



Gowin Clock ユーザーガイド

UG286-1.9.8J, 2024-02-02

著作権について(2024)

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

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

免責事項

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

バージョン履歴

日付	バージョン	説明
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"> ● rPLL の CLKOUTD3 のタイミングを追加。 ● PLLO の説明を更新。 ● GW1NR-2 および GW2AN-55C デバイスのサポートを追加。
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"> ● デバイス(GW1N-2B、GW1N-1P5、GW1N-1P5B、GW1NR-2B、GW2AN-18X、GW2AN-9X)のサポートを追加。 ● IP 呼び出しの図面を更新し、Help 内容を削除。
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"> ● PLL IP コアを構成するときの各モードの紹介を更新。 ● CLKDIVG の入力ソースの説明を更新。
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"> ● PLL モジュールの説明を更新。 ● OSCZ 対応デバイスを更新。
2022/08/10	1.9.5J	一部のデバイスのバージョン情報を更新。
2022/11/01	1.9.6J	GW1NS-2、GW1NS-2C、GW1NSE-2C、GW1NSR-2、および GW1NSR-2C デバイスを削除。
2023/09/12	1.9.7J	<ul style="list-style-type: none"> ● IP 呼び出しの一部の図面を更新、Device Version オプションを追加。 ● PLLO のデューティサイクルの微調整に関する内容を更新。
2024/02/02	1.9.8J	セクション「5.4.1 プリミティブの紹介」を更新、DLLDLY プリミティブのステップサイズの情報を追加。

目次

目次.....	i
図一覧.....	iii
表一覧.....	v
1 本マニュアルについて	1
1.1 マニュアルの内容.....	1
1.2 関連ドキュメント.....	1
1.3 用語、略語.....	2
1.4 テクニカル・サポートとフィードバック.....	2
2 概要.....	3
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
3 グローバルクロック	10
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
4 高速クロック	19
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
5 システムクロック	30
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	41
5.2 PLLVR	45
5.2.1 プリミティブの紹介	45
5.2.2 IP の呼び出し	53
5.3 PLLO	54
5.3.1 プリミティブの紹介	54
5.3.2 IP の呼び出し	73
5.4 DLLDLY	79
5.4.1 プリミティブの紹介	79
5.4.2 IP の呼び出し	82
5.5 CLKDIV	84
5.5.1 プリミティブの紹介	84
5.5.2 IP の呼び出し	86
5.6 CLKDIVG.....	88
5.6.1 プリミティブの紹介	88
5.6.2 IP の呼び出し	90
5.7 DQS.....	92
5.7.1 プリミティブの紹介	92
6 オシレータ	98
6.1 プリミティブの紹介	98
6.1.1 OSC.....	98
6.1.2 OSCZ	100
6.1.3 OSCH	102
6.1.4 OSCO.....	104
6.1.5 OSCW	106
6.2 IP の呼び出し	107

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 のタイミング図	32
図 5-3 入力ソースが CLKOUTP の場合の CLKOUTD3 のタイミング図.....	32
図 5-4 rPLL IP の構成ウィンドウ.....	41
図 5-5 PLLVR のポート図	46
図 5-6 PLLVR IP の構成ウィンドウ	53
図 5-7 PLLO のポート図.....	55
図 5-8 チャンネル B のデューティサイクルの微調整タイミング図(微調整方向が 1'b1、ステップ数が 1)	66

図 5-9 チャンネル B のデューティサイクルの微調整タイミング図(微調整方向が 1'b0、ステップ数が 1)	66
図 5-10 PLL0 IP の構成ウィンドウ	74
図 5-11 DLLDLY のポート図	80
図 5-12 DLLDLY IP の構成ウィンドウ	83
図 5-13 CLKDIV のポート図	84
図 5-14 CLKDIV IP の構成ウィンドウ	87
図 5-15 CLKDIVG のポート図	88
図 5-16 CLKDIVG IP の構成ウィンドウ	91
図 5-17 DQS のポート図	92
図 6-1 OSC のポート図	99
図 6-2 OSCZ のポート図	101
図 6-3 OSCH のポート図	103
図 6-4 OSCO のポート図	104
図 6-5 OSCW のポート図	106
図 6-6 OSC IP の構成ウィンドウ	108

表一覧

表 1-1 用語、略語	2
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	35
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	36
表 5-8 rPLL の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.....	47

表 5-13 PLL0 対応デバイス.....	54
表 5-14 PLL0 のポートの説明.....	55
表 5-15 PLL0 のパラメータの説明.....	57
表 5-16 IDSEL ポートのパラメータ参照テーブル.....	62
表 5-17 FBDSEL ポートのパラメータ参照テーブル.....	63
表 5-18 ODSELX(X=A/B/C/D)ポートのパラメータ参照テーブル.....	63
表 5-19 PLL0 のデューティサイクルの微調整の参照テーブル.....	65
表 5-20 DLLDLY のポートの説明.....	80
表 5-21 DLLDLY のパラメータの説明.....	80
表 5-22 CLKDIV のポートの説明.....	84
表 5-23 CLKDIV のパラメータの説明.....	85
表 5-24 CLKDIVG 対応デバイス.....	88
表 5-25 CLKDIVG のポートの説明.....	89
表 5-26 CLKDIVG のパラメータの説明.....	89
表 5-27 DQS 対応デバイス.....	92
表 5-28 DQS のポートの説明.....	93
表 5-29 DQS のパラメータの説明.....	94
表 6-1 OSC 対応デバイス.....	98
表 6-2 OSC のポートの説明.....	99
表 6-3 OSC のパラメータの説明.....	99
表 6-4 OSCZ 対応デバイス.....	100
表 6-5 OSCZ のポートの説明.....	101
表 6-6 OSCZ のパラメータの説明.....	101
表 6-7 OSCH 対応デバイス.....	102
表 6-8 OSCH のポートの説明.....	103
表 6-9 OSCH のパラメータの説明.....	103
表 6-10 OSCO 対応デバイス.....	104
表 6-11 OSCO のポートの説明.....	104
表 6-12 OSCO のパラメータの説明.....	105
表 6-13 OSCW 対応デバイス.....	106
表 6-14 OSCO のポートの説明.....	106
表 6-15 OSCW のパラメータの説明.....	106

1 本マニュアルについて

1.1 マニュアルの内容

本マニュアルでは、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ホームページ 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	コンフィギュラブル・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	コンフィギュラブル配線ユニット
DCC	Duty Cycle Correction	デューティサイクル補正モジュール
DCS	Dynamic Clock Selector	ダイナミック・クロック・セレクタ
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
LW	Long Wire	ロングワイヤ
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

1.4 テクニカル・サポートとフィードバック

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

ホームページ : www.gowinsemi.com/ja

E-mail : support@gowinsemi.com

2 概要

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(PCLK および 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ると同時に、論理配線として、つまり通常のリソース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

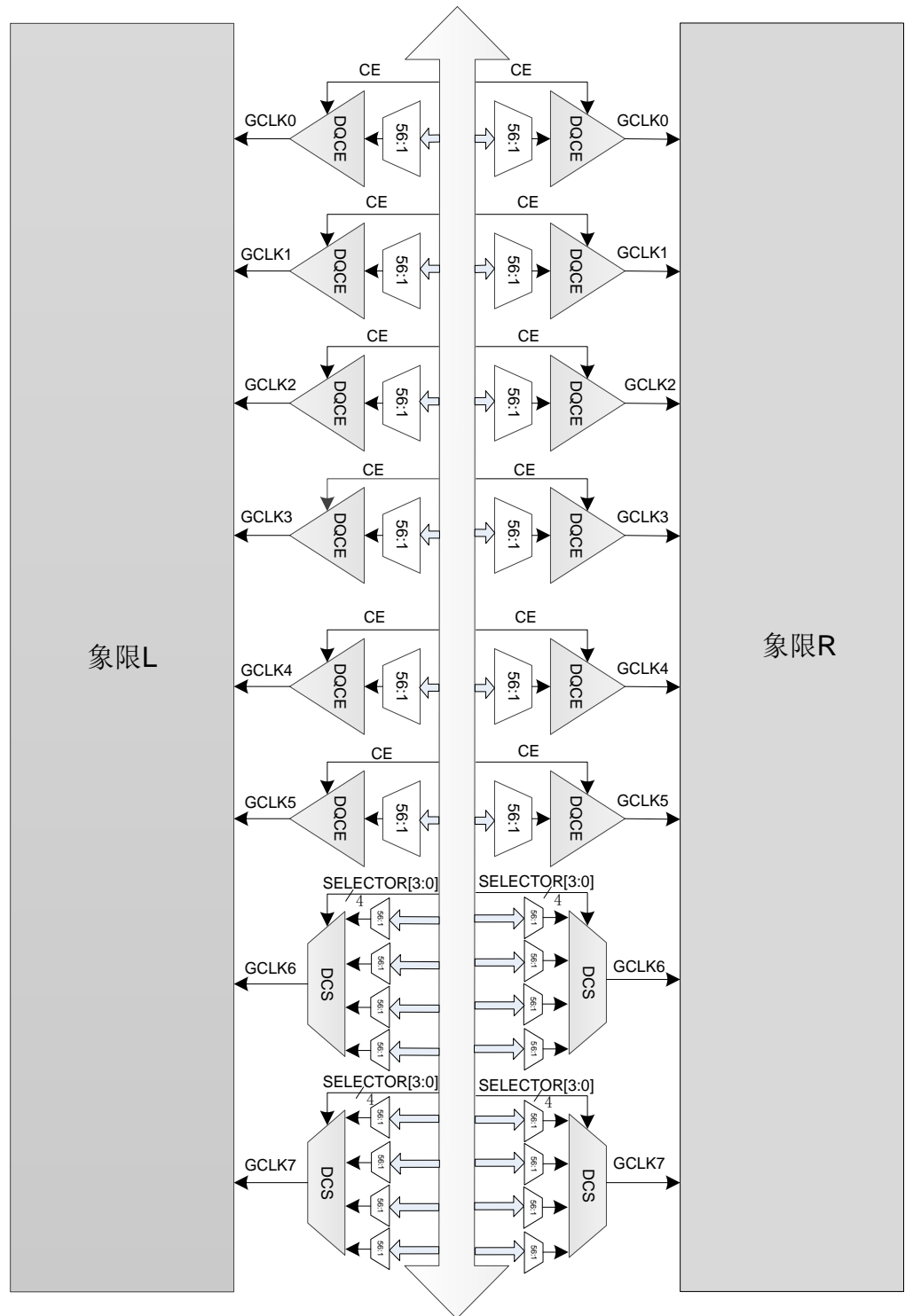
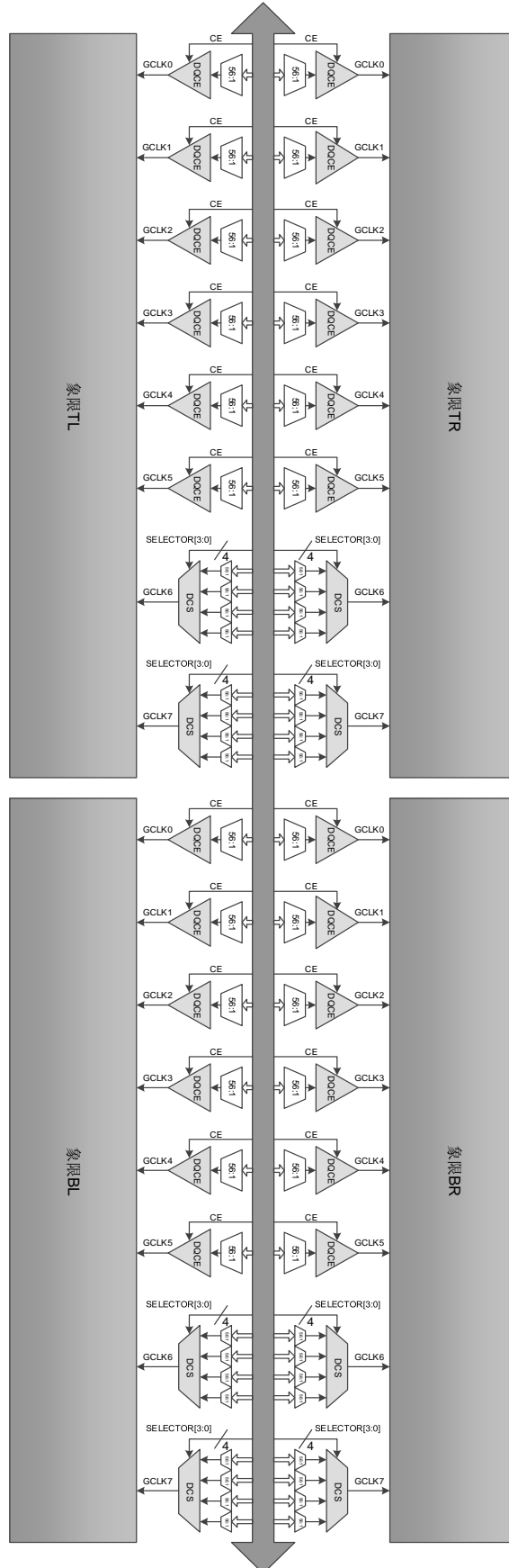


