




# Gowin Virtual Input Output ツール ユーザーガイド

SUG1189-1.1J, 2024-08-09

## 著作権について(2024)

著作権に関する全ての権利は、**Guangdong Gowin Semiconductor Corporation** に留保されています。

 **Gowin**、及び**LittleBee**は、当社により、中国、米国特許商標庁、及びその他の国において登録されています。商標又はサービスマークとして特定されたその他全ての文字やロゴは、それぞれの権利者に帰属しています。何れの団体及び個人も、当社の書面による許可を得ず、本文書の内容の一部もしくは全部を、いかなる視聴覚的、電子的、機械的、複写、録音等の手段によりもしくは形式により、伝搬又は複製をしてはなりません。

## 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
2024/06/28	1.0J	初版。
2024/08/09	1.1J	GAO と GVIO による共同デバッグの場合の画面のツールバーから「save」と「save as」ボタンを削除。

# 目次

目次 .....	i
図一覧 .....	ii
表一覧 .....	iv
<b>1 本マニュアルについて .....</b>	<b>1</b>
1.1 マニュアルの内容 .....	1
1.2 関連ドキュメント .....	1
1.3 用語、略語 .....	1
1.4 テクニカル・サポートとフィードバック .....	2
<b>2 概要 .....</b>	<b>3</b>
<b>3 GVIO 構成ファイル .....</b>	<b>5</b>
3.1 GVIO 構成ファイル .....	5
3.1.1 GVIO 構成ウィンドウの起動 .....	5
3.1.2 GVIO ファイルの構成 .....	7
3.1.3 ビットストリームファイルの生成 .....	15
<b>4 GVIO ツールの使用 .....</b>	<b>16</b>
4.1 GVIO ツールによる単独デバッグ .....	16
4.1.1 GVIO の起動 .....	16
4.1.2 GVIO の実行 .....	17
4.2 GVIO と GAO による共同デバッグ .....	22
4.2.1 GAO 構成ファイル .....	22
4.2.2 GVIO 構成ファイル .....	23
4.2.3 GAO と GVIO の実行 .....	23

## 図一覧

図 3-1 GVIO 構成ファイルの作成 .....	5
図 3-2 New GVIO Config File ダイアログボックス .....	6
図 3-3 GVIO ツールの構成ウィンドウ .....	6
図 3-4 「GVIO Core」 ビュー .....	7
図 3-5 選択された Core の構成ウィンドウ .....	8
図 3-6 唯一の Core を削除する場合のプロンプトボックス .....	8
図 3-7 GVIO コア数の上限を超えた場合のプロンプトボックス .....	8
図 3-8 「Probe In Port」 ダイアログボックス .....	9
図 3-9 Probe In Port の「Search Nets」 ダイアログボックス .....	10
図 3-10 Normal モード .....	11
図 3-11 ワイルドカードモード .....	11
図 3-12 正規表現モード .....	11
図 3-13 高度なフィルタリング方法 .....	12
図 3-14 「Probe Out Port」 ダイアログボックス .....	13
図 3-15 Probe Out Port の「Search Nets」 ダイアログボックス .....	14
図 3-16 同期クロック信号が存在しないことを示すプロンプトボックス .....	14
図 3-17 同期クロックが選択されていないことを示すプロンプトボックス .....	15
図 3-18 スティミュラス・ポートとサンプリング・ポートが両方とも有効になっていないことを示すプロンプトボックス .....	15
図 4-1 GVIO ツールの構成ウィンドウ .....	17
図 4-2 Reload プロンプトボックス .....	17
図 4-3 ツールバー .....	18
図 4-4 Cable 選択エラー .....	18
図 4-5 GVIO Programmer .....	19
図 4-6 GVIO 機能コアの構成 .....	19
図 4-7 スティミュラス信号とサンプリング信号の構成ビュー .....	19
図 4-8 Edge 列信号の変化方向 .....	21
図 4-9 サンプリング信号の右クリックメニュー .....	21
図 4-10 スティミュラス信号の右クリックメニュー .....	22

図 4-11 GAO と GVIO による共同デバッグ画面 .....	23
図 4-12 GVIO と GAO による共同デバッグの場合のサンプリング結果.....	24

# 表一覧

表 1-1 用語、略語 .....	1
-------------------	---

# 1 本マニュアルについて

## 1.1 マニュアルの内容

本マニュアルは、主に Gowin 仮想入出力デバッグ・ツール(Gowin Virtual Input Output、GVIO)の構成と使用方法について説明します。本マニュアルに記載のソフトウェア GUI のスクリーンショットは、Gowin ソフトウェア 1.9.10.01 バージョンの場合のものです。ソフトウェアのアップデートにより、一部の内容が変更される場合があります。

## 1.2 関連ドキュメント

GOWIN セミコンダクターのホームページ [www.gowinsemi.com/ja](http://www.gowinsemi.com/ja) から、以下の関連ドキュメントをダウンロード及び閲覧できます。

- Gowin ソフトウェア ユーザーガイド([SUG100](#))
- Gowin ソフトウェア クイックスタートガイド([SUG918](#))
- Gowin アナライザオシロスコープ ユーザーガイド([SUG114](#))

## 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
AO Core	Analysis Oscilloscope Core	機能コア
FPGA	Field Programmable Gate Array	フィールド・プログラマブル・ゲート・アレイ
GAO	Gowin Analyzer Oscilloscope	Gowin アナライザオシロスコープ
GVIO	Gowin Virtual Input Output	仮想入力出力
JTAG	Joint Test Action Group	ジョイント・テスト・アクション・グループ



## 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](http://www.gowinsemi.com/ja)

E-mail : [support@gowinsemi.com](mailto:support@gowinsemi.com)

# 2 概要

**GVIO** は、Gowin が自社で研究開発した、FPGA の内部信号をリアルタイムで監視および駆動できるデジタル信号動的デバッグ・ツールです。オンライン・ロジック・アナライザの **GAO(Gowin Analyzer Oscilloscope)** と併用してデバッグすると、より強力なデバッグが可能になります。このデバッグ環境では、内部信号ステイミュラスを生成し、**GAO** ツールを通じてロジック応答を取得することができます。これにより、ユーザーはシステム分析と障害特定を迅速に実行できるようになり、設計効率が向上します。

**GVIO** の動作原理:**GVIO** のコアは主にコントロールコアと機能コアの2つからなります。コントロールコアはすべての機能コアと **JTAG** スキャニング回路の通信コントローラです。機能コアは主にステイミュラス信号の送信と信号のサンプリング(キャプチャ)に使用されます。コントロールコアは、ホストコンピュータと機能コアを接続し、構成プロセスでホストコンピュータの命令を受信し、機能コアに送信します。データ読み出し手順で機能コアがキャプチャしたデータをホストコンピュータに送信し、Gowin ソフトウェアの **GUI** に表示させます。機能コアはコントロールコアと直接通信し、コントロールコアが転送した命令を受け取り、その命令に応じてデータのキャプチャとステイミュラス信号の送信を行います。

**GVIO** にはステイミュラス・ポートとサンプリング・ポートが含まれています。ステイミュラス・ポートはデザイン内の信号の駆動に使用され、サンプリング・ポートはデザイン内の信号のサンプリングに使用されます。**GVIO** は、RTL 信号に対するステイミュラス提供およびデータ・サンプリングをサポートしています。また、**GVIO** は、「For RTL Design」タイプの **GAO** と組み合わせてデバッグできますが、「For Post-Synthesis Netlist」タイプの **GAO** と組み合わせてデバッグすることはできません。また、**GVIO** ツールでステイミュラス信号の初期値を設定して、電源投入時の動作状態の分析を容易にすることもできます。

