



# Gowin Clock ユーザーガイド

UG286-1.9.6J, 2022-11-11

## 著作権について（2022）

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**、Gowin、及びGOWINSEMI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。本文書における全ての情報は、予備的情報として取り扱われなければなりません。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
2016/05/18	1.05J	初版。
2016/07/15	1.06J	図面を更新。
2016/08/31	1.07J	GW2A シリーズ FPGA 製品のサポートを追加。
2016/10/27	1.08J	GW2AR シリーズ FPGA 製品のサポートを追加。
2017/09/22	1.09J	最新のプリミティブライブラリに基づいて変更。
2017/10/16	1.10J	GW1N の説明及び説明図を追加。
2018/01/05	1.2J	高速クロックに関する内容を更新。
2018/04/20	1.3J	PLL ポート及びパラメータの情報を更新。
2019/12/04	1.4J	プリミティブの説明を更新。
2020/08/18	1.5J	マニュアルの構造を最適化。
2021/01/14	1.6J	PLLO、OSCO、CLKDIVG、および DCCG モジュールを追加。
2021/02/01	1.7J	<ul style="list-style-type: none"> <li>● rPLL および PLLO を更新。</li> <li>● GW1NR-2、GW2AN-55C デバイスのサポートを追加。</li> </ul>
2021/04/13	1.7.1J	GW1NZ-2 デバイスを削除。
2021/06/18	1.8J	<ul style="list-style-type: none"> <li>● GW1N-2B、GW1N-1P5、GW1N-1P5B、GW1NR-2B、GW2AN-18X、および GW2AN-9X デバイスを追加。</li> <li>● IP 呼び出しの図面を更新し、Help 内容を削除。</li> </ul>
2021/9/10	1.9J	位相同期回路の内容を追加。
2021/10/12	1.9.1J	<ul style="list-style-type: none"> <li>● PLL IP コアを構成するときの各モードの紹介を更新。</li> <li>● CLKDIVG の入力ソースの説明を更新。</li> </ul>
2022/01/24	1.9.2J	コード例のフォーマットを微調整。
2022/05/20	1.9.3J	OSCZ の説明を更新。
2022/07/22	1.9.4J	<ul style="list-style-type: none"> <li>● PLL モジュールの説明を更新。</li> <li>● OSCZ 対応デバイスを更新。</li> </ul>
2022/08/11	1.9.5J	一部のデバイスのバージョン情報を更新。
2022/11/11	1.9.6J	GW1NS-2、GW1NS-2C、GW1NSE-2C、GW1NSR-2、および GW1NSR-2C デバイスを削除。

# 目次

目次.....	i
図一覧.....	iii
表一覧.....	v
<b>1 本マニュアルについて .....</b>	<b>1</b>
1.1 マニュアル内容 .....	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>
2.1 グローバルクロック .....	3
2.2 高速クロック .....	6
2.3 位相同期回路.....	6
2.3.1 GW1N-1P5 / GW1N-2 / GW1NR-2 / GW2AN-18X / GW2AN-9X .....	6
2.3.2 LittleBee®ファミリーと Arora ファミリーのその他のデバイス.....	8
<b>3 グローバルクロック .....</b>	<b>10</b>
3.1 DQCE .....	10
3.1.1 プリミティブの紹介 .....	10
3.1.2 IP の呼び出し .....	11
3.2 DCS.....	13
3.2.1 プリミティブの紹介 .....	13
3.2.2 IP の呼び出し .....	17
<b>4 高速クロック .....</b>	<b>19</b>
4.1 DHCEN.....	19
4.1.1 プリミティブの紹介 .....	19
4.1.2 IP の呼び出し .....	20
4.2 DHCENC .....	22
4.2.1 プリミティブの紹介 .....	22
4.2.2 IP の呼び出し .....	23

4.3 DCC.....	23
4.3.1 プリミティブの紹介 .....	23
4.4 DCCG.....	25
4.4.1 プリミティブの紹介 .....	25
4.5 CLKDIV2 .....	27
4.5.1 プリミティブの紹介 .....	27
4.5.2 IP の呼び出し .....	28
<b>5 システムクロック .....</b>	<b>30</b>
5.1 rPLL .....	30
5.1.1 プリミティブの紹介 .....	30
5.1.2 IP の呼び出し .....	40
5.2 PLLVR .....	44
5.2.1 プリミティブの紹介 .....	44
5.2.2 IP の呼び出し .....	52
5.3 PLLO .....	53
5.3.1 プリミティブの紹介 .....	53
5.3.2 IP の呼び出し .....	71
5.4 DLLDL .....	77
5.4.1 プリミティブの紹介 .....	77
5.4.2 IP の呼び出し .....	80
5.5 CLKDIV .....	82
5.5.1 プリミティブの紹介 .....	82
5.5.2 IP の呼び出し .....	84
5.6 CLKDIVG.....	86
5.6.1 プリミティブの紹介 .....	86
5.6.2 IP の呼び出し .....	88
5.7 DQS.....	89
5.7.1 プリミティブの紹介 .....	89
<b>6 オシレータ .....</b>	<b>95</b>
6.1 プリミティブの紹介 .....	95
6.1.1 OSC.....	95
6.1.2 OSCZ .....	97
6.1.3 OSCH .....	99
6.1.4 OSCO.....	101
6.1.5 OSCW .....	102
6.2 IP の呼び出し .....	104

## 図一覧

図 2-1 LittleBee®ファミリーの 1K、2K、および 4K FPGA 製品での GCLK の配置 .....	4
図 2-2 LittleBee®ファミリーの 9K FPGA 製品および Arora ファミリーの FPGA 製品での GCLK の配置 .....	5
図 2-3 PLL の説明図 .....	7
図 2-4 PLL の説明図 .....	8
図 3-1 DQCE のポート図 .....	10
図 3-2 DQCE IP の構成ウィンドウ .....	12
図 3-3 DCS のポート図 .....	14
図 3-4 Non-Glitchless モードのタイミング図 .....	16
図 3-5 DCS mode が RISING の場合のタイミング .....	16
図 3-6 DCS mode が FALLING の場合のタイミング .....	16
図 3-7 DCS mode が CLK0_GND の場合のタイミング .....	16
図 3-8 DCS mode が CLK0_VCC の場合のタイミング .....	17
図 3-9 DCS IP の構成ウィンドウ .....	17
図 4-1 DHCEN のポート図 .....	19
図 4-2 DHCEN IP の構成ウィンドウ .....	21
図 4-3 DHCENC のポート図 .....	22
図 4-4 DCC のポート図 .....	24
図 4-5 DCCG のポート図 .....	25
図 4-6 CLKDIV2 のポート図 .....	27
図 4-7 CLKDIV2 IP の構成ウィンドウ .....	29
図 5-1 rPLL のポート図 .....	31
図 5-2 入力ソースが CLKOUT の場合の CLKOUTD3 のタイミング .....	32
図 5-3 入力ソースが CLKOUTP の場合の CLKOUTD3 のタイミング .....	32
図 5-4 rPLL IP の構成ウィンドウ .....	41
図 5-5 PLLVR のポート図 .....	45
図 5-6 PLLVR IP の構成ウィンドウ .....	52
図 5-7 PLLO のポート図 .....	54
図 5-8 B チャネルデューティサイクルの微調整タイミング図（微調整方向は 1'b1、ステップサイズは 1） .....	63

図 5-9 B チャンネルデューティサイクルの微調整タイミング図（微調整方向は 1'b0、ステップサイズは 1） .....	63
図 5-10 PLLO IP の構成ウィンドウ .....	72
図 5-11 DLLDLY のポート図 .....	78
図 5-12 DLLDLY IP の構成ウィンドウ .....	81
図 5-13 CLKDIV のポート図 .....	82
図 5-14 CLKDIV IP の構成ウィンドウ .....	85
図 5-15 CLKDIVG のポート図 .....	86
図 5-16 CLKDIVG IP の構成ウィンドウ .....	88
図 5-17 DQS のポート図 .....	90
図 6-1 OSC のポート図 .....	96
図 6-2 OSCZ のポート図 .....	98
図 6-4 OSCH のポート図 .....	100
図 6-5 OSCO のポート図 .....	101
図 6-6 OSCW のポート図 .....	103
図 6-7 OSC IP の構成ウィンドウ .....	105

# 表一覧

表 1-1 用語、略語 .....	1
表 2-1 PLL ポートの定義 .....	7
表 2-2 PLL ポートの定義 .....	8
表 3-1 DQCE のポートの説明 .....	10
表 3-2 DCS のポートの説明 .....	14
表 3-3 DCS のパラメータの説明 .....	14
表 4-1 DHCEN のポートの説明 .....	19
表 4-2 DHCENC 対応デバイス .....	22
表 4-3 DHCENC のポートの説明 .....	22
表 4-4 DCC 対応デバイス .....	23
表 4-5 DCC のポートの説明 .....	24
表 4-6 DCC のパラメータの説明 .....	24
表 4-7 DCCG 対応デバイス .....	25
表 4-8 DCCG のポートの説明 .....	25
表 4-9 DCCG のパラメータの説明 .....	26
表 4-10 CLKDIV2 のポートの説明 .....	27
表 4-11 CLKDIV2 のパラメータの説明 .....	27
表 5-1 rPLL 対応デバイス .....	30
表 5-2 rPLL のポートの説明 .....	31
表 5-3 rPLL のパラメータの説明 .....	32
表 5-4 IDSEL ポートのパラメータ参照テーブル .....	34
表 5-5 FBDSEL ポートのパラメータ参照テーブル .....	34
表 5-6 ODSEL ポートのパラメータ参照テーブル .....	35
表 5-7 rPLL 位相調整の参照テーブル .....	35
表 5-8 rPLL デューティサイクル調整の参照テーブル .....	36
表 5-9 rPLL 遅延調整の参照テーブル .....	37
表 5-10 PLLVR 対応デバイス .....	45
表 5-11 PLLVR のポートの説明 .....	46
表 5-12 PLLVR のパラメータの説明 .....	46



表 5-13 PLLO 対応デバイス .....	53
表 5-14 PLLO のポートの説明 .....	55
表 5-15 PLLO のパラメータの説明 .....	56
表 5-16 IDSEL ポートのパラメータ参照テーブル .....	60
表 5-17 FBDSEL ポートのパラメータ参照テーブル .....	61
表 5-18 ODSELX (X=A/B/C/D) ポートのパラメータ参照テーブル .....	61
表 5-19 PLLO デューティサイクル微調整の参照テーブル .....	63
表 5-20 DLLDLY のポートの説明 .....	78
表 5-21 DLLDLY のパラメータの説明 .....	78
表 5-22 CLKDIV のポートの説明 .....	82
表 5-23 CLKDIV のパラメータの説明 .....	83
表 5-24 CLKDIVG 対応デバイス .....	86
表 5-25 CLKDIVG のポートの説明 .....	86
表 5-26 CLKDIVG のパラメータの説明 .....	87
表 5-27 DQS 対応デバイス .....	89
表 5-28 DQS のポートの説明 .....	90
表 5-29 DQS のパラメータの説明 .....	91
表 6-1 OSC 対応デバイス .....	95
表 6-2 OSC のポートの説明 .....	96
表 6-3 OSC のパラメータの説明 .....	96
表 6-4 OSCZ 対応デバイス .....	97
表 6-5 OSCZ のポートの説明 .....	98
表 6-6 OSCZ のパラメータの説明 .....	98
表 6-10 OSCH 対応デバイス .....	99
表 6-11 OSCH のポートの説明 .....	100
表 6-12 OSCH のパラメータの説明 .....	100
表 6-13 OSCO 対応デバイス .....	101
表 6-14 OSCO のポートの説明 .....	101
表 6-15 OSCO のパラメータの説明 .....	101
表 6-16 OSCW 対応デバイス .....	103
表 6-17 OSCW のポートの説明 .....	103
表 6-18 OSCW パラメータの説明 .....	103

# 1 本マニュアルについて

## 1.1 マニュアル内容

本マニュアルでは、クロックリソースの機能、プリミティブの定義、およびその使用方法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 Web サイト [www.gowinsemi.com/ja](http://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：

- GW1N シリーズ FPGA 製品データシート ([DS100](#))
- GW1NR シリーズ FPGA 製品データシート([DS117](#))
- GW1NS シリーズ FPGA 製品データシート([DS821](#))
- GW1NSR シリーズ FPGA 製品データシート([DS861](#))
- GW1NSE シリーズ FPGA 製品データシート([DS871](#))
- GW1NZ シリーズ FPGA 製品データシート([DS841](#))
- GW2A シリーズ FPGA 製品データシート ([DS102](#))
- GW2AR シリーズ FPGA 製品データシート([DS226](#))

## 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
CIU	Configurable Interface Unit	コンフィギャラブル・インターフェース・ユニット
CLKDIV	Clock Divider	クロック分周器
CRU	Configurable Routing Unit	コンフィギャラブル配線ユニット
DCC	Duty Cycle Correction	デューティーサイクル補正モジュール

用語、略語	正式名称	意味
DCS	Dynamic Clock Selector	ダイナミック・クロック・セレクト
DHCEN	Dynamic HCLK Clock Enable with Inverted Gate	ダイナミック高速クロック・イネーブル
DLLDLY	DLL Delay	DLL遅延
DQCE	Dynamic Quadrant Clock Enable	ダイナミック・クオドラント・クロック・イネーブル
DQS	Bidirectional Data Strobe Circuit for DDR Memory	双方向データストロブ回路
GCLK	Global Clock	グローバルクロック
HCLK	High-speed Clock	高速クロック
OSC	Oscillator	オシレータ
PCLK	Primary Clock	プライマリクロック
PLL	Phase-locked Loop	位相同期回路
SCLK	Segmented Clock	セグメントクロック

## 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。

Web サイト : [www.gowinsemi.com/ja](http://www.gowinsemi.com/ja)

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

# 2 概要

本セクションでは、GOWIN セミコンダクターFPGA 製品の、専用のクロック入力、バッファ、及び配線リソースを含むクロックリソースについて紹介します。基本的なクロックリソースとして、高周波信号に適した低電気容量、低スキューの配線が提供されています。これらのリソースは最大限までクロックスキューを減少してパフォーマンスを向上させることができ、すべてのクロック信号に適用できます。

クロックリソースと配線は、FPGA の高性能アプリケーションにとって重要です。GOWIN セミコンダクターFPGA 製品は、直接にデバイスのあらゆるリソースに接続される専用のグローバルクロック (GCLK(プライマリクロック PCLK およびロングワイヤ LW を含む)) を提供しています。さらに、位相同期回路 (PLL)、高速クロック (HCLK)、および DQS 等のクロックリソースも提供されています。

## 2.1 グローバルクロック

GCLK は製品で複数の象限に配置されています。LittleBee®ファミリーの 1K、2K、および 4K FPGA 製品では、GCLK は、L と R の 2 つの象限に配置されています (図 2-1)。LittleBee®ファミリーの 9K FPGA 製品および Arora ファミリーの FPGA 製品では、GCLK は、BL、BR、TL、および TR の 4 つの象限に配置されています (図 2-2)。各象限には 8 つの GCLK ネットワークがあります。各 GCLK のクロックソースとして、専用のクロック入力ピン及び汎用配線リソースが使用できます。専用のクロック入力ピンは、より良好なクロックパフォーマンスを提供できます。

LW は、DFF にクロックイネーブル (CE) およびセット/リセット (SET/RESET) 信号を提供するための制御ラインとして使用できます。それと同時に、論理配線としてつまり通常のデータ信号として使用することもできます。

図 2-1 LittleBee®ファミリーの1K、2K、および4K FPGA 製品での GCLK の配置

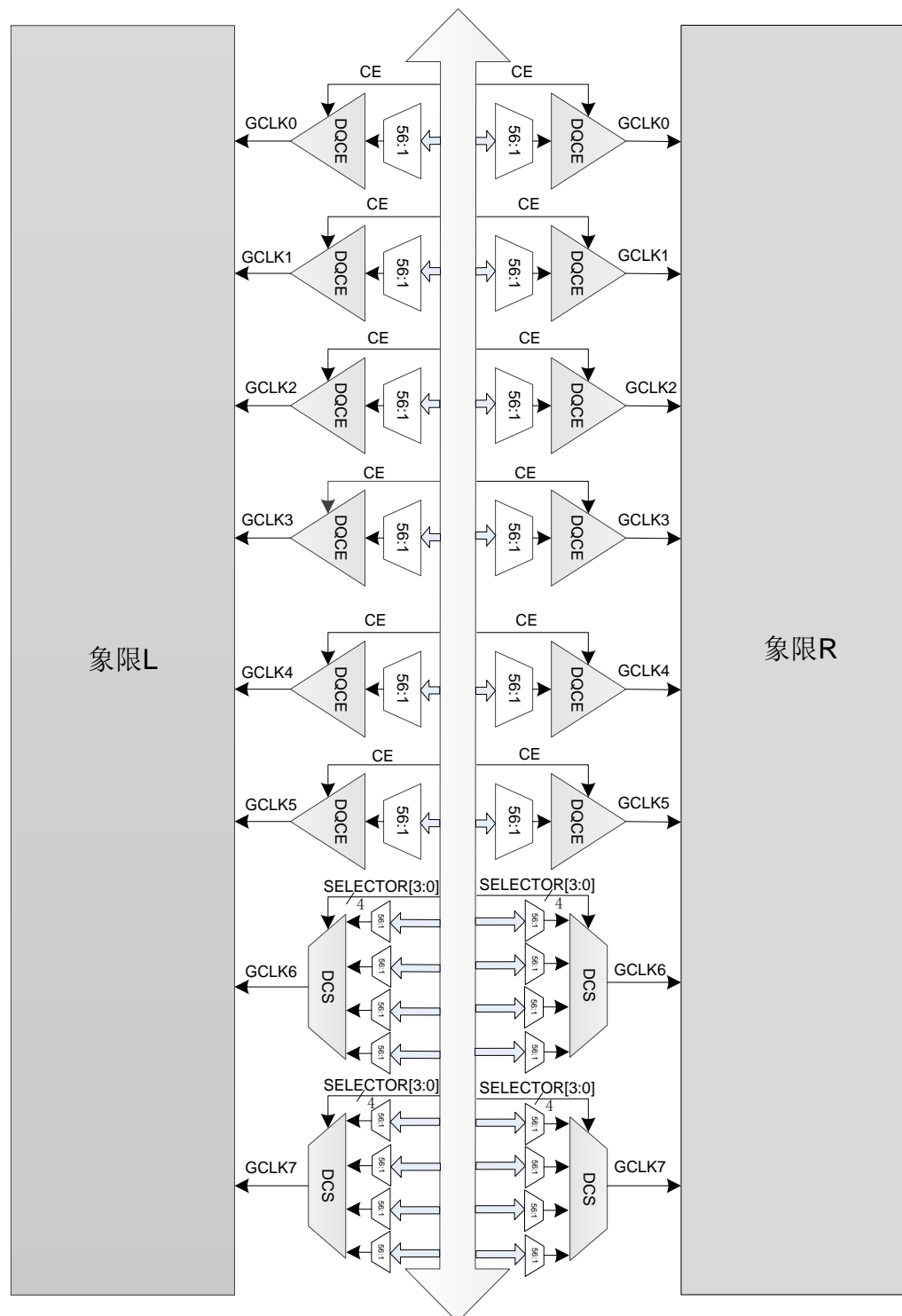
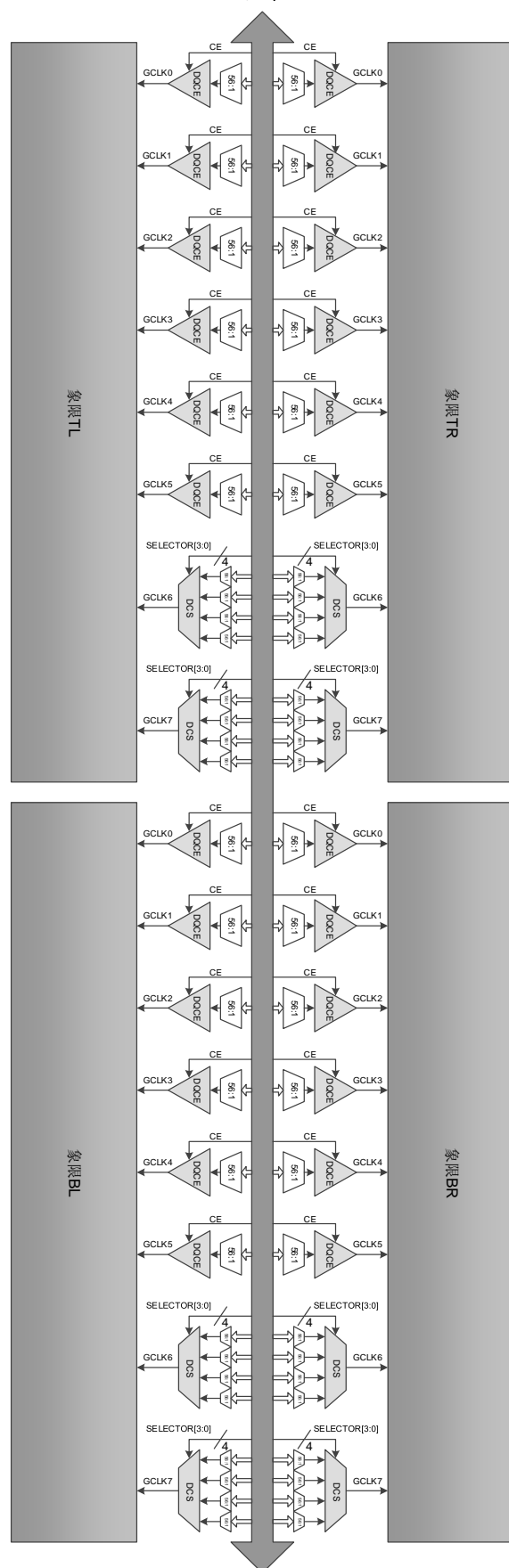


図 2-2 LittleBee®ファミリーの 9K FPGA 製品および Arora ファミリーの FPGA 製品での GCLK の配置



各象限の GCLK0~GCLK5 は、DQCE によって動的にオン/オフされます。GCLK0~GCLK5 クロックをオフにすると、GCLK0~GCLK5 によって駆動される内部ロジックは反転しなくなり、それによりデバイスの総消費電力が低下します。

各象限の GCLK6~GCLK7 は、DCS により制御され、内部ロジックは CRU を介して 4 つのクロック入力から動的に選択し、グリッチのないクロックを出力することができます。

## 2.2 高速クロック

低ジッタと低スキューの特徴を備えた高速クロック(HCLK)は、I/O 高性能のデータ転送を可能にし、主にソース同期データ転送プロトコルに適しています。高速クロック HCLK には 1 つの HCLKMUX モジュールがあります。HCLKMUX は、何れか 1 つの Bank 内の HCLK クロック入力信号をその他何れか 1 つの Bank 内に送信することができるため、HCLK の使用をより柔軟にしています。

HCLK リソースの配置については、次のマニュアルを参照してください。

- GW1N シリーズ FPGA 製品データシート ([DS100](#))
- GW1NR シリーズ FPGA 製品データシート([DS117](#))
- GW1NS シリーズ FPGA 製品データシート([DS821](#))
- GW1NSR シリーズ FPGA 製品データシート([DS861](#))
- GW1NSE シリーズ FPGA 製品データシート([DS871](#))
- GW1NZ シリーズ FPGA 製品データシート([DS841](#))
- GW2A シリーズ FPGA 製品データシート ([DS102](#))
- GW2AR シリーズ FPGA 製品データシート([DS226](#))

## 2.3 位相同期回路

位相同期回路(Phase-Locked Loop、PLL)はフィードバック制御回路です。PLL は、外部からのリファレンスクロック信号により、ループ内部のオシレータ信号の周波数と位相を制御します。

Gowin FPGA の PLL モジュールは、合成可能なクロック周波数を提供しており、パラメータを設定することで、クロックの周波数調整(逡倍及び分周)、位相調整、デューティサイクル調整等の機能を実現できます。

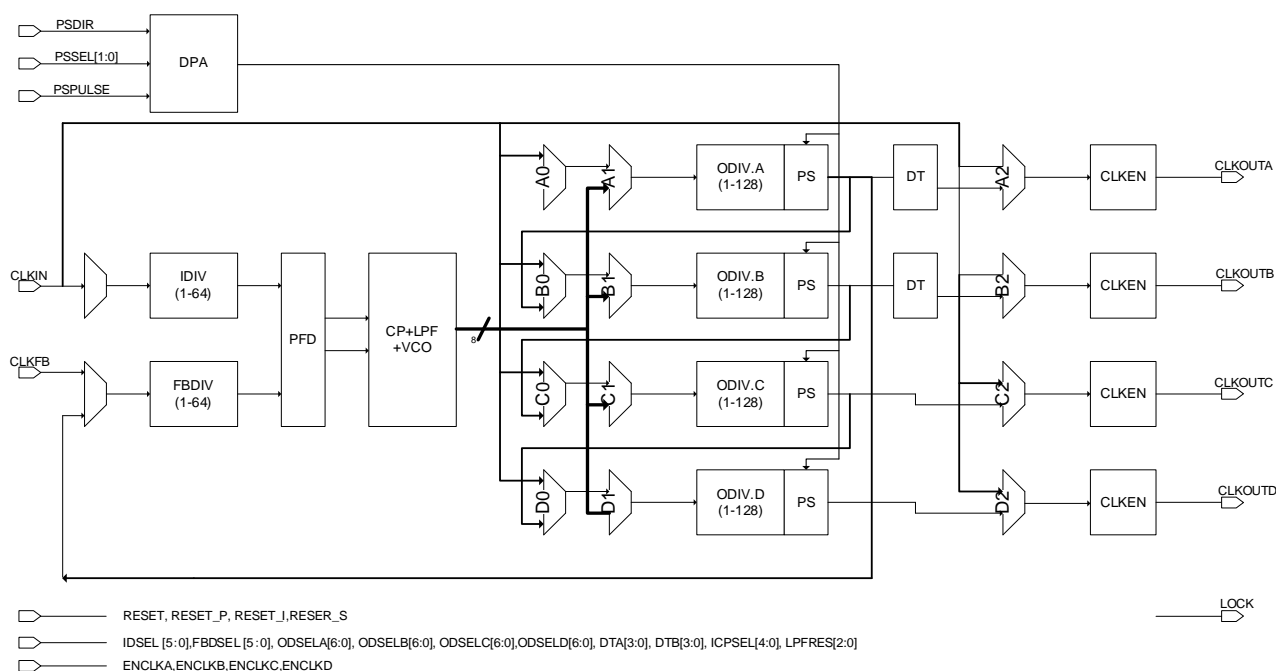
### 2.3.1 GW1N-1P5/ GW1N-2/ GW1NR-2/ GW2AN-18X/ GW2AN-9X

注記：

このセクションで説明されている内容は、GW1N-1P5/GW1N-2/GW1NR-2/GW2AN-18X/GW2AN-9X デバイスにのみ適用されます。

PLL のアーキテクチャは、図 2-3 に示す通りです。

図 2-3 PLL の説明図



PLL ポートの定義は表 2-1 に示す通りです。

表 2-1 PLL ポートの定義

ポート名	信号	説明
CLKIN	入力	リファレンスクロック入力
CLKFB	入力	フィードバッククロック入力
RESET	入力	PLL 全部リセット
RESET_P	入力	PLL のパワーダウン(Power Down)信号
RESET_I	入力	IDIV 付き PLL リセット
RESET_S	入力	B/C/D の 3 つのチャンネルのリセット
IDSEL[5:0]	入力	IDIV の動的制御、範囲は 1~64
FBDSEL[5:0]	入力	FBDIV の動的制御、範囲は 1~64
ODSELA[6:0]	入力	ODIVA の動的制御、範囲は 1~128
ODSELB[6:0]	入力	ODIVB の動的制御、範囲は 1~128
ODSELC[6:0]	入力	ODIVC の動的制御、範囲は 1~128
ODSELD[6:0]	入力	ODIVD の動的制御、範囲は 1~128
DTA[3:0]	入力	CLKOUTA のデューティサイクルの動的制御
DTB[3:0]	入力	CLKOUTB のデューティサイクルの動的制御
ICPSEL[4:0]	入力	ICP サイズの動的制御
LPFRES[2:0]	入力	LPFRES サイズの動的制御
PS DIR	入力	位相シフト方向の動的制御
PSSEL[1:0]	入力	位相シフトチャンネル選択の動的制御
PSPULSE	入力	位相シフトクロックの動的制御
ENCLKA ENCLKB ENCLKC ENCLKD	出力	クロック出力イネーブルの動的制御
CLKOUTA	出力	A チャンネルのクロック出力(デフォルト)
CLKOUTB	出力	B チャンネルのクロック出力(デフォルト)