図 2-2 LittleBee®ファミリーの 9K FPGA 製品および Arora ファミリーの FPGA 製品での GCLK の配置



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 の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インターフェースに適しています。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基づいてループ内部の発振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。

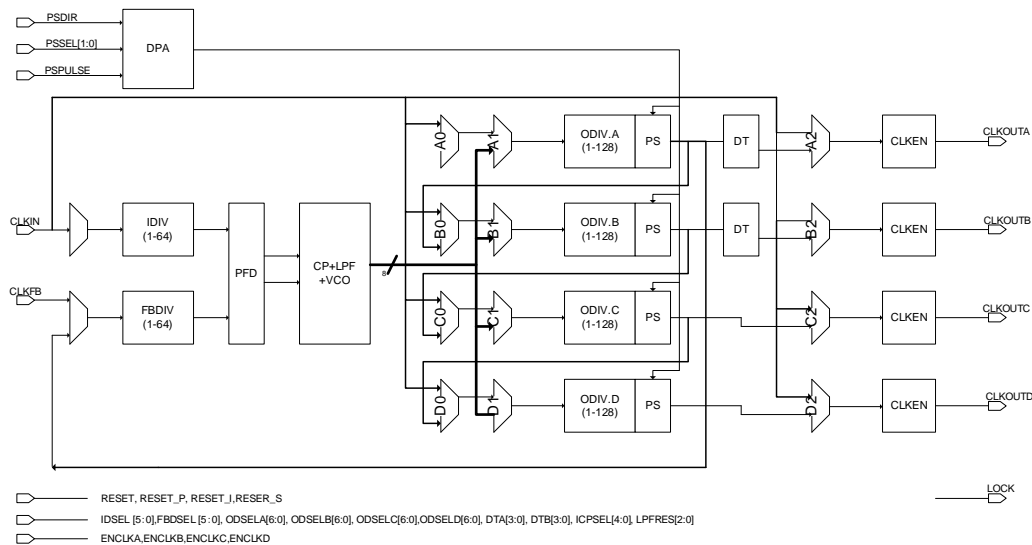
2.3.1 GW1N-1P5 / GW1N-2 / GW1NR-2 / GW2AN-18X / GW2AN-9X

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 全部リセット
RESET_S	入力	B/C/D の 3 つのチャネルのリセット
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
PSDIR	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	出力	クロック出力イネーブルの動的制御

ポート名	信号	説明
ENCLKC ENCLKD		
CLKOUTA	出力	チャンネル A のクロック出力(デフォルト)
CLKOUTB	出力	チャンネル B のクロック出力(デフォルト)
CLKOUTC	出力	チャンネル C のクロック出力(デフォルト)
CLKOUTD	出力	チャンネル D のクロック出力(デフォルト)
LOCK	出力	PLL のロック状態を示します。 1 : ロック 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、または通常のリファレンス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、または通常のリファレンス信号により提供されます。

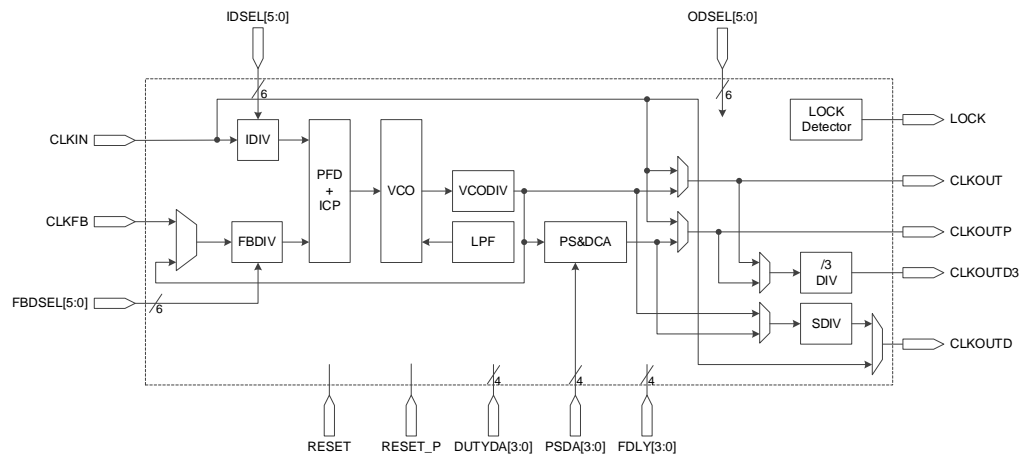
2.3.2 LittleBee®ファミリーと Arora ファミリーのその他のデバイス

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

ポート名	信号	説明
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
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
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 の分周クロック(SDIV により制御)
CLKOUTD3	出力	CLKOUT または CLKOUTP の分周クロック(DIV3 により制御され、分周値は 3)
LOCK	出力	PLL のロック状態を示します。 1 : ロック 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、または通常のリファレンス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、または通常のリファレンス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

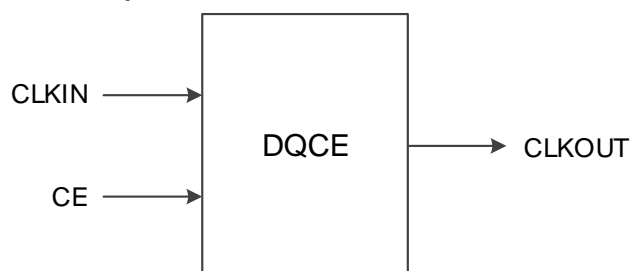
FPGA デバイスには、象限内の PCLK ネットワークを内部ロジックにより動的に有効または無効にできる動的クロック制御機能が備わっています。さらに、DQCE はパラメータの構成により無効にすることができ、この場合、PCLK ネットワークは常に有効です。PCLK クロックネットワークが無効にされると、そのクロックによって駆動されるすべてのロジックは反転しなくなるため、デバイスの全体的な消費電力が低下します。

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
);
```

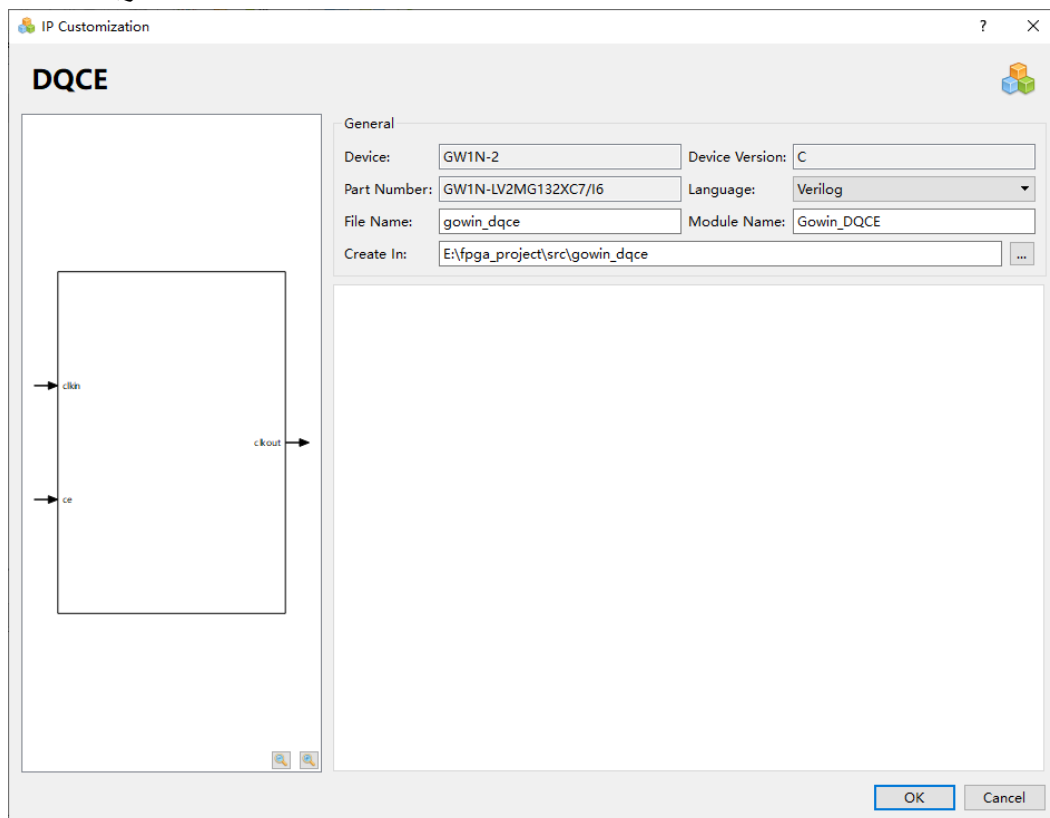
3.1.2 IP の呼び出し

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、General 構成タブおよび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. General 構成タブ

General 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。
- **Device Version** : デバイスのバージョン。
- **Part Number** : 部品番号。
- **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。同じである場合、エラーが報告されません。
- **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。

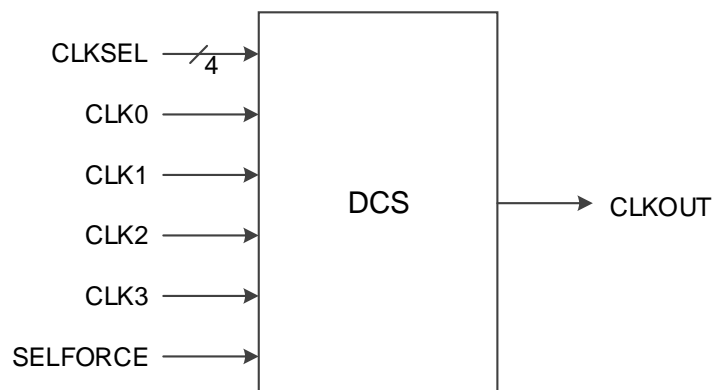
DCS には、「Non-Glitchless」モードと「Glitchless」モードの 2 つのクロック切り替えモードがあります。

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
SELFORCE	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]),  
    .SELFORCE(selfforce),  
    .CLKOUT(clkout)  
);  
defparam dcs_inst.DCS_MODE="RISING";
```

VHDL でのインスタンス化 :

