

はじめに

フェーズロック・ループ (PLL) は、いくつかの分周カウンタおよび異なる電圧制御発振器 (VCO) の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。Stratix[®] III の PLL では、カウンタ設定をリコンフィギュレーションして、PLL 出力クロックを動的に位相シフトすることができます。またチャージ・ポンプおよびループ・フィルタ・コンポーネントを変更して、PLL 帯域幅を動的に変化させることもできます。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、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。

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有用です。PLL リコンフィギュレーションは、PLL 出力周波数をスイープし、クロックの出力位相をデザインのどの段階でも調整できるため、プロトタイプ環境でも有用です。例えば、あるテスト・パターン生成システムは、被試験デバイスの要件に応じて、50 MHz または 100 MHz でパターンを生成および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できるため、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、この機能を使用して出力クロックの位相シフトを変更して、clock-to-out (t_{co}) 遅延をリアルタイムで調整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。



このアプリケーション・ノートでは、Stratix III デバイスにおける PLL リコンフィギュレーションの実装フローについて説明します。このアプリケーション・ノートを、以下の Stratix III に関する資料と併せて使用してください。

- 「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスのクロック・ネットワークおよび PLL」の章
- ALTPLL メガファンクション ユーザーガイド
- Phase-Locked Loops Reconfiguration (ALTPLL_RECONFIG) MegaFunction User Guide

このアプリケーション・ノートでは、ハードウェアおよびソフトウェアを統合して、PLL リコンフィギュレーション機能の使用方法について説明します。このアプリケーション・ノートでは、以下について説明します。

- Stratix III デバイスにおける PLL リコンフィギュレーションの概要
- 周波数プリスケラ・アプリケーションにおけるリアルタイム PLL リコンフィギュレーション機能の実装の完全フロー
- ダイナミック位相シフト機能の実装の完全フロー
- リコンフィギュレーションのための PLL パラメータを選択するときに考慮しなければならないデザイン検討事項

このドキュメントで説明されたデザイン例は、アルテラ・ウェブサイトの [資料：アプリケーション・ノートのページからダウンロード](#) できます。デザイン例のダウンロードおよび使用について手順は 18 ページの「[デザイン例](#)」を参照してください。

Stratix III デバイスにおける PLL リコンフィギュレーションの概要

次の PLL コンポーネントはリアルタイムでリコンフィギュレーション可能です。

- 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・カウンタ (C0-C9)
- ポスト VCO 分周器 (K)
- チャージ・ポンプ電流 (ICP) とループ・フィルタ・コンポーネント (R, C)

図 1 に、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って、PLL カウンタ設定をダイナミックに調整する方法を示します。シリアル・データが scandataport を通じてスキャン・チェーンに入力され、シフト・レジスタが scancclk でクロックされます。シリアル・データは、scanclkena 信号がアサートされている間には、スキャン・チェーンを通じてシフトされます。データの最終ビットがクロックされた後、リコンフィギュレーション・ステート・マシン信号の configupdate 信号を最低 1 scancclk サイクルアサートすると、PLL コンフィギュレーション・ビットがスキャン・レジスタのデータで同期的に更新されます。また、16 進ファイル (.hex) フォーマットまたはメモリ初期化ファイル (.mif) フォーマットの .mif ファイルを用いて、スキャン・チェーンを初期化または変更することもできます。.mif の設定について詳しくは、9 ページの「PLL リコンフィギュレーションのスキャン・レジスタ・ビットマップ」を参照してください。

図 1. PLL リコンフィギュレーション・スキャン・チェーン (1)、(3)

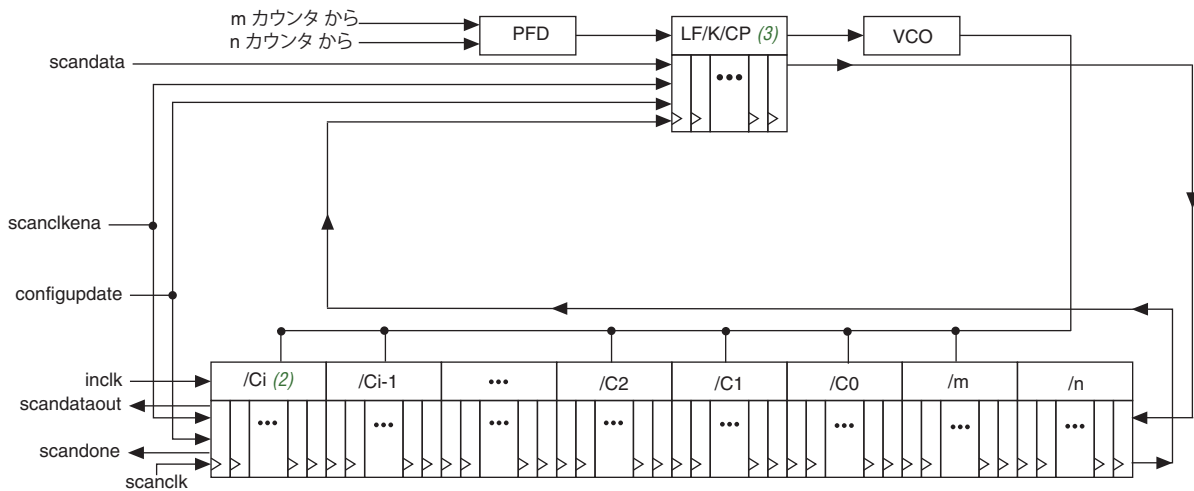


図 1 の注:

- (1) StratixIII のレフト/ライト PLL は、C0 ~ C6 カウンタをサポートします。
- (2) $i = 6$ または $i = 9$ 。
- (3) この図では、K カウンタの対応スキャン・レジスタは、チャージ・ポンプとループ・フィルタのスキャン・レジスタ間にあります。K カウンタは物理的に VCO の後に配置されます。

- ハードウェアおよびソフトウェアの実装について詳しくは、「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスのクロック・ネットワークおよび PLL」の章および「ALTPLL メガファンクション ユーザーガイド」をそれぞれ参照してください。

Quartus II ソフトウェアによる PLL リコンフィギュレーションの実装

ALTPLL MegaWizard® Plug-in Manager を使用して、デザイン内の ALTPLL メガファンクション・インスタンスにおけるリコンフィギュレーション回路をイネーブルすることができます。ALTPLL_RECONFIG メガファンクションは、Stratix III PLL のリコンフィギュレーション・プロセスを簡素化します。

- 詳細については、「Phase-Locked Loops Reconfiguration (ALTPLL_RECONFIG) MegaFunction User Guide」および「ALTPLL メガファンクション ユーザーガイド」をそれぞれ参照してください。

次の 2 つの方法では、ALTPLL_RECONFIG メガファンクションを用いて PLL リコンフィギュレーションを実現する手順について説明します。これらの方法のいずれかもリコンフィギュレーションに適用します。

