



MAX 10 のクロッキングおよび PLL ユーザー ガイド

UG-M10CLKPLL
2017.02.21

インテル® Quartus® Prime デザインスイートのための更新 16.0





目次

1 MAX[®] 10 のクロッキングおよび PLL の概要	4
1.1 クロック・ネットワークの概要.....	4
1.2 内蔵オシレータの概要.....	4
1.3 PLL の概要.....	4
2 MAX 10 のクロッキングおよび PLL のアーキテクチャと機能	5
2.1 クロック・ネットワークのアーキテクチャと機能.....	5
2.1.1 グローバル・クロック・ネットワーク.....	5
2.1.2 クロック・ピンの概要.....	5
2.1.3 クロック・リソース.....	6
2.1.4 グローバル・クロック・ネットワーク・ソース.....	6
2.1.5 グローバル・クロック・コントロール・ブロック.....	8
2.1.6 グローバル・クロック・ネットワークのパワーダウン.....	10
2.1.7 クロック・イネーブル信号.....	11
2.2 内蔵オシレータのアーキテクチャと機能.....	12
2.3 PLL のアーキテクチャと機能.....	12
2.3.1 PLL のアーキテクチャ.....	12
2.3.2 PLL の機能.....	13
2.3.3 PLL の位置.....	14
2.3.4 クロック・ピンの PLL への接続.....	16
2.3.5 PLL カウンタの GCLK への接続.....	16
2.3.6 PLL コントロール信号.....	17
2.3.7 クロック・フィードバック・モード.....	18
2.3.8 PLL の外部クロック出力.....	21
2.3.9 PLL からの ADC クロック入力.....	23
2.3.10 スペクトラム拡散クロッキング.....	23
2.3.11 PLL のプログラマブル・パラメータ.....	23
2.3.12 クロック・スイッチオーバー.....	26
2.3.13 PLL のカスケード接続.....	30
2.3.14 PLL リコンフィギュレーション.....	31
3 MAX 10 のクロッキングおよび PLL デザイン検討事項	33
3.1 クロック・ネットワークのデザイン検討事項.....	33
3.1.1 ガイドライン: クロック・イネーブル信号.....	33
3.1.2 ガイドライン: 接続制限.....	33
3.2 内蔵オシレータ・デザインの検討事項.....	34
3.2.1 ガイドライン: 接続制限.....	34
3.3 PLL のデザイン検討事項.....	34
3.3.1 ガイドライン: PLL コントロール信号.....	34
3.3.2 ガイドライン: 接続制限.....	34
3.3.3 ガイドライン: セルフ・リセット.....	34
3.3.4 ガイドライン: 出力クロック.....	35
3.3.5 ガイドライン: PLL のカスケード接続.....	35
3.3.6 ガイドライン: クロック・スイッチオーバー.....	36
3.3.7 ガイドライン: PLL リコンフィギュレーションにおける.mif ストリーミング.....	37



3.3.8 ガイドライン:PLL リコンフィギュレーションの scandone 信号.....	37
4 MAX 10 のクロッキングおよび PLL 実装ガイド.....	38
4.1 altclkctrl IP コア.....	38
4.2 ALTPLL IP コア.....	38
4.2.1 PLL ロック範囲の拡張.....	39
4.2.2 アドバンスド・パラメータを有するプログラマブル帯域幅.....	40
4.2.3 PLL ダイナミック・リコンフィギュレーションの実装.....	41
4.2.4 ダイナミック位相コンフィギュレーションの実装.....	45
4.3 ALTPLL_RECONFIG IP コア.....	48
4.3.1 リソース使用率レポートの取得.....	48
4.4 内蔵オシレーター IP コア.....	48
5 altclkctrl IP コア・リファレンス.....	50
5.1 altclkctrl パラメータ.....	50
5.2 altclkctrl ポートおよび信号.....	51
6 ALTPLL IP コア・リファレンス.....	52
6.1 ALTPLL パラメータ.....	52
6.1.1 動作モードのパラメータ設定.....	52
6.1.2 PLL コントロール信号のパラメータ設定.....	52
6.1.3 プログラマブル帯域幅のパラメータ設定.....	53
6.1.4 クロック・スイッチオーバーのパラメータ設定.....	53
6.1.5 PLL ダイナミック・リコンフィギュレーションのパラメータ設定.....	54
6.1.6 ダイナミック位相コンフィギュレーションのパラメータ設定.....	54
6.1.7 出力クロックのパラメータ設定.....	55
6.2 ALTPLL ポートおよび信号.....	56
7 ALTPLL_RECONFIG IP コア・リファレンス.....	59
7.1 ALTPLL_RECONFIG パラメータ.....	59
7.2 ALTPLL_RECONFIG ポートおよび信号.....	60
7.3 ALTPLL_RECONFIG カウンタ設定.....	62
8 内蔵オシレーター IP コア・リファレンス.....	65
8.1 altclkctrl パラメータ.....	65
8.2 内蔵オシレーターポートおよび信号.....	65
A MAX 10 のクロッキングおよび PLL ユーザーガイドのアーカイブ.....	66
B MAX 10 のクロッキングおよび PLL ユーザーガイド改訂履歴.....	67



1 MAX[®] 10 のクロッキングおよび PLL の概要

1.1 クロック・ネットワークの概要

MAX[®] 10 デバイスは、グローバル・クロック(GCLK)ネットワークをサポートします。

クロック・ネットワークはコアのクロック・ソースを提供します。クロック・ネットワークは、リセットやクリアなど高ファンアウトのグローバル信号ネットワークで使用することができます。

1.2 内蔵オシレータの概要

MAX 10 デバイスは、最大 116 MHz の組み込み型の内蔵オシレータを提供します。

この内蔵オシレータは、ユーザーによって有効または無効にすることができます。

1.3 PLL の概要

PLL(Phase-Locked Loop)は、デバイスのクロック管理、外部システムのクロック管理、および I/O インタフェース・クロッキングのために堅牢なクロック管理機能と合成機能を提供します。

以下の PLL を使用することができます。

- ゼロ遅延バッファ
- ジッター・アッテネータ
- 低スキュー・ファンアウト・バッファ
- 周波数合成器
- ボード上で必要なオシレータ数を削減
- 1 つのリファレンス・クロック・ソースから複数のクロック周波数を合成することにより、デバイスで使用されるクロック・ピンを削減
- オンチップ・クロック・デスキュー
- ダイナミック位相シフト
- カウンタ・リコンフィギュレーション
- 帯域幅のリコンフィギュレーション
- プログラマブル出力デューティ・サイクル
- PLL カスケード接続
- リファレンス・クロック・スイッチオーバー
- アナログ・デジタル・コンバータ(ADC)クロックの駆動



2 MAX 10 のクロッキングおよび PLL のアーキテクチャと機能

2.1 クロック・ネットワークのアーキテクチャと機能

2.1.1 グローバル・クロック・ネットワーク

GCLK はすべてのデバイスのクアドラントを供給し、デバイス全体を駆動します。I/O エレメント、ロジック・アレイ・ブロック (LAB)、専用マルチプライヤ・ブロック、および M9K メモリ・ブロックといったデバイス内のすべてのリソースは、クロック・ソースとして GCLK を使用することができます。これらのクロック・ネットワーク・リソースは、クロック・イネーブルやクリアなどの外部ピンによって供給されるコントロール信号に使用します。また、内部ロジックは、内部生成の GCLK と非同期クリア、クロック・イネーブル、またはその他の高ファンアウト・コントロール信号の GCLK を駆動することができます。

2.1.2 クロック・ピンの概要

GCLK ネットワークを駆動することができる外部クロック・ピンは 2 種類あります。

専用クロック入力ピン

専用クロック入力ピン (CLK<#>[p, n]) は、GCLK ネットワークの非同期クリア、プリセット、およびクロック・イネーブルといったクロックおよびグローバル信号を駆動するために使用することができます。

クロック入力に専用クロック入力ピンを使用しない場合は、専用クロック入力ピンを汎用入力ピンや出力ピンとして使用することもできます。

CLK ピンは、シングルエンドまたは差動入力として使用することができます。CLK ピンをシングルエンド・クロック入力として使用する場合、CLK<#>p ピンと CLK<#>n ピンの両方が GCLK ネットワークへの専用の接続を有します。CLK ピンを差動入力として使用する場合、同じ数の 2 つのクロック・ピンをペアにして、差動信号を受信します。

兼用クロック・ピン

プロトコル信号のような高ファンアウト・コントロール信号には兼用クロック (DPCLK) ピンを使用し、GCLK ネットワーク経由の PCI には TRDY および IRDY 信号を使用することができます。

DPCLK ピンは、I/O バンクの左右でのみ使用可能です。



2.1.3 クロック・リソース

表 1. MAX 10 のクロック・リソース

クロック・リソース	デバイス	利用可能なリソース数	クロック・リソースのソース
専用クロック入力ピン	<ul style="list-style-type: none"> 10M02 10M04 10M08 	シングルエンド:8、差動:4	I/O バンクの左右にある CLK[3..0][p,n]ピン
	<ul style="list-style-type: none"> 10M16 10M25 10M40 10M50 	シングルエンド:16、差動:8	I/O バンクの上、左、下および右にある CLK[7..0][p,n]ピン
DPCLK ピン	すべて	4	I/O バンクの左右にある DPCLK[3..0]ピン

クロック入力ピンの接続についての詳細は、ピン接続ガイドラインを参照してください。

関連情報

[MAX 10 FPGA Device Family Pin Connection Guidelines](#)

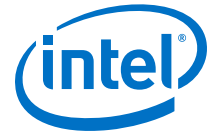
2.1.4 グローバル・クロック・ネットワーク・ソース

表 2. MAX 10 クロック・ピンの GCLK ネットワークへの接続

CLK ピン	GCLK
CLK0p	GCLK[0,2,4]
CLK0n	GCLK[1,2]
CLK1p	GCLK[1,3,4]
CLK1n	GCLK[0,3]
CLK2p	GCLK[5,7,9]
CLK2n	GCLK[6,7]
CLK3p	GCLK[6,8,9]
CLK3n	GCLK[5,8]
CLK4p ⁽¹⁾	GCLK[10,12,14]
CLK4n ⁽¹⁾	GCLK[11,12]
CLK5p ⁽¹⁾	GCLK[11,13,14]
CLK5n ⁽¹⁾	GCLK[10,13]
CLK6p ⁽¹⁾	GCLK[15,17,19]
CLK6n ⁽¹⁾	GCLK[16,17]
CLK7p ⁽¹⁾	GCLK[16,18,19]
CLK7n ⁽¹⁾	GCLK[15,18]

continued...

⁽¹⁾ これは 10M16、10M25、10M40、および 10M50 デバイスにのみ適用します。



CLK ピン	GCLK
DPCLK0	GCLK[0, 2]
DPCLK1	GCLK[1, 3, 4]
DPCLK2	GCLK[5, 7]
DPCLK3	GCLK[6, 8, 9]

図 -1: 10M02、10M04、および 10M08 デバイスの GCLK ネットワーク・ソース

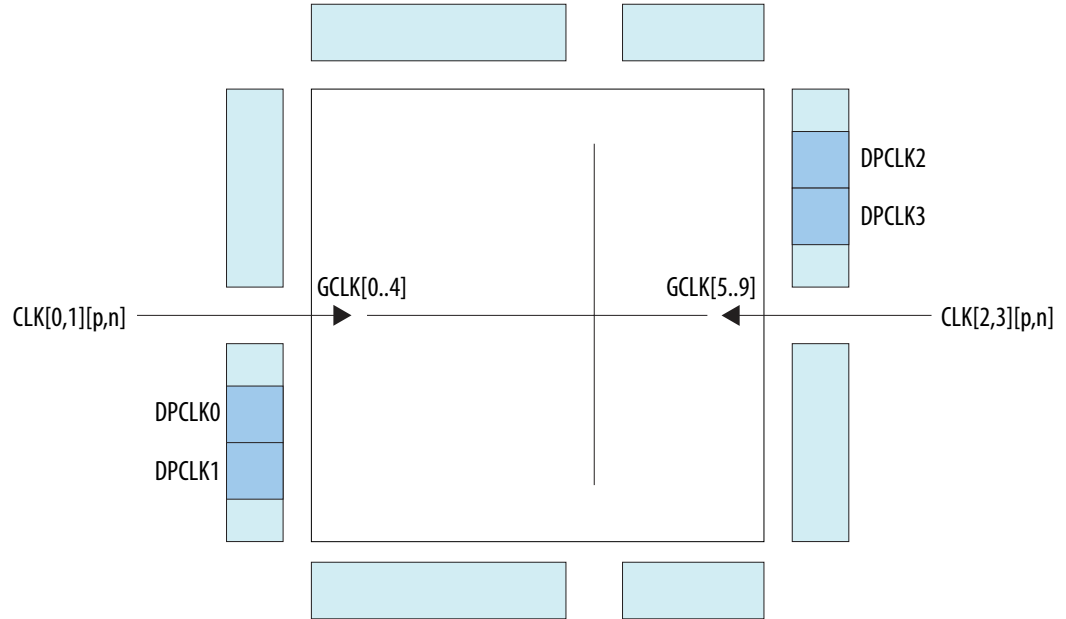
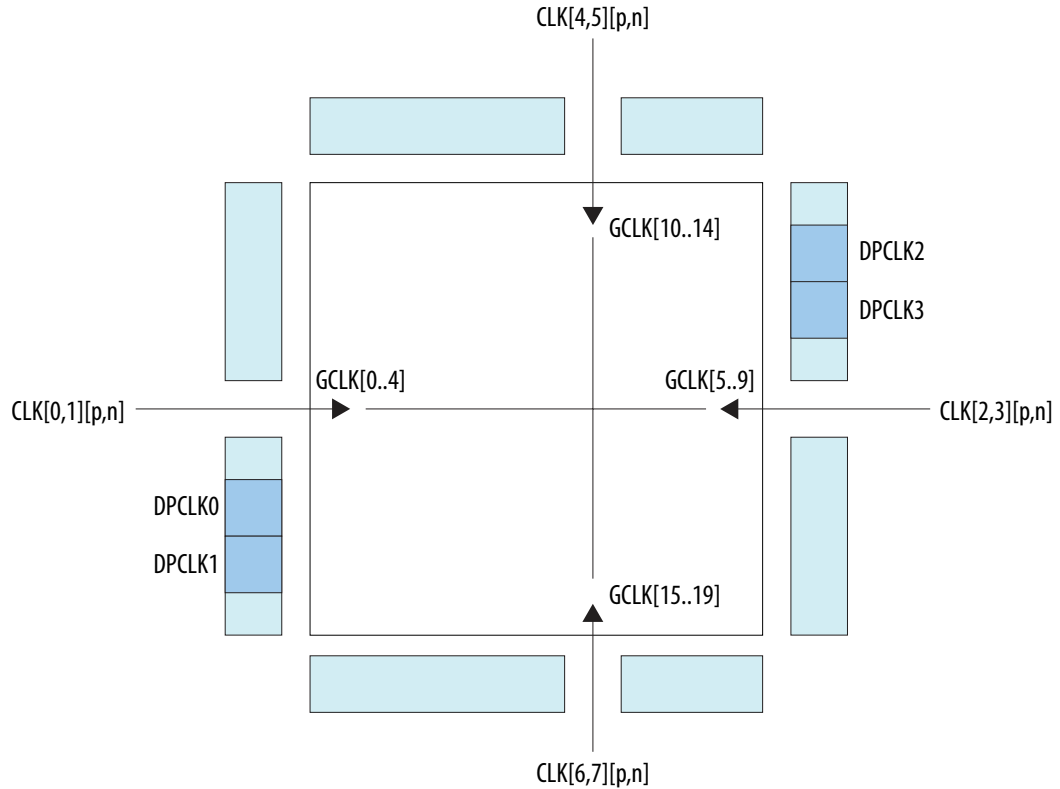


図 -2: 10M16、10M25、10M40、および 10M50 デバイスの GCLK ネットワーク・ソース



2.1.5 グローバル・クロック・コントロール・ブロック

クロック・コントロール・ブロックは GCLK を駆動します。クロック・コントロール・ブロックは専用クロック入力ピンに近い、デバイスの両側に配置されています。GCLK は最小のクロック・スキューと遅延に最適化されます。

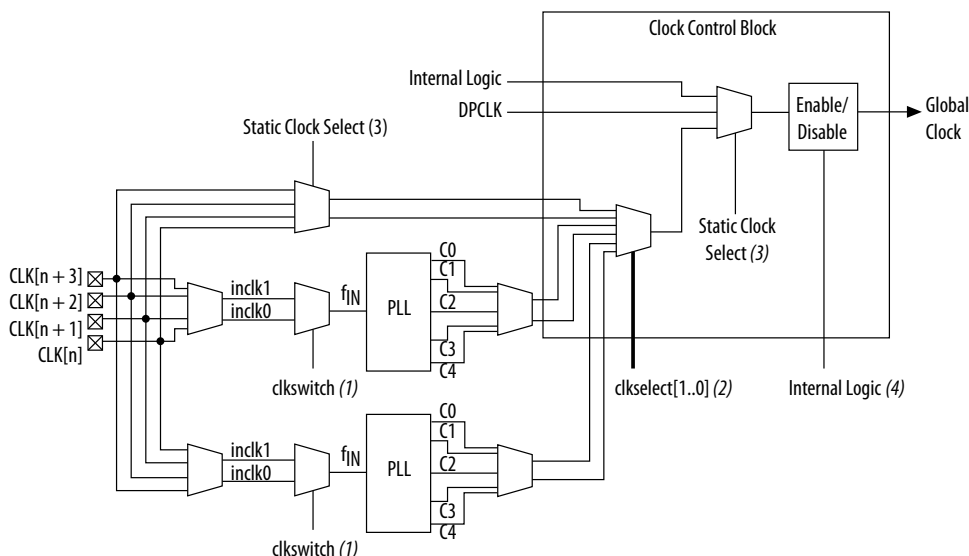
クロック・コントロール・ブロックは、以下の機能を有します。

- ダイナミックな GCLK クロック・ソースの選択 (DPCLK ピンと内部ロジック入力には適用しません)
- GCLK のマルチプレクサ化
- GCLK ネットワークのパワー・ダウン (ダイナミック・イネーブルおよびディセーブル)

表 3. クロック・コントロール・ブロックの入力

入力	説明
専用クロック入力ピン	専用クロック入力ピンは、クロック、または同期および非同期クリア、プリセット、クロック・イネーブルなどのグローバル信号を GCLK に駆動することができます。
DPCLK ピン	DPCLK ピンは、GCLK を介する PCI のプロトコル信号、TRDY および IRDY 信号といった高ファンアウトのコントロール信号に使用される双方向の兼用ピンです。DPCLK ピンによって駆動される入力を有するクロック・コントロール・ブロックは、PLL 入力を駆動することはできません。
PLL カウンタ出力	PLL カウンタ出力は、GCLK を駆動することができます。
内部ロジック	内部ロジック・エレメント (LE) を有効にして高ファンアウト、低スキュー信号パスを駆動するために、ロジック・アレイ配線を介して GCLK を駆動することができます。内部ロジックによって駆動される入力を有するクロック・コントロール・ブロックは、PLL 入力を駆動することはできません。

図 -3: クロック・コントロール・ブロック



注:

- (1) マニュアルPLLスイッチオーバー機能を使用している際、clkswitch信号はコンフィギュレーション・ファイルを通じて設定するかあるいはダイナミックに設定することができます。マルチプレクサの出力はPLLの入力クロック (fin) です。
- (2) clkselect[1..0]信号は、内部ロジックによって供給されます。デバイスがユーザー・モードの動的に選択するために使用することができます。GCLKへのクロック・ソースとして、1つのPLL (同じ側のPLLに適用可能なみ選択することができます)。
- (3) スタティック・クロック選択信号は、コンフィギュレーション・ファイル内に設定されます。そのため、デバイスがユーザー・モードの動的なコントロールを実行することはできません。
- (4) 内部ロジックは、ユーザー・モードのGCLKを有効または無効にするために使用することができます。

各 MAX 10 デバイスは、最大 20 個のクロック・コントロール・ブロックを有します。デバイスのそれぞれの側には 5 個のクロック・コントロール・ブロックがあります。

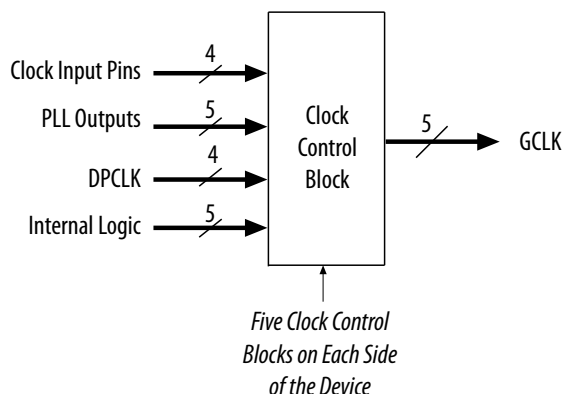
各 PLL は、c[4..0]カウンタを介して 5 個のクロック出力を生成します。これら 5 個のうち 2 個のクロックは、クロック・コントロール・ブロックを介して GCLK を駆動することができます。