ポート名	信号	説明
CLKOUTC	出力	C チャンネルのクロック出力(デフォルト)
CLKOUTD	出力	D チャンネルのクロック出力(デフォルト)
LOCK	出力	PLL ロック指示 : 1 : ロックする 0 : ロック解除

PLL のリファレンスクロック信号は、外部 PLL ピンから提供されるか、内部配線を使用した GCLK、HCLK、または通常のデータ信号により提供されます。PLL のフィードバック信号は、外部 PLL ピンから提供されるか、内部配線を使用した GCLK、HCLK、または通常のデータ信号により提供されます。

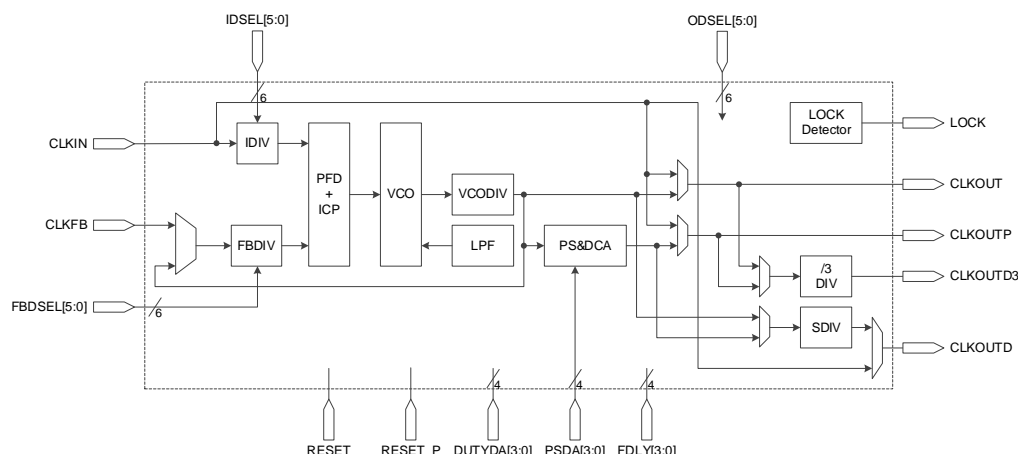
## 2.3.2 LittleBee®ファミリーと Arora ファミリーのその他のデバイス

注記：

このセクションで説明されている内容は、GW1N-1P5/GW1N-2/GW1NR-2/GW2AN-18X/GW2AN-9X デバイスを除く LittleBee®ファミリーと Arora ファミリーのデバイスに適用されます。

PLL のアーキテクチャは、図 2-4 に示す通りです。

図 2-4 PLL の説明図



PLL ポートの定義は表 2-2 に示す通りです。

表 2-2 PLL ポートの定義

ポート名	信号	説明
CLKIN	入力	リファレンスクロック入力
CLKFB	入力	フィードバッククロック入力
RESET	入力	PLL 全部リセット
RESET_P	入力	PLL のパワーダウン(Power Down)信号
IDSEL [5 : 0]	入力	IDIV の動的制御、範囲は 1~64
FBDSEL [5 : 0]	入力	FB DIV の動的制御、範囲は 1~64
PSDA [3 : 0]	入力	位相の動的制御(立ち上がりエッジで有効)
DUTYDA [3 : 0]	入力	デューティサイクルの動的制御(立ち下がりエッジで有効)
FDLY [3 : 0]	入力	CLKOUTP 遅延の動的制御

ポート名	信号	説明
CLKOUT	出力	位相とデューティサイクル調整なしのクロック出力
CLKOUTP	出力	位相とデューティサイクル調整ありのクロック出力
CLKOUTD	出力	CLKOUT または CLKOUTP からの分周クロック (SDIV により制御)
CLKOUTD3	出力	CLKOUT または CLKOUTP からの分周クロック (DIV3 により制御され、分周値は 3)
LOCK	出力	PLL ロック指示 : 1 : ロックする 0 : ロック解除

PLL のリファレンスクロック信号は、外部 PLL ピンから提供されるか、内部配線を使用した GCLK、HCLK、または通常のデータ信号により提供されます。PLL のフィードバック信号は、外部 PLL ピンから提供されるか、内部配線を使用した GCLK、HCLK、または通常のデータ信号により提供されます。

# 3 グローバルクロック

## 3.1 DQCE

### 3.1.1 プリミティブの紹介

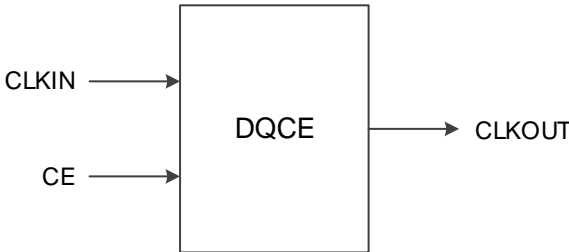
FPGA デバイスには、象限内の PCLK ネットワークを内部ロジックにより動的に有効または無効にできる動的クロック制御機能（DQCE）が備わっています。さらに、DQCE はパラメータの構成により無効にすることができ、この場合、PCLK ネットワークは常に有効です。PCLK クロックネットワークが無効になると、そのクロックによって駆動されるすべてのロジックはトグルしなくなるため、デバイスの全体的な消費電力が低下します。

#### 機能の説明

DQCE により、GCLK0~GCLK5 を動的にオン/オフできます。GCLK0~GCLK5 クロックをオフにすると、GCLK0~GCLK5 によって駆動される内部ロジックは反転しなくなり、それによりデバイスの総消費電力が低下します。DQCE を正常に動作させるには、CLKIN 信号に少なくとも 1 つの立ち下がりエッジが必要です。

#### ポート図

図 3-1 DQCE のポート図



#### ポートの説明

表 3-1 DQCE のポートの説明

ポート名	I/O	説明
CLKIN	入力	クロック入力信号

ポート名	I/O	説明
CE	入力	クロックイネーブル信号、 アクティブHigh
CLKOUT	出力	クロック出力信号

### プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。

#### Verilog でのインスタンス化：

```
DQCE dqce_inst (
    .CLKIN(clkin),
    .CE(ce),
    .CLKOUT(clkout)
);
```

#### VHDL でのインスタンス化：

```
COMPONENT DQCE
    PORT(
        CLKOUT:OUT std_logic;
        CE:IN std_logic;
        CLKIN:IN std_logic
    );
END COMPONENT;

uut:DQCE
PORT MAP(
    CLKIN=>clkin,
    CLKOUT=>clkout,
    CE=>ce
);
```

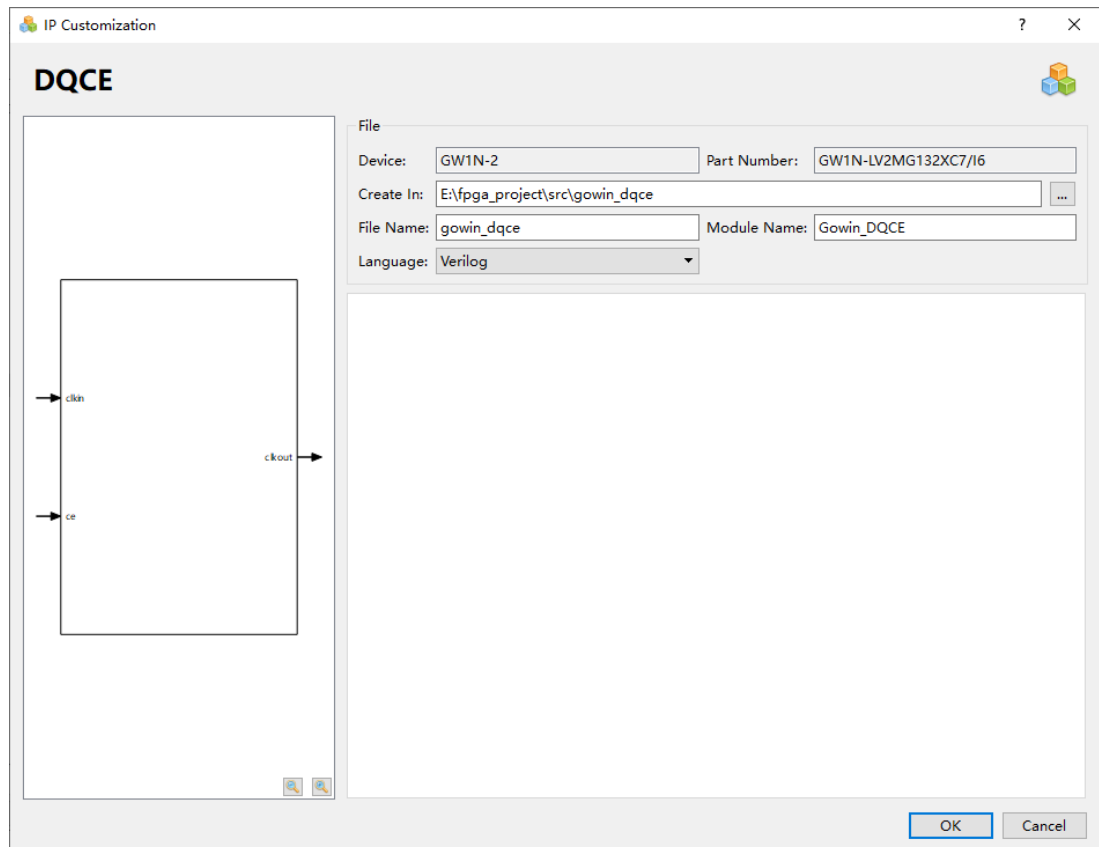
### 3.1.2 IP の呼び出し

IP Core Generator のインターフェースで DQCE をクリックすると、右側に DQCE の概要が表示されます。

#### IP の構成

IP Core Generator インターフェースで"DQCE"をダブルクリックすると、"IP Customization"ウィンドウがポップアップします。このウィンドウには File 構成タブとポート図があります（図 3-2）。

図 3-2 DQCE IP の構成ウィンドウ



#### 1. File 構成タブ

File 構成タブは、IP ファイルの構成に使用されます。

- **Device** : 対象デバイス。
- **Part Number** : パーツ番号。
- **Language** : IP を実現するハードウェア記述言語。右側のドロップダウンリストからターゲット言語（Verilog または VHDL）を選択します。
- **Module Name** : 生成される IP ファイルのモジュール名。右側のテキストボックスで編集できます。**Module Name** をプリミティブ名と同じにすることはできません。同じ場合、エラーメッセージがポップアップします。
- **File Name** : 生成される IP ファイルのファイル名。右側のテキストボックスで再編集できます。
- **Create In** : 生成される IP ファイルのパス。右側のテキストボックスでパスを直接編集するか、テキストボックスの右側にある選択ボタンを使用してパスを選択できます。

#### 2. ポート図

ポート図は、IP Core の構成結果を表示します(図 3-2)。

### 生成されるファイル

IP の構成が完了したら、構成ファイルの"File Name"によって命名された 3 つのファイルが生成されます：

- "gowin\_dqce.v"は完全な verilog モジュールです。
- "gowin\_dqce\_tmp.v"は IP のテンプレートファイルです。
- "gowin\_dqce.ipc"は IP の構成ファイルです。

注記：

VHDL が設計の言語として選択されている場合、生成される最初の 2 つのファイル名のサフィックスは.vhd になります。

## 3.2 DCS

### 3.2.1 プリミティブの紹介

各象限には GCLK6 と GCLK7 に対応する 2 つの DCS があります。DCS の出力は GCLK6 または GCLK7 に接続されます。つまり 1 象限内の 8 つの GCLK のうち GCLK6 と GCLK7 のみが動的クロック選択 (DCS) 機能を備えています。DCS のクロック選択信号 CLKSEL は CIU から供給され、内部ロジックは CRU を介して 4 つのクロック入力を動的に切り替えることができます。

#### 機能の説明

各象限の GCLK6~GCLK7 は DCS で制御され、4 つの入力クロックのうち 1 つがグローバルクロックとして選択されます。内部ロジックは CRU を介して 4 つのクロック入力から動的に選択し、グリッチのないクロックを出力することができます。

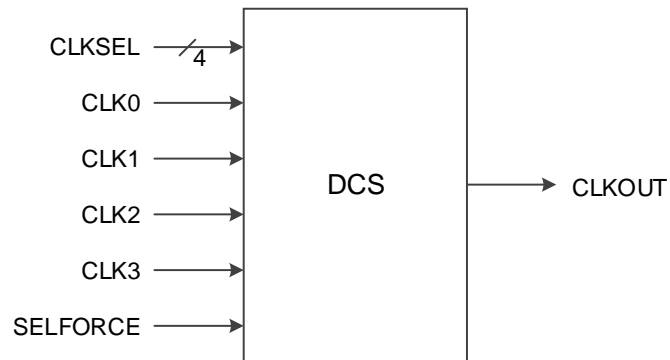
DCS には、「Non-Glitchless」モードと「Glitchless」モードの 2 つのクロック切り替えモードがあります。

Non-Glitchless モードでは、DCS は通常のマルチプレクサとして機能し、CLKSEL 信号のみを利用してクロック信号を切り替えます。出力のグリッチが許容されます。

Glitchless モードでは、パラメータ DCS\_MODE を使用してモードを設定できます。CLKSEL 信号を利用してクロック信号を動的に切り替えることにより、出力クロックのグリッチを回避できます。

## ポート図

図 3-3 DCS のポート図



## ポートの説明

表 3-2 DCS のポートの説明

ポート名	I/O	説明
CLK0	入力	クロック入力信号0
CLK1	入力	クロック入力信号1
CLK2	入力	クロック入力信号2
CLK3	入力	クロック入力信号3
CLKSEL[3:0]	入力	クロック選択信号
SELFFORCE	入力	強制モード選択 0 : glitchlessモード 1 : Non-glitchlessモード
CLKOUT	出力	クロック出力信号

## パラメータの説明

表 3-3 DCS のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
DCS_MODE	"CLK0", "CLK1", "CLK2", "CLK3", "GND", "VCC", "RISING", "FALLING", "CLK0_GND", "CLK1_GND", "CLK2_GND", "CLK3_GND", "CLK0_VCC", "CLK1_VCC", "CLK2_VCC", "CLK3_VCC"	"RISING"	DCSモードを設定します。

## プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。

**Verilog** でのインスタンス化 :

```
DCS dcs_inst (
    .CLK0(clk0),
```

```

        .CLK1(clk1),
        .CLK2(clk2),
        .CLK3(clk3),
        .CLKSEL(clksel[3:0]),
        .SELFFORCE(selfforce),
        .CLKOUT(clkout)
    );
    defparam dcs_inst.DCS_MODE="RISING";
VHDL でのインスタンス化 :
    COMPONENT DCS
        GENERIC(DCS_MODE:string:="RISING");
        PORT(
            CLK0:IN std_logic;
            CLK1:IN std_logic;
            CLK2:IN std_logic;
            CLK3:IN std_logic;
            CLKSEL:IN std_logic_vector(3 downto 0);
            SELFFORCE:IN std_logic;
            CLKOUT:OUT std_logic
        );
    END COMPONENT;
    uut:DCS
        GENERIC MAP(DCS_MODE=>"RISING")
        PORT MAP(
            CLK0=>clk0,
            CLK1=>clk1,
            CLK2=>clk2,
            CLK3=>clk3,
            CLKSEL=>clksel,
            SELFFORCE=>selfforce,
            CLKOUT=>clkout
        );

```

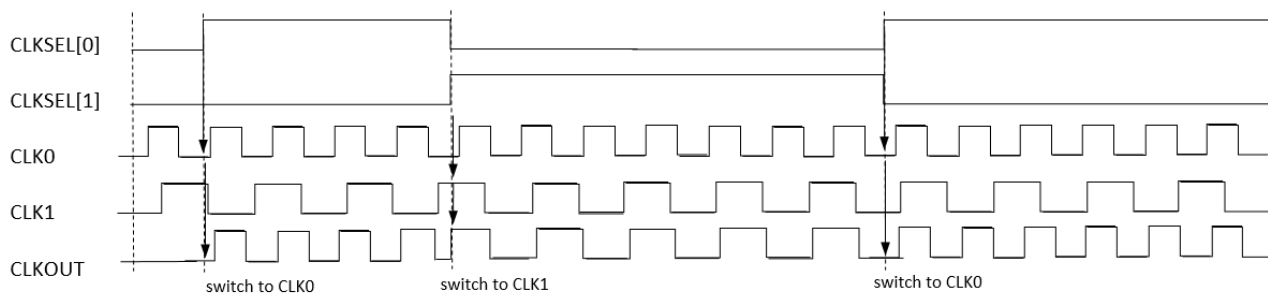
#### タイミング図

Non-Glitchless モードのタイミングは図 3-4 に示すとおりです。  
CLKSEL[3]~CLKSEL[0]はそれぞれ CLK3~CLK0 を選択するために使用さ



れ、アクティブ High で、切り替えタイミングは同じです。

図 3-4 Non-Glitchless モードのタイミング図



Glitchless モードのタイミングは図 3-5～図 3-8 に示すとおりです。  
CLKSEL[3]~CLKSEL[0]はそれぞれ CLK3~CLK0 を選択するために使用され、切り替えタイミングは同じです。

図 3-5 DCS mode が RISING の場合のタイミング

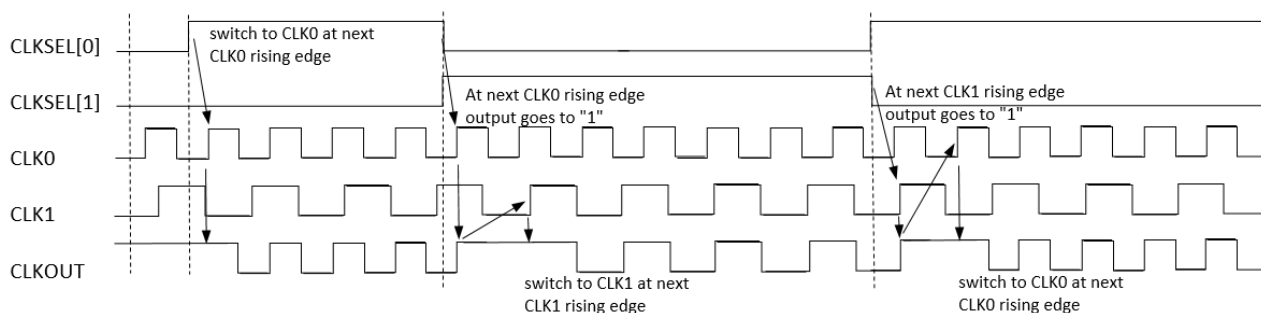


図 3-6 DCS mode が FALLING の場合のタイミング

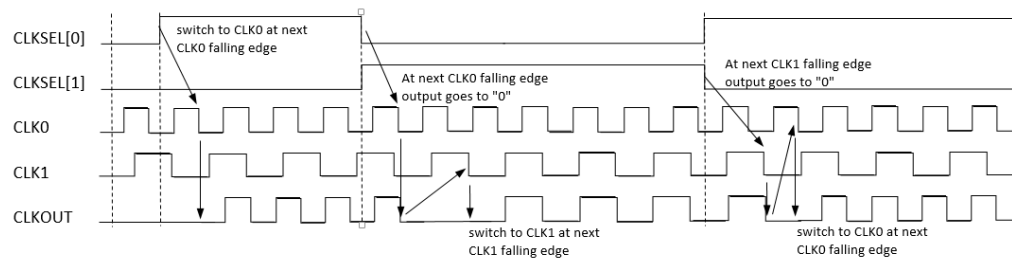


図 3-7 DCS mode が CLK0\_GND の場合のタイミング

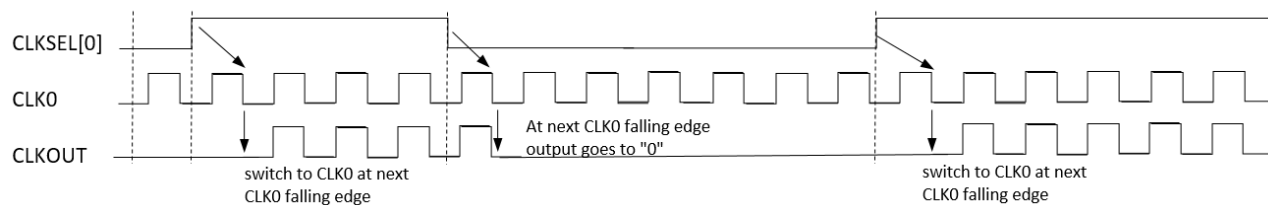
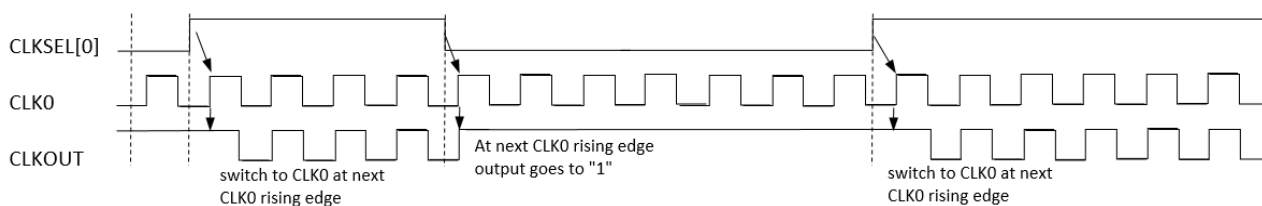


図 3-8 DCS mode が CLK0\_VCC の場合のタイミング



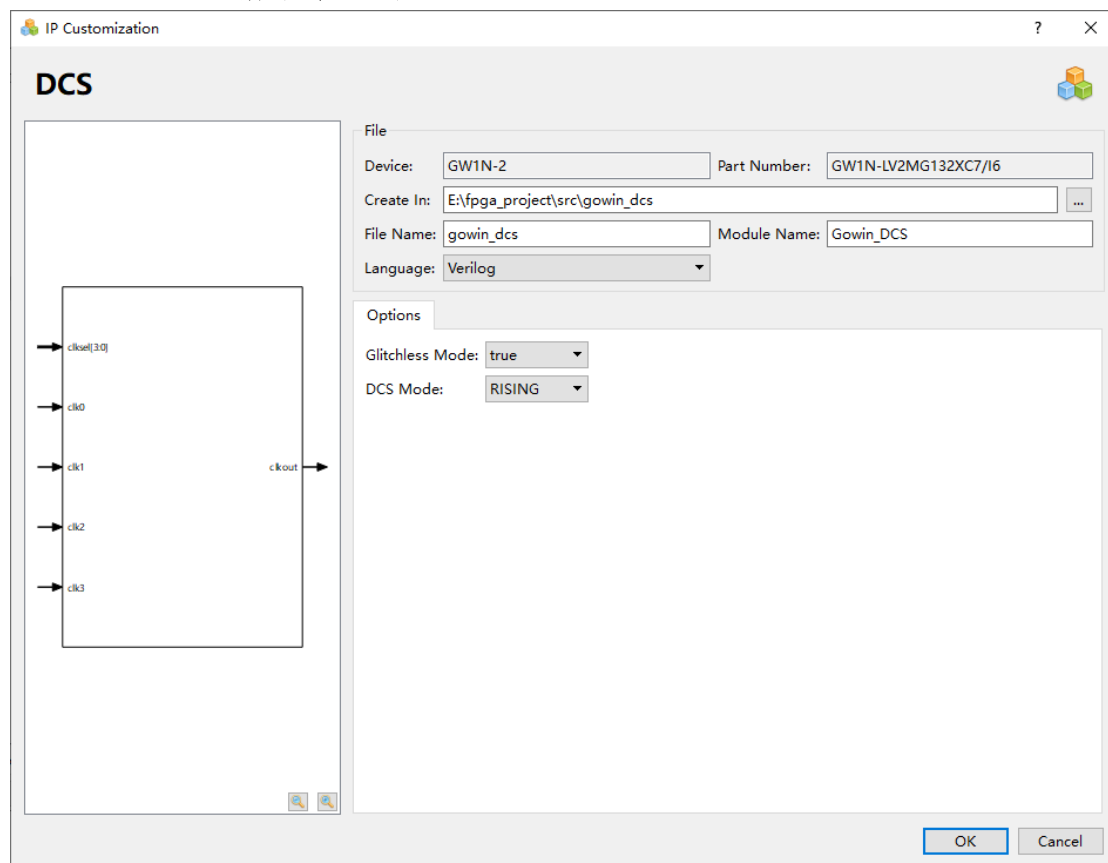
### 3.2.2 IP の呼び出し

IP Core Generator のインターフェースで **DCS** をクリックすると、右側に DCS の概要が表示されます。

#### IP の構成

IP Core Generator インターフェースで**"DCS"**をダブルクリックすると、**"IP Customization"**ウィンドウがポップアップします。このウィンドウには **File** 構成タブ、**Options** 構成タブ、およびポート図があります (図 3-9)。

図 3-9 DCS IP の構成ウィンドウ



#### 1. File 構成タブ

**File** 構成タブは、IP ファイルの構成に使用されます。DCS の **File** 構成タブの使用は **DQCE** モジュールと同様です。DQCE の **File** 構成タブを

参照してください。

## 2. Options 構成タブ

Options 構成タブは IP のカスタマイズに使用されます(図 3-9)。

- Glitchless Mode : Glitchless モードのイネーブルオプション。
- DCS Mode : DCS モードの設定。

## 3. ポート図

ポート図は、IP Core の構成結果を表示します(図 3-9)。

### 生成されるファイル

IP の構成が完了したら、構成ファイルの"File Name"によって命名された 3 つのファイルが生成されます：

- "gowin\_dcs.v"は完全な verilog モジュールです。
- gowin\_dcs\_tmp.v は IP のテンプレートファイルです。
- gowin\_dcs.ipc は IP の構成ファイルです。

注記：

VHDL が設計の言語として選択されている場合、生成される最初の 2 つのファイル名のサフィックスは.vhd になります。

# 4高速クロック

## 4.1 DHCEN

### 4.1.1 プリミティブの紹介

DHCEN は、HCLK 高速クロック信号を動的にオン/オフでき、CE が Low のときに有効になります。

ポート図

図 4-1 DHCEN のポート図



ポートの説明

表 4-1 DHCEN のポートの説明

ポート名	I/O	説明
CLKIN	入力	クロック入力信号
CE	入力	クロックイネーブル信号、アクティブ Low
CLKOUT	出力	クロック出力信号

### プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。

**Verilog** でのインスタンス化：

```
DHCEN dhcen_inst (  
    .CLKIN(clkin),  
    .CE(ce),  
    .CLKOUT(clkout)
```

```
);
```

**VHDL** でのインスタンス化 :

```
COMPONENT DHCEN
```

```
    PORT(
```

```
        CLKOUT:OUT std_logic;
```

```
        CE:IN std_logic;
```

```
        CLKIN:IN std_logic
```

```
    );
```

```
END COMPONENT;
```

```
uut:DHCEN
```

```
PORT MAP(
```

```
    CLKIN=>clkin,
```

```
    CLKOUT=>clkout,
```

```
    CE=>ce
```

```
);
```