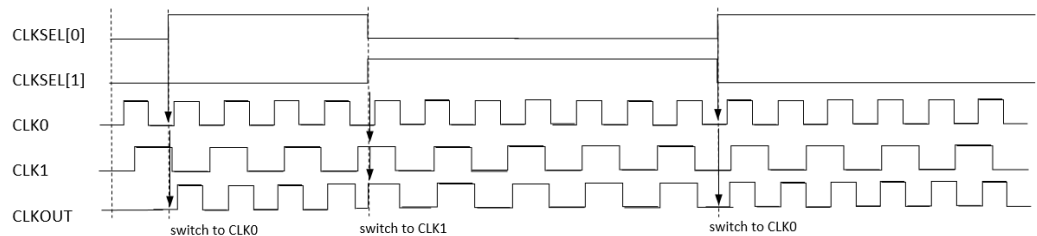
```
COMPONENT DCS  
    GENERIC(DCS_MODE:string:="RISING");  
    PORT(  
        CLK0:IN std_logic;  
        CLK1:IN std_logic;  
        CLK2:IN std_logic;  
        CLK3:IN std_logic;  
        CLKSEL:IN std_logic_vector(3 downto 0);  
        SELFORCE:IN std_logic;  
        CLKOUT:OUT std_logic  
    );  
END COMPONENT;  
 uut:DCS  
     GENERIC MAP(DCS_MODE=>"RISING")  
     PORT MAP(  
         CLK0=>clk0,  
         CLK1=>clk1,  
         CLK2=>clk2,  
         CLK3=>clk3,  
         CLKSEL=>clksel,  
         SELFORCE=>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

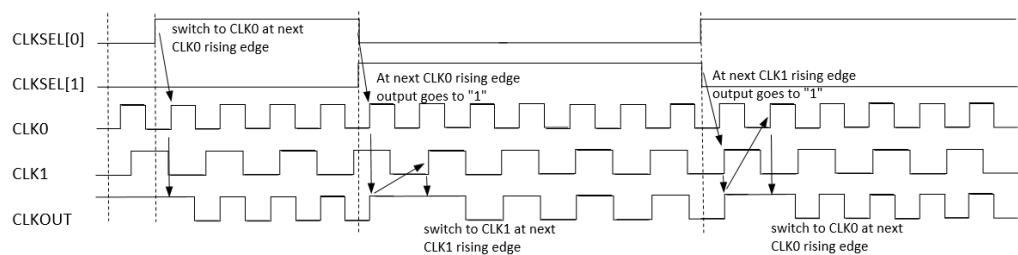


図 3-6 DCS mode が FALLING の場合のタイミング

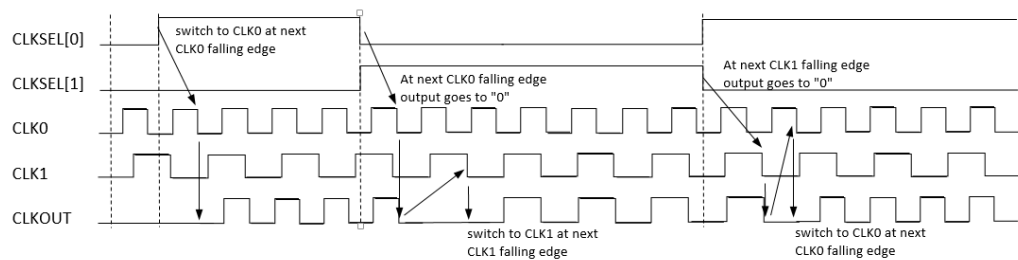


図 3-7 DCS mode が CLK0_GND の場合のタイミング

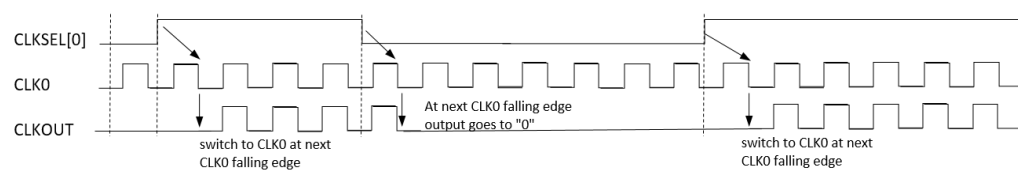
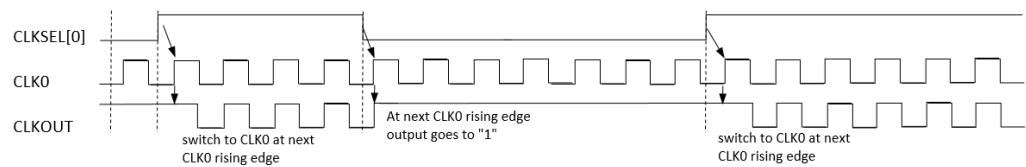


図 3-8 DCS mode が CLK0_VCC の場合のタイミング



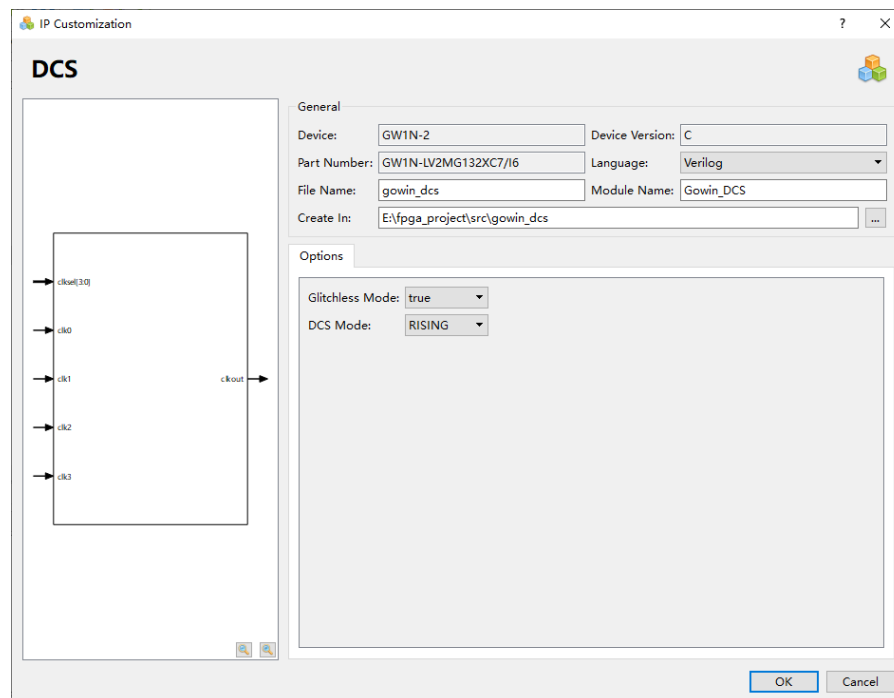
3.2.2 IP の呼び出し

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、**General** 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. General 構成タブ

General 構成タブは、IP ファイルの構成に使用されます。DCS の General 構成タブの使用は **DQCE** モジュールと同様であるので、**DQCE** の General 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 のときにオンです。

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
);
```

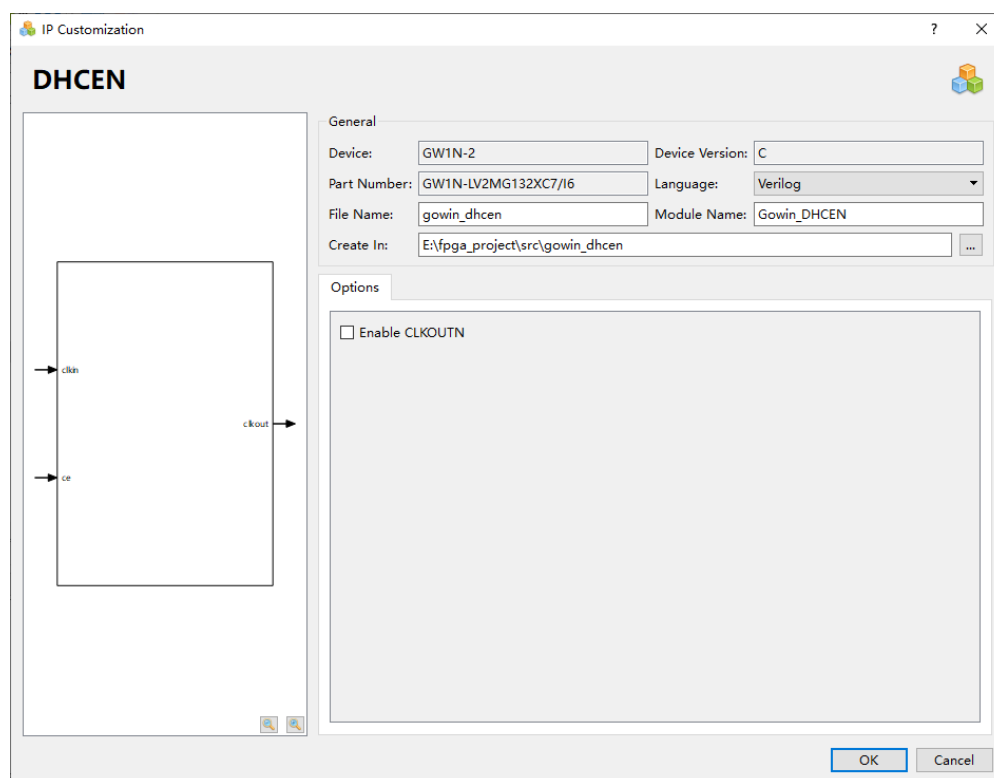
4.1.2 IP の呼び出し

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、General 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. General 構成タブ

General 構成タブは、IP ファイルの構成に使用されます。DHCEN の General 構成タブの使用は DQCE モジュールと同様であるので、DQCE の General 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せん。

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_dhcen.v" は完全な verilog モジュールです。
- "gowin_dhcen_tmp.v" は IP のテンプレートファイルです。
- "gowin_dhcen.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 のときにオンです。

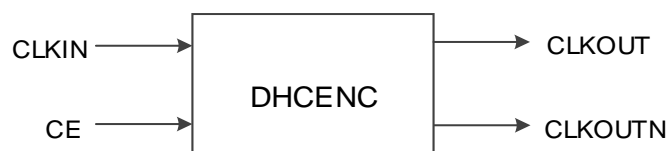
サポートされるデバイス

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
CLKOUTN	出力	クロック出力信号。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 と同様に、[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。

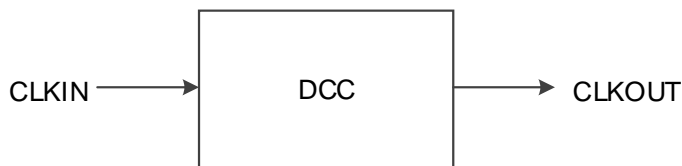
サポートされるデバイス

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	クロック入力信号
CLKOUT	出力	クロック出力信号

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 します 1' b0 : DCC を無効に します
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。

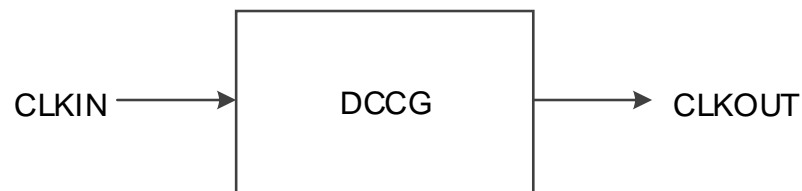
サポートされるデバイス

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	クロック入力信号
CLKOUT	出力	クロック出力信号

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

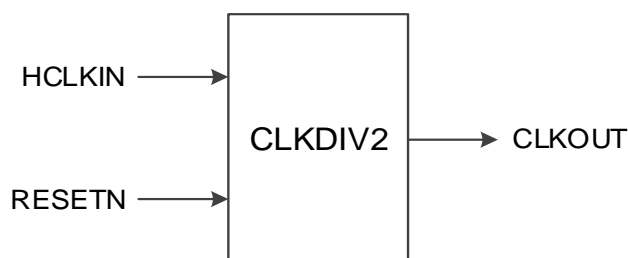
CLKDIV2 は、クロック周波数を 2 分周するクロック分周器です。CLKDIV2 の出力は、DCC/DCCG の CLKIN、IOLOGIC の FCLK、PLL の CLKIN と CLKFB、DQS の FCLK、および CLKDIV の HCLKIN のみを駆動できます。

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	クロック入力信号
RESETN	入力	非同期リセット信号、アクティブ Low。
CLKOUT	出力	クロック出力信号

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 :

```
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  
    );
```

