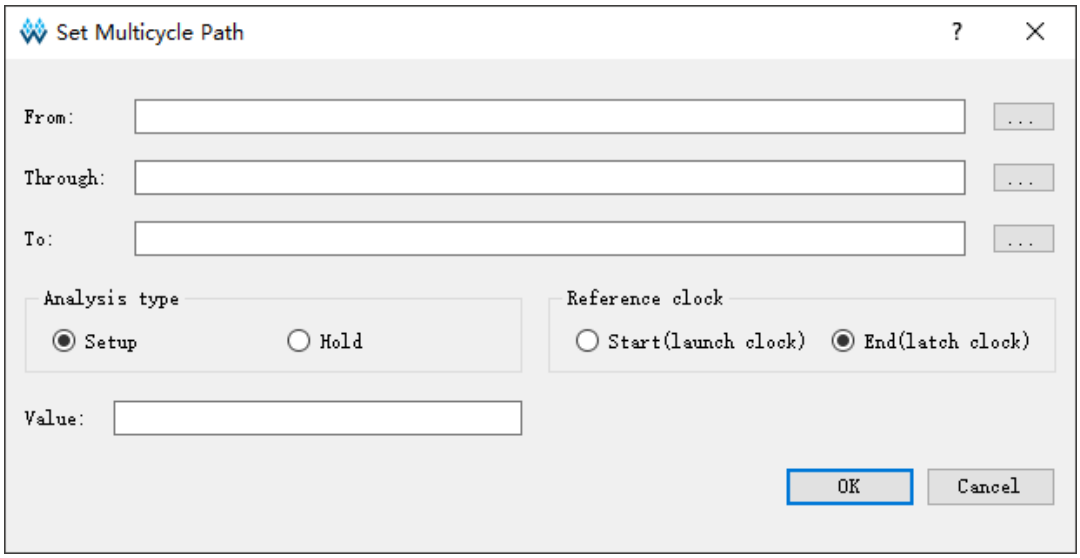
1. 「Constraints>Set Multicycle Path」を選択すると、「Set Multicycle Path」ダイアログボックスがポップアップします(図 4-26)。

- Reference clock は、リファレンスクロックが開始クロックかラッチクロックかを示します。
- Analysis type は、Setup または Hold のチェックを示します。
- From：パスの始点を指定します。右側の「…」をクリックして選択します。
- Through：パスの通過点トを指定します。右側の「…」をクリックして選択します。
- To：パスの終点を指定します。右側の「…」をクリックして選択します。
- Value：マルチサイクルの数を指定します。正または負の整数。負の場合は繰り上げを意味し、正の場合は繰り下げを意味します。

注記：

From、To、および Through は、単独で使用するか、併用することができます。

図 4-26 Multicycle Path 制約の作成



2. 構成したあと、「OK」をクリックして制約を保存します。

4.7.4 動作条件の制約

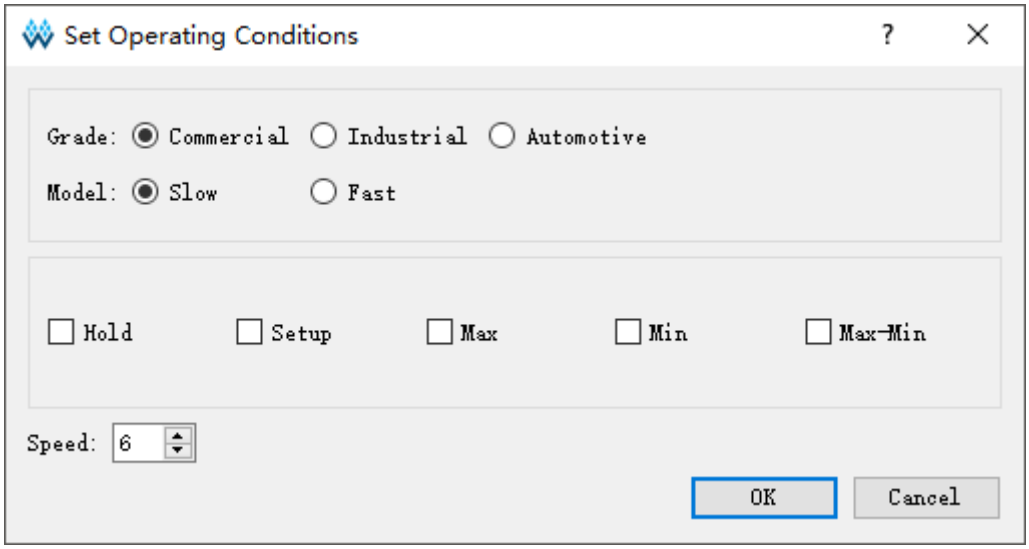
スピードグレード、モデルタイプなどを指定できます。デフォルトでは、Gowin ソフトウェアは、Setup 解析を実行するときに Slow Model(低速遅延モデル)を使用し、Hold 解析を実行するときに Fast Model(高速遅延モデル)を使用します。

また、ユーザーは、タイミングモデルの使用をカスタマイズすることもできます。たとえば、高温で電力が不安定な場合、低速遅延モデルを指定することができます。完了後、STA Tool Run Summary で使用されている遅延モデルを確認できます。

「Constraints>Set Operating Conditions」を選択すると、「Set Operating Conditions」ダイアログボックスがポップアップします(図 4-27)。

- Grade には、Commercial(コマーシャル)、Industrial(インダストリアル)、Automotive(オートモーティブ)があります。
- Model には、低速および高速があります。
- Hold と Setup は、ホールド時間適用またはセットアップ時間適用を示します。
- Max と Min の機能はそれぞれ Setup と Hold と同様です。
- Max-Min は、Max と Min を同時に選択することと同じです。

図 4-27 Operating Conditions 制約の作成



注記：

- 設定された Grade と Speed がチップの型番と一致しない場合、実際の制約が優先されます。
- 実際の制約の Grade と Speed が現在のプロジェクトをサポートしていない場合、警告メッセージが表示されます。
- Setup のみが選択されている場合、Hold は、Setup の場合の Grade-Speed に従って分析されます。
- Hold のみが選択されている場合、Setup は、Hold の場合の Grade-Speed に従って分析されます。
- エンジニアリングサンプル(ES)の場合、タイミング解析はデフォルトで最も遅いスピードグレードで実行されます。ユーザーは必要に応じてスピードグレードを設定できます。

4.7.5 タイミングレポート内容の制約

Report Timing

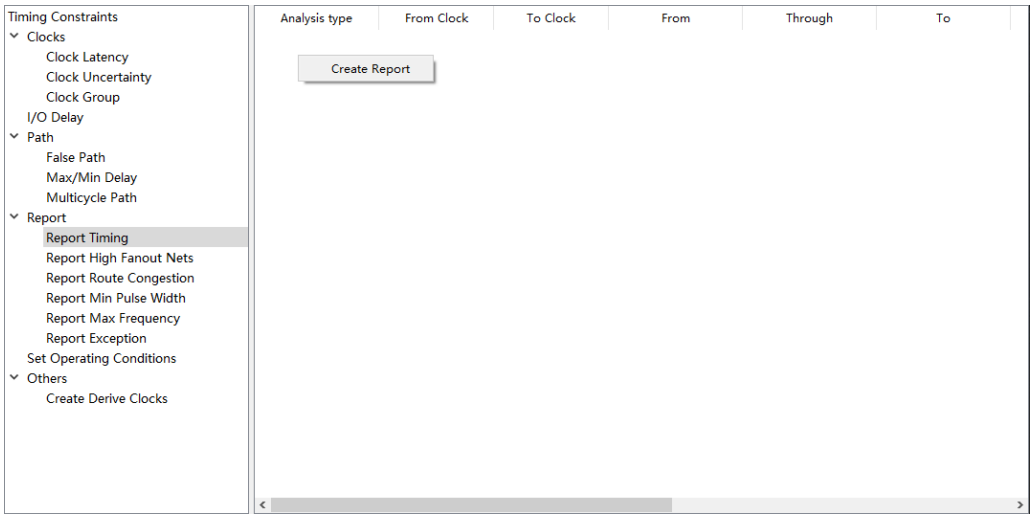
設定されたパラメータに従って、対応するレポート内容を出力します。これにより、より具体的なタイミングレポートと解析を実現できます。

たとえば、Gowin ソフトウェアはデフォルトで 25 の Setup 解析パスを報告します。ユーザーが 35 の最悪の Setup パス解析情報を見たい場合、図 4-29 の「Max Paths」に 35 を直接入力します。その結果については、Setup、Hold レポートを参照してください。

その操作手順は以下のとおりです。

1. メインウィンドウで「Timing Constraints > Report Timing」を選択し、空白スペースで右クリックすると、「Create Report」が表示されます。

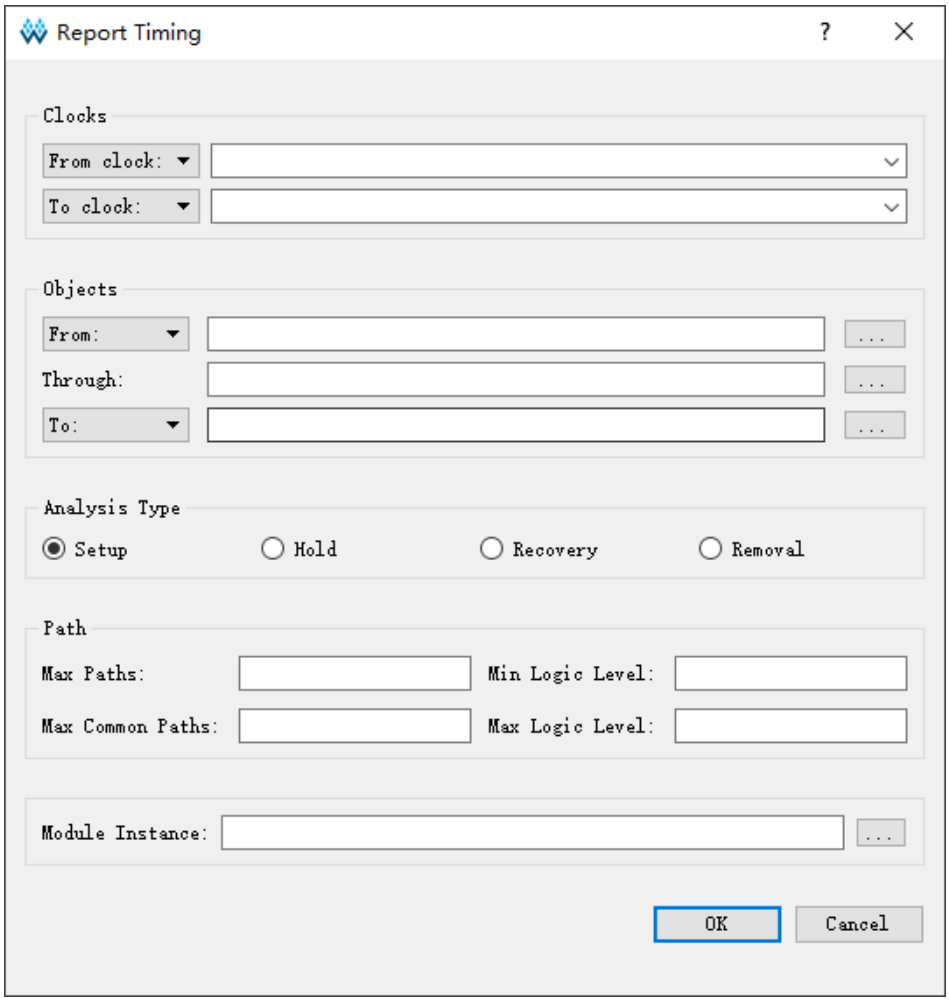
図 4-28 Report Timing の新規作成



2. 「Create Report」を選択すると、図 4-29 に示すダイアログボックスがポップアップします。

- Path は、タイミングレポートの最大パス数(Max Paths)、最大共通パス数(Max Common Paths)、最大および最小ロジックレベル(Max/Min Logic Level)を指定します。全部正の整数です。
- Clocks は、タイミングレポートパスの関連クロックを指定します。From/To Clock はそれぞれ送信クロックとサンプリングクロックを指定します。右側の「⌄」をクリックして選択します。
- Objects は、解析の開始と終了のオブジェクトを指定します。右側の「…」をクリックして選択します。
- Analysis Type には、セットアップ時間(Setup)、ホールド時間(Hold)、リカバリ時間(Recovery)、およびリムーバル時間(Removal)があります。
- Module Instance は、モジュールのインスタンスを指定します。右側の「…」をクリックして選択します。

図 4-29 Report Timing ダイアログボックス



3. 構成したあと、「OK」をクリックして保存します。

Report High Fanout Nets

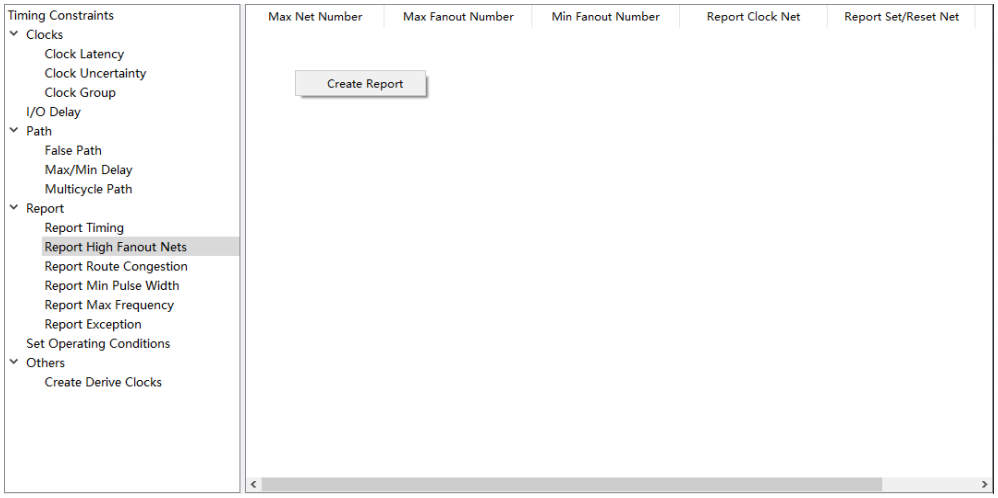
Net のファンアウト数を報告します。デフォルトでは、最大の 10 個のレポートが報告されます。

例えば、ユーザーは、ファンアウトが 5~7 の Net を表示したい場合、Min Fanout を 5、Max Fanout を 7 と指定できます。生成されたレポートは、High Fanout Nets Report で確認できます。

その操作手順は以下のとおりです。

1. メインウィンドウで「Timing Constraints>Report High Fanout Nets」を選択します。
2. 右側の空白で右クリックすると、「Create Report」が表示されます(図 4-30)。