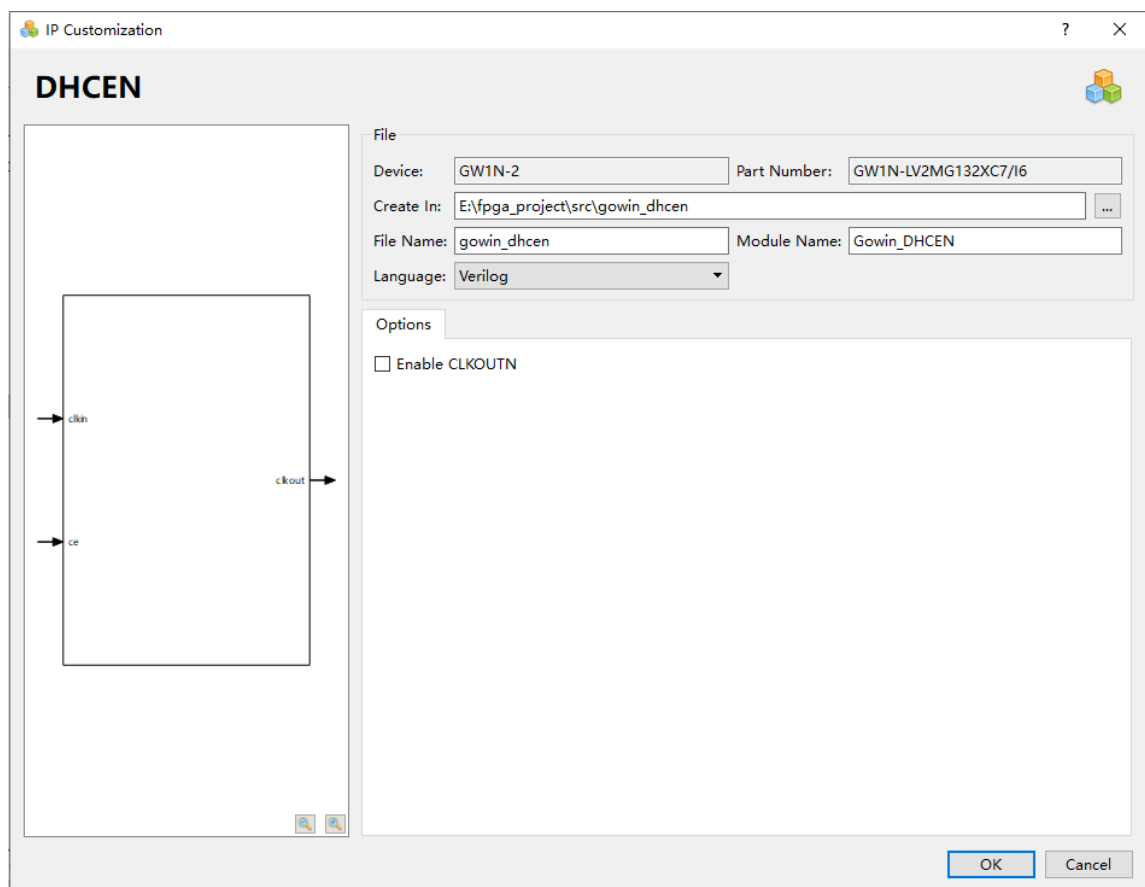
### 4.1.2 IP の呼び出し

IP Core Generator のインターフェースで **DHCEN** をクリックすると、右側に **DHCEN** の概要が表示されます。

#### IP の構成

IP Core Generator インターフェースで"**DHCEN**"をダブルクリックすると、"**IP Customization**"ウィンドウがポップアップします。このウィンドウには **File** 構成タブ、**Options** 構成タブ、およびポート図があります (図 4-2)。

図 4-2 DHCEN IP の構成ウィンドウ



### 1. File 構成タブ

File 構成タブは、IP ファイルの構成に使用されます。DHCEN の File 構成タブの使用は DQCE モジュールと同様です。DQCE の File 構成タブを参照してください。

### 2. Options 構成タブ

Options 構成タブは IP のカスタマイズに使用されます(図 4-2)。

- **Enable CLKOUTN** : チェックされる場合は DHCENC がインスタンス化され、チェックされない場合は DHCEN がインスタンス化されます。

### 3. ポート図

ポート図は、IP Core の構成結果を表示します(図 4-2)。

## 生成されるファイル

IP の構成が完了したら、構成ファイルの"File Name"によって命名された 3 つのファイルが生成されます：

- "gowin\_dhcn.v"は完全な verilog モジュールです。
- "gowin\_dhcn\_tmp.v"は IP のテンプレートファイルです。
- "gowin\_dhcn.ipc"は IP の構成ファイルです。

注記：

VHDL が設計の言語として選択されている場合、生成される最初の 2 つのファイル名のサフィックスは.vhd になります。

## 4.2 DHCENC

### 4.2.1 プリミティブの紹介

DHCENC は、HCLK 高速クロック信号を動的にオン/オフでき、CE が Low のときに有効になります。

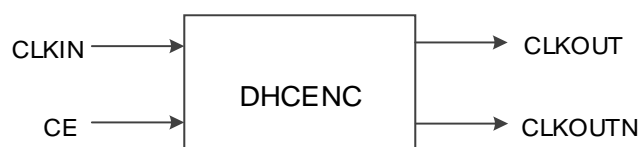
サポートされるデバイス

表 4-2 DHCENC 対応デバイス

ファミリー	シリーズ	デバイス
LittleBee®	GW1N	GW1N-9C, GW1N-2, GW1N-1P5, GW1N-2B, GW1N-1P5B
	GW1NR	GW1NR-9C, GW1NR-2, GW1NR-2B

## ポート図

図 4-3 DHCENC のポート図



## ポートの説明

表 4-3 DHCENC のポートの説明

ポート名	I/O	説明
CLKIN	入力	クロック入力
CE	入力	クロックイネーブル信号、アクティブ Low。
CLKOUT	出力	クロック出力
CLKOUTN	出力	クロック出力。CLKOUTの反転。

## プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。

**Verilog** でのインスタンス化 :

```
DHCENC dhcenc_inst (  
    .CLKIN(clkin),  
    .CE(ce),  
    .CLKOUT(clkout),  
    .CLKOUTN(clkoutn)  
);
```

**VHDL** でのインスタンス化 :

```
COMPONENT DHCENC  
    PORT(  
        CLKOUT:OUT std_logic;  
        CLKOUTN:OUT std_logic;  
        CE:IN std_logic;  
        CLKIN:IN std_logic  
    );  
END COMPONENT;  
uut:DHCENC  
PORT MAP(  
    CLKIN=>clkin,  
    CLKOUT=>clkout,  
    CLKOUTN=>clkoutn,  
    CE=>ce  
);
```

4.2.2 IP の呼び出し

DHCENC の呼び出しは DHCEN と同様で、4.1.2 IP の呼び出しを参照してください。

4.3 DCC

4.3.1 プリミティブの紹介

DCC は高速クロックのデューティサイクル補正モジュールです。  
サポートされるデバイス

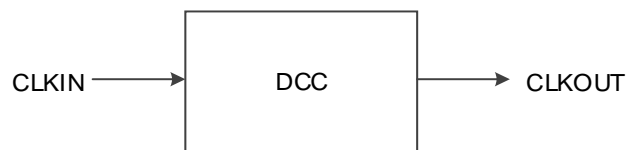
表 4-4 DCC 対応デバイス

ファミリー	シリーズ	デバイス
LittleBee®	GW1N	GW1N-9C
	GW1NR	GW1NR-9C



## ポート図

図 4-4 DCC のポート図



## ポートの説明

表 4-5 DCC のポートの説明

ポート名	I/O	説明
CLKIN	入力	クロック入力
CLKOUT	出力	クロック出力

## パラメータの説明

表 4-6 DCC のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
DCC_EN	1'b1, 1'b0	1'b1	1'b1 : DCCを有効にする 1'b0 : DCCを無効にする
FCLKIN	-	50.0	入力クロックの周波数

## プリミティブのインスタンス化

**Verilog** でのインスタンス化 :

```

DCC dcc_inst (
    .CLKIN(clkin),
    .CLKOUT(clkout)
);
defparam dcc_inst.DCC_EN=1'b1;
defparam dcc_inst.FCLKIN=50.0;

```

**VHDL** でのインスタンス化 :

```

COMPONENT DCC
    GENERIC (
        DCC_EN : bit := '1';  --'1':enable dcc; '0': disable dcc
        FCLKIN : REAL := 50.0 --frequency of the clkin(M)
    );
    PORT(
        CLKOUT:OUT std_logic;

```

```

        CLKIN:IN std_logic
    );
END COMPONENT;
uut:DCC
GENERIC MAP(
    DCC_EN=>'1',
    FCLKIN=>50.0
)

PORT MAP(
    CLKIN=>clkin,
    CLKOUT=>clkout
);
```

4.4 DCCG

4.4.1 プリミティブの紹介

DCCG は高速クロックのデューティサイクル補正モジュールです。

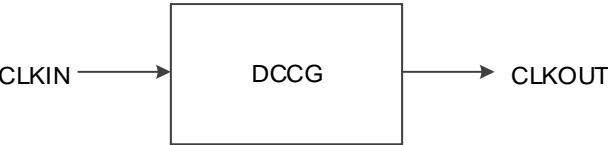
サポートされるデバイス

表 4-7 DCCG 対応デバイス

ファミリー	シリーズ	デバイス
LittleBee®	GW1N	GW1N-2, GW1N-1P5, GW1N-2B, GW1N-1P5B
	GW1NR	GW1NR-2, GW1NR-2B

ポート図

図 4-5 DCCG のポート図



ポートの説明

表 4-8 DCCG のポートの説明

ポート名	I/O	説明
CLKIN	入力	クロック入力
CLKOUT	出力	クロック出力

## パラメータの説明

表 4-9 DCCG のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
DCC_MODE	2'b00, 2'b01, 2'b10, 2'b11	2'b00	2'b00/2'b01:Buffered 2'b10: +80ps 2'b11: -80ps
FCLKIN	-	50.0	入力クロックの周波数

## プリミティブのインスタンス化

### Verilog でのインスタンス化 :

```
DCCG dccg_inst (
    .CLKIN(clkin),
    .CLKOUT(clkout)
);
defparam dccg_inst.DCC_MODE=2'b00;
defparam dccg_inst.FCLKIN=50.0;
```

### VHDL でのインスタンス化 :

```
COMPONENT DCCG
    GENERIC (
        DCC_MODE : bit_vector := "00";
        FCLKIN : REAL := 50.0 --frequency of the clkin(M)
    );
    PORT(
        CLKOUT:OUT std_logic;
        CLKIN:IN std_logic
    );
END COMPONENT;

uut:DCCG
GENERIC MAP(
    DCC_MODE=>"00",
    FCLKIN=>50.0
)

PORT MAP(
    CLKIN=>clkin,
    CLKOUT=>clkout
);
```

# 4.5 CLKDIV2

## 4.5.1 プリミティブの紹介

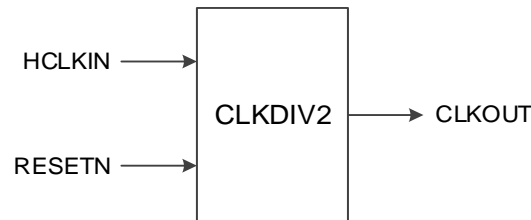
CLKDIV2 は、クロック周波数を 2 分周するクロック分周器です。  
CLKDIV2 の出力は、DCC/DCCG の CLKIN、IOLOGIC の FCLK、PLL の CLKIN と CLKFB、DQS の FCLK、および CLKDIV の HCLKIN のみを駆動できます。

### 機能の説明

CLKDIV2 は、入力クロックと位相が一致する 2 分周クロックを生成する高速クロック分周モジュールです。

### ポート図

図 4-6 CLKDIV2 のポート図



### ポートの説明

表 4-10 CLKDIV2 のポートの説明

ポート名	I/O	説明
HCLKIN	入力	クロック入力
RESETN	入力	非同期リセット信号、アクティブLow
CLKOUT	出力	クロック出力

### パラメータの説明

表 4-11 CLKDIV2 のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
GSREN	"false", "true"	"false"	グローバルリセット GSRを有効にする

### プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。

**Verilog** でのインスタンス化：

```
CLKDIV2 clkdiv2_inst (
```

```

        .HCLKIN(hclkin),
        .RESETN(resetn),
        .CLKOUT(clkout)
    );
    defparam clkdiv2_inst.GSREN="false";
VHDL でのインスタンス化 :
    COMPONENT CLKDIV2
        GENERIC(
            GSREN:STRING:="false"
        );
        PORT(
            HCLKIN:IN std_logic;
            RESETN:IN std_logic;
            CLKOUT:OUT std_logic
        );
    END COMPONENT;
    uut:CLKDIV2
        GENERIC MAP(
            GSREN=>"false"
        )
        PORT MAP(
            HCLKIN=>hclkin,
            RESETN=>resetn,
            CLKOUT=>clkout
        );

```

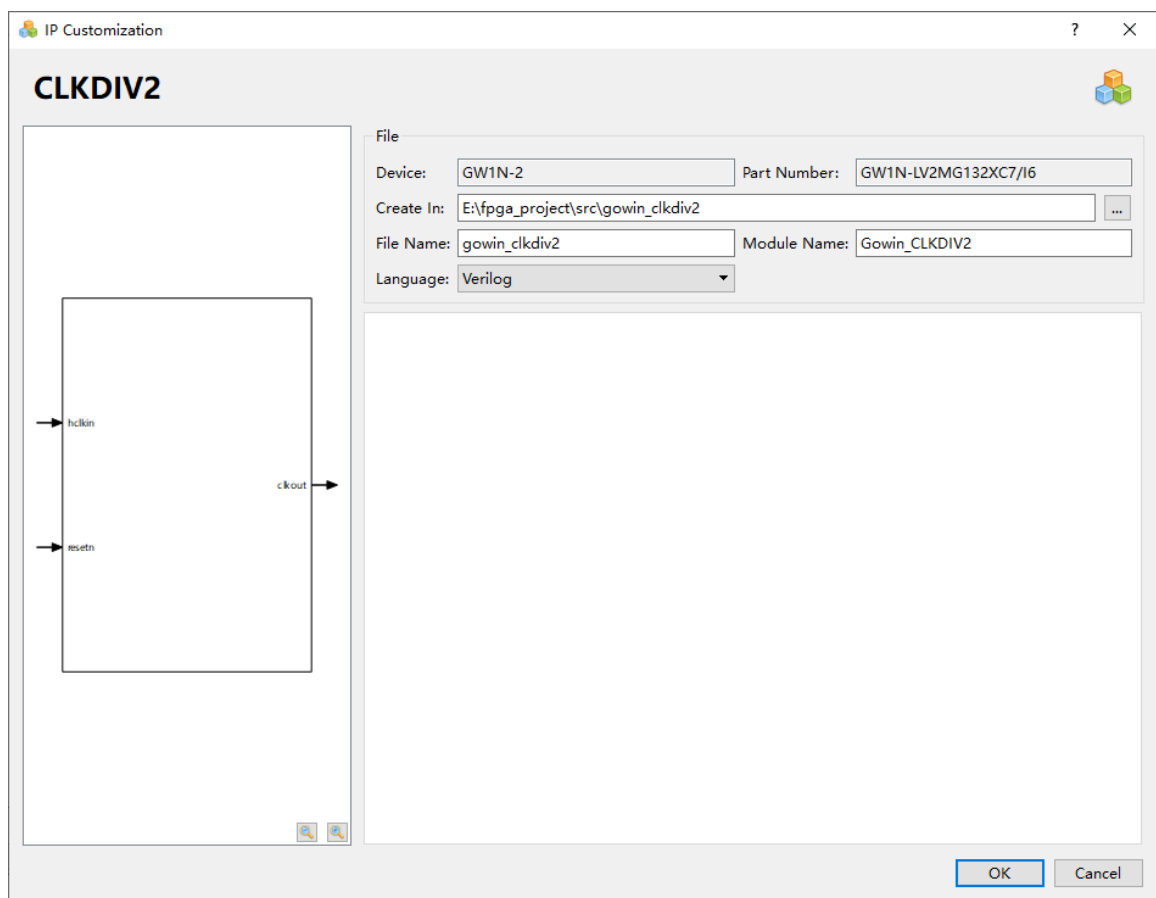
## 4.5.2 IP の呼び出し

IP Core Generator のインターフェースで **CLKDIV2** をクリックすると、右側に **CLKDIV2** の概要が表示されます。

### IP の構成

IP Core Generator インターフェースで "**CLKDIV2**" をダブルクリックすると、"**IP Customization**" ウィンドウがポップアップします。このウィンドウには **File** 構成タブ、**Options** 構成タブ、およびポート図があります (図 4-7)。

図 4-7 CLKDIV2 IP の構成ウィンドウ



### 1. File 構成タブ

File 構成タブは、IP ファイルの構成に使用されます。CLKDIV2 の File 構成タブの使用は DQCE モジュールと同様です。DQCE の File 構成タブを参照してください。

### 2. ポート図

ポート図は、IP Core の構成結果を表示します(図 4-7)。

### 生成されるファイル

IP の構成が完了したら、構成ファイルの"File Name"によって命名された 3 つのファイルが生成されます：

- "gowin\_clkdiv2.v"は完全な verilog モジュールです。
- "gowin\_clkdiv2\_tmp.v"は IP のテンプレートファイルです。
- "gowin\_clkdiv2.ipc"は IP の構成ファイルです。

### 注記：

VHDL が設計の言語として選択されている場合、生成される最初の 2 つのファイル名のサフィックスは.vhd になります。

# 5 システムクロック

## 5.1 rPLL

### 5.1.1 プリミティブの紹介

GOWIN FPGA の rPLL は、外部からのリファレンスクロック信号により、ループ内部のオシレータ信号の周波数と位相を制御します。

サポートされるデバイス

表 5-1 rPLL 対応デバイス

ファミリー	シリーズ	デバイス
Arora	GW2A	GW2A-18, GW2A-18C, GW2A-55, GW2A-55C
	GW2AN	GW2AN-55C
	GW2AR	GW2AR-18, GW2AR-18C
	GW2ANR	GW2ANR-18C
LittleBee®	GW1N	GW1N-1, GW1N-1S, GW1N-4, GW1N-4B, GW1N-4D, GW1N-9, GW1N-9C
	GW1NR	GW1NR-1, GW1NR-4, GW1NR-4B, GW1NR-4D, GW1NR-9, GW1NR-9C
	GW1NRF	GW1NRF-4B
	GW1NZ	GW1NZ-1, GW1NZ-1C

機能の説明

rPLL は、入力クロックに基づいてクロック位相調整、デューティサイクル調整、および周波数調整（逡倍と分周）を実行し、さまざまな位相と周波数のクロック出力を生成できます。

rPLL は入力クロック **CLKIN** に周波数調整（逡倍及び分周）を行うことができ、計算式は以下の通りです：

$$f_{CLKOUT} = (f_{CLKIN} * FBDIV) / IDIV$$

$$f_{VCO} = f_{CLKOUT} * ODIV$$

$$f_{CLKOUTD} = f_{CLKOUT} / SDIV$$

$$f_{PFD} = f_{CLKIN} / IDIV = f_{CLKOUT} / FBDIV$$

注記：

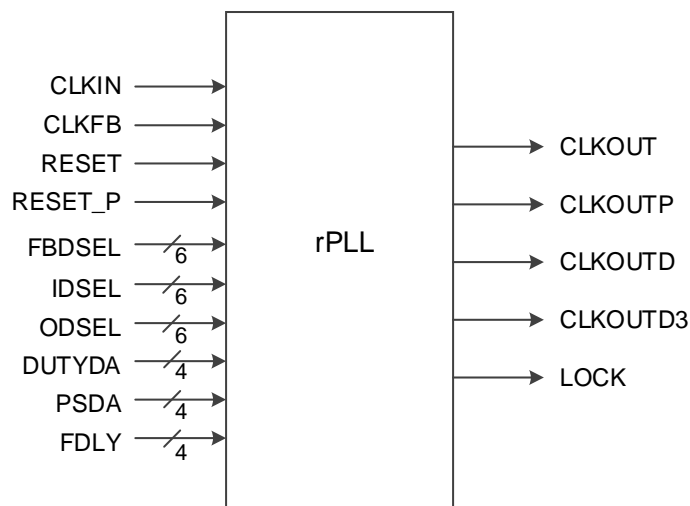
- $f_{CLKIN}$  は入力クロック **CLKIN** の周波数、 $f_{CLKOUT}$  は **CLKOUT** および **CLKOUTP** の周

波数、 $f_{CLKOUTD}$  は  $CLKOUTD$  の周波数、 $f_{PFD}$  は位相検出器周波数です。

- IDIV、FBDIV、ODIV、SDIV は、各分周器の分周係数です。分周係数を調整することにより、所望の周波数のクロック信号を生成することができます。
- rPLL の周波数の範囲については、[FPGA 製品データシート](#)を参照してください。

### ポート図

図 5-1 rPLL のポート図



### ポートの説明

表 5-2 rPLL のポートの説明

ポート名	I/O	説明
CLKIN	入力	リファレンスクロック入力
CLKFB	入力	フィードバッククロック入力
RESET	入力	rPLL非同期リセット入力、アクティブHigh
RESET_P	入力	rPLLパワーダウン（Power Down）入力、アクティブHigh。非bypassモードでは、RESET_PがHighの場合、CLKOUT/CLKOUTP/CLKOUTD/CLKOUTD3は0です。
FBDSEL[5:0]	入力	FBDIVの動的制御値、範囲は0~63、実際の値は64-FBDSEL
IDSEL[5:0]	入力	IDIVの動的制御値、範囲は0~63、実際の値は64-IDSEL
ODSEL[5:0]	入力	ODIVの動的制御値、2,4,8,16,32,48,64,80,96,112,128
DUTYDA[3:0]	入力	デューティサイクルの動的調整信号
PSDA[3:0]	入力	位相の動的調整信号
FDLY[3:0]	入力	遅延の動的微調整信号
CLKOUT	出力	rPLLクロック出力
LOCK	出力	rPLLロック指示。1はロック、0はロック解除を示す
CLKOUTP	出力	CLKOUTに基づく位相とデューティサイクル調整ありのrPLLクロック出力信号
CLKOUTD	出力	SDIVを経たrPLLクロック出力信号、SDIVを経たCLKOUTまたはCLKOUTPの出力信号



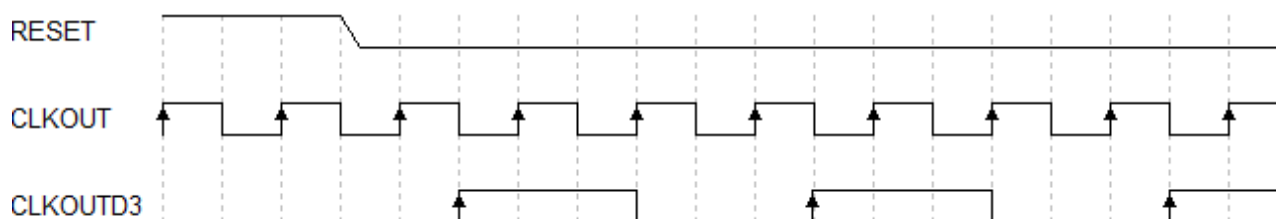
ポート名	I/O	説明
CLKOUTD3	出力	DIV3を経たrPLLクロック出力信号、DIV3を経たCLKOUTまたはCLKOUTPの出力信号

CLKOUTD3 は、3 分周された出力クロック信号で、2 つの入力ソースがあります。

- CLKOUTD3 の入力ソースが CLKOUT の場合：

図 5-2 に示すように、リセット終了後、CLKOUTD3 はクロック CLKOUT の最初の立ち下がりエッジで High になり、その後の 2 番目の立ち上がりエッジで Low になります。

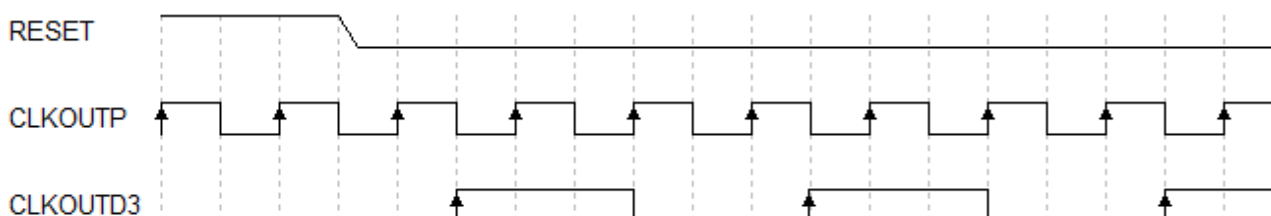
図 5-2 入力ソースが CLKOUT の場合の CLKOUTD3 のタイミング



- CLKOUTD3 の入力ソースが CLKOUTP の場合：

図 5-3 に示すように、リセット終了後、CLKOUTD3 はクロック CLKOUTP の最初の立ち下がりエッジで High になり、その後の 2 番目の立ち上がりエッジで Low になります。

図 5-3 入力ソースが CLKOUTP の場合の CLKOUTD3 のタイミング



## パラメータの説明

表 5-3 rPLL のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
FCLKIN	"3"~"500"	"100"	リファレンスクロックの周波数
IDIV_SEL	0~63	0	IDIV分周係数の静的設定
DYN_IDIV_SEL	"true", "false"	"false"	IDIV分周係数の静的制御パラメータまたは動的制御信号の選択 false: 静的制御、つまり IDIV_SEL を選択 true: 動的制御、つまり IDSEL を選択
FBDIV_SEL	0~63	0	FBDIV分周係数の静的設定
DYN_FBDIV_SEL	"true", "false"	"false"	FBDIV分周係数の静的制御パラメ

パラメータ名	値の範囲	デフォルト値	説明
			ータまたは動的制御信号の選択 false: 静的制御、つまり FBDIV_SELを選択 true: 動的制御、つまりFBDSELを選択
ODIV_SEL	2,4,8,16,32,48, 64,80,96,112,1 28	8	ODIV分周係数の静的設定
DYN_ODIV_SEL	"true", "false"	"false"	ODIV分周係数の静的制御パラメータまたは動的制御信号の選択 false: 静的制御、つまりODIV_SELを選択 true: 動的制御、つまりODSELを選択
PSDA_SEL	"0000"~"1111"	"0000"	位相の静的調整
DUTYDA_SEL	"0010"~"1110"	"1000"	デューティサイクルの静的調整
DYN_DA_EN	"true", "false"	"false"	位相およびデューティサイクル調整を制御するために動的信号を選択 false: スタティック制御 true: 動的制御
CLKOUT_FT_DIR	1'b1	1'b1	CLKOUTの微調整方向の設定 1'b1: 減らす
CLKOUT_DLY_STEP	0,1,2,4	0	CLKOUTの微調整係数の設定 CLKOUT_DLY_STEP*delay(delay=50ps)
CLKOUTP_FT_DIR	1'b1	1'b1	CLKOUTPの微調整方向の設定 1'b1: 減らす
CLKOUTP_DLY_STEP	0,1,2	0	CLKOUTPの微調整係数の設定 CLKOUTP_DLY_STEP*delay(delay=50ps)
DYN_SDIV_SEL	2~128 (偶数のみ)	2	SDIV分周係数の静的設定
CLKFB_SEL	"internal", "external"	"internal"	CLKFBソースの選択 internal: 内部CLKOUTからのフィードバック external: 外部信号からのフィードバック
CLKOUTD_SRC	"CLKOUT", "CLKOUTP"	"CLKOUT"	CLKOUTDソースの選択
CLKOUTD3_SRC	"CLKOUT", "CLKOUTP"	"CLKOUT"	CLKOUTD3ソースの選択
CLKOUT_BYPASS	"true", "false"	"false"	rPLLをバイパス。CLKOUTDはCLKINから直接取得 true: rPLLをバイパス。CLKINはCLKOUTに直接作用 false: ノーマルモード
CLKOUTP_BYPASS	"true", "false"	"false"	rPLLをバイパス。CLKOUTPはCLKINから直接取得 true: rPLLをバイパス。CLKINは

パラメータ名	値の範囲	デフォルト値	説明
			CLKOUTPに直接作用 false: ノーマルモード
CLKOUTD_BYPASS	"true", "false"	"false"	rPLLをバイパス。CLKOUTDはCLKINから直接取得 true: rPLLをバイパス。CLKINはCLKOUTDに直接作用 false: ノーマルモード
DEVICE	"GW1N-1", "GW1NR-1", "GW1N-1S", "GW1NZ-1", "GW1NZ-1C", "GW1N-4", "GW1N-4B", "GW1N-4D", "GW1NR-4", "GW1NR-4B", "GW1NR-4D", "GW1NRF-4B", "GW1N-9", "GW1N-9C", "GW1NR-9", "GW1NR-9C", "GW2A-18", "GW2AR-18", "GW2A-55", "GW2A-55C"	"GW1N-4"	デバイスの選択

