

## はじめに

このアプリケーション・ノートでは、複数のレガシー DDR2 コントローラを1個の FPGA で設計するためのステップを詳細に説明しています。このアプリケーション・ノートを読めば、PLL および DLL リソースの割り当て、タイミング解析、デバッグなどの重要なデザイン検討事項を理解できるようになるはずです。



ALTMEMPHY ベースの複数のコントローラを実装する場合については、ここでは説明していませんので、「AN 462: ALTMEMPHY メガファンクションを使用した複数のメモリ・インタフェースの実装」を参照してください。

複数コントローラのデザイン・フローは、DDR/DDR2 SDRAM コントローラを使用した2コントローラ・デザインと3コントローラ・デザインの例を示しながら説明しています。

- 2つの独立したコントローラ (図 1)
  - 1つがデバイスのトップに、もう1つがデバイスのボトムにあります。
  - コントローラ間でリソースの共有なし
- 3つのコントローラ (図 2)
  - 2つがデバイスのトップに、もう1つがデバイスのボトムにあります。
  - デバイスのトップにある2つのインタフェースのためのPLLおよびDLLリソースの共有

## デザインの 詳細

この項では、メモリ周波数、メモリ・インタフェース幅、コントローラの PLL 数などのデザイン仕様を説明します。正しいデバイスを使用することが最大性能の達成に役立つため、これらの詳細をあらかじめ把握しておくことが大切です。以下に仕様を示します。

- ターゲット・デバイス: EP2S60F1020C3
- DDR2 コントローラ・バージョン: Quartus® II 6.1 以降
- Quartus II ソフトウェア: バージョン 6.1 以降
- コントローラ・インタフェース周波数: デザインのすべてのコントローラで 266.67 MHz
- インタフェースの幅: 各コントローラで 32 ビット
- PLL 数: 両デザインの各コントローラとも 2 個 (システム PLL および フィードバック PLL)