**GVIO** には以下の特徴があります。

- デザイン内信号駆動用のステイミュラスを提供可能
- デザイン内の信号をサンプリング可能

- 最大 16 個の機能コアをサポート
- 各機能コアは最大 64 個のステイミュラス・ポートと最大 64 個のサンプリング・ポートをサポート
- 各ステイミュラス・ポートとサンプリング・ポートは最大 256 ビット幅をサポート
- ステイミュラス・ポートの初期化データのカスタマイズをサポート
- サンプリング信号のレベルの変化を検出するアクティビティ検出器を提供

# 3 GVIO 構成ファイル

GVIO 構成ウィンドウは主にコントロールコアと機能コアのパラメータの構成と変更に使われます。

## 3.1 GVIO 構成ファイル

### 3.1.1 GVIO 構成ウィンドウの起動

GVIO 構成ウィンドウは、構成ファイル(.gvio)を作成またはロードすることで起動できます。以下にこの 2 つの起動方法を紹介します。

#### GVIO 構成ファイルの作成

その操作手順は以下のとおりです。

1. Gowin ソフトウェアの **Design** ウィンドウで右クリックし、「**New File...**」を選択すると「**New**」ダイアログボックスがポップアップします(図 3-1)。
2. 「**GVIO Config File**」を選択して「**OK**」をクリックすると、「**New GVIO Config File**」ダイアログボックスがポップアップします(図 3-2)。
3. 「**Name**」編集ボックスで構成ファイル名を入力し、「**OK**」をクリックします。

図 3-1 GVIO 構成ファイルの作成

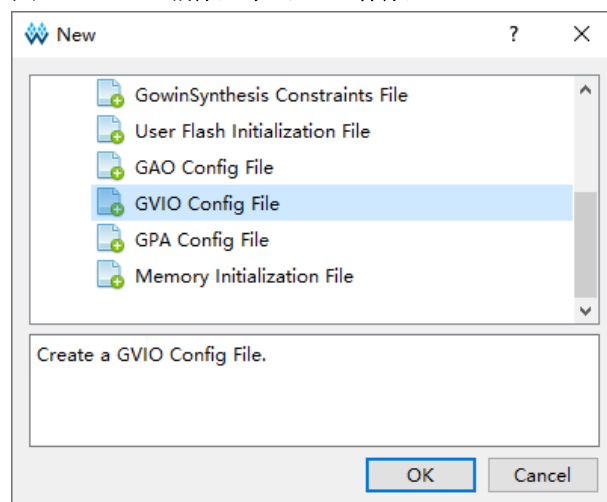
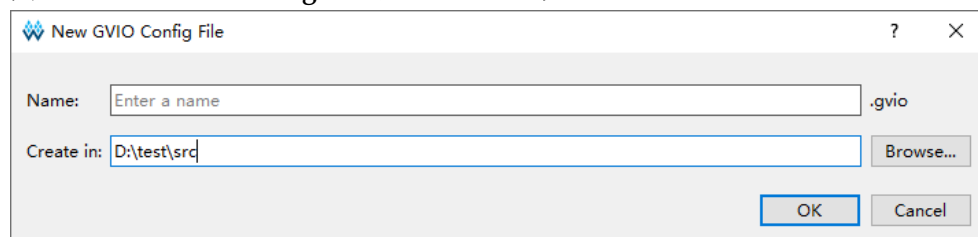


図 3-2 New GVIO Config File ダイアログボックス



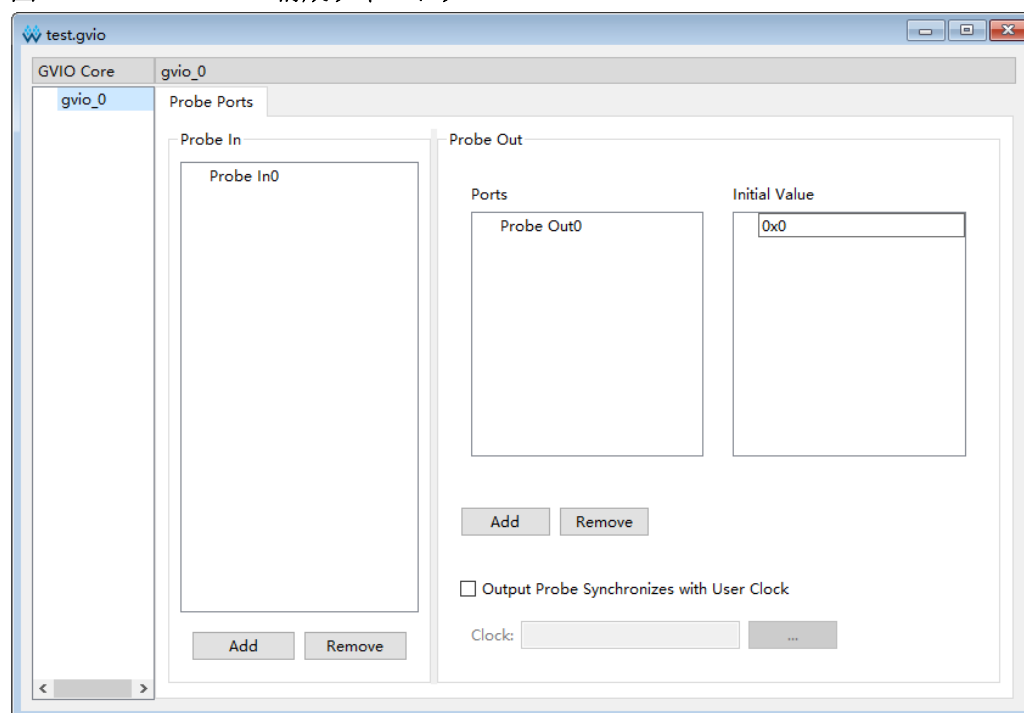
## GVIO 構成ファイルの追加

その操作手順は以下のとおりです。

1. Design ウィンドウで右クリックし、「Add File…」を選択すると「Select Files」ダイアログボックスがポップアップします。
2. 既存の GVIO 構成ファイル(.gvio)を選択して、「Design」ウィンドウに追加します。

Design ウィンドウで構成ファイル(.gvio)をダブルクリックすると、Gowin ソフトウェアのメインウィンドウで GVIO 構成ウィンドウがポップアップします(図 3-3)。GVIO 構成ウィンドウには、機能コアの数を構成するための「GVIO Core」ビューと、対応する Core の信号構成ビューが含まれます。そのうち Core の信号構成ビューには、サンプリング信号を構成するための「Probe In」ビューとステイミュラス信号を構成するための「Probe Out」ビューが含まれます。

図 3-3 GVIO ツールの構成ウィンドウ



### 3.1.2 GVIO ファイルの構成

GVIO ツールの構成ウィンドウは、機能コアの数、サンプリング信号、およびステミュラス信号を構成するために使用されます。

#### 機能コアの数の構成

図 3-4 に示すように、「GVIO Core」ビューは、現在のプロジェクトで使用されている機能コアの数を表示および管理するために使用されます。

「GVIO Core」ビューにはデフォルトで `gvio_0` のみが含まれ、最大 16 個の Core(`gvio_0`~`gvio_15`)をサポートします。その関連操作は次のとおりです。

1. 「GVIO Core」ビューの任意の場所を右クリックし、「Add」をクリックして新しい GVIO Core を追加します。
2. 「GVIO Core」ビューで Core を右クリックして「Remove」をクリックすると、対応する Core が削除されます。
3. 中間番号の Core が削除されると、それに応じて後続の Core 番号も減少するため、Core 番号は常に連続的にインクリメントされます。
4. Core を左クリックすると、右側に対応する Core の構成ビューが表示されます。たとえば、「GVIO Core」ビューで「`gvio_3`」を選択すると、「`gvio_3`」構成ビューが右側に表示されます(図 3-5)。