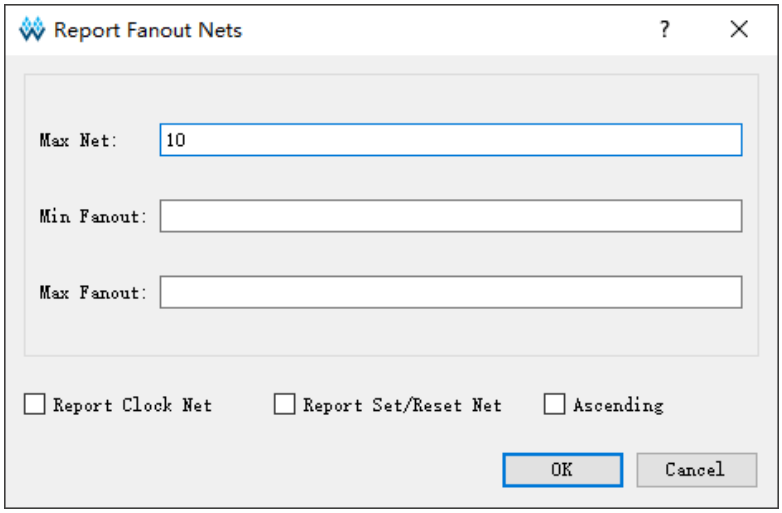
図 4-30 Report High Fanout Nets の作成



3. 「Create Report」を選択すると、図 4-31 に示すダイアログボックスがポップアップします。

- Max Net は、レポートの最大数を指定します。
- Min Fanout と Max Fanout は、それぞれファンアウトの下限と上限を指定します。正の整数です。
- Report Clock Net は、シーケンシャル・エレメントのクロック入力に接続されている Net を報告します。
- Report Set/Reset は、シーケンシャル・エレメントのリセット入力に接続されている Net を報告します。
- Ascending は Net のソート順を指し、デフォルトでは昇順が採用されています。

図 4-31 Report High Fanout Nets ダイアログボックス



4. 構成したあと、「OK」をクリックして保存します。

Report Route Congestion

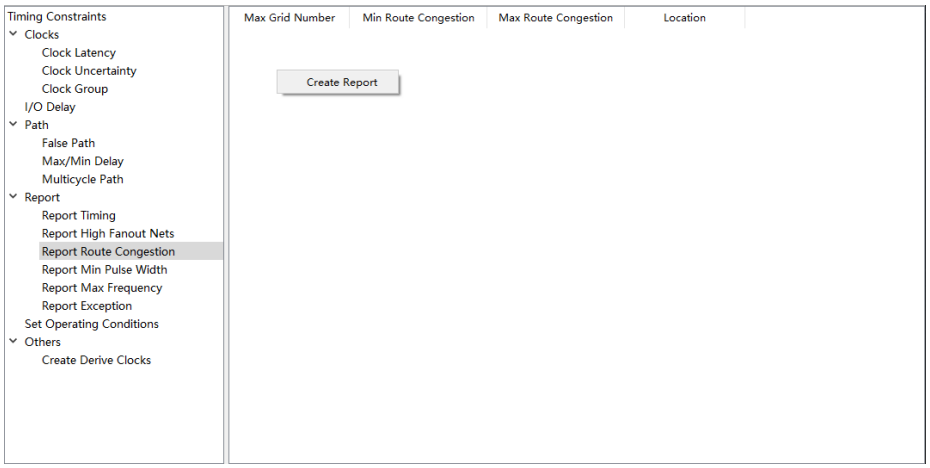
密集レベルを報告します。デフォルトでは、最悪の 10 個の Grid が報告されます。

これは通常、特定の Grid の配線の密集レベルの報告に使用されます。例えば、ユーザーが Grid R4C4 の密集レベルを報告したい場合は、Grid Location を R4C4 と指定します。生成されたレポートは、Route Congestions Report で確認できます。

その手順は以下のとおりです。

1. メインウィンドウで、「Timing Constraints>Report Route Congestion」を選択します。
2. 右側の空白で右クリックすると、「Create Report」が表示されます(図 4-32)。

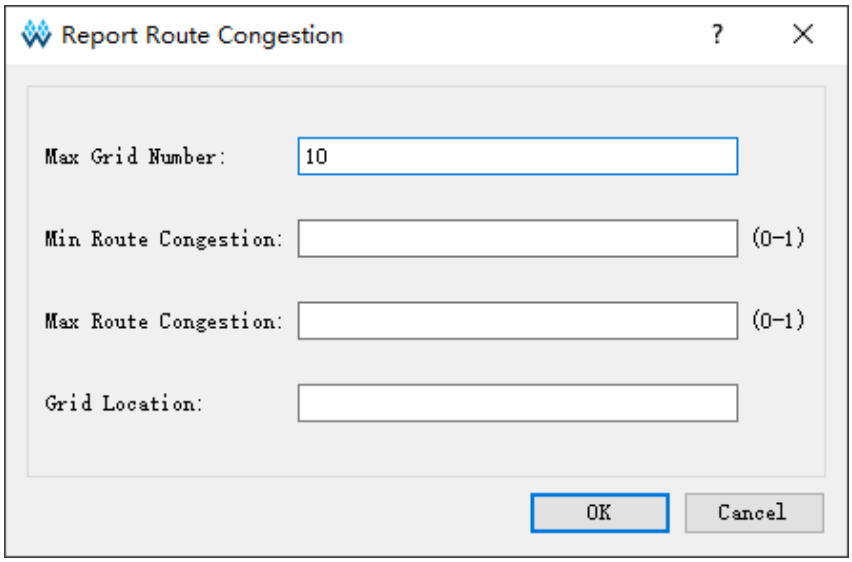
図 4-32 Report Route Congestion の作成



3. 「Create Report」を選択すると、図 4-33 に示すダイアログボックスがポップアップします。

- Max Grid Number はレポートの数を指定します。
- Min、Max Route Congestion は、それぞれ配線の密集レベルの下限と上限を指定します。浮動小数点数型、小数点以下 3 桁。
- Grid Location は、報告される Grid を指定します(例えば、R4C4)。

図 4-33 Report Route Congestion ダイアログボックス



4. 構成したあと、「OK」をクリックして保存します。

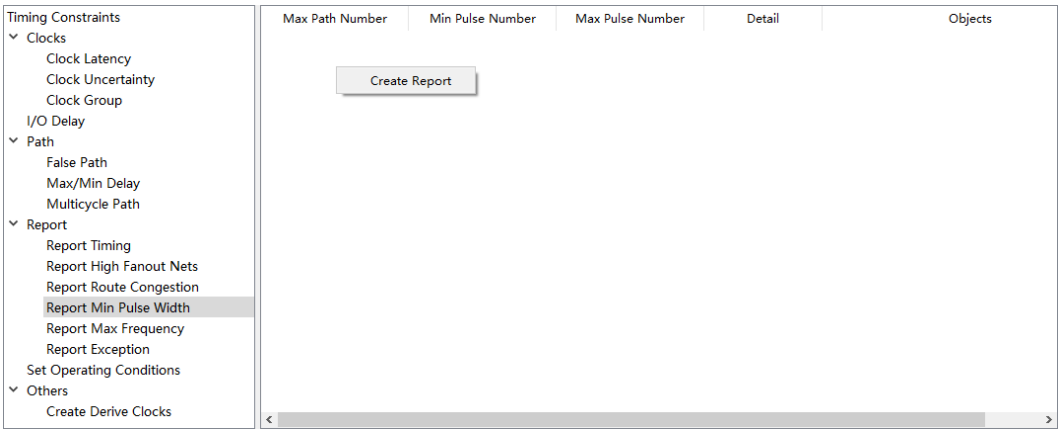
Report Min Pulse Width

最小パルス幅を報告します。デフォルトでは、10 個のレポートが報告されます。ユーザーはこの制約文を使用して、特定の範囲内のパルス幅または特定のオブジェクトでのパルス幅を報告できます。例えば、フリップフロップのインスタンス Reg11_Z の場合、ユーザーは Objects を Reg11_Z として指定できます。生成されたレポートは、Minimum Pulse Width Report で確認できます。

手順は以下のとおりです。

1. メインウィンドウで「Timing Constraints>Report Min Pulse Width」を選択します。
2. 右側の空白で右クリックすると、「Create Report」が表示されます(図 4-34)。

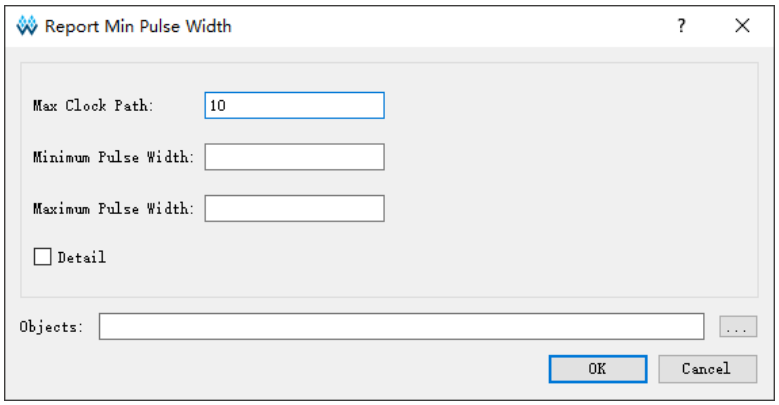
図 4-34 Report Min Pulse Width の作成



3. 「Create Report」を選択すると、図 4-35 に示すダイアログボックスがポップアップします。

- Max Clock Path は、レポートの最大数を指定します。正の整数です。
- Minimum、Maximum Pulse Width は、報告されるパルス幅の下限と上限を指定します。浮動小数点数型、小数点以下 3 桁。
- Detail は、詳細なパスを報告するかどうかを指定します。
- Objects は、報告する必要のあるシーケンシャル・エレメントを指定します。DFF などのフリップフロップのみがサポートされます。右側の「...」をクリックして選択します。

図 4-35 Report Min Pulse Width ダイアログボックス



4. 構成したあと、「OK」をクリックして保存します。

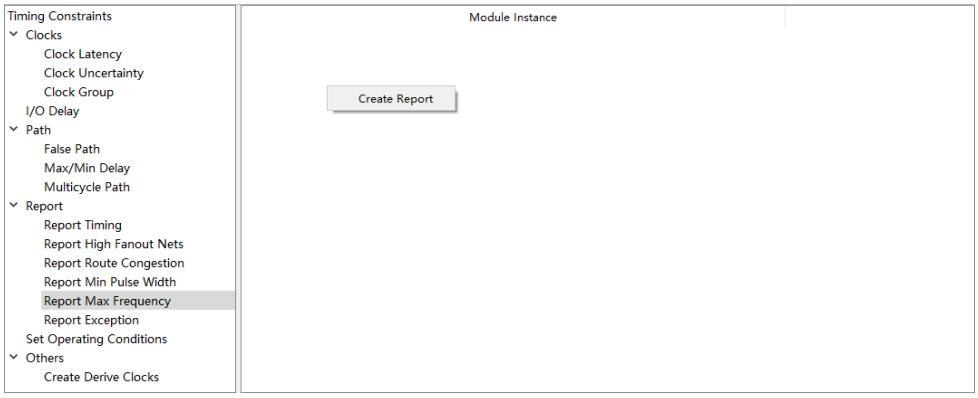
Report Max Frequency

最大動作周波数を報告します。デフォルトでは、Gowin ソフトウェアは Top 層のクロックの最大周波数のみを報告します。ユーザーは、特定のモジュールの最大動作クロック周波数のレポートを指定できます。このモジュールの最大動作クロック周波数のクリティカルパスは、このモジュール内に限定されるものではなく、このモジュールの同期に関連するものです。

その操作手順は以下のとおりです。

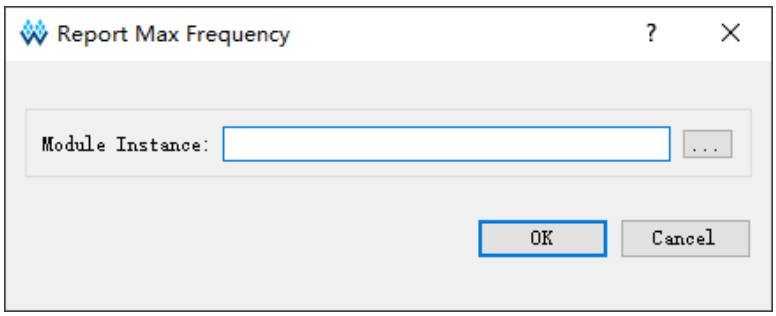
1. メインウィンドウで「Timing Constraints > Report > Report Max Frequency」を選択します。
2. 右側の空白で右クリックすると、「Create Report」が表示されます(図 4-36)。

図 4-36 Report Max Frequency の新規作成



3. 「Create Report」を選択すると、図 4-37 に示すダイアログボックスがポップアップします。「Module Instance」はモジュールのインスタンスの名前です。右側の「...」をクリックして選択します。

図 4-37 Report Max Frequency ダイアログボックス



4. 構成したあと、「OK」をクリックして保存します。

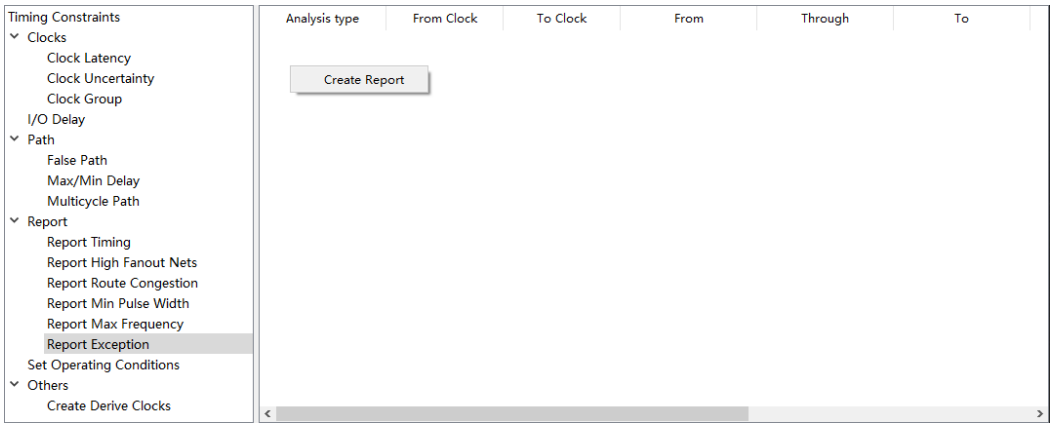
Report Exception

例外制約文の影響を受けるタイミングパスをさらに制約して、ユーザーが関心のあるタイミングパスを報告します。

その手順は以下のとおりです。

1. メインウィンドウで「Timing Constraints > Report > Report Exception」を選択します。
2. 右側の空白で右クリックすると、「Create Report」が表示されます(図 4-38)。