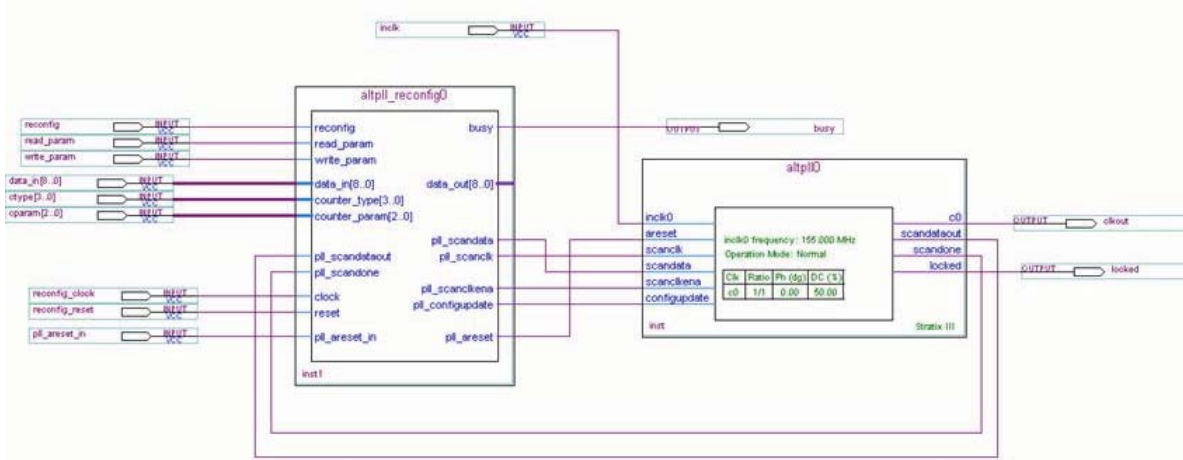
- 方法1は、他の PLL を影響せずに特定の PLL カウンタをリコンフィギュレーションするのに使用されます。この方法は 1 回に 1 つまたは一部の PLL 設定を変更する必要がある時に使用されます。
- 方法2は、.mif ファイルを用いて全ての PLL カウンタ設定をリコンフィギュレーションするのに使用されます。スキャン・チェーンは .mif または .hex フォーマットの .mif ファイルによってアップデートされ、そして PLL カウンタはスキャン・チェーンの内容によってアップデートされます。

PLL リコンフィギュレーション：方法 1

特定の PLL カウンタをリコンフィギュレーションするには、ALTPLL_RECONFIG メガファンクションに特定の counter_type、counter_param および data_in をシフト・インします。図 2 に、ALTPLL メガファンクションと ALTPLL_RECONFIG メガファンクション間の接続を示します。

- 👉 Quartus® II ソフトウェア v 5.0 以降から、Quartus II ソフトウェア・ロジック・オプションでの PRESERVE_PLL_COUNTER_ORDER 設定は、PLL において PLL リコンフィギュレーションが使用される時、自動的にオンになります。ソフトウェア・ロジック・オプションがオンになっているため、Quartus II ソフトウェアはカウンタを循環してクロック配線を向上しないので、クロック配線エラーが発生することになります。これを改正するには、まず PLL リコンフィギュレーションをオフにして、Quartus II ソフトウェアを最適なカウンタ選択をさせ、そして PLL リコンフィギュレーションをオンに戻し、Quartus II ソフトウェアが選択したカウンタを使用して PLL 出力を変更します (c0 を c1 に移動など)。

図 2. QuartusII ソフトウェアでの ALTPLL_RECONFIG および ALTPLL メガファンクション



方法 1 による PLL リコンフィギュレーションのステップは次の通りです。

1. 4 ビット counter_type (C0-C9、M、N、CP/LF、VCO) を選択します。

 有効な counter_type 設定については、『*Phase-Locked Loops Reconfiguration (ALTPLL_RECONFIG) MegaFunction User Guide*』を参照してください。

2. ステップ 1 で選択されたカウンタに 3 ビット counter_param を選択します。有効なパラメータ設定については、表 1 を参照してください。

表 1. Stratix III ALTPLL_RECONFIG counter_param 設定 (1 / 2)

Counter Type	Counter Param 名	Counter Param 値	Data_in 幅 (ビット)
C0-C9: トップ / ボトム PLL C0-C6: レフト / ライト PLL	High count	000	8
	Low count	001	8
	Bypass	100	1
	Mode (奇数 / 偶数の分周)	101	1
チャージ・ポンプ / ループ・フィルタ	チャージ・ポンプ電流	000	3
	ループ・フィルタ抵抗	001	5
	ループ・フィルタ・コンデンサ	010	2
VCO	VCO ポスト・スケール	000	1

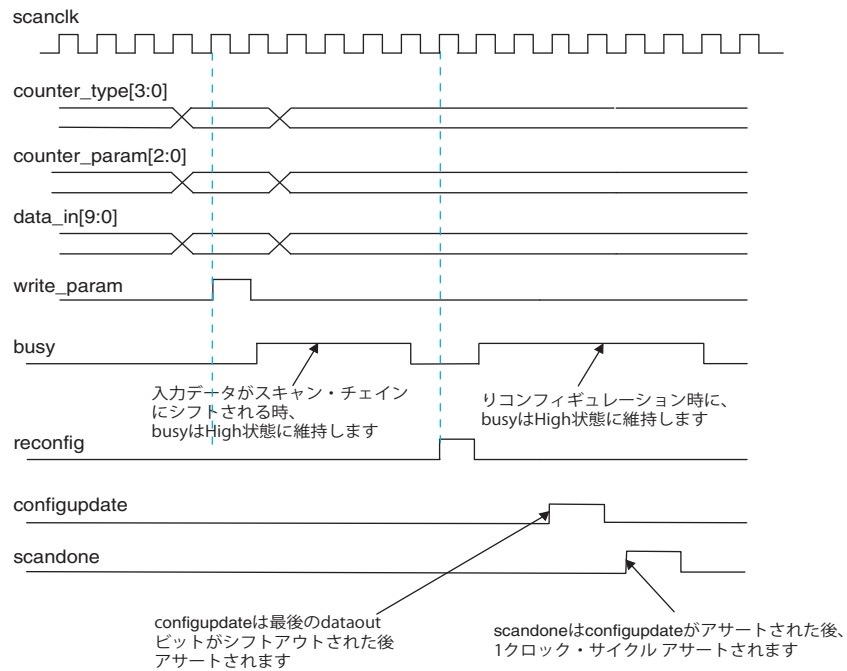
表 1. Stratix III ALTPLL_RECONFIG counter_param 設定 (2 / 2)

Counter Type	Counter Param 名	Counter Param 値	Data_in 幅 (ビット)
M/N カウンタ	High count	000	8
	Low count	001	8
	Bypass	100	1
	Mode (奇数 / 偶数の分周)	101	1
	Nominal count	111	9

- ステップ 2 での counter_param に 9 ビットの新しい data_in 値を選択します。
- ステップ 1 ~ 3 からの入力がスキャン・チェーンに書き込まれるように、write_param を 1 scanclk サイクル アサートします。
- busy 信号は、write_param がアサートされた直後の立ち上がりエッジにおいてアサートされます。パラメータの書込み中に、busy 信号はアサート状態を維持します。busy 信号がアサートされる時間はスキャン・チェーンにシフトされたデータによって異なります。
- busy 信号がディアサートされる時、ステップ 1 ~ 5 の手順で異なる counter_type をスキャン・チェーンに書き込むことができます。
- すべてのデータがスキャン・チェーンにシフトされたら、busy 信号がディアサートされた後、reconfig 信号をアサートし、スキャン・チェーンの内容によって PLL をリコンフィギュレーションします。