注記：

- 「GVIO Core」ビューに Core が 1 つしかない場合、削除は禁止されています。Core を右クリックして Remove を選択すると、削除禁止プロンプトボックスがポップアップします(図 3-6)。
- 最大 16 個の Core がサポートされます。16 個の Core の上にさらに Core を追加すると、プロンプトボックスがポップアップします(図 3-7)。

図 3-4 「GVIO Core」ビュー

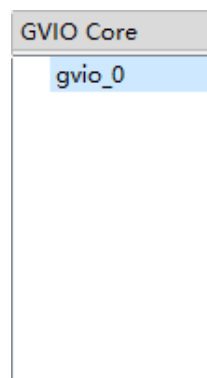


図 3-5 選択された Core の構成ウィンドウ

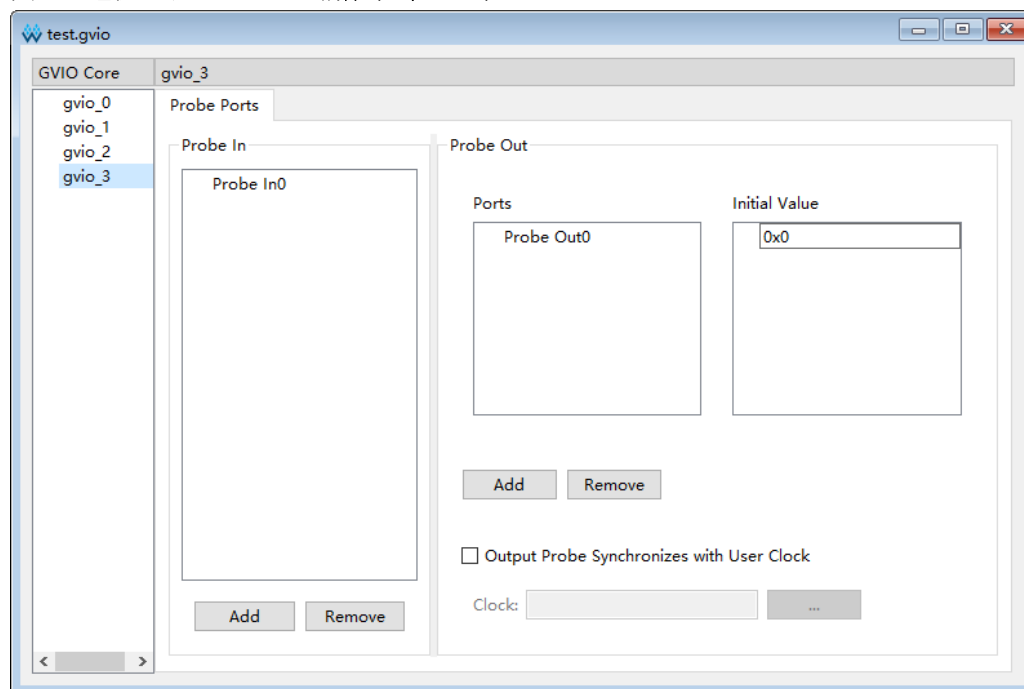


図 3-6 唯一の Core を削除する場合のプロンプトボックス

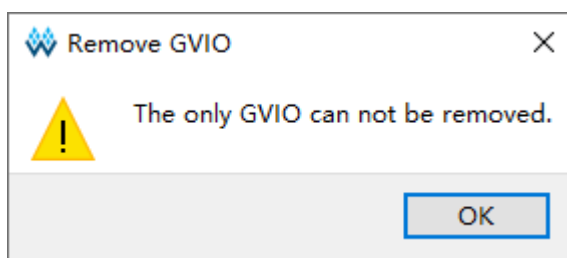
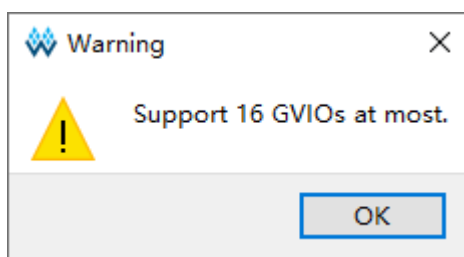


図 3-7 GVIO コア数の上限を超えた場合のプロンプトボックス



### サンプリング信号の構成

「Probe In」ビューは、機能コアのサンプリング・ポートを構成するために使用されます(図 3-3)。「Probe In」ビューは、Probe In0 ～ Probe In63 を含む最大 64 個のサンプリング・ポートをサポートし、各ポートの幅の範囲は 1 ～ 256 ビットです。「Probe In」ビューでサポートされる操作は次のとおりです。

1. 「Add」をクリックすると、「Probe In」ポートが追加されます。


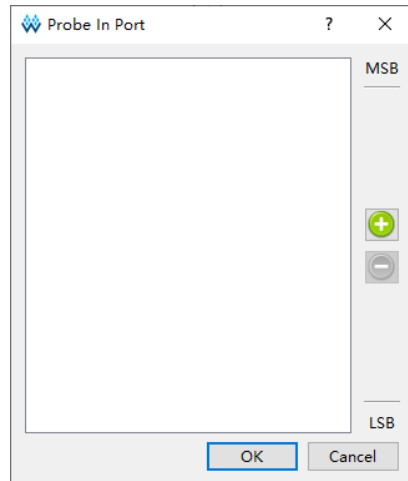
2. 「Remove」をクリックすると、「Probe In」ポートが削除されます。
3. いずれかの Probe In ポートをダブルクリックすると、「Probe In Port」ダイアログボックスがポップアップします(図 3-8)。
4. 「」をクリックすると「Search Nets」ダイアログボックスがポップアップします。「Search」ボタンをクリックして検索します(図 3-9)。サンプリングできない信号はグレー表示され、選択できません。
5. サンプリング信号を選択して「OK」をクリックします。

図 3-8 「Probe In Port」ダイアログボックス



注記：

上図の MSB と LSB は、それぞれサンプリング・ポートの上位と下位を表します。

「Probe In Port」ダイアログボックスの信号は、次の操作をサポートします。


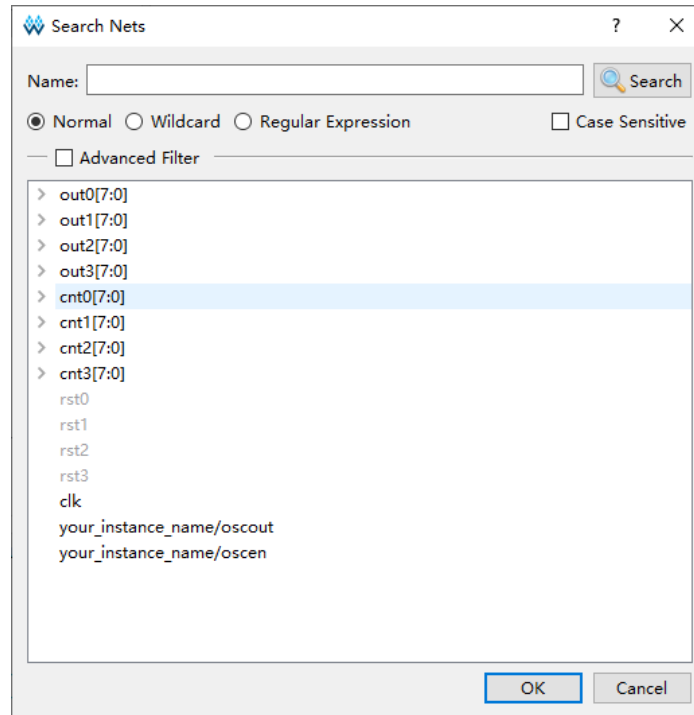
- サンプリング信号の削除をサポート：サンプリング信号をクリックして選択するか、**Shift+左キー**または**Ctrl+左キー**で複数のトリガ信号を選択し、「」をクリックして削除します。
- ドラッグ&ドロップによる信号の並べ替えをサポート：トリガ信号をクリックするか、**Shift+左キー**と**Ctrl+左キー**で複数のトリガ信号を選択し、そして左クリックしてドラッグし、信号の並べ替えを完了します。
- 同じ Probe In Port に同じ信号の繰り返し追加はできません。そのルールは以下の通り：
  - 個々の信号を繰り返し追加すると、追加失敗になります。
  - バス信号のサブ信号が既に追加されている場合、このバス信号を再度追加すると、追加されたサブ信号は削除され、バス信号全体は保持されます。
  - バス信号が既に追加されている場合、そのサブ信号を追加すると、追加失敗になります。