前出の「クロック・コントロール・ブロックの入力」の表からは、以下の入力のみ特定のクロック・コントロール・ブロックに駆動することができます。

- 2つの専用クロック入力ピン
- 2つの PLL カウンタ出力
- 1つの DPCLK ピン
- 内部ロジックからの1つのソース

クロック・コントロール・ブロックからの出力は、対応する GCLK を供給します。クロック・コントロール・ブロックの入力が別の PLL または専用クロック入力ピンの出力である場合、GCLK は PLL 入力を駆動することができます。通常の I/O ピンは PLL 入力クロック・ポートを駆動することができません。

図 -4: デバイスの各側のクロック・コントロール・ブロック



クロック・コントロール・ブロックへの5つの入力のうち、2つのクロック入力ピンと2つのPLL出力がGCLKを供給するために動的に選択されます。クロック・コントロール・ブロックは、内部ロジックからの信号の静的な選択をサポートします。

関連情報

- 50 ページの `altclkctrl` パラメータ
- 51 ページの `altclkctrl` ポートおよび信号

2.1.6 グローバル・クロック・ネットワークのパワーダウン

MAX 10 GCLK (パワーダウン) は、静的および動的、両方のアプローチを使用して無効にすることができます。静的なアプローチでは、コンフィギュレーション・ビットは Quartus® Prime ソフトウェアによって生成されるコンフィギュレーション・ファイルに設定され、未使用の GCLK を自動的に無効にします。また、動的なクロック・イネーブルあるいはディセーブル機能により、内部ロジックによって GCLK のクロック・イネーブルまたはディセーブルを制御することが可能になります。

クロック・ネットワークが無効になると、クロック・ネットワークによって供給されるすべてのロジックはオフ状態になり、デバイス全体の消費電力を削減します。この機能は PLL から独立しており、クロック・ネットワークに直接適用されます。

Quartus Prime ソフトウェアの `altclkctrl` IP コア・パラメータ・エディタを介して、GCLK マルチプレクサの入力クロック・ソースと `clkena` 信号を設定することができます。

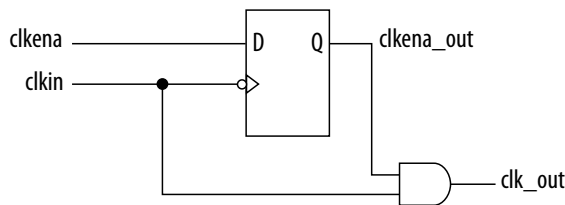
関連情報

- 50 ページの [altclkctrl パラメータ](#)
- 51 ページの [altclkctrl ポートおよび信号](#)

2.1.7 クロック・イネーブル信号

MAX 10 デバイスは、GCLK ネットワーク・レベルで `clkena` 信号をサポートします。これにより、PLL が使用されているときでもクロックをゲート・オフすることができます。出力クロックを再度イネーブルすると、回路がクロック・ネットワーク・レベルでクロックをゲート・オフするため、PLL は再同期または再ロック期間を必要としません。また、ループ関連のカウンタに影響しないため、`clkena` 信号とは無関係に PLL をロック状態に維持することができます。

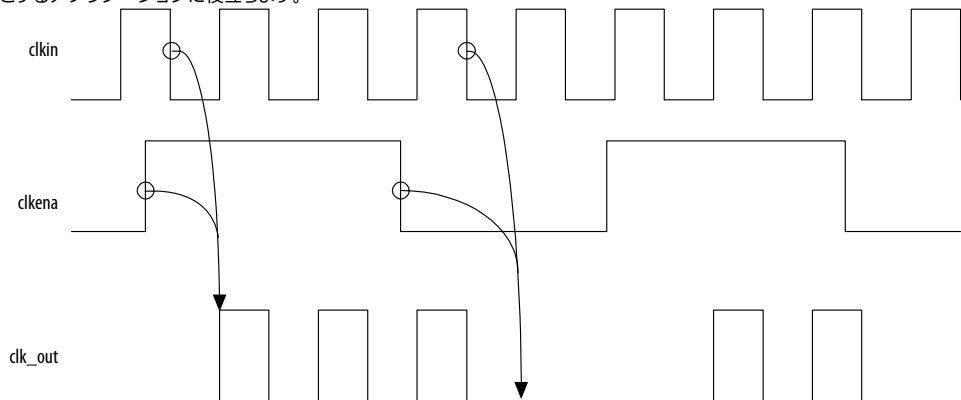
図 -5: `clkena` の実装



注意: 出力ピンに PLL の C0 出力を制御する `clkena` 回路は、1 個のレジスタではなく 2 個のレジスタと実装されます。

図 -6: `clkena` の実装(出力イネーブル)の波形例

`clkena` 信号は、クロック(`clkin`)の立ち下がりエッジでサンプリングされます。この機能は、低消費電力またはスリープ・モードを必要とするアプリケーションに役立ちます。



また、`clkena` 信号は、PLL の再同期中にシステムが周波数オーバーシュートを許容できない場合、クロック出力をディセーブルすることもできます。

関連情報

- 33 ページの [ガイドライン:クロック・イネーブル信号](#)
- 50 ページの [altclkctrl パラメータ](#)
- 51 ページの [altclkctrl ポートおよび信号](#)

2.2 内蔵オシレータのアーキテクチャと機能

MAX 10 デバイスは、クロック・マルチプレクサとディバイダを備えた内蔵リング・オシレータを有します。この内蔵リング・オシレータは最大 232 MHz で動作しますが、これには接続することはできません。この動作周波数はさらに低い周波数に分周されます。

デフォルトでは、内蔵オシレータはユーザー・モードでオフになっています。オシレータをオンにするには、内蔵オシレーター IP コアの `oscena` 信号をアサートします。

`oscena` 入力信号がアサートされると、オシレータが有効になり、`clkout` 出力信号を介して出力をロジック・アレイに配線することができます。`oscena` 信号が Low に設定される場合、`clkout` 信号は High のままです。この遅延は、TimeQuest timing analyzer を使用して解析することができます。

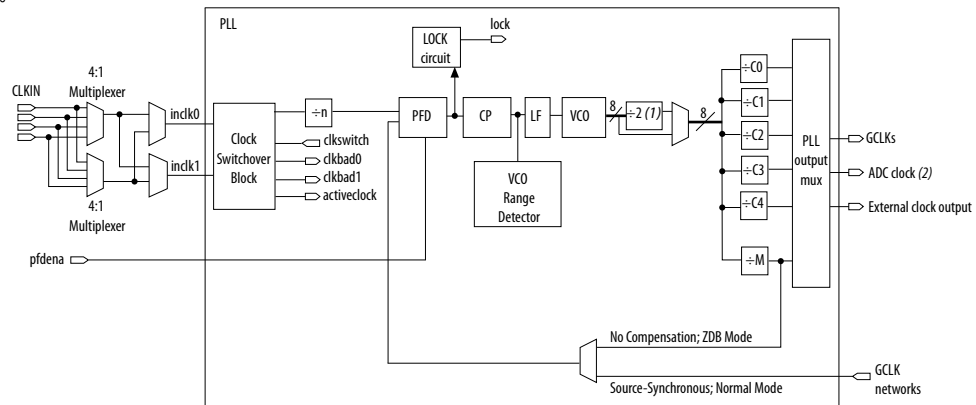
2.3 PLL のアーキテクチャと機能

2.3.1 PLL のアーキテクチャ

PLL の主な目的は、電圧制御オシレータ (VCO) の位相と周波数を入力リファレンス・クロックに同期させることです。

図 -7: MAX 10 PLL の上位レベルのブロック図

各クロック・ソースは、PLL としてデバイスの同じサイドに配置されている 2~4 本のクロック・ピンのどれからでも供給することができます。



注:

- (1) これは VCO ポストスケール・カウンタ K です。
- (2) ADC クロックは、PLL1 と PLL3 の CO カウンタによってのみ駆動することができます。

位相周波数検出器 (PFD)

PFD は、フィードバック・クロックからの入力 f_{FB} と入力リファレンス・クロックからの入力 f_{REF} を有します。PLL は、PFD を使用してフィードバック・クロックへの入力リファレンス・クロックの立ち上がりエッジを比較します。PFD は、VCO が高い周波数で動作する必要があるか、低い周波数で動作する必要があるかを決定する Up 信号または Down 信号を生成します。

チャージ・ポンプ (CP)

チャージ・ポンプが Up 信号でロジック High を受信する場合、電流はループ・フィルタに駆動されます。チャージ・ポンプが Down 信号でロジック High を受信する場合、電流はループ・フィルタから引き出されます。



ループ・フィルタ(LF)

ループ・フィルタは、Up 信号と Down 信号を PFD から VCO のバイアスに使用する電圧に変換します。また、ループ・フィルタはチャージ・ポンプからグリッチをフィルタし、電圧のオーバーシュートを防ぎます。これにより VCO のジッタを最小限に抑えます。

電圧制御オシレータ(VCO)

VCO の動作速度は、チャージ・ポンプからの電圧によって決まります。VCO は、4 ステージの差動リング・オシレータとして実装されます。VCO の周波数(f_{VCO})を入力基準周波数(f_{REF})よりも高くするために、分周カウンタ(M)がフィードバック・ループに挿入されます。

VCO の周波数は、次の式を用いて算出します。

$$f_{VCO} = f_{REF} \times M = f_{IN} \times M/N$$

式中の f_{IN} は PLL への入力クロック周波数であり、N はプリスケール・カウンタです。

VCO 周波数は重要なパラメータであり、PLL の適切な動作を確実にするために 600 ~ 1,300 MHz である必要があります。Quartus Prime ソフトウェアは、デザインのクロック出力および位相シフト要件に基づいて、VCO 周波数を推奨される範囲内に自動的に設定します。

ポスト・スケール・カウンタ(C)

VCO 出力は、最大 5 つのポストスケール・カウンタ(C0、C1、C2、C3、および C4)を供給することができます。これらのポストスケール・カウンタにより、PLL が多数の調和関係にある周波数を生成することが可能になります。

内部遅延エレメント

MAX 10 PLL は、GCLK ネットワークと I/O バッファ上の配線を補償するために内部遅延エレメントを有します。これらの内部遅延は固定されています。

PLL 出力

MAX 10 PLL は最大 5 つの GCLK 出力と 1 つの専用外部クロック出力をサポートします。GCLK ネットワークまたは専用外部クロック出力への出力周波数(f_{OUT})は、以下の式を用いて決定します。

$$f_{REF} = f_{IN}/N \text{ と}$$

$$f_{OUT} = f_{VCO}/C = (f_{REF} \times M)/C = (f_{IN} \times M)/(N \times C)$$

式中で、C は C0、C1、C2、C3、または C4 カウンタの設定を表します。

2.3.2 PLL の機能

表 4. MAX 10 PLL の機能

機能	サポート
C 出力カウンタ	5
M、N、C カウンタ・サイズ	1 ~ 512 ⁽²⁾
専用クロック出力	シングルエンド: 1、差動: 1
<i>continued...</i>	



機能	サポート
専用クロック入力ピン	シングルエンド:4, 差動:2
スペクトラム拡散入力クロック・トラッキング	あり ⁽³⁾
PLL カスケード接続	GCLK 経由
ソース・シンクロナス補償	あり
非補償モード	あり
通常補償	あり
ゼロ遅延バッファ補償	あり
位相シフト分解能	最小 96 ps の増分量 ⁽⁴⁾
プログラマブル・デューティ・サイクル	あり
出力カウンタ・カスケード接続	あり
入力クロック・スイッチオーバー	あり
ユーザー・モード・リコンフィギュレーション	あり
ロック検出の喪失	あり
4:1 マルチプレクサ CLK 入力選択	あり

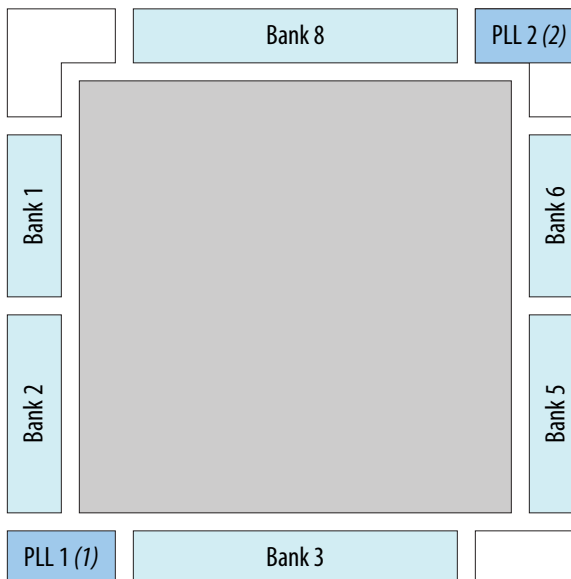
2.3.3 PLL の位置

次の図は、PLL の物理的な位置を示しています。各インデックスはデバイス内の 1 つの PLL を表します。PLL の物理的な位置は Quartus Prime Chip Planner の座標に対応します。

-
- (2) 出力クロックが 50%のデューティ・サイクルを使用する場合、C カウンタの範囲は 1~512 です。50%以外のデューティ・サイクルを使用する出力クロックでは、ポストスケール・カウンタの範囲は 1~256 です。
- (3) 入力クロック・ジッタが入力ジッタ許容仕様範囲内である場合にのみ適用可能です。
- (4) 最小の位相シフトは、VCO 周期を 8 で除算することによって求められます。また、MAX 10 デバイス・ファミリは、すべての出力周波数を最小 45°の増分でシフトすることができます。周波数と分周パラメータによっては、より細かい調整も可能です。



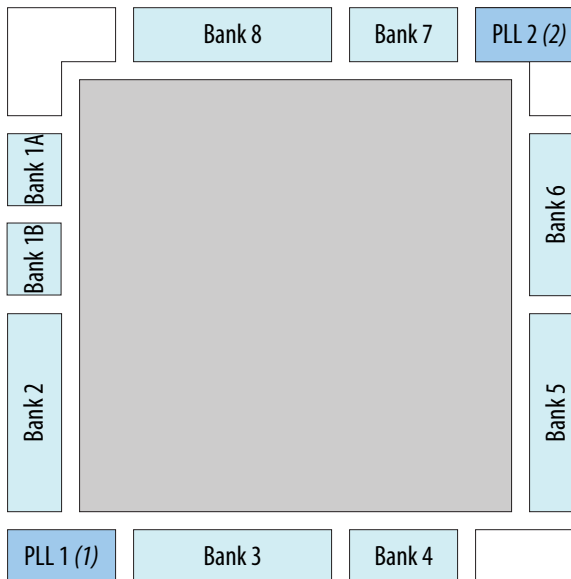
図 -8: 10M02 デバイスの PLL の位置



注:

- (1) V36パッケージを除くすべてのパッケージで使用可能です。
- (2) U324およびV36パッケージでのみ使用可能です。

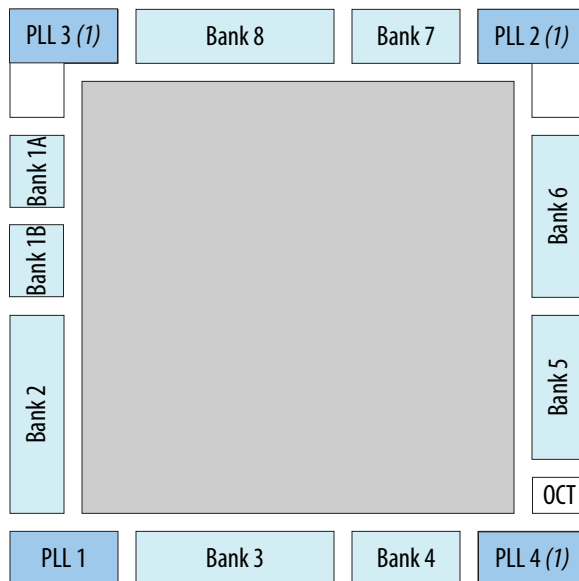
図 -9: 10M04 および 10M08 デバイスの PLL の位置



注:

- (1) V81パッケージを除くすべてのパッケージで使用可能です。
- (2) F256, F484, U324, およびV81パッケージでのみ使用可能です。

図 -10: 10M16、10M25、10M40、および 10M50 デバイスの PLL の位置



注:
 (1) E144およびU169パッケージを除くすべてのパッケージで使用可能です。

2.3.4 クロック・ピンの PLL への接続

表 5. MAX 10 専用クロック入力ピンの PLL への接続

専用クロック・ピン	PLL
CLK[0,1][p,n]	PLL1, PLL3
CLK[2,3][p,n]	PLL2, PLL4
CLK[4,5][p,n]	PLL2, PLL3
CLK[6,7][p,n]	PLL1, PLL4

2.3.5 PLL カウンタの GCLK への接続

表 6. MAX 10 PLL カウンタの GCLK ネットワークへの接続

PLL カウンタ出力	GCLK
PLL1_C0	GCLK[0,3,15,18]
PLL1_C1	GCLK[1,4,16,19]
PLL1_C2	GCLK[0,2,15,17]
PLL1_C3	GCLK[1,3,16,18]
PLL1_C4	GCLK[2,4,17,19]
PLL2_C0	GCLK[5,8,10,13]
PLL2_C1	GCLK[6,9,11,14]

continued...



PLL カウンタ出力	GCLK
PLL2_C2	GCLK[5, 7, 10, 12]
PLL2_C3	GCLK[6, 8, 11, 13]
PLL2_C4	GCLK[7, 9, 12, 14]
PLL3_C0 ⁽⁵⁾	GCLK[0, 3, 10, 13]
PLL3_C1 ⁽⁵⁾	GCLK[1, 4, 11, 14]
PLL3_C2 ⁽⁵⁾	GCLK[0, 2, 10, 12]
PLL3_C3 ⁽⁵⁾	GCLK[1, 3, 11, 13]
PLL3_C4 ⁽⁵⁾	GCLK[2, 4, 12, 14]
PLL4_C0 ⁽⁵⁾	GCLK[5, 8, 15, 18]
PLL4_C1 ⁽⁵⁾	GCLK[6, 9, 16, 19]
PLL4_C2 ⁽⁵⁾	GCLK[5, 7, 15, 17]
PLL4_C3 ⁽⁵⁾	GCLK[6, 8, 16, 18]
PLL4_C4 ⁽⁵⁾	GCLK[7, 9, 17, 19]

2.3.6 PLL コントロール信号

PLL の演算と再同期を観察および制御するにあたって、以下の 3 つの信号を使用することができます。

pfdena

pfdena 信号は、最後にロックされた周波数を維持し、システムがシャットダウンする前に現在の設定を保存する時間を確保するために使用します。

pfdena 信号は、プログラマブル・ゲートで PFD 出力を制御します。PFD 回路はデフォルトで有効になっています。PFD 回路が無効になっている場合、PLL 出力は入力クロックに依存せず、ロック・ウィンドウの外側にドリフトする傾向があります。

areset

areset 信号は、各 PLL のリセット入力または再同期化入力です。これらの入力信号は、デバイス入力ピンまたは内部ロジックによって駆動することができます。

areset 信号をアサートすると、PLL カウンタがリセットし、PLL 出力をクリアして PLL のロックを解除します。VCO は標準設定に戻ります。areset 信号をディアサートすると、PLL は再びロックして入力に再同期します。

areset 信号のアサートは、VCO を無効にするものではなく、VCO を公称値にリセットするものです。VCO が完全に無効になるのは、デザインにインスタンス化された PLL を有しない場合のみです。

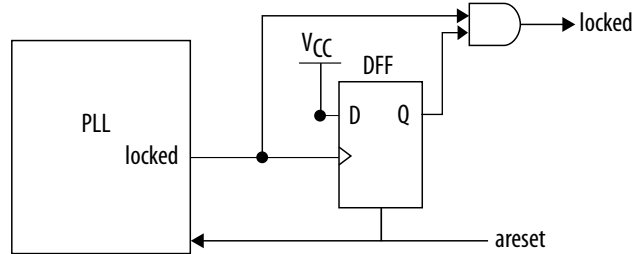
locked

locked 出力は、PLL が基準クロックにロックし、PLL クロック出力が ALTPLL IP コアのパラメータ・エディタで設定された目的の位相および周波数で動作していることを示します。

⁽⁵⁾ これは 10M16、10M25、10M40、および 10M50 デバイスにのみ適用します。

Intel は、PLL のステータスを制御および監視するために、デザインで areset および locked 信号を使用することを推奨します。この実装を次の図に示します。

図 -11: ロックされた信号の実装



注意: D フリップ・フロップの前に locked 信号をプローブするために SignalTap® II ツールを使用すると、areset がディアサートされたときにのみ locked 信号が Low になります。areset 信号がイネーブルされていない場合、追加のロジックは ALTPLL IP コアに実装されません。