表 5-4 IDSEL ポートのパラメータ参照テーブル

IDSEL[5:0]	IDIVの静的パラメータ値	IDIVの実際の値
111111	0	1
111110	1	2
111101	2	3
111100	3	4
111011	4	5
111010	5	6
111001	6	7
111000	7	8
110111	8	9
.....	.....	.....
000000	63	64

表 5-5 FBDSEL ポートのパラメータ参照テーブル

FBDSEL [5:0]	FBDIVの静的パラメータ値	FBDIVの実際の値
111111	0	1

FBDSEL [5:0]	FBDIVの静的パラメータ値	FBDIVの実際の値
111110	1	2
111101	2	3
111100	3	4
111011	4	5
111010	5	6
111001	6	7
111000	7	8
110111	8	9
.....	.....	.....
000000	63	64

表 5-6 ODSEL ポートのパラメータ参照テーブル

ODSEL [5:0]	ODIVのパラメータ値	ODIVの実際の値
111111	2	2
111110	4	4
111100	8	8
111000	16	16
110000	32	32
101000	48	48
100000	64	64
011000	80	80
010000	96	96
001000	112	112
000000	128	128

表 5-7 rPLL 位相調整の参照テーブル

パラメータPSDA_SELまたはポートPSDA設定	位相の調整
0000	0°
0001	22.5°
0010	45°
0011	67.5°
0100	90°
0101	112.5°
0110	135°
0111	157.5°
1000	180°

パラメータPSDA_SELまたはポート PSDA設定	位相の調整
1001	202.5°
1010	225°
1011	247.5°
1100	270°
1101	292.5°
1110	315°
1111	337.5°

表 5-8 rPLL デューティサイクル調整の参照テーブル

パラメータDUTYDA_SEL設定	デューティサイクルの設定値 (/16)
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14

デューティサイクル動的調整の場合、位相シフト設定を参照する必要があります。例えば、位相シフトが"0" (0000) の場合、50%デューティサイクルの DUTYDA\_SEL の値は"8" (1000) です。位相シフトが"180°"の場合、50%デューティサイクルの DUTYDA\_SEL の値は"0" (0000) になります。

動的デューティサイクルの計算：

- DUTYDA [3:0] > PSDA [3:0] の場合、DutyCycle =  $1/16 \times (\text{DUTYDA [3:0]} - \text{PSDA [3:0]})$ 。
- DUTYDA [3:0] < PSDA [3:0] の場合、DutyCycle =  $1/16 \times (16 + \text{DUTYDA [3:0]} - \text{PSDA [3:0]})$ 。

注記：

DutyCycle = 0/16、1/16、および 15/16 はサポートされていません。

ポート FDLY[3 : 0]を介して、出力クロック CLKOUTP の遅延を動的に

制御できます。1 ステップごとに **0.125ns** 増加します。ヒステリシス（クロック信号 **CLKOUTP** が入力クロックより遅い）とリード（クロック信号 **CLKOUTP** が入力クロックより早い）は位相オフセットに合わせて実装されます。

表 5-9 rPLL 遅延調整の参照テーブル

ポートFDLY [3:0] (GW1N-1/GW1N-1S)	ポートFDLY [3:0] (その他のデバイス)	遅延ステップ数
0000	1111	0
0001	1110	1
0010	1101	2
0100	1011	4
1000	0111	8

### プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。

#### Verilog でのインスタンス化：

```
rPLL rpll_inst(
    .CLKOUT(clkout),
    .LOCK(lock),
    .CLKOUTP(clkoutp),
    .CLKOUTD(clkoutd),
    .CLKOUTD3(clkoutd3),
    .RESET(reset),
    .RESET_P(reset_p),
    .CLKIN(clkin),
    .CLKFB(clkfb),
    .FBDSEL(fbdsel),
    .IDSEL(idsel),
    .ODSEL(odsel),
    .PSDA(psda),
    .DUTYDA(dutyda),
    .FDLY(fdly)
);
defparam rpll_inst.FCLKIN = "50";
defparam rpll_inst.DYN_IDIV_SEL = "false";
defparam rpll_inst.IDIV_SEL = 0;
```

```

defparam rpll_inst.DYN_FBDIV_SEL = "false";
defparam rpll_inst.FBDIV_SEL = 1;
defparam rpll_inst.ODIV_SEL = 8;
defparam rpll_inst.PSDA_SEL = "0100";
defparam rpll_inst.DYN_DA_EN = "false";
defparam rpll_inst.DUTYDA_SEL = "1000";
defparam rpll_inst.CLKOUT_FT_DIR = 1'b1;
defparam rpll_inst.CLKOUTP_FT_DIR = 1'b1;
defparam rpll_inst.CLKOUT_DLY_STEP = 0;
defparam rpll_inst.CLKOUTP_DLY_STEP = 0;
defparam rpll_inst.CLKFB_SEL = "external";
defparam rpll_inst.CLKOUT_BYPASS = "false";
defparam rpll_inst.CLKOUTP_BYPASS = "false";
defparam rpll_inst.CLKOUTD_BYPASS = "false";
defparam rpll_inst.DYN_SDIV_SEL = 2;
defparam rpll_inst.CLKOUTD_SRC = "CLKOUT";
defparam rpll_inst.CLKOUTD3_SRC = "CLKOUT";
defparam rpll_inst.DEVICE = "GW1N-4";

```

**VHDL** でのインスタンス化 :

```

COMPONENT rPLL
    GENERIC(
        FCLKIN:STRING:= "100.0";
        DEVICE:STRING:= "GW1N-4";
        DYN_IDIV_SEL:STRING:="false";
        IDIV_SEL:integer:=0;
        DYN_FBDIV_SEL:STRING:="false";
        FBDIV_SEL:integer:=0;
        DYN_ODIV_SEL:STRING:="false";
        ODIV_SEL:integer:=8;
        PSDA_SEL:STRING:="0000";
        DYN_DA_EN:STRING:="false";
        DUTYDA_SEL:STRING:="1000";
        CLKOUT_FT_DIR:bit:='1';
        CLKOUTP_FT_DIR:bit:='1';
        CLKOUT_DLY_STEP:integer:=0;

```

```

        CLKOUTP_DLY_STEP:integer:=0;
        CLKOUTD3_SRC:STRING:="CLKOUT";
        CLKFB_SEL : STRING:="internal";
        CLKOUT_BYPASS:STRING:="false";
        CLKOUTP_BYPASS:STRING:="false";
        CLKOUTD_BYPASS:STRING:="false";
        CLKOUTD_SRC:STRING:="CLKOUT";
        DYN_SDIV_SEL:integer:=2
    );
    PORT(
        CLKIN:IN std_logic;
        CLKFB:IN std_logic;
        IDSEL:IN std_logic_vector(5 downto 0);
        FBDSEL:IN std_logic_vector(5 downto 0);
        ODSEL:IN std_logic_vector(5 downto 0);
        RESET:IN std_logic;
        RESET_P:IN std_logic;
        PSDA,FDLY:IN std_logic_vector(3 downto 0);
        DUTYDA:IN std_logic_vector(3 downto 0);
        LOCK:OUT std_logic;
        CLKOUT:OUT std_logic;
        CLKOUTD:OUT std_logic;
        CLKOUTP:OUT std_logic;
        CLKOUTD3:OUT std_logic
    );
END COMPONENT;
uut:rPLL
    GENERIC MAP(
        FCLKIN =>"100.0",
        DEVICE =>"GW2A-18",
        DYN_IDIV_SEL=>"false",
        IDIV_SEL=>0,
        DYN_FBDIV_SEL=>"false",
        FBDIV_SEL=>0,
        DYN_ODIV_SEL=>"false",

```



```

        ODIV_SEL=>8,
        PSDA_SEL=>"0000",
        DYN_DA_EN=>"false",
        DUTYDA_SEL=>"1000",
        CLKOUT_FT_DIR=>'1',
        CLKOUTP_FT_DIR=>'1',
        CLKOUT_DLY_STEP=>0,
        CLKOUTP_DLY_STEP=>0,
        CLKOUTD3_SRC=>"CLKOUT",
        CLKFB_SEL=>"internal",
        CLKOUT_BYPASS=>"false",
        CLKOUTP_BYPASS=>"false",
        CLKOUTD_BYPASS=>"false",
        CLKOUTD_SRC=>"CLKOUT",
        DYN_SDIV_SEL=>2
    )
    PORT MAP(
        CLKIN=>clk_in,
        CLKFB=>clkfb,
        IDSEL=>idsel,
        FBDSEL=>fbdsel,
        ODSEL=>odsel,
        RESET=>reset,
        RESET_P=>reset_p,
        PSDA=>psda,
        FDLY=>fdly,
        DUTYDA=>dutyda,
        LOCK=>lock,
        CLKOUT=>clkout,
        CLKOUTD=>clkoutd,
        CLKOUTP=>clkoutp,
        CLKOUTD3=>clkoutd3
    );

```

### 5.1.2 IP の呼び出し

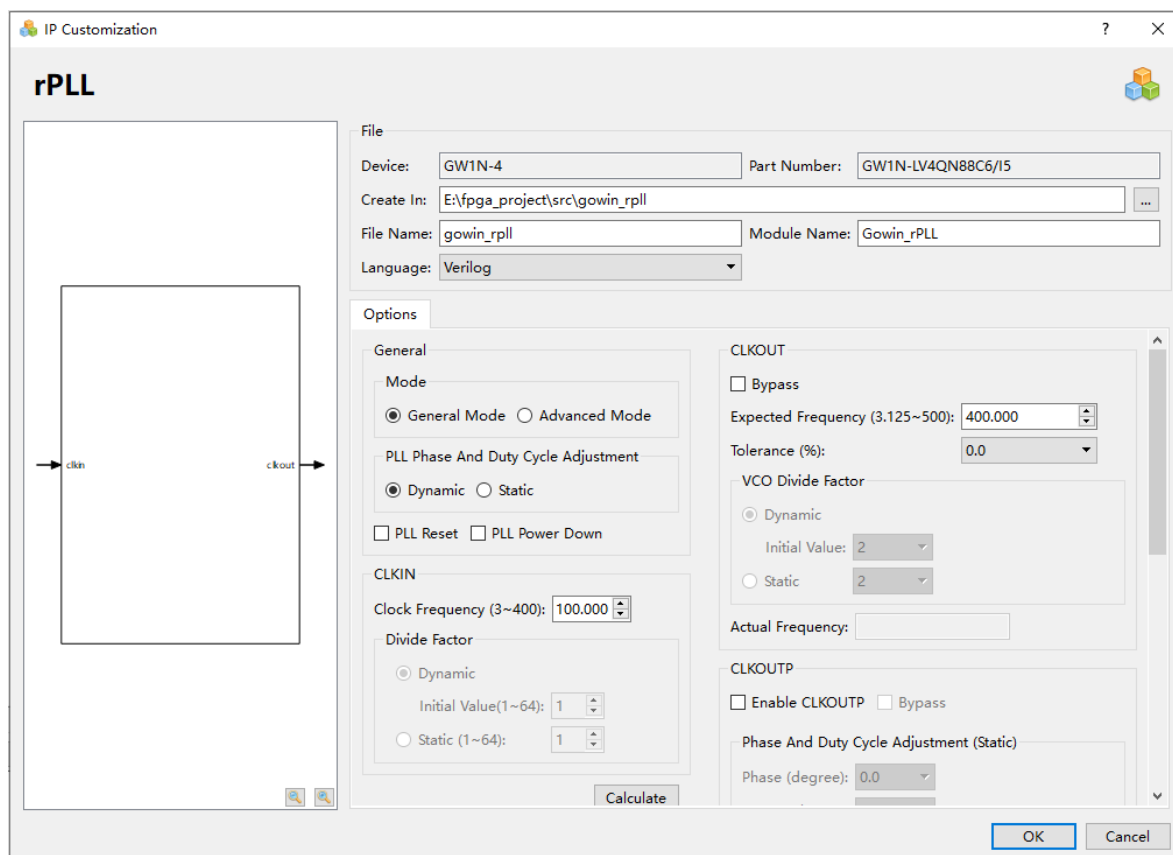
IP Core Generator のインターフェースで"rPLL"をクリックすると、右

側に rPLL の概要が表示されます。

## IP の構成

IP Core Generator インターフェースで rPLL をダブルクリックすると、rPLL の"IP Customization"ウィンドウがポップアップします。このウィンドウには File 構成タブ、Options 構成タブ、およびポート図があります (図 5-4)。

図 5-4 rPLL IP の構成ウィンドウ



### 1. File 構成タブ

File 構成タブは、IP ファイルの構成に使用されます。rPLL の File 構成タブの使用は DQCE モジュールと同様です。DQCE の File 構成タブを参照してください。

### 2. Options 構成タブ

Options 構成タブは IP のカスタマイズに使用されます(図 5-4)。

- **General** : 一般モードとアドバンスモードの構成、出力位相とデューティサイクルの調整の動的モード/静的モードの構成、PLL Reset のイネーブルの構成を実行します。
  - **"Mode"** : IP Core 構成のモードを設定し、一般モード"General Mode"とアドバンスモード"Advanced Mode"をサポートします。一般モードでは、入力クロック周波数と出力クロック周波数を

入力すると、ソフトウェアが自動的に周波数分割係数を計算します。アドバンスモードはアドバンスユーザーに適しており、アドバンスモードでは、入力周波数と周波数分割係数を入力して期待される出力を得ることができます。

- "PLL Phase And Duty Cycle Adjustment" : 出力デューティサイクルおよび位相調整モードを構成し、動的調整"Dynamic"および静的調整"Static"をサポートします。
- "PLL Reset" : チェックすると、rPLL のリセットが有効になります。
- "PLL Power Down" : チェックすると、reset\_p ポートを介して rPLL をパワーダウンモードにします。
- CLKIN : rPLL 入力クロックの周波数、分周パラメータなどを構成します。
  - "Clock Frequency (周波数範囲)" : 入力クロックの周波数を構成し、範囲は device により異なります。
  - "Divide Factor" : アドバンスモードで分周パラメータを構成し、動的モード"Dynamic"と静的モード"Static"をサポートします。静的モードでは分周パラメータの具体的な数値を構成でき、範囲は 1~64 です。CLKOUT の出力周波数が device の推奨動作要件の範囲外にある場合、"Calculate"または"OK"をクリックするとエラーメッセージのウィンドウがポップアップします。CLKIN/IDIV の出力周波数が device の推奨動作要件の範囲外にある場合、"Calculate"または"OK"をクリックするとエラーメッセージのウィンドウがポップアップします。
- CLKFB : rPLL フィードバッククロックのソースと周波数通倍パラメータを構成します。
  - フィードバッククロックのソース (Source) として Internal と External を選択できます。
  - "Divide Factor" : アドバンスモードで周波数通倍パラメータを構成し、動的モード"Dynamic"と静的モード"Static"をサポートします。静的モードでは周波数通倍パラメータの具体的な数値を構成でき、範囲は 1~64 です。構成が不適切な場合、"Calculate"ボタンまたは"OK"ボタンをクリックするとエラーメッセージのウィンドウがポップアップします。
- Enable LOCK : チェックすると、LOCK ポートがイネーブルされます。
- CLKOUT : rPLL 出力クロックの目的の周波数、VCO パラメータ、出力クロックサイクルの微調整パラメータを構成します。
  - "Bypass" : チェックすると、バイパス機能がイネーブルされます。

- **"Expected Frequency (周波数範囲)"** : 一般モードで目的の出力クロック CLKOUT の周波数を構成し、範囲は **device** により異なります。
- **"Tolerance (%)"** は CLKOUT の目的の周波数と算出した実際周波数の許容誤差を構成します。
- **"VCO Divide Factor"** : アドバンスモードで VCO パラメータを構成し、動的モード **"Dynamic"** と静的モード **"Static"** をサポートします。静的モードでは分周パラメータの具体的な数値を構成でき、範囲は **2/4/8/16/32/48/64/80/96/112/128** です。構成が不適切な場合、**"Calculate"** または **"OK"** をクリックするとエラーメッセージのウィンドウがポップアップします。
- **"Actual Frequency"** : 計算で得られた CLKOUT の実際出力周波数を表示します。
- **CLKOUTP** : 位相シフトクロックの位相とデューティサイクルの調整パラメータを構成し、位相シフトクロックのリセットを有効/無効にします。
  - **"Enable CLKOUTP"** : チェックすると、CLKOUTP がイネーブルされます。
  - **"Bypass"** : チェックすると、バイパス機能がイネーブルされます。
  - **"Phase And Duty Cycle Adjustment (Static)"** : 静的モードで位相 (Phase (degree)) およびデューティサイクル (Duty Cycle) を構成できます。
- **CLKOUTD** : 分周クロック出力のクロックソース、目的の分周クロック出力周波数、および分周クロックの出力パラメータを構成し、分周クロック出力のリセットを有効/無効にします。
  - **"Enable CLKOUTD"** : チェックすると、CLKOUTD がイネーブルされます。
  - **"Bypass"** : チェックすると、バイパス機能がイネーブルされます。
  - **"Source"** : 分周クロック出力のクロックソースを構成し、CLKOUT と CLKOUTP を選択できます。
  - **"Expected Frequency (周波数範囲)"** : 一般モードで目的の分周クロック出力の周波数を構成し、範囲は **device** により異なります。
  - **"Tolerance (%)"** : 分周クロック出力の目的の周波数と算出した実際周波数の許容誤差を構成します。
  - **"Divide Factor (2~128)"** : アドバンスモードで分周クロック出力の分周パラメータを構成します。範囲は **2~128** の偶数です。

奇数に設定されている場合、"OK"をクリックするとエラーが表示されます。

- "Actual Frequency" : 計算で得られた分周クロックの実際出力周波数を表示します。
- CLKOUTD3:3分周クロック出力のクロックソースを構成します。
  - "Enable CLKOUTP" : チェックすると、CLKOUTD3 がイネーブルされます。
  - "Source" : 3 分周クロック出力のクロックソースを構成し、CLKOUT と CLKOUTP を選択できます。
- Calculate : 現在の構成が適切か計算します。
  - 一般モード"General Mode"で入力・出力周波数の計算に従って分周パラメータ、周波数逡倍パラメータ、VCO パラメータを構成します。計算された実際の周波数が目的の周波数と等しくない場合、"Calculate"ボタンをクリックすると"error"ウィンドウがポップアップし、不当な位置を赤でマークします。
  - アドバンスモード"Advanced Mode"で、構成された静的分周パラメータ、周波数逡倍パラメータ、VCO パラメータが適切か計算します。適切でない場合は、"Calculate"をクリックすると"error"メッセージが表示されます。正しい場合は、"Calculate"をクリックすると、構成が成功したことを示す"info"ウィンドウが表示されます。

### 3. ポート図

ポート図は、現在の IP Core の構成結果を表示し、入力・出力ポートの数は Options 構成に従ってリアルタイムで更新されます (図 5-4)。

#### 生成されるファイル

IP の構成が完了したら、構成ファイルの"File Name"によって命名された 3 つのファイルが生成されます :

- "gowin\_rpll.v"は完全な verilog モジュールです。
- gowin\_rpll\_tmp.v は IP のテンプレートファイルです。
- gowin\_rpll.ipc は IP の構成ファイルです。

注記 :

VHDL が設計の言語として選択されている場合、生成される最初の 2 つのファイル名のサフィックスは.vhd になります。

## 5.2 PLLVR

### 5.2.1 プリミティブの紹介

GOWIN FPGA の PLLVR (Phase\_Locked Loop with regulator、電力調整機能付き位相同期回路) は、外部からのリファレンスクロック信号によ

り、ループ内部のオシレータ信号の周波数と位相を制御します。

### サポートされるデバイス

表 5-10 PLLVR 対応デバイス

ファミリー	シリーズ	デバイス
LittleBee®	GW1NS	GW1NS-4, GW1NS-4C, GW1NSR-4, GW1NSR-4C, GW1NSER-4C

### 機能の説明

PLLVR は、電力調整機能を備えた PLL です。入力クロックに基づいてクロック位相調整、デューティサイクル調整、および周波数調整（逡倍と分周）を実行し、さまざまな位相と周波数の出力クロックを生成できます。

PLLVR の機能は下表のとおりです。

PLLVR は入力クロック **CLKIN** に周波数調整（逡倍及び分周）を行うことができ、計算式は以下の通りです：

$$f_{CLKOUT} = (f_{CLKIN} * FBDIV) / IDIV$$

$$f_{VCO} = f_{CLKOUT} * ODIV$$

$$f_{CLKOUTD} = f_{CLKOUT} / SDIV$$

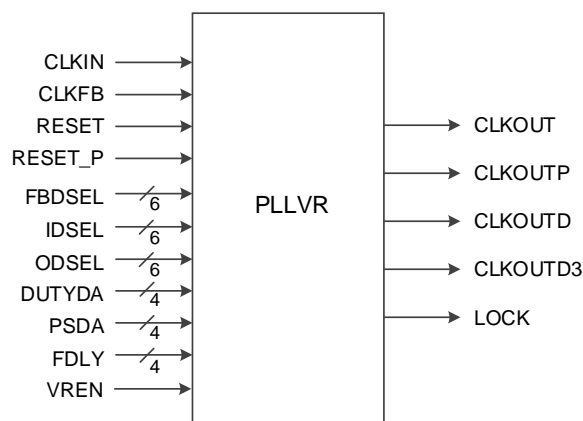
$$f_{PFD} = f_{CLKIN} / IDIV = f_{CLKOUT} / FBDIV$$

注記：

- $f_{CLKIN}$  は入力クロック **CLKIN** の周波数、 $f_{CLKOUT}$  は **CLKOUT** および **CLKOUTP** の周波数、 $f_{CLKOUTD}$  は **CLKOUTD** の周波数、 $f_{PFD}$  は位相検出器周波数です。
- **IDIV**、**FBDIV**、**ODIV**、**SDIV** は、各分周器の分周係数です。分周係数を調整することにより、所望の周波数のクロック信号を生成することができます。
- PLLVR の周波数の範囲については、[FPGA 製品データシート](#)を参照してください。

### ポート図

図 5-5 PLLVR のポート図



## ポートの説明

表 5-11 PLLVR のポートの説明

ポート名	I/O	説明
CLKIN	入力	リファレンスクロック入力
CLKFB	入力	フィードバッククロック入力
RESET	入力	PLLVR非同期リセット入力、アクティブHigh
RESET_P	入力	PLLVRパワーダウン (Power Down) 入力、アクティブHigh。非bypassモードでは、RESET_PがHighの場合、CLKOUT/CLKOUTP/CLKOUTD/CLKOUTD3は0です。
FBDSEL[5:0]	入力	FBDIVの動的制御値、範囲は0~63、実際の値は64-FBDSEL
IDSEL[5:0]	入力	IDIVの動的制御値、範囲は0~63、実際の値は64-IDSEL
ODSEL[5:0]	入力	ODIVの動的制御値、 2,4,8,16,32,48,64,80,96,112,128
DUTYDA[3:0]	入力	デューティサイクルの動的調整信号
PSDA[3:0]	入力	位相の動的調整信号
FDLY[3:0]	入力	遅延の動的微調整信号
VREN	入力	PLLVR電源調整イネーブル信号、アクティブHigh
CLKOUT	出力	PLLVRクロック出力
LOCK	出力	PLLVRロック指示。1はロック、0はロック解除を示す
CLKOUTP	出力	位相とデューティサイクル調整ありのPLLVRクロック出力
CLKOUTD	出力	SDIVを経たPLLVRクロック出力信号、SDIVを経たCLKOUTまたはCLKOUTPの出力信号
CLKOUTD3	出力	DIV3を経たPLLVRクロック出力信号、DIV3を経たCLKOUTまたはCLKOUTPの出力信号

注記：

- CLKOUTD3 は、CLKOUT または CLKOUTP を 3 分周した後の出力クロック信号であり、CLKOUT または CLKOUTP とのタイミング関係については 5.1 rPLL を参照してください。

## パラメータの説明

表 5-12 PLLVR のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
FCLKIN	3~500	100	リファレンスクロックの周波数
IDIV_SEL	0~63	0	IDIV分周係数の静的設定
DYN_IDIV_SEL	"true", "false"	"false"	IDIV分周係数の静的制御パラメータまたは動的制御信号の選択 false: 静的制御、つまりIDIV_SELを選択 true: 動的制御、つまりIDSELを選択
FBDIV_SEL	0~63	0	FBDIV分周係数の静的設定

パラメータ名	値の範囲	デフォルト値	説明
DYN_FBDIV_SEL	"true", "false"	"false"	FBDIV分周係数の静的制御パラメータまたは動的制御信号の選択 false: 静的制御、つまりFBDIV_SELを選択 true: 動的制御、つまりFBDSELを選択
ODIV_SEL	2,4,8,16,32,48,64,80,96,112,128	8	ODIV分周係数の静的設定
DYN_ODIV_SEL	"true", "false"	"false"	ODIV分周係数の静的制御パラメータまたは動的制御信号の選択 false: 静的制御、つまりODIV_SELを選択 true: 動的制御、つまりODSELを選択
PSDA_SEL	"0000" ~ "1111"	"0000"	位相の静的調整
DUTYDA_SEL	"0010" ~ "1110"	"1000"	デューティサイクルの静的調整
DYN_DA_EN	"true", "false"	"false"	位相およびデューティサイクル調整を制御するために動的信号を選択 false: スタティック制御 true: 動的制御
CLKOUT_FT_DIR	1'b1	1'b1	CLKOUTの微調整方向の設定 1'b1: 減らす
CLKOUT_DLY_STEP	0,1,2,4	0	CLKOUTの微調整係数の設定 $\text{CLKOUT\_DLY\_STEP} \times \text{delay} (\text{delay}=50\text{ps})$
CLKOUTP_FT_DIR	1'b1	1'b1	CLKOUTPの微調整方向の設定 1'b1: 減らす
CLKOUTP_DLY_STEP	0,1,2	0	CLKOUTPの微調整係数の設定 $\text{CLKOUTP\_DLY\_STEP} \times \text{delay} (\text{delay}=50\text{ps})$
DYN_SDIV_SEL	2~128 (偶数のみ)	2	SDIV分周係数の静的設定
CLKFB_SEL	"internal", "external"	"internal"	CLKFBソースの選択 internal: 内部CLKOUTからのフィードバック external: 外部信号からのフィードバック
CLKOUTD_SRC	"CLKOUT", "CLKOUTP"	"CLKOUT"	CLKOUTDソースの選択
CLKOUTD3_SRC	"CLKOUT", "CLKOUTP"	"CLKOUT"	CLKOUTD3ソースの選択
CLKOUT_BYPASS	"true", "false"	"false"	PLLVRをバイパス。CLKOUTはCLKINから直接取得 true: PLLVRをバイパス。CLKINはCLKOUTに直接作用 false: ノーマルモード
CLKOUTP_BYPASS	"true", "false"	"false"	PLLVRをバイパス。CLKOUTPはCLKINから直接取得 true: PLLVRをバイパス。CLKINはCLKOUTPに直接作用 false: ノーマルモード



パラメータ名	値の範囲	デフォルト値	説明
CLKOUTD_BYPASS	"true", "false"	"false"	PLLVRをバイパス。CLKOUTDはCLKINから直接取得 true: PLLVRをバイパス。CLKINはCLKOUTDに直接作用 false: ノーマルモード
DEVICE	"GW1NS-4", "GW1NS-4C", "GW1NSR-4", "GW1NSR-4C", "GW1NSER-4C"	"GW1NS-4"	デバイスの選択

注記：

IDSEL、FBDESL、ODSEL ポートのパラメータ参照テーブル、位相調整の参照テーブル、およびデューティサイクル調整の参照テーブルは rPLL と同様です。

#### プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。

#### Verilog でのインスタンス化：

```

PLLVR pllvr_inst(
    .CLKOUT(clkout),
    .LOCK(lock),
    .CLKOUTP(clkoutp),
    .CLKOUTD(clkoutd),
    .CLKOUTD3(clkoutd3),
    .VREN(vren),
    .RESET(reset),
    .RESET_P(reset_p),
    .CLKIN(clkin),
    .CLKFB(clkfb),
    .FBDESEL(fbdsel),
    .IDSEL(idsel),
    .ODSEL(odsel),
    .PSDA(psda),
    .DUTYDA(dutyda),