図 3-9 Probe In Port の「Search Nets」ダイアログボックス



「Search Nets」ダイアログボックスでは、Normal、Wildcard、Regular Expression の 3 つのオプションは相互に排他的です。

- **Normal** オプションは、通常の方法による検索です。このオプションを選択した場合、**Search** ボタンをクリックすると「Name」テキストボックスに含まれる文字列が検索されます(図 3-10)。
- **Wildcard** オプションは、ワイルドカードによる検索です。このオプションを選択した場合、**Search** ボタンをクリックすると **Name** テキストボックスに含まれる文字列が検索されます。ワイルドカード(\*、?)が使用できます(図 3-11)。
- **Regular Expression** オプションは正規表現による検索です。このオプションを選択した場合、**Search** ボタンをクリックすると、**Name** テキストボックスに含まれる文字列が検索されます。正規表現を使用できます(図 3-12)。
- **Case Sensitive** にチェックを入れると、信号のマッチングを行う時、大文字と小文字を区別します。**Search Nets** ダイアログの下にある **Signal** エリアはクリックで 1 つ選択、**Shift+左キー** と **Ctrl+左キー** で複数選択などの機能をサポートします。

図 3-10 Normal モード

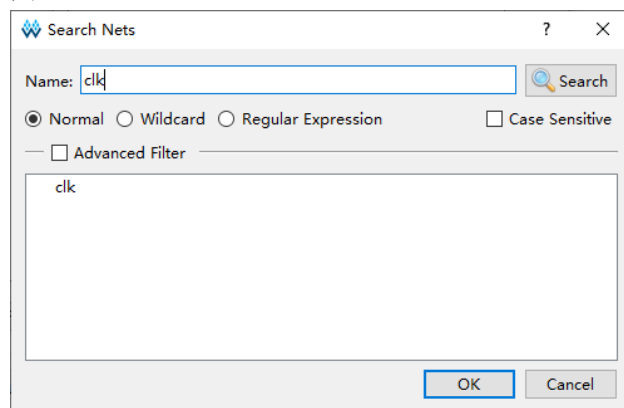


図 3-11 ワイルドカードモード

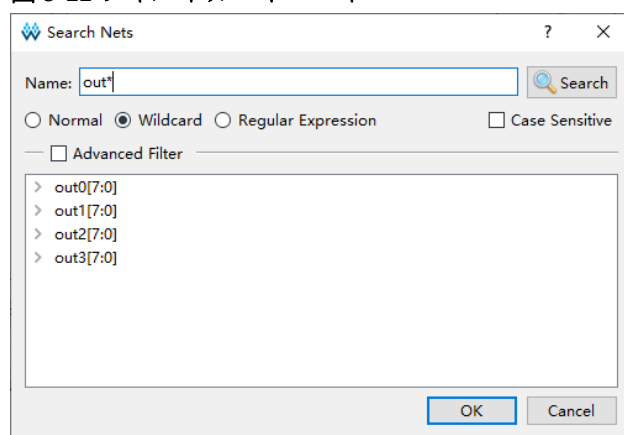
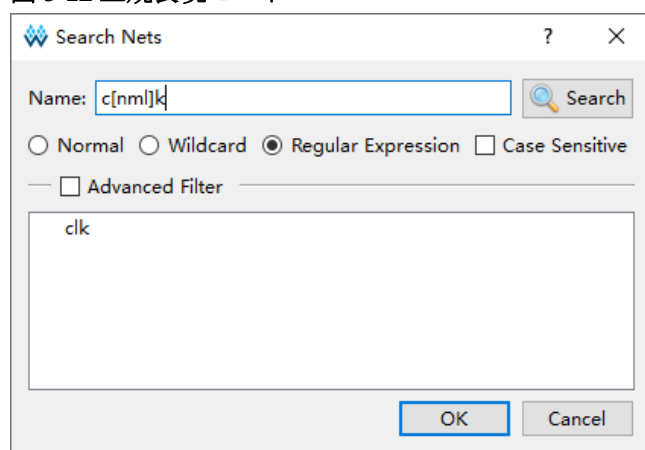


図 3-12 正規表現モード



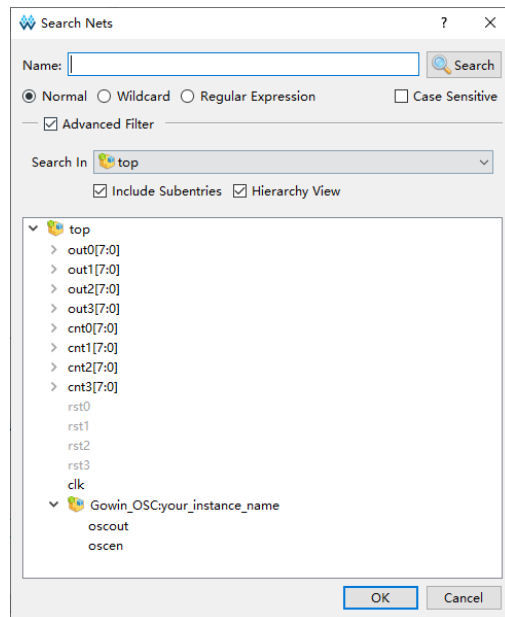
**Advanced Filter** にチェックを入れると、高度なフィルタリング方法が使用されます。これにより、さらにフィルタリング条件を設定して、目的の信号をより正確に見つけることができます。ここで：

- **Search In** オプションでは、どのモジュールから信号をフィルタリングするか設定できます。

- **Include Subentries** オプションでは、サブモジュールから信号をフィルタリングするかどうか設定できます。
- **Hierarchy View** オプションは、階層表示のために使用されます。


図 3-13 に示すように、「Search In」で「top」を選択し、かつ「Include Subentries」および「Hierarchy View」をチェックした後、「Search」ボタンをクリックすると、**top** モジュールとそのサブモジュールに関連するすべての出力信号が階層形式で表示されます。

図 3-13 高度なフィルタリング方法



## ステイミュラス信号の構成

「Probe Out」ビューは、機能コアのステイミュラス・ポートを構成するために使用され、「Ports」、「Initial Value」、および「Output Probe Synchronizes with User Clock」という 3 つの部分で構成されています。そのうち「Ports」は、Probe Out0 ~ Probe Out63 を含む最大 64 個のステイミュラス・ポートをサポートし、各ポートの幅の範囲は 1 ~ 256 ビットです。「Initial Value」は、対応するステイミュラス信号の初期値を設定するために使用されます。「Output Probe Synchronizes with User Clock」は、ステイミュラス信号をユーザーが設定したクロック信号に同期させるかどうかを設定するために使用されます。「Probe Out」ビューでサポートされる操作は次のとおりです。