図 4-38 Report Exception の作成

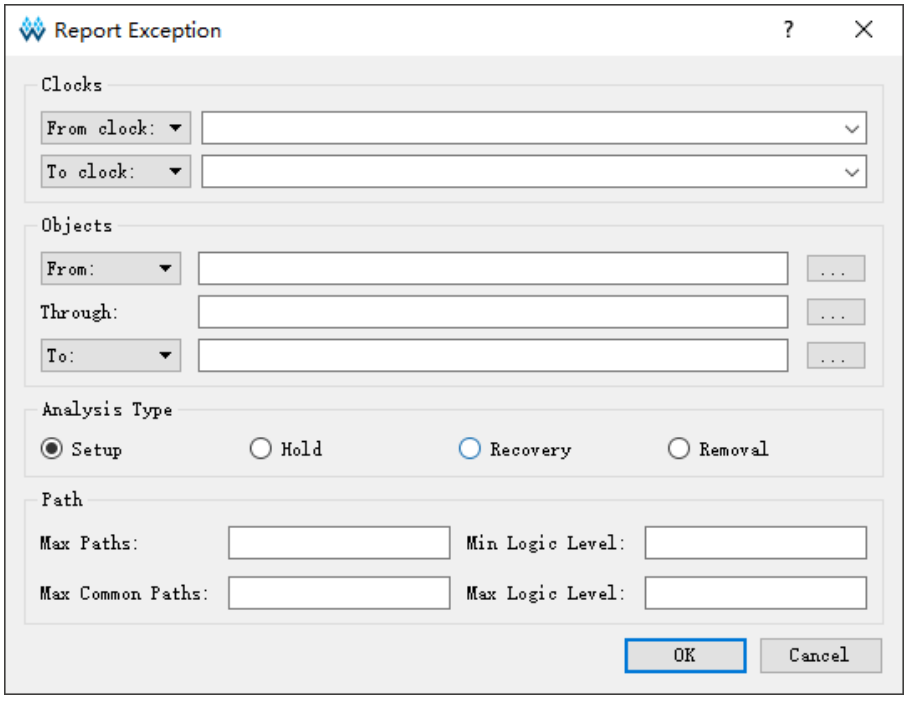


3. 「Create Report」を選択すると、図 4-39 に示すダイアログボックスがポップアップします。

注記：

各オプションの詳細については、Report Timing を参照してください。

図 4-39 Report Exception ダイアログボックス



4. 構成したあと、「OK」をクリックして保存します。

4.7.6 その他の制約

Create Derive Clocks

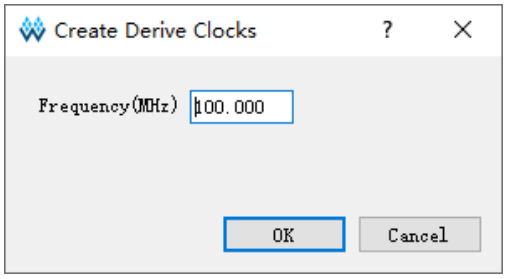
- デザインのためにグローバルのクロックを作成します。
- 最大 1200MHz までの周波数設定をサポートします。

Derive Clocks 制約を追加するには、2 つの方法があります：

1. 「Constraints」メニューで Derive Clocks 制約を追加します。

a). 「Constraints > Create Derive Clocks…」を選択すると「Create Derive Clocks」ダイアログボックスがポップアップします(図 4-40)。

図 4-40 Create Derive Clocks

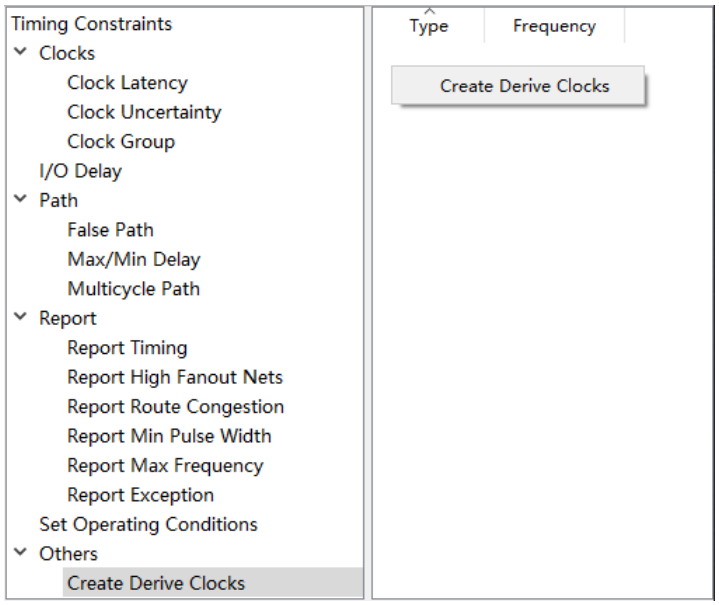


b). 情報を入力します。

- Frequency(Mhz)：グローバルの周波数、1200 以下の正の浮動小数点数、小数点以下 3 桁。

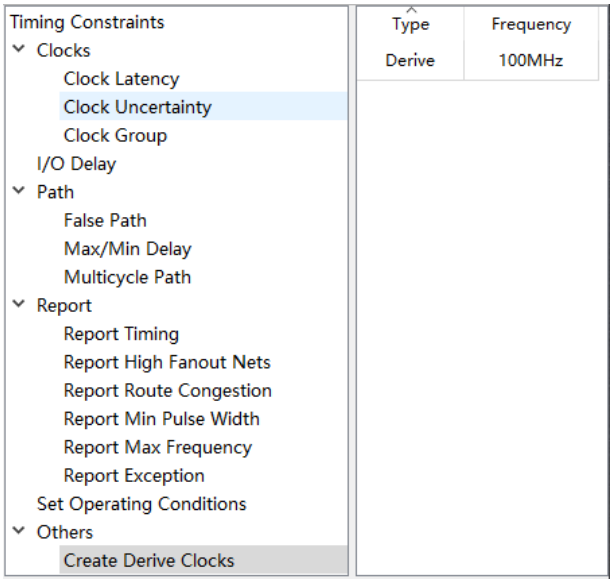
2. 「Others > Create Derive Clocks」で「Derive Clocks」を作成します。空白で右クリックし、「Create Derive Clocks」を選択して Derive Clocks を作成します(図 4-41)。

図 4-41 选择 Create Derive Clocks



クロックが作成されると、Derive Clocks リストには対応する制約が追加されます(図 4-42)。

図 4-42 Derive Clocks リスト



4.7.6 保存とエクスポート

すべての制約の編集が完了後、「File>Save」または「File>Save As」をクリックすると、現在のタイミング制約エディタの制約情報がタイミング制約ファイル(.sdc)に保存されます。タイミング制約ファイルの内容形式は、付録 A.タイミング制約構文仕様を参照してください。

4.8 タイミング制約の優先度

Gowin ソフトウェアにより提供されるタイミング制約の優先度は次に示すとおりです(低い順)。

1. create_clock と create_generated_clock
2. set_multicycle_path
3. set_max_delay と set_min_delay
4. set_false_path
5. set_clock_groups

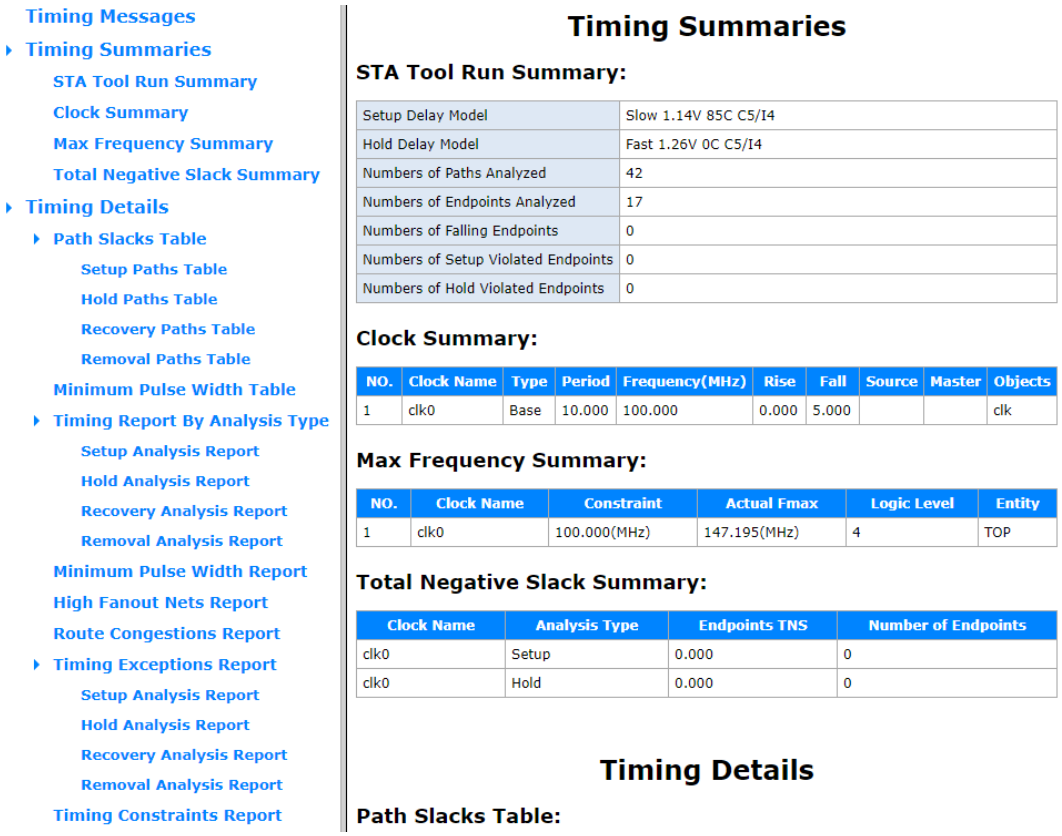
注記：

同じタイミングパスで競合が生じる可能性のあるタイミング制約のみが優先度で並べ替えられます。言及されていない他の制約の場合は、異なるタイプの制約間の競合が生じません。

5 タイミングレポート

このセクションでは、GOWIN セミコンダクターの静的タイミング解析のレポートについて説明します。図 5-1 に示すように、レポートは左側のナビゲーションバーと右側のコンテンツバーからなり、要件を満たさない項目は赤で表示されます。

図 5-1 静的タイミング解析レポート



5.1 Timing Summaries

タイミングサマリ(Timing Summaries)は、STA Tool Run Summary、Clock Summary、Max Frequency Summary、および Total Negative Slack

Summary の 4 つの部分で構成されています(図 5-2)。

図 5-2 Timing Summaries

Timing Summaries

STA Tool Run Summary:

Setup Delay Model	Slow 1.14V 85C C5/I4
Hold Delay Model	Fast 1.26V 0C C5/I4
Numbers of Paths Analyzed	42
Numbers of Endpoints Analyzed	17
Numbers of Falling Endpoints	0
Numbers of Setup Violated Endpoints	0
Numbers of Hold Violated Endpoints	0

Clock Summary:

NO.	Clock Name	Type	Period	Frequency(MHz)	Rise	Fall	Source	Master	Objects
1	clk0	Base	10.000	100.000	0.000	5.000			clk

Max Frequency Summary:

NO.	Clock Name	Constraint	Actual Fmax	Logic Level	Entity
1	clk0	100.000(MHz)	147.195(MHz)	4	TOP

Total Negative Slack Summary:

Clock Name	Analysis Type	Endpoints TNS	Number of Endpoints
clk0	Setup	0.000	0
clk0	Hold	0.000	0

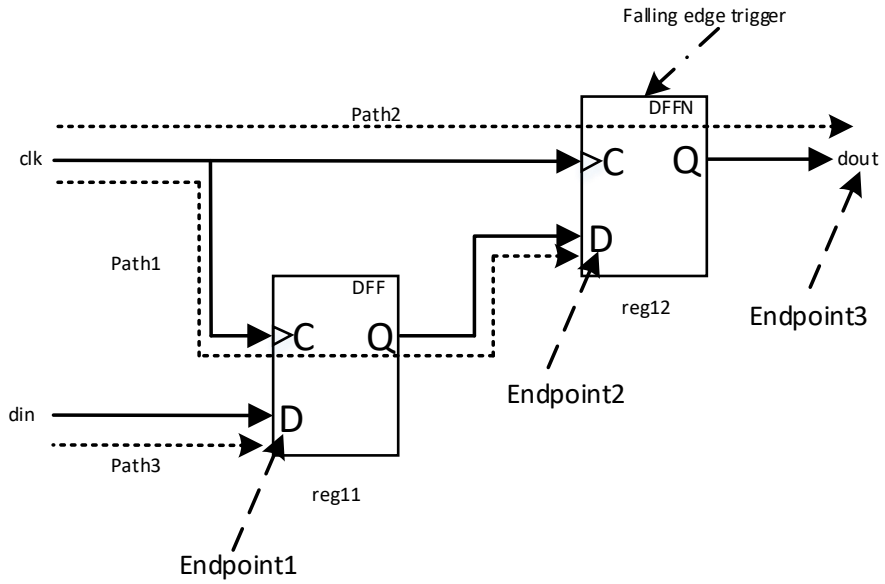
5.1.1 STA Tool Run Summary

- Setup Delay Model：セットアップ時間の解析のために使用されるデータモデル。デフォルトでは、Slow モデル(つまり、高温と低圧)が使用されます。
- Hold Delay Model：ホールド時間の解析のために使用されるデータモデル。デフォルトでは、Fast モデル(つまり、低温と高圧)が使用されます。
- Numbers of Paths Analyzed：静的タイミング解析パスの数。図 5-3 に示すように、Path1、Path2、および Path3 という合計 3 つのタイミングパスが解析されました。
- Numbers of Endpoints Analyzed：解析されたタイミングパス終点の数。図 5-3 に示すように、Endpoint1、Endpoint2、Endpoint3 という

合計 3 つの終点が解析されました。

- Numbers of Falling Endpoints：立ち下がりエッジ・トリガの終点の数。図 5-3 に示すように、reg12 は立ち下がりエッジ・トリガの DFFN であるため、終点 D は立ち下がりエッジ・トリガの終点となります。
- Numbers of Setup Violated Endpoints：セットアップ時間を満たさない終点の数。
- Numbers of Hold Violated Endpoints：ホールド時間を満たさない終点の数。

図 5-3 Path & Endpoints



5.1.2 Clock Summary

ユーザーデザインのすべてのクロックを報告します。デザイン内のクロックが制約されていない場合、ソフトウェアはデフォルトのクロックを作成します。Arora ファミリーの場合、クロック周波数はデフォルトで 100MHz、LittleBee ファミリーの場合、クロック周波数はデフォルトで 50MHz です。また、GAO を含むデザインの場合、TCK の周波数は 20MHz です。

- NO.：番号。
- Clock Name：クロックの名前。デフォルトでクロックを作成する際、クロック名が重複している場合、Gowin ソフトウェアは自動的にサフィックス「_gowin」を追加します。PLL、OSC、および CLKDIV タイプの場合、Gowin ソフトウェアはデフォルトでクロック名にサフィックス「.default_gen_clk」を追加します。
- Type：Base と Generated の 2 つのタイプがあります。Base は基本ク

ロックを表し、Generated は派生クロックを表します。

- Period：クロックサイクル。
- Frequency (MHz) ：クロック周波数(MHz)。
- Rise：クロックの立ち上がりエッジ時間。
- Fall：クロックの立ち下がりエッジ時間。
- Source：クロックソース。port、pin、net、reg から取得できます。
- Master：クロックの派生元のクロック。マスタークロック。
- Objects：port、pin、net、reg などのクロックオブジェクト。