```

```

        .FDLY(fdly)
    );
    defparam pllvr_inst.FCLKIN = "50";
    defparam pllvr_inst.DYN_IDIV_SEL = "false";
    defparam pllvr_inst.IDIV_SEL = 0;
    defparam pllvr_inst.DYN_FBDIV_SEL = "false";
    defparam pllvr_inst.FBDIV_SEL = 1;
    defparam pllvr_inst.ODIV_SEL = 8;
    defparam pllvr_inst.PSDA_SEL = "0100";
    defparam pllvr_inst.DYN_DA_EN = "false";
    defparam pllvr_inst.DUTYDA_SEL = "1000";
    defparam pllvr_inst.CLKOUT_FT_DIR = 1'b1;
    defparam pllvr_inst.CLKOUTP_FT_DIR = 1'b1;
    defparam pllvr_inst.CLKOUT_DLY_STEP = 0;
    defparam pllvr_inst.CLKOUTP_DLY_STEP = 0;
    defparam pllvr_inst.CLKFB_SEL = "external";
    defparam pllvr_inst.CLKOUT_BYPASS = "false";
    defparam pllvr_inst.CLKOUTP_BYPASS = "false";
    defparam pllvr_inst.CLKOUTD_BYPASS = "false";
    defparam pllvr_inst.DYN_SDIV_SEL = 2;
    defparam pllvr_inst.CLKOUTD_SRC = "CLKOUT";
    defparam pllvr_inst.CLKOUTD3_SRC = "CLKOUT";
    defparam pllvr_inst.DEVICE = "GW1NS-4";

```

**VHDL** でのインスタンス化 :

```

COMPONENT PLLVR
    GENERIC(
        FCLKIN:STRING:= "100.0";
        DEVICE:STRING:= "GW1NS-4";
        DYN_IDIV_SEL:STRING:="false";
        IDIV_SEL:integer:=0;
        DYN_FBDIV_SEL:STRING:="false";
        FBDIV_SEL:integer:=0;
        DYN_ODIV_SEL:STRING:="false";
        ODIV_SEL:integer:=8;
        PSDA_SEL:STRING:="0000";

```

```

        DYN_DA_EN:STRING:="false";
        DUTYDA_SEL:STRING:="1000";
        CLKOUT_FT_DIR:bit:='1';
        CLKOUTP_FT_DIR:bit:='1';
        CLKOUT_DLY_STEP:integer:=0;
        CLKOUTP_DLY_STEP:integer:=0;
        CLKOUTD3_SRC:STRING:="CLKOUT";
        CLKFB_SEL : STRING:="internal";
        CLKOUT_BYPASS:STRING:="false";
        CLKOUTP_BYPASS:STRING:="false";
        CLKOUTD_BYPASS:STRING:="false";
        CLKOUTD_SRC:STRING:="CLKOUT";
        DYN_SDIV_SEL:integer:=2
    );
    PORT(
        CLKIN:IN std_logic;
        CLKFB:IN std_logic;
        IDSEL:IN std_logic_vector(5 downto 0);
        FBDSEL:IN std_logic_vector(5 downto 0);
        ODSEL:IN std_logic_vector(5 downto 0);
        VREN:IN std_logic;
        RESET:IN std_logic;
        RESET_P:IN std_logic;
        PSDA,FDLY:IN std_logic_vector(3 downto 0);
        DUTYDA:IN std_logic_vector(3 downto 0);
        LOCK:OUT std_logic;
        CLKOUT:OUT std_logic;
        CLKOUTD:OUT std_logic;
        CLKOUTP:OUT std_logic;
        CLKOUTD3:OUT std_logic
    );
END COMPONENT;
 uut:PLLVR
    GENERIC MAP(
        FCLKIN =>"100.0",

```

```
    DEVICE =>"GW1NS-4",
    DYN_IDIV_SEL=>"false",
    IDIV_SEL=>0,
    DYN_FBDIV_SEL=>"false",
    FBDIV_SEL=>0,
    DYN_ODIV_SEL=>"false",
    ODIV_SEL=>8,
    PSDA_SEL=>"0000",
    DYN_DA_EN=>"false",
    DUTYDA_SEL=>"1000",
    CLKOUT_FT_DIR=>'1',
    CLKOUTP_FT_DIR=>'1',
    CLKOUT_DLY_STEP=>0,
    CLKOUTP_DLY_STEP=>0,
    CLKOUTD3_SRC=>"CLKOUT",
    CLKFB_SEL=>"internal",
    CLKOUT_BYPASS=>"false",
    CLKOUTP_BYPASS=>"false",
    CLKOUTD_BYPASS=>"false",
    CLKOUTD_SRC=>"CLKOUT",
    DYN_SDIV_SEL=>2
)
PORT MAP(
    CLKIN=>clk_in,
    CLKFB=>clkfb,
    IDSEL=>idsel,
    FBDSSEL=>fbdsel,
    ODSEL=>odsel,
    VREN=>vren,
    RESET=>reset,
    RESET_P=>reset_p,
    PSDA=>psda,
    FDLY=>fdly,
    DUTYDA=>dutyda,
    LOCK=>lock,
```

```
CLKOUT=>clkout,
CLKOUTD=>clkoutd,
CLKOUTP=>clkoutp,
CLKOUTD3=>clkoutd3
```

```
);
```

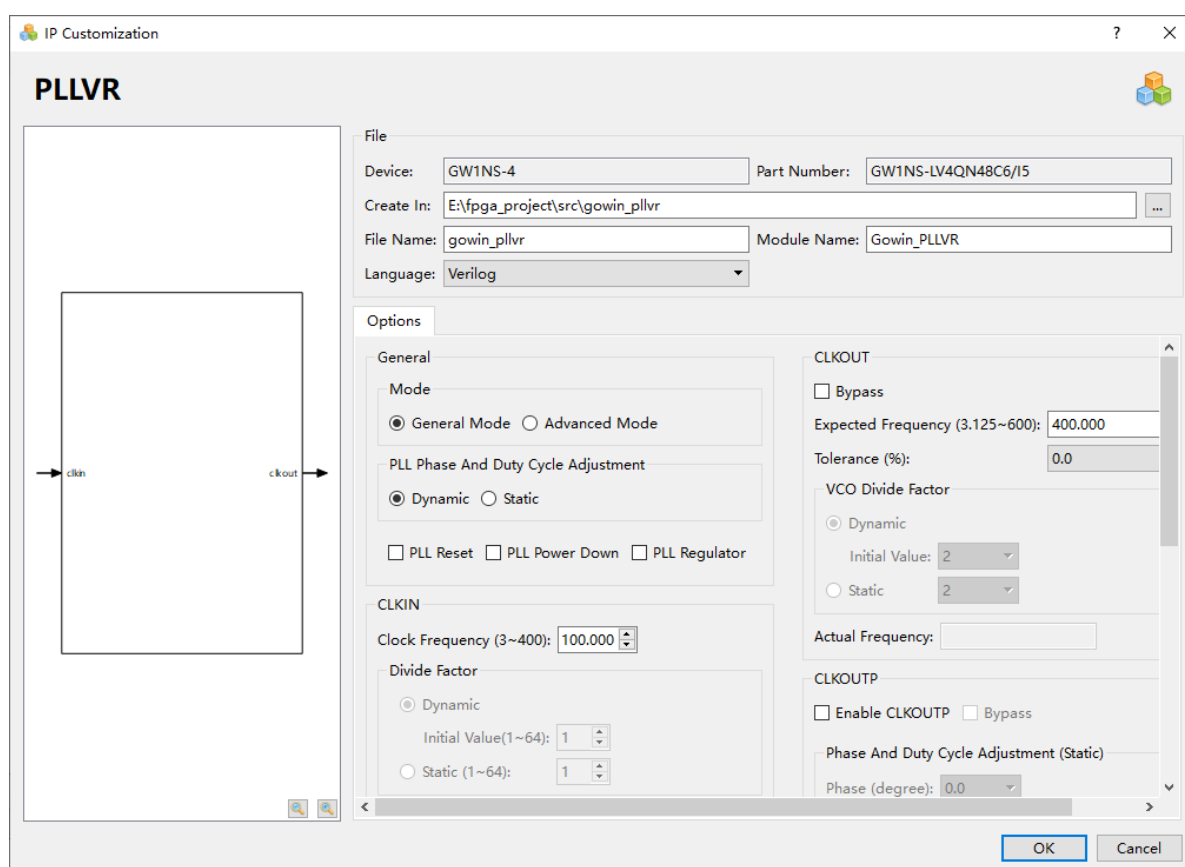
### 5.2.2 IP の呼び出し

IP Core Generator のインターフェースで"PLLVR"をクリックすると、右側に PLLVR の概要が表示されます。

#### IP の構成

IP Core Generator インターフェースで"PLLVR"をダブルクリックすると、PLLVR の"IP Customization"ウィンドウがポップアップします。このウィンドウには File 構成タブ、Options 構成タブ、ポート図、および"Help"ボタンがあります (図 5-6)。

図 5-6 PLLVR IP の構成ウィンドウ



#### 1. File 構成タブ

File 構成タブは、IP ファイルの構成に使用されます。PLLVR の File 構成タブの使用は DQCE モジュールと同様です。DQCE の File 構成タブを参照してください。

## 2. Options 構成タブ

Options 構成タブは IP のカスタマイズに使用されます(図 5-6)。PLLVR の Options 構成タブの使用は rPLL モジュールと同様です。rPLL の Options 構成タブを参照してください。PLL Regulator オプションが追加されました。

## 3. ポート図

ポート図は、現在の IP Core の構成結果を表示し、入力・出力ポートのビット幅は Options 構成に従ってリアルタイムで更新されます(図 5-6)。

### 生成されるファイル

IP の構成が完了したら、構成ファイルの"File Name"によって命名された 3 つのファイルが生成されます：

- "gowin\_pllvr.v"は完全な verilog モジュールです。
- gowin\_pllvr\_tmp.v は IP のテンプレートファイルです。
- gowin\_pllvr.ipc は IP の構成ファイルです。

注記：

VHDL が設計の言語として選択されている場合、生成される最初の 2 つのファイル名のサフィックスは.vhd になります。

## 5.3 PLLO

### 5.3.1 プリミティブの紹介

Gowin は、4 チャネルクロック出力をサポートし、特定の入力クロックに基づいて周波数、位相、およびデューティサイクルを調整できる位相ロックループ PLLO を提供します。

#### サポートされるデバイス

表 5-13 PLLO 対応デバイス

ファミリー	シリーズ	デバイス
LittleBee®	GW1N	GW1N-2, GW1N-1P5, GW1N-2B, GW1N-1P5B
	GW1NR	GW1NR-2, GW1NR-2B
Arora	GW2AN	GW2AN-18X, GW2AN-9X

#### 機能の説明

PLLO は、4 チャネルクロック出力をサポートし、入力クロックに基づいてクロック位相調整、デューティサイクル調整、および周波数調整（逡倍と分周）を実行し、さまざまな位相と周波数のクロック出力を生成できます。正しいクロック出力を得るには、[FPGA 製品データシート](#)に記載されている周波数範囲に従って入力クロック周波数を設定する必要があります。

PLLO は入力クロック CLKIN に周波数調整（逡倍及び分周）を行うこ

とができ、計算式は以下の通りです：

$$f_{CLKOUTA} = (f_{CLKIN} * FBDIV) / IDIV$$

$$f_{VCO} = f_{CLKOUTA} * ODIVA$$

$$f_{CLKOUTX} = f_{IN\_ODIVX} / ODIVX$$

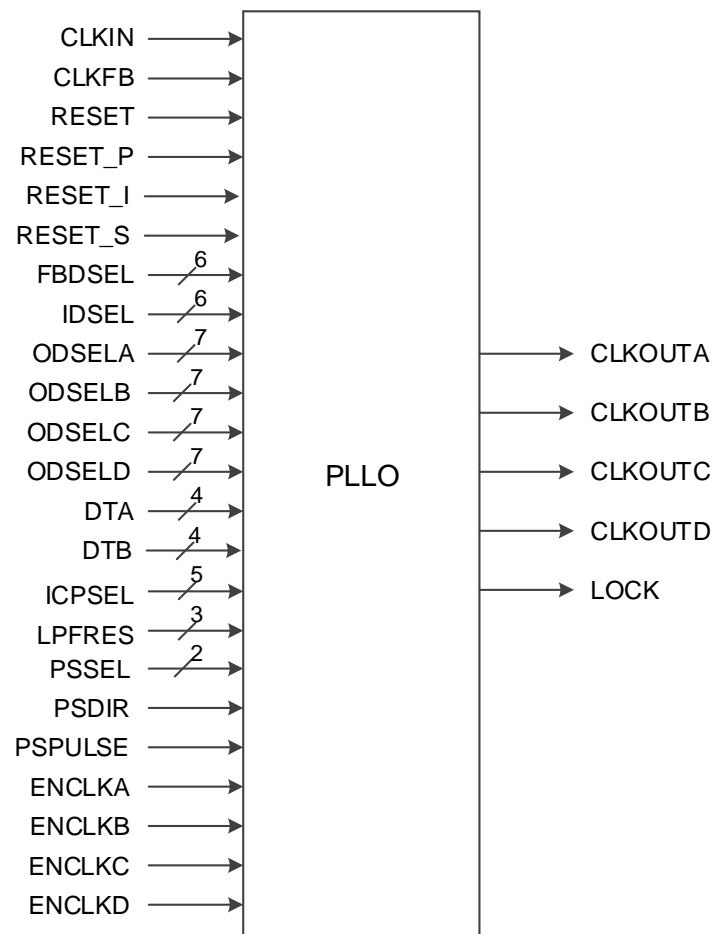
$$f_{PFD} = f_{CLKIN} / IDIV = f_{CLKOUTA} / FBDIV$$

注記：

- $f_{CLKIN}$  は入力クロック CLKIN の周波数です。
- $f_{CLKOUTX}$  は A/B/C/D チャンネルの出力クロックの周波数です ( $x=A/B/C/D$ )。ODIVX は A/B/C/D チャンネルの出力分周係数です ( $x=A/B/C/D$ )
- $f_{IN\_ODIVX}$  は ODIVX の入力クロックの周波数です ( $x=A/B/C/D$ )。デフォルトは  $f_{VCO}$  で、カスケードまたはバイパス時に実際の回路に従います。
- $f_{PFD}$  は位相検出器周波数です (3MHz 以上)。
- IDIV、FBDIV、ODIVX は、各分周器の分周係数です。分周係数を調整することにより、所望の周波数のクロック信号を生成することができます。
- PLLO の周波数の範囲については、[FPGA 製品データシート](#)を参照してください。

#### ポート図

図 5-7 PLLO のポート図



## ポートの説明

表 5-14 PLLO のポートの説明

ポート名	I/O	説明
CLKIN	入力	リファレンスクロック入力
CLKFB	入力	フィードバッククロック入力
RESET	入力	PLL全部リセット信号、アクティブHigh
RESET_P	入力	PLLパワーダウン (Power Down) 信号、アクティブHigh
RESET_I	入力	IDIV付き全部リセット信号、アクティブHigh
RESET_S	入力	B/C/Dの3つのチャンネルのリセット、アクティブHigh
FBDSEL[5:0]	入力	FBDIVの動的制御値、範囲は0~63、実際の値は64 - FBDSEL
IDSEL[5:0]	入力	IDIVの動的制御値、範囲は0~63、実際の値は64 - IDSEL
ODSELA[6:0]	入力	ODIVAの動的制御値、範囲は0~127、実際の値は128 - ODSELA
ODSELB[6:0]	入力	ODIVBの動的制御値、範囲は0~127、実際の値は128 - ODSELB
ODSELC[6:0]	入力	ODIVCの動的制御値、範囲は0~127、実際の値は128 - ODSELC
ODSELD[6:0]	入力	ODIVDの動的制御値、範囲は0~127、実際の値は128 - ODSELD
DTA[3:0]	入力	CLKOUTAのデューティサイクルの動的微調整
DTB[3:0]	入力	CLKOUTBのデューティサイクルの動的微調整
ICPSEL[4:0]	入力	ICP電流の動的制御。ICPSEL[4:0]の値の増加とともに電流が増加
LPFRES[2:0]	入力	LPFRESの値の動的制御。LPFRESの値は、R0からR7の範囲です。R0は最大の帯域幅に対応し、R7は最小の帯域幅に対応。
PSSEL[1:0]	入力	位相シフトチャンネル選択の動的制御
PSDIR	入力	位相シフト方向の動的制御
PSPULSE	入力	位相シフトパルスの動的制御
ENCLKA	入力	チャンネルAのクロック出力イネーブルの動的制御。動的イネーブルを使用するには、静的パラメータCLKOUTA_EN = "TRUE"にする必要がある
ENCLKB	入力	チャンネルBのクロック出力イネーブルの動的制御。動的イネーブルを使用するには、静的パラメータCLKOUTB_EN = "TRUE"にする必要がある
ENCLKC	入力	チャンネルCのクロック出力イネーブルの動的制御。動的イネーブルを使用するには、静的パラメータCLKOUTC_EN = "TRUE"にする必要がある
ENCLKD	入力	チャンネルDのクロック出力イネーブルの動的制御。動的イネーブルを使用するには、静的パラメータCLKOUTD_EN = "TRUE"にする必要がある
CLKOUTA	出力	Aチャンネルのクロック出力
CLKOUTB	出力	Bチャンネルのクロック出力
CLKOUTC	出力	Cチャンネルのクロック出力



ポート名	I/O	説明
CLKOUTD	出力	Dチャネルのクロック出力
LOCK	出力	PLLロック指示。1はロック、0はロック解除を示す

### パラメータの説明

表 5-15 PLLO のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
FCLKIN	"3"~"400"	"100.0"	リファレンスクロックの周波数(MHz)
IDIV_SEL	0~63	0	IDIV分周係数の静的設定。分周係数1~64に対応。
DYN_IDIV_SEL	"true", "false"	"false"	IDIV分周係数の静的制御パラメータまたは動的制御信号の選択 FALSE: 静的制御、つまりIDIV_SELを選択 TRUE: 動的制御、つまりIDSELを選択
FBDIV_SEL	0~63	0	FBDIV分周係数の静的設定。分周係数1~64に対応。
DYN_FBDIV_SEL	"true", "false"	"false"	FBDIV分周係数の静的制御パラメータまたは動的制御信号の選択 FALSE: 静的制御、つまりFBDIV_SELを選択 TRUE: 動的制御、つまりFBDSELを選択
ODIVA_SEL	1~128	4	ODIVA分周係数の静的設定
DYN_ODIVA_SEL	"true", "false"	"false"	ODIVA分周係数の静的制御パラメータまたは動的制御信号の選択 FALSE: 静的制御、つまりODIVA_SELを選択 TRUE: 動的制御、つまりODSELAを選択
ODIVB_SEL	1~128	4	ODIVB分周係数の静的設定
DYN_ODIVB_SEL	"true", "false"	"false"	ODIVB分周係数の静的制御パラメータまたは動的制御信号の選択 FALSE: 静的制御、つまりODIVB_SELを選択 TRUE: 動的制御、つまりODSELBを選択
ODIVC_SEL	1~128	4	ODIVC分周係数の静的設定
DYN_ODIVC_SEL	"true", "false"	"false"	ODIVC分周係数の静的制御パラメータまたは動的制御信号の選択 FALSE: 静的制御、つまりODIVC_SELを選択 TRUE: 動的制御、つまりODSELCを選択

パラメータ名	値の範囲	デフォルト値	説明
ODIVD_SEL	1~128	4	ODIVD分周係数の静的設定
DYN_ODIVD_SEL	"true", "false"	"false"	ODIVD分周係数の静的制御パラメータまたは動的制御信号の選択 FALSE: 静的制御、つまりODIVD_SELを選択 TRUE: 動的制御、つまりODSELDを選択
CLKOUTA_EN	"true", "false"	"true"	Aチャンネルのクロック出力イネーブル
CLKOUTB_EN	"true", "false"	"true"	Bチャンネルのクロック出力イネーブル
CLKOUTC_EN	"true", "false"	"true"	Cチャンネルのクロック出力イネーブル
CLKOUTD_EN	"true", "false"	"true"	Dチャンネルのクロック出力イネーブル
DYN_DTA_SEL	"true", "false"	"false"	Aチャンネルデューティサイクル微調整の静的制御パラメータまたは動的制御信号の選択 FALSE: 静的制御、つまりCLKOUTA_DT_DIR & CLKOUTA_DT_STEPを選択 TRUE: 動的制御、つまりDTAを選択
DYN_DTB_SEL	"true", "false"	"false"	Bチャンネルデューティサイクル微調整の静的制御パラメータまたは動的制御信号の選択 FALSE: 静的制御、つまりCLKOUTB_DT_DIR & CLKOUTB_DT_STEPを選択 TRUE: 動的制御、つまりDTBを選択
CLKOUTA_DT_DIR	1'b1, 1'b0	1'b1	Aチャンネルデューティサイクルの静的微調整方向 1'b1: 立ち上がりエッジ揃えでデューティサイクルが増加。 1'b0: 立ち下がりエッジ揃えでデューティサイクルが減少。
CLKOUTB_DT_DIR	1'b1, 1'b0	1'b1	Bチャンネルデューティサイクルの静的微調整方向 1'b1: 立ち上がりエッジ揃えでデューティサイクルが増加。 1'b0: 立ち下がりエッジ揃えでデューティサイクルが減少。
CLKOUTA_DT_STEP	0,1,2,4	0	Aチャンネルデューティサイクル静的微調整のステップサイズ。ステップごとに50ps。
CLKOUTB_DT_STEP	0,1,2,4	0	Bチャンネルデューティサイクル静的微調整のステップサイズ。ステップごとに50ps。
CLKA_IN_SEL	2'b00,2'b01,2'b11	2'b00	ODIVA入力クロックソースの選択 2'b00/2'b01: VCO出力から 2'b11: CLKINから(バイパス)
CLKA_OUT_SEL	1'b0, 1'b1	1'b0	Aチャンネル出力クロックソースの選

パラメータ名	値の範囲	デフォルト値	説明
			択 1'b0: ODIVA出力から 1'b1: CLKINから(バイパス)
CLKB_IN_SEL	2'b00,2'b01,2'b10,2'b11	2'b00	ODIVB入力クロックソースの選択 2'b00/2'b01: VCO出力から 2'b10: CLKCAS_Aから(カスケード) 2'b11: CLKINから(バイパス)
CLKB_OUT_SEL	1'b0, 1'b1	1'b0	Bチャンネル出力クロックソースの選択 1'b0: ODIVB出力から 1'b1: CLKINから(バイパス)
CLKC_IN_SEL	2'b00,2'b01,2'b10,2'b11	2'b00	ODIVC入力クロックソースの選択 2'b00/2'b01: VCO出力から 2'b10: CLKCAS_Bから(カスケード) 2'b11: CLKINから(バイパス)
CLKC_OUT_SEL	1'b0, 1'b1	1'b0	Cチャンネル出力クロックソースの選択 1'b0: ODIVC出力から 1'b1: CLKINから(バイパス)
CLKD_IN_SEL	2'b00,2'b01,2'b10,2'b11	2'b00	ODIVD入力クロックソースの選択 2'b00/2'b01: VCO出力から 2'b10: CLKCAS_Cから(カスケード) 2'b11: CLKINから(バイパス)
CLKD_OUT_SEL	1'b0, 1'b1	1'b0	Dチャンネル出力クロックソースの選択 1'b0: ODIVD出力から 1'b1: CLKINから(バイパス)
CLKFB_SEL	"INTERNAL", "EXTERNAL"	"INTERNAL"	CLKFBソースの選択 INTERNAL: 内部CLKOUTAからのフィードバック EXTERNAL: 外部信号からのフィードバック
DYN_DPA_EN	"true", "false"	"false"	動的位相シフト調整イネーブル
DYN_PSB_SEL	"true", "false"	"false"	Bチャンネル位相調整の静的制御パラメータまたは動的制御信号の選択 FALSE: 静的制御、つまりPSB_COARSE & PSB_FINE TRUEを選択 TRUE: 動的制御、つまりDPA (PSSEL & PSDIR & PSPULSE) を選択。DYN_DPA_EN="TRUE" にすることが必要。
DYN_PSC_SEL	"true", "false"	"false"	Cチャンネル位相調整の静的制御パラメータまたは動的制御信号の選択 FALSE: 静的制御、つまりPSC_COARSE & PSC_FINE TRUEを選択 TRUE: 動的制御、つまりDPA

パラメータ名	値の範囲	デフォルト値	説明
			(PSSEL& PSDIR& PSPULSE) を選択。DYN_DPA_EN="TRUE" にすることが必要。
DYN_PSD_SEL	"true", "false"	"false"	Dチャンネル位相調整の静的制御パラメータまたは動的制御信号の選択 FALSE: 静的制御、つまり PSD_COARSE & PSD_FINE TRUE を選択 TRUE: 動的制御、つまり DPA (PSSEL& PSDIR& PSPULSE) を選択。DYN_DPA_EN="TRUE" にすることが必要。
PSB_COARSE	0~128	1	Bチャンネル位相シフト粗調整静的設定
PSB_FINE	0~7	0	Bチャンネル位相シフト微調整静的設定
PSC_COARSE	0~128	1	Cチャンネル位相シフト粗調整静的設定
PSC_FINE	0~7	0	Cチャンネル位相シフト微調整静的設定
PSD_COARSE	0~128	1	Dチャンネル位相シフト粗調整静的設定
PSD_FINE	0~7	0	Dチャンネル位相シフト微調整静的設定
DTMS_ENB	"true", "false"	"false"	Bチャンネル (ODIVB = 2~128) のデューティサイクル調整イネーブル FALSE: 50%のデューティサイクル。 TRUE: DYN_PSB_SEL = "TRUE" の場合、PSB_COARSE & PSB_FINE を立ち下がりエッジとして設定し、動的位相調整を行って立ち上がりエッジを設定し、動的デューティサイクル調整 (立ち下がりエッジ - 立ち上がりエッジ) を実現。
DTMS_ENC	"true", "false"	"false"	Cチャンネル (ODIVC = 2~128) のデューティサイクル調整イネーブル FALSE: 50%のデューティサイクル。 TRUE: DYN_PSC_SEL = "TRUE" の場合、PSC_COARSE & PSC_FINE を立ち下がりエッジとして設定し、動的位相調整を行って立ち上がりエッジを設定し、動的デューティサイクル調整 (立ち下がりエッジ - 立ち上がりエッジ) を実現。
DTMS_END	"true", "false"	"false"	Dチャンネル (ODIVD = 2~128) のデューティサイクル調整イネーブル FALSE: 50%のデューティサイクル。 TRUE: DYN_PSD_SEL="true" の場合、PSD_COARSE & PSD_FINE を立ち下がりエッジとして設定し、動的位相調整を行って立ち上がりエッジを設定し、動的デューティサイクル調整 (立ち下がりエッジ - 立ち上がりエ

パラメータ名	値の範囲	デフォルト値	説明
			ッジ) を実現。
RESET_I_EN	"true", "false"	"false"	動的信号RESET_Iイネーブル。 RESET_Iポートを使用するには、このパラメータをTRUEに設定する必要がある。
RESET_S_EN	"true", "false"	"false"	動的信号RESET_Sイネーブル。 RESET_Sポートを使用するには、このパラメータをTRUEに設定する必要がある。
DYN_ICP_SEL	"true", "false"	"false"	ICPSEL静的制御パラメータまたは動的制御信号の選択 FALSE: 静的制御、つまりICP_SELを選択 TRUE: 動的制御、つまり動的信号ICPSELを選択
ICP_SEL	5'bXXXXX, 5'b00000~5'b11111	5'bXXXXX	ICP電流の静的設定 5'bXXXXX: ソフトウェアが自動的に計算してこのパラメータを設定。 5'b00000~5'b11111: ユーザーは、必要に応じてパラメータ範囲内で設定可能。
DYN_RES_SEL	"true", "false"	"false"	LPRREF静的制御パラメータまたは動的制御信号の選択 FALSE: 静的制御、つまりLPR_REFを選択 TRUE: 動的制御、つまり動的信号LPRFRESを選択
LPR_REF	7'bXXXXXXXX, 7'b00000000(R0), 7'b00000001(R1), 7'b00000010(R2), 7'b00000100(R3), 7'b00010000(R4), 7'b00100000(R5), 7'b01000000(R6), 7'b10000000(R7)	7'bXXXXXX XX	LPRRESの静的設定 7'bXXXXXXXX: ソフトウェアが自動的に計算してこのパラメータを設定。 7'b00000000~7'b10000000 (8つの値あり): ユーザーは、必要に応じてこの8つの値を設定可能。