図 1 に 2 コントローラ実装を示します。このデザインでは、各コントローラは独自の PLL リソースと DLL リソースを備えています。

図 1. 2 コントローラの実装

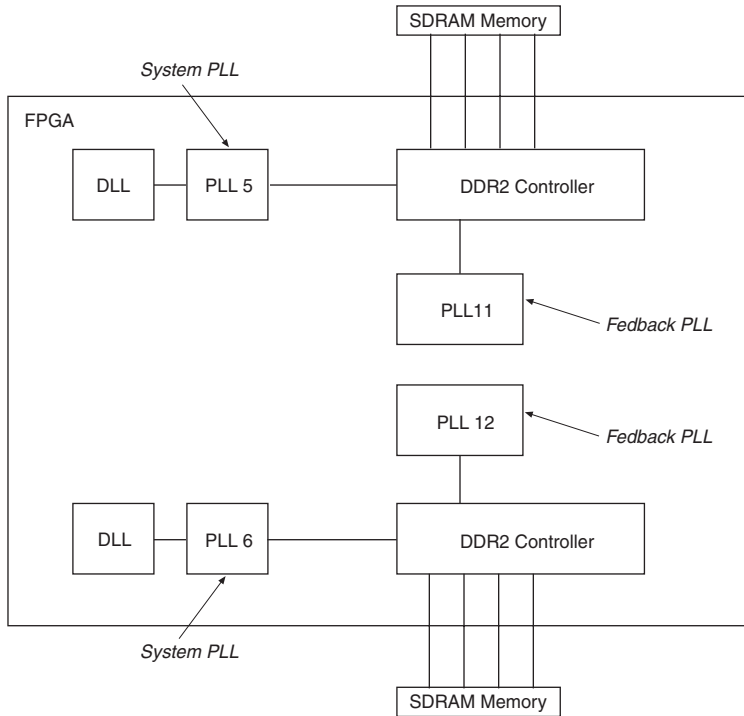
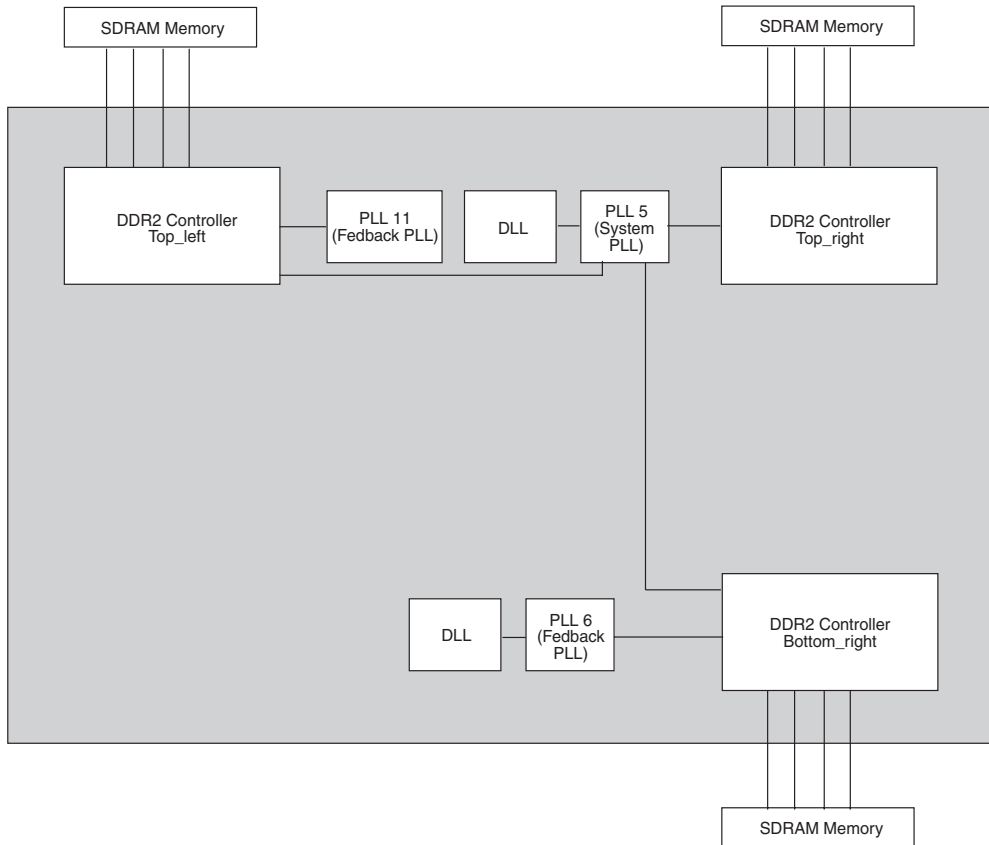


図 2 に 3 コントローラ実装を示します。この実装で、DLL リソースと PLL リソースは 2 つのコントローラで共有されます。



3つのコントローラで共有されるクロック・リソースについて詳しくは、[図 4](#)を参照してください。

図 2. 3 コントローラ実装



## デザインの検討事項

設計者がカスタム・ロジックとコントローラ間で、FPGA のどのリソースを共有する必要があるかを知っておくことが大切です。以下のデザイン検討事項は、2 または 3 コントローラ・デザインに適用されます。