図 3 に示すタイミング図は、ALTPLL_RECONFIG メガファンクションによって PLL リコンフィギュレーションを実現するための counter_type、counter_param、datain、reconfig および busy 信号間の関係を示します。

図 3. 方法 1 による PLL リンコンフィギュレーションのタイミング図



PLL リンコンフィギュレーション：方法 2

ALTPLL_RECONFIG メガファンクションを **.mif** または **.hex** フォーマットの **.mif** ファイルによって初期化します。**.mif** ファイルは PLL リンコンフィギュレーション・スキャン・チェーンのビットマップです。reconfig 信号をパルスして ALTPLL メガファンクションを **.mif** ファイルの内容によって更新します。

counter_type、counter_param および datain ポートが使用されていないことを除き、このデザイン・セットアップは方法 1 に似ています。制御信号が適切にアサートされる限り、方法 1 と方法 2 を同一デザインに使用することも可能です。

.mif ファイルが有効な設定を持つことを確保するために、リンコンフィギュレーション後に PLL に新しい入出力設定で個別の ALTPLL メガファンクション・インスタンスを生成し、ALTPLL MegaWizard Plug-in Manager で **Generate a .mif file** を選択します。生成された **.mif** ファイルをデザインのリンコンフィギュレーションに使用します。

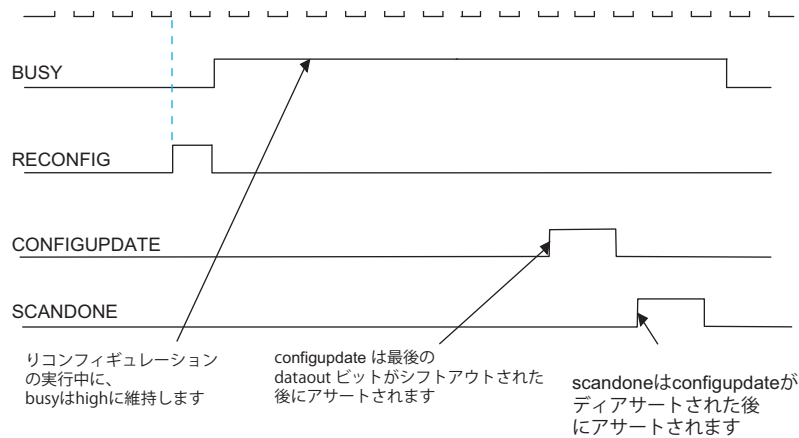
Stratix III デバイスの場合に、複数の **.mif** ファイルによる複数のリンコンフィギュレーションが可能です。MegaWizard Plug-in Manager を使用して、デザインを再コンパイルすることなく複数の **.mif** ファイルを生成できます。詳細は、11 ページの「アブリケーション例 1」を参照してください。

方法 2 による PLL リンコンフィギュレーションのステップは次の通りです：

1. 新しいリンコンフィギュレーション設定の **.mif** ファイルを生成します。
2. ALTPLL_RECONFIG メガファンクションを **.mif** ファイルに関連付けます。
3. reconfig 信号をアサートして、スキャン・チェーンおよび PLL カウンタを **.mif** ファイルの内容によって更新します。

図 4 に、ALTPLL_RECONFIG メガファンクションにおける .mif ファイルによるリコンフィギュレーションのタイミング図を示します。

図 4. ALTPLL_RECONFIG メガファンクションにおける .mif ファイルによるリコンフィギュレーションのタイミング図



リコンフィギュレーション信号

- write_param 信号は scanclk の立ち上がりエッジでサンプリングされます。ほかのクロック・サイクルでパラメータが偶然に再書き込まないように、write_param 信号は 1 scanclk サイクルのみアサートされる必要があります。
- reconfig 信号は scanclk の立ち上がりエッジでサンプリングされます。リコンフィギュレーション後にほかの PLL カウンタがリロードされないように、reconfig 信号は 1 scanclk サイクルのみアサートされる必要があります。
- read_param 信号がアサートされると、スキャン・チェーンの内容を scandataout ポートを通して読み出してシフト・アウトされる必要があることを示します。読み出されたビット位置およびビット数は counter_type と counter_param の組み合わせによって異なります。read_param 信号は scanclk の立ち上がりエッジでサンプリングされます。ほかのクロック・サイクルでパラメータがよけいに読み出さないように、read_param 信号は 1 scanclk サイクルのみアサートされる必要があります。スキャン・チェーン内容のコピーが読み出されます。read_param の動作後、スキャン・チェーンの内容が保持されます。
- busy 信号は read_param、write_param、または reconfig がアサートされた直後の scanclk 信号の立ち上がりエッジでアサートされ、動作完了までは High にしておきます。busy 信号がアサートされる間に、すべてのスキャン・チェーン入力が無視され、スキャン・チェーンの内容は busy 信号がデアサートされるまでに変更できません。

ALTPLL_RECONFIG メガファンクション・ポートの機能に関する説明は、*「Phase-Locked Loops Reconfiguration (ALTPLL_RECONFIG) MegaFunction User Guide」* に参照してください。

リコンフィギュレーション後に、スキャン・チェーン内容は変更されません。ほかのパラメータが前の値に維持するため、複数のリコンフィギュレーションをしながら、1 回に 1 つのパラメータを選択的に変更することができます。

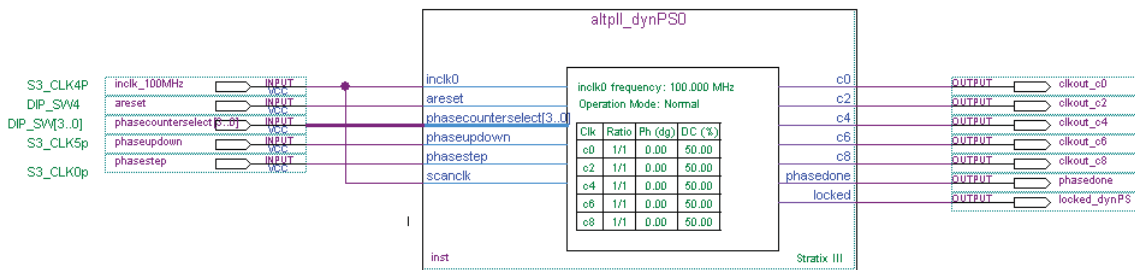
Quartus II による PLL ダイナミック位相シフト

Quartus II による PLL ダイナミック位相シフトの実装

ダイナミック位相シフト機能により、スキャン・チェーンをロードせずに個々の PLL 出力の出力位相をほかの PLL 出力や基準クロックに対してダイナミックに調整できます。位相は 1 度に VCO 周期の 1/8 ずつシフトされます。出力クロックは、このダイナミック位相シフトの動作中にアクティブです。