5.1.3 Max Frequency Summary

- NO.：番号。
- Clock Name：タイミングモデルを駆動するクロックの名前。
- Constraint：SDC 制約のクロック周波数、または SDC 制約なしの場合のデフォルトのクロック周波数。
- Actual Fmax：配置配線の後に Gowin ソフトウェアの分析による最大の実際の周波数。
- Logic Level：クロックによって駆動される最悪のタイミングパスのロジックレベル数。
- Entity：最大周波数のモジュール。デフォルトは TOP です。

注記：

- 配置配線の後にクロックがタイミングモデルを駆動しない場合は、「No timing paths to get frequency of *」が表示されます。
- 最大クロック周波数レオートでは、同じクロックで駆動されるタイミングモデル(派生クロックを含む)のクロックのみが報告されます。
- Gowin ソフトウェアがより正確な解析を実行できるように、デザインに完全なタイミング制約を追加することをお勧めします。

5.1.4 Total Negative Slack Summary

- Clock Name：クロックの名前。
- Analysis Type：Setup と Hold の 2 種類があります。
- Endpoints TNS：クロック(ClockName に対応)によって駆動されるタイミングパスの終点スラックが負の場合の合計時間。同じ終点を持つパスに対しては、最悪のパスのみが集計されます。
- Number of Endpoints：クロック(ClockName に対応)によって駆動されるタイミングパスの終点スラックが負の場合の合計終点数。同じ終点

を持つパスに対しては、最悪のパスのみが集計されます。

注記：

同じクロックで駆動されるタイミングモデルのみが報告されます。

5.2 Timing Details

5.2.1 Path Slacks Table

Path Slacks Table は、Setup Paths Table(セットアップ時間パス解析テーブル)、Hold Paths Table(ホールド時間パス解析テーブル)、Recovery Paths Table(リカバリー時間パス解析テーブル)、Removal Paths Table(リムーバル時間パス解析テーブル)で構成された、タイミングパスの静的解析のスラックテーブルです。上記の 4 種類のテーブルのヘッダー(図 5-4 参照)の説明は次のとおりです。

- Path Number：パス番号。デフォルトの最大レポート数は 25 です。
- Path Slack：データ要求時間からデータ到着時間を引いた値であり、この値が負の場合、タイミングは満たされません。
- From Node：前段シーケンシャル・エレメントのタイミング解析の開始ノード。
- To Node：後段のシーケンシャル・エレメントのタイミング解析の終了ノード。
- From Clock：前段シーケンシャル・エレメントのデータ送信クロックと送信エッジ・タイプ(立ち上がりエッジまたは立ち下がりエッジ)。
- To Clock：後段のシーケンシャル・エレメントのデータ・ラッチ・クロックとラッチ・エッジ・タイプ。
- Relation：送信クロックとサンプリングクロックの時間関係を示します。
- Clock Skew：クロックスキュー。送信クロックとラッチクロックが前段と後段のシーケンシャル・エレメントに到着する時間差を指します。
- Data Delay：データ到着パスにおけるデータ遅延です。その値は、データ到着パス全体における遅延値の一部です。

注記：

- 解析に使用できるタイミングパスがない場合、「Nothing to report!」と表示されます。
- Path Slacks Table では、デフォルトで最悪の 25 パスが解析されます。ユーザーは Report Timing を利用してこの 25 パス以外のパスを確認できます。
- Path Slacks Table のデフォルト解析には、クロック・ドメイン・クロッシングのタ

イミングパスの解析が含まれます。解析したくないパスは、Set Clock Group または Set False Path を使用して指定できます。

図 5-4 Path Slacks Table

Path Slacks Table:

Setup Paths Table

Report Command:report_timing -setup -max_paths 25 -max_common_paths 1

Path Number	Path Slack	From Node	To Node	From Clock	To Clock	Relation	Clock Skew	Data Delay
1	8.806	synS_r_s0/Q	synE_r_s0/D	ck0:[R]	ck0:[R]	10.000	0.000	0.794

Hold Paths Table

Report Command:report_timing -hold -max_paths 25 -max_common_paths 1

Path Number	Path Slack	From Node	To Node	From Clock	To Clock	Relation	Clock Skew	Data Delay
1	0.570	synS_r_s0/Q	synE_r_s0/D	ck0:[R]	ck0:[R]	0.000	0.000	0.570

Recovery Paths Table

Report Command:report_timing -recovery -max_paths 25 -max_common_paths 1

Path Number	Path Slack	From Node	To Node	From Clock	To Clock	Relation	Clock Skew	Data Delay
1	8.649	rstSrc_r_s0/Q	rstObj_r_s0/CLEAR	ck0:[R]	ck1:[R]	10.000	0.000	1.278

Removal Paths Table

Report Command:report_timing -removal -max_paths 25 -max_common_paths 1

Path Number	Path Slack	From Node	To Node	From Clock	To Clock	Relation	Clock Skew	Data Delay
1	0.788	rstSrc_r_s0/Q	rstObj_r_s0/CLEAR	ck0:[R]	ck1:[R]	0.000	0.000	0.833

5.2.2 Minimum Pulse Width Table

シーケンシャル・エレメントが認識できる最小パルス幅の静的タイミング解析テーブルです。パルス幅とは、有効な High/Low レベルの信号が持続する時間を指します。デフォルトで最悪の 10 件が報告されます。図 5-5 のヘッダーについて説明します。

- Number：番号。デフォルトは 10 個報告されます。
- Slack：シーケンシャル・エレメントが認識できる最小パルス幅のスラック値。
- Actual Width：配置配線後の静的タイミング解析後にシーケンシャル・エレメントが認識できる実際のパルス幅。
- Required Width：シーケンシャル・エレメントが認識できるために必要な最小パルス幅。
- Type：パルス幅のタイプ。Low Pulse Width(論理 Low のパルス幅)と High Pulse Width(論理 High のパルス幅)の 2 つのタイプがあります。
- Clock：最小パルス幅解析用のクロック。
- Objects：最小パルス幅解析用のシーケンシャル・エレメントのインスタンス。

注記：

最小パルス幅解析レポートがない場合、「Nothing to report!」と表示されます。

図 5-5 Minimum Pulse Width Table

Minimum Pulse Width Table:

Report Command:report_min_pulse_width -nworst 10 -detail

Number	Slack	Actual Width	Required Width	Type	Clock	Objects
1	2.738	4.238	1.500	Low Pulse Width	DEFAULT_CLK	reg12
2	2.738	4.238	1.500	Low Pulse Width	DEFAULT_CLK	reg11_Z
3	2.813	4.313	1.500	High Pulse Width	DEFAULT_CLK	reg12
4	2.813	4.313	1.500	High Pulse Width	DEFAULT_CLK	reg11_Z

5.2.3 Timing Report By Analysis Type

Setup Analysis Report、Hold Analysis Report、Recovery Analysis Report、および Removal Analysis Report の 4 つのタイプの静的タイミング解析があります。その中で、Setup Analysis Report には Recovery Analysis Report が含まれ、Hold Analysis Report には Removal Analysis Report が含まれます。同じ解析・計算方法が使用されます。以下にこの 4 種類の解析を紹介します。

Setup Analysis Report

シーケンシャル・エレメントのクロック信号の立ち上がりエッジの前にデータが安定している時間を解析するためのセットアップ時間解析レポートです。この時間が十分でない場合、データはクロックの立ち上がりエッジでシーケンシャル・エレメントに安定的に供給されません。

Gowin ソフトウェアは、詳細な計算と解析を経て、データ到着時間、データ要求時間、サンプリングクロック、送信クロックなどをユーザー参照用の Setup Analysis Report に出力します。

このレポートは、コマンド report_timing -setup によって生成されます。Gowin ソフトウェアは、デフォルトで 25 の最悪のスラックのタイミングパスを分析して報告します。レポートには、Path Summary、Data Arrival Path、および Path Statistics が含まれます。

1. Path Summary。図 5-6 は静的タイミング解析のパス情報の概要です。その詳細は次のとおりです。

- Slack：許容されるデータ最大遅延時間から実際のデータ到着時間を差し引いた時間です。正の値はタイミング収束を示し、負の値はタイミングが収束しないことを示します。
- Data Arrival Time：Launch edge が後続のシーケンシャル・エレメントのデータポートに到着するのにかかる時間。
- Data Required Time：Latch edge が後続のシーケンシャル・エレメントのクロックポートに到着するのにかかる時間。
- From：前段のシーケンシャル・エレメント。

- To：後段のシーケンシャル・エレメント。
- Launch Clock：開始クロック。対象エッジは、R(立ち上がりエッジ)または F(立ち下がりエッジ)です。
- Latch Clock：ラッチクロック。対象エッジは、R(立ち上がりエッジ)または F(立ち下がりエッジ)です。

図 5-6 Path Summary

Path Summary:

Slack	5.789
Data Arrival Time	6.767
Data Required Time	12.556
From	reg11_Z
To	reg12_Z
Launch Clk	sysclk1:[R]
Latch Clk	sysclk1:[R]

2. Data Arrival Path。図 5-7 にはデータ到着パスを示します。その詳細は次のとおりです。

- AT：タイミングパス上の時間ポイントを指します。
- DELAY：遅延値を示します。
- TYPE：タイミング解析パス上のノードのタイプを示し、空の場合は使用できないことを示します。

  注記：

  図 5-7 の各 TYPE の意味は次のとおりです。

  - tCL：time of clock latency、クロックソース遅延。
  - tINS：time of module instance、インスタンス化されたモジュールの遅延。
  - tNET：time of net、net の遅延。
  - tC2Q：time of clock to quit、シーケンシャル・エレメントの内部遅延。

- RF：現在解析されているエレメントの反転のタイプを指します。RR は正のパルスが反転しないことを意味し、FF は負のパルスが反転しないことを意味し、RF は正のパルスが負のパルスに反転することを意味し、FR は負のパルスが正のパルスに反転することを意味します。
- FANOUT：ファンアウト。
- LOC：現在解析されているエレメントの、デバイス内の物理的な位置。位置情報なしの場合は DHCEN などの UNPLACE マークを使用します。
- NODE：静的タイミング解析のパス上のノード。インスタンス名とポート、クロック、およびクロックエッジのアクティブ化時間(active clock edge time)が含まれます。

図 5-7 Data Arrival Path

Data Arrival Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysclk1
0.000	0.000	tCL	RR	1	IOL7[A]	clk1_ibuf/I
0.943	0.943	tINS	RR	2	IOL7[A]	clk1_ibuf/O
3.236	2.293	tNET	RR	1	IOL2[B]	reg11_Z/CLK
3.786	0.550	tC2Q	RF	1	IOL2[B]	**reg11_Z/Q**
6.767	2.981	tNET	FF	1	R5C9[1][A]	**reg12_Z/D**

3. Data Required Path。図 5-8 に示すように、データ要求パスとは、クロックが有効なエッジからシーケンシャル・エレメントのクロックポートに到着するまでのパスを指します。

注記：

図 5-8 の TYPE の意味は次のとおりです。

- tUnc：time of clock uncertainty、クロックのばらつき値。
- tSu：time of setup、セットアップ時間。

図 5-8 Data Required Path

Data Required Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
10.000	10.000					active clock edge time
10.000	0.000					sysclk1
10.000	0.000	tCL	RR	1	IOL7[A]	clk1_ibuf/I
10.943	0.943	tINS	RR	2	IOL7[A]	clk1_ibuf/O
13.236	2.293	tNET	RR	1	R5C9[1][A]	reg12_Z/CLK
13.036	-0.200	tUnc				reg12_Z
12.556	-0.480	tSu		1	R5C9[1][A]	reg12_Z

4. Path Statistics
   図 5-9 にパスの統計情報を示します。

- Clock Skew：クロックスキュー。
- Setup Relationship：前段のシーケンシャル・エレメントのデータ送信と後段のシーケンシャル・エレメントのデータラッチ間の時間関係。
- Logic Level：2 つのシーケンシャル・エレメント間の論理エレメントの数。0 は直接接続されていることを意味します。
- Arrival Clock Path Delay：Data Arrival Path のクロック遅延状況の統計を取ります。cell は論理エレメントの遅延、route は配線の遅延、tC2Q はシーケンシャル・エレメントの内部遅延を表します。
- Arrival Data Path Delay：Data Arrival Path のデータ遅延状況の統計を取ります。
- Required Clock Path Delay：Data Required Path のクロック遅延状況の統計を取ります。

図 5-9 Path Statistics

Path Statistics:

Clock Skew	0.000
Setup Relationship	10.000
Logic Level	1
Arrival Clock Path Delay	cell: 0.943, 29.131%; route: 2.293, 70.869%
Arrival Data Path Delay	cell: 0.000, 0.000%; route: 2.981, 84.423%; tC2Q: 0.550, 15.577%
Required Clock Path Delay	cell: 0.943, 29.131%; route: 2.293, 70.869%

Hold Analysis Report

図 5-10 はシーケンシャル・エレメントのクロック信号の立ち上がりエッジの前にデータが安定している時間を解析するためのホールド時間解析レポートです。この時間が十分でない場合、データはシーケンシャル・エレメントに安定して供給されません。Gowin ソフトウェアは詳細な計算と分析を実行し、最終的にデータ到着時間、データ要求時間、サンプリングクロック、送信クロックなどをユーザー参照用のレポートに出力します。このレポートは、コマンド report_timing -hold によって生成されます。Gowin ソフトウェアは、デフォルトで 25 の最悪のスラックのタイミングパスを分析して報告します。レポートのヘッダーの詳細については、Setup Analysis Report を参照してください。

図 5-10 Hold Analysis Report

Hold Analysis Report

Report Command:report_timing -hold -max_paths 25 -max_common_paths 1

Path1

Path Summary:

Slack	1.003
Data Arrival Time	3.554
Data Required Time	2.551
From	reg11_s0
To	reg12_s0
Launch Clk	sysyclk:[R]
Latch Clk	sysyclk:[R]

Data Arrival Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysyclk
0.000	0.000	tCL	RR	1	IOL11[A]	clk_ibuf/I
0.811	0.811	tINS	RR	2	IOL11[A]	clk_ibuf/O
2.533	1.723	tNET	RR	1	R2C9[0][A]	reg11_s0/CLK
2.933	0.400	tC2Q	RR	1	R2C9[0][A]	**reg11_s0/Q**
3.554	0.621	tNET	RR	1	R2C9[1][A]	**reg12_s0/CLEAR**

Data Required Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysyclk
0.000	0.000	tCL	RR	1	IOL11[A]	clk_ibuf/I
0.811	0.811	tINS	RR	2	IOL11[A]	clk_ibuf/O
2.533	1.723	tNET	RR	1	R2C9[1][A]	reg12_s0/CLK
2.533	0.000	tUnc				reg12_s0
2.551	0.018	tHld		1	R2C9[1][A]	reg12_s0

Path Statistics:

Clock Skew	0.000
Hold Relationship	0.000
Logic Level	1
Arrival Clock Path Delay	cell: 0.811, 31.998%; route: 1.723, 68.002%
Arrival Data Path Delay	cell: 0.000, 0.000%; route: 0.621, 60.818%; tC2Q: 0.400, 39.182%
Required Clock Path Delay	cell: 0.811, 31.998%; route: 1.723, 68.002%