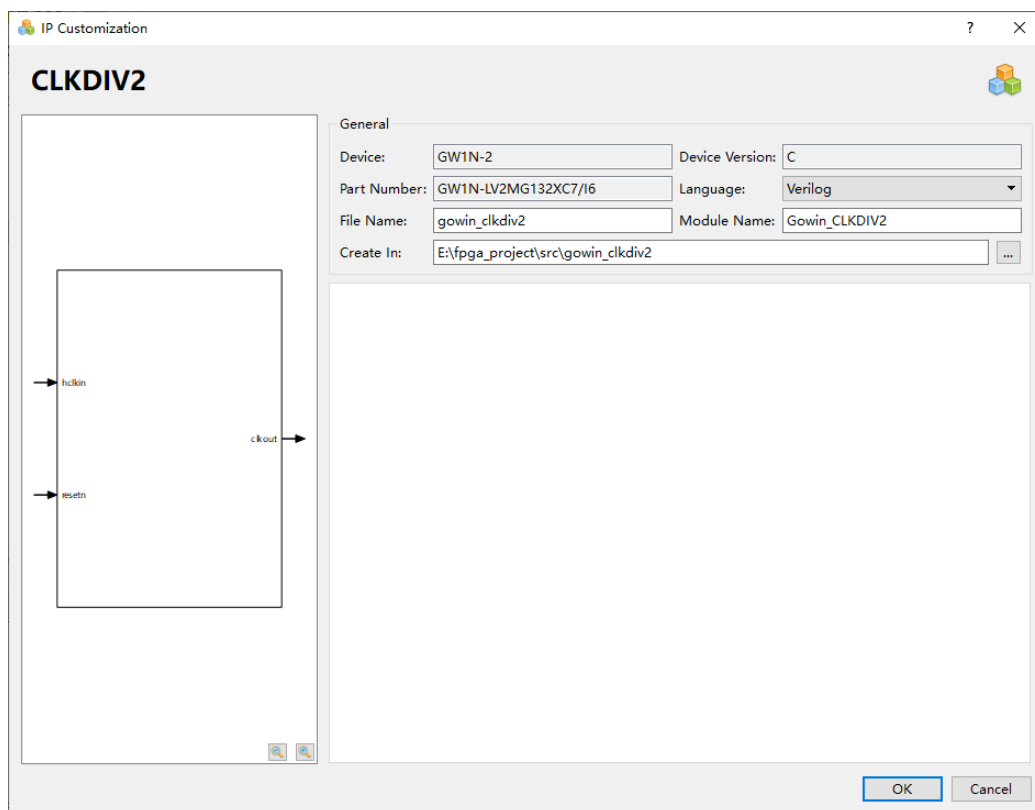
4.5.2 IP の呼び出し

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、**General** 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. General 構成タブ

General 構成タブは、IP ファイルの構成に使用されます。CLKDIV2 の General 構成タブの使用は DQCE モジュールと同様であるので、DQCE の General 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基づいてループ内部の発振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) に対して周波数調整(逡倍及び分周)を行うことができます。その計算式は以下の通りです：

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

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

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

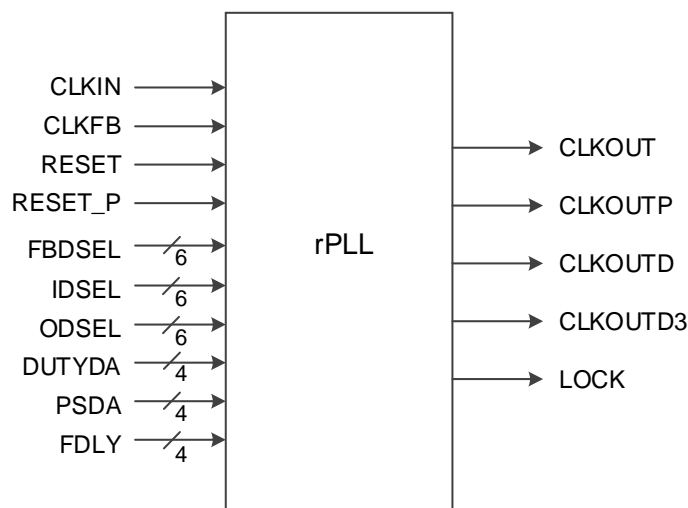
$$f_{PFD} = f_{CLKIN} / IDIV = f_{CLKOUT} / FBDIV$$

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信号
CLKFB	入力	フィードバッククロック入力信号
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 の出力は 0 です。
FBDSEL[5:0]	入力	FBDIV の動的制御、範囲は 0~63、実際の値は 64 - FBDSEL
IDSEL[5:0]	入力	IDIV の動的制御、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 クロック出力信号
LOCK	出力	rPLL ロック指示。1 はロック、0 はロック解除を示す

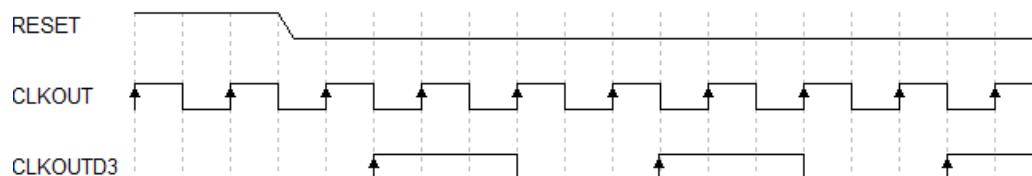
ポート名	I/O	説明
CLKOUTP	出力	rPLL の、CLKOUT に基づいた位相とデューティサイクル調整ありのクロック出力信号
CLKOUTD	出力	rPLL の、SDIV を経たクロック出力信号。CLKOUT または CLKOUTP の、SDIV を経た出力信号
CLKOUTD3	出力	rPLL の、DIV3 を経たクロック出力信号。CLKOUT または CLKOUTP の、DIV3 を経た出力信号

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、リセットが解放された後、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。

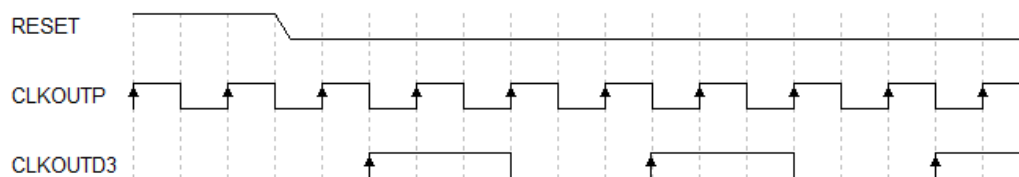
図 5-2 入力ソースが CLKOUT の場合の CLKOUTD3 のタイミング図



- CLKOUTD3 の入力ソースが CLKOUTP の場合：

図 5-3 に示すように、リセットが解放された後、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 のタイミング図



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

パラメータ名	値の範囲	デフォルト値	説明
			false: 静的制御、つまりパラメータ IDIV_SEL を選択します true: 動的制御、つまり IDSEL を選択します
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ータまたは動的制御信号の選択 false: 静的制御、つまりパラメータ FBDIV_SEL を選択します true: 動的制御、つまり FBDSEL を選択します
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 を選択します true: 動的制御、つまり ODSEL を選択します
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します false: 静的制御 true: 動的制御
CLKOUT_FT_DIR	1'b1	1'b1	CLKOUT の微調整方向の設定 1'b1: 減算
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: 減算
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 の偶数	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

パラメータ名	値の範囲	デフォルト値	説明
CLKOUTD3_SRC	"CLKOUT", "CLKOUTP"	"CLKOUT"	CLKOUTD3 ソースの選択
CLKOUT_BYPASS	"true","false"	"false"	rPLL をバイパス。CLKOUT は CLKIN から直接取得されます true: rPLL をバイパス。CLKIN は CLKOUT に直接作用します false: ノーマルモード
CLKOUTP_BYPASS	"true","false"	"false"	rPLL をバイパス。CLKOUTP は CLKIN から直接取得されます true: rPLL をバイパス。CLKIN は CLKOUTP に直接作用します false: ノーマルモード
CLKOUTD_BYPASS	"true","false"	"false"	rPLL をバイパス。CLKOUTD は CLKIN から直接取得されます true: rPLL をバイパス。CLKIN は CLKOUTD に直接作用します 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", "GW2AN-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 の実際値
111111	0	1
111110	1	2
111101	2	3
111100	3	4
111011	4	5
111010	5	6

IDSEL[5:0]	IDIV の静的パラメータ値	IDIV の実際値
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 5-6 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 の設定	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°
1001	202.5°
1010	225°
1011	247.5°
1100	270°
1101	292.5°
1110	315°
1111	337.5°

表 5-8 rPLL の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)です。

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)は位相オフセットの設定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
);

```

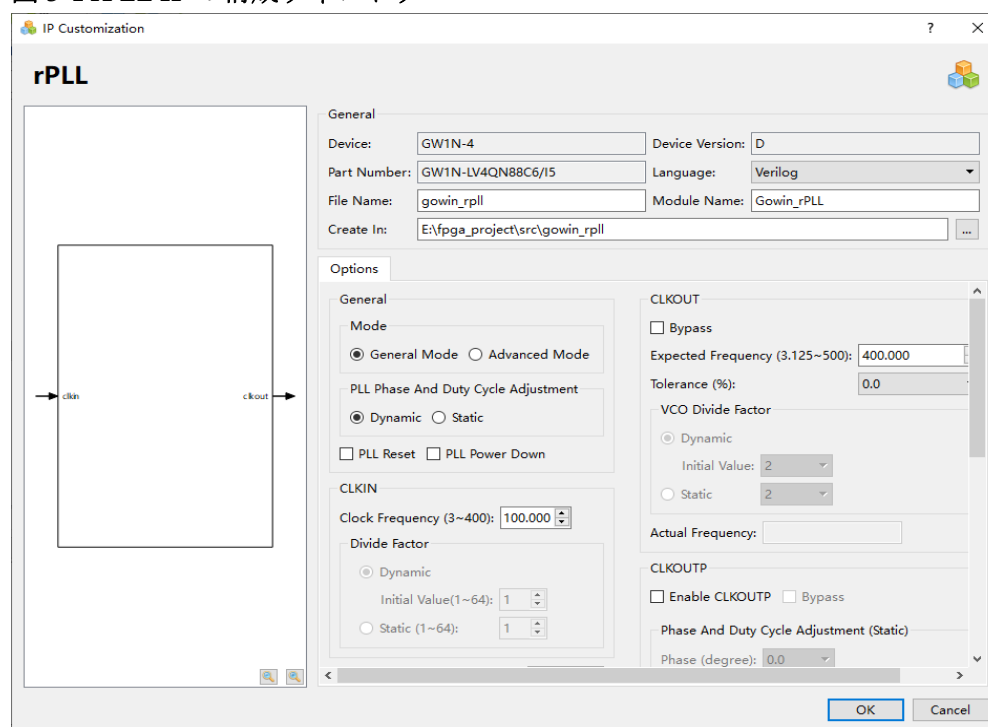
5.1.2 IP の呼び出し

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 General 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. General 構成タブ

General 構成タブは、IP ファイルの構成に使用されます。rPLL の General 構成タブの使用は DQCE モジュールと同様であるので、DQCE の General 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ます。一般モード **"General Mode"**とアドバンスモード**"Advanced Mode"**がサポートされます。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ます。動的調整 **"Dynamic"**および静的調整**"Static"**がサポートされます。
 - **"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するために使用され、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数値(範囲は 1~64)を構成できます。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するために使用され、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数値(範囲は 1~64)を構成できます。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(%)"** : **CLKOUT** の目的の周波数と算出された実際の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** パラメータを構成するために使用され、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数値(範囲は **2/4/8/16/32/48/64/80/96/112/128**)を構成できます。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** : 計算された **CLKOUT** の実際の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された実際の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報告されます。
- "Actual Frequency" : 計算された分周クロックの実際の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、構成された分周パラメータ、周波数逡倍パラメータ、および VCO パラメータに基づいて計算します。計算された実際の周波数が目的の周波数と同じでない場合、"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、不当な位置が赤でマークされます。
 - 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。構成が正しい場合は、"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、IP Core の構成結果を表示します。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基づいてループ内部の発振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(通倍と分周)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)に対して周波数調整(通倍及び分周)を行うことができます。その計算式は以下の通りです：

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

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

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

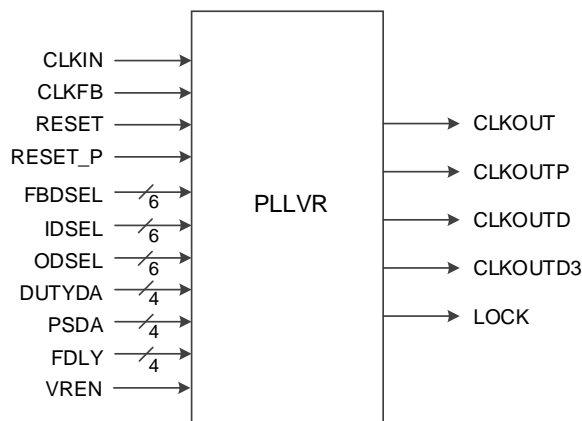
$$f_{PPFD} = f_{CLKIN} / IDIV = f_{CLKOUT} / FBDIV$$

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_{PPFD} 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信号
CLKFB	入力	フィードバッククロック入力信号
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 の出力は 0 です。
FBDSEL[5:0]	入力	FBDIV の動的制御、範囲は 0~63、実際の値は 64 - FBDSEL
IDSEL[5:0]	入力	IDIV の動的制御、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 クロック出力信号
LOCK	出力	PLLVR ロック指示。1 はロック、0 はロック解除を示します
CLKOUTP	出力	位相とデューティサイクル調整ありの PLLVR クロック出力信号
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 の 3 分周された出力クロック信号です。CLKOUTD3 と CLKOUT または CLKOUTP のタイミング関係については、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、つまりパラメータ IDIV_SEL を選択します true: 動的制御、つまり IDSEL を選択します
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ータまたは動的制御信号の選択 false: 静的制御、つまりパラメータ FBDIV_SEL を選択します true: 動的制御、つまり FBDSEL を選択します
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 を選択します true: 動的制御、つまり ODSEL を選択します
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します

パラメータ名	値の範囲	デフォルト値	説明
			false : 静的制御 true : 動的制御
CLKOUT_FT_DIR	1'b1	1'b1	CLKOUT の微調整方向の設定 1'b1: 減算
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: 減算
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 の偶数	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 から直接取得されます true : PLLVR をバイパス。CLKIN は CLKOUT に直接作用します false : ノーマルモード
CLKOUTP_BYPASS	"true", "false"	"false"	PLLVR をバイパス。 CLKOUTP は CLKIN から直接取得されます true : PLLVR をバイパス。CLKIN は CLKOUTP に直接作用します false : ノーマルモード
CLKOUTD_BYPASS	"true", "false"	"false"	PLLVR をバイパス。 CLKOUTD は CLKIN から直接取得されます

パラメータ名	値の範囲	デフォルト値	説明
			true: PLLVR をバイパス。CLKIN は CLKOUTD に直接作用します 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 と同様であるため、rPLL を参照してください。

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：