関連情報

- 34 ページの [ガイドライン:PLL コントロール信号](#)
- 52 ページの [PLL コントロール信号のパラメータ設定](#)
- 56 ページの [ALTPLL ポートおよび信号](#)

2.3.7 クロック・フィードバック・モード

MAX 10 PLL は、最大 4 つの異なるクロック・フィードバック・モードをサポートします。各モードでは、クロックの通倍と分周、位相シフト、およびプログラマブル・デューティ・サイクルが可能になります。

PLL が入出力遅延を完全に補償するのは、特定の PLL に対応する専用クロック入力ピンをクロック・ソースとして使用する場合に限られます。

たとえば、PLL1 をノーマル・モードで使用している場合、以下に示すクロック入力ピンのいずれかから PLL および PLL クロックの出力-デスティネーション・レジスタまでのクロック遅延は、完全に補償されます。

- CLK0
- CLK1
- CLK2
- CLK3

GCLK ネットワークを使用して PLL を駆動する場合、入出力遅延は Quartus Prime ソフトウェアで完全に補償されないことがあります。

関連情報

- 52 ページの [動作モードのパラメータ設定](#)

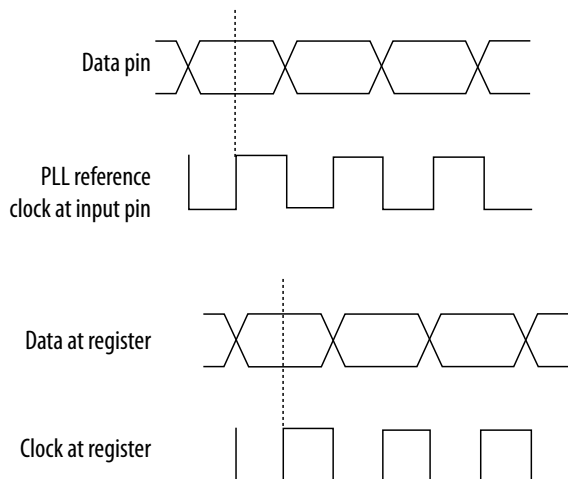
2.3.7.1 ソース・シンクロナス・モード

入力ピンにデータとクロックが同時に到達する場合、データとクロックの位相関係はどの I/O エLEMENT 入力レジスタのデータ・ポートとクロック・ポートでも同じのままです。



このモードは、ソース・シンクロナス・データ転送に使用することができます。I/O エLEMENTのデータ信号とクロック信号では、両信号が同じ I/O 規格を使用している限り同様のバッファ遅延が発生します。

図 -12: ソース・シンクロナス・モードのクロックおよびデータ間の位相関係の例



ソース・シンクロナス・モードは、以下の 2 つのパス間における遅延の差を含む、クロック・ネットワーク遅延を補償します。

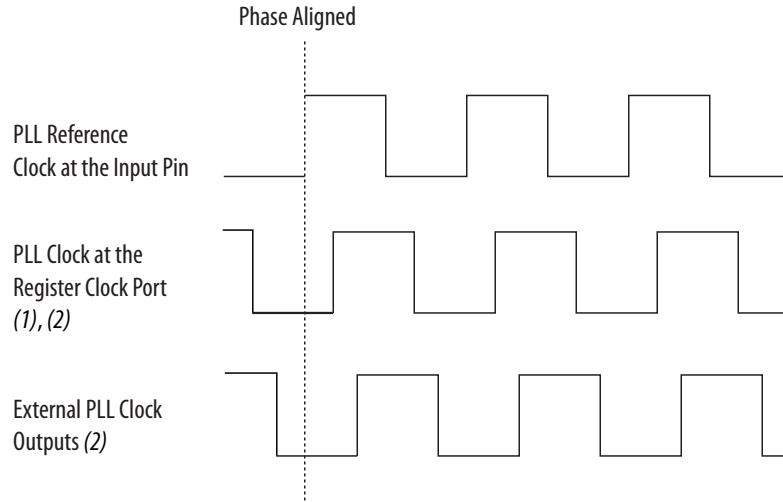
- データ・ピンから I/O エLEMENT・レジスタ入力
- クロック入力ピンから PLL PFD 入力

ソース・シンクロナス・モードの PLL でクロックされるすべてのデータ・ピンについては、Quartus Prime ソフトウェアで I/O エLEMENTのレジスタ遅延チェーンへの入力ピンをゼロに設定します。すべてのデータ・ピンは Quartus Prime ソフトウェアの **PLL COMPENSATED logic** オプションを使用する必要があります。

2.3.7.2 非補償モード

非補償モードでは、PLL はいかなるクロック・ネットワークも補償しません。このモードでは、PFD へのクロック・フィードバックが通過する回路がそれほど多くないため、ジッタ性能が向上します。PLL の内部クロック出力と外部クロック出力はいずれも、PLL クロック入力を基準にして位相シフトされます。

図 -13: 非補償モードの PLL クロック間における位相関係の例



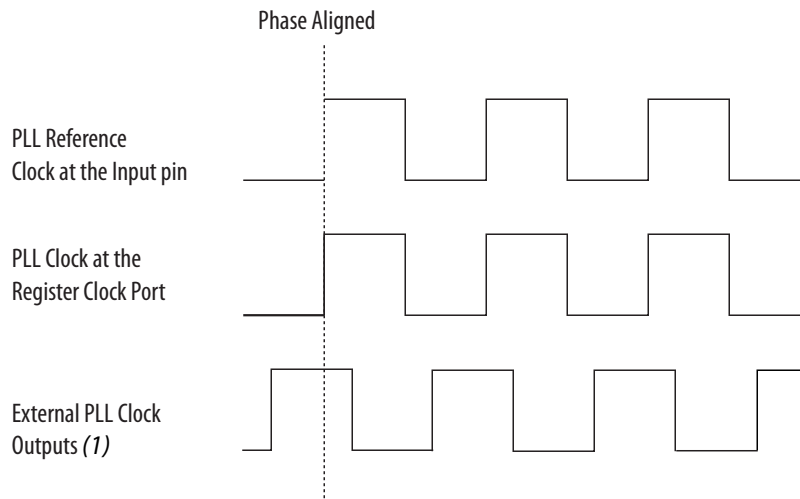
注:

- (1) PLLによって供給される内部クロックは、相互に位相アラインメントします。
- (2) PLLのクロック出力はPLL入力クロックよりも速くなるか、または遅くなる可能性があります。PLLクロック出力は、配線遅延によってPLL入力クロックより遅くなります。

2.3.7.3 ノーマル・モード

ノーマル・モードでは、PLL は GCLK ネットワークによってもたらされる遅延を完全に補償します。ノーマル・モードの内部クロックは、入力クロック・ピンに位相アラインメントされます。このモードでは、外部クロック出力ピンは入力クロック・ピンに相対する位相遅延を有します。両ピン間の位相差は、Quartus Prime ソフトウェアのタイミング・アナライザによってレポートされます。

図 -14: ノーマル補償モードの PLL クロック間における位相関係の例



注:

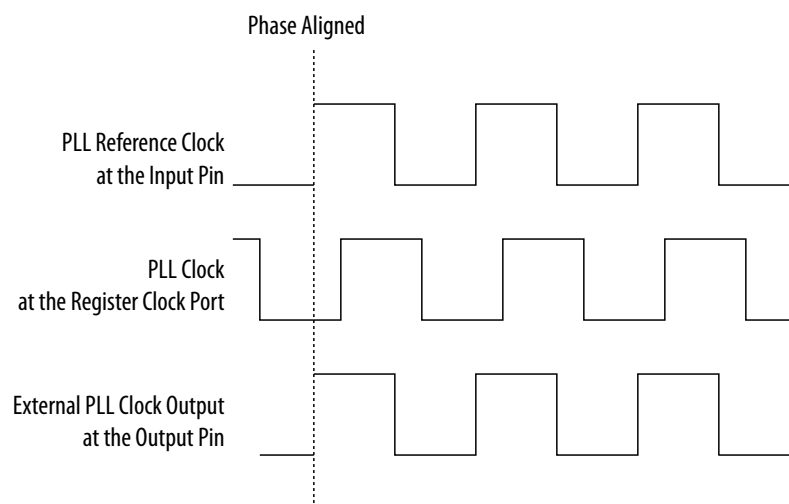
- (1) 外部クロック出力はPLL内部クロック信号よりも速くなるか、または遅くなること



2.3.7.4 ゼロ遅延バッファ・モード

ゼロ遅延バッファ(ZDB)モードでは、デバイス全体のゼロ遅延を目的として、外部クロック出力ピンはクロック入力ピンと位相アラインメントされます。このモードを使用する場合、入力および出力ピンにおけるクロック・アラインメントを確実にするために、入力クロックと出力クロックに同じ I/O 規格を使用します。

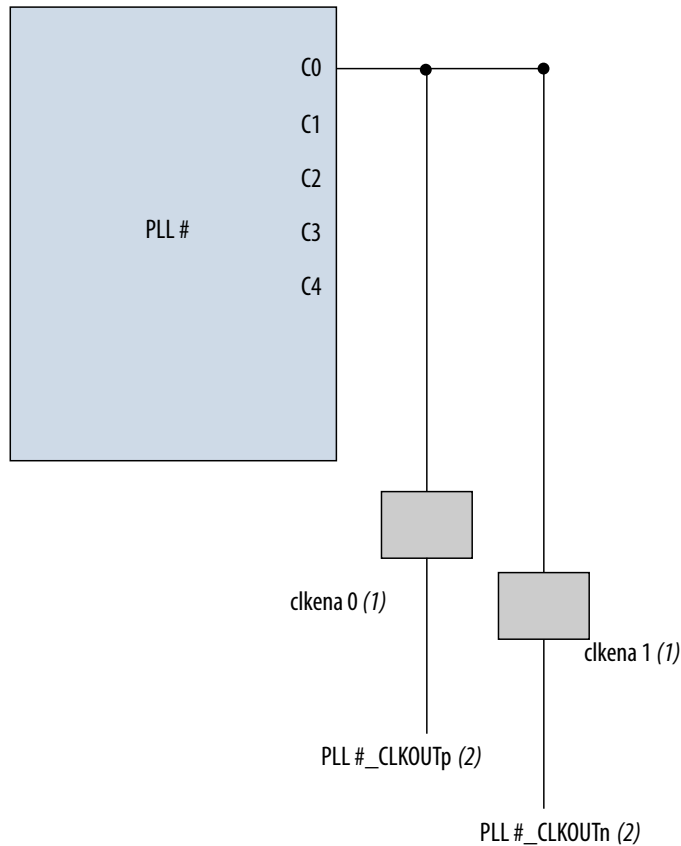
図 -15: ZDB モードの PLL クロック間における位相関係の例



2.3.8 PLL の外部クロック出力

MAX 10 デバイスの各 PLL は、1 つのシングル・エンド・クロック出力または 1 つの差動クロック出力をサポートします。C0 出力カウンタのみが GCLK を介さずに専用外部クロック出力を供給することができます。その他の出力カウンタは GCLK を介して他の I/O ピンを供給することができます。

図 -16: PLL の外部クロック出力



注:

- (1) これらの外部クロック・イネーブル信号は、ALTCLKCTRL IPコアを使用しているときにのみ利用可能
- (2) PLL#_CLKOUTpピンとPLL#_CLKOUTnピンは、1つのシングルエンド・クロック出力または1つの差動クロック出力として使用可能な兼用I/Oピンです。

差動出力ペアの各ピンの位相差は 180°です。180°の位相差をピンのペアに実装するにあたって、Quartus Prime ソフトウェアはデザインの NOT ゲートを I/O エlement に配置します。

クロック出力ピンのペアは、次の I/O 規格をサポートします。

- 標準出力ピン(トップおよびボトム・バンク内)と同じ I/O 規格
- LVDS
- LVPECL
- 差動高速トランシーバ・ロジック (HSTL)
- 差動 SSTL

MAX 10 PLL は、GCLK を介して通常の I/O ピンにドライブ・アウトすることができます。また、外部 PLL クロック機能が必要ない場合は、外部クロック出力ピンを汎用 I/O ピンとして使用することもできます。



関連情報

MAX 10 General Purpose I/O User Guide

PLL クロック出力ピンでサポートされている I/O 規格に関する詳細情報を提供します。

2.3.9 PLL からの ADC クロック入力

PLL1 と PLL3 からの C0 出力カウンタでのみ ADC クロックを駆動することができます。

カウンタ C0 は ADC クロック入力への専用パスを有します。

2.3.10 スペクトラム拡散クロッキング

MAX 10 デバイスでは、標準的な変調周波数のスペクトラム拡散入力が可能です。ただし、入力がスペクトラム拡散信号であることをデバイスが自動的に検出することはなく、入力信号は PLL の入力で確定的ジッタのように映ります。

入力信号が以下の条件を満たしている場合、MAX 10 PLL はスペクトラム拡散入力クロックをトラッキングすることができます。

- 入力信号が入力ジッタ許容仕様の範囲内である。
- 入力クロックの変調周波数がフィッターレポートで指定された PLL 帯域幅を下回っている。

MAX 10 デバイスは、スペクトラム拡散信号を内部で生成することはできません。

2.3.11 PLL のプログラマブル・パラメータ

2.3.11.1 プログラマブル・デューティ・サイクル

プログラマブル・デューティ・サイクルにより、PLL は可変デューティ・サイクルのクロック出力を生成することができます。この機能は PLL ポストスケール・カウンタでサポートされます。

デューティ・サイクルは、ポストスケール・カウンタに Low および High の時間カウントをセッティングすることによって設定します。デューティ・サイクルを選択するにあたって、Quartus Prime ソフトウェアは周波数入力と必要な逡倍レートまたは分周レートを使用します。

デューティ・サイクルの精度は、ポストスケール・カウンタ値によって決まります。精度は 50% をポストスケール・カウンタ値で除算した値で定義されます。たとえば、C0 カウンタが 10 の場合、5~90% のデューティ・サイクルでは 5% のステップが可能です。

プログラマブル・デューティ・サイクルをプログラマブル位相シフトと組み合わせることにより、オーバーラップのない正確なクロックを生成することができます。

関連情報

42 ページの [ポストスケール・カウンタ \(C0 ~ C4\)](#)

ポストスケールのカウンタのデューティサイクルをリアルタイムで設定する方法の詳細を説明します。

2.3.11.2 プログラマブル帯域幅

PLL の帯域幅は、入力クロックおよびそれに関連するジッタへの PLL のトラッキング能力を示す測度です。MAX 10 PLL は、ループ・フィルタやチャージ・ポンプを含む PLL ループのプログラマブル特性を使用して、PLL 帯域幅を高度にコントロールします。PLL の帯域幅は、PLL のクローズド・ループ・ゲインの 3-dB 周波数によって決まります。この帯域幅は、オープン・ループ PLL 応答のユニティ・ゲイン・ポイントとほぼ一致します。

関連情報

- 40 ページの [アドバンスド・パラメータを有するプログラマブル帯域幅](#)
- 44 ページの [チャージ・ポンプとループ・フィルタ](#)
PLL 帯域幅をリアルタイムで更新するための PLL コンポーネントに関する詳細情報を提供します。
- 53 ページの [プログラマブル帯域幅のパラメータ設定](#)

2.3.11.3 プログラマブル位相シフト

MAX 10 デバイスは、位相シフトを使用してクロック遅延を実装します。以下に示す方法のいずれかを使用して MAX 10 PLL から出力クロックを位相シフトすることができます。

- VCO 位相タップを使用する高分解能
- カウンタ始動時間を使用する低分解能

VCO 位相出力とカウンタ始動時間は、遅延を挿入するにあたって最も正確な方法です。これらの方法は純粋にカウンタ設定に基づいており、プロセス、電圧、および温度とは関係しません。

MAX 10 デバイスは、VCO 位相タップのダイナミック位相シフトのみをサポートします。この位相シフトは何度でもコンフィギュレーションすることができます。各位相シフトにはおよそ 1scanclk サイクルを要し、大きな位相シフトを迅速に実装することができます。

分解能の高い位相シフト

分解能の高い位相シフトを実装するには、出力カウンタのいずれか (C[4..0]) または M カウンタが VCO の 8 つの位相のいずれかを基準クロックとして使用できるようにします。これにより、遅延時間を高い分解能で調整することができます。以下に示す式は、この方法を使用して挿入可能な最小遅延時間を示しています。

図 -17: 高分解能の位相シフトの式

式中の f_{REF} は、入力基準クロック周波数を示します。

$$\Phi_{fine} = \frac{T_{VCO}}{8} = \frac{1}{8f_{VCO}} = \frac{1}{8} \times \frac{N}{M \times f_{REF}}$$

たとえば、 f_{REF} が 100 MHz、 $N = 1$ 、 $M = 8$ の場合、 $f_{VCO} = 800$ MHz、 $\Phi_{fine} = 156.25$ ps となります。この位相シフトは PLL の動作周波数で定義され、位相シフトの値は基準クロック周波数とカウンタ設定によって決まります。

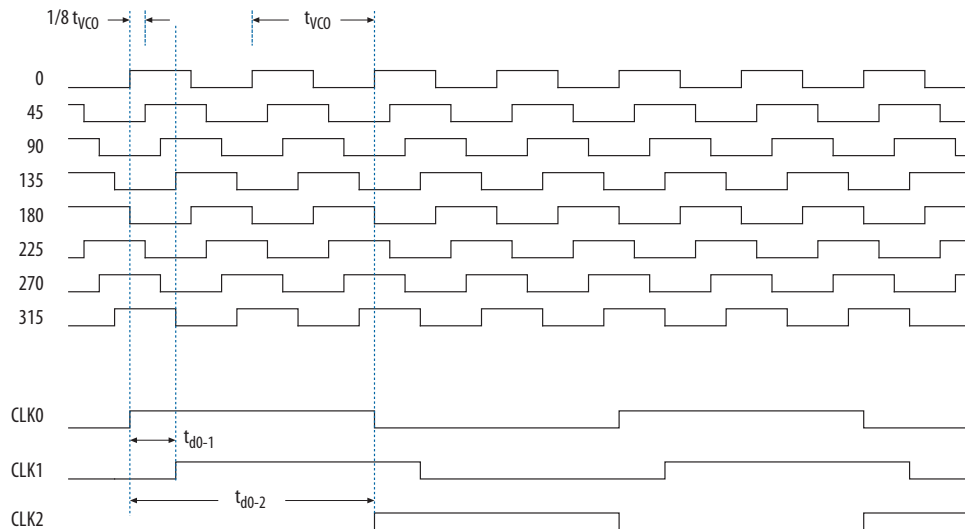
次の図は、VCO 位相タップ方式の高分解能を使用した位相シフトの挿入例を示しています。VCO からの 8 つの位相は、参考用に名前を付けて示しています。



図 -18: VCO 位相出力とカウンタ遅延時間を使用した遅延挿入の例

この例では、

- CLK0 は VCO からの 0°位相を基準とし、カウンタの C 値はゼロに設定されています。
- CLK1 信号は 4 で分周され、High 時間と Low 時間の長さはそれぞれ 2VCO クロックになります。CLK1 は VCO からの 135° 位相タップを基準とし、カウンタの C 値は 1 に設定されています。
- CLK2 信号も 4 で分周されます。この例では、2 つのクロックは 3 Φ_{fine} だけオフセットされます。CLK2 は VCO からの 0°位相を基準としますが、カウンタの C 値は 3 に設定されています。これにより、最終的な遅延は 2 Φ_{coarse} (2 つの完全な VCO 期間) になります。



分解能の低い位相シフト

分解能の低い位相シフトを実装するには、あらかじめ設定されたカウンタ・クロックの数だけカウンタの始動を延期させます。

図 -19: 低分解能の位相シフトの式

式中の C は、カウンタ遅延時間に設定されたカウント値 (Quartus Prime ソフトウェアのコンパイル・レポートの PLL 使用セクションにおける初期設定) です。初期値が 1 の場合、 $C - 1 = 0^\circ$ 位相シフトです。

$$\Phi_{coarse} = \frac{C - 1}{f_{VCO}} = \frac{(C - 1)N}{Mf_{REF}}$$

関連情報

- 45 ページの [ダイナミック位相コンフィギュレーションの実装](#)
- 46 ページの [ダイナミック位相コンフィギュレーション・カウンタ選択](#)
- 47 ページの [アドバンスド・パラメータを有するダイナミック位相コンフィギュレーション](#)
- 54 ページの [ダイナミック位相コンフィギュレーションのパラメータ設定](#)
Quartus Prime フトウェアの ALTPLL IP コアパラメータ設定に関する詳細情報を提供します。
- 59 ページの [ALTPLL_RECONFIG パラメータ](#)
Quartus Prime ソフトウェアの ALTPLL_RECONFIG IP コアパラメータ設定の詳細について説明します。