Recovery Analysis Report

図 5-11 はリカバリ時間解析レポートです。リカバリ時間：シーケンシャル・エレメントの有効クロックエッジの前に非同期クリア/セット/リセット信号が安定していなければならない最小時間。この時間が満たされていなければ、フリップフロップは正常な動作状態に入れないことがあります。リカバリー時間の解析と計算の方法はセットアップ時間と一致しています。このレポートは、コマンド report_timing -recovery によって生成されます。Gowin ソフトウェアは、デフォルトで 25 の最悪のスラックのタイミングパスを分析して報告します。レポートのヘッダーの詳細については、Setup Analysis Report を参照してください。

図 5-11 Recovery Analysis Report

Recovery Analysis Report

Report Command:report_timing -recovery -max_paths 25 -max_common_paths 1

Path1

Path Summary:

Slack	8.355
Data Arrival Time	4.629
Data Required Time	12.984
From	reg11_s0
To	reg12_s0
Launch Clk	sysyclk:[R]
Latch Clk	sysyclk:[R]

Data Arrival Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysyclk
0.000	0.000	tCL	RR	1	IOL11[A]	clk_ibuf/I
0.943	0.943	tINS	RR	2	IOL11[A]	clk_ibuf/O
3.236	2.293	tNET	RR	1	R2C9[0][A]	reg11_s0/CLK
3.786	0.550	tC2Q	RF	1	R2C9[0][A]	**reg11_s0/Q**
4.629	0.843	tNET	FF	1	R2C9[1][A]	**reg12_s0/CLEAR**

Data Required Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
10.000	10.000					active clock edge time
10.000	0.000					sysyclk
10.000	0.000	tCL	RR	1	IOL11[A]	clk_ibuf/I
10.943	0.943	tINS	RR	2	IOL11[A]	clk_ibuf/O
13.236	2.293	tNET	RR	1	R2C9[1][A]	reg12_s0/CLK
13.036	-0.200	tUnc				reg12_s0
12.984	-0.052	tSu		1	R2C9[1][A]	reg12_s0

Path Statistics:

Clock Skew	0.000
Setup Relationship	10.000
Logic Level	1
Arrival Clock Path Delay	cell: 0.943, 29.131%; route: 2.293, 70.869%
Arrival Data Path Delay	cell: 0.000, 0.000%; route: 0.843, 60.531%; tC2Q: 0.550, 39.469%
Required Clock Path Delay	cell: 0.943, 29.131%; route: 2.293, 70.869%

Removal Analysis Report

図 5-12 はリムーバル時間解析レポートです。リムーバル時間：シーケンシャル・エレメントの有効クロックエッジの後に非同期クリア/セット/リセット信号が安定していなければならない最小時間。この時間が満たされていなければ、フリップフロップは正常な動作状態に入れないことがあります。リムーバル時間の解析と計算の方法はホールド時間と一致しています。このレポートは、コマンド report_timing -removal によって生成されます。Gowin ソフトウェアは、デフォルトで 25 の最悪のスラックのタイミングパスを分析して報告します。レポートのヘッダーの詳細については、Hold Analysis Report を参照してください。

図 5-12 Removal Analysis Report

Removal Analysis Report

Report Command:report_timing -removal -max_paths 25 -max_common_paths 1

Path1

Path Summary:

Slack	1.003
Data Arrival Time	3.554
Data Required Time	2.551
From	reg11_s0
To	reg12_s0
Launch Clk	sysyclk:[R]
Latch Clk	sysyclk:[R]

Data Arrival Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysyclk
0.000	0.000	tCL	RR	1	IOL11[A]	clk_ibuf/I
0.811	0.811	tINS	RR	2	IOL11[A]	clk_ibuf/O
2.533	1.723	tNET	RR	1	R2C9[0][A]	reg11_s0/CLK
2.933	0.400	tC2Q	RR	1	R2C9[0][A]	**reg11_s0/Q**
3.554	0.621	tNET	RR	1	R2C9[1][A]	**reg12_s0/CLEAR**

Data Required Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysyclk
0.000	0.000	tCL	RR	1	IOL11[A]	clk_ibuf/I
0.811	0.811	tINS	RR	2	IOL11[A]	clk_ibuf/O
2.533	1.723	tNET	RR	1	R2C9[1][A]	reg12_s0/CLK
2.533	0.000	tUnc				reg12_s0
2.551	0.018	tHld		1	R2C9[1][A]	reg12_s0

Path Statistics:

Clock Skew	0.000
Hold Relationship	0.000
Logic Level	1
Arrival Clock Path Delay	cell: 0.811, 31.998%; route: 1.723, 68.002%
Arrival Data Path Delay	cell: 0.000, 0.000%; route: 0.621, 60.818%; tC2Q: 0.400, 39.182%
Required Clock Path Delay	cell: 0.811, 31.998%; route: 1.723, 68.002%

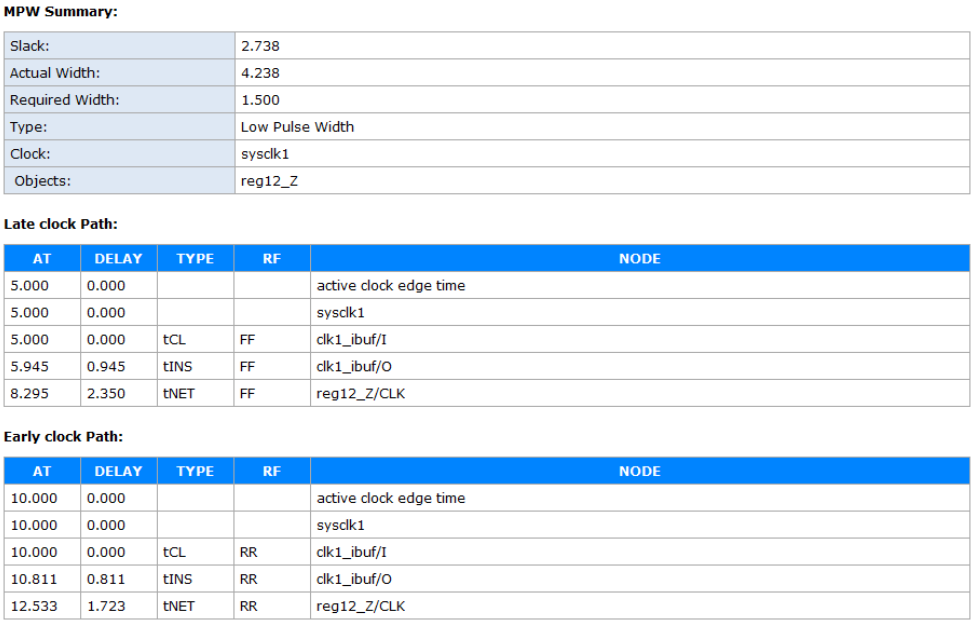
5.2.4 Minimum Pulse Width Report

最小パルス幅レポートです。High レベルの最小パルスと Low レベルの最小パルスを含む、タイミング解析に関係するパス上のすべてのシーケンシャル・エレメントの最小パルス幅を解析します。図 5-13 に示すとおりです。

- Actual Width：実際のパルス幅。この値は、解析対象で実際に維持されているパルス幅であり、つまり Early clock Path から Late clock Path を引いた値です。
- Required Width：必要な最小認識幅、つまりパルス信号が維持される最小時間。
- Slack：パルス幅のスラック。Actual Width から Required Width を引いた値です。
- Type：パルスのタイプ。Low Pulse Width と High Pulse Width の 2 つのタイプがあります。
- Clock：静的タイミング解析用のクロック。

- Objects：現在解析されているシーケンシャル・エレメント。
- Late clock Path：パルスの開始時に解析を開始するパス。High Pulse Width の場合は、論理 High 信号の開始時に解析を開始するパスです。Low Pulse Width の場合は、論理 Low 信号の開始時に解析を開始するパスです。
- Early clock Path：パルスの終了時に解析を開始するパス。High Pulse Width の場合は、論理 High 信号の終了時に解析を開始するパスです。Low Pulse Width の場合は、論理 Low 信号の終了時に解析を開始するパスです。

図 5-13 Minimum Pulse Width Report



MPW Summary:

Slack:	2.738
Actual Width:	4.238
Required Width:	1.500
Type:	Low Pulse Width
Clock:	sysclk1
Objects:	reg12_Z

Late clock Path:

AT	DELAY	TYPE	RF	NODE
5.000	0.000			active clock edge time
5.000	0.000			sysclk1
5.000	0.000	tCL	FF	clk1_ibuf/I
5.945	0.945	tINS	FF	clk1_ibuf/O
8.295	2.350	tNET	FF	reg12_Z/CLK

Early clock Path:

AT	DELAY	TYPE	RF	NODE
10.000	0.000			active clock edge time
10.000	0.000			sysclk1
10.000	0.000	tCL	RR	clk1_ibuf/I
10.811	0.811	tINS	RR	clk1_ibuf/O
12.533	1.723	tNET	RR	reg12_Z/CLK

5.2.5 High Fanout Nets Report

High Fanout Nets Report は、タイミング解析に関係するパス上の net のファンアウトを解析すると同時に、この net の最小 Slack と最大遅延を解析します。デフォルトでは 10 個解析されます。図 5-14 に示すとおりです(FANOUT 値の大きい順)。

- FANOUT：net のファンアウトを示します。
- NET NAME：net 名。
- WORST SLACK：net 上の最悪のスラック。net には複数のスラックが存在する可能性があります。
- MAX DELAY：net の最大遅延。

図 5-14 High Fanout Nets Report

High Fanout Nets Report:

Report Command:report_high_fanout_nets -max_nets 10

FANOUT	NET NAME	WORST SLACK	MAX DELAY
2	clk1_c	5.789	2.350
2	clk2_c	17.616	2.350
1	reg21_i	17.616	0.000
1	reg11	5.789	2.981
1	reg21	17.616	0.403

5.2.6 Route Congestions Report

図 5-15 は配線の密集レベルレポートです。

- GRID LOC：Grid の位置。
- ROUTE CONGESTIONS：Grid 上の配線の密集レベル。たとえば、0.056 は、5.6%の密集レベルを示します。
- デフォルトでは、最悪の 10 個が報告され、ROUTE CONGESTIONS 値の大きい順にリストされます。

図 5-15 Route Congestions Report

Route Congestions Report:

Report Command:report_route_congestion -max_grids 10

GRID LOC	ROUTE CONGESTIONS
R5C9	0.056
R2C1	0.028
R3C1	0.028
R3C9	0.028
R1C1	0.014
R5C1	0.014

5.2.7 Timing Exceptions Report

次に、実際のケースを使用してタイミング例外レポートを解説します。

図 5-16 のケースに合わせた特定の SDC ファイルを図 5-17 に示します。

図 5-16 テストケース

```
module timing(
output dout,
input din, clk1, clk2
);

reg reg11, reg12;
reg reg21, reg22;


always @(posedge clk1)
begin
    reg11 <= din;
    reg12 <= reg11;
end

always @(posedge clk2)
begin
    reg21 <= din;
    reg22 <= ~reg21;
end

assign dout = reg22 & reg12;

endmodule
```

図 5-17 Timing Exceptions 制約

```
create_clock -name sysclk1 -period 10 -waveform {0 5} [get_ports {clk1}]
create_clock -name sysclk2 -period 10 -waveform {0 5} [get_ports {clk2}]
set_max_delay -from [get_clocks {sysclk1}] -to [get_clocks {sysclk1}] 5
set_max_delay -from [get_clocks {sysclk2}] -to [get_clocks {sysclk2}] 4
```

図 5-17 に示すように、タイミング例外制約文 set_max_delay は、sysclk1 と sysclk2 の影響を受けるタイミングパスの最大絶対遅延値を、それぞれ 5ns と 4ns に設定します。set_max_delay は setup 解析に影響し、影響を受けるパスはデフォルトでタイミング例外レポートに表示されます。デフォルトのレポートを図 5-18 に示します。

図 5-18 Timing Exceptions Report

Timing Exceptions Report:

Setup Analysis Report

Report Command:report_exceptions -setup -max_paths 5 -max_common_paths 1

Timing Path Constraint[1]: set_max_delay -from [get_clocks {sysclk1}] -to [get_clocks {sysclk1}] 5

Path1

Path Summary:

Slack	0.789
Data Arrival Time	6.767
Data Required Time	7.556
From	reg11_Z
To	reg12_Z
Launch Clk	sysclk1:[R]
Latch Clk	sysclk1:[R]

Data Arrival Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysclk1
0.000	0.000	tCL	RR	1	IOL7[A]	clk1_ibuf/I
0.943	0.943	tINS	RR	2	IOL7[A]	clk1_ibuf/O
3.236	2.293	tNET	RR	1	IOL2[B]	reg11_Z/CLK
3.786	0.550	tC2Q	RF	1	IOL2[B]	**reg11_Z/Q**
6.767	2.981	tNET	FF	1	R5C9[1][A]	**reg12_Z/D**

Data Required Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
5.000	5.000					active clock edge time
5.000	0.000					sysclk1
5.000	0.000	tCL	RR	1	IOL7[A]	clk1_ibuf/I
5.943	0.943	tINS	RR	2	IOL7[A]	clk1_ibuf/O
8.236	2.293	tNET	RR	1	R5C9[1][A]	reg12_Z/CLK
8.036	-0.200	tUnc				reg12_Z
7.556	-0.480	tSu		1	R5C9[1][A]	reg12_Z

Path Statistics:

Clock Skew	0.000
Setup Relationship	5.000
Logic Level	1
Arrival Clock Path Delay	cell: 0.943, 29.131%; route: 2.293, 70.869%
Arrival Data Path Delay	cell: 0.000, 0.000%; route: 2.981, 84.423%; tC2Q: 0.550, 15.577%
Required Clock Path Delay	cell: 0.943, 29.131%; route: 2.293, 70.869%

Timing Path Constraint[14]: set_max_delay -from [get_clocks {sysclk2}] -to [get_clocks {sysclk2}] 4

Path1

Path Summary:

Slack	1.616
Data Arrival Time	4.940
Data Required Time	6.556
From	reg21_Z
To	reg22_Z
Launch Clk	sysclk2:[R]
Latch Clk	sysclk2:[R]

Data Arrival Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysclk2
0.000	0.000	tCL	RR	1	IOL5[A]	clk2_ibuf/I
0.943	0.943	tINS	RR	2	IOL5[A]	clk2_ibuf/O
3.236	2.293	tNET	RR	1	R5C9[0][B]	reg21_Z/CLK
3.786	0.550	tC2Q	RR	1	R5C9[0][B]	**reg21_Z/Q**
4.189	0.403	tNET	RR	1	R5C9[0][A]	reg21_i_cZ/I0
4.940	0.751	tINS	RF	1	R5C9[0][A]	reg21_i_cZ/F
4.940	0.000	tNET	FF	1	R5C9[0][A]	**reg22_Z/D**

Data Required Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
4.000	4.000					active clock edge time
4.000	0.000					sysclk2
4.000	0.000	tCL	RR	1	IOL5[A]	clk2_ibuf/I
4.943	0.943	tINS	RR	2	IOL5[A]	clk2_ibuf/O
7.236	2.293	tNET	RR	1	R5C9[0][A]	reg22_Z/CLK