- 「Add」をクリックすると、「Probe Out」ポートが追加されます。
- 「Remove」をクリックすると、「Probe Out」ポートが削除されます。
- いずれかの Probe Out ポートをダブルクリックすると、「Probe Out Port」ダイアログボックスがポップアップします(図 3-14)。
- 「」をクリックすると「Search Nets」ダイアログボックスがポッ

ブアップします。「Search」ボタンをクリックして検索します(図 3-15)。ステイミュラス信号として使用できない信号はグレー表示され、選択できません。このうち、ソースのない信号はステイミュラス信号として使用できます。

- ステイミュラス信号を選択して「OK」をクリックします。
- ステイミュラス信号の右側で対応する **Initial Value** を設定します。そのデフォルト値は 0 で、16 進数で表示されます。
- 「Output Probe Synchronizes with User Clock」をチェックすると、ステイミュラス信号はユーザーが設定したクロック信号に同期するようになります。[Clock]の右側にある「...」ボタンをクリックすると、クロック信号を追加するための「Search Nets」ダイアログボックスが開きます。

図 3-14 「Probe Out Port」ダイアログボックス

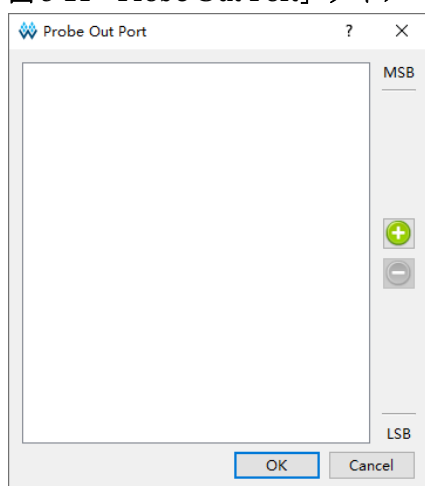
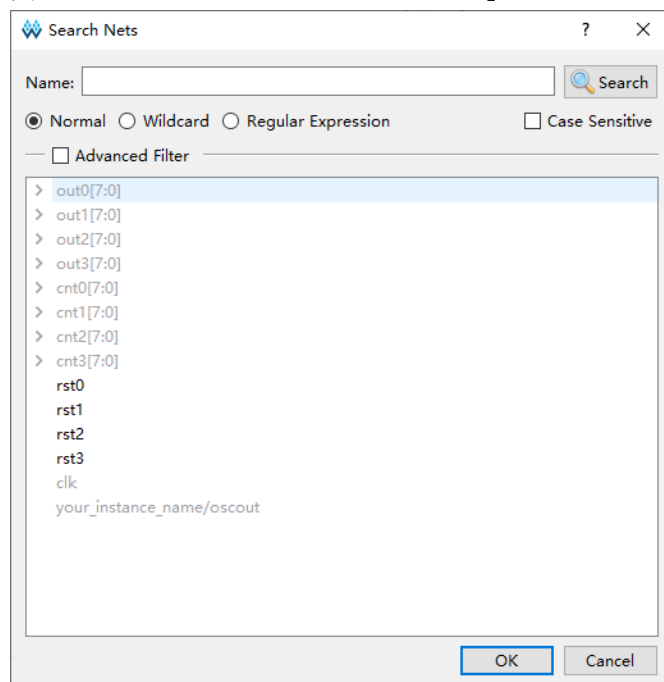


図 3-15 Probe Out Port の「Search Nets」ダイアログボックス



注記：

- 構成ファイル(.gvio)を保存する時、構成した同期クロック信号が存在しない場合、この同期クロック信号が存在しないことを示すメッセージがポップアップします(図 3-16)。
- 「Output Probe Synchronizes with User Clock」がチェックされているが、同期クロックが構成されていない場合は、同期クロックが選択されていないことを示すメッセージがポップアップします。
- Probe In ポートと Probe Out ポートの少なくとも 1 つのために信号を構成する必要があります。そうしないと、エラーが報告されます(図 3-18)。

図 3-16 同期クロック信号が存在しないことを示すプロンプトボックス

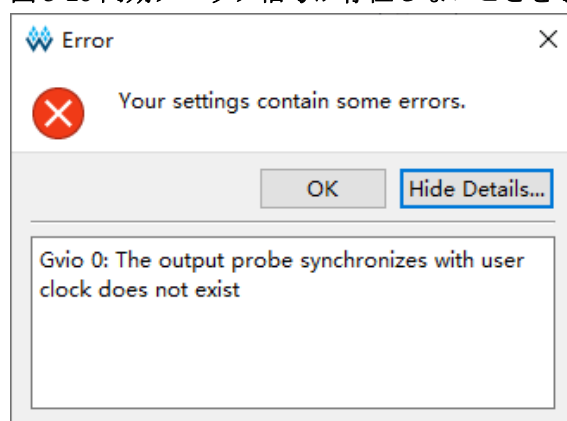


図 3-17 同期クロックが選択されていないことを示すプロンプトボックス

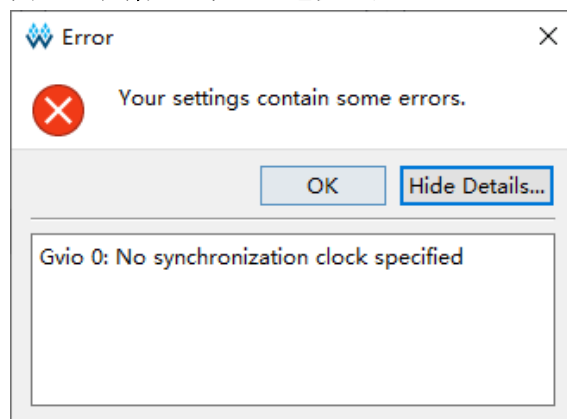
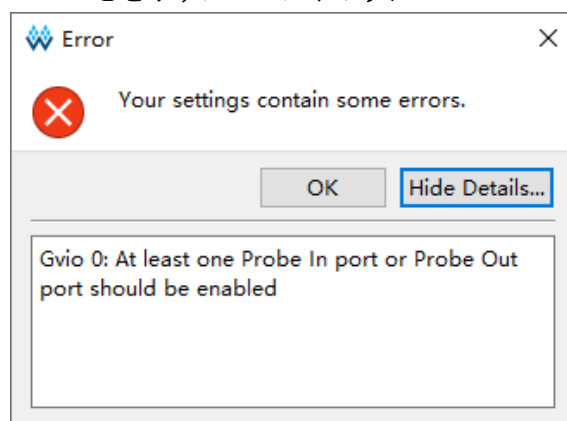


図 3-18 スティミュラス・ポートとサンプリング・ポートが両方とも有効になっていないことを示すプロンプトボックス



### 3.1.3 ビットストリームファイルの生成

GVIO ファイルの構成が完了したら、「**Process**」ウィンドウで「**Place&Route**」をダブルクリックしてユーザーデザイン全体の配置配線を実行すると、ユーザーデザインと **GVIO** 構成情報を含む、デフォルトのファイル名がプロジェクト名のビットストリームファイルが生成されます。このファイルはデフォルトでプロジェクトパスの下に「**/impl/pnr/**」に配置されます。

# 4 GVIO ツールの使用

GVIO ツールは主に、JTAG インターフェースを介してステイミュラス信号の値を再構成し、信号をサンプリングして表示するために使用されます。これにより、ユーザーは信号をより簡単にデバッグおよび観察できるようになります。