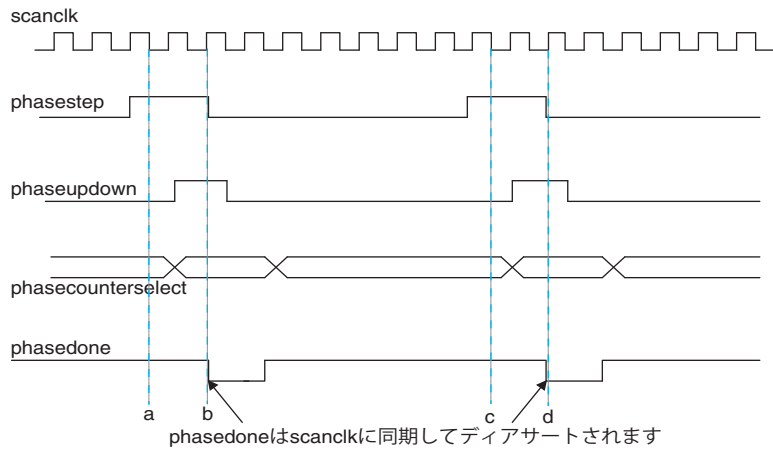
ALTPLL MegaWizard Plug-In Manager を使用して、ALTPLL メガファンクションのインスタンスでダイナミック位相シフト回路をイネーブルします (図 5 を参照)。

図 5. ダイナミック位相シフトがイネーブルした ALTPLL



ダイナミック位相シフトに必要な各信号間のタイミング関係については、図 6 のタイミング図を参照してください。

図 6. ダイナミック位相シフトのタイミング図



次はダイナミック位相シフトの信号：

- phasestep は scandclk の立ち下りエッジ (a、c) でサンプリングされ、最低 2 scandclk サイクル アサートする必要があります。

- phaseupdown および phasecounterselect は、phasestep がサンプリングされた直後 scanclk の二番目の立ち上がりエッジ (b、d) でサンプリングされず。
- phasedone は scanclk の二番目の立ち上がりエッジ (b、d) に同期してディアサートされます。phasedone は PLL ダイナミック位相シフトが終了した後でアサートされます。phasedone の low 時間は、VCO 周波数および scanclk 周波数によって、1 scanclk サイクルより多いまたは 1 scanclk サイクルより少ないこととなります。
- phasestep パルスごとに、一回の位相シフトが実行されます。phasestep パルス同士には、1 scanclk サイクル以上の間隔が必要です。
- ダイナミック位相シフト後に PLL をリセットすると、すべての PLL カウンタが元の位相設定に戻ります。

PLL リンコンフィギュレーションのスキュン・レジスタ・ビットマップ

上級 PLL ユーザーは、「StratixIII デバイス・ハンドブック Volume 1」の「StratixIII デバイスのクロック・ネットワークおよび PLL」の章を参照して、カウンタおよび位相シフト設定を手動で選択することができます。さまざまなカウンタ、ループ・フィルタおよびチャージ・ポンプの設定に対して、個別のコンフィギュレーション・ビット設定を決定した後、表 2 に示すビットマップのどおりビットを配置します。

表 2 に、トップ / ボトム PLL スキュン・チェーン・レジスタのビットマップを示します。レフト / ライト PLL の場合に、同じビットマップを基準にしてもかまいません。ただし、レフト / ライト PLL が 7 つの出力カウンタのみを有するため、スキュン・チェーンはより短いです。スキュン・チェーンにシフトされる最後のビットは Bit0 です。Bit233 は、トップ / ボトム PLL で最初にスキュン・チェーンにシフトされるビットです。Bit179 は、レフト / ライト PLL で最初にスキュン・チェーンにシフトされるビットです。各カウンタのデューティ・サイクルは、high_count および low_count ビットで設定できます。カウンタが奇数分周に設定したら、カウンタ奇数分周ビットが 1 に設定することを確認してください。奇数分周のデューティ・サイクルは以下のどおりにセットすべきです。

```
high_count = (CounterDivVal + 1)/2
low_count = CounterDivVal - high_count
奇数 / 偶数分周のビット = 1
```

Nominal count を使用して Mカウンタまたは Nカウンタをリコンフィギュレーションする場合、カウンタ・ビットは自動的に次のどおりに設定されます。

Nominal count が偶数の場合、カウンタ・ビット設定は：

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

Nominal count が奇数の場合、カウンタ・ビット設定は：

```
high_count = (Nominalcount + 1)/2
low_count = Nominalcount - high_count
奇数 / 偶数分周のビット = 1
```

Nominal count が 1 とする場合、カウンタ・ビット設定は：

```
Bypass bit = 1
```

表 2. PLL リコンフィギュレーション・スキャン・チェーンのビットマップ

ビット	PLL スキャン・チェーンのビットマップ								
Bit225-Bit233	C9_low_0	C9_low_1	C9_low_2	C9_low_3	C9_low_4	C9_low_5	C9_low_6	C9_low_7	C9_mode_odd/even
Bit216-Bit224	C9_high_0	C9_high_1	C9_high_2	C9_high_3	C9_high_4	C9_high_5	C9_high_6	C9_high_7	C9_mode_bypass
Bit207-Bit215	C8_low_0	C8_low_1	C8_low_2	C8_low_3	C8_low_4	C8_low_5	C8_low_6	C8_low_7	C8_mode_odd/even
Bit198-Bit206	C8_high_0	C8_high_1	C8_high_2	C8_high_3	C8_high_4	C8_high_5	C8_high_6	C8_high_7	C8_mode_bypass
Bit189-Bit197	C7_low_0	C7_low_1	C7_low_2	C7_low_3	C7_low_4	C7_low_5	C7_low_6	C7_low_7	C7_mode_odd/even
Bit180-Bit188	C7_high_0	C7_high_1	C7_high_2	C7_high_3	C7_high_4	C7_high_5	C7_high_6	C7_high_7	C7_mode_bypass
Bit171-Bit179	C6_low_0	C6_low_1	C6_low_2	C6_low_3	C6_low_4	C6_low_5	C6_low_6	C6_low_7	C6_C6ode_odd/even
Bit162-Bit170	C6_high_0	C6_high_1	C6_high_2	C6_high_3	C6_high_4	C6_high_5	C6_high_6	C6_high_7	C6_C6ode_bypass
Bit153-Bit161	C5_low_0	C5_low_1	C5_low_2	C5_low_3	C5_low_4	C5_low_5	C5_low_6	C5_low_7	C5_mode_odd/even
Bit144-Bit152	C5_high_0	C5_high_1	C5_high_2	C5_high_3	C5_high_4	C5_high_5	C5_high_6	C5_high_7	C5_mode_bypass
Bit135-Bit143	C4_low_0	C4_low_1	C4_low_2	C4_low_3	C4_low_4	C4_low_5	C4_low_6	C4_low_7	C4_mode_odd/even
Bit126-Bit134	C4_high_0	C4_high_1	C4_high_2	C4_high_3	C4_high_4	C4_high_5	C4_high_6	C4_high_7	C4_mode_bypass
Bit117-Bit125	C3_low_0	C3_low_1	C3_low_2	C3_low_3	C3_low_4	C3_low_5	C3_low_6	C3_low_7	C3_mode_odd/even
Bit108-Bit116	C3_high_0	C3_high_1	C3_high_2	C3_high_3	C3_high_4	C3_high_5	C3_high_6	C3_high_7	C3_mode_bypass
Bit99-Bit107	C2_low_0	C2_low_1	C2_low_2	C2_low_3	C2_low_4	C2_low_5	C2_low_6	C2_low_7	C2_mode_odd/even
Bit90-Bit98	C2_high_0	C2_high_1	C2_high_2	C2_high_3	C2_high_4	C2_high_5	C2_high_6	C2_high_7	C2_mode_bypass
Bit81-Bit89	C1_low_0	C1_low_1	C1_low_2	C1_low_3	C1_low_4	C1_low_5	C1_low_6	C1_low_7	C1_mode_odd/even
Bit72-Bit80	C1_high_0	C1_high_1	C1_high_2	C1_high_3	C1_high_4	C1_high_5	C1_high_6	C1_high_7	C1_mode_bypass
Bit63-Bit71	C0_low_0	C0_low_1	C0_low_2	C0_low_3	C0_low_4	C0_low_5	C0_low_6	C0_low_7	C0_mode_odd/even
Bit54-Bit62	C0_high_0	C0_high_1	C0_high_2	C0_high_3	C0_high_4	C0_high_5	C0_high_6	C0_high_7	C0_mode_bypass
Bit45-Bit53	N_low_0	N_low_1	N_low_2	N_low_3	N_low_4	N_low_5	N_low_6	N_low_7	N_mode_odd/even
Bit36-Bit44	N_high_0	N_high_1	N_high_2	N_high_3	N_high_4	N_high_5	N_high_6	N_high_7	N_mode_bypass
Bit27-Bit35	M_low_0	M_low_1	M_low_2	M_low_3	M_low_4	M_low_5	M_low_6	M_low_7	M_mode_odd/even
Bit18-Bit26	M_high_0	M_high_1	M_high_2	M_high_3	M_high_4	M_high_5	M_high_6	M_high_7	M_mode_bypass
Bit15-Bit17	CP_0	CP_1	CP_2						
Bit10-Bit14	Reserved_0	Reserved_1	Reserved_2	Reserved_3	Reserved_4				
Bit9	VCO_postscale_0								
Bit4-Bit8	LF_R_0	LF_R_1	LF_R_2	LF_R_3	LF_R_4				
Bit2-Bit3	LF_C_0	LF_C_1							
Bit0-Bit1	Reserved_5	Reserved_6							