2.3.12 クロック・スイッチオーバー

クロック・スイッチオーバー機能により、PLL は 2 つの基準入カクロックを切り換えることができます。この機能はクロックの冗長に、あるいはデュアル・クロック・ドメイン・アプリケーション(前のクロックが動作を停止した場合に、冗長クロックをオンにするシステム)に使用します。デザインは、クロックがそれ以上トグルしないとき、またはユーザー・コントロール信号 `clkswitch` に基づいて自動的にクロック・スイッチオーバーを実行することができます。

MAX 10 PLL では、以下のクロック・スイッチオーバー・モードがサポートされます。

- 自動スイッチオーバー・クロック・センス回路が現在のリファレンス・クロックをモニタします。現在のリファレンス・クロックがトグルを停止した場合、リファレンス・クロックは自動的に `inclk0` クロックまたは `inclk1` クロックに切り替わります。
- マニュアル・クロック・スイッチオーバー—`clkswitch` 信号によってクロック・スイッチオーバーを制御します。`clkswitch` 信号がロジック Low からロジック High になり、3 クロック・サイクル以上 High の状態が続く場合、PLL へのリファレンス・クロックは `inclk0` から `inclk1` に、またはその逆に切り替わります。
- マニュアル・オーバーライドの自動スイッチオーバー—このモードは自動スイッチオーバーとマニュアル・クロック・スイッチオーバーを組み合わせたものです。`clkswitch` 信号が High になると、自動クロック・スイッチオーバー機能をオーバーライドします。`clkswitch` 信号が High である限り、それ以上のスイッチオーバー動作はブロックされます。

関連情報

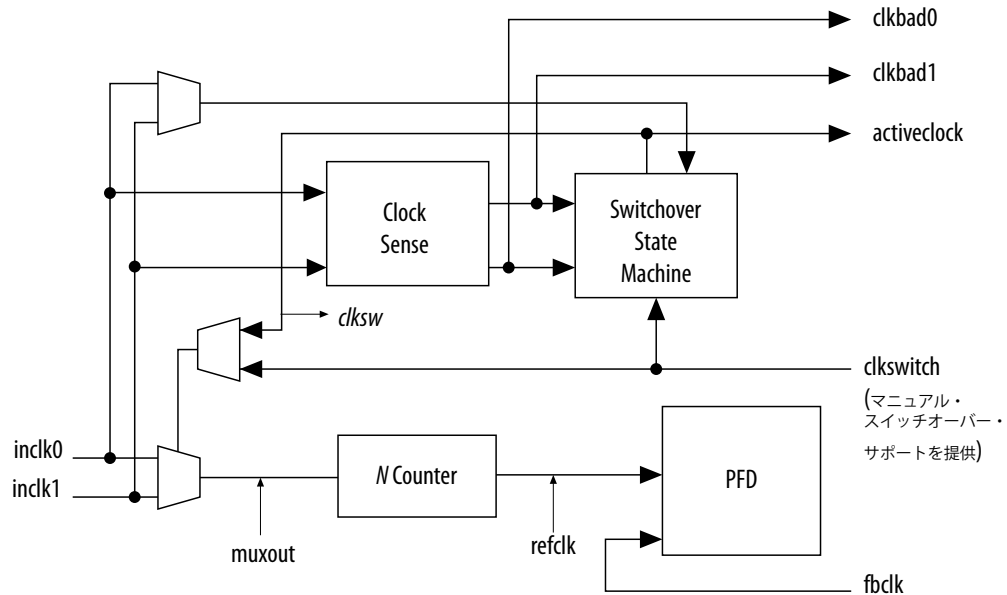
- 36 ページの [ガイドライン:クロック・スイッチオーバー](#)
- 53 ページの [クロック・スイッチオーバーのパラメータ設定](#)

2.3.12.1 自動クロック・スイッチオーバー

MAX 10 PLL は、完全にコンフィギュレーション可能なクロック・スイッチオーバー機能をサポートしません。

図 -20: 自動クロック・スイッチオーバー回路のブロック図

次の図は、PLL に組み込まれた自動スイッチオーバー回路のブロック図を示しています。



現在のリファレンス・クロックが存在しない場合、クロック・センス・ブロックは自動的に PLL リファレンスのバックアップ・クロックに切り換わります。デザイン内の PLL の inclk1 ポートに接続することにより、クロック・ソースをバックアップ・クロックで選択することができます。

このクロック・スイッチオーバー回路は、PLL から 3 つのステータス信号 (clkbad[0]、clkbad[1]、および activeclock) を送信し、カスタム・スイッチオーバー回路をロジック・アレイに実装します。

自動スイッチオーバー・モードでは、clkbad[0] 信号と clkbad[1] 信号は 2 つのクロック入力のステータスを示します。clkbad[0] および clkbad[1] 信号がアサートされると、クロック・センス・ブロックは対応するクロック入力が入力を停止したことを検知します。inclk0 と inclk1 間の周波数差が 20% を超える場合、これら 2 つの信号は無効です。

activeclock 信号は、2 つのクロック入力 (inclk0 または inclk1) のどちらかが PLL のリファレンス・クロックとして選択されているかを示します。2 つのクロック入力の周波数差が 20% を超える場合、activeclock 信号が唯一有効なステータス信号となります。

注意: 入力クロックのグリッチにより、入力クロック間の周波数差が 20% 以上になることがあります。

PLL の現在のリファレンス・クロックが入力を停止した場合、スイッチオーバー回路を使用して、同じ周波数で動作する inclk0 から inclk1 に自動的に切り替えます。この自動スイッチオーバーでは、inclk0 クロックと inclk1 クロックの 2 つのクロックのうちいずれかが停止し、他方が使用可能ときには、これらを何回でも切り換えることができます。

たとえば、リファレンス・クロックと同じ周波数の冗長クロックが必要なアプリケーションでは、スイッチオーバー・ステート・マシンはマルチプレクサ選択入力を制御する信号 (clksw) を生成します。この場合、inclk1 が PLL のリファレンス・クロックになります。

自動クロック・スイッチオーバー・モードを使用する場合、次の条件を満たしている必要があります。

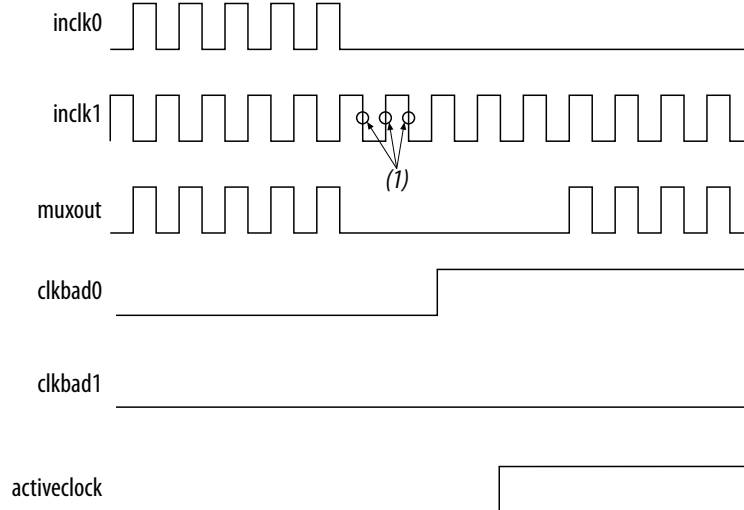
- FPGA がコンフィギュレーションされる際、両方のクロック入力の実行されている。
- 2つのクロック入力の周期の差が 20%未満である。

現在のクロック入力がトグルを停止し、他のクロックもトグルしていない場合、スイッチオーバーは開始されず、`clkbad[0..1]` 信号は無効です。両方のクロック入力の周波数が異なり、周期の差が 20%以内である場合、クロック・センス・ブロックはクロックがトグルを停止したことを検知します。他方で、PLL はスイッチオーバーが完了した後にロックを喪失し、再ロックの時間を必要とすることがあります。

注意: Intel は、クロック切り替えを使用しているときに、PLL の入力クロックと出力クロックの間の位相関係を維持するために、`areset` 信号を使用して PLL をリセットすることを推奨します。

図 -21: クロック検出喪失後の自動スイッチオーバーの例

次の図は、自動スイッチオーバー・モードのスイッチオーバー機能の波形例を示しています。この例では、`inclk0` 信号は Low に保持されています。`inclk0` 信号が約 2 クロック・サイクルの間 Low に保持された後、クロック・センス回路は `clkbad[0]` 信号を High にドライブします。リファレンス・クロック信号はトグルしていないため、スイッチオーバー・スタート・マシンが `clksw` 信号を介してマルチプレクサを制御し、バックアップ・クロック `inclk1` に切り換えます。



注:

- (1) スイッチオーバーは、使用可能なクロックに応じて `inclk0` または `inclk1` の立ち下がりエッジで有効になります。この図では、スイッチオーバーは `inclk1` の立ち下がりエッジで有効になっています。

2.3.12.2 マニュアル・オーバーライドの自動スイッチオーバー

マニュアル・オーバーライドの自動スイッチオーバー・モードでは、ユーザー制御またはシステム制御の切り換え条件に `clkswitch` 信号を使用することができます。このモードは、同じ周波数での切り換え、または異なる周波数の入力間での切り換えに使用可能です。

たとえば、`inclk0` が 66 MHz で `inclk1` が 200 MHz である場合、`clkswitch` 信号を使用してスイッチオーバーを制御する必要があります。自動クロック・センス回路は周波数の差が 20%を超えるクロック入力 (`inclk0` および `inclk1`) 周波数をモニタすることはできません。

この機能は、クロック・ソースがバックプレーン上の複数のカードから生じていて、動作の周波数間でシステム制御のスイッチオーバーを必要とする場合に役立ちます。

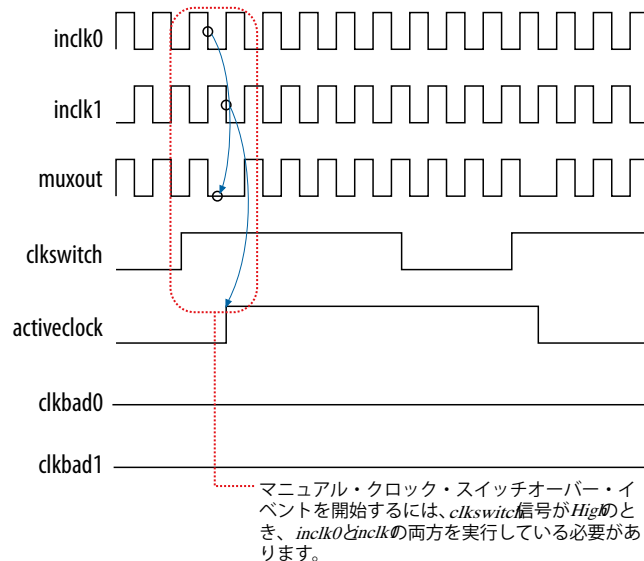


VCO が推奨される周波数範囲で動作するよう、バックアップ・クロック周波数を選択し、M、N、および C カウンタを設定する必要があります。

次の図は、clkswitch 信号で制御されるクロック・スイッチオーバーの波形を示しています。この例では、両方のクロック・ソースが動作し、inclk0 はリファレンス・クロックとして選択されています。clkswitch 信号が High になると、スイッチオーバー・シーケンスを開始します。inclk0 の立ち下りエッジでは、カウンタのリファレンス・クロックである muxout がゲート・オフされ、クロックのグリッチを防ぎます。inclk1 の立ち下りエッジでは、リファレンス・クロックのマルチプレクサが PLL 基準を inclk0 から inclk1 に切り替えます。activeclock 信号は、現在 PLL (inclk1) に信号を供給しているクロックを示すためにアサートされます。

マニュアル・スイッチオーバーの自動オーバーライド・モードでは、activeclock 信号は clkswitch 信号を反映します。マニュアル切り換えの間、両方のクロックが機能しているため、clkbad 信号が High になることはありません。また、スイッチオーバー回路はポジティブ・エッジ・センシティブであるので、clkswitch 信号の立ち下りエッジは回路を inclk1 から inclk0 に再度切り替えることはありません。clkswitch 信号が再び High になると、このプロセスを繰り返します。

図 -22: clkswitch(マニュアル)コントロールを使用したクロック・スイッチオーバーの例



clkswitch 信号と自動スイッチは、切り換えられているクロックが使用可能な場合のみ機能します。クロックが使用できない場合、ステート・マシンはクロックが使用可能になるまで待機します。

2.3.12.3 マニュアル・クロック・スイッチオーバー

マニュアル・クロック・スイッチオーバー・モードでは、clkswitch 信号は、inclk0 または inclk1 のどちらかが PLL の入力クロックとして選択されるかを制御します。デフォルトでは inclk0 が選択されています。

クロック・スイッチオーバー・イベントは、clkswitch 信号がロジック Low からロジック High に遷移され、3inclk サイクル以上 High の状態が保持されているときに開始されます。別のスイッチオーバー・イベントを実行するには、clkswitch 信号を再び Low に戻す必要があります。別のスイッチオー

バー・イベントが必要ない場合は、最初の切り替えの後、clkswitch 信号をロジック High の状態のままにしておくことができます。少なくとも 3inclk サイクルの間 clkswitch 信号を High にパルスすると、別のスイッチオーバー・イベントが実行されます。

inclk0 と inclk1 の周波数が異なり、常に動作している場合、clkswitch 信号が High である最少時間は、inclk0 と inclk1 の周波数が近い方のクロック・サイクルで 3 サイクル以上である必要があります。

2.3.13 PLL のカスケード接続

関連情報

35 ページの [ガイドライン:PLL のカスケード接続](#)

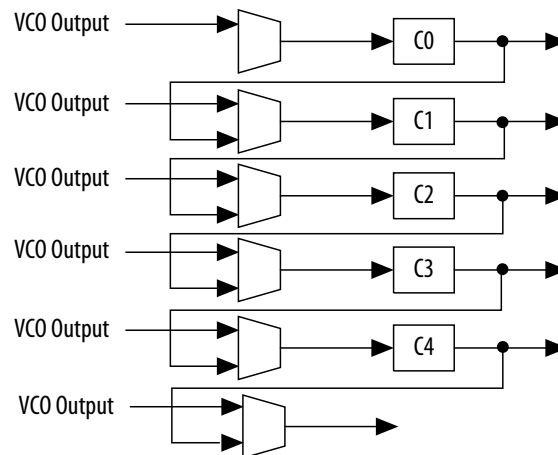
2.3.13.1 PLL 間のカスケード接続

クロック・ネットワークを使用して、2 つの PLL をカスケード接続します。デザインで PLL をカスケード接続する場合、ソース(アップストリーム)PLL は狭帯域幅設定を有し、デスティネーション(ダウンストリーム)PLL は広帯域幅設定を有している必要があります。

2.3.13.2 カウンタ間のカスケード接続

MAX 10 PLL は、512 を超えるカウンタを作成するためのポストスケール・カウンタのカスケード接続をサポートします。これは、ある C カウンタの出力を次の C カウンタの入力に供給することによって実装されます。

図 -23: カウンタ間のカスケード接続



カウンタをカスケード接続して高周波数 VCO クロックのより大きな分周を実装する場合、カスケード接続されたカウンタは個々のカウンタ設定の積を有する 1 つのカウンタとして動作します。

たとえば、C0 = 4 で C1 = 2 の場合、カスケード接続された値は $C0 \times C1 = 8$ となります。

Quartus Prime ソフトウェアは、カスケード接続するすべてのポストスケール・カウンタ値をコンフィギュレーション・ファイルで自動的に設定します。ポストスケール・カウンタのカスケード接続は、PLL リコンフィギュレーションを使用して実行することはできません。



2.3.14 PLL リコンフィギュレーション

PLL は、いくつかの分周カウンタと異なる VCO 位相タップを使用して周波数合成と位相シフトを実行します。MAX 10 PLL では、両方のカウンタ設定をリコンフィギュレーションし、PLL 出力クロックをリアルタイムで位相シフトすることができます。また、チャージ・ポンプとループ・フィルタ・コンポーネントを変更することも可能ですが、PLL の帯域幅に動的に影響します。

以下の PLL コンポーネントは、リアルタイムでコンフィギュレーションすることができます。

- プリスケール・カウンタ(N)
- フィードバック・カウンタ(M)
- ポストスケール出力カウンタ(C0 ~ C4)
- チャージ・ポンプ電流(I_{CP})
- ループ・フィルタ・コンポーネント(R, C)

これらの PLL コンポーネントを使用して、FPGA 全体をリコンフィギュレーションすることなく、以下の設定をリアル・タイムで更新することができます。

- 出力クロック周波数
- PLL 帯域幅
- 位相シフト

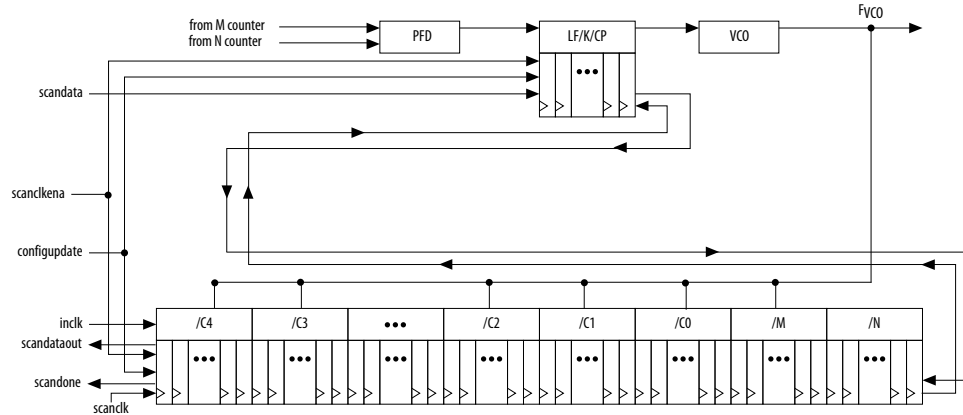
PLL をリアルタイムでリコンフィギュレーションする機能は、複数の周波数で動作するアプリケーションにおいて便利な機能となります。また、プロトタイピング環境でも有用な機能であり、PLL 出力周波数をスイープし、出力クロック位相を動的に調整することができます。

たとえば、被試験デバイスの要件に応じて、テスト・パターンを生成するシステムが 75 MHz または 150MHz でパターンを生成して送信する必要があります。PLL コンポーネントをリアルタイムでリコンフィギュレーションすることにより、このような 2 つの出力周波数を数マイクロ秒以内に切り換えることができます。

また、この機能は、PLL 出力クロックの位相シフトを変更することによって clock-to-out (t_{CO}) 遅延をリアルタイムで調整するために使用することもできます。この手法により、新しい PLL 設定でコンフィギュレーション・ファイルを再生成する必要がなくなります。

図 -24: PLL リンコンフィグレーション・スキャン・チェーン

次の図は、新しい設定をシリアル・シフト・レジスタ・チェーンまたはスキャン・チェーンにシフトすることによる PLL カウンタ設定の動的な調整を示しています。シリアル・データは、scandata ポートを通じてスキャン・チェーンにシフトし、シフト・レジスタは scanc1k によってロックされます。scanc1k の最大周波数は 100 MHz です。データの最後のビットをシフトした後、1scanc1k クロック・サイクル以上 configupdate 信号をアサートすると、スキャン・レジスタのデータで PLL コンフィグレーション・ビットを同期的に更新します。



カウンタ設定は、個々のカウンタのクロック周波数に同期して更新されます。そのため、すべてのカウンタが同時に更新されるわけではありません。

ダイナミック・リコンフィグレーション手法では、16 進形式ファイル (.hex) やメモリ初期化ファイル (.mif) といったコンフィグレーション・ファイルを使用します。これらのファイルは ALTPLL_RECONFIG IP コアと併せて使用され、ダイナミック・リコンフィグレーションを実行します。

関連情報

- 37 ページの [ガイドライン:PLL リンコンフィグレーションにおける.mif ストリーミング](#)
- 41 ページの [PLL ダイナミック・リコンフィグレーションの実装](#)
- 54 ページの [PLL ダイナミック・リコンフィグレーションのパラメータ設定](#)
Quartus Prime ソフトウェアの ALTPLL IP コアパラメータ設定に関する詳細情報を提供します。
- 59 ページの [ALTPLL_RECONFIG パラメータ](#)
Quartus Prime ソフトウェアの ALTPLL_RECONFIG IP コアパラメータ設定に関する詳細情報を提供します。



3 MAX 10 のクロッキングおよび PLL デザイン検討事項

3.1 クロック・ネットワークのデザイン検討事項

3.1.1 ガイドライン:クロック・イネーブル信号

Intel では、クロックソースを PLL または GCLK に切り替えるときに `clkena` 信号を使用することを推奨しています。推奨されるシーケンスは次のとおりです。