「For RTL Design」タイプの GAO(.rao)の場合、GVIO ツールと共同でデバッグできますが、「For Post-Synthesis Netlist」タイプ GAO(.gao)の場合、共同デバッグがサポートされません。以下では、「GVIO による単独デバッグ」と「GVIO と GAO による共同デバッグ」について、それぞれ説明します。

## 4.1 GVIO ツールによる単独デバッグ

### 4.1.1 GVIO の起動

ロジック・アナライザとして、GVIO は GAO と同様に、Gowin Analyzer Oscilloscope ツール上で動作します。.gvio ファイルをロードする手順は以下のとおりです。

1. メニューバーで「Tools」を選択します。
2. 表示されるドロップダウン・リストから「Gowin Analyzer Oscilloscope」を選択し、GVIO ツールを起動します。デフォルトでは、プロジェクト内の有効な.gvio ファイルがロードされます。または、「Open」ボタンをクリックして、ロードする.gvio ファイルを選択します。


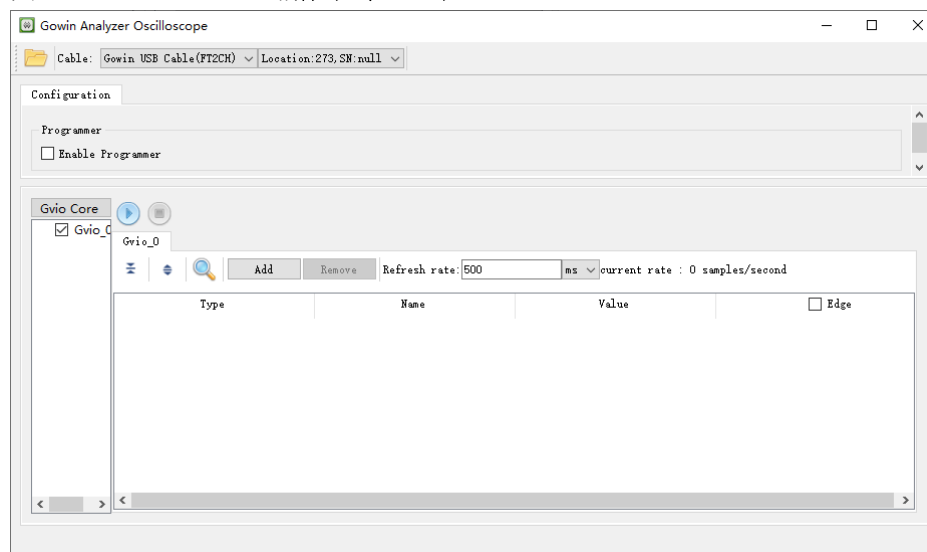
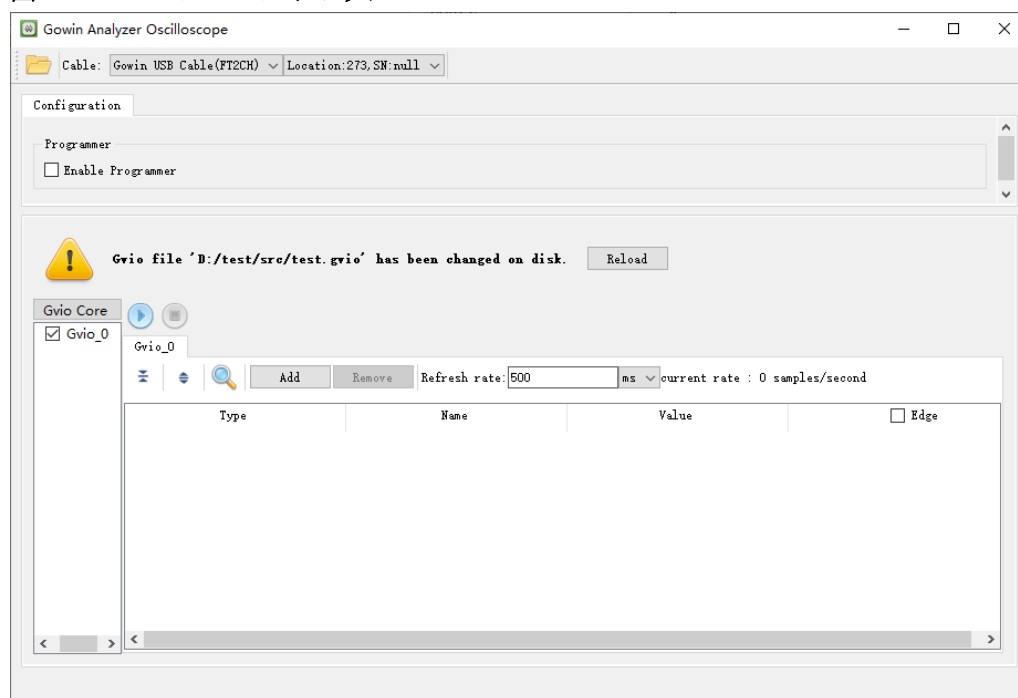
また、IDE ツールバー上のアイコン「」をクリックして GVIO ツールを起動することもできます。.gvio ファイルの構成については、[3.1.2GVIO ファイルの構成](#)を参照してください。

図 4-1 GVIO ツールの構成ウィンドウ



.gvio ファイルが更新されると、Reload プロンプトボックスが表示されます(図 4-2)。「Reload」ボタンをクリックすると、更新された.gvio ファイルが再ロードされます。

図 4-2 Reload プロンプトボックス



## 4.1.2 GVIO の実行

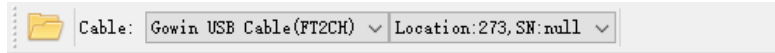
図 4-1 に示すように、GVIO ウィンドウにはツールバーと Configuration ビューがあります。ツールバーでは構成ファイル(.gvio) をロードし、Cable タイプを構成することができます。「Configuration」ビューには Programmer と機能コアの構成が含まれており、そのうち機能コアの構成には GVIO コアの有効化、ステイミュラス信号の構成、およびサンプリング信号の表示が含まれます。



## ツールバーの操作

ツールバーには、「」と「**Cable**」オプションがあります(図 4-3)。

図 4-3 ツールバー



各ツールの機能の詳細は以下のとおりです。



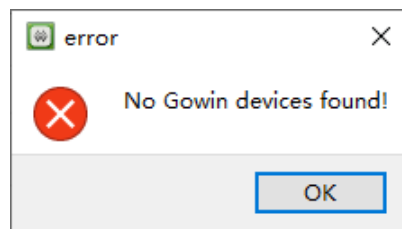
- 「」: 構成ファイル (.gvio) をロードするために使用されます。
- 「」: キャプチャウィンドウは、シリアルポート(Gowin USB Cable (GWU2X) )、Gowin USB Cable(FT2CH)、およびパラレルポート(Parallel Port (LPT) )をサポートします。GVIO 実行画面を開くと、Cable タイプが自動的にスキャンされます。Programmer を使用してビットストリームをダウンロードする場合、または GVIO を使用してステミュラス信号の送信とデータのサンプリングを実行する場合は、正しい Cable タイプを選択する必要があります。そうしないと、エラーが報告されます(図 4-4)。
- 「Location: 17, SN: U7DJFGEN」: 接続されたケーブルが自動的にスキャンされ、対応する Location パラメータと SN コードパラメータが表示されます。デュアルチャネルのケーブルまたは複数のケーブルが接続されている場合、パラメータに基づいて対応するケーブルを手動で選択することができます。