```

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,
    FBDSEL=>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
```

```
);
```

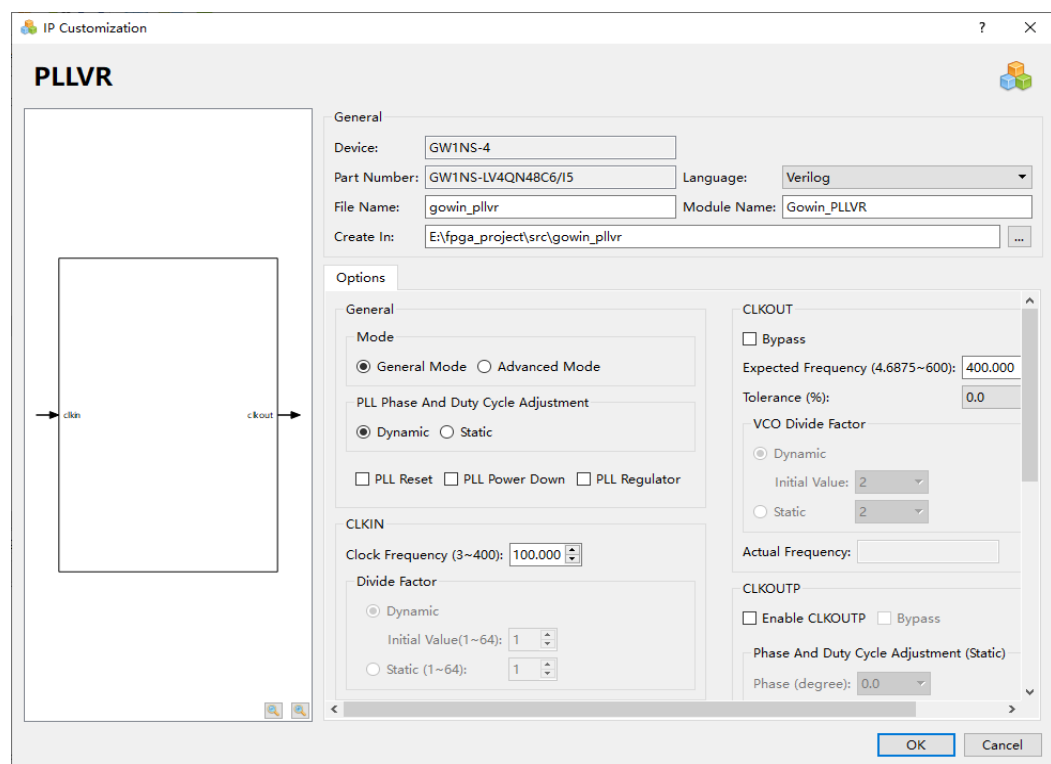
5.2.2 IP の呼び出し

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、General 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. General 構成タブ

General 構成タブは、IP ファイルの構成に使用されます。PLLVR の General 構成タブの使用は DQCE モジュールと同様であるので、DQCE の General 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、IP Core の構成結果を表示します。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 5-6)。

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 FPGA は、4 つのクロック出力をサポートする位相同期回路 PLLO を提供します。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することができます。

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 つのクロック出力をサポートします。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(通倍と分周)を実行することにより、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。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)に対して周波数調整(通倍及び分周)を行うことができます。その計算式は以下の通りです：

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

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

$$f_{CLKOUTX} = f_{IN_ODIVX} / ODIVX$$

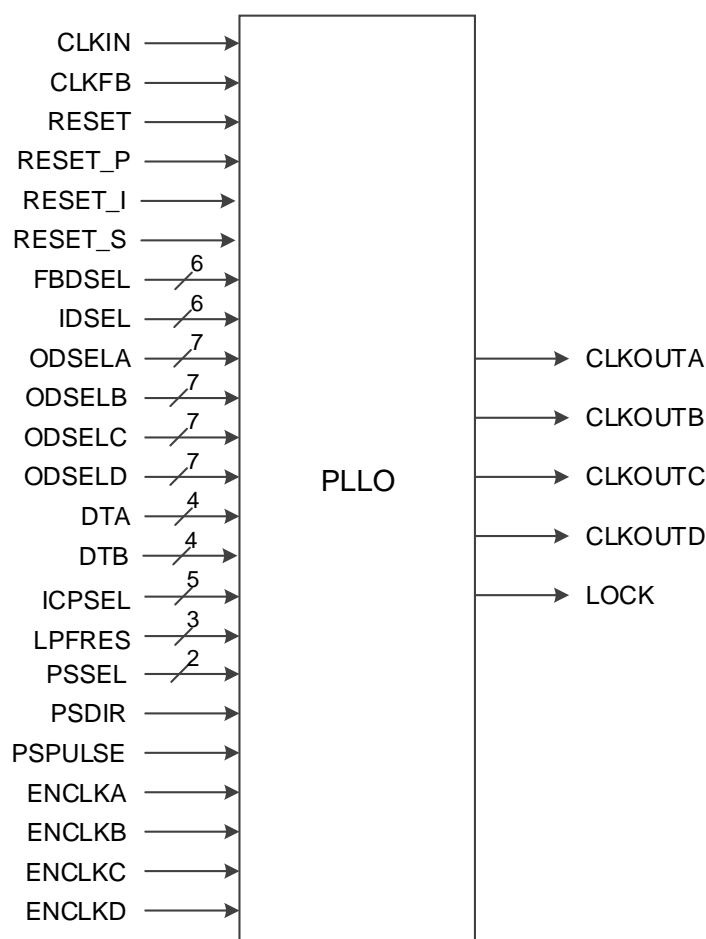
$$f_{PFD} = f_{CLKIN} / IDIV = f_{CLKOUTA} / FBDIV$$

注記：

- f_{CLKIN} は CLKIN の周波数です。
- $f_{CLKOUTX}$ はチャンネル A/B/C/D の出力クロックの周波数です($x=A/B/C/D$)。ODIVX はチャンネル A/B/C/D 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信号
CLKFB	入力	フィードバッククロック入力信号

ポート名	I/O	説明
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 リセット機能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、範囲は 0~63、実際の値は 64 - FBDSEL
IDSEL[5:0]	入力	IDIV の動的制御、範囲は 0~63、実際の値は 64 - IDSEL
ODSELA[6:0]	入力	ODIVA の動的制御、範囲は 0~127、実際の値は 128 - ODSELA
ODSELB[6:0]	入力	ODIVB の動的制御、範囲は 0~127、実際の値は 128 - ODSELB
ODSELC[6:0]	入力	ODIVC の動的制御、範囲は 0~127、実際の値は 128 - ODSELC
ODSELD[6:0]	入力	ODIVD の動的制御、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。この値の増加とともに電流が増加します。
LPFRES[2:0]	入力	LPFRES の値の動的制御。範囲は R0~R7。R0 は最大の帯域幅に対応し、R7 は最小の帯域幅に対応します。
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"にする必要があります。
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"にする必要があります。
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"にする必要があります。
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"にする必要があります。
CLKOUTA	出力	チャンネル A のクロック出力

ポート名	I/O	説明
CLKOUTB	出力	チャンネル B のクロック出力
CLKOUTC	出力	チャンネル C のクロック出力
CLKOUTD	出力	チャンネル D のクロック出力
LOCK	出力	PLL ロック指示。1 はロック、0 はロック解除を示します

パラメータの説明

表 5-15 PLL0 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。実際の 1~64 に対応します。
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、つまりパラメータ IDIV_SEL を選択します TRUE: 動的制御、つまり IDSEL を 選択します
FBDIV_SEL	0~63	0	FBDIV 分周係数の静的設定。実際の 1~64 に対応します。
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、つまりパラメータ FBDIV_SEL を選択します TRUE: 動的制御、つまり FBDSEL を 選択します
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 を選択します TRUE: 動的制御、つまり ODSELA を選択します
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 を選択します TRUE: 動的制御、つまり ODSELB を選択します
ODIVC_SEL	1~128	4	ODIVC 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_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 を選択します TRUE: 動的制御、つまり ODSELC を選択を選択します
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 を選択します TRUE: 動的制御、つまり ODSELD を選択します
CLKOUTA_EN	"TRUE", "FALSE"	"TRUE"	チャンネル A のクロック出力イネーブル
CLKOUTB_EN	"TRUE", "FALSE"	"TRUE"	チャンネル B のクロック出力イネーブル
CLKOUTC_EN	"TRUE", "FALSE"	"TRUE"	チャンネル C のクロック出力イネーブル
CLKOUTD_EN	"TRUE", "FALSE"	"TRUE"	チャンネル D のクロック出力イネーブル
DYN_DTA_SEL	"TRUE", "FALSE"	"FALSE"	チャンネル A 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 を選択します TRUE: 動的制御、つまり DTA を選択します
DYN_DTB_SEL	"TRUE", "FALSE"	"FALSE"	チャンネル B 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 を選択します TRUE: 動的制御、つまり DTB を選択します
CLKOUTA_DT_DIR	1'b1, 1'b0	1'b1	チャンネル A 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します 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します

パラメータ名	値の範囲	デフォルト値	説明
CLKOUTB_DT_DIR	1'b1, 1'b0	1'b1	チャンネル B 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します 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します
CLKOUTA_DT_STEP	0,1,2,4	0	チャンネル A デューティサイクル静的微調整のステップ数。ステップごとに 50ps。
CLKOUTB_DT_STEP	0,1,2,4	0	チャンネル B デューティサイクル静的微調整のステップ数。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 の出力 2'b11: CLKIN(バイパス)
CLKA_OUT_SEL	1'b0, 1'b1	1'b0	チャンネル A 出力クロックソースの選択 1'b0: ODIVA の出力 1'b1: CLKIN(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 の出力 2'b10: CLKCAS_A(カスケード) 2'b11: CLKIN(バイパス)
CLKB_OUT_SEL	1'b0, 1'b1	1'b0	チャンネル B 出力クロックソースの選択 1' b0 : ODIVB の出力 1' b1 : CLKIN(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 の出力 2' b10 : CLKCAS_B(カスケード) 2' b11 : CLKIN(バイパス)
CLKC_OUT_SEL	1'b0, 1'b1	1'b0	チャンネル C 出力クロックソースの選択 1'b0: ODIVC の出力 1'b1: CLKIN(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 の出力 2'b10: CLKCAS_C(カスケード) 2'b11: CLKIN(バイパス)
CLKD_OUT_SEL	1'b0, 1'b1	1'b0	チャンネル D 出力クロックソースの選択 1'b0: ODIVD の出力

パラメータ名	値の範囲	デフォルト値	説明
			1'b1: CLKIN(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"	チャンネル B 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 を選択します "TRUE" : 動的制御、つまり DPA 動的信号(PSSEL、PSDIR、および PSPULSE)を選択します。また、DYN_DPA_EN="TRUE"にする必要があります。
DYN_PSC_SEL	"TRUE", "FALSE"	"FALSE"	チャンネル C 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 を選択します 動的制御、つまり DPA 動的信号(PSSEL、PSDIR、および PSPULSE)を選択します。また、DYN_DPA_EN="TRUE"にする必要があります。
DYN_PSD_SEL	"TRUE", "FALSE"	"FALSE"	チャンネル D 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 を選択します "TRUE" : 動的制御、つまり DPA 動的信号(PSSEL、PSDIR、および PSPULSE)を選択します。また、DYN_DPA_EN="TRUE"にする必要があります。
PSB_COARSE	1~128	1	チャンネル B 位相シフト粗調整の静的設定
PSB_FINE	0~7	0	チャンネル B 位相シフト微調整の静的設定
PSC_COARSE	1~128	1	チャンネル C 位相シフト粗調整の静的設定
PSC_FINE	0~7	0	チャンネル C 位相シフト微調整の静的設定

パラメータ名	値の範囲	デフォルト値	説明
PSD_COARSE	1~128	1	チャンネル D 位相シフト粗調整の静的設定
PSD_FINE	0~7	0	チャンネル D 位相シフト微調整の静的設定
DTMS_ENB	"TRUE", "FALSE"	"FALSE"	チャンネル B(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、動的位相調整を立ち上がりエッジとして設定することで、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)を実現します。
DTMS_ENC	"TRUE", "FALSE"	"FALSE"	チャンネル C(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、動的位相調整を立ち上がりエッジとして設定することで、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)を実現します。
DTMS_END	"TRUE", "FALSE"	"FALSE"	チャンネル D(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、動的位相調整を立ち上がりエッジとして設定することで、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)を実現します。
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ります。
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ります。
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

パラメータ名	値の範囲	デフォルト値	説明
			FALSE: 静的制御、つまりパラメータ ICP_SEL を選択します TRUE: 動的制御、つまり動的信号 ICPSEL を選択します
ICP_SEL	5'bXXXXX, 5'b00000 ~5'b11111	5'bXXXXX	ICP 電流の静的設定 5'bXXXXXX: 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します 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できます。
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 を選択します TRUE: 動的制御、つまり動的信号 LPFRES を選択します
LPR_REF	7'bXXXXXX, 7'b000000(R0),7'b0000001(R1),7'b0000010(R2),7'b0000100(R3),7'b0001000(R4),7'b0010000(R5),7'b0100000(R6),7'b1000000(R7)	7'bXXXXXX	LPRRES の静的設定 7'bXXXXXXX: 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します 7'b0000000~7'b1000000(8つの値あり): ユーザーは、必要に応じてこの8つの値から選択できます。

表 5-16 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

IDSEL[5:0]	IDIV の静的パラメータ値	IDIV の実際値
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 の静的パラメータ値	FBDIV の実際値
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 の静的パラメータ値	ODIVX の実際値
1111111	1	1
1111110	2	2
1111101	3	3
1111100	4	4
1111011	5	5
1111010	6	6
1111001	7	7
1111000	8	8
1110111	9	9
.....
0000000	128	128

位相の調整

PLL0 は、静的位相調整と動的位相調整をサポートします。動的位相調

整はチャンネル B/C/D 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=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 はチャン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 パルスの立ち下がりエッジごとに、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 $\leq ODIV$ です。

位相調整は、次の式で計算できます(例としてチャンネル B を使用)。

COARSE_B < ODIVB の場合、 $ps = (FINE_B/8 + COARSE_B)/ODIVB * 360$

COARSE_B = ODIVB の場合、 $ps = (FINE_B/8)/ODIVB * 360$

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の調整

PLL0 の動的デューティサイクル調整は、チャンネル B/C/D 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(チャンネル B)。

$$\text{DUTY} = (\text{PSB_FINE}/8 + \text{PSB_COARSE})$$

$$\text{PHASE} = (\text{DYN_FINEB}/8 + \text{DYN_COARSEB})$$

動的デューティサイクルの計算：

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

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)内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で 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 (High の期間 < Low の期間、すなわち 50%未満)。

デューティサイクルの微調整

PLL0 のチャンネル A/B 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数の設定によって実現されます。静的モードと動的モードがサポートされます。

1) 静的 :

- a) 微調整の方向 : パラメータ

CLKOUTA_DT_DIR/CLKOUTB_DT_DIR により制御されます。

- b) 微調整のステップ数 : パラメータ

CLKOUTA_DT_STEP/CLKOUTB_DT_STEP で制御されます。

2) 動的 :

- a) 微調整の方向 : ポート DTA[3]/DTB[3]により制御されます。

- b) 微調整のステップ数 : ポート 7-DTA[2:0]/7-DTB[2:0]により制御されます。

微調整方向が 1'b1 の場合、立ち下がりエッジの遅延が調整されてデューティサイクルが増加します。微調整方向が 1'b0 の場合、立ち上がりエッジの遅延が調整されてデューティサイクルが減少します。

表 5-19 PLL0 のデューティサイクルの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	デューティサイクルの微調整ステップ数	デューティサイクルの微調整の遅延値
1'b0	0	0
	1	-50ps
	2	-100ps
	4	-200ps
1'b1	0	0
	1	+50ps
	2	+100ps
	4	+200ps

例えば、同じ周波数のクロックを出力するチャンネル A とチャンネル B がある場合、チャンネル B のクロックのデューティサイクルの微調整タイミングは、図 5-8 と図 5-9 に示すとおりです(チャンネル A は基準として使用される)。

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

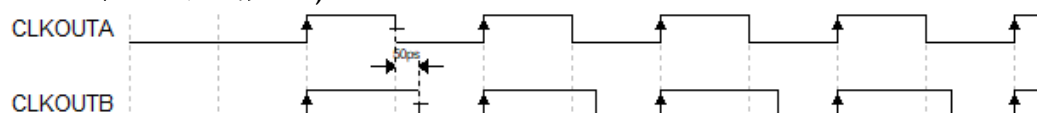
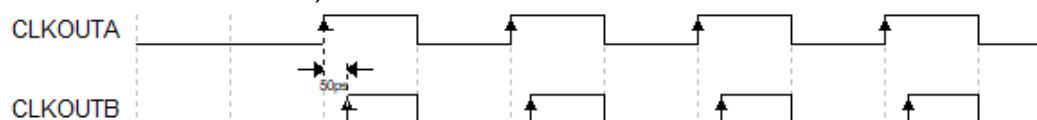


図 5-9 チャンネル B のデューティサイクルの微調整タイミング図(微調整方向が 1' b0、ステップ数が 1)



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。動的設定の場合、ユーザーは、必要に応じて設定できます。静的設定の場合、デフォルトの X は、ソフトウェアが自動的に計算して構成することを示し、また、手動でこのパラメータを設定することも可能です。

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。

LPFRES 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(psdir),
.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) := "XXXXXXXX"
    );
    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,FBDSSEL : 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_DT_B_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 => "XXXXX";
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
```

```
);
```