1. `clkena` 信号をディアサートして、プライマリ出カクロックを無効にします。
2. クロック・コントロール・ブロックのダイナミック選択信号を使用して、セカンダリ・クロックに切り換えます。
3. セカンダリ・クロックの一部のクロック・サイクルが `clkena` 信号の再アサート前に通過できるようにします。セカンダリ・クロックを有効にするまで待機するクロック・サイクルの正確な数はデザインによって異なります。異なるクロック・ソースを切り替える際、グリッチのない移行を確実にするためにカスタム・ロジックをビルドすることができます。

関連情報

- 11 ページの [クロック・イネーブル信号](#)
- 50 ページの [altclkctrl パラメータ](#)
- 51 ページの [altclkctrl ポートおよび信号](#)

3.1.2 ガイドライン:接続制限

以下のガイドラインでは、`inclck` 入力を駆動することができる信号ソースに関連した制限事項について説明します。

- `clkselect` ポートと一致する `inclck` ポートを使用する必要があります。
- 複数の入力ソースを使用している場合、`inclck` ポートは専用クロック入力ピンと PLL クロック出力によってのみ駆動することができます。
- クロック・コントロール・ブロックが別のクロック・コントロール・ブロックの `inclck` ポートを供給する場合、双方のクロック・コントロール・ブロックは同等の機能のシングル・クロック・コントロール・ブロックにすることができなければなりません。
- グリッチのないスイッチオーバー機能を使用している際、切り替えられたクロックはアクティブである必要があります。このクロックがアクティブでない場合、スイッチオーバー回路はもともと選択していたクロックから移行することができません。

3.2 内蔵オシレータ・デザインの検討事項

3.2.1 ガイドライン: 接続制限

内蔵オシレータで PLL は駆動できません。

3.3 PLL のデザイン検討事項

3.3.1 ガイドライン: PLL コントロール信号

以下の条件のいずれかに当てはまる場合、デザインに `areset` 信号を含める必要があります。

- PLL リコンフィギュレーションまたはクロック・スイッチオーバーがデザインで有効になっている。
- ロック状態喪失後に、PLL 入力クロックと出力クロック間の位相関係を維持する必要がある。
- パワーアップ時、PLL への入力クロックがトグルしているか不安定である。
- 入力クロックが安定し、仕様範囲内になった後に `areset` 信号がアサートされる。

関連情報

17 ページの [PLL コントロール信号](#)

3.3.2 ガイドライン: 接続制限

同時スイッチング・ノイズ (SSN) のデザイン・ガイドラインに準拠するには、Intel は PLL への入力クロック信号と同じバンク内で終端されていない I/O を使用しないことを推奨します。

関連情報

[Guidelines: Clock and Asynchronous Control Input Signal](#)

I/O 接続制限の使用に関する詳細情報を提供します。

3.3.3 ガイドライン: セルフ・リセット

PLL のロック時間は、デバイスのパワーアップ後、PLL 出力周波数における遷移後、または PLL のリセット後に、PLL がターゲットの周波数と位相関係を達成するにあたって必要な時間です。

PLL は、以下に示すような様々な理由でロックを喪失する可能性があります。

- 入力クロックにおける過度のジッタ。
- PLL のクロック入力における過度のスイッチング・ノイズ。
- 高い出力ジッタとロック喪失の原因となる、電源からの過度のノイズ。
- PLL への入力クロックのグリッチまたは停止。
- PLL の `areset` ポートをアサートすることによる PLL のリセット。
- PLL のリコンフィギュレーションは M カウンタ、N カウンタ、または位相シフトを遷移させる可能性があり、PLL のロック喪失の原因となります。ただし、ポストスケール・カウンタへの遷移は、PLL の `locked` 信号には影響しません。
- ロック範囲仕様外の PLL 入力クロック周波数。
- PFD は `pfdena` ポートを使用してディセーブルされます。このとき、PLL の出力位相および周波数はロック・ウィンドウの外側にドリフトする傾向があります。



ALTPLL IP コアは、locked というロック信号を使用して PLL のロック・プロセスをモニタすることを可能にし、また、ロック喪失で PLL をセルフ・リセットに設定することを可能にします。

3.3.4 ガイドライン: 出力クロック

各 MAX 10 PLL は、最大 5 つの出力クロックをサポートします。出力クロック・ポートは、コア出力クロックまたは外部出力クロック・ポートとして使用することができます。コア出力クロックは FPGA コアに信号を供給し、外部出力クロックは FPGA 上の専用ピンに信号を供給します。

ALTPLL IP コアは専用出力イネーブル・ポートを有しません。PLL 出力は、PLL 出力カウンタを無効にするための areset 信号を使用して無効にすることができます。

3.3.5 ガイドライン: PLL のカスケード接続

PLL をカスケード接続する際は、以下のガイドラインを参考にしてください。

- ジッタをフィルタできるよう、プライマリ PLL を狭帯域幅に設定します。また、プライマリ PLL からジッタをトラッキングするためにセカンダリ PLL を広帯域幅に設定します。PLL 帯域幅の範囲がオーバーラップしないよう、Quartus Prime ソフトウェアのコンパイル・レポート・ファイルを確認することができます。帯域幅範囲がオーバーラップした場合、カスケード接続された PLL 方式でジッタのピーキングが生じる可能性があります。
注 Quartus Prime ソフトウェアの TimeQuest Timing Analyzer を使用して、PLL の確定的意: ジッタとスタティック位相エラー (SPE) を見積もることができます。SDC コマンド `derive_clock_uncertainty` を使用して、プロジェクト・ディレクトリに `PLLJ_PLLSPE_INFO.txt` というタイトルのレポートを生成します。その後、`set_clock_uncertainty` コマンドを使用して、クロック制約にジッタと SPE 値を追加します。
- セカンダリ PLL で正しい位相設定ができるよう、プライマリ PLL がロックされるまでセカンダリ PLL のリセット状態を維持します。
- カスケード接続された方式の PLL のいずれの `inclk` ポートも、カスケード接続された方式の PLL からのクロック出力に接続することはできません。

関連情報

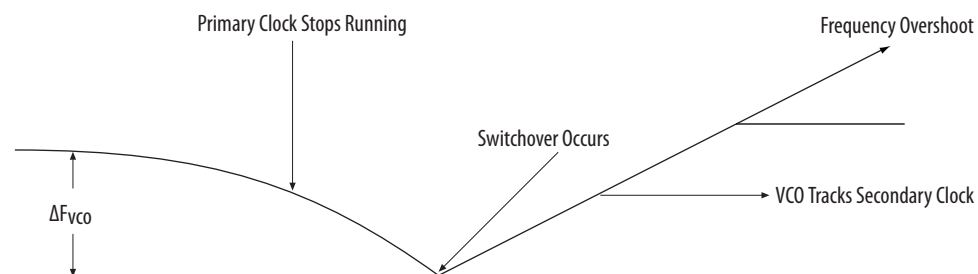
30 ページの [PLL のカスケード接続](#)

3.3.6 ガイドライン: クロック・スイッチオーバー

PLL でクロック・スイッチオーバーを使用してデザインする場合は、以下のガイドラインに従ってください。

- クロック喪失検出および自動クロック・スイッチオーバーを使用するには、`inclnk0` と `inclnk1` の周波数差が 20% 以内である必要があります。この要件を満たせない場合、`clkbad[0]` 信号と `clkbad[1]` 信号は正しく機能しません。
- マニュアル・クロック・スイッチオーバーを使用する場合、`inclnk0` と `inclnk1` 周波数差は 20% を超えても構いません。ただし、2 つのクロック・ソース (周波数、位相、またはその両方) の差によって、PLL がロックを失う可能性があります。PLL をリセットすると、入力クロックと出力クロック間の正しい位相関係が維持されます。
- マニュアル・クロック・スイッチオーバーイベントを開始するために `clkswitch` 信号が High になるとき、`inclnk0` と `inclnk1` の両方を実行している必要があります。この要件を満たせない場合、クロック・スイッチオーバーが正しく機能しません。
- クロック・スイッチオーバー機能と小さい周波数ドリフトを必要とするアプリケーションでは、狭帯域幅 PLL を使用する必要があります。狭帯域幅 PLL は、基準入力クロックの変動に対する反応が広帯域幅 PLL よりも遅くなります。また、スイッチオーバーが起こる際、狭帯域幅 PLL が出力にクロック停止を伝える速度は広帯域幅 PLL よりも遅くなります。狭帯域幅 PLL は、基準クロックのジッタをフィルタします。なお、狭帯域幅 PLL ではロック時間も長くなることに注意してください。
- スwitchオーバーが起こると、PLL が新しいクロックにロックするための有限の再同期期間が生じることがあります。PLL が再ロックするにあたって必要な正確な時間は、PLL のコンフィギュレーションによって異なります。
- PLL への入力クロックと PLL からの出力クロックの位相関係は、デザインにおいて重要です。クロック・スイッチオーバーを実行した後、10 ns の間 `areset` をアサートします。ロックされた信号 (またはゲート・ロックされた信号) が High になるのを待ってから、PLL からの出力クロックを再度イネーブルします。
- PLL の再同期化期間中にシステムが周波数変動に対応できない場合は、スイッチオーバー中にシステムをディセーブルします。`clkbad[0]` および `clkbad[1]` ステータス信号を使用して PFD をオフ (`pfdena = 0`) にすることにより、VCO は最後の周波数を維持します。また、スイッチオーバー・スタート・マシンを使用して、セカンダリ・クロックに切り換えることもできます。PFD をイネーブルすると、出力クロック・イネーブル信号 (`clkena`) がスイッチオーバーおよび再同期化期間中にクロック出力をディセーブルすることができます。ロック表示が安定した後、システムは出力クロックを再度イネーブルすることができます。
- プライマリ・クロックが失われると VCO 周波数は徐々に低下し、セカンダリ・クロックにロックすると VCO は上昇します。次の図はこの状況を図示しています。VCO がセカンダリ・クロックにロックした後、VCO 周波数で多少のオーバーシュート (過周波数状態) が生じることがあります。

図 -25: VCO のスイッチオーバー動作周波数





関連情報

- 26 ページの [クロック・スイッチオーバー](#)
- 53 ページの [クロック・スイッチオーバーのパラメータ設定](#)

3.3.7 ガイドライン:PLL リコンフィギュレーションにおける.mif ストリーミング

PLL リコンフィギュレーションで .mif ストリーミングを使用する際は、以下のガイドラインを参考にしてください。

- 10M02 デバイスは、フラッシュサイズの制限のために PLL リコンフィギュレーションで .mif ストリーミングをサポートしません。Intel は、外部フラッシュを使用することを推奨します。
- 10M04、10M08、10M16、10M25、10M40、および 10M50 デバイスは、シングル・イメージ・モードで .mif ストリーミングのみをサポートします。Intel は、デュアル・イメージ・モードに外部フラッシュを使用することを推奨します。MAX 10 デバイスは、.mif とともにデュアル・イメージ・モードと PLL リコンフィギュレーションの両方を同時に使用することはサポートしていません。

関連情報

31 ページの [PLL リコンフィギュレーション](#)

3.3.8 ガイドライン:PLL リコンフィギュレーションの scandone 信号

scandone 信号は、2 番目の PLL リコンフィギュレーションの前に Low になる必要があります。scandone 信号を Low にするには、PLL の areset 信号をアサートする必要があります。



4 MAX 10 のクロッキングおよび PLL 実装ガイド

4.1 altclkctrl IP コア

クロック・コントロール・ブロック (altclkctrl) IP コアは、クロック・コントロール・ブロックをコンフィギュレーションするためのクロック制御機能です。

altclkctrl IP コアの一般的なアプリケーションは以下のとおりです。

- ダイナミック・クロック・ソースの選択—クロック・コントロール・ブロックを使用する際、グローバル・クロック・ネットワークを駆動するダイナミック・クロック・ソースを選択することができます。
- クロック・ネットワークのダイナミック・パワーダウン—ダイナミック・クロック・イネーブルまたはディセーブル機能は、内部ロジックによるクロック・ネットワークのパワーダウンを可能にします。クロック・ネットワークがパワーダウンされると、クロック・ネットワークによって供給されるすべてのロジックはトグルしません。そのため、デバイス全体の消費電力量を低減します。

altclkctrl IP コアは、以下の機能を提供します。

- クロック・コントロール・ブロックの動作モード仕様のサポート。
- 入力クロック・ソース数の仕様のサポート。
- アクティブ High のクロック・イネーブル・コントロール入力の提供。

関連情報

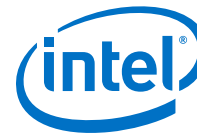
- [Introduction to Intel FPGA IP Cores](#)
パラメーター化、アップグレード、IP コアのシミュレーションを含むすべての IntelFPGA IP コアに関する基本的な情報を提供します。
- [Creating Version-Independent IP and Qsys Simulation Scripts](#)
ソフトウェアあるいは IP のバージョンのアップグレードのためのマニュアルでの更新を必要としないシミュレーション・スクリプトの作成について詳しい情報を提供します。
- [Project Management Best Practices](#)
プロジェクトと IP ファイルの効果的な管理および移植性のためのガイドラインを提供します。
- 66 ページの [MAX 10 のクロッキングおよび PLL ユーザーガイドのアーカイブ](#)
以前のバージョンの ALTCLKCTRL IP コアのユーザーガイドのリストを提供します。

4.2 ALTPLL IP コア

ALTPLL IP コアは PLL 回路を指定します。この IP コアは、PLL タイプ、動作モード、および PLL の高度な機能をコンフィギュレーションするために使用することができます。

関連情報

- [Introduction to Intel FPGA IP Cores](#)
パラメーター化、アップグレード、IP コアのシミュレーションを含むすべての IntelFPGA IP コアに関する基本的な情報を提供します。



- [Creating Version-Independent IP and Qsys Simulation Scripts](#)
ソフトウェアあるいは IP のバージョンのアップグレードのためのマニュアルでの更新を必要としないシミュレーション・スクリプトの作成について詳しい情報を提供します。
- [Project Management Best Practices](#)
プロジェクトと IP ファイルの効果的な管理および移植性のためのガイドラインを提供します。
- 66 ページの [MAX 10 のクロッキングおよび PLL ユーザーガイドのアーカイブ](#)
以前のバージョンの ALTPLL IP コアのユーザーガイドのリストを提供します。

4.2.1 PLL ロック範囲の拡張

PLL のロック範囲は、PLL がロックを達成できる最小 (Freq min lock パラメータ) の入力周波数の値と最大 (Freq max lock パラメータ) の入力周波数の値の間です。入力周波数を変更すると、PLL がロックを喪失することがあります。ただし、入力クロックが最小および最大周波数仕様の範囲内である場合、PLL はロックを達成することができます。入力周波数の値は、Quartus Prime ソフトウェアのコンパイルレポートの Fitter フォルダ、Resource Section にある PLL Summary レポートで示されます。

Quartus Prime ソフトウェアは、ロック範囲の最大化を実行する際、PLL パラメータに対し必ずしも値を取得するわけではありません。たとえば、ALTPLL パラメータ・エディタで 75 MHz の入力クロックを指定する場合、実際の PLL ロック範囲は 70 MHz から 90 MHz の間となりえます。使用中のアプリケーションが 50 MHz から 100 MHz の範囲を必要とする場合、デフォルトの PLL ロック範囲は十分ではありません。

PLL のクロック・スイッチオーバーをサポートするデバイスでは、ALTPLL IP コアのパラメータ・エディタを使用してロック範囲を最大化することができます。

PLL のロック範囲を最大化するために有効なパラメータ値を抽出するには、以下の手順を実行します。

1. 回路図エディタで、デザイン内の ALTPLL インスタンスをダブルクリックして ALTPLL パラメータ・エディタを開きます。
2. **General/Modes** ページの **What is the frequency of the inclk0 input?** で、必要な PLL ロック範囲の下限の値を入力します。
たとえば、アプリケーションが 50 MHz ~ 100 MHz のロック範囲を必要とする場合、50 MHz と入力します。
3. **Inputs/Lock** ページで、**Create output file(s) using the 'Advanced' PLL parameters** をオンにします。
4. **Clock switchover** ページで、**Create an 'inclk1' input for a second input clock** をオンにし、inclk1 の周波数としてロック範囲の上限値を入力します。
たとえば、アプリケーションが 50 MHz ~ 100 MHz のロック範囲を必要とする場合、100 MHz と入力します。
5. ALTPLL IP コア・パラメータ・エディタの残りのページで、その他のパラメータを設定します。
6. プロジェクトをコンパイルします。PLL Summary レポートに表示されるロック範囲に注意してください。コンパイルが良好に実行される場合、M 値、N 値、チャージ・ポンプ電流、ループ・フィルタ抵抗、およびループ・フィルタ・キャパシタンスといった、このレポートからの PLL のすべての値に留意します。
7. 回路図エディタで、デザイン内の ALTPLL インスタンスをダブルクリックして ALTPLL パラメータ・エディタを開きます。
8. **Clock switchover** ページで、**Create an 'inclk1' input for a second input clock** をオフにします。

9. **Finish** をクリックして、PLL のラッパー・ファイルを更新します。
10. テキスト・エディタで、PLL ラッパー・ファイルを開きます。ステップ 6 にリストされているパラメータのすべての値を修正します。変更を保存します。
 - ラッパー・ファイルが Verilog フォーマットである場合、**defparam** セクションに移動します。
 - ラッパー・ファイルが VHDL HDL である場合、**Generic Map** セクションに移動します。
11. プロジェクトをコンパイルします。
12. PLL のロック範囲が必要条件を満たしていることを確認するために、PLL Summary レポートを確認してください。変更済みの PLL は必要なロック範囲を有している必要があります。

入力クロック周波数が必要な PLL ロック範囲の上下限に極めて近い場合 (たとえば、必要なロック範囲の下限が 50 MHz で、入力クロック周波数が 50 MHz の場合)、入力クロックがジッタを有するか、周波数が 50 MHz 以下をドリフトする際に、PLL がロックを維持しないことがあります。こうした場合、入力クロック周波数が範囲の上下限から遠くなるよう、PLL のロック範囲を拡張することができます。たとえば、必要なロック範囲である 50 MHz ~ 100 MHz が PLL のロック範囲内になるよう、45 MHz ~ 105 MHz と入力することができます。

この手順を使用して所望のロック範囲を実装することができない場合、Quartus Prime ソフトウェアはエラー・メッセージを表示します。その場合、入力周波数範囲をサポートするために、PLL リコンフィギュレーションなど他のオプションを検討する必要があります。

4.2.2 アドバンスド・パラメータを有するプログラマブル帯域幅

アドバンスド・レベルのコントロールは、PLL ループ・フィルタ特性を正確にコントロールすることができます。このレベルでは、以下のアドバンスド・パラメータを明示的に選択することができます。

- チャージ・ポンプ電流 (charge_pump_current)
- ループ・フィルタ抵抗 (loop_filter_r)
- ループ・フィルタ・キャパシタンス (loop_filter_c)

このオプションは、PLL コンフィギュレーションに習熟した上級ユーザー向けです。このオプションは、パラメータについて十分理解し、それらを最適に設定することができる場合に使用することができます。生成されるファイルは、ALTPLL IP コア・パラメータ・エディタで再利用するものではありません。アドバンスド・パラメータを使用して ALTPLL IP コアの出力ファイルを指定した後、Quartus Prime コンパイラはそれらを変更することはできません。たとえば、コンパイラは最適化を実行することができません。そのため、デザインはコンパイラの改良されたアルゴリズムがもたらすメリットを得ることができません。Quartus Prime コンパイラは、より適した設定を選択したり、ALTPLL IP コア・パラメータ・エディタがデザインに対応していないことを検出した設定を変更したりすることはできません。

アドバンスド PLL パラメータを使用して出力ファイルを生成するためのパラメータ設定は、ALTPLL IP コア・パラメータ・エディタの **Inputs/Lock** ページにあります。

機能を有効にするには、**Create output file(s) using the 'Advanced' PLL parameters** をオンにします。

このオプションをオンにすると、生成される出力ファイルに PLL で使用するすべての初期カウンタ値が含まれます。これらの値は、サードパーティ・シミュレータの機能シミュレーションに使用することができます。

これらのパラメータ設定は、追加のトップレベル・ポートを作成しません。



関連情報

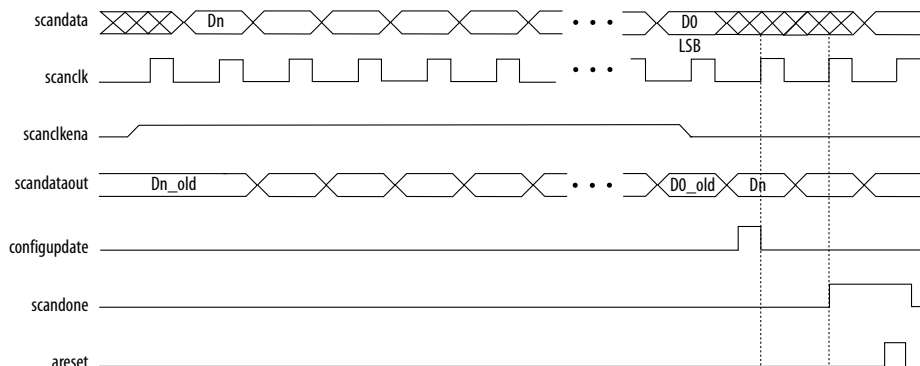
- 24 ページの [プログラマブル帯域幅](#)
- 44 ページの [チャージ・ポンプとループ・フィルタ](#)
PLL 帯域幅をリアルタイムで更新するための PLL コンポーネントに関する詳細情報を提供しません。
- 53 ページの [プログラマブル帯域幅のパラメータ設定](#)