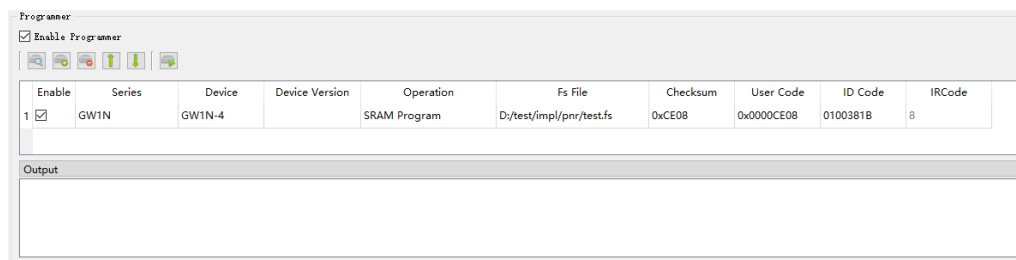
図 4-4 Cable 選択エラー



## Programmer

GVIO の実行画面には Programmer 機能が統合されています(図 4-5)。GVIO Programmer の使用法は、GAO Programmer の使用方法と同じです。詳細については、『Gowin アナライザオシロスコープ ユーザーガイド ([SUG114](#))』のセクション 4.1.2 を参照してください。

図 4-5 GVIO Programmer



## GVIO Core のイネーブル

GVIO は最大 16 個のコアをサポートします。GVIO の実行画面で、Gvio\_0 ~ Gvio\_15 をチェックすることで、対応するコアを有効にすることができます。有効になっている GVIO コアのみがステイミュラス信号の送信とデータのサンプリングを実現できます。

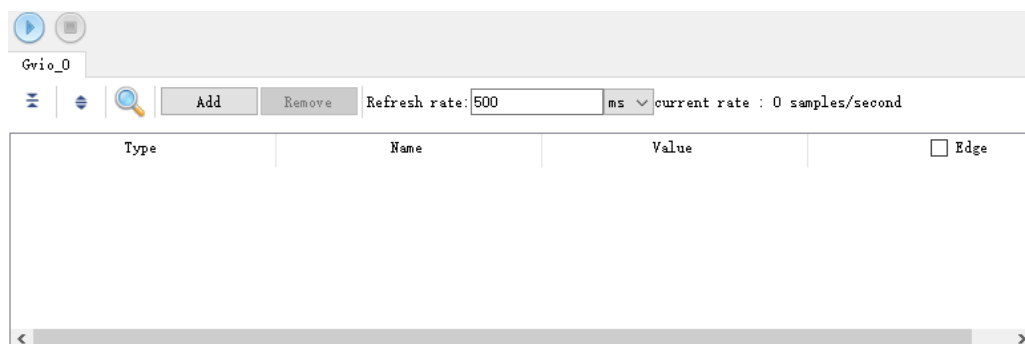
図 4-6 GVIO 機能コアの構成








## ステイミュラス信号とサンプリング信号の構成

ステイミュラス信号とサンプリング信号の構成ビューを図 4-7 に示します。GVIO の動作中に、ステイミュラス信号の値を動的に構成し、サンプリング信号を定期的にリードバックすることができます。このビューにはツールバーと信号リスト領域が含まれています。

図 4-7 ステイミュラス信号とサンプリング信号の構成ビュー



ツールバー：

- 「」: **Start**。GVIO を起動し、ステイミュラス信号の送信とサンプリング信号のリードバックを完了します。
- 「」: **Stop**。ステイミュラス信号の送信とサンプリング信号のリードバックを停止します。
- 「」: **Collapse**。信号リスト領域内のすべてのバス信号を折りたたむために使用されます。
- 「」: **Expand**。信号リスト領域内のすべてのバス信号を展開するために使用されます。
- 「」: **Search**。信号リスト領域内の信号を検索するために使用されます。
- 「**Add**」: このボタンをクリックすると、「**Search Nets**」ダイアログボックスがポップアップします。このダイアログボックスは、ステイミュラス信号とサンプリング信号を追加するために使用されます。リストに追加されている信号は、「**Search Nets**」ダイアログボックスには表示されません。
- 「**Remove**」: このボタンをクリックすると、選択したリスト内の信号が削除されます。
- 「**Refresh rate**」: サンプリング信号をリードバックする周期を設定するために使用されます。単位には **ms** と **s** があります。デフォルト値は **500ms** で、最小値は **1ms** です。「**Start**」をクリックすると、GVIO ツールは **Refresh rate** でサンプリング信号をリードバックし、ユーザーがステイミュラス信号値を更新した後に **JTAG** ポートを介してステイミュラス信号をデザインに駆動します。
- 「**current rate**」: サンプリング信号をリードバックする実際の周波数を表示するために使用されます。デフォルトの表示は **0 Samples/second** です。

信号リスト:

- **Type**: タイプ: 信号の方向を表示するために使用されます。**Input** タイプ信号はサンプリング信号、**Output** タイプ信号はステイミュラス信号です。
- **Name**: 信号の名称を表示するために使用されます。
- **Value**: サンプリング信号の値とステイミュラス信号の値を表示するために使用されます。
- **Edge**: この列を有効にすると、立ち上がりエッジの変化、立ち下がりエッジの変化、立ち上がりエッジと立ち下がりエッジの両方の変化など、特定の周期内のサンプリング信号値の変化方向が表示されます。

注記:

信号リスト内の信号は、上下にドラッグすることで並べ替えることができます。

図 4-8 Edge 列信号の変化方向

Type	Name	Value	<input checked="" type="checkbox"/> Edge
Input	out0[7:0]	[H]62	
Input	out0[7]	0	↕
Input	out0[6]	1	↑
Input	out0[5]	1	↑
Input	out0[4]	0	↕
Input	out0[3]	0	↕
Input	out0[2]	0	↕
Input	out0[1]	1	↑
Input	out0[0]	0	↓
Output	rst0	[B]1	

信号リストでサンプリング信号を選択して右クリックすると、右クリックメニューが表示されます(図 4-9)。右クリックメニューには次のものが含まれます。

- **Digit** : サンプリング信号の値をデジタル形式で表示します。
- **LED** : サンプリング信号の値を **LED** 形式で表示します。緑色は **High** レベル、灰色は **Low** レベルを表します。たとえば、図 4-9 に示すように、out0[0]は灰色の LED(Low レベル)、out0[2]は緑色の LED(High レベル)です。
- **Radix** : バス信号の進法を変更します。**Binary**(2 進数)、**Octal**(8 進数)、**Signed Decimal**(符号付き 10 進数)、**Unsigned Decimal**(符号なし 10 進数)、**Hexadecimal**(16 進数)などのオプションが含まれます。デフォルトでは、16 進数です。
- **Edge Persistence** : Edge 列のレベル変化を示す矢印の表示時間。8 Samples、40 Samples、80 Samples、および Infinite は、それぞれ Refresh Rate \*8、Refresh Rate \*40、Refresh Rate \*80、および無限の 4 つの時間範囲を示します。
- **Remove** : 選択したサンプリング信号を削除するために使用されます。

注記 :

- バス信号の表示形式が **Digit** または **LED** に設定されている場合、そのサブ信号は設定された表示形式で表示されますが、バス信号は常に **Digit** 形式で表示されます。
- サブ信号の表示形式が **Digit** または **LED** に設定されている場合、選択されたサブ信号のみが設定された表示形式で表示され、バス信号は常に **Digit** 形式で表示されます。