次の項では、複数の **.mif** ファイルによる PLL のリコンフィギュレーションおよび **PLL** ダイナミック位相シフトの使用を含む2つのアプリケーションについて説明します。ここで提供されるデザイン例では、実装の詳細が示されます。このアプリケーション・ノート用のデザイン例は [アルテラ・ウェブサイトの資料ページ](#) からダウンロードできます。

アプリケーション例 1

マルチレート・ライン・カードにおける周波数プリスケアラ

マルチレート・ライン・カード・アプリケーションにおいては、**Stratix III PLL** は異なるデータ・レートをサポートするために、複数の周波数にロックする必要があります。このアプリケーションは、2つの **PLL** システムをカスケードで接続して構成しており、アップストリーム **PLL** は周波数プリスケアラとして働いて、ダウンストリーム **PLL** に一定の周波数を提供します（例えば、**311 MHz**）。このプリスケアラのデザイン例はここで示されます。プリスケールされた周波数は **Stratix III PLL** のリコンフィギュレーション機能によって実現されます。周波数プリスケアラがデータレートを切り換えるたびに、**PLL** は新しい入力クロックおよび出力クロック関係にリコンフィギュレーションされます。

デザイン例 1

ここで示される周波数プリスケアラの実装は **SONET** アプリケーション用のものです。**SONET** データ・レートのサンプルは **622 MHz**、**155.5 MHz** および **38.875 MHz** です。この周波数プリスケアラはすべての入力データ・レートに **311 MHz** の出力を提供する必要があります。入力信号は、データ・レート変更の制御に使用されます。ここで示される周波数プリスケアラ例は、複数の **.mif** ファイルを使用してリコンフィギュレーションをして、6 ページの「**PLL リコンフィギュレーション：方法 2**」で述べられたステップで **PLL** リコンフィギュレーションを実現します。

このデザインは、**ALTPLL** メガファンクション、**ALTPLL_RECONFIG** メガファンクション、3つの異なる入力周波数に対応する3つの **.mif** ファイルを持つ3つの **ROM**、およびステート・マシンへの信号が変わるたびに適切な **.mif** ファイルを選択して適切な制御信号を出力してダイナミック・リコンフィギュレーションを実現するステート・マシンで構成されています。ステート・マシン内のロジックは、自動的に必要な信号を生成して **311 MHz** の固定周波数を得ます。

ブロック図

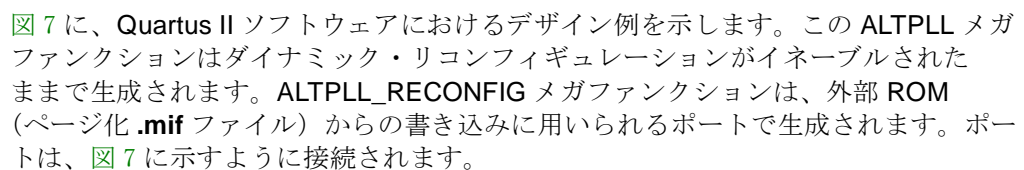
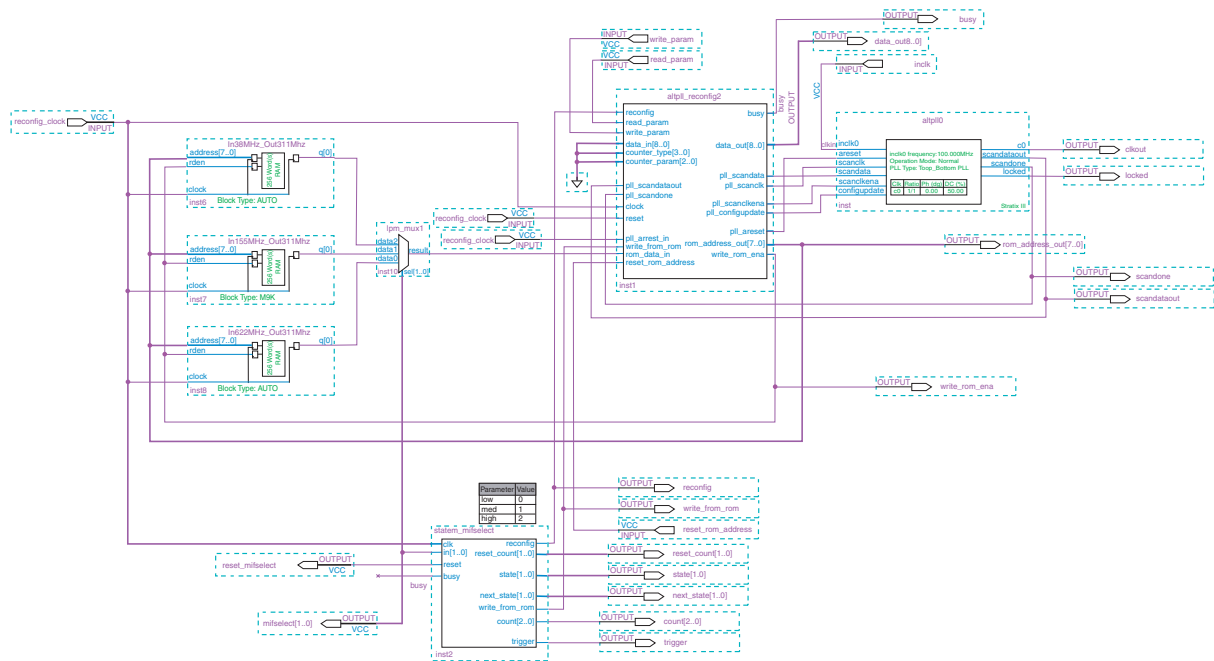
 **図 7** に、**Quartus II** ソフトウェアにおけるデザイン例を示します。この **ALTPLL** メガファンクションはダイナミック・リコンフィギュレーションがイネーブルされたままで生成されます。**ALTPLL_RECONFIG** メガファンクションは、外部 **ROM**（ページ化 **.mif** ファイル）からの書き込みに用いられるポートで生成されます。ポートは、**図 7** に示すように接続されます。