4.2.3 PLL ダイナミック・リコンフィギュレーションの実装

PLL カウンタをリコンフィギュレーションするには、以下の手順を実行します。

1. scandata(Dn)の最初のビットにシフトする前に、scanc1k サイクル以上 scanc1kena 信号をアサートします。
2. シリアル・データ(scandata)を scanc1k の 2 番目の立ち上がりエッジのスキャン・チェーンにシフトします。
3. 144 ビットがすべてスキャン・チェーンにスキャンされた後、scanc1kena 信号をディアサートして、スキャン・チェーンのビットの偶発的なシフトを防ぎます。
4. configupdate 信号を 1scanc1k サイクルの間アサートし、PLL カウンタをスキャン・チェーンのコンテンツで更新します。
scandone 信号は High になり、PLL がリコンフィギュレーションされていることを示します。立ち下がりエッジは PLL カウンタが新しい設定に更新されたことを示します。
5. M、N、ポストスケール出力 C カウンタ、または I_{CP}、R、および C の設定を変更する場合は、areset 信号を使用して PLL をリセットします。
6. 手順 1~5 を繰り返して、PLL を何度でもリコンフィギュレーションすることができます。

図 -26: PLL リコンフィギュレーション・スキャン・チェーンの機能シミュレーション



カウンタのクロック周波数をリコンフィギュレーションする際、同じインタフェースを使用して対応するカウンタの位相シフト設定をリコンフィギュレーションすることはできません。位相シフトは、ダイナミック位相シフト・リコンフィギュレーション・インタフェースを使用してリアル・タイムでリコンフィギュレーションすることができます。クロック出力で同じ非ゼロ位相シフト設定(たとえば 90°)を維持する場合、カウンタのクロック周波数をリコンフィギュレーションした後に位相シフトをリコンフィギュレーションする必要があります。

関連情報

- 31 ページの [PLL リコンフィギュレーション](#)

4.2.3.1 ポストスケール・カウンタ(C0~C4)

ポストスケール・カウンタの逡倍値または分周値およびデューティ・サイクルは、リアルタイムでコンフィギュレーションすることができます。各カウンタは 8 ビットの High タイム設定と 8 ビットの Low タイム設定を有します。デューティ・サイクルは、全サイクル・タイム (High タイムと Low タイムの合計) に対する出力 High タイムまたは Low タイムの比率です。

ポストスケール・カウンタは 2 つのコントロール・ビットを有します。

- `rbypass`—カウンタをバイパスする
- `rselodd`—出力クロックのデューティ・サイクルを選択する

`rbypass` ビットが 1 に設定されると、カウンタはバイパスされ、1 分周になります。このビットが 0 に設定されると、PLL は High タイム・カウンタと Low タイム・カウンタに基づいて VCO 出力周波数の有効分周比を計算します。PLL は、VCO 出力クロックの立ち上がりエッジで出力クロックを High から Low に遷移させることにより、このデューティ・サイクルを実装します。

たとえば、ポストスケール分周係数が 10 の場合、50-50%のデューティ・サイクルを達成するために High および Low のカウント値をそれぞれ 5 と 5 に設定します。なお、High および Low カウント値をそれぞれ 4 と 6 に設定すると、40-60%のデューティ・サイクルを有する出力クロックを生成します。

`rselodd` ビットは、50%のデューティ・サイクルを有する VCO 出力周波数の分周係数が奇数になることを示します。PLL は、VCO 出力クロックの立ち下がりエッジで出力クロックを High から Low に遷移させることにより、このデューティ・サイクルを実装します。

たとえば、ポストスケール分周係数が 3 の場合、この分周を達成するために High および Low タイム・カウント値をそれぞれ 2 と 1 に設定します。これは 67%-33%のデューティ・サイクルを意味します。50%-50%のデューティ・サイクルが必要な場合は、奇数の分周係数でもこのデューティ・サイクルを達成するために、`rselodd` コントロール・ビットを 1 に設定する必要があります。`rselodd` = 1 に設定する場合、High タイムから 0.5 サイクルを減算し、Low タイムに 0.5 サイクルを加算します。

計算例は以下に示すとおりです。

- High タイム・カウント = 2 サイクル
- Low タイム・カウント = 1 サイクル
- `rselodd` = 1 は実質的に以下に等しくなります。
 - High タイム・カウント = 1.5 サイクル
 - Low タイム・カウント = 1.5 サイクル
 - デューティ・サイクル = (1.5/3)% High タイム・カウントと(1.5/3)% Low タイム・カウント

関連情報

23 ページの [プログラマブル・デューティ・サイクル](#)

4.2.3.2 スキャン・チェーン

MAX 10 PLL は 144 ビットのスキャン・チェーンを有します。



表 7. PLL コンポーネントの再プログラミング・ビット

ブロック名	ビット数		
	Counter	コントロール・ビット	Total
C4 ⁽⁶⁾	16	2 ⁽⁷⁾	18
C3	16	2 ⁽⁷⁾	18
C2	16	2 ⁽⁷⁾	18
C1	16	2 ⁽⁷⁾	18
C0	16	2 ⁽⁷⁾	18
M	16	2 ⁽⁷⁾	18
N	16	2 ⁽⁷⁾	18
チャージ・ポンプ	9	0	9
ループ・フィルタ ⁽⁸⁾	9	0	9
ビットの総数			144

図 -27: PLL コンポーネントのスキャン・チェーン・オーダ

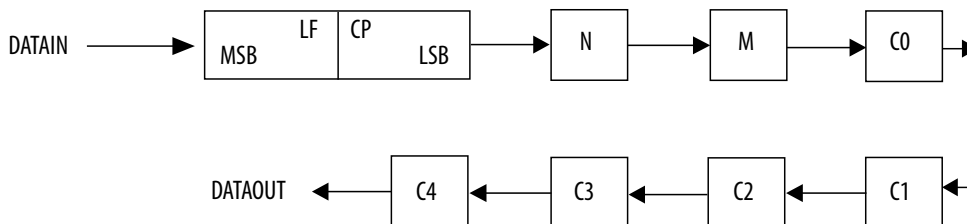
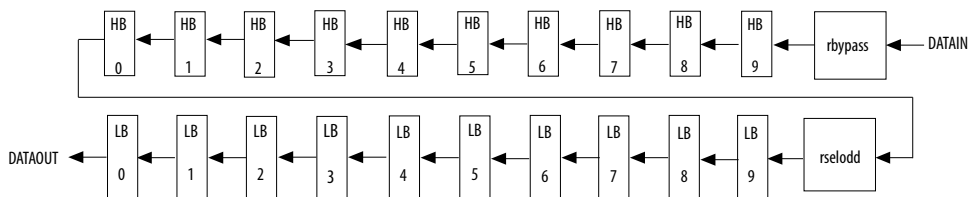


図 -28: PLL ポストスケール・カウンタのスキャン・チェーン・ビット・オーダ



- (6) C4 Low カウント値の LSB ビットは、スキャン・チェーンに最初にシフトされるビットです。
- (7) 2 つのコントロール・ビットには、カウンタをバイパスする rbypass と出カクロックのデューティ・サイクルを選択する rselodd があります。
- (8) ループ・フィルタの MSB ビットは、最後にスキャン・チェーンにシフトされるビットです。

4.2.3.3 チャージ・ポンプとループ・フィルタ

以下の設定をリコンフィギュレーションして、PLL 帯域幅をリアルタイムで更新することができます。

- チャージ・ポンプ (I_{CP})
- ループ・フィルタ抵抗 (R)
- ループ・フィルタ・コンデンサ (C)

表 8. チャージ・ポンプのビット・コントロール

CP[2]	CP[1]	CP[0]	設定 (10 進数)
0	0	0	0
0	0	1	1
0	1	1	3
1	1	1	7

表 9. ループ・フィルタの抵抗値コントロール

LFR[4]	LFR[3]	LFR[2]	LFR[1]	LFR[0]	設定 (10 進数)
0	0	0	0	0	0
0	0	0	1	1	3
0	0	1	0	0	4
0	1	0	0	0	8
1	0	0	0	0	16
1	0	0	1	1	19
1	0	1	0	0	20
1	1	0	0	0	24
1	1	0	1	1	27
1	1	1	0	0	28
1	1	1	1	0	30

表 10. ループ・フィルタの高周波数コンデンサ・コントロール

LFC[1]	LFC[0]	設定 (10 進数)
0	0	0
0	1	1
1	1	3

関連情報

- 24 ページの [プログラマブル帯域幅](#)
- 40 ページの [アドバンスド・パラメータを有するプログラマブル帯域幅](#)
- 53 ページの [プログラマブル帯域幅のパラメータ設定](#)

4.2.3.4 PLL カウンタのバイパス

PLL カウンタをバイパスすると、逓倍(M カウンタ)または分周(N、C0 ~ C4 カウンタ)の係数が 1 になります。

表 11. PLL カウンタ設定

説明	PLL スキャン・チェイン・ビット[0..8]設定								
	LSB								MSB
バイパスされる PLL カウンタ	X	X	X	X	X	X	X	X	1 ⁽⁹⁾
バイパスされない PLL カウンタ	X	X	X	X	X	X	X	X	0 ⁽⁹⁾

PLL カウンタのいずれかをバイパスするには、バイパス・ビットを 1 に設定します。他のビットの値は無視されます。

4.2.4 ダイナミック位相コンフィギュレーションの実装

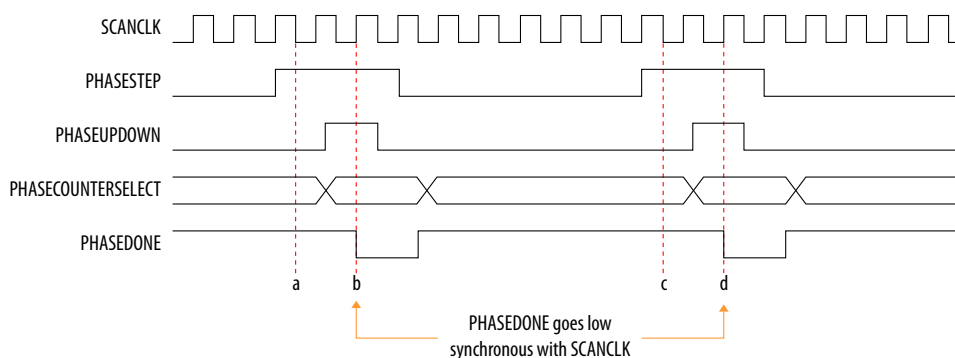
ダイナミック位相シフトを 1 ステップを実行するには、以下の手順を使用します。

1. 必要に応じて PHASEUPDOWN と PHASECOUNTERSELECT を設定します。
2. PHASESTEP を 2SCANCLK サイクル以上アサートします。PHASESTEP パルスごとに、1 位相シフトします。
3. PHASEDONE が Low になった後、PHASESTEP をデアサートします。
4. PHASEDONE が High になるのを待ちます。
5. 手順 1 ~ 4 を必要な回数だけ繰り返し、複数の位相シフトを実行します。

PHASEUPDOWN 信号と PHASECOUNTERSELECT 信号は SCANCLK に同期し、SCANCLK エッジを基準とする t_{su} および t_h 要件を満たす必要があります。

ダイナミック位相シフトは何度でも繰り返すことができます。たとえば、VCO 周波数が 1,000 MHz、出力クロック周波数が 100 MHz に設定されているデザインでは、40 ダイナミック位相シフト(それぞれ 125 ps の位相シフトを生成)を実行すると、出力クロックを 180°シフトします。これは 5ns の位相シフトです。

図 -29: ダイナミック位相シフトのタイミング図



(9) バイパス・ビット

PHASESTEP 信号は、SCANCLK (a, c) のネガティブ・エッジでラッチされ、アサートされた状態を 2SCANCLK サイクル以上維持する必要があります。PHASEDONE が Low になった後、PHASESTEP をディアサートします。

PHASESTEP がラッチされた後の 2 番目の SCANCLK の立ち上がりエッジ (b, d) では、PHASEUPDOWN と PHASECOUNTERSELECT の値がラッチされます。PLL は指定されたカウンタに対して、指定された方向でダイナミック位相シフトを開始します。

PHASEDONE 信号はディアサートされ、2 番目の立ち上がりエッジ (b, d) で SCANCLK に同期します。そして、PLL がダイナミック位相シフトを完了するまで Low の状態を維持します。PHASEDONE の Low タイムは、VCO 周波数と SCANCLK 周波数によって 1SCANCLK サイクル以上であったり、以下であったりします。

PHASEDONE 信号が Low から High に遷移した後、別のダイナミック位相シフトを実行することができます。PHASESTEP パルスごとに、1 位相シフトします。PHASESTEP パルスは少なくとも 1SCANCLK サイクル以上離す必要があります。

関連情報

- 24 ページの [プログラマブル位相シフト](#)
- 54 ページの [ダイナミック位相コンフィギュレーションのパラメータ設定](#)
Quartus Prime ソフトウェアの ALTPLL IP コアパラメータ設定に関する詳細情報を提供します。
- 59 ページの [ALTPLL_RECONFIG パラメータ](#)
Quartus Prime ソフトウェアの ALTPLL_RECONFIG IP コアパラメータ設定に関する詳細情報を提供します。
- 54 ページの [PLL ダイナミック・リコンフィギュレーションのパラメータ設定](#)
Quartus Prime ソフトウェアの ALTPLL IP コアパラメータ設定に関する詳細情報を提供します。
- 59 ページの [ALTPLL_RECONFIG パラメータ](#)
Quartus Prime ソフトウェアの ALTPLL_RECONFIG IP コアパラメータ設定に関する詳細情報を提供します。

4.2.4.1 ダイナミック位相コンフィギュレーション・カウンタ選択

表 12. 位相カウンタ選択マッピング

PLL カウンタ選択	PHASECOUNTERSELECT [2]	[1]	[0]
すべての出力カウンタ	0	0	0
M カウンタ	0	0	1
C0 カウンタ	0	1	0
C1 カウンタ	0	1	1
C2 カウンタ	1	0	0
C3 カウンタ	1	0	1
C4 カウンタ	1	1	0

関連情報

24 ページの [プログラマブル位相シフト](#)



4.2.4.2 アドバンスド・パラメータを有するダイナミック位相コンフィギュレーション

ALTPLL IP コアで取得することができる最高の位相シフト・ステップ分解能は、VCO 周期の 1/8 です。VCO 周波数がサポートされる VCO 範囲の下限である場合、位相シフト・ステップ分解能はデザインに適したものよりも大きくなる場合があります。

位相シフトの分解能は、PLL のダイナミック位相リコンフィギュレーション機能を使用して変更することができます。ダイナミック位相リコンフィギュレーション機能を有効にせずに位相シフトの分解能を変更する場合は、以下の手順を実行します。

1. ALTPLL インスタンスを作成します。必ずターゲット・デバイスのスピード・グレードと PLL タイプを指定してください。
2. **PLL Reconfiguration** ページで、**Create optional inputs for dynamic phase reconfiguration** と **Enable phase shift step resolution** をオンにします。
3. **Output Clocks** ページで、必要な出力クロックごとに所望の位相シフトを設定します。表示されるすべての内部 PLL 設定に留意してください。
4. **Bandwidth/SS** ページで、**More Details** をクリックして内部 PLL 設定を確認します。表示されるすべての設定に留意してください。
5. **Inputs/Lock** ページで、**Create output file(s) using the 'Advanced' PLL Parameters** をオンにします。
6. **PLL Reconfiguration** ページに戻り、**Create Optional Inputs for Dynamic Phase Reconfiguration** をオフにします。
7. **Finish** をクリックして、PLL のインスタンス・ファイルを生成します。
アドバンスド・パラメータを使用している場合、PLL のラッパー・ファイル(< ALTPLL_instantiation_name > .v または < ALTPLL_instantiation_name > .vhd) は PLL パラメータを識別することが可能なフォーマットで書かれます。パラメータは、VHDL ファイルの **Generic Map** セクション、または Verilog ファイルの defparam セクションにリストされます。
8. PLL のインスタンス・ラッパー・ファイルを開き、**Generic Map** セクションまたは defparam セクションのいずれかを検索します。
9. 手順 3 と 4 で留意した設定に合うよう設定を変更します。
10. PLL インスタンス・ラッパー・ファイルを保存し、デザインをコンパイルします。
11. コンパイル・レポートの Fitter フォルダ内、Resource セクションにある PLL Usage レポートで、出力クロック周波数および位相が正しいことを確認します。

この手法を使用して、デザインの設定を最適化するために、ALTPLL IP コア・パラメータ・エディタで提供される有効な PLL パラメータを適用することができます。

Advanced PLL Parameters オプションを使用して PLL ラッパー・ファイルの編集を手動で行わない場合、ダイナミック・フェーズ・リコンフィギュレーション・オプションをイネーブルにした状態で関連する入力ポートの phasecounterselect[3..0]、phaseupdown、phasesstep および scanclk を定数に接続するという別の方法もあります。

関連情報

24 ページの [プログラマブル位相シフト](#)

4.3 ALTPLL_RECONFIG IP コア

ALTPLL_RECONFIG IP コアは、PLL のダイナミック・リアルタイム・リコンフィギュレーションを容易にするためにリコンフィギュレーション・ロジックを実装しています。この IP コアを使用して、FPGA 全体をリコンフィギュレーションすることなく、出力クロック周波数、PLL 帯域幅、および位相シフトをリアルタイムで更新することができます。

ALTPLL_RECONFIG IP コアは、クロックおよびその他の周波数信号の周波数と位相シフトの動的な遷移をサポートする必要があるデザインで使用します。また、この IP コアは PLL の出力周波数をスワイプし、出力クロック位相を動的に調整することを可能にするため、プロトタイピング環境でも有用です。なお、出力クロック位相シフトを遷移させることにより、リアルタイムで clock-to-output (t_{CO}) 遅延を調整することもできます。このアプローチにより、新しい PLL 設定でコンフィギュレーション・ファイルを再生成する必要がなくなります。この動作にはダイナミック位相シフトが必要です。

関連情報

- [Introduction to Intel FPGA IP Cores](#)
パラメーター化、アップグレード、IP コアのシミュレーションを含むすべての IntelFPGA IP コアに関する基本的な情報を提供します。
- [Creating Version-Independent IP and Qsys Simulation Scripts](#)
ソフトウェアあるいは IP のバージョンのアップグレードのためのマニュアルでの更新を必要としないシミュレーション・スクリプトの作成について詳しい情報を提供します。
- [Project Management Best Practices](#)
プロジェクトと IP ファイルの効果的な管理および移植性のためのガイドラインを提供します。
- 66 ページの [MAX 10 のクロッキングおよび PLL ユーザーガイドのアーカイブ](#)
以前のバージョンの ALTPLL_RECONFIG IP コアのユーザーガイドのリストを提供します。

4.3.1 リソース使用率レポートの取得

ALTPLL_RECONFIG IP コアのリソース使用率および性能についての詳細は、Quartus Prime ソフトウェアのコンパイル・レポートを参照してください。

Quartus Prime ソフトウェアで ALTPLL_RECONFIG IP コアのコンパイル・レポートを確認するには、以下の手順を実行します。

1. Processing メニューで、**Start Compilation** をクリックしてフル・コンパイルを実行します。
2. デザインをコンパイルした後、Processing メニューで **Compilation Report** をクリックします。
3. Contents ブラウザの Table で、“+”アイコンをクリックして Fitter フォルダを展開します。
4. **Fitter** フォルダで **Resource section** を展開し、**Resource Usage Summary** を選択してリソース使用量を表示します。
5. **Fitter** フォルダで **Resource section** を展開し、**Resource Utilization by Entity** を選択してリソース使用率を表示します。

4.4 内蔵オシレーター IP コア

内蔵オシレーター IP コアは、デバイスの内蔵オシレータの周波数を指定します。



関連情報

- [Introduction to Intel FPGA IP Cores](#)
パラメーター化、アップグレード、IP コアのシミュレーションを含むすべての IntelFPGA IP コアに関する基本的な情報を提供します。
- [Creating Version-Independent IP and Qsys Simulation Scripts](#)
ソフトウェアあるいは IP のバージョンのアップグレードのためのマニュアルでの更新を必要としないシミュレーション・スクリプトの作成について詳しい情報を提供します。
- [Project Management Best Practices](#)
プロジェクトと IP ファイルの効果的な管理および移植性のためのガイドラインを提供します。
- 66 ページの [MAX 10 のクロッキングおよび PLL ユーザーガイドのアーカイブ](#)
以前のバージョンの 内部オシレーター IP コアのユーザーガイドのリストを提供します。