表 5-16 IDSEL ポートのパラメータ参照テーブル

IDSEL[5:0]	IDIV静的パラメータ値	IDIV実際値
111111	0	1
111110	1	2
111101	2	3
111100	3	4
111011	4	5
111010	5	6
111001	6	7

111000	7	8
110111	8	9
.....	.....	.....
000000	63	64

表 5-17 FBDSEL ポートのパラメータ参照テーブル

FBDSEL [5:0]	FBDIV静的パラメータ値	FBDIV実際値
111111	0	1
111110	1	2
111101	2	3
111100	3	4
111011	4	5
111010	5	6
111001	6	7
111000	7	8
110111	8	9
.....	.....	.....
000000	63	64

表 5-18 ODSELX (X=A/B/C/D) ポートのパラメータ参照テーブル

ODSELX [6:0]	ODIVX 静的パラメータ値	ODIVX 実際値
111111	1	1
111110	2	2
111101	3	3
111100	4	4
111011	5	5
111010	6	6
111001	7	7
111000	8	8
110111	9	9
.....	.....	.....
000000	128	128

## 位相の調整

PLLO は、静的位相調整と動的位相調整をサポートします。その中で、動的位相調整は B/C/D チャネルでのみサポートされます。静的位相調整は、パラメータ PSX\_COARSE および PSX\_FINE (X=A/B/C/D) の設定によって実現されます。動的位相調整は、信号 PSSEL、PSDIR、

PSPULSE によって実現されます。PSSEL はチャネルの選択に使用され、PSDIR は加算または減算の制御に使用されます。PSPULSE パルスの立ち上がりエッジの場合、DYN\_FINE は 1 増加/減少し、DYN\_FINE のオーバーフローまたはアンダーフローの場合、DYN\_COARSE は 1 増加/減少します。DYN\_COARSE の値は ODIV より小さいです。

位相調整は、次の式で計算できます（例として B チャネルを使用）。

COARSE\_B < ODIVB の場合、 $ps = (FINE\_B/8 + COARSE\_B)/ODIVB \times 360$

COARSE\_B = ODIVB の場合、 $ps = (FINE\_B/8)/ODIVB \times 360$

注記：

- DYN\_FINE および DYN\_COARSE は、DPA によって生成される内部信号であり、PSSEL、PSDIR、および PSPULSE に応じて生成されます。
- FINE\_B は、DYN\_PSB\_SEL によって選択された動的 DYN\_FINE\_B または静的パラメータ PSB\_FINE であり、COARSE\_B は、DYN\_PSB\_SEL によって選択された動的 DYN\_COARSE\_B または静的パラメータ PSB\_COARSE です。
- CLKX\_IN\_SEL (X = B/C/D) がバイパスまたはカスケードの場合、FINE\_X (X = B/C/D) を 0 に設定する必要があります。

### デューティサイクル調整

PLL0 動的デューティサイクル調整は、B/C/D チャネルでのみサポートされます。デューティサイクルは次のように定義されます。

$$\text{Duty cycle} = (\text{falling edge} - \text{rising edge}) / \text{cycle\_period}$$

falling edge の位置は、静的位相シフト設定によって決定され、DUTY として定義されます。rising edge の位置は、動的位相シフト設定の PHASE によって決定されます。DYN\_FINE および DYN\_COARSE は、DPA によって生成される内部信号です。位相調整セクションの関連説明を参照してください。DUTY と PHASE の計算式は次のとおりです（B チャネル）。

$$\text{DUTY} = ((\text{PSB\_FINE}) / 8 + \text{PSB\_COARSE})$$

$$\text{PHASE} = ((\text{DYN\_FINEB}) / 8 + \text{DYN\_COARSEB})$$

動的デューティサイクルの計算：

- DUTY > PHASE の場合、 $\text{DutyCycle} = (\text{DUTY} - \text{PHASE}) / \text{ODIVB}$ 。
- DUTY < PHASE の場合、 $\text{DutyCycle} = (\text{DUTY} - \text{PHASE}) / \text{ODIVB} + 1$ 。

注記：

- ODIV = 1 の場合、動的デューティサイクル調整はサポートされておらず、デューティサイクルは 50% です。
- ODIV >= 2 の場合、DUTY-PHASE は (-0.5、0.5) の間の値をサポートしません。
- CLKX\_IN\_SEL (X = B/C/D) がバイパスまたはカスケードの場合、ODIV (> 2) が奇数の場合、デューティサイクルは 50% ではありません（高レベル < 低レベル、つまり 50% 未満）。

## デューティサイクルの微調整

PLLO の A/B チャンネルは、デューティサイクルの微調整をサポートします。これは、デューティサイクルの微調整の方向とステップサイズの設定によって実現でき、静的モードと動的モードをサポートします。微調整方向が 1'b1 の場合、立ち下がりエッジの遅延を調整し、デューティサイクルを増やします。微調整方向が 1'b0 の場合、立ち上がりエッジの遅延を調整し、デューティサイクルを減らします。

表 5-19 PLL0 デューティサイクル微調整の参照テーブル

デューティサイクルの微調整方向 <sup>[1]</sup>	デューティサイクルの微調整ステップサイズ <sup>[2]</sup>	デューティサイクルの微調整遅延
1'b0	0	0
	1	-50ps
	2	-100ps
	4	-200ps
1'b1	0	0
	1	+50ps
	2	+100ps
	4	+200ps

注記：

- [1] A：パラメータ CLKOUTA\_DT\_DIR/信号 DTA[3] B：パラメータ CLKOUTB\_DT\_DIR/信号 DTB[3]。
- [2] A：パラメータ CLKOUTA\_DT\_STEP または信号(7 - DTA[2:0]) B：パラメータ CLKOUTB\_DT\_STEP または信号(7 - DTB[2:0])。

A チャンネルと B チャンネルは同じ周波数クロックを出力し、B チャンネルクロックのデューティサイクルは、A チャンネルクロックを基準として微調整されます。そのタイミングを図 5-8 と図 5-9 に示します。

図 5-8 B チャンネルデューティサイクルの微調整タイミング図（微調整方向は 1'b1、ステップサイズは 1）

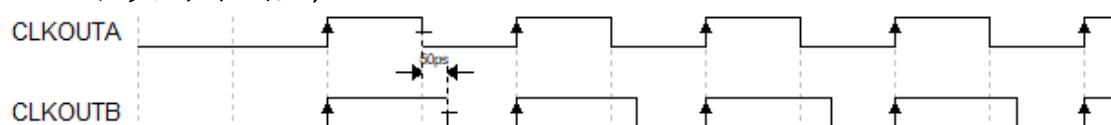
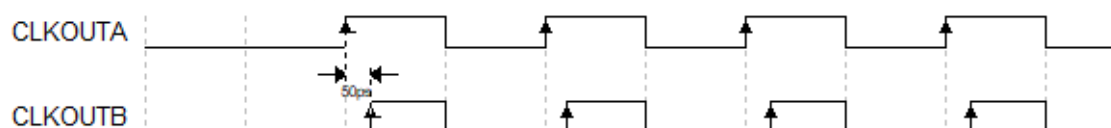


図 5-9 B チャンネルデューティサイクルの微調整タイミング図（微調整方向は 1'b0、ステップサイズは 1）



## ICPSEL/LPFRES の設定

PLLO は、ICPSEL および LPFRES の設定をサポートします。静的設定および動的設定があります。動的な場合、ユーザーは、実際のニーズに応



じて設定できます。静的な場合はデフォルトで X であり、Gowin ソフトウェアが自動的に計算して構成します。

ICPSEL の値の範囲は、小さい順で、ICP1、ICP2、..... ICPN ..... ICP31、ICP32、合計 32 のレベルに分割できます。ICP1 は最小電流に対応し、ICP32 は最大電流に対応します。すなわち、ICP の値は、N が大きいほど大きいです。

LPRRES の値の範囲は、小さい順で、R0、R1、R2、R3、R4、R5、R6、R7 です。R0 は最大帯域幅に対応し、R7 は最小帯域幅に対応します。次はいくつかの典型的な値です：R7-> 250KHz、R4-> 1.6MHz、R1-> 12MHz。

### プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。

**Verilog** でのインスタンス化：

```
PLLO pllo_inst (
    .LOCK(lock),
    .CLKOUTA(clkouta),
    .CLKOUTB(clkoutb),
    .CLKOUTC(clkoutc),
    .CLKOUTD(clkoutd),
    .CLKIN(clkin),
    .CLKFB(clkfb),
    .RESET(reset),
    .RESET_P(reset_p),
    .RESET_I(reset_i),
    .RESET_S(reset_s),
    .FBDSEL(fbdsel),
    .IDSEL(idsel),
    .ODSELA(odsel_a),
    .ODSELB(odsel_b),
    .ODSELC(odsel_c),
    .ODSELD(odsel_d),
    .DTA(dta),
    .DTB(dtb),
    .ICPSEL(icpsel),
```

```
.LPFRES(lpfres),  
.PSSEL(pssel),  
.PSDIR(psdire),  
.PSPULSE(phpulse),  
.ENCLKA(enclka),  
.ENCLKB(enclkb),  
.ENCLKC(enclkc),  
.ENCLKD(enclkd)  
);  
  
defparam pllo_inst.FCLKIN = "100";  
defparam pllo_inst.DYN_IDIV_SEL = "false";  
defparam pllo_inst.IDIV_SEL = 0;  
defparam pllo_inst.DYN_FBDIV_SEL = "false";  
defparam pllo_inst.FBDIV_SEL = 0;  
defparam pllo_inst.DYN_ODIVA_SEL = "false";  
defparam pllo_inst.ODIVA_SEL = 4;  
defparam pllo_inst.DYN_ODIVB_SEL = "false";  
defparam pllo_inst.ODIVB_SEL = 4;  
defparam pllo_inst.DYN_ODIVC_SEL = "false";  
defparam pllo_inst.ODIVC_SEL = 4;  
defparam pllo_inst.DYN_ODIVD_SEL = "false";  
defparam pllo_inst.ODIVD_SEL = 4;  
defparam pllo_inst.CLKOUTA_EN = "TRUE";  
defparam pllo_inst.CLKOUTB_EN = "false";  
defparam pllo_inst.CLKOUTC_EN = "false";  
defparam pllo_inst.CLKOUTD_EN = "false";  
defparam pllo_inst.DYN_DTA_SEL = "false";  
defparam pllo_inst.DYN_DTB_SEL = "false";  
defparam pllo_inst.CLKOUTA_DT_DIR = 1'b1;  
defparam pllo_inst.CLKOUTB_DT_DIR = 1'b1;  
defparam pllo_inst.CLKOUTA_DT_STEP = 0;  
defparam pllo_inst.CLKOUTB_DT_STEP = 0;  
defparam pllo_inst.CLKA_IN_SEL = 2'b00;  
defparam pllo_inst.CLKA_OUT_SEL = 1'b0;
```

```

defparam pllo_inst.CLKB_IN_SEL = 2'b00;
defparam pllo_inst.CLKB_OUT_SEL = 1'b0;
defparam pllo_inst.CLKC_IN_SEL = 2'b00;
defparam pllo_inst.CLKC_OUT_SEL = 1'b0;
defparam pllo_inst.CLKD_IN_SEL = 2'b00;
defparam pllo_inst.CLKD_OUT_SEL = 1'b0;
defparam pllo_inst.CLKFB_SEL = "INTERNAL";
defparam pllo_inst.DYN_DPA_EN = "false";
defparam pllo_inst.DYN_PSB_SEL = "false";
defparam pllo_inst.DYN_PSC_SEL = "false";
defparam pllo_inst.DYN_PSD_SEL = "false";
defparam pllo_inst.PSB_COARSE = 1;
defparam pllo_inst.PSB_FINE = 0;
defparam pllo_inst.PSC_COARSE = 1;
defparam pllo_inst.PSC_FINE = 0;
defparam pllo_inst.PSD_COARSE = 1;
defparam pllo_inst.PSD_FINE = 0;
defparam pllo_inst.DTMS_ENB = "false";
defparam pllo_inst.DTMS_ENC = "false";
defparam pllo_inst.DTMS_END = "false";
defparam pllo_inst.RESET_I_EN = "false";
defparam pllo_inst.RESET_S_EN = "false";
defparam pllo_inst.DYN_ICP_SEL = "false";
defparam pllo_inst.ICP_SEL = 5'bXXXXXX;
defparam pllo_inst.DYN_RES_SEL = "false";
defparam pllo_inst.LPR_REF = 7'bXXXXXXXX;

```

**VHDL** でのインスタンス化 :

COMPONENT PLLO

GENERIC (

FCLKIN : STRING := "100.0";

DYN\_IDIV\_SEL : STRING := "false";

IDIV\_SEL : integer := 0;

DYN\_FBDIV\_SEL : STRING := "false";

FBDIV\_SEL : integer := 0;

```
DYN_ODIVA_SEL : STRING := "false";
ODIVA_SEL : integer := 4;
DYN_ODIVB_SEL : STRING := "false";
ODIVB_SEL : integer := 4;
DYN_ODIVC_SEL : STRING := "false";
ODIVC_SEL : integer := 4;
DYN_ODIVD_SEL : STRING := "false";
ODIVD_SEL : integer := 4;
CLKOUTA_EN : STRING := "TRUE";
CLKOUTB_EN : STRING := "TRUE";
CLKOUTC_EN : STRING := "TRUE";
CLKOUTD_EN : STRING := "TRUE";

DYN_DTA_SEL : STRING := "false";
DYN_DTB_SEL : STRING := "false";
CLKOUTA_DT_DIR : bit := '1';
CLKOUTB_DT_DIR : bit := '1';
CLKOUTA_DT_STEP : integer := 0;
CLKOUTB_DT_STEP : integer := 0;
CLKA_IN_SEL : bit_vector := "00";
CLKA_OUT_SEL : bit := '0';
CLKB_IN_SEL : bit_vector := "00";
CLKB_OUT_SEL : bit := '0';
CLKC_IN_SEL : bit_vector := "00";
CLKC_OUT_SEL : bit := '0';
CLKD_IN_SEL : bit_vector := "00";
CLKD_OUT_SEL : bit := '0';
CLKFB_SEL : STRING := "INTERNAL";
DYN_DPA_EN : STRING := "false";
DYN_PSB_SEL : STRING := "false";
DYN_PSC_SEL : STRING := "false";
DYN_PSD_SEL : STRING := "false";
PSB_COARSE : integer := 1;
PSB_FINE : integer := 0;
PSC_COARSE : integer := 1;
```

```

PSC_FINE : integer := 0;
PSD_COARSE : integer := 1;
PSD_FINE : integer := 0;
DTMS_ENB : STRING := "false";
DTMS_ENC : STRING := "false";
DTMS_END : STRING := "false";
RESET_I_EN : STRING := "false";
RESET_S_EN : STRING := "false";
DYN_ICP_SEL : STRING := "false";
ICP_SEL : std_logic_vector(4 downto 0) := "XXXXX";
DYN_RES_SEL : STRING := "false";
LPR_REF : std_logic_vector(6 downto 0) := "XXXXXXX"
);
PORT (
    CLKIN : IN std_logic;
    CLKFB : IN std_logic:= '0';
    RESET, RESET_P : IN std_logic:= '0';
    RESET_I, RESET_S : IN std_logic:= '0';
    IDSEL, FBDSEL : IN std_logic_vector(5 downto 0);
    ODSELA, ODSELB, ODSELC, ODSELD : IN
std_logic_vector(6 downto 0);
    DTA, DTB : IN std_logic_vector(3 downto 0);
    ICPSEL : IN std_logic_vector(4 downto 0);
    LPFRES : IN std_logic_vector(2 downto 0);
    PSSEL : IN std_logic_vector(1 downto 0);
    PSDIR, PSPULSE : IN std_logic;
    ENCLKA, ENCLKB, ENCLKC, ENCLKD : IN std_logic;
    LOCK : OUT std_logic;
    CLKOUTA : OUT std_logic;
    CLKOUTB : OUT std_logic;
    CLKOUTC : OUT std_logic;
    CLKOUTD : OUT std_logic
);
END COMPONENT;
 uut:PLLO

```

## GENERIC MAP(

```
FCLKIN : STRING => "100.0";
DYN_IDIV_SEL => "false";
IDIV_SEL => 0;
DYN_FBDIV_SEL => "false";
FBDIV_SEL => 0;
DYN_ODIVA_SEL => "false";
ODIVA_SEL => 4;
DYN_ODIVB_SEL => "false";
ODIVB_SEL => 4;
DYN_ODIVC_SEL => "false";
ODIVC_SEL => 4;
DYN_ODIVD_SEL => "false";
ODIVD_SEL => 4;
CLKOUTA_EN => "TRUE";
CLKOUTB_EN => "TRUE";
CLKOUTC_EN => "TRUE";
CLKOUTD_EN => "TRUE";
DYN_DTA_SEL => "false";
DYN_DTB_SEL => "false";
CLKOUTA_DT_DIR => '1';
CLKOUTB_DT_DIR => '1';
CLKOUTA_DT_STEP => 0;
CLKOUTB_DT_STEP => 0;
CLKA_IN_SEL => "00";
CLKA_OUT_SEL => '0';
CLKB_IN_SEL => "00";
CLKB_OUT_SEL => '0';
CLKC_IN_SEL => "00";
CLKC_OUT_SEL => '0';
CLKD_IN_SEL => "00";
CLKD_OUT_SEL => '0';
CLKFB_SEL => "INTERNAL";
DYN_DPA_EN => "false";
DYN_PSB_SEL => "false";
```

```
DYN_PSC_SEL => "false";
DYN_PSD_SEL => "false";
PSA_COARSE => 0;
PSA_FINE => 0;
PSB_COARSE => 0;
PSB_FINE => 0;
PSC_COARSE => 0;
PSC_FINE => 0;
PSD_COARSE => 0;
PSD_FINE => 0;
DTMS_ENB => "false";
DTMS_ENC => "false";
DTMS_END => "false";
RESET_I_EN => "false";
RESET_S_EN => "false";
DYN_ICP_SEL => "false";
ICP_SEL => "XXXXXX";
DYN_RES_SEL => "false";
LPR_REF => "XXXXXXXX"
)
PORT MAP(
    LOCK=>lock,
    CLKOUTA=> clkouta,
    CLKOUTB=>clkoutb,
    CLKOUTC=>clkoutc,
    CLKOUTD=>clkoutd,
    CLKIN=>clkin,
    CLKFB=>clkfb,
    RESET=>reset,
    RESET_P=>reset_p,
    RESET_I=>reset_i,
    RESET_S=>reset_s,
    FBDSEL=>fbdsel,
    IDSEL=>idsel,
    ODSELA=>odsela,
```

```
ODSELB=>odselb,  
ODSELC=>odselc,  
ODSELD=>odseld,  
DTA=>dta,  
DTB=>dtb,  
ICPSEL=>icpsel,  
LPFRES=>lpfres,  
PSSEL=>pssel,  
PSDIR=>psdir,  
PSPULSE=>pspulse,  
ENCLKA=>enclka,  
ENCLKB=>enclkb,  
ENCLKC=>enclkc,  
ENCLKD=>enclkd
```

```
);
```

### 5.3.2 IP の呼び出し

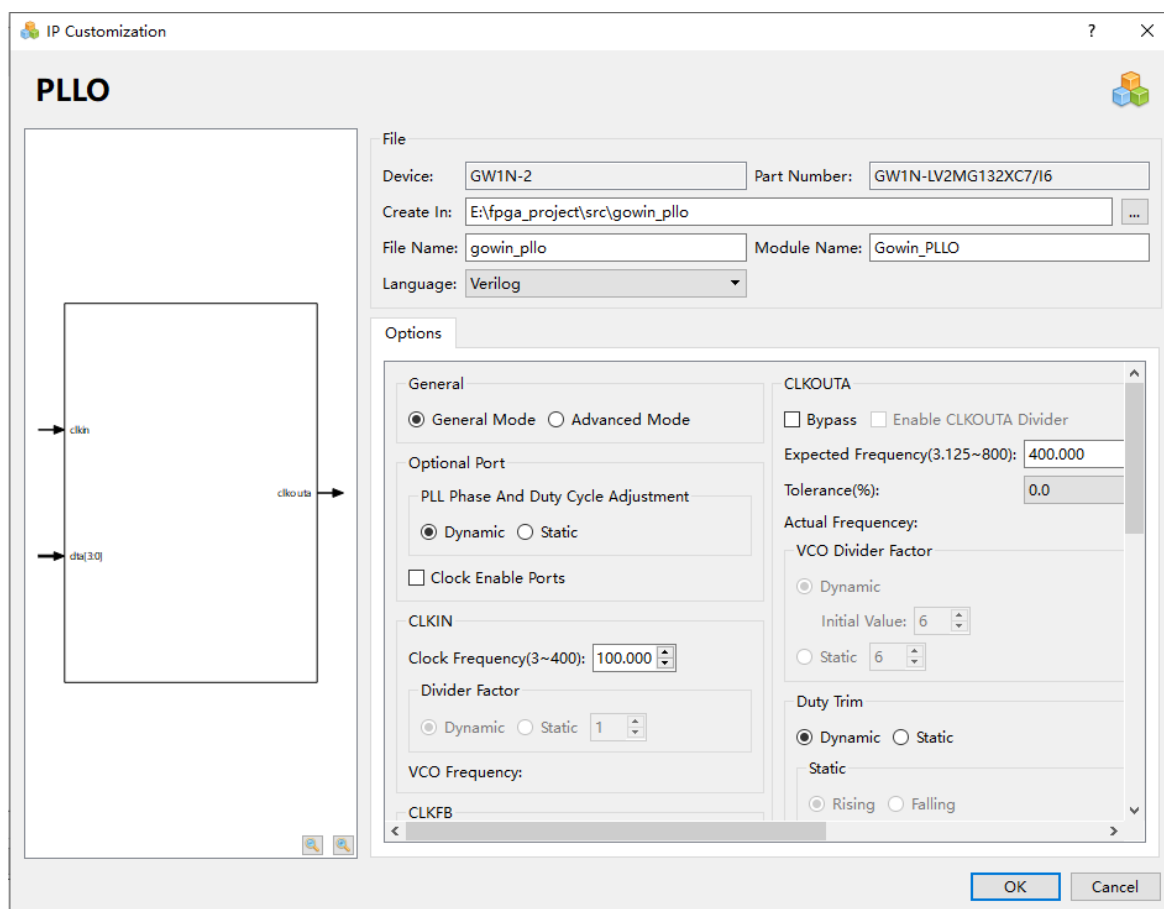
IP Core Generator のインターフェースで"PLLO"をクリックすると、右側に PLLO の概要が表示されます。

#### IP の構成

IP Core Generator インターフェースで"PLLO"をダブルクリックすると、PLLO の"IP Customization"ウィンドウがポップアップします。このウィンドウには File 構成タブ、Options 構成タブ、およびポート図があります (図 5-10)。



図 5-10 PLLO IP の構成ウィンドウ



### 1. File 構成タブ

File 構成タブは、IP ファイルの構成に使用されます。PLLO の File 構成タブの使用は DQCE モジュールと同様です。DQCE の File 構成タブを参照してください。

### 2. Options 構成タブ

Options 構成タブは IP のカスタマイズに使用されます(図 5-10)。

- **General** : IP Core 構成のモードを設定し、一般モード"General Mode"とアドバンスモード"Advanced Mode"をサポートします。
- **Optional Port** : 出力位相とデューティサイクルの調整の動的モード/静的モードの構成、PLLO 出力クロックのイネーブルの構成を実行します。一般モードでは、入力クロック周波数と出力クロック周波数を入力すると、ソフトウェアが自動的に周波数分割係数を計算します。アドバンスモードはアドバンスユーザーに適しており、アドバンスモードでは、入力周波数と周波数分割係数を入力して期待される出力を得ることができます。
  - **"PLL Phase And Duty Cycle Adjustment"** : 出力デューティサイクルおよび位相調整モードを構成し、動的調整"Dynamic"および静的調整"Static"をサポートします。

- **Clock Enable Ports** : PLL0 の出力クロックのポートを構成します。
- **CLKIN** : PLL0 入力クロックの周波数、分周パラメータなどを構成します。
  - **Clock Frequency** (周波数範囲) : 入力クロックの周波数を構成し、範囲は **3~400MHz** です。
  - **"Divide Factor"** : アドバンスモードで分周パラメータを構成し、動的モード**"Dynamic"**と静的モード**"Static"**をサポートします。静的モードでは分周パラメータの具体的な数値を構成でき、範囲は **1~64** です。**CLKOUT** の出力周波数が **device** の推奨動作要件の範囲外にある場合、**"Calculate"**または**"OK"**をクリックするとエラーメッセージのウィンドウがポップアップします。**CLKIN/IDIV** の出力周波数が **device** の推奨動作要件の範囲外にある場合、**"Calculate"**または**"OK"**をクリックするとエラーメッセージのウィンドウがポップアップします。
  - **VCO Frequency** : 計算された **VCO** の周波数であり、読み出し専用です。
- **CLKFB** : PLL0 フィードバッククロックのソースと周波数通倍パラメータを構成します。
  - フィードバッククロックのソース (**Source**) として **Internal** と **External** を選択できます。
  - **"Divide Factor"** : アドバンスモードで周波数通倍パラメータを構成し、動的モード**"Dynamic"**と静的モード**"Static"**をサポートします。静的モードでは周波数通倍パラメータの具体的な数値を構成でき、範囲は **1~64** です。構成が不適切な場合、**"Calculate"**ボタンまたは**"OK"**ボタンをクリックするとエラーメッセージのウィンドウがポップアップします。
- **ICP and LPF**
  - **ICPSEL** オプションは、**ICP** 電流を構成し、動的調整**"Dynamic"**と静的調整**"Static"**をサポートします。静的モードでは、**ICP** の特定の値を **ICP1~ICP32** の範囲で構成できます。デフォルトは **X** であり、この場合、ソフトウェアが自動的に計算して構成します。
  - **LPFRES** オプションは、ローパスフィルター抵抗を構成し、動的調整**"Dynamic"**と静的調整**"Static"**をサポートします。静的モードでは、**RES** の特定の値を **R0~R7** の範囲で構成できます。デフォルトは **X** であり、この場合、ソフトウェアが自動的に計算して構成します。
- **PLL Reset**
  - **PLL Reset** : チェックすると、**PLL0** のリセットが有効になり

ます。

- **PLL Power Down** : チェックすると、**RESET\_P** ポートを介して **PLLO** をパワーダウンモードにします。
- **CLKIN Divider Reset** : **RESET\_I** のイネーブルを構成します。
- **CLKOUTB/CLKOUTC/CLKOUTD Divider Reset** : **RESET\_S** のイネーブルを構成します。
- **Enable LOCK** : チェックすると、**LOCK** ポートがイネーブルされます。
- **CLKOUTA : A チャンネル PLLO 出力クロックの目的の周波数、VCO パラメータ、出力クロックの微調整デューティサイクルパラメータを構成します。**
  - **"Bypass"** : チェックすると、バイパス機能がイネーブルされます。
  - **Enable CLKOUTA Divider** : **VCO** クロックのバイパス機能を構成します。
  - **Expected Frequency** (周波数範囲) : 一般モードで目的の出力クロック **CLKOUTA** の周波数を構成します。非 **bypass** モードでは、範囲は **3.125~800MHz** です。
  - **Tolerance (%)** : **CLKOUTA** の目的の周波数と算出した実際周波数の許容誤差を構成します。
  - **Actual Frequency** : 計算で得られた **CLKOUTA** の実際出力周波数を表示します。
  - **VCO Divide Factor** : アドバンスモードで **VCO** パラメータを構成し、動的モード**"Dynamic"**と静的モード**"Static"**をサポートします。静的モードでは分周パラメータの具体的な数値を構成でき、範囲は **1~128** です。構成が不適切な場合、**"Calculate"**または**"OK"**をクリックするとエラーメッセージのウィンドウがポップアップします。
  - **Duty Trim** : デューティサイクルの微調整を構成し、動的モード**"Dynamic"**と静的モード**"Static"**をサポートします。静的モードは**"Rising"**と**"Falling"**があり、**"Step"**の値 (**0、1、2、4**) を構成できます。
- **CLKOUTB : B チャンネル PLLO 出力クロックの目的の周波数、VCO パラメータ、出力クロックの微調整デューティサイクルパラメータ、位相、およびデューティサイクル調整パラメータを構成します。**
  - **"Bypass"** : チェックすると、バイパス機能がイネーブルされます。
  - **Enable CLKOUTB Divider** : **VCO** クロックのバイパス機能を構成します。