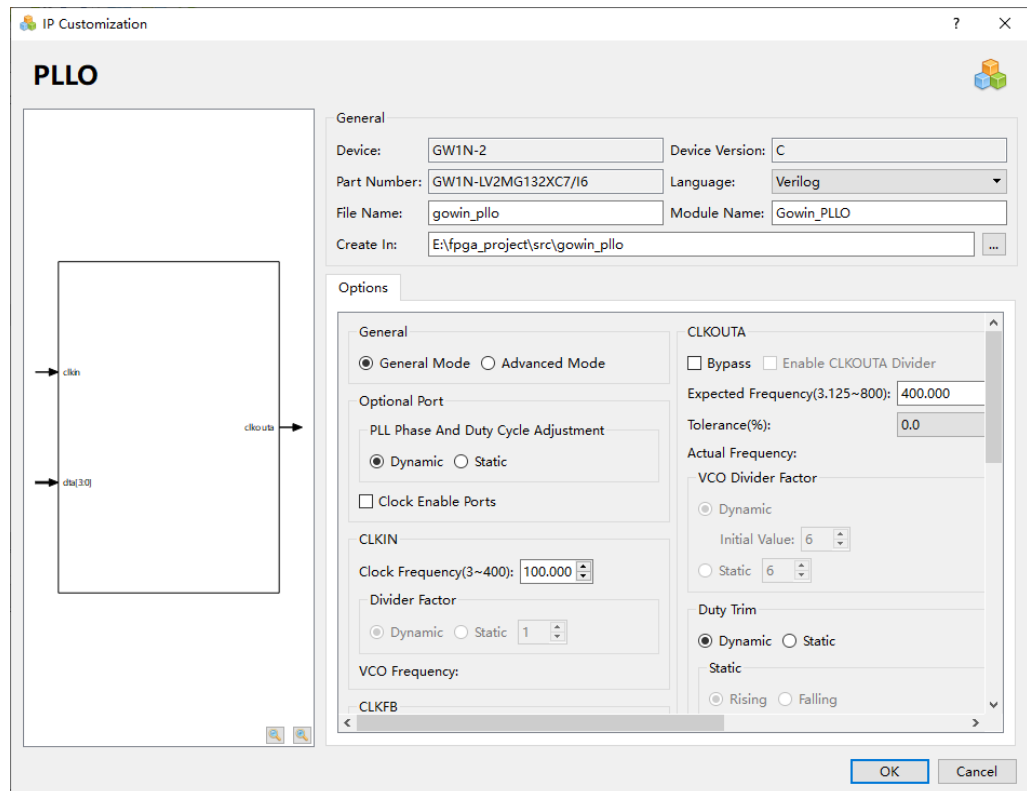
5.3.2 IP の呼び出し

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 General 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 PLL0 IP の構成ウィンドウ



1. General 構成タブ

General 構成タブは、IP ファイルの構成に使用されます。PLLVO の General 構成タブの使用は DQCE モジュールと同様であるので、DQCE の General 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ます。一般モード "General Mode"とアドバンスモード"Advanced Mode"がサポートされます。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。
- 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、PLL0 出力クロックのイネーブルの構成を実行します。
 - "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ます。動的調整 "Dynamic"および静的調整"Static"がサポートされます。
 - "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ます。その範囲は **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するために使用され、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数値(範囲は **1~64**)を構成できます。**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逡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逡倍パラメータを構成するために使用され、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逡倍パラメータの数値(範囲は **1~64**)を構成できます。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** 電流を構成するために使用され、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** の値を **ICP1~ICP32** の範囲で指定でき、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するために使用され、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** の値を **R0~R7** の範囲で指定でき、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 PLL0 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 : PLL0 のチャンネル A 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 3.125MHz~800MHz です。
 - "Tolerance(%)" : CLKOUTA の目的の周波数と算出された実際の周波数の許容誤差を構成します。
 - "Actual Frequency" : 算出された CLKOUTA の実際の出力周波数を表示します。
 - "VCO Divide Factor" : アドバンスモードで VCO パラメータを構成するために使用され、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数値(範囲は 1~128)を構成できます。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するために使用され、動的モード"Dynamic"と静的モード"Static"をサポートします。静的モードの場合は"Rising"と"Falling"があり、"Step"の値(0、1、2、4)を構成できます。
- CLKOUTB : PLL0 のチャンネル B 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 3.125MHz~800MHz です。
 - "Tolerance(%)" : CLKOUTB の目的の周波数と算出された実際の周波数の許容誤差を構成します。
 - "Actual Frequency" : 算出された CLKOUTB の実際の出力周波数を表示します。
 - "VCO Divide Factor" : アドバンスモードで VCO パラメータを構成するために使用され、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数値(範囲は 1~128)を構成できます。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するために使用され、動的モード"Dynamic"と静的モード"Static"をサポートします。静的モードの場合は"Rising"と"Falling"があり、"Step"の値(0、1、2、4)を構成できます。
 - "Phase(degree)" : 位相(度)の調整を構成するために使用され、動的モード"Dynamic"および静的モード"Static"をサポートします。静的モードでは位相(度)を指定できます。
 - "DutyCycle" : デューティサイクルを構成するために使用され、動的モード"Dynamic"と静的モード"Static"をサポートします。静的モードの場合は 50%です。デューティサイクルの動的調整は、位相の調整と動的 DPA 調整によって実現されません。
- CLKOUTC : PLL0 のチャンネル C 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 3.125MHz~800MHz です。
 - "Tolerance(%)" : CLKOUTB の目的の周波数と算出された実際の周波数の許容誤差を構成します。
 - "Actual Frequency" : 算出された CLKOUTC の実際の出力周波