タイミング例外レポートは、デフォルトでタイミング例外制約文の影響を受けるすべてのパスを報告します。Gowin ソフトウェアは、ユーザーが必要なレポートの一部のコンテンツを構成および表示し、不要なレポートパスをフィルターすることを可能にする report_exception 制約コマン

ドを提供します。図 5-19 に示すように、赤いボックスの最初の行では、sysclk1 の影響を受けるパスの setup 解析が報告され、2 番目の行では、sysclk2 の影響を受けるパスの setup 解析が報告されません。

図 5-19 report_exception 文

```
create_clock -name sysclk1 -period 10 -waveform {0 5} [get_ports {clk1}]
create_clock -name sysclk2 -period 10 -waveform {0 5} [get_ports {clk2}]
set_max_delay -from [get_clocks {sysclk1}] -to [get_clocks {sysclk1}] 5
set_max_delay -from [get_clocks {sysclk2}] -to [get_clocks {sysclk2}] 4
report_exceptions -setup -from_clock [get_clocks {sysclk1}] -to_clock [get_clocks {sysclk1}] -max_paths 1 -max_common_paths 1
report_exceptions -setup -from_clock [get_clocks {sysclk2}] -to_clock [get_clocks {sysclk2}] -max_paths 0 -max_common_paths 0
```

図 5-19 の制約後のタイミング例外レポートを図 5-20 に示します。

図 5-20 report_exception レポート

Timing Exceptions Report:

Setup Analysis Report

Setup Analysis Report[1]:

Report Command:report_exceptions -setup -from_clock [get_clocks {sysclk1}] -to_clock [get_clocks {sysclk1}] -max_paths 1 -max_common_paths 1

Timing Path Constraint[1]: set_max_delay -from [get_clocks {sysclk1}] -to [get_clocks {sysclk1}] 5

Path1

Path Summary:

Slack	-0.654
Data Arrival Time	7.947
Data Required Time	7.293
From	reg11_ins23
To	reg12_ins20
Launch Clk	sysclk1:[R]
Latch Clk	sysclk1:[R]

Data Arrival Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysclk1
0.000	0.000	tCL	RR	1	IOL15[A]	clk1_ibuf13/I
0.982	0.982	tINS	RR	2	IOL15[A]	clk1_ibuf13/O
2.893	1.911	tNET	RR	1	IOL2[B]	reg11_ins23/CLK
3.351	0.458	tC2Q	RF	1	IOL2[B]	**reg11_ins23/Q**
7.947	4.596	tNET	FF	1	R15C23[1][A]	**reg12_ins20/D**

Data Required Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
5.000	5.000					active clock edge time
5.000	0.000					sysclk1
5.000	0.000	tCL	RR	1	IOL15[A]	clk1_ibuf13/I
5.982	0.982	tINS	RR	2	IOL15[A]	clk1_ibuf13/O
7.893	1.911	tNET	RR	1	R15C23[1][A]	reg12_ins20/CLK
7.693	-0.200	tUnc				reg12_ins20
7.293	-0.400	tSu		1	R15C23[1][A]	reg12_ins20

Path Statistics:

Clock Skew	0.000
Setup Relationship	5.000
Logic Level	1
Arrival Clock Path Delay	cell: 0.982, 33.942%; route: 1.911, 66.058%
Arrival Data Path Delay	cell: 0.000, 0.000%; route: 4.596, 90.932%; tC2Q: 0.458, 9.068%
Required Clock Path Delay	cell: 0.982, 33.942%; route: 1.911, 66.058%

5.2.8 Timing Constraints Report

図 5-21 はタイミング制約レポートです。

- SDC Command Type：静的タイミング制約コマンドのタイプ。
- State：Invalid と Actived の 2 つの値があります。Actived はコマンドが有効であることを示し、Invalid はコマンドが無効であることを示します。
- Detail Command：SDC ファイル内の対応するタイミング制約文で

す。

図 5-21 Timing Constraints Report

Timing Constraints Report:

SDC Command Type	State	Detail Command
TC_CLOCK	Actived	create_clock -name main -period 18.182 -waveform {0 9.091} [get_ports {clk}]
TC_GENERATED_CLOCK	Actived	create_generated_clock -name main_gen -source [get_ports {clk}] -master_clock main -divide_by 5 -duty_cycle 40 -phase 22 -offset 50 [get_ports {in}]
TC_INPUT_DELAY	Actived	set_input_delay -clock main_gen 0.2 -clock_fall -add_delay -source_latency_included [get_ports {in}]
TC_CLOCK_LATENCY	Actived	set_clock_latency -source 1.2 [get_clocks {main}]
TC_CLOCK_UNCERTAINTY	Actived	set_clock_uncertainty 2.3 -setup -from [get_clocks {main}] -to [get_clocks {main}]
TC_FALSE_PATH	Actived	set_false_path -from [get_clocks {main_gen}] -to [get_clocks {main_gen}]
TC_MULTICYCLE	Actived	set_multicycle_path -from [get_clocks {main_gen}] -to [get_clocks {main_gen}] -setup -end 3
TC_MAX_DELAY	Actived	set_max_delay -from [get_clocks {main}] -to [get_clocks {main}] 1.11
TC_CLOCK_GROUP	Actived	set_clock_groups -exclusive -group [get_clocks {main}] -group [get_clocks {main_gen}]

付録 A.タイミング制約構文仕様

Gowin のタイミング制約構文仕様は、標準の SDC（Synopsys Design Constraint）構文形式を参照しています。タイミング制約を実行することにより、特定のタイミング要件を満たすことができます。

ワイルドカード文字「？」と「*」の使用がサポートされています。「?」は 1 文字の一致を実現し、「*」は 0 文字以上の一致(階層間の一致をサポート)を実現します。また、複数の行に分割されるタイミング制約もサポートされています。

A.1 クロック制約

A.1.1 create_clock

構文

コマンド：create_clock

パラメータ：-period <period_value>

[-name <clock_name>]

[-waveform <edge_list>]

<objects>

[-add]

-period：クロックサイクルを指定します。0 より大きい数(単位は ns)に設定する必要があります。

-name：クロック名を指定します。このパラメータは、クロックの一意の識別マークであるため、重複した名前のクロックを作成してはなりません。そうすると、後で作成されたクロックによって、最初に作成されたクロックが上書きされます。このパラメータが指定されていない場合、クロックのデフォルト名は source objects の最初のエレメント名となりま

す。

-waveform：クロックの立ち上がりエッジと立ち下がりエッジの時間を指定します。この 2 つの時間は次第に増える正数となり、かつ両者の差は 1 クロックサイクル未満です。通常、立ち上がりエッジが先に到着すると指定した場合、立ち上がりエッジと立ち下がりエッジの時間はいずれも 1 クロックサイクル未満に設定します。例えば「{0 5}」はこのクロックの立ち上がりエッジが 0ns の時先に到着し、立ち下がりエッジが 5ns の時到着することを表します。クロックの立ち下がりエッジが先に到着する場合、立ち上がりエッジ時間を 1 クロックサイクル未満、立ち下がりエッジ時間を 1 クロックサイクル以上にします。例えば周期を 10ns に設定すると、「-waveform {5 10}」はこのクロックの立ち下がりエッジが 0ns の時に到着し、立ち上がりエッジが 5ns の時に到着することを表します。

-add：複数のクロックを同じソースに追加する場合、2 番目以降のクロック作成文で-add パラメータを使用する必要があります。そうしない場合、2 番目以降のクロック作成文は無視されます(クロックは正常に作成されません)。

<objects>：クロックのオブジェクトを指定します。コレクション get_ports、get_pins、get_nets、および get_regs をサポートします。ユーザーの選択したクロックソースにクロックが作成済みの場合、ユーザーは-add コマンドで新しいクロックを作成できます。ユーザーが-add コマンドを使用しない場合、Gowin ソフトウェアはこのコマンドを無視し、新しいクロックは作成されません。ユーザーが create_clock コマンドを使用してクロックを作成する時にオブジェクトを指定しない場合、Gowin ソフトウェアはこのコマンドを無視し、クロックは正しく作成されません。

例

#「?」を使用して 1 文字の一致(例えば、clk、cck)を実現します。

```
create_clock -name ck -period 100 -waveform {0 50} [get_ports {c?k}]
```

#「*」を使用して 0 文字以上の一致(例えば、clk、clock)を実現します。

```
create_clock -name ck -period 100 -waveform {0 50} [get_ports {c*k}]
```

#「*」を使用して一致(例えば、uut/rpll_inst/CLKOUT)を実現します。

```
create_clock -name cck0 -period 25 -waveform {0 12.5} [get_pins {uut/r*_inst/CLKOUT}]
```

#エスケープ文字の使用例。

```
create_clock -name cck -period 25 -waveform {0 12.5} [get_pins {uut/¥*ll_inst/CLKOUT}]
```

```
create_clock -name cck -period 25 -waveform {0 12.5} [get_pins {¥?ut/rpll_inst/CLKOUT}]
```

周期 10ns、立ち下がりエッジが先に到着するクロックを作成します。デフォルトは、clk です。

create_clock -period 10.000 -waveform {5 10} [get_ports {clk}]

デューティサイクルが 40%のクロックを作成します。

create_clock -name clk -period 10.000 -waveform {6 10} [get_ports {clk}]

#同じポートに 2 つの有効クロックを追加します。

create_clock -period 10 -name clk [get_ports {clk}] #クロック clk は正常に作成されます

create_clock -period 10 -name clk1 [get_ports {clk}] #-add パラメータがないため、コマンド 2 は無視され、クロック clk1 は作成されません。

create_clock -period 20 -name clk1 -add [get_ports {clk}]#クロック clk1 は正常に作成されます

#-add を使用することにより、4 つの入力を持つ DCS のクロック出力ポートに 4 つのクロックを作成します。

create_clock -name clk0 -period 10 -waveform {0 5} [get_pins {dcs_inst/CLKOUT}]

create_clock -name clk1 -period 10 -waveform {0 4} [get_pins {dcs_inst/CLKOUT}] –add

create_clock -name clk2 -period 10 -waveform {0 3} [get_pins {dcs_inst/CLKOUT}] –add

create_clock -name clk3 -period 10 -waveform {0 2} [get_pins {dcs_inst/CLKOUT}] –add

A.1.2 create_generated_clock

構文

コマンド：create_generated_clock

パラメータ：[-name <clock name>]

-source <master pin>

[-edges <edge list>]

[-edge_shift <shift list>]

[-divide_by <factor>]

[-multiply_by<factor>]

[-duty_cycle <percent>]

[-add]

[-invert]

[-master_clock <clock>]

[-phase <phase>]

[-offset <offset>]

<objects>

-name：派生クロックの名前を指定します。このパラメータが未指定の場合、1 つ目の「source object」を派生クロックの名前とします。派生クロック名は一意でなければならず、派生クロック名がすでに存在する場合、以前作成された同名のクロックは上書きされます。

-source：派生クロックのソースを指定します。ソースに複数のクロックが存在する場合、「-master_clock」で特定のマスタークロックを指定します。コレクション get_ports、get_pins、get_nets、および get_regs をサポートします。

-master_clock：派生クロックに対応するマスタークロックを指定します。

-edges：派生クロックのクロックエッジ時間を指定します。このパラメータリストは 3 つの次第に増える正整数で構成され、派生クロックの最初の立ち上がりエッジと立ち下がりエッジ、2 つ目の立ち上がりエッジとマスタークロックエッジの関係を表します。例えば、マスタークロックの最初の立ち上がりエッジを 1、最初の立ち下がりエッジを 2、2 つ目の立ち上がりエッジを 3 とし、順に計数すると、このパラメータを使用して 2 分周の派生クロックを作成する方法は「-edge {1 3 5}」です。

-edge_shift：このパラメータは、「-edges」パラメータと併用することで-edges パラメータによって設定されるエッジにシフトを追加します。その値は、エッジが隣接するエッジを超えないように設定する必要があります。

注記：

「-edge」と「-edge_shift」は、「-invert」を除く波形調整パラメータと同時に使用できません。

-divide_by：派生クロックのマスタークロックに対する分周数を設定します。

-multiply_by：派生クロックのマスタークロックに対する逓倍数を設定します。

-duty_cycle：派生クロックのデューティサイクルを設定します。

-add：同じソースのクロックを追加し、同時に有効にします。

-invert：このパラメータを使用すると、派生クロックを反転できます。Gowin ソフトウェアは 1/2 サイクルをシフトすることで反転を実現します。

-phase：マスタークロックのクロックエッジのオフセットを設定します(単位は度)。

-offset：派生クロックエッジのオフセットを設定します。

<objects>：クロックのオブジェクトを指定します。コレクション get_ports、get_pins、get_nets、および get_regs をサポートします。

例

「-divide_by」を用いてポート a で 2 分周の派生クロックを作成します。

create_clock -period 10 [get_ports clk]

create_generated_clock -name genClk -source [get_ports {clk}] -divide_by 2 [get_ports {a}]

「-edges」を用いてポート a で 2 分周の派生クロックを作成します。

create_generated_clock -name genClk -source [get_ports {clk}] -edges {1 3 5} [get_ports {a}]

#デューティサイクルが 40%の 2 分周派生クロックを作成します。

create_generated_clock -name genClk0 -source [get_ports {clk}] -multiply_by 2 -duty_cycle 40 [get_pins {pll_out}]

#マスタークロックの 2 分周反転の派生クロックを作成します。

create_generated_clock -name genClk1 -source [get_ports {clk}] -divide_by 2 -invert [get_pins {pll_out}]

2 分周かつシフトが 90 度の派生クロックを作成します。

create_generated_clock -name genClk2 -source [get_ports{clk}] -multiply_by 2 -phase 90[get_pins {pll_out}]

#2 分周の派生クロックを作成します。

create_generated_clock -name genClk3 -source [get_ports {clk}] -edges {2 4 6}[get_pins {pll_out}]

#同じソースでマスタークロックが異なる派生クロックのペアを作成します。

create_clock -period 10 -name clk [get_ports {clk}]

create_clock -period 20 -name clk1 -add [get_ports {clk}]

create_generated_clock -name genClk -source [get_ports {clk}] -divide_by 2 -master_clock clk -add [get_pins {pll_out}]

create_generated_clock -name genClk1 -source [get_ports {clk}] -master_clock clk1 -divide_by 2 -add [get_pins {pll_out}]

A.1.3 set_clock_latency

構文

コマンド：set_clock_latency

パラメータ：-source [-rise | -fall]

[-late | -early]

<delay>

[-clock <clock list>]

<object list>

-source：クロックソース遅延を示します。必須項目です。

-rise | -fall：立ち上がりエッジまたは立ち下がりエッジの遅延を指定します。これら 2 つのパラメータを同じ文に同時に使用することはできません。これら 2 つのパラメータが存在しない場合、立ち上がりエッジと立ち下がりエッジの遅延は両方ともこの文で指定された値に設定されます。

-late | -early：最大または最小遅延を指定します。セットアップ分析の場合、late は launch clock に作用し、early は latch clock に作用します。ホールド分析の場合、セットアップ分析の反対です。