- **Expected Frequency**（周波数範囲）：一般モードで目的の出力クロック CLKOUTB の周波数を構成します。非 **bypass** モードでは、範囲は 3.125~800MHz です。
  - **Tolerance (%)**：CLKOUTB の目的の周波数と算出した実際周波数の許容誤差を構成します。
  - **Actual Frequency**：計算された CLKOUT の実際出力周波数を表示します。
  - **VCO Divide Factor**：アドバンスモードで VCO パラメータを構成し、動的モード"**Dynamic**"と静的モード"**Static**"をサポートします。静的モードでは分周パラメータの具体的な数値を構成でき、範囲は 1~128 です。構成が不適切な場合、"**Calculate**"または"**OK**"をクリックするとエラーメッセージのウィンドウがポップアップします。
  - **Duty Trim**：デューティサイクルの微調整を構成し、動的モード"**Dynamic**"と静的モード"**Static**"をサポートします。静的モードは"**Rising**"と"**Falling**"があり、"**Step**"の値（0、1、2、4）を構成できます。
  - **"Phase (degree)"** 配置調整的相位度数，支持动态模式"**Dynamic**"和静态模式"**Static**"，静态模式下配置相位度数；
  - **DutyCycle**：デューティサイクルを構成します。動的モード"**Dynamic**"と静的モード"**Static**"をサポートし、静的モードでは 50%です。動的デューティサイクル調整は、位相の調整と動的 DPA 調整によって実現されます。
- **CLKOUTC**：C チャンネル PLL0 出力クロックの目的の周波数、VCO パラメータ、位相、およびデューティサイクル調整パラメータを構成します。
    - **"Bypass"**：チェックすると、バイパス機能がイネーブルされます。
    - **Enable CLKOUTC Divider**：VCO クロックのバイパス機能を構成します。
    - **Expected Frequency**（周波数範囲）：一般モードで目的の出力クロック CLKOUTC の周波数を構成します。非 **bypass** モードでは、範囲は 3.125~800MHz です。
    - **Tolerance (%)**：CLKOUTC の目的の周波数と算出した実際周波数の許容誤差を構成します。
    - **Actual Frequency**：計算された CLKOUTC の実際出力周波数を表示します。
    - **VCO Divide Factor**：アドバンスモードで VCO パラメータを構成し、動的モード"**Dynamic**"と静的モード"**Static**"をサポートし

ます。静的モードでは分周パラメータの具体的な数値を構成でき、範囲は 1~128 です。構成が不適切な場合、"Calculate"または"OK"をクリックするとエラーメッセージのウィンドウがポップアップします。

- **Phase (degree)** : 位相度の調整を構成し、動的モード "Dynamic" および静的モード "Static" をサポートします。静的モードで位相度を構成します。
- **DutyCycle** : デューティサイクルを構成します。動的モード "Dynamic" と静的モード "Static" をサポートし、静的モードでは 50% です。動的デューティサイクル調整は、位相の調整と動的 DPA 調整によって実現されます。
- **CLKOUTD : D チャンネル PLL0 出力クロックの目的の周波数、VCO パラメータ、位相、およびデューティサイクル調整パラメータを構成します。**
  - **"Bypass"** : チェックすると、バイパス機能がイネーブルされます。
  - **Enable CLKOUTD Divider** : VCO クロックのバイパス機能を構成します。
  - **Expected Frequency (周波数範囲)** : 一般モードで目的の出力クロック CLKOUTD の周波数を構成します。非 bypass モードでは、範囲は 3.125~800MHz です。
  - **Tolerance (%)** : CLKOUTD の目的の周波数と算出した実際周波数の許容誤差を構成します。
  - **Actual Frequency** : 計算された CLKOUTD の実際出力周波数を表示します。
  - **VCO Divide Factor** : アドバンスモードで VCO パラメータを構成し、動的モード "Dynamic" と静的モード "Static" をサポートします。静的モードでは分周パラメータの具体的な数値を構成でき、範囲は 1~128 です。構成が不適切な場合、"Calculate"または"OK"をクリックするとエラーメッセージのウィンドウがポップアップします。
  - **Phase (degree)** : 位相度の調整を構成し、動的モード "Dynamic" および静的モード "Static" をサポートします。静的モードで位相度を構成します。
  - **DutyCycle** : デューティサイクルを構成します。動的モード "Dynamic" と静的モード "Static" をサポートし、静的モードでは 50% です。動的デューティサイクル調整は、位相の調整と動的 DPA 調整によって実現されます。
- **Calculate** : 現在の構成が適切か計算します。

- 一般モード"**General Mode**"で入力・出力周波数の計算に従って分周パラメータ、周波数逡倍パラメータ、**VCO** パラメータを構成します。計算された実際の周波数が目的の周波数と等しくない場合、"**Calculate**"ボタンをクリックすると"**error**"ウィンドウがポップアップします。
- アドバンスモード"**Advanced Mode**"で、構成された静的分周パラメータ、周波数逡倍パラメータ、**VCO** パラメータが適切か計算します。適切でない場合は、"**Calculate**"をクリックすると"**error**"メッセージが表示されます。正しい場合は、"**Calculate**"をクリックすると、構成が成功したことを示す"**info**"ウィンドウが表示されます。

### 3. ポート図

ポート図は、現在の **IP Core** の構成結果を表示し、入力・出力ポートのビット幅は **Options** 構成に従ってリアルタイムで更新されます（図 5-10）。

#### 生成されるファイル

IP の構成が完了したら、構成ファイルの"**File Name**"によって命名された 3 つのファイルが生成されます：

- "**gowin\_pllo.v**"は完全な **verilog** モジュールです。
- **gowin\_pllo\_tmp.v** は IP のテンプレートファイルです。
- **gowin\_pllo.ipc** は IP の構成ファイルです。

注記：

VHDL が設計の言語として選択されている場合、生成される最初の 2 つのファイル名のサフィックスは **.vhd** になります。

## 5.4 DLLDLY

### 5.4.1 プリミティブの紹介

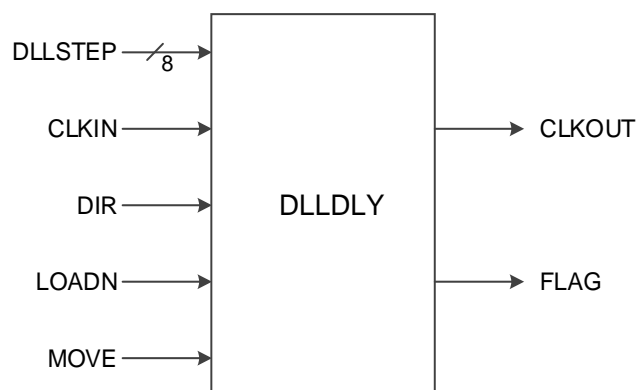
クロック遅延モジュールとして、**DLLDLY** は **DLLSTEP** 信号に基づきクロックを遅延して出力します。

#### 機能の説明

**DLLDLY** は **DLLSTEP** に従って位相に対応する遅延を生成します。これによって **CLKIN** に基づいた遅延出力が実現されます。

## ポート図

図 5-11 DLLDLY のポート図



## ポートの説明

表 5-20 DLLDLY のポートの説明

ポート名	I/O	説明
CLKOUT	出力	クロック出力
FLAG	出力	動的遅延調整のunder-flowまたはover-flowを示す出力フラグビット
DLLSTEP[7:0]	入力	遅延ステップ入力
CLKIN	入力	クロック入力
DIR	入力	遅延の動的調整の方向を設定 0: 遅延を増やす 1: 遅延を減らす
LOADN	入力	遅延ステップのロードの制御 0: 遅延ステップのロード。 1: 遅延の動的調整
MOVE	入力	MOVEは立ち下がりエッジの場合の動的調整された遅延で、パルスごとに1遅延ステップ移動します

## パラメータの説明

表 5-21 DLLDLY のパラメータの説明

パラメータ名	パラメータのタイプ	値の範囲	デフォルト値	説明
DLL_INSEL	Integer	1'b1	1'b1	1'b1: ノーマルモード。DLLDLY遅延モジュールを使用。
DLY_SIGN	String	1'b0,1'b1	1'b0	遅延調整の記号を設定: 1'b0: '+' 1'b1: '-'
DLY_ADJ	Integer	0~255	0	遅延調整の設定 : dly_sign=0 DLY_ADJ; dly_sign=1 -256+ DLY_ADJ

### プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。

#### Verilog でのインスタンス化 :

```
DLLDLY dlldly_0 (
    .CLKIN(clkin),
    .DLLSTEP(step[7:0]),
    .DIR(dir),
    .LOADN(loadn),
    .MOVE(move),
    .CLKOUT(clkout),
    .FLAG(flag)
);
```

```
defparam dlldly_0.DLL_INSEL=1'b1;
```

```
defparam dlldly_0.DLY_SIGN=1'b1;
```

```
defparam dlldly_0.DLY_ADJ=0;
```

#### VHDL でのインスタンス化 :

```
COMPONENT DLLDLY
    GENERIC(
        DLL_INSEL:bit:=0';
        DLY_SIGN:bit:=0';
        LY_ADJ:integer:=0
    );
    PORT(
        DLLSTEP:IN std_logic_vector(7 downto 0);
        CLKIN:IN std_logic;
        DIR,LOADN,MOVE:IN std_logic;
        CLKOUT:OUT std_logic;
        FLAG:OUT std_logic
    );
END COMPONENT;

 uut:DLLDLY
    GENERIC MAP(
        DLL_INSEL=>'1',
```



```
        DLY_SIGN=>'0',  
        LY_ADJ=>0  
    )  
    PORT MAP(  
        DLLSTEP=>step,  
        CLKIN=>clkin,  
        DIR=>dir,  
        LOADN=>loadn,  
        MOVE=>move,  
        CLKOUT=>clkout,  
        FLAG=>flag  
    );
```

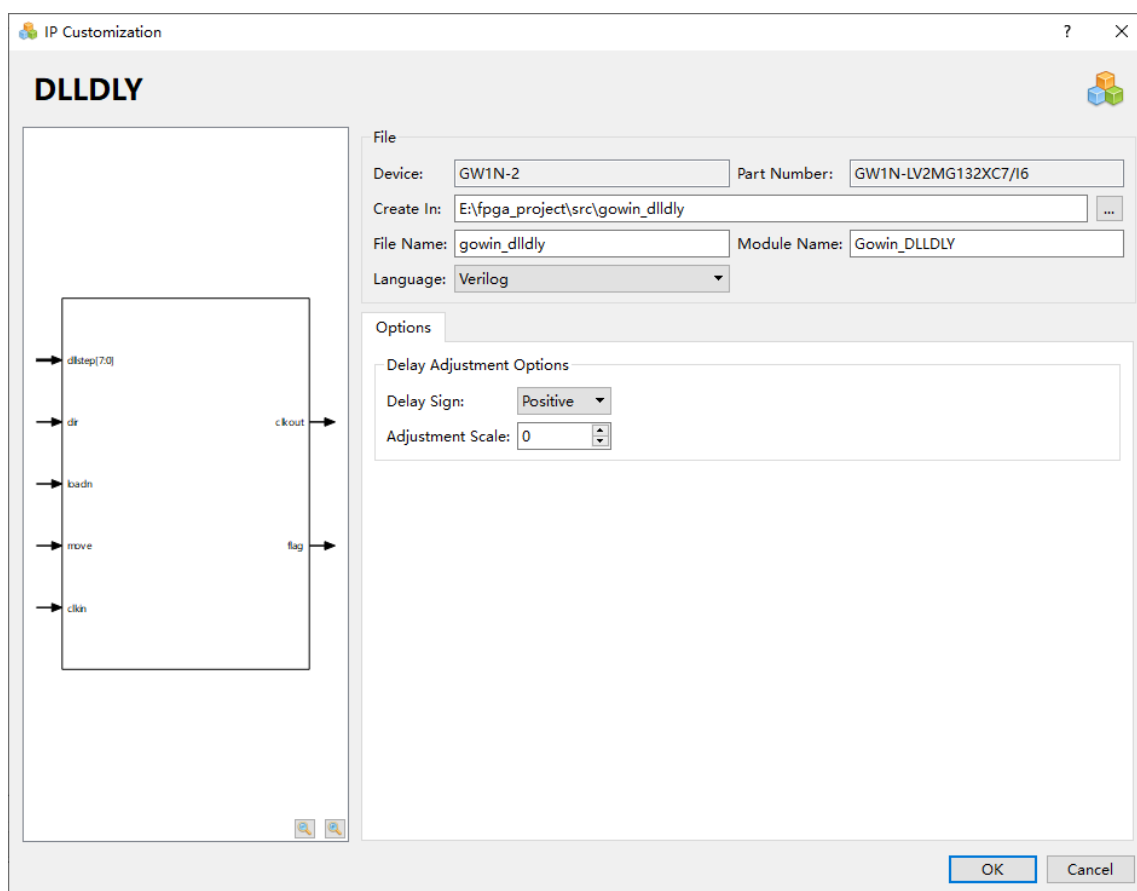
### 5.4.2 IP の呼び出し

IP Core Generator のインターフェースで DLLDLY をクリックすると、右側に DLLDLY の概要が表示されます。

#### IP の構成

IP Core Generator インターフェースで"DLLDLY"をダブルクリックすると、"IP Customization"ウィンドウがポップアップします。このウィンドウには File 構成タブ、Options 構成タブ、およびポート図があります (図 5-12)。

図 5-12 DLLDLY IP の構成ウィンドウ



### 1. File 構成タブ

File 構成タブは、IP ファイルの構成に使用されます。DLLDLY の File 構成タブの使用は DQCE モジュールと同様です。DQCE の File 構成タブを参照してください。

### 2. Options 構成タブ

Options 構成タブは IP のカスタマイズに使用されます(図 5-12)。

- Delay Sign : 遅延調整の記号を設定します。
- Adjustment Scale : 調整のスケールを設定します。

### 3. ポート図

ポート図は、IP Core の構成結果を表示します(図 5-12)。

### 生成されるファイル

IP の構成が完了したら、構成ファイルの"File Name"によって命名された 3 つのファイルが生成されます :

- "gowin\_dllily.v"は完全な verilog モジュールです。
- gowin\_dllily\_tmp.v は IP のテンプレートファイルです。

- gowin\_dllldly.ipc は IP の構成ファイルです。

注記：

VHDL が設計の言語として選択されている場合、生成される最初の 2 つのファイル名のサフィックスは.vhd になります。

## 5.5 CLKDIV

### 5.5.1 プリミティブの紹介

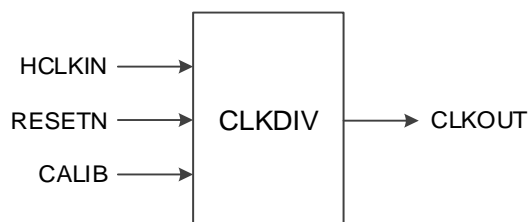
CLKDIV は、クロック周波数を調整するクロック分周器です。

#### 機能の説明

CLKDIV は、入力クロックと位相が一致し、IO ロジックに使用される分周クロックを生成する高速クロック分周モジュールです。GW1N-1S、GW1NS-4、GW1NS-4C、GW1NSR-4、GW1NSR-4C、GW1NSER-4C、GW1N-9、GW1N-9C、GW1NR-9、GW1NR-9C、GW1N-2、GW1N-1P5、GW1N-2B、GW1N-1P5B、GW1NR-2、および GW1NR-2B は 2/3.5/4/5/8 分周をサポートし、その他デバイスは 2/3.5/4/5 分周をサポートします。

#### ポート図

図 5-13 CLKDIV のポート図



#### ポートの説明

表 5-22 CLKDIV のポートの説明

ポート名	I/O	説明
HCLKIN	入力	クロック入力
RESETN	入力	非同期リセット信号、アクティブ Low
CALIB	入力	CALIB 入力。クロック出力を調整。
CLKOUT	出力	クロック出力

CALIB 信号は IOLOGIC の CALIB と併用することで、以下のような機能を提供できます。

- 2 分周の場合、2 立ち下がりエッジごとに位相を 180 度調整し、2 回の調整を 1 サイクルとします。
- 3.5 分周の場合、1 立ち下がりエッジごとに位相を約 102.8 度調整し、7 回の調整を 1 サイクルとします。

- 4 分周の場合、2 立ち下がりエッジごとに位相を 90 度調整し、4 回の調整を 1 サイクルとします。
- 5 分周の場合、2 立ち下がりエッジごとに位相を 72 度調整し、5 回の調整を 1 サイクルとします。
- 8 分周の場合、2 立ち下がりエッジごとに位相を 45 度調整し、8 回の調整を 1 サイクルとします。

### パラメータの説明

表 5-23 CLKDIV のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
DIV_MODE	2, 3.5, 4, 5 (8)	2	クロック分周係数を設定。
GSREN	"false", "true"	"false"	グローバルリセットGSRを有効にする

### プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。

#### Verilog でのインスタンス化 :

```
CLKDIV clkdiv_inst (
    .HCLKIN(hclkin),
    .RESETN(resetn),
    .CALIB(calib),
    .CLKOUT(clkout)
);
defparam clkdiv_inst.DIV_MODE="3.5";
defparam clkdiv_inst.GSREN="false";
```

#### VHDL でのインスタンス化 :

```
COMPONENT CLKDIV
    GENERIC(
        DIV_MODE:STRING:="2";
        GSREN:STRING:"false"
    );
    PORT(
        HCLKIN:IN std_logic;
        RESETN:IN std_logic;
        CALIB:IN std_logic;
```

```
        CLKOUT:OUT std_logic
    );
END COMPONENT;
uut:CLKDIV
    GENERIC MAP(
        DIV_MODE=>"2",
        GSREN=>"false"
    )
    PORT MAP(
        HCLKIN=>hclkin,
        RESETN=>resetn,
        CALIB=>calib,
        CLKOUT=>clkout
    );
```

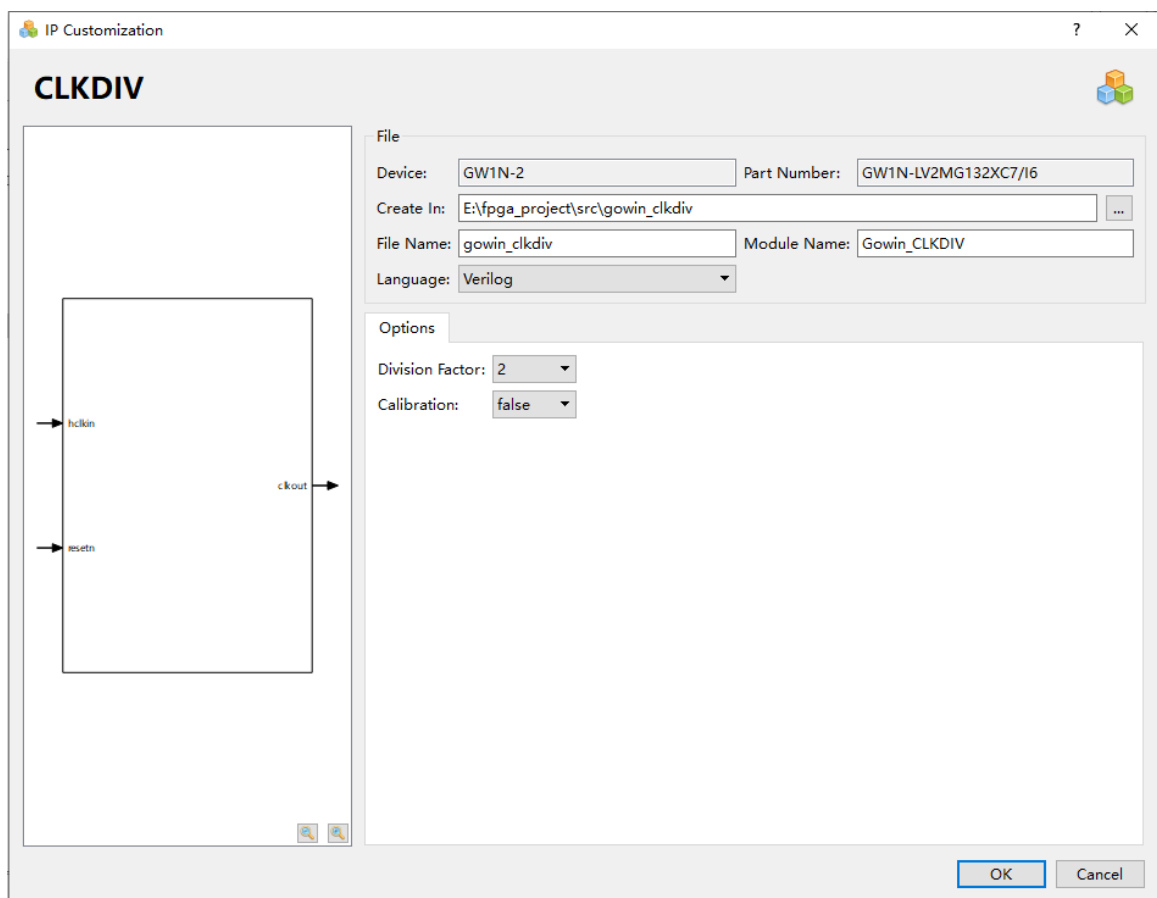
### 5.5.2 IP の呼び出し

IP Core Generator のインターフェースで **CLKDIV** をクリックすると、右側に **CLKDIV** の概要が表示されます。

#### IP の構成

IP Core Generator インターフェースで"**CLKDIV**"をダブルクリックすると、"**IP Customization**"ウィンドウがポップアップします。このウィンドウには **File** 構成タブ、**Options** 構成タブ、およびポート図があります (図 5-14)。

図 5-14 CLKDIV IP の構成ウィンドウ



### 1. File 構成タブ

File 構成タブは、IP ファイルの構成に使用されます。CLKDIV の File 構成タブの使用は DQCE モジュールと同様です。DQCE の File 構成タブを参照してください。

### 2. Options 構成タブ

Options 構成タブは IP のカスタマイズに使用されます(図 5-14)。

- Division Factor : 除算係数。
- Calibration : 校正クロックのイネーブルオプション。

### 3. ポート図

ポート図は、IP Core の構成結果を表示します(図 5-14)。

### 生成されるファイル

IP の構成が完了したら、構成ファイルの"File Name"によって命名された 3 つのファイルが生成されます :

- "gowin\_clkdiv.v"は完全な verilog モジュールです。
- gowin\_clkdiv\_tmp.v は IP のテンプレートファイルです。

- gowin\_clkdiv.ipc は IP の構成ファイルです。

注記：

VHDL が設計の言語として選択されている場合、生成される最初の 2 つのファイル名のサフィックスは.vhd になります。

## 5.6 CLKDIVG

### 5.6.1 プリミティブの紹介

CLKDIVG は、クロック周波数を調整するクロック分周器です。

サポートされるデバイス

表 5-24 CLKDIVG 対応デバイス

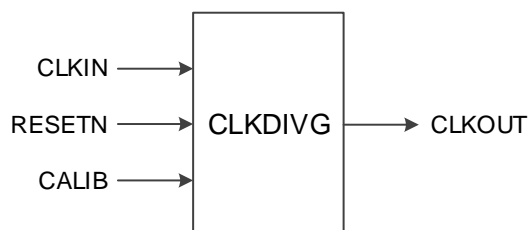
ファミリー	シリーズ	デバイス
LittleBee®	GW1N	GW1N-2, GW1N-1P5, GW1N-2B, GW1N-1P5B
	GW1NR	GW1NR-2, GW1NR-2B
Arora	GW2AN	GW2AN-18X, GW2AN-9X

機能の説明

CLKDIVG は、入力クロックと位相が一致する分周クロックを生成するクロック分周モジュールです。CLKDIVG は 1 つだけで、位置は固定されています。その入力は固定の IO からのものであり、機能は CLKDIV と同じです。

ポート図

図 5-15 CLKDIVG のポート図



ポートの説明

表 5-25 CLKDIVG のポートの説明

ポート名	I/O	説明
CLKIN	入力	クロック入力
RESETN	入力	非同期リセット信号、アクティブ Low
CALIB	入力	CALIB 入力。クロック出力を調整。
CLKOUT	出力	クロック出力

CALIB 信号は IOLOGIC の CALIB と併用することができます(5.5 CLKDIV 参照)。

## パラメータの説明

表 5-26 CLKDIVG のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
DIV_MODE	2, 3.5, 4, 5, 8	2	クロック分周係数を設定。
GSREN	"false", "true"	"false"	グローバルリセットGSRを有効にする

## プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。

**Verilog** でのインスタンス化：

```
CLKDIVG CLKDIVG_inst (
    .CLKIN(clkin),
    .RESETN(resetn),
    .CALIB(calib),
    .CLKOUT(clkout)
);
defparam CLKDIVG_inst.DIV_MODE="2";
defparam CLKDIVG_inst.GSREN="false";
```

**VHDL** でのインスタンス化：

```
COMPONENT CLKDIVG
    GENERIC(
        DIV_MODE:STRING:="2";
        GSREN:STRING:="false"
    );
    PORT(
        CLKIN:IN std_logic;
        RESETN:IN std_logic;
        CALIB:IN std_logic;
        CLKOUT:OUT std_logic
    );
END COMPONENT;

uut:CLKDIVG
    GENERIC MAP(
        DIV_MODE=>"2",
```



```

        GSREN=>"false"
    )
    PORT MAP(
        CLKIN=>clkIn,
        RESETN=>resetn,
        CALIB=>calib,
        CLKOUT=>clkout
    );

```

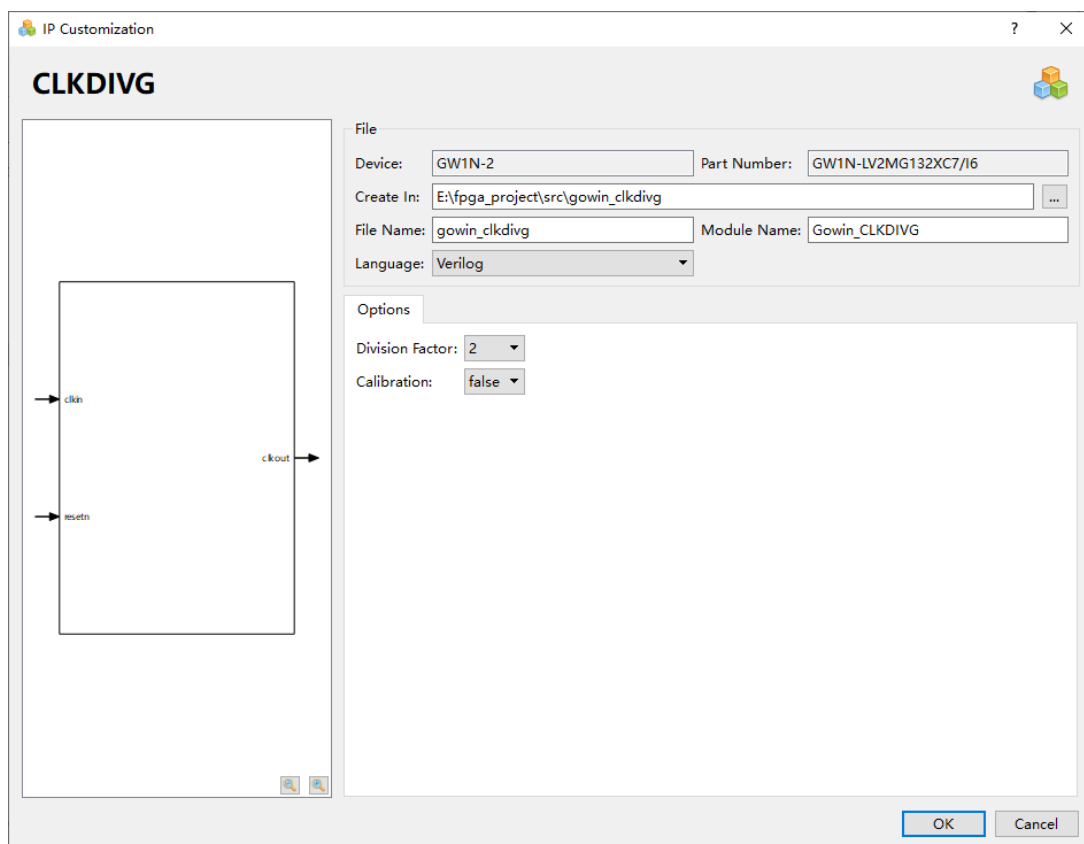
## 5.6.2 IP の呼び出し

IP Core Generator のインターフェースで **CLKDIVG** をクリックすると、右側に **CLKDIVG** の概要が表示されます。

### IP の構成

IP Core Generator インターフェースで"**CLKDIVG**"をダブルクリックすると、"**IP Customization**"ウィンドウがポップアップします。このウィンドウには **File** 構成タブ、**Options** 構成タブ、およびポート図があります (図 5-16)。