図 7. ALTPLL_RECONFIG メガファンクションのページ化 .mif 機能による PLL リコンフィギュレーション




プロシージャ

次のステップでは、11 ページの「デザイン例 1」のデザイン・フローについて説明します。

1. ダイナミック・リコンフィギュレーションがイネーブルされたままで ALTPLL メガファンクションをインスタンス化します。PLL を 3 つのデータ・レートのうちどれかでコンフィギュレーションします（この場合では、38.875 MHz の入力周波数、311 MHz の出力周波数）。デザインに必要とされる PLL カウンタ出力を選択します。このデザインでは、c0 カウンタ出力が使用されます。
MegaWizard Plug-In Manager を終了しないでください。
2. MegaWizard Plug-in Manager の 5 ページでの **Generate a Configuration File** ボタンを使用してこの PLL コンフィギュレーションの .mif ファイルを生成します。
3. MegaWizard Plug-in Manager で、PLL 入力周波数を 622 MHz に、カウンタ出力を 311 MHz に変更して .mif ファイルをもう 1 回生成します。ステップ 2 で生成されたファイルを上書きしないように、異なるファイル名で命名します。
4. MegaWizard Plug-in Manager で、PLL 入力周波数を 155 MHz に、カウンタ出力を 311 MHz に変更して .mif ファイルをもう 1 回生成します。ステップ 3 で生成されたファイルを上書きしないように、異なるファイル名で命名します。
5. これで、SONET プリスケラ・アプリケーション用の 3 つの異なる入力データレートを持つ .mif ファイルが生成されました。
6. MegaWizard Plug-In Manager で PLL 設定を保存します。これで、PLL は 155 MHz の入力周波数および 311 MHz の出力周波数にコンフィギュレーションされました。

7. ALTPLL_RECONFIG メガファンクションをインスタンス化します。**add ports to write to the scan chain from external ROM during run-time** を選択して複数の .mif ファイルをサポートする機能をイネーブルします。図 7 に示すように、すべてのポートを ALTPLL メガファンクションに接続します。
8. 1 ビット幅のデータ・バスおよび 8 ビット幅のアドレス・バスで 3 つの ROM をインスタンス化します。すべての ROM にリード・イネーブル (rden) ポートを作成します。ステップ 1 ~ 4 で作成した .mif ファイルで 3 つの ROM を初期化します。

 ROM インスタンス化について詳しくは、*「ROM Megafunction User Guide」* を参照してください。

9. 3 入力の lpm_mux をインスタンス化して、図 7 に示すようにポートを接続します。
10. 入力 mifselect が変える時に適切な ROM を選択するシンプルなステート・マシンを作成します。図 7 に示すようにポートを接続します。ステート・マシンは、mux への select 信号によって ROM を選択して、適切な制御信号を ALTPLL_RECONFIG メガファンクションに出力することでリコンフィギュレーションを実現します。リコンフィギュレーションは、busy 信号がディアサートされる時のみ実行できます。busy が high の状態では、mifselect の変化が無視されます。

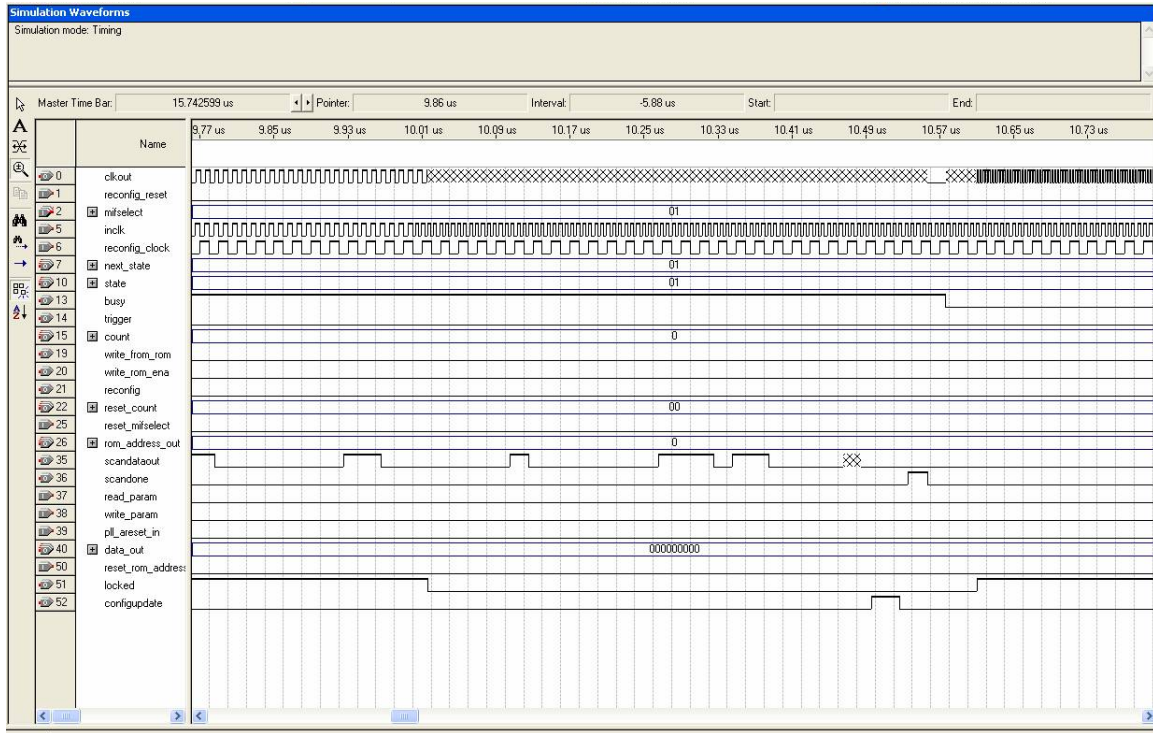
表 3 に、入力信号および入力信号がステート・マシン出力に対する影響の詳細を記載します。

表 3. デザイン例 1：ステート・マシンの入力信号および出力信号

Mifselect	入力周波数—出力周波数	影響を受けるカウンタ
00	38.875 MHz — 311 MHz	C0
01	155 MHz — 311 MHz	C0
10	622 MHz — 311 MHz	C0
11	変更なし	なし

11. 入力を mifselect に適用して、デザインをコンパイルしてシミュレーションします。mifselect が変えるたびに、外部 ROM からの異なる .mif ファイルはスキャン・チェーンにロードされ、PLL はスキャン・チェーンの内容でリコンフィギュレーションされます。図 8 にシミュレーションの波形を示します。