<delay>：クロックソースの遅延値を設定します。デフォルト値は 0 です。

注記：

late の値は、early の値以上である必要があります。そうでない場合、late に early の値が割り当てられます。

-clock：複数のクロックを作成した時、このパラメータを使用してどのクロックに遅延を設定するかを決定する必要があります。このパラメータを設定しない場合、すべてのクロックに同じレイテンシを設定します。コレクション get_clocks をサポートします。

<source objects>：どのクロックのアクセスポイント、またはどのクロックに遅延の設定をするか指定します。コレクション get_clocks、get_ports、get_pins、get_nets、および get_regs をサポートします。

例

create_clock -period 10 -name clk [get_ports {clk}]

create_clock -period 10 -name clk0 [get_ports {clk}] -add

clk に 2ns のクロック遅延を指定します。

set_clock_latency -source 2 [get_clocks {clk}]

クロックポート上の clk0 にクロック遅延を指定します。

set_clock_latency -source 2 -clock [get_clocks {clk0}] [get_ports {clk}]

#clk0 上のクロック cck の立ち上がりエッジのクロックソース遅延を設定し、late 値と early 値をそれぞれ 0.111 と 0.011 に指定します。

set_clock_latency -source -rise -late 0.111 [get_ports {clk0}] -clock [get_clocks {cck}]

set_clock_latency -source -rise -early 0.011 [get_ports {clk0}] -clock [get_clocks {cck}]

#clk0 上のクロック cck の立ち下がりエッジのクロックソース遅延を設定し、late 値と early 値をそれぞれ 0.222 と 0.022 に指定します。

set_clock_latency -source -fall -late 0.222 [get_ports {clk0}] -clock [get_clocks {cck}]

set_clock_latency -source -fall -early 0.022 [get_ports {clk0}] -clock [get_clocks {cck}]

#ワイルドカードを使用して一致(例えば：、uut/rpll_inst/CLKOUT)を実現します。

set_clock_latency -source 0.123 [get_pins {u?t/r*_inst/¥*OUT}] -clock [get_clocks {cck0}]

A.1.4 set_clock_uncertainty

構文

コマンド：set_clock_uncertainty

パラメータ：[-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>]

[-setup | -hold]

<uncertainty value>

-from/-rise_from/-fall_from：ばらつきの送信側のクロックを指定します。「-rise_from」と「-fall_from」でこのばらつきの有効クロックエッジを指定できます。コレクション get_clocks をサポートします。

-to/-rise_to/-fall_to：ばらつきの受信側のクロックを指定します。「-rise_to」と「-fall_to」でこのばらつきの有効クロックエッジを指定できます。コレクション get_clocks をサポートします。

-from/-rise_from/-fall_from：ばらつきがセットアップ時間またはホールド時間のどちらに影響するかを指定します。同じ制約文は相互に排他的です。どちらも指定されていない場合、両方のチェックが有効です。

<uncertainty value>：ばらつき値。

注記：

少なくとも 1 つの launch クロックまたは latch クロックを指定する必要があります。指定しない場合、制約は無効です。

例

clk から clk のセットアップ時間のばらつきを 0.5 として設定します。

set_clock_uncertainty -setup -from clk -to clk 0.5

#clk0 から clk のホールド時間のばらつきを 0.0 として設定します。

set_clock_uncertainty -hold -from clk0 -to clk 0.0

#launch クロックが clk0 の場合のホールド時間、セットアップ時間のばらつきを 0.111、0.222 に設定します。

set_clock_uncertainty 0.222 -setup -from [get_clocks {clk0}]

set_clock_uncertainty 0.111 -hold -from [get_clocks {clk0}]

#latch クロックが clk1 の場合のホールド時間、セットアップ時間のばらつきを 0.111、0.222 に設定します。

set_clock_uncertainty 0.222 -setup -to [get_clocks {clk1}]

set_clock_uncertainty 0.111 -hold -to [get_clocks {clk1}]

#launch クロックが clk の場合のホールド時間、セットアップ時間のばらつきを 0.111 に設定します。

set_clock_uncertainty 0.111 -from [get_clocks {clk}]

A.1.5 set_clock_groups

構文

コマンド：set_clock_groups

パラメータ：[-asynchronous | -Exclusive]

[-group <clock name>] ...

-asynchronous | -Exclusive：クロック間の関係を非同期または相互に排他的に指定します。

-group：クロックを同じグループとして指定します。コレクション get_clocks を使用して 1 つ以上のクロックを収集することをサポートします。

例

クロック clk とクロック clk0 の関係を排他的に設定します。

set_clock_groups -exclusive -group [get_clocks {clk}] -group [get_clocks {clk0}]

A.2 I/O 遅延の制約

A.2.1 set_input_delay

構文

コマンド：set_input_delay

パラメータ：-clock clock_name

[-clock_fall]

[-rise]

[-fall]

[-max]

[-min]

[-add_delay]

[-source_latency_included]

<delay_value>

<port_list>

-clock：関連するクロックを指定します。

-clock_fall：この入力ポートとクロックの立ち下がりエッジの関連付けを示します。このパラメータがない場合、デフォルトはクロックの立ち上がりエッジと関連付けられます。

-rise/-fall：立ち上がりエッジまたは立ち下がりエッジデータの入力遅延を指定します。1 つだけを指定すると、もう 1 つに同じ値が自動的に割り当てられます。

-max/-min：データの最大または最小入力遅延を指定します。それぞ

れ setup と hold に影響します。1 つだけを指定すると、もう 1 つに同じ値が自動的に割り当てられます。

-add_delay：このタイプの複数の制約を同時に有効にします。

-source_latency_included：外部クロック遅延が既に入力遅延に含まれていることを表します。指定しない場合、外部クロック遅延は入力遅延に含まれません。

<delay_value>：入力遅延値を指定します。デフォルトは 0ns です。

<port_list>：制約される入力ポート(PORT)を指定します。コレクション get_ports をサポートします。

例

ポート a の clk 立ち上がりエッジに基づく入力遅延を 0.8ns に設定します。

```
set_input_delay -clock clk 0.8 [get_ports {a}]
```

すべての入力ポートの clk 立ち上がりエッジに基づく遅延を 0.8 に設定します。

```
set_input_delay -clock clk 0.8 [all_inputs]
```

ポート a の clk 立ち下がりエッジに基づく入力遅延を 0.8ns に設定します。

```
set_input_delay -clock clk -clock_fall 0.8 [get_ports {a}]
```

ポート a の clk 立ち上がりエッジに基づく 4 タイプの遅延を設定します。

```
set_input_delay -clock clk -max -rise 1.4 [get_ports {a}]
set_input_delay -clock clk -max -fall 1.5 [get_ports {a}]
set_input_delay -clock clk -min -rise 0.7 [get_ports {a}]
set_input_delay -clock clk -min -fall 0.8 [get_ports {a}]
#-add_delay の使用例。
set_input_delay -clock clk1 -max 1.5 [get_ports {a}]
set_input_delay -clock clk1 -max 2.5 -add_delay [get_ports {a}]
#ワイルドカードの使用例。
set_input_delay -clock cck0 -max 1.4 [get_ports {d*}]
```

A.2.2 set_output_delay

構文

コマンド：set_output_delay

パラメータ：-clock clock_name

[-clock_fall]

[-rise]

[-fall]

[-max]

[-min]

[-add_delay]

[-source_latency_included]

<delay_value>

<port_list>

-clock：出力遅延に関するクロックを指定します。

-clock_fall：この出力遅延とクロックの立ち下がりエッジの関連付けを指定します。指定しない場合、デフォルトで立ち上がりエッジに関連付けられます。

-rise/-fall：立ち上がりエッジまたは立ち下がりエッジに基づく出力遅延を指定します。1 つだけを指定すると、もう 1 つに同じ値が自動的に割り当てられます。

-max/-min：データの最大または最小出力遅延を指定します。それぞれ setup と hold に影響します。1 つだけを指定すると、もう 1 つに同じ値が自動的に割り当てられます。

-add_delay：このタイプの複数の制約を同時に有効にします。

-source_latency_included：外部クロック遅延が既に出力遅延に含まれていることを表します。

<delay_value>：出力遅延値を指定します。デフォルトは 0ns です。

<port_list>：制約される出力ポート(PORT)を指定します。コレクション get_ports をサポートします。

例

ポート b の外部出力遅延を 0.5ns に設定します。

set_output_delay -clock clk 0.5 [get_ports {b}]

すべての出力ポートの外部出力遅延を 0.5ns に設定します。

set_output_delay -clock clk 0.5 [all_outputs]

#ポート b のクロック立ち下がりエッジに基づく外部出力遅延を 0.5ns に設定します。

```
set_output_delay -clock clk -clock_fall 0.5 [get_ports {b}]
```

ポート b のクロック立ち上がりエッジに基づく外部出力遅延を設定します。

```
set_output_delay -clock clk -max -rise 0.3 [get_ports {b}]
set_output_delay -clock clk -max -fall 0.5 [get_ports {b}]
set_output_delay -clock clk -min -rise 0.8 [get_ports {b}]
set_output_delay -clock clk -min -fall 0.7 [get_ports {b}]
```

パラメータ「-add_delay」によって異なるクロックエッジに基づく外部出力遅延を同時に有効にします。

```
set_output_delay -clock clk0 -min 0.5 [get_ports {b}]
set_output_delay -clock clk0 -max 0.6 [get_ports {b}]
set_output_delay -clock clk0 -clock_fall 0.7 -add_delay [get_ports {b}]
set_output_delay -clock clk1 -min 0.8 -add_delay [get_ports {b}]
set_output_delay -clock clk1 -max 0.9 -add_delay [get_ports {b}]
```

A.3 タイミングパスの制約

A.3.1 set_max_delay / set_min_delay

構文

コマンド：set_max_delay

パラメータ：[-from <from list>]

[-to <to list>]

[-through <through_list>]

<delay value>

コマンド：set_min_delay

パラメータ：[-from <from list>]

[-to <to list>]

[-through <through_list>]

<delay value>

-from：パスの始点を指定します。コレクション get_clocks、get_ports、get_regs、get_pins をサポートします。

-to：パスの終点を指定します。コレクション get_clocks、get_ports、get_regs、get_pins をサポートします。

-through：パスの通過点を指定します。コレクション get_pins と get_nets をサポートします。このパラメータを使用してピン(PIN)を収集する場合、このピンは非シーケンシャル・エレメントのピン(PIN)である必要があります。同一の制約では複数の「-through」パラメータを使用することはできません。

<delay value>：出力遅延値を指定します。

注記：

- set_max_delay 制約は setup 解析に影響し、set_min_delay 制約は hold 解析に影響します。
- 以上の 3 つのパラメータは併用するか、単独で使用することができます。この 3 つのパラメータが指定する基本ユニットが同じパスにない時、Gowin ソフトウェアはこの制約を無視し、タイミング計算に影響を及ぼしません。

例

#clk0 により駆動されるエレメントから clk1 により駆動されるエレメントまでのタイミングパスの最小遅延を 0.5ns に設定します。

set_max_delay -from [get_clocks {clk0}] -to [get_clocks {clk1}] 5

#ワイルドカードの使用例。

set_max_delay -from [get_ports {d*}] -to [get_regs {r?}] 2

#入力ポートをピンに制約することは、setup 解析に影響し、出力ポートをピンに制約することは、hold 解析に影響します。

set_max_delay -from [all_inputs] -to [get_pins {r*/D}] 1.234

set_max_delay -from [get_pins {r?_s0/CLK}] -to [all_outputs] 0.989

#すべての、クロックにより駆動されるシーケンシャル・エレメントの最大遅延を 5ns に設定します。

set_max_delay -from [all_clocks] 5 -to [get_ports {out*}]

#ポート a からポート b の最大遅延を 2ns に設定します。

set_max_delay -from [get_ports {a}] -to [get_ports {b}] 2

#フリップフロップ reg0 から clk 立ち下がりエッジで駆動されるシーケンシャル・エレメントまでの最大遅延を 2ns に設定します。

set_max_delay -from [get_regs {reg0}] -to [get_clocks {clk}] 2

#clock により駆動されるエレメントから clock により駆動されるエレメントまでのタイミングパスの最小遅延を 0.5ns に設定します。

set_min_delay -from [get_clocks {clk}] -to [get_clocks {clk}] 0.5

#ポート a からフリップフロップ reg0 の最小遅延を 0.5ns に設定します。

set_min_delay -from [get_ports {a}] -to [get_regs {reg0}] 0.5

フリップフロップ reg0 からポート b の最小遅延を 0.5ns に設定します。

set_min_delay -from [get_regs {reg0}] -to [get_ports {b}] 0.5

ポート a からポート b の最小遅延を 0.5ns に設定します。

set_min_delay -from [get_ports {a}] -to [get_ports {b}] 0.5

#ポート a からクロック clk までの遅延、すべてのデータポートから関連クロックまでの遅延を設定します。

set_max_delay -from [get_ports {a}] -to [get_clocks {clk}] 0.5

set_max_delay -from [all_inputs] -to [all_clocks] 0.111

A.3.2 set_false_path

構文

コマンド：set_false_path

パラメータ：[-from <from list>]

[-to <to list>]

[-through <through list>]

[-setup]

[-hold]

-setup/-hold：現在の制約がセットアップ時間のチェックまたはホールド時間のチェックのどちらに有効であるかを指定します。この 2 つのパラメータは相互に排他的です。指定されていない場合、setup と hold の両方に有効です。

-from：パスの始点を指定します。コレクション get_ports、get_regs、ger_pins、および get_clocks をサポートします。Gowin ソフトウェアは関連する終点を自動的に取得するので、単独で使用できます。

-to：パスの終点を指定します。コレクション get_ports、get_regs、ger_pins、および get_clocks をサポートします。Gowin ソフトウェアは関連する始点を自動的に取得するので、単独で使用できます。

-through：パスの通過点またはネットを指定します。コレクション get_pins または get_nets で通過点またはネットを収集できます。このパラメータリストでは複数のピン(PIN)またはネット(NET)を指定できます。これらは同じパスまたは異なるパスにあります。同一の制約では複数の「-through」パラメータを使用できません。

注記：

コレクション get_pins を使用する場合、-from の値はクロックピン、-to の値は非クロックピン、-through の値はデータパスの出力ピン(DFF.Q など)または入力ピン(DFF.D、DFF.CE など)である必要があります。

例

#クロック clk0 とクロック clk1 により駆動されるパスの場合のタイミング解析を行わないように設定します。

set_false_path -from [get_clocks {clk0}] -to [get_clocks {clk1}]

#フリップフロップ reg0 からフリップフロップ reg1 のパスの場合のタイミング解析を行わないように設定します。

set_false_path -from [get_regs {reg0}] -to [get_regs {reg1}]

#クロック clk の立ち上がりエッジの駆動からクロック clk1 の立ち下がりエッジの駆動までのパスの場合のタイミング解析を行わないように設定します。

set_false_path –from [get_clocks {clk}] -to [get_clocks {clk1}]

#ポート a からポート b のパスの場合のタイミング解析を行わないように設定します。

set_false_path －from [get_ports {a}] to [get_ports {b}]

#-from を単独で使用します。setup と hold の両方に対して有効です。

set_false_path -from [get_pins {reg0_s0/CLK}]

set_false_path -from [get_regs {reg0_s0}]

set_false_path -from [get_clocks {cck}]