図 5-16 CLKDIVG IP の構成ウィンドウ



### 1. File 構成タブ

File 構成タブは、IP ファイルの構成に使用されます。

- CLKDIVG の File 構成タブの使用は DQCE モジュールと同様です。DQCE の File 構成タブを参照してください。

## 2. Options 構成タブ

Options 構成タブは IP のカスタマイズに使用されます(図 5-16)。

- Division Factor : 除算係数。
- Calibration : 校正クロックのイネーブルオプション。

## 3. ポート図

ポート図は、IP Core の構成結果を表示します(図 5-16)。

## 生成されるファイル

IP の構成が完了したら、構成ファイルの"File Name"によって命名された 3 つのファイルが生成されます：

- "gowin\_clkdivg.v"は完全な verilog モジュールです。
- gowin\_clkdivg\_tmp.v は IP のテンプレートファイルです。
- "gowin\_clkdivg.ipc"は IP の構成ファイルです。

注記：

VHDL が設計の言語として選択されている場合、生成される最初の 2 つのファイル名のサフィックスは.vhd になります。

# 5.7 DQS

## 5.7.1 プリミティブの紹介

DQS は DDR メモリインターフェースの双方向データストロブ回路です。

## サポートされるデバイス

表 5-27 DQS 対応デバイス

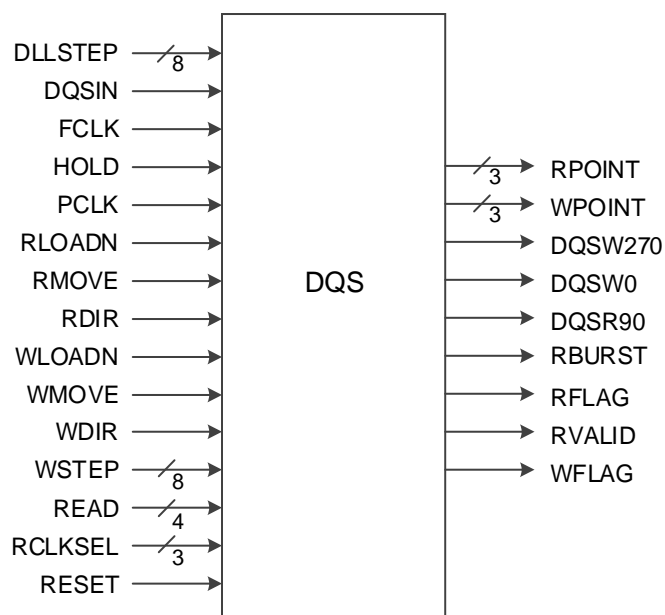
ファミリー	シリーズ	デバイス
Arora	GW2A	GW2A-18, GW2A-18C, GW2A-55, GW2A-55C
	GW2AN	GW2AN-55C, GW2AN-18X, GW2AN-9X
	GW2AR	GW2AR-18, GW2AR-18C
	GW2ANR	GW2ANR-18C

## 機能の説明

DQS は、主に DQSIN と DQSR90、DQSW0 と DQSW270 の信号間の位相関係の調整、および書き込みバランスと読み出し較正に用いるメモリコントローラ IP のキーコンポーネントです。

## ポート図

図 5-17 DQS のポート図



## ポートの説明

表 5-28 DQS のポートの説明

ポート名	I/O	説明
DLLSTEP[7:0]	入力	DQS遅延ステップ制御入力
DQSIN	入力	IO PADからのDQS入力
FCLK	入力	二つの異なるFCLKクロックツリーの出力から得る高速クロック
HOLD	入力	DQS書き込みに使用の場合、関連する信号の書き込みを停止して出力クロックを同期。DQS読み出しに使用の場合、FIFOカウンタをリセット
PCLK	入力	PCLKクロックツリーからのプライマリクロック
RDIR	入力	DDR読み出しの遅延方向を調整 "0"は遅延を増やす "1"は遅延を減らす
RLOADN	入力	DDR読み出しの最終遅延ステップサイズを初期化、アクティブLow
RMOVE	入力	RMOVEが立ち下がりエッジのときにDDR読み出しの遅延ステップサイズを変更、パルスごとに1回変更
WDIR	入力	DDR書き込み遅延方向を調整 "0"は遅延を増やす "1"は遅延を減らす
WLOADN	入力	DDR書き込みの最終遅延ステップを初期化、アクティブLow
WMOVE	入力	WMOVEが立ち下がりエッジのときにDDR書き込み遅延ステップを変更、各パルスは1回変更
WSTEP[7:0]	入力	DDR書き込みイコライゼーション遅延制御に使用
READ[3:0]	入力	DDR読み出しモード用のREAD信号

ポート名	I/O	説明
RCLKSEL[2:0]	入力	読み出しクロックソースの選択と極性制御
RESET	入力	DQSリセット入力、アクティブHigh
RPOINT[2:0]	出力	IOLOGICのRADDRに作用する、または配線によりユーザーロジックに作用するFIFO制御読み出しポインター
WPOINT[2:0]	出力	IOLOGICのWADDRに作用する、または配線によりユーザーロジックに作用するFIFO制御書き込みポインター
DQSW0	出力	IOLOGICのTCLKに作用する、または配線によりユーザーロジックに作用するPCLK/FCLK 0° 位相シフト出力。
DQSW270	出力	IOLOGICのTCLKに作用する、または配線によりユーザーロジックに作用するPCLK/FCLK 270° 位相シフト出力。
DQSR90	出力	IOLOGICのICLKに作用する、または配線によりユーザーロジックに作用するDQSI 90° 位相シフト出力。
RFLAG	出力	読み出し遅延調整under-flowまたはover-flowを示すREAD遅延調整出力フラグ
WFLAG	出力	書き込み遅延調整under-flowまたはover-flowを示すWRITE遅延調整出力フラグ
RVALID	出力	READモードのデータ有効フラグ
RBURST	出力	READバースト検出出力

### パラメータの説明

表 5-29 DQS のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
FIFO_MODE_SEL	1'b0 , 1'b1	1'b0	FIFOモード選択 1'b0: DDR memoryモード 1'b1: GDDRモード
RD_PNTR	"000", "001", "010", "011", "100", "101", "110", "111"	3'b000	FIFO読み出しポインター設定
DQS_MODE	"X1", "X2_DDR2", "X2_DDR3", "X4", "X2_DDR3_EXT"	"X1"	DQSモード選択
HWL	"false", "true"	"false"	update0/1タイミング関係制御 "false": update1はupdate0より1サイクル先です。 "true": update1とupdate0のタイミングは同じです。
GSREN	"false", "true"	"false"	グローバルリセットGSRを有効にする

### 接続ルール

- DQSI の入力 DQSI は IO PAD からのものです。
- DQS の出力 RPOINT は、IOLOGIC の RADDR に接続するか、ユーザ

ーロジックに作用できます。

- DQS の出力 WPOINT は、IOLOGIC の WADDR に接続するか、ユーザーロジックに作用できます。
- DQS の出力 DQSR90 は、IOLOGIC の ICLK に接続するか、ユーザーロジックに作用できます。
- DQS の出力 DQSW0/DQSW270 は、IOLOGIC の TCLK に接続するか、ユーザーロジックに作用できます。

#### プリミティブのインスタンス化

##### Verilog でのインスタンス化 :

```
DQS uut (
    .DQSIN(dqs),
    .PCLK(pclk),
    .FCLK(fclk),
    .RESET(reset),
    .READ(read),
    .RCLKSEL(rsel),
    .DLLSTEP(step),
    .WSTEP(wstep),
    .RLOADN(1'b0),
    .RMOVE(1'b0),
    .RDIR(1'b0),
    .WLOADN(1'b0),
    .WMOVE(1'b0),
    .WDIR(1'b0),
    .HOLD(hold),
    .DQSR90(dqsr90),
    .DQSW0(dqsw0),
    .DQSW270(dqsw270),
    .RPOINT(rpoint),
    .WPOINT(wpoint),
    .RVALID(rvalid),
    .RBURST(rburst),
    .RFLAG(rflag),
    .WFLAG(wflag)
);
```

```

defparam uut.DQS_MODE = "X1";
defparam uut.FIFO_MODE_SEL = 1'b0;
defparam uut.RD_PNTR = 3'b001;

```

**VHDL** でのインスタンス化 :

COMPONENT DQS

    GENERIC(

```

        FIFO_MODE_SEL:bit:='0';
        RD_PNTR : bit_vector:="000";
        DQS_MODE:string:="X1";
        HWL:string:="false";
        GSREN : string:="false"

```

    );

    PORT(

```

        DQSIN,PCLK,FCLK,RESET:IN std_logic;
        READ:IN std_logic_vector(3 downto 0);
        RCLKSEL:IN std_logic_vector(2 downto 0);
        DLLSTEP,WSTEP:IN std_logic_vector(7 downto 0);
        RLOADN,RMOVE,RDIR,HOLD:IN std_logic;
        WLOADN,WMOVE,WDIR:IN std_logic;
        DQSR90,DQSW0,DQSW270:OUT std_logic;
        RPOINT, WPOINT:OUT std_logic_vector(2 downto 0);
        RVALID,RBURST,RFLAG,WFLAG:OUT std_logic

```

    );

END COMPONENT;

uut:DQS

    GENERIC MAP(

```

        FIFO_MODE_SEL=>'0',
        RD_PNTR=>"000",
        DQS_MODE=>"X1",
        HWL=>"false",
        GSREN=>"false"

```

    )

    PORT MAP(

```

        DQSIN=>dqsin,
        PCLK=>pclk,

```

```
FCLK=>fclk,  
RESET=>reset,  
READ=>read,  
RCLKSEL=>rclkssel,  
DLLSTEP=>step,  
WSTEP=>wstep,  
RLOADN=>rloadn,  
RMOVE=>rmove,  
RDIR=>rdir,  
HOLD=>hold,  
WLOADN=>wloadn,  
WMOVE=>wmove,  
WDIR=>wdir,  
DQSR90=>dqsr90,  
DQSW0=>dqsw0,  
DQSW270=>dqsw270,  
RPOINT=>rpoint,  
WPOINT=>wpoint,  
RVALID=>rvalid,  
RBURST=>rburst,  
RFLAG=>rflag,  
WFLAG=>wflag  
);
```

# 6 オシレータ

## 6.1 プリミティブの紹介

### 6.1.1 OSC

OSC はオンチップオシレータです。

サポートされるデバイス

表 6-1 OSC 対応デバイス

ファミリ	シリーズ	デバイス
Arora	GW2A	GW2A-18, GW2A-18C, GW2A-55, GW2A-55C
	GW2AN	GW2AN-55C
	GW2AR	GW2AR-18, GW2AR-18C
	GW2ANR	GW2ANR-18C
LittleBee®	GW1N	GW1N-4, GW1N-4B, GW1N-4D, GW1N-9, GW1N-9C
	GW1NR	GW1NR-4, GW1NR-4B, GW1NR-4D, GW1NR-9, GW1NR-9C
	GW1NRF	GW1NRF-4B

#### 機能の説明

GOWIN FPGA に組み込まれたプログラム可能なオンチップオシレータは、プログラミング中に **MSPI** プログラミングモードのクロックソースを提供するほか、ユーザーデザイン用のクロックリソースも提供します。動作パラメータを設定して最大 **64** のクロック周波数を得ることができます。

出力クロック周波数を取得するには、次の式を使用します。

$$f_{CLKOUT} = f_{osc} / \text{FREQ\_DIV} ;$$

このうち、 $f_{osc}$  は **OSC** の発振周波数で、除数 **FREQ\_DIV** は設定されたパラメータ（2～128 の偶数）です。

注記：

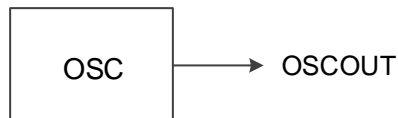
$f_{osc}$  はデバイスにより異なります。GW1N-4, GW1NR-4, GW1N-4B, GW1NR-4B, GW1NRF-4B, GW1N-4D, GW1NR-4D デバイスは 210MHz をサポートし、その他のデ



バースは 250MHz をサポートします。

## ポート図

図 6-1 OSC のポート図



## ポートの説明

表 6-2 OSC のポートの説明

ポート名	I/O	説明
OSCOUT	出力	OSCの出力

## パラメータの説明

表 6-3 OSC のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
FREQ_DIV	2~128(偶数)	100	OSC分周係数の設定
DEVICE	"GW1N-4"、"GW1N-4B"、"GW1N-4D"、 "GW1NR-4"、"GW1NR-4B"、 "GW1NR-4D"、"GW1NRF-4B"、 "GW1N-9"、"GW1N-9C"、 "GW1NR-9"、"GW1NR-9C"、 "GW2A-18"、"GW2AR-18"、 "GW2A-55"、"GW2A-55C"、 "GW2AN-55C"。	GW1N-4(LittleBeeファミリ) GW2A-18(Aroraファミリ)	デバイスの選択

## プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。

### Verilog でのインスタンス化：

```

OSC uut(
    .OSCOUT(oscout)
);
defparam uut.FREQ_DIV=100;
  
```

```

defparam uut.DEVICE="GW2A-18";
VHDL でのインスタンス化 :
COMPONENT OSC
    GENERIC(
        FREQ_DIV:integer:=100;
        DEVICE:string:="GW2A-18"
    );
    PORT(OSCOUT:OUT STD_LOGIC);
END COMPONENT;
uut:OSC
    GENERIC MAP(
        FREQ_DIV=>100,
        DEVICE=>"GW2A-18"
    )
    PORT MAP(OSCOUT=>oscout);

```

### 6.1.2 OSCZ

OSCZ は、OSC を動的にシャットダウンできるオンチップオシレータです。

サポートされるデバイス

表 6-4 OSCZ 対応デバイス

ファミリー	シリーズ	デバイス
LittleBee®	GW1NS	GW1NS-4, GW1NS-4C
	GW1NSR	GW1NSR-4, GW1NSR-4C
	GW1NSER	GW1NSER-4C
	GW1NZ	GW1NZ-1, GW1NZ-1C

#### 機能の説明

GW1NZ などのシリーズの FPGA 製品には、OSC 機能を動的にオン/オフできる、±5%のクロック精度のプログラム可能なオンチップオシレータが備わっています。このオンチップオシレータは、プログラミング中に MSPI プログラミングモードのクロックソースを提供するほか、ユーザーデザイン用のクロックリソースも提供します。動作パラメータを設定して最大 64 のクロック周波数を得ることができます。出力クロック周波数を取得するには、次の式を使用します。

$$f_{CLKOUT} = f_{oscz} / \text{FREQ\_DIV} ;$$

"f<sub>oscz</sub>"は OSCZ の発振周波数で、"FREQ\_DIV"は 2~128 の偶数の分周パラメータです。

注記：

- $f_{oscZ}$  は、デバイスやスピードグレードによって異なります。GW1NS-4/GW1NS-4C/GW1NSR-4/GW1NSR-4C/GW1NSER-4C デバイスの C7 スピードグレードの場合は 260MHz、その他のサポートされているデバイスとスピードグレードの場合は 250MHz です。

ポート図

図 6-2 OSCZ のポート図



ポートの説明

表 6-5 OSCZ のポートの説明

ポート名	I/O	説明
OSCEN	入力	OSCイネーブル信号、アクティブHigh
OSCOUT	出力	OSCの出力

パラメータの説明

表 6-6 OSCZ のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
FREQ_DIV	2~128(偶数)	100	OSC分周係数の設定
S_RATE	"SLOW", "FAST"	"SLOW"	GW1NS-4/GW1NS-4C/GW1NSR-4/GW1NSR-4C/GW1NSER-4CデバイスC7スピードグレードの場合は"FAST"、その他の場合は"SLOW"に設定してください。

プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。

**Verilog** でのインスタンス化：

```

OSCZ uut(
    .OSCOUT(oscout),
    .OSCEN(oscen)
);

defparam uut.FREQ_DIV=100;
defparam uut.S_RATE="SLOW";
  
```

**VHDL** でのインスタンス化 :

```

COMPONENT OSCZ
    GENERIC(
        FREQ_DIV:integer:=100
        S_RATE:string:="SLOW"
    );
    PORT(
        OSCOUT:OUT STD_LOGIC;
        OSCEN:IN std_logic
    );
END COMPONENT;
uut:OSCZ
    GENERIC MAP(
        FREQ_DIV=>100
        S_RATE=>"SLOW"
    )
    PORT MAP(
        OSCOUT=>oscout,
        OSCEN(oscen)
    );

```

### 6.1.3 OSCH

OSCH はオンチップオシレータです。

サポートされるデバイス

表 6-7 OSCH 対応デバイス

ファミリー	シリーズ	デバイス
LittleBee®	GW1N	GW1N-1, GW1N-1S
	GW1NR	GW1NR-1

#### 機能の説明

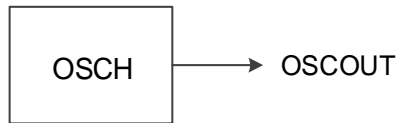
OSCH オンチップオシレータは、プログラミング中に **MSPI** プログラミングモードのクロックソースを提供するほか、ユーザーデザイン用のクロックリソースも提供します。動作パラメータを設定して最大 **64** のクロック周波数を得ることができます。出力クロック周波数を取得するには、次の式を使用します。

$$f_{CLKOUT} = 240MHz / FREQ\_DIV ;$$

"FREQ\_DIV"は 2~128 の偶数のパラメータです。

## ポート図

図 6-3 OSCH のポート図



## ポートの説明

表 6-8 OSCH のポートの説明

ポート名	I/O	説明
OSCOUT	出力	OSCの出力

## パラメータの説明

表 6-9 OSCH のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
FREQ_DIV	2~128(偶数)	100	OSC分周係数の設定

## プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。

**Verilog** でのインスタンス化 :

```

OSCH uut(
    .OSCOUT(oscout)
);
defparam uut.FREQ_DIV=100;
  
```

**VHDL** でのインスタンス化 :

```

COMPONENT OSCH
    GENERIC(
        FREQ_DIV:integer:=100
    );
    PORT(OSCOUT:OUT STD_LOGIC);
END COMPONENT;
uut:OSCH
    GENERIC MAP(
        FREQ_DIV=>100
    )
    PORT MAP(OSCOUT=>oscout);
  
```

## 6.1.4 OSCO

OSCO は、OSC を動的にシャットダウンできるオンチップオシレータであり、Regulator 電源供給をサポートします。

サポートされるデバイス

表 6-10 OSCO 対応デバイス

ファミリー	シリーズ	デバイス
LittleBee®	GW1N	GW1N-2, GW1N-1P5, GW1N-2B, GW1N-1P5B
	GW1NR	GW1NR-2, GW1NR-2B

### 機能の説明

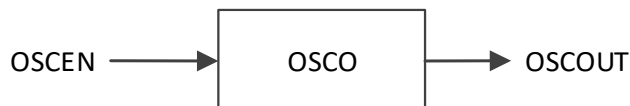
FPGA 製品には、OSC 機能を動的にオン/オフでき、Regulator 電源供給をサポートする、±5%のクロック精度のプログラム可能なオンチップオシレータが備わっています。このオンチップオシレータは、プログラミング中に MSPI プログラミングモードのクロックソースを提供するほか、ユーザーデザイン用のクロックリソースも提供します。動作パラメータを設定して最大 64 のクロック周波数を得ることができます。出力クロック周波数を取得するには、次の式を使用します。

$$f_{CLKOUT} = 250MHz / \text{FREQ\_DIV} ;$$

" FREQ\_DIV"は 2~128 の偶数のパラメータです。

### ポート図

図 6-4 OSCO のポート図



### ポートの説明

表 6-11 OSCO のポートの説明

ポート名	I/O	説明
OSCEN	入力	OSCイネーブル信号、アクティブHigh
OSCOUT	出力	OSCの出力

### パラメータの説明

表 6-12 OSCO のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
FREQ_DIV	2~128(偶数)	100	OSC分周係数の設定
REGULATOR_EN	1'b0, 1'b1	1'b0	1'b0:VCCによる電源供給 1'b1:Regulatorによる電源供給

### プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。

#### Verilog でのインスタンス化 :

```
OSCO uut(  
    .OSCOOUT(oscout),  
    .OSCEN(oscen)  
);  
defparam uut.FREQ_DIV=100;  
defparam uut.REGULATOR_EN =1'b0;
```

#### VHDL でのインスタンス化 :

```
COMPONENT OSCO  
    GENERIC(  
        FREQ_DIV:integer:=100;  
        REGULATOR_EN : bit := '0'  
    );  
    PORT(  
        OSCOUT:OUT STD_LOGIC;  
        OSCEN:IN std_logic  
    );  
END COMPONENT;  
uut:OSCO  
    GENERIC MAP(  
        FREQ_DIV=>100,  
        REGULATOR_EN=> '0'  
    )  
    PORT MAP(  
        OSCOUT=>oscout,  
        OSCEN(oscen)  
    );
```

## 6.1.5 OSCW

OSCW はオンチップオシレータです。

## サポートされるデバイス

表 6-13 OSCW 対応デバイス

ファミリー	シリーズ	デバイス
Arora	GW2AN	GW2AN-18X, GW2AN-9X

## 機能の説明

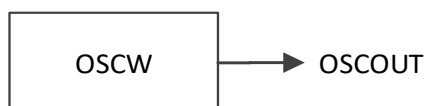
FPGA 製品には、±5%のクロック精度のプログラム可能なオンチップオシレータが備わっています。このオンチップオシレータは、プログラミング中に **MSPI** プログラミングモードのクロックソースを提供するほか、ユーザーデザイン用のクロックリソースも提供します。動作パラメータを設定して最大 **64** のクロック周波数を得ることができます。出力クロック周波数を取得するには、次の式を使用します。

$$f_{CLKOUT} = 200MHz / \text{FREQ\_DIV} ;$$

"FREQ\_DIV"は 2~128 の偶数のパラメータです。

## ポート図

図 6-5 OSCW のポート図



## ポートの説明

表 6-14 OSCW のポートの説明

ポート名	I/O	説明
OSCOUT	出力	OSC の出力

## パラメータの説明

表 6-15 OSCW パラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
FREQ_DIV	2~128(even )	80	OSC 分周係数の設定

## プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。

**Verilog** でのインスタンス化 :

```

OSCW uut(
.OSCOUT(oscout)
);
defparam uut.FREQ_DIV=80;
  
```

**VHDL** でのインスタンス化 :



```
COMPONENT OSCW
    GENERIC(
        FREQ_DIV:integer:=100
    );
    PORT(
        OSCOUT:OUT STD_LOGIC
    );
END COMPONENT;

uut:OSCW
    GENERIC MAP(
        FREQ_DIV=>80
    )
    PORT MAP(
        OSCOUT=>oscout
    );
```

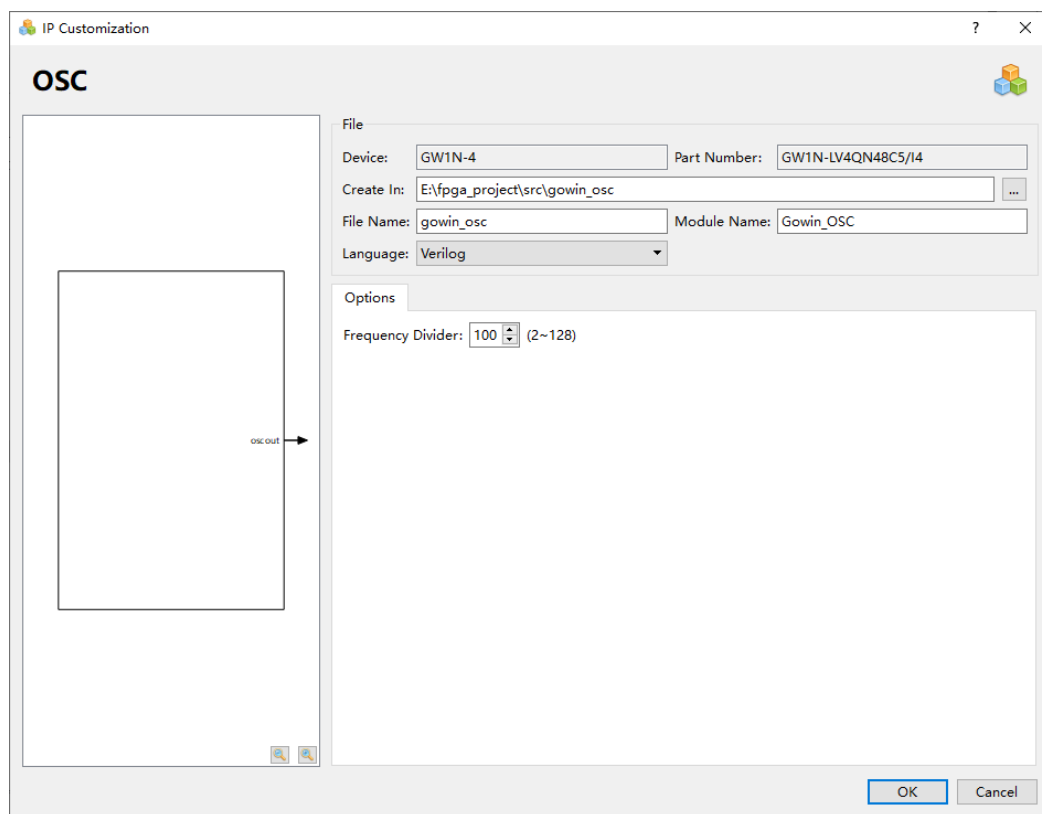
## 6.2 IPの呼び出し

IP Core Generator のインターフェースで **OSC** をクリックすると、右側に **OSC** の概要が表示されます。

### IP の構成

IP Core Generator インターフェースで"**OSC**"をダブルクリックすると、"**IP Customization**"ウィンドウがポップアップします。このウィンドウには **File** 構成タブ、**Options** 構成タブ、およびポート図があります (図 6-6)。

図 6-6 OSC IP の構成ウィンドウ



### 1. File 構成タブ

File 構成タブは、IP ファイルの構成に使用されます。OSC の File 構成タブの使用は DQCE モジュールと同様です。DQCE の File 構成タブを参照してください。

### 2. Options 構成タブ

Options 構成タブは IP のカスタマイズに使用されます(図 6-6)。

- Frequency Divider : 分周値。この値の範囲は 2~128 の偶数です。

### 3. ポート図

ポート図は、IP Core の構成結果を表示します(図 6-6)。

### 生成されるファイル

IP の構成が完了したら、構成ファイルの"File Name"によって命名された 3 つのファイルが生成されます：

- "gowin\_osc.v"は完全な verilog モジュールです。
- gowin\_osc\_tmp.v は IP のテンプレートファイルです。
- gowin\_osc.ipc は IP の構成ファイルです。

注記：

VHDL が設計の言語として選択されている場合、生成される最初の 2 つのファイル名のサフィックスは.vhd になります。