5 altclkctrl IP コア・リファレンス

5.1 altclkctrl パラメータ

表 13. デバイスの altclkctrl IP コアのパラメータ

この表は、MAX 10 デバイスに適用可能な IP コアのパラメータをリストしています。

パラメータ	Value	説明
How do you want to use the altclkctrl	For global clock または For external path	altclkctrl のバッファリング・モードを指定します。以下のモードから選択することができます。 <ul style="list-style-type: none"> For global clock—クロック信号がチップのすべてのパスに同じスキューの量で到達することを可能にします。入力ポート clkselect を選択して、4 つのクロック入力間を切り替えることができます。 For external path—PLL の出力から専用クロック出力ピンへのクロック・パスを示します。1 つのクロック出力のみ受け入れます。
How many clock inputs would you like?	1、2、3、または 4	クロック・コントロール・ブロックの入力クロック・ソース数を指定します。最大 4 つのクロック入力を選択することができます。 For global clock オプションを選択する場合のみ、クロック入力の数を変更することができます。
Create 'ena' port to enable or disable the clock network driven by this buffer	On または Off	クロック・ネットワークを有効または無効にするためにアクティブ High のクロック・イネーブル信号を作成する場合、このオプションをオンにします。
Ensure glitch-free switchover implementation	On または Off	複数のクロック入力を使用する際、グリッチのないスイッチオーバーを実装するには、このオプションをオンにします。 他のソースに切り替える前に、現在選択されているクロックが実行していることを確認する必要があります。選択されているクロックが実行していない場合、グリッチのないスイッチオーバーの実装は新しいクロック・ソースに切り替えることができません。 デフォルトでは、clkselect ポートは 00 で設定されています。読み込まれるには、クロックは clkselect ポートの値に inclk0x を適用する必要があります。

関連情報

- 8 ページの [グローバル・クロック・コントロール・ブロック](#)
- 10 ページの [グローバル・クロック・ネットワークのパワーダウン](#)
- 11 ページの [クロック・イネーブル信号](#)
- 33 ページの [ガイドライン:クロック・イネーブル信号](#)



5.2 altclkctrl ポートおよび信号

表 14. デバイスの altclkctrl 入力ポート

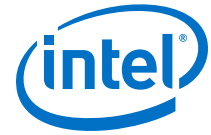
ポート名	条件	説明
clkselect[]	オプション	<p>クロック・バッファによって駆動されるクロック・ネットワークを駆動するために、クロック・ソースを動的に選択する入力です。</p> <p>入力ポート[1 DOWNTO 0]幅です。</p> <p>これを省略する場合、デフォルトは GND です。</p> <p>この信号が接続される場合、クロック・コントロール・ブロックはグローバル・クロック・ネットワークのみを駆動することができます。</p> <p>以下のリストは、バイナリ値の信号選択を示しています。</p> <ul style="list-style-type: none"> • 00—inclk[0] • 01—inclk[1]
ena	オプション	<p>クロック・バッファのクロック・イネーブルです。</p> <p>これを省略する場合、デフォルト値は V_{CC} です。</p>
inclk[]	必須	<p>クロック・バッファのクロック入力です。</p> <p>入力ポート[1 DOWNTO 0]幅です。</p> <p>最大 2 つのクロック入力、inclk[1..0]を指定することができます。</p> <p>クロック・ピン、PLL からのクロック出力、およびコア信号によって inclk[]ポートを駆動することができます。</p> <p>複数のクロック入力はグローバル・クロック・ネットワークでのみサポートされます。</p>

表 15. デバイスの altclkctrl 出力ポート

ポート名	条件	説明
outclk	必須	クロック・バッファの出力です。

関連情報

- 8 ページの [グローバル・クロック・コントロール・ブロック](#)
- 10 ページの [グローバル・クロック・ネットワークのパワーダウン](#)
- 11 ページの [クロック・イネーブル信号](#)
- 33 ページの [ガイドライン:クロック・イネーブル信号](#)



6 ALTPLL IP コア・リファレンス

6.1 ALTPLL パラメータ

次の表は、MAX 10 デバイスに適用可能な IP コアのパラメータをリストしています。

6.1.1 動作モードのパラメータ設定

PLL の動作モードは、ALTPLL IP コア・パラメータ・エディタの **General/Modes** ページで設定することができます。

表 16. 動作モードのパラメータ・エディタ設定

パラメータ	Value	説明
Which device speed grade will you be using?	Any, 7, または 8	デバイスを最速スピードで使用していない場合、スピード・グレードを指定します。値が低いほどスピード・グレードは速くなります。
What is the frequency of the inlock0 input?	—	入力クロック信号の周波数を指定します。
Use the feedback path inside the PLL	In normal mode, In source-synchronous compensation mode, In zero-delay buffer mode, または With no compensation	使用する動作モードを指定します。 Source-synchronous mode と Zero-delay buffer mode では、IP コアに適切なモードを設定することに加えて、アサインメント・エディタを使用して PLL Compensation アサインメントを割り当てる必要があります。このアサインメントにより、出力ピンをゼロ遅延バッファ・モードの PLL の補償ターゲットとして指定する、あるいは、入力ピンまたは入力ピンのグループをソース・シンクロナス・モードの PLL の補償ターゲットとして指定することができます。
Which output clock will be compensated for?	C0, C1, C2, C3, または C4	補償する PLL 出力ポートを指定します。 ドロップ・ダウン・リストには、選択したデバイスのすべての出力クロック・ポートが含まれます。適切な出力クロックの選択は、選択した動作モードによって異なります。 たとえば、ノーマル・モードではコア出力クロックを選択します。ゼロ遅延バッファ・モードでは外部出力クロックを選択します。

関連情報

18 ページの [クロック・フィードバック・モード](#)

6.1.2 PLL コントロール信号のパラメータ設定

コントロール信号のパラメータ設定は、ALTPLL IP コア・パラメータ・エディタの **Inputs/Lock** ページにあります。

使用可能なオプションから作成するコントロール信号をオンにします。

関連情報

17 ページの [PLL コントロール信号](#)



6.1.3 プログラマブル帯域幅のパラメータ設定

ALTPLL IP コアの帯域幅は、ALTPLL IP コア・パラメータ・エディタの **Bandwidth/SS** ページでコンフィギュレーションすることができます。

表 17. 帯域幅コンフィギュレーションのパラメータ・エディタ設定

パラメータ	Value	説明
Auto	—	ALTPLL パラメータ・エディタは、目的の PLL 設定を達成するにあたって最適な帯域幅の値を選択します。場合によっては、帯域幅の値が Low および High プリセット範囲外になることもあります。 プログラマブル帯域幅機能をクロック・スイッチオーバー機能を併せて使用して、必要な PLL 出力設定を取得することができます。スペクトラム拡散機能を有効にする場合は、帯域幅を Auto に設定する必要があります。
Preset	Low	狭帯域幅の PLL では、ジッタ除去は向上しますがロック時間が遅くなります。
	Medium	中帯域幅の PLL では、ジッタ除去とロック時間は均衡します。
	High	広帯域幅の PLL では、ロック時間はより高速になりますがより多くのジッタをトラッキングします。

Bandwidth/SS ページの右側の表では、以下のコンポーネントの値が表示されています。

- Charge pump current
- Loop filter resistance
- Loop filter capacitance
- M カウンタ

これらのパラメータ設定は、追加のトップレベル・ポートを作成しません。

関連情報

- 24 ページの [プログラマブル帯域幅](#)
- 40 ページの [アドバンスド・パラメータを有するプログラマブル帯域幅](#)
- 44 ページの [チャージ・ポンプとループ・フィルタ](#)

6.1.4 クロック・スイッチオーバーのパラメータ設定

クロック・スイッチオーバー機能のパラメータ設定は、ALTPLL IP コア・パラメータ・エディタの **Clock switchover** ページにあります。

表 18. クロック・スイッチオーバーのパラメータ・エディタ設定

パラメータ	Value	説明
Create an 'inclk1' input for a second input clock	On または Off	スイッチオーバー機能を有効にするには、このオプションをオンにします。inclk0 信号は、デフォルトでは ALTPLL IP コアのプライマリ入カクロック信号です。
Create a 'clkswitch' input to manually select between the input clocks	—	このオプションは、マニュアル・クロック・スイッチオーバー・モードで選択しません。
Allow PLL to automatically control the switching between input clocks	—	このオプションは、自動クロック・スイッチオーバー・モードで選択します。自動スイッチオーバーは、ロックの喪失時、あるいは inclk0 信号がトグルを停止した時に開始されます。
<i>continued...</i>		

パラメータ	Value	説明
Create a 'clkswitch' input to dynamically control the switching between input clocks	On または Off	このオプションは、マニュアル・オーバーライドの自動クロック・スイッチオーバー・モードでオンにします。自動スイッチオーバーは、ロックの喪失時、あるいは clkswitch 信号がアサートされた時に開始されます。
Perform the input clock switchover after (number) input clock cycles	On または Off	PLL がクロック・スイッチオーバーを実行する前に待機するクロック・サイクル数を指定するには、このオプションをオンにします。待機するクロック・サイクルの許容数は、デバイスによって異なります。
Create an 'activeclock' output to indicate the input clock being used	On または Off	どの入力クロック信号が PLL を駆動しているかをモニタするには、このオプションをオンにします。現在のクロック信号が inclk0 であるとき、activeclock 信号は Low です。現在のクロック信号が inclk1 であるとき、activeclock 信号は High です。
Create a 'clkbad' output for each input clock	On または Off	入力クロック信号がトグルを停止したときをモニタするには、このオプションをオンにします。clkbad0 信号は inclk0 信号をモニタします。clkbad1 信号は inclk1 信号をモニタします。inclk0 信号がトグルを停止すると、clkbad0 信号は High になります。また、inclk1 信号がトグルを停止すると、clkbad1 信号は High になります。入力クロック信号がトグルしているとき、clkbad 信号は Low のままです。

関連情報

- 26 ページの [クロック・スイッチオーバー](#)
- 36 ページの [ガイドライン:クロック・スイッチオーバー](#)

6.1.5 PLL ダイナミック・リコンフィギュレーションのパラメータ設定

通常のダイナミック・リコンフィギュレーション手法のパラメータ設定は、ALTPLL IP コア・パラメータ・エディタの **PLL Reconfiguration** ページにあります。

表 19. PLL ダイナミック・リコンフィギュレーションのパラメータ・エディタ設定

パラメータ	Value	説明
Create optional inputs for dynamic reconfiguration	On または Off	scanclock, scanclockena, scandata, scandone, scandataout、および configupdate のインスタンス化のためのすべての PLL リコンフィギュレーション・ポートを有効にするには、このオプションをオンにします。
Initial Configuration File	—	ALTPLL_RECONFIG IP コアを初期化するために使用されるコンフィギュレーション・ファイルの位置を指定します。
Additional Configuration File(s)	—	追加のコンフィギュレーション・ファイルを指定します。このファイルには PLL の追加の設定が含まれる場合があります。また、ALTPLL_RECONFIG IP コアを初期化するために使用されることがあります。

関連情報

- 31 ページの [PLL リコンフィギュレーション](#)
- 45 ページの [ダイナミック位相コンフィギュレーションの実装](#)

6.1.6 ダイナミック位相コンフィギュレーションのパラメータ設定

ダイナミック位相コンフィギュレーション機能を有効にするパラメータ設定は、ALTPLL IP コア・パラメータ・エディタの **PLL Reconfiguration** ページにあります。



表 20. ダイナミック位相コンフィギュレーションのパラメータ・エディタ設定

パラメータ	Value	説明
Create optional inputs for dynamic phase reconfiguration	On または Off	ダイナミック位相コンフィギュレーション機能を有効にするには、このオプションをオンにします。以下のポートが作成されます。 <ul style="list-style-type: none"> • phasecounterselect[2..0] • phaseupdown • phasestep • scanclk • phasedone
Enable phase shift step resolution edit	On または Off	Output Clocks ページで個々の PLL 出力クロックの Phase shift step resolution(ps) の値を変更するには、このオプションをオンにします。 デフォルトでは、最高の位相シフト分解能の値は VCO 周期の 1/8 です。VCO 周波数がサポートされる VCO 範囲の下限である場合、位相シフト分解能はデザインに適したものよりも大きくなる場合があります。位相シフト・ステップ分解能を調整するには、このオプションを使用します。

関連情報

- 24 ページの [プログラマブル位相シフト](#)
- 45 ページの [ダイナミック位相コンフィギュレーションの実装](#)

6.1.7 出力クロックのパラメータ設定

ALTPLL パラメータ・エディタの **Output Clocks** ページには、クロック出力信号のパラメータ設定が含まれています。ALTPLL IP コアの c0、c1、c2、c3、および c4 クロック出力信号をコンフィギュレーションすることができます。

各オプションは、次の 2 つのカラムを有します。

- Requested settings—実装する設定。
- Actual settings—要求された設定に最も近似するように、PLL 回路で実装可能な値に一番近い値を設定。

Actual settings カラムの値は、Requested settings を調整するためのガイドとして使用します。いずれかの出力クロックの Requested settings が近似できない場合、ALTPLL IP コア・パラメータ・エディタは各ページのトップで警告メッセージを発生します。

表 21. 出力クロックのパラメータ・エディタ設定

パラメータ	Value	説明
Use this clock	On または Off	ALTPLL インスタンスで出力クロック・ポートを生成するには、このオプションをオンにします。 補償される出力クロック・ポートは、デフォルトで有効になっています。これとは異なる補償される出力クロック・ポートを選択しない限り、無効にすることはできません。
Enter output clock frequency	—	出力クロック信号の周波数を指定します。
Enter output clock parameters	—	周波数の代わりに出力クロックのパラメータを指定します。
Clock multiplication factor	—	信号のクロック乗数係数を指定します。
Clock division factor	—	信号のクロック分周係数を指定します。
<i>continued...</i>		

パラメータ	Value	説明
Clock phase shift	—	出力クロック信号のプログラマブル位相シフトを設定します。 最小の位相シフトは VCO 周期の 1/8 です。位相の増分については、最大のステップ・サイズは 45 度です。 Clock multiplication factor オプションと Clock division factor オプションを使用して、より小さいステップを設定することができます。 たとえば、ポストスケール・カウンタが 32 である場合、最小の位相シフト・ステップは 0.1°です。Up および Down ボタンを使用して、位相シフトの値を循環することができます。あるいは、これらのボタンを使用する代わりに、位相シフト・フィールドに手動で数値を入力することができます。
Clock duty cycle (%)	—	出力クロック信号のデューティ・サイクルを設定します。
Per Clock Feasibility Indicators	—	実現不可能な設定を有する出力クロックを示します。 出力クロック名が赤で示されるものが実現不可能な設定を有するクロックの名前です。緑色でリストされるクロックには設定上の問題はありませぬ。また、グレーで表示されるクロック名は選択されていない出力クロックを示します。警告メッセージの内容を解決するには、影響のある出力クロックの Requested settings を調整する必要があります。

ALTPLL IP コア・パラメータ・エディタは最も単純な有理数を計算し、Actual settings カラムで表示します。コピー・ボタンを使用して、Actual settings から Requested settings に値をコピーすることができます。

図 -30: PLL の出力クロック周波数

$$\text{Output clock frequency} = \text{Input clock frequency} \times \frac{\text{Multiplication factor}}{\text{Division factor}}$$

たとえば、入力クロック周波数が 100 MHz であり、必要な通倍係数と分周係数がそれぞれ 205 と 1025 である場合、出力クロック周波数は、 $100 \times 205/1025 = 20$ MHz と計算されます。Actual settings は最も単純な有理数を反映します。そのため、実際の通倍係数と分周係数はそれぞれ 1 と 5 になります。

6.2 ALTPLL ポートおよび信号

表 22. MAX 10 デバイスの ALTPLL 入力ポート

ポート名 ⁽¹⁰⁾	条件	説明
areset	オプション	GATE_LOCK_COUNTER パラメータを含むすべてのカウンタを初期値にリセットします。
clkswitch	オプション	クロック入力ポート (inclk0 ポートと inclk1 ポート) 間を動的にトグルする、あるいは自動クロック・スイッチオーバーを手動でオーバーライドするコントロール入力ポートです。 inclk1 ポートのみが作成されている場合、clkswitch ポートを作成する必要があります。
configupdate	オプション	ダイナミック・フル PLL リコンフィギュレーション。
inclk[]	必須	クロック・ネットワークを駆動するクロック入力です。 複数の inclk[] ポートが作成されている場合、clkselect ポートを使用して使用するクロックを指定する必要があります。inclk0 ポートは常に接続している必要があります。なお、切り替えが必要な場合は他のクロック入力を接続します。

continued...

(10) 正確な名前を得るには、ポート名内の角括弧 [] を整数に置き換えてください。例: inclk0、inclk1



ポート名 ⁽¹⁰⁾	条件	説明
		このポートは、専用クロック・ピンまたは PLL 出力クロックで駆動することができます。
pfdena	オプション	位相周波数検出器(PFD)を有効にします。PFDが無効のとき、PLLは入力クロックに関係なく動作を継続します。PLL出力クロック周波数は一定時間変化しないため、信頼性の高い入力クロックが存在しないときは、pfdenaポートをシャットダウンまたはクリーンアップ機能として使用することができます。
phasecounterselect[]	オプション	カウンタ選択を指定します。phasecounterselect[2..0]ビットを使用して、位相調整のためにMカウンタまたはCカウンタのうちの1つのいずれかを選択することができます。1つのアドレス・マップですべてのCカウンタを選択します。この信号は、SCANCLKの立ち上がりエッジでPLLのレジスタに格納されます。
phasestep	オプション	ダイナミック位相シフトを指定します。ロジック Highでダイナミック位相シフトを有効にします。
phaseupdown	オプション	ダイナミック位相シフトの方向を指定します。1 = UP、0 = DOWNです。信号はSCANCLKの立ち上がりエッジのPLLでレジスタに格納されています。
scanclk	オプション	シリアル・スキャン・チェーンの入力クロック・ポート。PHASESTEPと組み合わせて使用され、ダイナミック位相シフトを有効または無効にする、コアからの自走クロックです。ダイナミック・リコンフィギュレーションではSCANCLKと共有されます。
scanclkena	オプション	シリアル・スキャン・チェーンのクロック・イネーブル・ポート。
scandata	オプション	シリアル・スキャン・チェーンのデータを格納します。

表 23. MAX 10 デバイスの ALTPLL 出力ポート

ポート名 ⁽¹¹⁾	条件	説明
activeclock	オプション	クロック・スイッチオーバー回路が開始する際、どのクロックがプライマリ基準クロックであるかを指定します。 inclk0が使用されている場合、activeclockポートはLowになります。また、inclk1が使用されている場合、activeclockポートはHighになります。 プライマリ基準クロックが正しくトグルしていないとき、クロック・スイッチオーバーを自動的に開始するようPLLを設定するか、あるいはclkswitch入力ポートを使用して、クロック・スイッチオーバーを手動で開始することができます。
c[]	必須	PLLのクロック出力。
clkbad[]	オプション	clkbad1ポートとclkbad0ポートは、入力クロックのトグルをチェックします。 inclk0ポートがトグルを停止すると、clkbad0ポートはHighになります。また、inclk1ポートがトグルを停止すると、clkbad1ポートはHighになります。
locked	オプション	この出力ポートは、PLLがPhase-Lockedに達した際にインジケータとして機能します。lockedポートは、PLLがロックされている限りHighのままです。また、PLLのロックが解除されるとLowになります。

continued...