数を表示します。

- "VCO Divide Factor" : アドバンスモードで VCO パラメータを構成するために使用され、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数値(範囲は 1~128)を構成できます。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)の調整を構成するために使用され、動的モード"Dynamic"および静的モード"Static"をサポートします。静的モードでは位相(度)を指定できます。
 - "DutyCycle" : デューティサイクルを構成するために使用され、動的モード"Dynamic"と静的モード"Static"をサポートします。静的モードの場合は 50%です。デューティサイクルの動的調整は、位相の調整と動的 DPA 調整によって実現されません。
- CLKOUTD : PLL0 のチャンネル D 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 3.125MHz~800MHz です。
 - "Tolerance(%)" : CLKOUTD の目的の周波数と算出された実際の周波数の許容誤差を構成します。
 - "Actual Frequency" : 算出された CLKOUTD の実際の出力周波数を表示します。
 - "VCO Divide Factor" : アドバンスモードで VCO パラメータを構成するために使用され、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数値(範囲は 1~128)を構成できます。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)の調整を構成するために使用され、動的モード"Dynamic"および静的モード"Static"をサポートします。静的モードでは位相(度)を指定できます。
 - "DutyCycle" : デューティサイクルを構成するために使用され、動的モード"Dynamic"と静的モード"Static"をサポートし

ます。静的モードの場合は 50% です。デューティサイクルの動的調整は、位相の調整と動的 DPA 調整によって実現されま

- **Calculate** : 現在の構成が適切か計算します。
 - 一般モード"**General Mode**"で、構成された分周パラメータ、周波数通倍パラメータ、および **VCO** パラメータが適切か計算します。算出された実際の周波数が目的の周波数と同じでない場合、"**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、周波数通倍パラメータ、**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。構成が正しい場合は、"**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、**IP Core** の構成結果を表示します。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 5-10)。

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

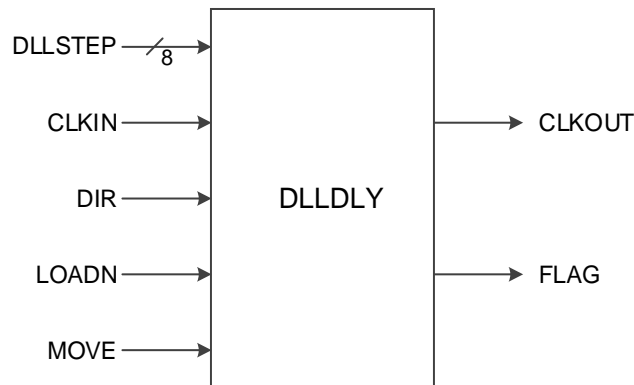
クロック遅延モジュールとして、**DLLDLY** は **DLLSTEP** 信号に基づきクロックを遅延して出力します。

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。**DLLSTEP** の遅延ステップサイズは、約 30ps (**GW1N(X)** **FPGA**)または約 18ps (**GW2A(X)** **FPGA**)です。

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	クロック出力信号
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。
DLLSTEP[7:0]	入力	遅延ステップ入力
CLKIN	入力	クロック入力信号
DIR	入力	遅延の動的調整の方向を設定します 0: 遅延を増やします 1: 遅延を減らします
LOADN	入力	遅延ステップのロードの制御 0: 遅延ステップをロードします。 1: 遅延を動的に調整します。
MOVE	入力	MOVE の立ち下がりエッジで遅延を動的に調整し、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します。
DLY_SIGN	String	1'b0,1'b1	1'b0	遅延調整の符号を設定します : 1'b0: '+' 1'b1: '-'
DLY_ADJ	Integer	0~255	0	遅延調整を設定します :

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
				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
    );
```

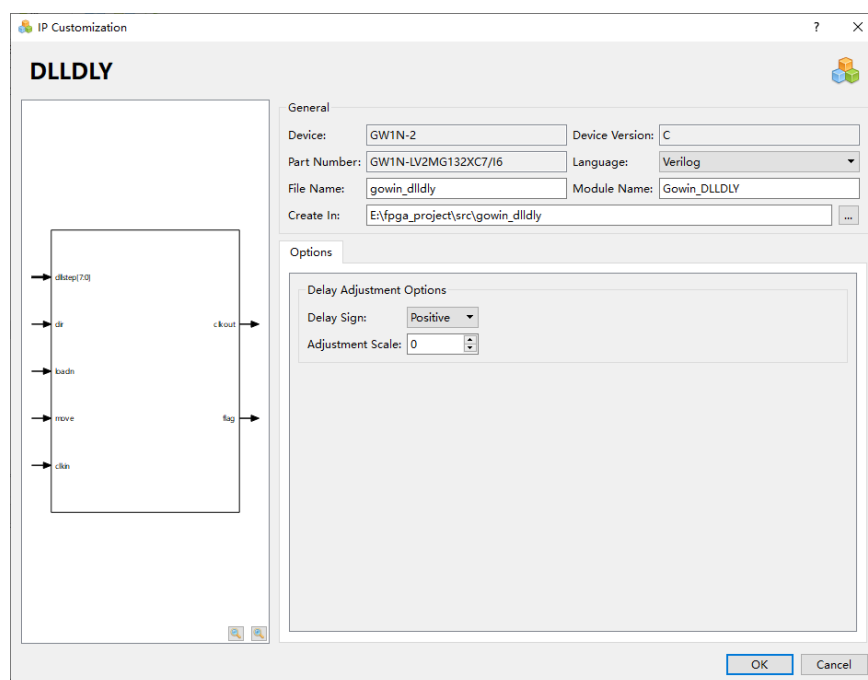
5.4.2 IP の呼び出し

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、General 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. General 構成タブ

General 構成タブは、IP ファイルの構成に使用されます。DLLDLY の General 構成タブの使用は DQCE モジュールと同様であるので、DQCE の General 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 : 遅延調整の符号を設定します。
- 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_dllily.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

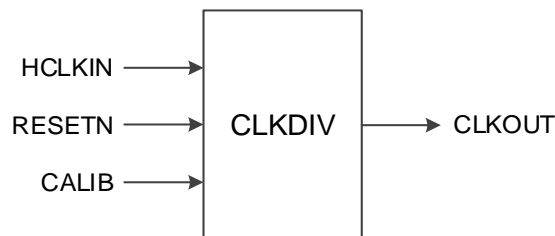
CLKDIV は、クロック周波数を調整するクロック分周器です。

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	クロック入力信号
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します。
CLKOUT	出力	クロック出力信号

CALIB 信号は、IOLOGIC の CALIB と併用することができます。その機能は次のとおりです。

- 2 分周の場合、2 立ち下がりエッジごとに 180 度の位相を調整し、2 回の調整で 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約 102.8 度の位相を調整し、7 回の調整で 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 90 度の位相を調整し、4 回の調整で 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約72度の位相を調整し、5回の調整で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約45度の位相を調整し、8回の調整で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	クロックの分周係数を設定します。
GSREN	"false", "true"	"false"	グローバルリセット GSR を有効にします

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 :

```
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  
    );
```

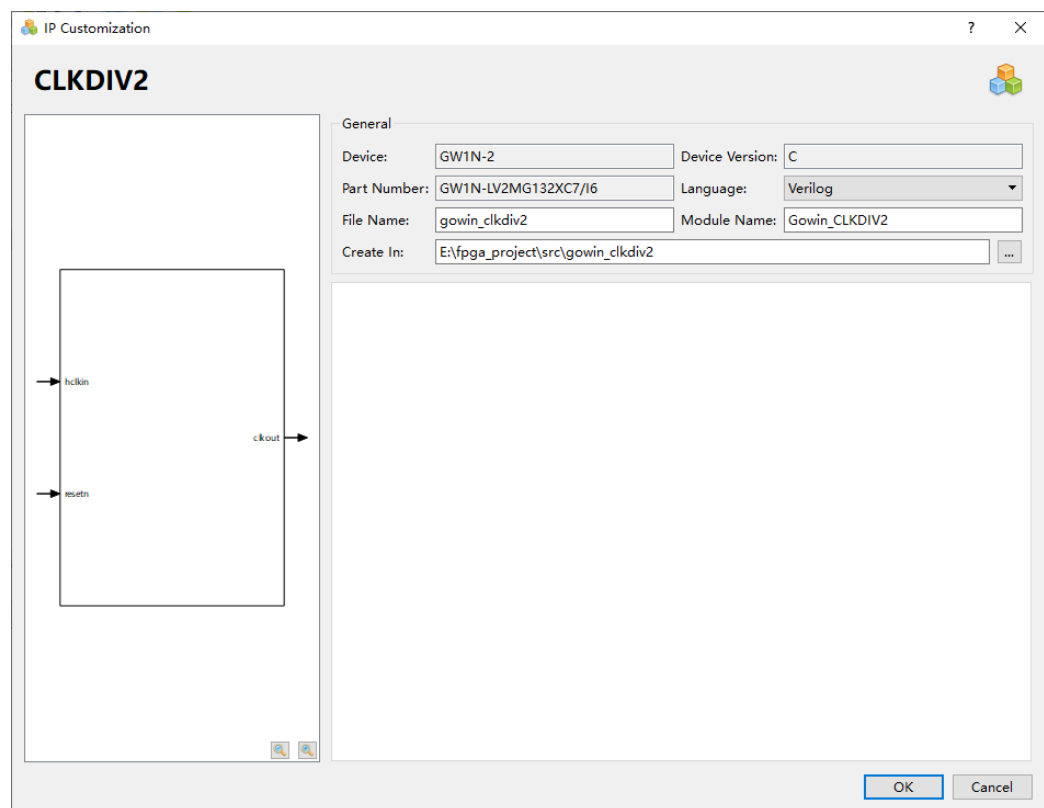
5.5.2 IP の呼び出し

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、General 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. General 構成タブ
General 構成タブは、IP ファイルの構成に使用されます。CLKDIV の General 構成タブの使用は DQCE モジュールと同様であるので、DQCE の General 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

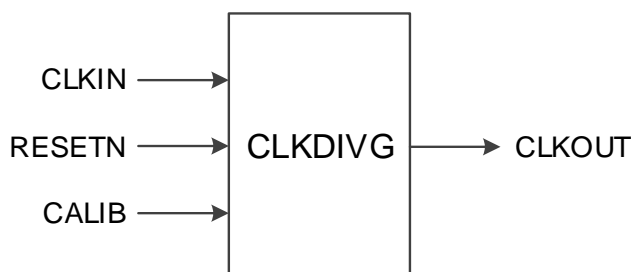
ファミリー	シリーズ	デバイス
LittleBee	GW1N	GW1N-2, GW1N-1P5, GW1N-2B, GW1N-1P5B
	GW1NR	GW1NR-2, GW1NR-2B
Arora	GW2AN	GW2AN-18X, GW2AN-9X

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 つのみあり、その位置は固定されており、その入力信号は固定の IO から供給されます。その機能は 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	クロック入力信号
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します。
CLKOUT	出力	クロック出力信号

CALIB 信号は、IOLOGIC の CALIB と併用することができます。その機能については、[5.5 CLKDIV](#) を参照してください。

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	クロックの分周係数を設定します。
GSREN	"false", "true"	"false"	グローバルリセット GSR を有効にします

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 :