#-to を単独で使用します。setup に対して有効です。

set_false_path -from [get_regs {reg0_s0}] －setup

#-to を単独で使用します。hold に対して有効です。

set_false_path -from [get_regs {reg0_s0}] －hold

#-through を単独で使用します。reg0_s0.Q を通るタイミングパスは解析されなくなります。

set_false_path - through [get_pins {reg0_s0/Q}]

#-through を単独で使用します。reg0_c を通るタイミングパスは解析されなくなります。

set_false_path - through [get_nets {reg0_c}]

#」*」の使用例。

set_false_path -from [get_regs {mi/r*0}] -to [get_regs {spi/R*}]

#」?」の使用例。

set_false_path -from [get_pins {mi/r?g0/CLK}] -to [get_pins {spi/DI}]

A.3.3 set_multicycle_path

構文

コマンド：set_multicycle_path

パラメータ：[-setup|-hold]

[-start|-end]

[-from <from_list>]

[-to <to list>]

[-through <through_list>]

<path multiplier>

-start/-end：この制約のリファレンスクロックが開始クロック(launch clock)またはラッチクロック(latch clock)であるかを指定します。パラメータ「-start」は開始クロック(launch clock)を指定し、パラメータ「-end」はラッチクロック(latch clock)を指定します。デフォルトではラッチクロック(latch clock)です。

-setup/-hold：現在の制約がセットアップ時間のチェックまたはホールド時間のチェックのどちらに影響するかを指定します。この 2 つのパラメータは相互に排他的です。デフォルトではセットアップ時間のチェックに影響します。

-from：パスの始点を指定します。コレクション get_pins、get_ports、get_regs、および get_clocks をサポートします。

-to：パスの終点を指定します。コレクション get_pins、get_ports、get_regs、および get_clocks をサポートします。

-through：パスの通過点またはネットを指定します。コレクション get_pins または get_nets で通過点またはネットを収集できます。このパラメータリストでは複数のピン(PIN)またはネット(NET)を指定できます。これらは同じパスまたは異なるパスにあります。同一の制約では複数の「-through」パラメータを使用できません。

<path multiplier>：サイクル数を指定します。

注記：

「-from」、「-to」、「-through」の 3 つのパラメータは併用するか、単独で使用することができます。この 3 つのパラメータが指定する点が同じパスにない時、Gowin ソフトウェアはこの制約を無視し、タイミング解析に影響を及ぼしません。

例

create_clock -name clk -period 10 [get_ports {clk}]

create_generated_clock -name genClk -multiply_by 2 -source [get_ports {clk}] [get_pins {pll_out}]

#マルチサイクルパスを設定：リファレンスクロックは genClk。セットアップ時間のチェックに影響します。

set_multicycle_path -end -setup -from [get_clocks {clk}] -to [get_clocks {genClk}] 2

#マルチサイクルパスを設定：リファレンスクロックはフリップフロップ reg0 のクロック。セットアップ時間とホールド時間のチェックに影響します。

set_multicycle_path -start -setup -from [get_regs {reg0}] -to [get_regs {reg1}] 3

set_multicycle_path -start -hold -from [get_regs {reg0}] -to [get_regs {reg1}] 1

#マルチサイクルパスを設定：リファレンスクロックは clk0。ソースクロック clk 立ち上がりエッジの駆動から clk0 立ち下がりエッジの駆動までのパスにのみ影響します。

set_multicycle_path -end -setup -from [get_clocks {clk}] -to [get_clocks {clk0}] 3

#ワイルドカード」?」と」*」の使用例。

set_multicycle_path -from [get_regs {SD/addr? }] -to [get_regs {RSG/D*_s0}

A.4 動作条件の制約

構文

コマンド：set_operating_conditions

パラメータ：[-grade <c|i|a>]

[-model <slow|fast>]

[-speed <speed>]

[-setup]

[-hold]

[-max]

[-min]

[-max_min]

-grade：デバイスの温度グレードを指定します。現在コマーシャル(commercial)、インダストリアル(industrial)、およびオートモーティブ(automotive)がサポートされます。

-model：タイミング解析のモデルを指定します。

-speed：デバイスのスピードグレードを指定します。

-setup：現在のプロセスコーナーでセットアップ時間のチェックを行うように指定し、-max 機能と一致します。

-hold：現在のプロセスコーナーでホールド時間のチェックを行うように指定し、-min 機能と一致します。

-max：現在のプロセスコーナーでセットアップ時間のチェックを行うように指定し、-setup 機能と一致します。

-min：現在のプロセスコーナーでホールド時間のチェックを行うように指定し、-hold 機能と一致します。

-max_min：現在のプロセスコーナーでセットアップ、ホールド時間のチェックを行うように指定し、-setup と-hold 機能の同時の指定と一致します。

例

#インダストリアル・スピードグレード 6。高速モデル。setup と hold 解析に影響します。

set_operating_conditions -grade i -model fast -speed 6 -setup -hold

#コマーシャル・スピードグレード 7。低速モデル。setup と hold 解析に影響します。

set_operating_conditions -grade c -model slow -speed 7 -max_min

A.5 タイミングレポート内容の制約

A.5.1 report_timing

構文

コマンド：report_timing

パラメータ：[-setup|-hold|-recovery|-removal]

[-max_paths <value>]

[-max_common_paths < value >]

[-rise_from <rise_from_list>]

[-fall_from <fall_from_list>]

[-to <to list>]

[-rise_to <rise_to_list>]

[-fall_to <fall_to_list>]

[-through <through list>]

[-from_clock<from clok>]

[-fall_from_clock <from clok>]

[-rise_from_clock <from clok>]

[-to_clock <to clok>]

[-rise_to_clock <to clok>]

[-fall_to_clock <to clok>]

[-min_logic_level < value >]

[-max_logic_level < value >]

[-mod_ins {mod_ins1 mod_ins2 ...}]

-setup|-hold|-recovery|-removal：タイミングレポートのチェックタイプを指定します。相互に排他的です。

-max_paths：タイミングレポートの最大パス数を指定します。デフ

ォルトは 25 です。指定されたパスの数が指定数に達しない場合、指定されていないパスの最悪のパスでこの指定数を補完します。

-max_common_paths：同じ終点を共有するパスの最大数を指定します。

-from/-rise_from/-fall_from：タイミングレポートパスの始点を指定します。-rise/fall_from はクロックである必要があります。コレクション get_clocks をサポートします。単独で使用する場合、Gowin ソフトウェアは関連する始点を自動的に取得します。

-to /-rise_to /-fall_to：タイミングレポートパスの終点を指定します。-rise/fall_to はクロックである必要があります。コレクション get_clocks をサポートします。単独で使用する場合、Gowin ソフトウェアは関連する終点を自動的に取得します。

-through：タイミングレポートパスの通過点を指定します。コレクション get_nets、get_pins をサポートします。

-from_clock /-fall_from_clock /-rise_from_clock：タイミングレポートパスの始点関連クロックを指定します。コレクション get_clocks をサポートします。単独で使用する場合、Gowin ソフトウェアは関連する終点を自動的に取得します。

-to_ clock /-rise_to_ clock /-fall_to_clock：タイミングレポートパスの終点関連クロックを指定します。コレクション get_clocks をサポートします。単独で使用する場合、Gowin ソフトウェアは関連する始点を自動的に取得します。

-min_logic_level/-max_logic_level：レポートパスの logic level を制限します。

-mod_ins {mod_ins1 mod_ins2 ...}：複数のモジュールのインスタンスを指定します。スペースで区切ります。このパラメータを使用しない場合、デフォルトでデザイン全体のタイミングが報告されます。

例

#セットアップ時間のチェックを報告するように指定します。レポート数は 100。

report_timing -setup -max_paths 100 -max_common_paths 5

#レポートの始点は ck であり、終点はワイルドカードを使用します(例えば、r0、r1 など)。

report_timing -hold -rise_from [get_clocks {ck}] -to [get_pins {r*/D }]

report_timing -setup -fall_from [get_clocks {ck }] -to [get_regs {r*}]

#パスのロジックレベルの数を 2 として指定します。最大 2 つのパスを報告できます。同じ終点のパスは最大 1 つ報告できます。

report_timing -recovery -from_clock [get_clocks {cck0}] -to_clock [get_clocks {cck1}] -max_paths 2 -max_common_paths 1 -max_logic_level 2 -min_logic_level 2

#hold 解析は、インスタンス uut 内のタイミングのみを報告します

report_timing -hold -mod_ins {uut}

A.5.2 report_high_fanout_nets

構文

コマンド：report_high_fanout_nets

パラメータ：[-clock_regions]

[-slr]

[-ascending]

[-max_nets <max_net_value>]

[-min_fanout <min_fanout_value>]

[-max_fanout <max_fanout_value>]

-clock_regions：オプションのパラメータ。このパラメータが指定されている場合、レポートの範囲はシーケンシャル・エレメントのクロック入力に接続された NET に制限されます。

-slr：オプションのパラメータ。このパラメータが指定されている場合、レポートの範囲は、シーケンシャル・エレメントのリセット/セット入力(同期または非同期)に接続された NET に制限されます。

-ascending：オプションのパラメータです。このパラメータが指定された場合、net のファンアウト値は降順にソートされます。このパラメータが指定されていない場合、デフォルトでは昇順にソートされます。

-max_net：オプションのパラメータです。レポートの最大 net 数を指定します。指定されていない場合、デフォルトで最大 10 net が報告されます。

-min_fanout：オプションのパラメータです。このパラメータ値以上のファンアウト数を持つ net のファンアウト状況のみ報告するよう指定します。

-max_fanout：オプションのパラメータです。このパラメータ値以下のファンアウト数を持つ net のファンアウト状況のみ報告するよう指定します。

例

シーケンシャル・エレメントのリセット/セット入力に接続された

net、ファンアウト数は[1,15]、最大レポート数は 10。

report_high_fanout_Nets -slr -max_nets 10 -min_fanout 1 -max_fanout 15

最大 10 net のファンアウト状況を報告します。

report_high_fanout_Nets -max_nets 10

A.5.3 report_route_congestion

構文

コマンド：report_route_congestion

パラメータ：[-max_grids <max grids value>]

[-min_route_congestion <min route congestion value>]

[-max_route_congestion <max route congestion>]

[-LOC <position>]

-max_grids：オプションのパラメータです。レポートの最大 grid 数を指定し、このパラメータを指定しない場合、デフォルトでは 10 個の grid の密集レベルを報告します。

-Min_route_congestion：オプションのパラメータです。grid の密集レベルの最小値を報告するよう指定し、このパラメータを指定しない場合、デフォルトでは 0 です。

-max_route_congestion：オプションのパラメータです。grid の密集レベルの最大値を報告するよう指定し、このパラメータを指定しない場合、デフォルトでは 1 です。このパラメータの値は、min_route_congestion のパラメータ値以上でなければなりません。そうでない場合、警告メッセージが報告され、当該文は無視されます。

-LOC：オプションのパラメータです。grid の物理位置を報告し、1 つの grid を指定できます。例えば、R1C3 の場合、第 1 行、第 3 列の grid が報告されます。範囲を指定することもできます。例えば、R[1:3]C3 の場合、1 行目から 3 行目の 3 列目の grid が報告されます。R[1:3]C[1:3]の場合、1 行目から 3 行目の 1 列目から 3 列目の grid が報告されます。R1C[1:3]の場合、1 行目の 1 列目から 3 列目の grid が報告されます。

例

物理アドレスが第 1 から 5 行第 1 から 5 列の密集レベルが 0 から 0.5 の grid の密集レベルを報告します。最も密集レベルが高い 5 つのみが報告されます。

report_route_congestion -max_grids 5 -min_route_congestion 0 -

max_route_congestion 0.5 -LOC R[1:5]C[1:5]

A.5.4 report_min_pulse_width

構文

コマンド：report_min_pulse_width

パラメータ：[-nworst <nworst value>]

[-min_pulse_width <min pulse width value>]

[-max_pulse_width <max pulse width value>]

[-detail]

[get_regs {regIns name}]

-nworst：報告する最悪のパスの数を指定します。

-min_pulse_width：レポートのシーケンシャル・エレメントの最小パルス幅を指定します。

-max_pulse_width：レポートのシーケンシャル・エレメントの最大パルス幅を指定します。

-detail：このパラメータを指定すると、クロックパスが含まれる詳細が報告されます。指定しない場合は簡単なレポートが作成されます。

get_regs {regIns name}：レポートの対象を指定します。このオプションを指定しない場合、デフォルトではすべてのフリップフロップにパルス幅タイミング解析を行います。1 つ以上の reg を指定できます。

例

#パルス幅が 0.1～4 の最悪の 3 つのクロックパスの最小パルス幅の状況を詳しく報告します。

report_min_pulse_width -nworst 3 -min_pulse_width 0.1 -max_pulse_width 4 –detail

#パルス幅が 0.001～4 の最悪の 20 のクロックパスの最小パルス幅の状況を簡単に報告します。

report_min_pulse_width -nworst 20 -min_pulse_width 0.001 -max_pulse_width 4

A.5.5 report_max_frequency

構文

コマンド：report_max_frequency

パラメータ：-mod_ins {mod_ins1 mod_ins2 ...}

-mod_ins {mod_ins1 mod_ins2 ...}：複数のモジュールのインスタンスを指定します。スペースで区切ります。このパラメータの指定に関わらず、設計の最大周波数はすべて報告されます。

例

bsram0 の最大動作周波数を報告します

report_max_frequency -mod_ins {bsram0}

A.5.6 report_exceptions

構文

コマンド：report_exceptions

パラメータ：-setup|-hold | -recovery | removal

[-max_paths<number>]

[-max_common_paths< number >]

[-max_logic_level <number>]

[-min_logic_level <number>]

[-rise_from <rise_from_list>]

[-fall_from <fall_from_list>]

[-to <to list>]

[-rise_to <rise_to_list>]

[-fall_to <fall_to_list>]

[-through <through list>]

[-rise_through <rise_through_list>]

[-fall_through <fall_through_list>]

[-from_clock<from clock>]

[-fall_from_clock<from clock>]

[-rise_from_clock<from clock>]

[-to_clock<to clock>]

[-rise_to_clock<to clock>]

[-fall_to_clock<to clock>]

キーワードの名前、意味、および使用法は、report_timing のものと

同じです。例外制約によって生成されたパスを報告します。

例

#リカバリー時間パスを報告します

create_clock -name mm -period 10 -waveform {0 5} [get_ports {clk}]

set_max_delay -from [get_clocks {mm}] -to [get_clocks {mm}] 0.22

report_exceptions -recovery -from_clock [get_clocks {mm}] -to_clock [get_clocks {mm}] -max_paths 1 -max_common_paths 1

A.6 その他の制約

A.6.1 derive_clocks

構文

コマンド：derive_clocks

パラメータ：- freq < value>

-freq：グローバルの周波数、1200 以下の正の浮動小数点数、小数点以下 3 桁、単位は MHz。

例

#周波数 100MHz のグローバルのクロックを作成します。

derive_clocks -freq 100


GOWIN高云
智慧逻辑 定制未来