⁽¹⁰⁾ 正確な名前を得るには、ポート名内の角括弧[]を整数に置き換えてください。例: inclk0、inclk1

⁽¹¹⁾ 正確な名前を得るには、ポート名内の角括弧[]を整数に置き換えてください。例: c0、c1

ポート名 ⁽¹¹⁾	条件	説明
		<p>locked 信号をゲートするにあたって必要なサイクル数は、PLL 入力クロックによって異なります。ゲート・ロック回路は PLL 入力クロックによってクロックされます。PLL の最大ロック時間は MAX 10 Device Datasheet で確認することができます。</p> <p>PLL の最大ロック時間を確保し、PLL 入力クロックの周期で分周します。結果は、locked 信号をゲートするために必要なクロック・サイクル数になります。</p> <p>ロック信号は PLL の非同期出力です。PLL ロック信号は、位相周波数検出器 (PFD) を以下のように供給する基準クロックとフィードバック・クロックから求められます。</p> <ul style="list-style-type: none"> • 基準クロック = 入力クロック/N • フィードバック・クロック = VCO/M <p>基準クロックとフィードバック・クロックの位相および周波数が同じであるかロック回路許容範囲内であるとき、PLL は locked ポートをアサートします。2 つのクロック信号間の差がロック回路許容範囲外になると、PLL はロックを喪失します。</p>
phasedone	オプション	<p>この出力ポートは、ダイナミック位相リコンフィギュレーションが完了したことを示します。</p> <p>phasedone 信号がアサートされる場合、位相調整が完了し、PLL が 2 番目の調整パルスで動作可能な状態であることをコア・ロジックに示します。この信号は、内部 PLL タイミングに基づいてアサートし、SCANCLK の立ち上がりエッジでデアサートします。</p>
scandataout	オプション	<p>シリアル・スキャン・チェーンのデータ出力です。</p> <p>scandataout ポートは、PLL のリコンフィギュレーションが完了したことを判断するために使用することができます。リコンフィギュレーションが完了すると最後の出力がクリアされます。</p>
scandone	オプション	<p>この出力ポートは、スキャン・チェーンの書き込み動作が開始されることを示します。</p> <p>scandone ポートは、スキャン・チェーンの書き込み動作が開始すると High になります。また、スキャン・チェーンの書き込み動作が完了すると Low になります。</p>

関連情報

17 ページの PLL コントロール信号

⁽¹¹⁾ 正確な名前を得るには、ポート名内の角括弧 [] を整数に置き換えてください。例: c0、c1



7 ALTPLL_RECONFIG IP コア・リファレンス

7.1 ALTPLL_RECONFIG パラメータ

表 24. MAX 10 デバイスの ALTPLL_RECONFIG IP コアのパラメータ

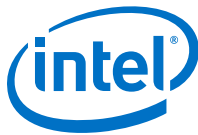
この表は、MAX 10 デバイスに適用可能な IP コアのパラメータをリストしています。

ページ	パラメータ	値	説明
Parameter Settings	Currently Selected Device Family	—	選択するデバイス・ファミリを指定します。
	Which scan chain type will you be using?	—	スキャン・チェーンは、設定を格納するために使用されるシリアル・ソフト・レジスタ・チェーンです。スキャン・チェーンはキャッシュのように動作します。reconfig 信号をアサートすると、PLL はキャッシュ内の値でリコンフィギュレーションされます。スキャン・チェーンのタイプは、リコンフィギュレーションされる PLL のタイプに従う必要があります。スキャン・チェーンのタイプは、 Top/Bottom のデフォルト値を有します。
	Do you want to specify the initial value of the scan chain?	No, leave it blank, Yes, use this file for the content data	スキャン・チェーンの初期値を指定します。ファイルを指定しない場合は No, leave it blank を選択し、.hex または .mif ファイルをブラウズするには Yes, use this file for the content data を選択します。 ROM から初期化するオプションは利用できませんが、 Add ports to write to the scan chain from external ROM during run time をオンにすることで、実行時に外部 ROM からスキャン・チェーンへ書き込むよう追加のポートを選択することが可能です。
	Add ports to write to the scan chain from external ROM during run time	On / Off	ユーザー・モード時に外部 ROM に格納される複数のコンフィギュレーション・ファイルのサイクルを利用するには、このオプションをオンにします。
EDA	Simulation Libraries	—	機能シミュレーションのライブラリを指定します。
	Generate netlist	On / Off	合成エリアとタイミング見積もりネットリストを生成するには、このオプションをオンにします。
Summary	—	—	生成するファイルのタイプを指定します。グレーのチェックマークは自動的に生成されるファイルを示します。チェックのないチェック・ボックスはオプションのファイルを示します。以下のタイプのファイルから選択します。 <ul style="list-style-type: none"> AHDL インクルード・ファイル(<function name> . inc) VHDL コンポーネント宣言ファイル((<function name> . cmp) Quartus Prime シンボル・ファイル(<function name> . bsf) インスタンス化テンプレート・ファイル(<function name> _inst . v または <function name> _inst . vhd) Verilog HDL ブラック・ボックス・ファイル(<function name> _bb . v) Generate netlist オプションがオンになっている場合、ネットリストのファイル(<function name> _syn . v)も使用可能です。

Intel Corporation. 無断での引用、転載を禁じます。Intel、インテル、Intel ロゴ、Altera、ARRIA、CYCLONE、ENPIRION、MAX、NIOS、QUARTUS および STRATIX の名称およびロゴは、アメリカ合衆国および/またはその他の国における Intel Corporation の商標です。インテルは FPGA 製品および半導体製品の性能がインテルの標準保証に準拠することを保証しますが、インテル製品およびサービスは、予告なく変更される場合があります。インテルが書面にて明示的に同意する場合を除き、インテルはここに記載されたアプリケーション、または、いかなる情報、製品、またはサービスの使用によって生じるいっさいの責任を負いません。インテル製品の顧客は、製品またはサービスを購入する前、および、公開済みの情報を信頼する前には、デバイスの仕様を最新のバージョンにしておくことをお勧めします。

*その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ISO
9001:2008
登録済



関連情報

- 24 ページの プログラマブル位相シフト
- 45 ページの ダイナミック位相コンフィギュレーションの実装
- 31 ページの PLL リコンフィギュレーション
- 45 ページの ダイナミック位相コンフィギュレーションの実装

7.2 ALTPLL_RECONFIG ポートおよび信号

表 25. MAX 10 デバイスの ALTPLL_RECONFIG 入力ポート

ポート名	条件	説明
clock	必須	個々のパラメータをロードするためのクロック入力です。また、この信号はリコンフィギュレーション時に PLL をクロックします。 このクロック入力ポートは有効なクロックに接続する必要があります。 clock f_{MAX} については、MAX 10 Device Datasheet を参照してください。
reset	必須	IP コアへの非同期リセット入力です。 Intel は、最初に使用する前にこの IP コアをリセットして、有効な状態にあることを保証することを推奨します。ただし、リセット状態では電源が入ります。このポートは接続されている必要があります。
data_in[]	オプション	パラメータを書き込む際に、パラメータ値を提供するデータ入力です。 この 9 ビットの入力ポートは、書き込み動作中にスキャン・キャッシュに書き込まれるデータを提供します。書き込まれるカウンタ・パラメータのビット幅によってキャッシュに読み取られる data_in[] のビット数が決まります。 たとえば、C0 カウンタの Low ビット・カウントが 8 ビット幅であると、data_in[7..0] は正しいキャッシュ位置に読み取られます。C0 カウンタのバイパス・モードが 1 ビット幅であると、data_in[0] はこのパラメータの値として読み取られません。 これを省略する場合、デフォルト値は 0 です。
counter_type[]	オプション	カウンタ・タイプを指定します。 対応する動作 (読み取り、書き込み、またはリコンフィギュレーション) 向けに選択すべきカウンタ・タイプを選択する 4 ビット・バスの入力ポートです。 counter_type の値と設定される物理カウンタ間をマッピングするには、counter_type[3..0] のセッティング・テーブルを参照してください。
counter_param[]	オプション	counter_type ポートで指定した値のパラメータを指定します。 所与のカウンタ・タイプのどのパラメータを更新する必要があるかを選択する 3 ビット・バスの入力ポートです。各パラメータ・タイプへのマッピングと対応するパラメータのビット幅は counter_param[3..0] セッティング・テーブルで定義されています。
read_param	オプション	キャッシュから counter_type および counter_param ポートで指定したパラメータを読み取り、data_out[] ポートに供給します。 read_param 信号がアサートされる場合、スキャン・キャッシュを読み取り、data_out[] に供給する必要があることを示します。スキャン・キャッシュのビット位置と、読み取られた後に data_out[] に供給されるビット数は、counter_type および counter_param の値によって決まります。read_param 信号は立ち上がりクロック・エッジでサンプリングされます。read_param 信号がアサートされる場合、パラメータ値はキャッシュから読み取られます。パラメータが再度読み取られることを防ぐため、read_param 信号を 1 クロック・サイクルの間だけアサートします。 read_param 信号のアサートに続いて、立ち上がりクロック・エッジで busy 信号がアサートされます。パラメータが読み取られている間、busy 信号はアサートされたままです。busy 信号がデアサートされると、data_out[] の値は有効になり、次のパラメータをロードすることができます。busy 信号がアサートされている間、data_out[] の値は無効です。

continued...



ポート名	条件	説明
		read_param 信号がアサートされる時、busy 信号はクロックの次の立ち上がりエッジでのみアサートされ、read_param 信号と同じクロック・サイクルではアサートされません。
write_param	オプション	<p>counter_type および counter_param ポートで指定したパラメータを、data_in[] ポートで指定した値と併せてキャッシュに書き込みます。</p> <p>write_param 信号がアサートされる場合、data_in[] の値は counter_type[] と counter_param[] で指定したパラメータに書き込まれる必要があることを示します。data_in[] ポートから読み取られるビット数はパラメータによって決まります。write_param 信号は立ち上がりクロック・エッジでサンプリングされます。write_param 信号がアサートされる場合、パラメータ値はキャッシュに書き込まれます。パラメータが再度書き込まれることを防ぐため、write_param 信号を 1 クロック・サイクルの間だけアサートします。</p> <p>write_param 信号のアサートに続いて、立ち上がりクロック・エッジで busy 信号がアサートされます。パラメータが書き込まれている間、busy 信号はアサートされたままで、data_in[] への入力は無視されます。busy 信号がデアサートされると、次のパラメータを書き込むことができます。</p> <p>write_param 信号がアサートされる時、busy 信号はクロックの次の立ち上がりエッジでのみアサートされます。busy 信号は write_param 信号と同じクロック・サイクルではアサートされません。</p>
reconfig	必須	<p>PLL が現在のキャッシュで指定した PLL 設定でリコンフィギュレーションされるよう指定します。</p> <p>reconfig 信号がアサートされる場合、キャッシュ内の値で PLL をリコンフィギュレーションする必要があることを示します。reconfig 信号は立ち上がりクロック・エッジでサンプリングされます。reconfig 信号がアサートされる場合、キャッシュされた設定は PLL にロードされます。PLL コンフィギュレーションの再ロード防くため、reconfig 信号を 1 クロック・サイクルの間だけアサートします。busy 信号は、reconfig 信号のアサートに続いて立ち上がりクロック・エッジでアサートされます。PLL がロードされている間、busy 信号はアサートされたままです。busy 信号がデアサートされると、再びパラメータ値を変更することができます。</p> <p>リコンフィギュレーション中およびリコンフィギュレーション後、スキャン・チェーンのデータ・キャッシュは変更されません。これにより、1 つのパラメータのみを使用して、リコンフィギュレーション設定の新しいセットを容易に作成することができます。</p> <p>reconfig の前のアサート以来 write_param がアサートされていない場合、スキャン・チェーン全体が再び PLL にシフトされます。</p> <p>reconfig 信号がアサートされる時、busy 信号はクロックの次の立ち上がりエッジでのみアサートされます。busy 信号は reconfig 信号と同じクロック・サイクルではアサートされません。</p>
pll_areset_in	オプション	<p>PLL をリセットする必要があることを示す入力信号です。</p> <p>pll_areset_in 信号がアサートされる場合、PLL の IP コアをリセットする必要があることを示します。未接続のままにする場合、このポートのデフォルトは 0 です。デザインに ALTPLL_RECONFIG IP コアを使用している場合、他の方法で PLL をリセットすることはできません。PLL を手動でリセットするには、この IP コア・ポートを使用する必要があります。</p>
pll_scandone	オプション	ALTPLL_RECONFIG IP コアの出力ポートです。このポートは PLL の scandone 出力信号によって駆動され、PLL がリコンフィギュレーションされることを判断します。
pll_scandataout	必須	ALTPLL IP コアからの scandataout 信号によって駆動される入力ポートです。ALTPLL IP コアの現在のコンフィギュレーションを読み取るにはこのポートを使用します。この入力ポートは、ダイナミックにリコンフィギュレーション可能なビットからの ALTPLL スキャン・データ出力を保持します。pll_scandataout ポートは PLL の scandataout ポートに接続する必要があります。reconfig 信号がアサートされる時、このポート上の動作を観察することができます。

表 26. MAX 10 デバイスの ALTPLL_RECONFIG 出力ポート

ポート名	条件	説明
data_out[]	オプション	read_param がアサートされる時、キャッシュから読み取られるデータです。この 9 ビット出力バスによって、ユーザーにパラメータ・データが提供されます。read_param 信号がアサートされる時、counter_type[] および counter_param[] の値によって、キャッシュからロードされ data_out[] バスで駆動されるパラメータ値が決まります。IP コアが busy 信号をデアサートするとき、バスの該当するビット(たとえば、[0] または [3..0]) は有効な値を保持します。
busy	オプション	PLL がパラメータをキャッシュから読み取っている、またはキャッシュに書き込んでいる、あるいは PLL をコンフィギュレーションしていることを示します。 busy 信号がアサートされている間、パラメータの読み取りと書き込み、およびリコンフィギュレーションの開始を実行することはできません。IP コアへの遷移は busy 信号がアサートされていないときのみ行うことができます。read_param、write_param、または reconfig 入力ポートがアサートされるとこの信号は High になり、指定された動作が完了するまで High のままです。リコンフィギュレーション動作の場合、busy 信号は pll_areset 信号がアサートされ、その後デアサートされるまで High のままです。
pll_areset	必須	リコンフィギュレーションする PLL の areset ポートを駆動します。 リコンフィギュレーションが正しく機能するよう、pll_areset ポートは ALTPLL IP コアの areset ポートに接続する必要があります。この信号はアクティブ High です。 pll_areset は、pll_areset_in がアサートされる時、あるいはリコンフィギュレーションの後、scandone 信号が High になった後の次の立ち上がりエッジでアサートされます。ALTPLL_RECONFIG IP コアを使用する場合、pll_areset 出力ポートを使用して PLL の areset ポートを駆動します。
pll_configupdate	オプション	リコンフィギュレーションする PLL の configupdate ポートを駆動します。アサートされると、pll_configupdate ポートは選択したデータを PLL コンフィギュレーション・ラッチにロードします。この信号は最後のデータ・ビットが送信された後にアサートされます。
pll_scancclk	必須	リコンフィギュレーションする PLL の scancclk ポートを駆動します。各デバイスの最大 scancclk 周波数については、それぞれのデバイスのデバイス・ハンドブックを参照してください。
pll_scancclkena	オプション	このポートは、リコンフィギュレーションする PLL の scancclk ポートのクロック・イネーブルとして機能します。 リコンフィギュレーションは、pll_scancclkena がアサートされた後、pll_scancclk の最初の立ち上がりエッジで開始します。pll_scancclkena 信号がデアサートされると、pll_scancclk の最初の立ち下がりエッジでは、IP コアが PLL へのデータのスキャンを停止します。
pll_scandata	必須	リコンフィギュレーションする PLL の scandata ポートを駆動します。 IP コアからの出力ポートは、ダイナミックにリコンフィギュレーション可能なビットのための PLL へのスキャン・データ入力を保持します。pll_scandata ポートは scandata を PLL に送信します。このポート上におけるすべての動作は、reconfig 信号がアサートされる時に観察することができます。

7.3 ALTPLL_RECONFIG カウンタ設定

表 27. MAX 10 デバイスの counter_type[3..0] 設定

カウンタ選択	バイナリ	デシマル
N	0000	0
M	0001	1
CP/LF	0010	2
VCO	0011	3
<i>continued...</i>		



カウンタ選択	バイナリ	デシマル
C0	0100	4
C1	0101	5
C2	0110	6
C3	0111	7
C4	1000	8
無効な値	1001	9
無効な値	1010	10
無効な値	1011	11
無効な値	1100	12
無効な値	1101	13
無効な値	1110	14
無効な値	1111	15

表 28. MAX 10 デバイスの counter_param[2..0] 設定

カウンタ・タイプ	カウンタ Param	バイナリ	デシマル	幅(ビット)
通常カウンタ (C0 ~ C4)	High count	000	0	8
	Low count	001	1	8
	Bypass	100	4	1
	Mode (奇数/偶数の分周)	101	5	1
CP/LF	Charge pump unused	101	5	5
	Charge pump current	000	0	3
	Loop filter unused	100	4	1
	Loop filter resistor	001	1	5
	Loop filter capacitance	010	2	2
VCO	VCO post scale	000	0	1
M/N カウンタ	High count	000	0	8
	Low count	001	1	8
	Bypass	100	4	1
	Mode (奇数/偶数の分周)	101	5	1
	Nominal count	111	7	9

Nominal count が偶数の場合、カウンタ・ビットは以下のように自動的に設定されます。

- `high_count = Nominalcount/2`
- `low_count = Nominalcount/2`



Nominal count が奇数の場合、カウンタ・ビットは以下のように自動的に設定されます。

- $high_count = (Nominalcount + 1)/2$
- $low_count = Nominalcount - high_count$
- 奇数/偶数の分周ビット = 1

Nominal count = 1 の場合、bypass bit = 1 となります。



8 内蔵オシレーター IP コア・リファレンス

8.1 altclkctrl パラメータ

表 29. デバイスの altclkctrl IP コアのパラメータ

この表は、MAX 10 デバイスに適用可能な IP コアのパラメータをリストしています。

パラメータ	Value	説明
クロック周波数	55, 116	シミュレーションでのクロック周波数を指定します。指定しない場合、デフォルト値は 55 MHz です。

8.2 内蔵オシレーターポートおよび信号

表 30. MAX 10 デバイスの内蔵オシレーター入力ポート

ポート名	条件	説明
oscena	必須	内蔵オシレータをオンまたはオフにする入力コントロール信号です。

表 31. MAX 10 デバイスの内蔵オシレーター出力ポート

ポート名	条件	説明
clkout	オプション	内蔵オシレータからの出力クロックです。



A MAX 10 のクロッキングおよび PLL ユーザーガイドのアーカイブ

使用する IP コアバージョンがリストにない場合、以前の IP コアバージョンのユーザーガイドが適用されます。

IP コアバージョン	ユーザーガイド
15.1	MAX 10 Clocking and PLL User Guide
15.0	MAX 10 Clocking and PLL User Guide
14.1	MAX 10 Clocking and PLL User Guide

Intel Corporation. 無断での引用、転載を禁じます。Intel、インテル、Intel ロゴ、Altera、ARRIA、CYCLONE、ENPIRION、MAX、NIOS、QUARTUS および STRATIX の名称およびロゴは、アメリカ合衆国および/ またはその他の国における Intel Corporation の商標です。インテルは FPGA 製品および半導体製品の性能がインテルの標準保証に準拠することを保証しますが、インテル製品およびサービスは、予告なく変更される場合があります。インテルが書面にて明示的に同意する場合を除き、インテルはここに記載されたアプリケーション、または、いかなる情報、製品、またはサービスの使用によって生じるいっさいの責任を負いません。インテル製品の顧客は、製品またはサービスを購入する前、および、公開済みの情報を信頼する前には、デバイスの仕様を最新のバージョンにしておくことをお勧めします。

*その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ISO
9001:2008
登録済



B MAX 10 のクロッキングおよび PLL ユーザーガイド改訂履歴

日付	バージョン	変更内容
2017年2月	2017.02.21	商標を「Intel」へ変更。
2015年11月	2015.11.02	<ul style="list-style-type: none"> IP カタログと Parameter Editor、IP コアの生成、IP コアによって生成されるファイルの項を削除し、Introduction to Altera IP Cores へのリンクを追加。 表記を Quartus II から Quartus Prime へ変更。
2015年6月	2015.06.12	PLL デザインの検討事項に接続制約のガイドラインを追加。
2015年5月	2015.05.04	分解能の高い位相シフト式を再配置。
2014年12月	2014.12.15	<ul style="list-style-type: none"> クロック入力に専用クロック入力ピンを使用しない場合は、専用クロック入力ピンを汎用入力ピンとして使用することもできます、と表現を訂正。 内蔵オシレータのアーキテクチャと機能のセクションに、内蔵リング・オシレータが最大 232 MHz で動作可能であるが、この周波数はアクセスできないこと説明する記述を追加。 内部オシレータ用の接続制限のガイドラインを追加。 内部オシレータ IP コア・パラメータにクロック周波数を追加。 内部オシレータ周波数の表を内部オシレータのアーキテクチャと機能の章から MAX 10 FPGA デバイス・データシートへ移動。
2014年9月	2014.09.22	初版

Intel Corporation. 無断での引用、転載を禁じます。Intel、インテル、Intel ロゴ、Altera、ARRIA、CYCLONE、ENPIRION、MAX、NIOS、QUARTUS および STRATIX の名称およびロゴは、アメリカ合衆国および/またはその他の国における Intel Corporation の商標です。インテルは FPGA 製品および半導体製品の性能がインテルの標準保証に準拠することを保証しますが、インテル製品およびサービスは、予告なく変更される場合があります。インテルが書面にて明示的に同意する場合を除き、インテルはここに記載されたアプリケーション、または、いかなる情報、製品、またはサービスの使用によって生じるいっさいの責任を負いません。インテル製品の顧客は、製品またはサービスを購入する前、および、公開済みの情報を信頼する前には、デバイスの仕様を最新のバージョンにしておくことをお勧めします。

*その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ISO
9001:2008
登録済