```
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=>clk_in,
        RESETN=>resetn,
        CALIB=>calib,
        CLKOUT=>clkout
    );
```

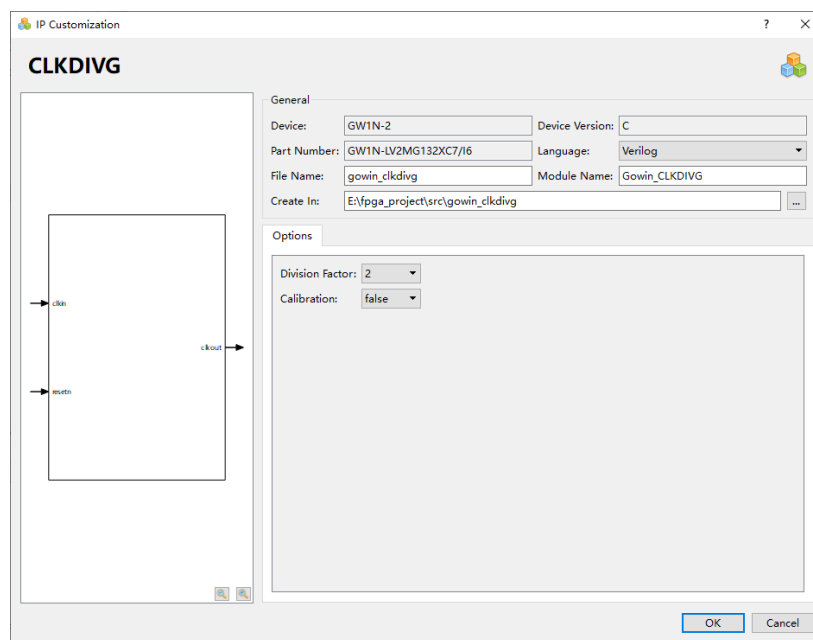
5.6.2 IP の呼び出し

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、**General** 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. General 構成タブ

General 構成タブは、IP ファイルの構成に使用されます。CLKDIVG の General 構成タブの使用は DQCE モジュールと同様であるので、DQCE の General 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

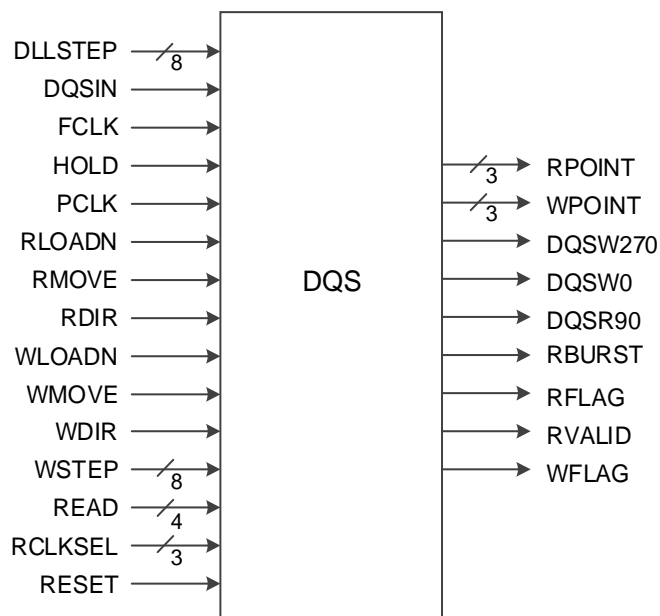
ファミリー	シリーズ	デバイス
Arora	GW2A	GW2A-18, GW2A-18C, GW2A-55, GW2A-55C
	GW2AN	GW2AN-55C, GW2AN-18X, GW2AN-9X
	GW2AR	GW2AR-18, GW2AR-18C
	GW2ANR	GW2ANR-18C

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、および書き込みレベリングと読み出しのキャリブレーションに使用される、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	2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に同期させるために使用されます。DQS 読み出しに使用の場合、FIFO カウンターをリセットするために使用されます。
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します "0" : 遅延を増やします "1" : 遅延を減らします
RLOADN	入力	DDR 読み出しの最終遅延ステップを初期化します。 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を変更します。パルスごとに 1 回変更します
WDIR	入力	DDR 書き込みの遅延方向を調整します "0" : 遅延を増やします "1" : 遅延を減らします
WLOADN	入力	DDR 書き込みの最終遅延ステップを初期化します。 アクティブ Low
WMOVE	入力	WMOVE が立ち下がりエッジのときに DDR 書き込みの遅延ステップを変更します。パルスごとに 1 回変更します
WSTEP[7:0]	入力	DDR 書き込みレベリング遅延制御に使用されます
READ[3:0]	入力	DDR 読み出しモードに使用される READ 信号
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 読み出しポインタ
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 書き込みポインタ
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。

ポート名	I/O	説明
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	読み出し遅延調整の under-flow または over-flow を示す READ 遅延調整出力フラグ
WFLAG	出力	書き込み遅延調整の 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 モードの選択 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 読み出しポインタの設定
DQS_MODE	"X1", "X2_DDR2", "X2_DDR3", "X4", "X2_DDR3_EXT"	"X1"	DQS モードの選択
HWL	"false", "true"	"false"	update0/1 のタイミング関係の制御 "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 を有効にします

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-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 に含まれるプログラマブルなオンチップオシレータは、MSPI コンフィギュレーションモードまたはユーザーデザイン用のクロックソースとして使用できます。動作パラメータの設定により、最大 64 のクロック周波数を取得できます。

出力クロック周波数は以下の式で計算できます。

$$f_{CLKOUT} = f_{osc} / \text{FREQ_DIV} ;$$

ここで、 f_{osc} は OSC の発振周波数で、除数 FREQ_DIV は設定されるパラメータ (2~128 の偶数) です。

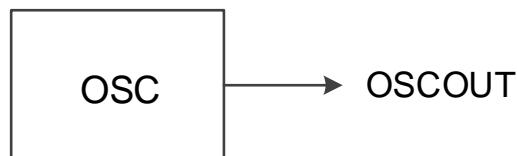
注記：

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

バイスは 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 の偶数	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 ファミリー) "GW2A-18"(Arora ファミリー)	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® ファミリー	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、MSPI コンフィギュレーションモードまたはユーザーデザイン用のクロックソースとして使用できます。動作パラメータの設定により、最大 64 のクロック周波数を取得できます。出力クロック周波数は、次の式により算出されます。

$$f_{CLKOUT} = f_{oscz} / FREQ_DIV ;$$

ここで、 f_{oscz} は OSCZ の発振周波数で、除数 $FREQ_DIV$ は設定されるパラメータ(2~128 の偶数)です。

注記：

f_{oscz} はデバイスおよびスピードグレードにより異なります。GW1NS-4/GW1NS-4C/GW1NSR-4/GW1NSR-4C/GW1NSER-4C デバイスの C7 スピードグレードは 260MHz をサポートし、その他のサポートされるデバイスおよびスピードグレードは 250MHz をサポートします。

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 クロック出力信号

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 の偶数	100	OSC 分周係数の設定
S_RATE	"SLOW", "FAST"	"SLOW"	GWINS-4 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

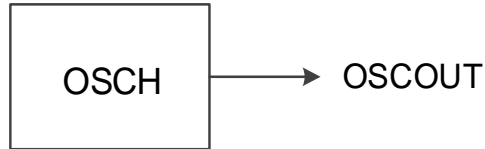
このプログラマブルなオンチップオシレータ **OSCH** は、**MSPI** コンフィギュレーションモードまたはユーザーデザイン用のクロックソースとして使用できます。動作パラメータの設定により、最大 **64** のクロック周波数を取得できます。出力クロック周波数は、次の式により算出されます。

$$f_{CLKOUT} = 240MHz / \text{FREQ_DIV} ;$$

"FREQ_DIV"は 2~128 の偶数のパラメータです。

ポート図

図 6-3 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 クロック出力信号

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 の偶数	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®ファミリー	GW1N	GW1N-2, GW1N-1P5, GW1N-2B, GW1N-1P5B
	GW1NR	GW1NR-2, GW1NR-2B

機能の説明

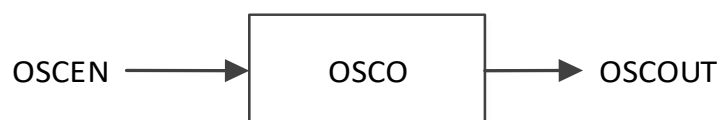
FPGA 製品には、OSC 機能を動的にオン/オフでき、Regulator 電源供給をサポートする、±5%のクロック精度のプログラマブルなオンチップオシレータが備わっています。このオンチップオシレータは、MSPI コンフィギュレーションモードまたはユーザーデザイン用のクロックソースとして使用できます。動作パラメータの設定により、最大 64 のクロック周波数を取得できます。出力クロック周波数は、次の式により算出されます。

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

"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
OSCO	出力	OSC クロック出力信号

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(
    .OSCOUT(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マブル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、MSPI コンフィギュレーションモードまたはユーザーデザイン用のクロックソースとして使用できます。動作パラメータの設定により、最大 64 のクロック周波数を取得できます。出力クロック周波数は、次の式により算出されます。

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

"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 OSCO のポートの説明

ポート名	I/O	説明
OSCOUT	出力	OSC クロック出力信号

パラメータの説明

表 6-15 OSCW 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 の偶数	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  
    );
```

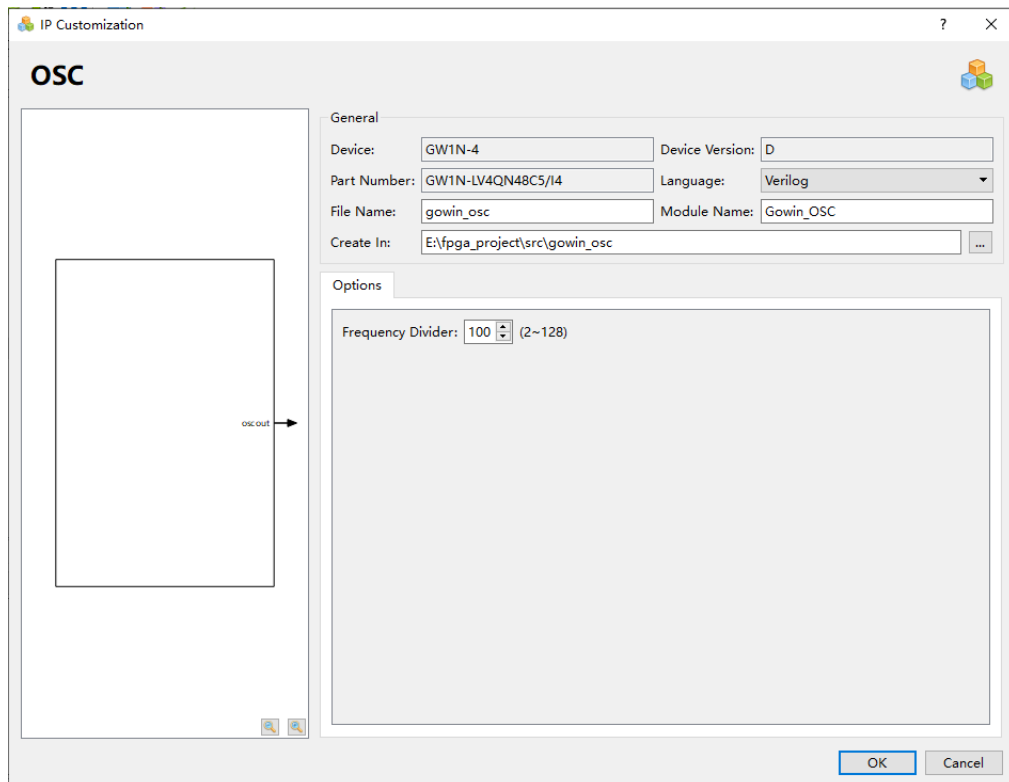
6.2 IPの呼び出し

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、**General** 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. **General 構成タブ**
General 構成タブは、IP ファイルの構成に使用されます。OSC の General 構成タブの使用は DQCE モジュールと同様であるので、DQCE の General 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。