図 4-9 サンプリング信号の右クリックメニュー

Type	Name	Value	<input type="checkbox"/> Edge
Input	out0[7:0]	[H]14	
Input	out0[7]		
Input	out0[6]		
Input	out0[5]		
Input	out0[4]		
Input	out0[3]		
Input	out0[2]		
Input	out0[1]		
Input	out0[0]		
Input	out1[7:0]	[H]80	
Output	rst0	[B]1	
Output	rst1	[B]1	

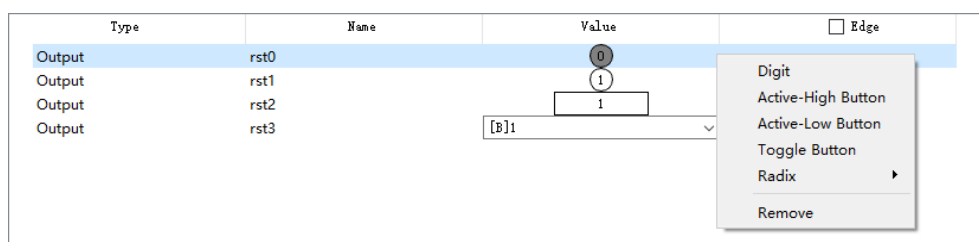
信号リストでステイミュラス信号を選択して右クリックすると、右クリックメニューが表示されます(図 4-10)。右クリックメニューには次のものが含まれます。

- **Digit** : ステイミュラス信号の値をデジタル形式で表示します。
- **Active-High Button** : ステイミュラス信号の値をボタンの形式で表示します。初期値は 0 です。ボタンをクリックすると、数値が 1 になり、ボタンの色が白になります。ボタンを放すと、数値が 0 になり、ボタンの色が灰色になります。図 4-10 の rst0 の Value 列に示すとおりです。
- **Active-Low Button** : ステイミュラス信号の値をボタンの形式で表示します。初期値は 1 です。ボタンをクリックすると、数値が 0 になり、ボタンの色が灰色になります。ボタンを放すと、数値が 1 になり、ボタンの色が白になります。図 4-10 の rst1 の Value 列に示すとおりです。
- **Toggle Button** : ステイミュラス信号の値をスイッチの形式で表示します。スイッチをクリックするたびに、スイッチの値がトグルします。図 4-10 の rst2 の Value 列に示すとおりです。
- **Radix** : バス信号の進法を変更します。Binary(2 進数)、Octal(8 進数)、Signed Decimal(符号付き 10 進数)、Unsigned Decimal(符号なし 10 進数)、Hexadecimal(16 進数)などのオプションが含まれます。デフォルトでは、16 進数です。
- **Remove** : 選択したステイミュラス信号を削除するために使用されます。

注記 :

- Active-High Button、Active-Low Button、Toggle Button はバス信号に対して無効であり、バス信号は常に Digit 形式で表示されます。
- バス信号に対する Active-High Button、Active-Low Button、または Toggle Button の選択は、すべてのサブ信号に対して有効です。

図 4-10 ステイミュラス信号の右クリックメニュー



## 4.2 GVIO と GAO による共同デバッグ

GVIO と GAO の組み合わせは、より強力なデバッグ環境を提供します。GVIO は、制御信号をリアルタイムで更新して GAO のトリガー条件を強制的に発生させることができ、それによって外部センサー・データなどのシミュレーションが実現されます。また、GAO のサンプリングを通じて対応するロジック応答を取得できるため、デバッグが容易になります。

### 4.2.1 GAO 構成ファイル

GVIO は、For RTL Design タイプの GAO(.rao)と共同でデバッグでき

ます。 .rao 構成ファイルの作成については、『Gowin アナライザオシロスコープ ユーザーガイド([SUG114](#))』の第3章を参照してください。

## 4.2.2 GVIO 構成ファイル

GVIO 構成ファイルの作成については、[3.1.2GVIO ファイルの構成](#)を参照してください。

## 4.2.3 GAO と GVIO の実行

共同デバッグの場合、GVIO と GAO は両方とも Gowin Analyzer Oscilloscope ツール上で動作します。gvio ファイルと .rao ファイルをロードする手順は以下のとおりです。

1. メニューバーで「Tools」を選択します。
2. 表示されるドロップダウン・リストから「Gowin Analyzer Oscilloscope」を選択します。デフォルトでは、プロジェクト内の有効な .gvio と .rao ファイルがロードされます(図 4-11)。


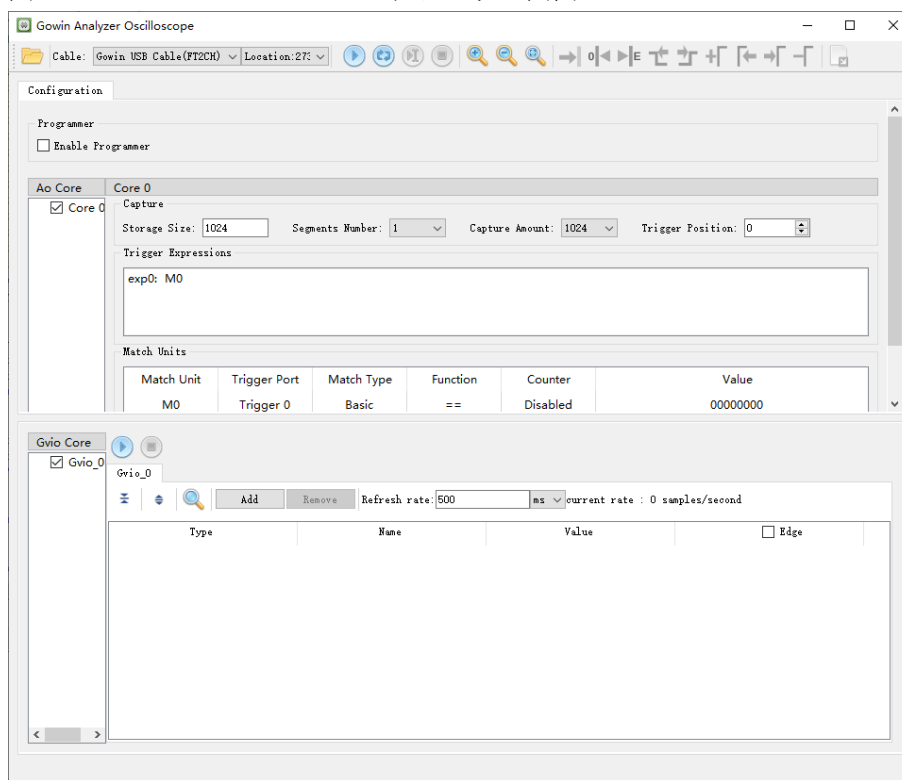
または、ツールバー上のアイコン「」をクリックして Gowin Analyzer Oscilloscope を起動します。

図 4-11 GAO と GVIO による共同デバッグ画面



GAO の実行などについては、『Gowin アナライザオシロスコープ ユーザーガイド([SUG114](#))』のセクション 4.1.2 を参照してください。

GVIO の実行などについては、[4.1.2GVIO の実行](#)を参照してください。

共同デバッグ画面には 2 つの Start ボタンがあり、上の Start ボタンは GAO の動作を制御し、下の Start ボタンは GVIO の動作を制御します。GAO

と GVIO は同時にまたは個別に実行できます。以下は、GAO と GVIO の同時実行を例に説明します(図 4-12)。

図 4-12 では、rst0、rst1、rst2、および rst3 はそれぞれ out0、out1、out2、および out3 のアクティブ Low のリセット信号です。GVIO は次のステイミュラスを提供します : rst0=0、rst1=1、rst2=1、rst3=1。そして、GAO のサンプリング結果から、out0 信号がリセットされていることがわかります。

図 4-12 GVIO と GAO による共同デバッグの場合のサンプリング結果