図 8. デザイン例 1：シミュレーション波形



アプリケーション例 2

ALTPLL メガファンクションによるダイナミック位相シフト


アルテラ ALTMEMPHY メガファンクションにより、Stratix III デバイスに物理層インタフェース (PHY) を迅速に作成できます。PHY は外部メモリとユーザー・ロジック間でデータを安全に転送します。Stratix III ALTMEMPHY メガファンクションは、初期キャリブレーション・シーケンスをサポートして、FPGA および外部メモリ・デバイスにおけるプロセス変動による影響を除去します。キャリブレーション・プロセスにより、セットアップおよびホールド・マージンが最大になるように、再同期化クロックの位相がデータ有効ウィンドウの中央に集められます。

詳細は、「ALTMEMPHY メガファンクション・ユーザーガイド」を参照してください。

ALTMEMPHY でのオート・キャリブレーション・コントローラは、クロックのダイナミック位相シフトを用いて、有効なデータ・キャプチャ・ウィンドを決めます。オート・キャリブレーション・コントローラには、入力用のデータ・ストリームおよびクロックがあります。このクロックはリコンフィギュレーション可能な PLL に供給され、VCO 期間の 1/8 単位で位相シフトされます。このコントローラは、位相シフトされたクロックを使用してデータをサンプリングします (所期のデータと比較される)。コントローラは、有効なデータ・キャプチャ・ウィンドウが決められるまでにクロックの位相をシフトします。

デザイン例 2

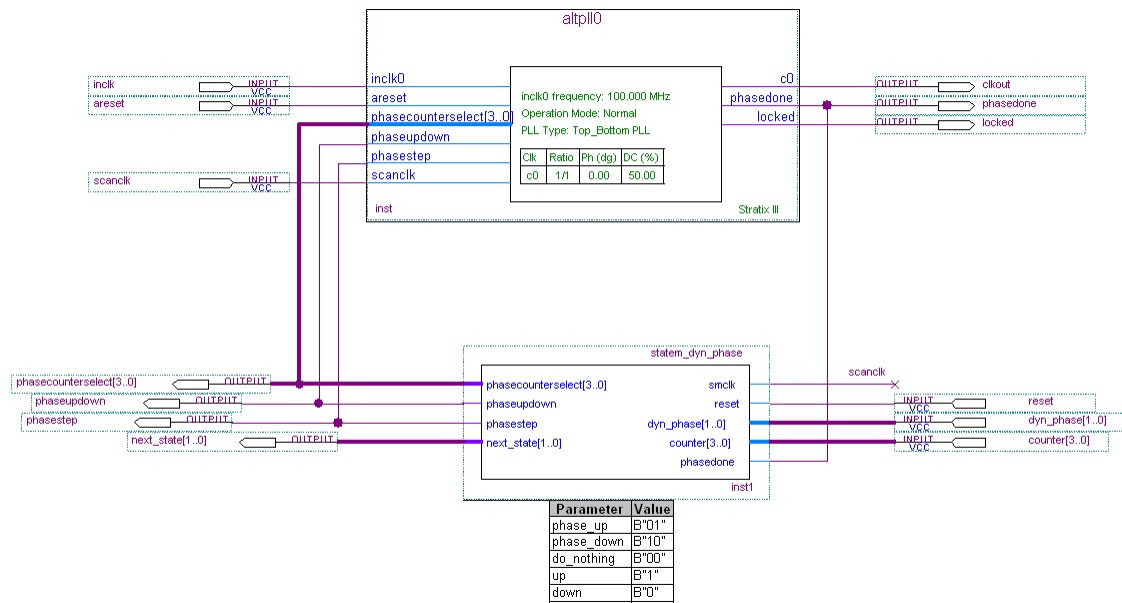
ここで述べられるアプリケーションは、ダイナミック位相シフトを用いてクロック・エッジをシフトします。このデザインは、`dyn_phase` と `counter` 入力を持つシンプルなステート・マシンおよび `ALTPLL` メガファンクションで構成されています。`counter` 信号は PLL への `phasecounterselect` 入力となります。`dyn_phase` に応じて、カウンタ出力位相を前または後にシフトできます。ステート・マシン内のロジックは位相シフトを実現するのに必要な信号を自動的に生成します。

 `phasecounterselect` 設定については、「StratixIII デバイス・ハンドブック Volume 1」の「StratixIII デバイスのクロック・ネットワークおよび PLL」の章を参照してください。

ブロック図

図 9 に、Quartus II ソフトウェアにおけるデザイン例を示します。この `ALTPLL` メガファンクションはダイナミック位相シフト・リコンフィギュレーションがイネーブルされたままで生成されます。ダイナミック位相シフト・コントロール・ポートは示されるようにステート・マシンに接続されます。この例では、PLL のすべてのカウンタをターゲットとします。

図 9. Stratix III ダイナミック位相シフト



プロシージャ

次のステップでは、15 ページの「デザイン例 2」のデザイン・フローについて説明します。

1. ダイナミック位相シフトがイネーブルされたままで `ALTPLL` メガファンクションをインスタンス化します。メガファンクションで PLL を目的の入力周波数および出力周波数にコンフィギュレーションします。

- 図 9 に示すように、入力と出力が付いたステート・マシンを作成します。入力信号およびそのステート・マシン出力に対する影響について詳しくは、表 4 を参照してください。dyn_phase が 10 になると、ステート・マシンは適切な信号を出力して位相を前進してシフトします。dyn_phase が 11 になると、ステート・マシンは適切な信号を出力して位相を後退してシフトします。これらの出力信号の設定方法について詳しくは、ページ 8 の図 6 のタイミング図を参照してください。

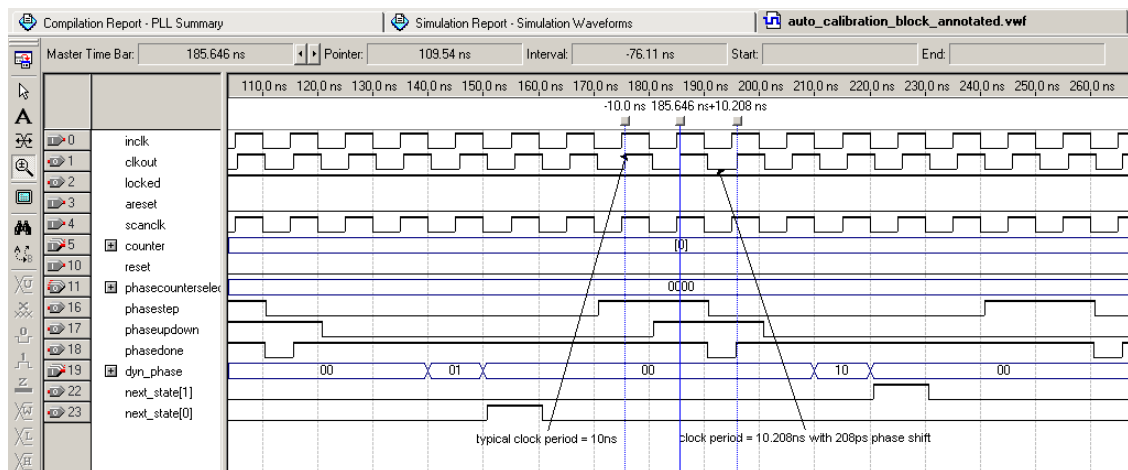
表 4. デザイン例 2 ステート・マシンの入力ポートおよび出力への影響

カウンタ	dyn_phase	位相	影響を受けるカウンタ
0	10	1 ステップ前進	全部
0	11	1 ステップ後退	全部
0	00	変更なし	なし
0	01	変更なし	なし

- 入力をカウンタおよび dyn_phase に適用し、デザインをコンパイルしてシミュレートします。dyn_phase の変更は、phasedone がアサートされる時のみ実行する必要があります。phasedone が low 状態では、dyn_phase への変更が無視されます。図 10 にシミュレーションの波形を示します。
- PLL カウンタ c0 はこのデザインで示されたが、ステート・マシンへのカウンタ入力はすべての PLL 出力カウンタをターゲットとします。ステート・マシンへのカウンタ入力を変えて特定の PLL 出力に影響することができます。

phasecounterselect 設定については、「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスのクロック・ネットワークおよび PLL」の章を参照してください。

図 10. デザイン例 2：ダイナミック位相シフト・シミュレーション出力



ダイナミック位相シフトが活用できるもう 1 つのアプリケーションは Nios® II アプリケーションです。Nios II プロセッサの使用時に、ほとんどのアプリケーションはオフチップ・メモリ・インタフェースを必要とします。メモリ・デバイスにおける不適切なクロッキングは、コードの実行不能やメモリにおけるバック・ツー・バック・トランザクションの実行不能などの問題を誘発する可能性があります。ダイナミック位相シフトにより、PLL を調整してメモリ・デバイスをアクセスする有効な信号ウィンドウを決めることができます。ALTPLL メガファンクションから PLL パラメータを読み出すのに必要な周辺装置を収容するためのハードウェアは、SOPC Builder によって作成できます。PLL のオフチップ・メモリ・クロック出力のシフト、SDRAM メモリのテスト、および SDRAM 有効信号ウィンドウの境界と中心の計算を実行するデザインの一部とするソフトウェアは、C 言語で記述された Nios IDE プロジェクトとして作成できます。

デザインの検討事項

PLL リコンフィギュレーションを使用するときは、以下の内容について考慮する必要があります。

- プリスケールおよびフィードバック・カウンタの設定 (m 、 n)、またはチャージポンプ/ループ・フィルタの設定を変更すると、PLL VCO 周波数が影響を受けるため、PLL を基準クロックに再ロックしなければならない場合があります。
- m カウンタの位相シフト設定を変更すると、基準クロックに対する出力クロックの位相関係が変化するため、PLL も再ロックする必要があります。プリスケールおよびフィードバック・カウンタの設定 (m 、 n) を変更した場合の正確な影響は、設定の変更内容によって異なりますが、どの変更を行った場合でも通常は再クロックが必要です。
- スキャン・チェーンの初期内容を指定する .mif ファイルを使用しない場合は、デバイスがユーザー・モードに入る時にスキャン・チェーンは空白となります。PLL はプログラマ・オブジェクト・ファイル (.pof) にプログラミングされた設定に基づいて初期状態にコンフィギュレーションされます。
- fm 位相シフト設定を使用して位相シフトを追加すると、すべての PLL クロック出力が基準クロックに対して制御され、効果的に負の位相シフトが追加されます。これは、 fm がフィードバック・パスに存在するためです。
- ループ・エレメント (m 、 n 、 m カウンタ位相、 lcp 、 R 、 C) を変更する場合に、アルテラは、ALTCLKCTRL メガファンクションにある $clkna$ 信号を使用して、ロジック・アレイへの PLL 出力をディセーブルすることを推奨しています。これにより、システム・ロジックに影響を与える過度の高周波数状態がなくなります。
- ポスト・スケール・カウンタ (C) や位相を変更しても、PLL ロックや VCO 周波数は影響を受けません。位相シフトの分解能は、常に最小の増分ステップ (VCO 周期の $1/8$ に等しい) を持つ VCO 周波数の関数としています。
- 出力クロック間の位相関係が重要な場合、アルテラは $areset$ 信号を使用して PLL を再同期化することを推奨しています。これにより、すべての内部 PLL カウンタがリセットされ、ロック・プロセスが再開されます。
- Stratix III PLL スキャン・チェーンでは自走式の $scanclk$ がサポートされているため、クロックの開始と停止を厳密に制御する必要はありません。
- m カウンタまたは n カウンタの値の変化はすべての出力クロック周波数に影響を与えます。また、出力カウンタは個別でリコンフィギュレーションできます。

- scandone 信号は scanclk のポジティブ・エッジと同期し、リコンフィギュレーションが完了した後 ALTPLL により最低 1 クロック・サイクル アサートされません。

デザイン例

デザイン例のダウンロードおよび使用については、以下の指示を参照してください。

デザイン例 1：マルチレート・ライン・カードにおける周波数プリスケアラ

Design Example 1 (an454_prescaler_de1.zip) を解凍して Quartus II ソフトウェアでコンパイルします。この例では、3 つの .mif ファイルはすでにセットアップされています。この例と共に提供されるベクタ波形ファイル (.vwf) を使用してタイミング・シミュレーションを実行します。mifselect が変更された後、PLL はリコンフィギュレーション後にロックを失うことがあります。ロックが回復した後、PLL 出力周波数は 38.875 MHz、155.5 MHz または 622 MHz の異なる周波数に対して、常に 311 MHz となります。

デザイン例 2：ALTPLL メガファンクションによるダイナミック位相シフト

Design Example 2 (an454_altpll_dynphase_de2.zip) を解凍して Quartus II ソフトウェアでコンパイルします。この例と共に提供される .vwf ファイルを使用してタイミング・シミュレーションを実行します。この例は位相を 2 回上昇させるように設定されます。PLL カウンタ出力の変化を観察することができます。エッジは、最初の位相シフト後に 1/8 の VCO 周期、二回目の位相シフト後に 1/4 の VCO 周期を進めるようにシフトされます。

結論

PLL リコンフィギュレーションは、PLL クロック出力周波数を変化でき、そして任意のステージでも位相シフトができる強力な機能です。ロック状態の喪失やグリッチ、出力位相関係などの重要な検討事項は、PLL カウンタの設定および位相シフトの設定に影響を与えます。PLL のリコンフィギュレーション時間は通常 15ms 未満なので、各動作モードにすばやく切り換え可能です。こうした柔軟性により、Stratix III は優れたクロック管理システムとなります。

参考資料

このアプリケーション・ノートでは、以下のドキュメントを参照しています。

- 「StratixIII デバイス・ハンドブック Volume 1」の「Stratix III デバイスのクロック・ネットワークおよびPLL」の章
- ALTPLL メガファンクション ユーザーガイド
- Phase-Locked Loops Reconfiguration (ALTPLL_RECONFIG) MegaFunction User Guide
- ROM Megafunction User Guide
- ALTMEMPHY メガファンクション・ユーザーガイド

改訂履歴

表 5 に、このアプリケーション・ノートの改訂履歴を示します。

表 5. 改訂履歴

日付およびリビジョン	変更内容	概要
2009 年 5 月 ver 1.2	<ul style="list-style-type: none">■ 「PLL リコンフィギュレーション: 方法 1」の項を更新■ 図 3 および図 4 を置き換え■ 「リコンフィギュレーション信号」の項を更新■ 表 2 を更新	—
2007 年 10 月 ver 1.1	<ul style="list-style-type: none">■ 図 4 を変更	—
2007 年 10 月 ver 1.0	<ul style="list-style-type: none">■ 初版	—



101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support
www.altera.com/support

Copyright © 2009 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.



I.S. EN ISO 9001