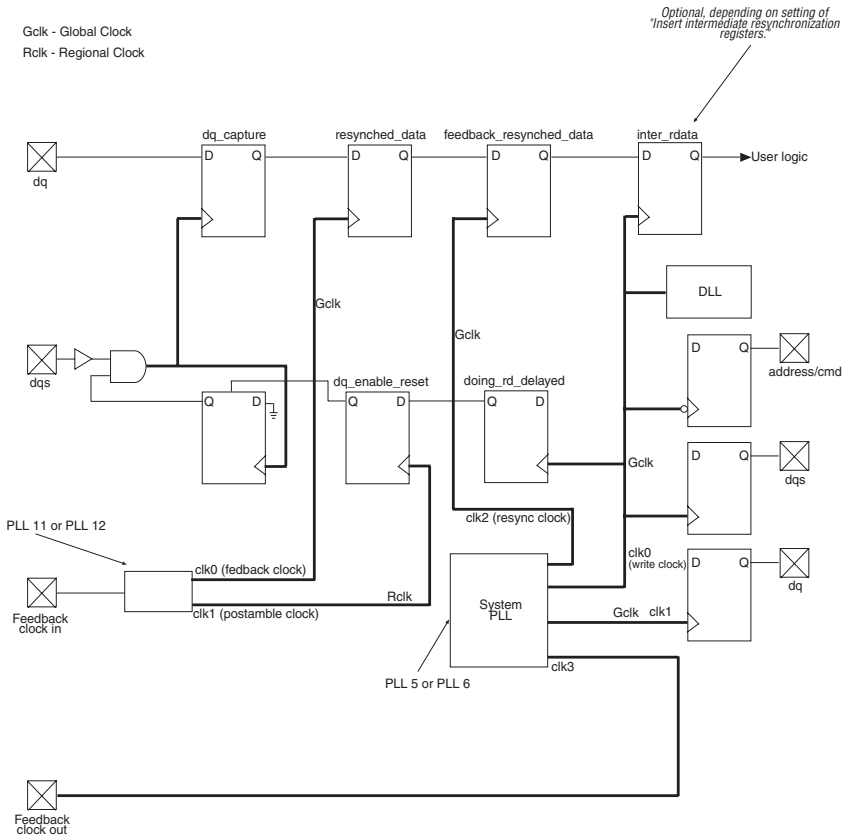
1. 各インタフェースに FPGA からメモリに 1 つまたは複数のクロック・ペア (`clk_to_sdram[]` および `clk_to_sdram_n[]`) が使用できること。
2. 各コントローラに 1 個のシステム PLL と 1 個のフィードバック PLL が使用可能なこと。各コントローラで、システム PLL がシステム・クロックを生成し、フィードバック PLL がフィードバック・クロックを生成します。図 3 に、200 MHz 以上のレガシー・ベース・メモリ・インタフェースのクローズ・タイミングに関してアルテラが推奨する構成を示します。



フィードバックおよびシステム PLL の機能について詳しくは、「[DDRおよびDDR2 SDRAM コントローラ・コンパイラ・ユーザガイド](#)」の「付録 A、再同期化」の項を参照してください。

図 3 に、システム PLL およびフィードバック PLL としてコンフィギュレーションされた PLL を含む、2 個の PLL リード・データパスの実装を示します。

図 3. フィードバック PLL とシステム PLL の両方を使用した DDR2 SDRAM コントローラ



## DLL リソースの検討事項

Stratix® II および Stratix II GX シリーズの FPGA デバイスは、FPGA のトップで 1 個の DLL、ボトムで 1 個の DLL をサポートします。したがって、すべてのコントローラが共通 DLL からの出力を共有しなければならないため、デバイスの同じサイドにあるメモリ・コントローラが同じ周波数で動作する必要があります。

## PLL リソース の検討事項



固定数の PLL 出力タップしか使用できないため PLL リソースは重要です。グローバル・クロック・リソースとリージョナル・クロック・リソースは、数に制限があり、またデバイスに応じて異なるためこれらを選択するときには注意が必要です。ここでは、各種クロックに使用するリソースのタイプについて説明します。

PLL リソースについては、「[Stratix II デバイス・ハンドブック volume 2](#)」の「[Stratix II および Stratix II GX デバイスの PLL](#)」の章にある「enhanced PLL」および「fast PLL」の項を参照してください。

2 コントローラ・デザインでは、2つのコントローラが独立しているため、PLL は IP ツール・ベンチでインスタンス化され、変更する必要はありません。



両方のメモリ・インターフェースが同じ周波数の場合は、PLL リソースを共有できます。



2 個の独立した PLL から出力されるクロックによって駆動されるデータが境界を越える必要がある場合は、適切なデータ同期化に配慮しなければなりません。

3 コントローラ・デザインでは、3つすべてのコントローラをドライブするのに必要な PLL の最小タップ数は 11 です。[表 1](#) に 3 コントローラのデザイン例に必要なクロックについて説明します。

クロック名	説明	クロック数
システム・クロック	1つのシステム・クロックで3つのコントローラをすべてドライブ。	1
ライト・クロック	1つのライト・クロックで3つのコントローラをすべてドライブ。	1
再同期化クロック	各コントローラに1つの再同期化クロックがあります。(1)	3
フィードバック再同期化クロック	各コントローラに1つのフィードバック再同期化クロックがあります。(1)	3
ポストアンブル・クロック	各コントローラに1つのポストアンブル・クロックがあります。(1)	3
合計		11

表 1 の注：

- (1) 2つ以上のコントローラが1つの再同期化 / フィードバック / ポストアンブル・クロックを共有する場合は、外部メモリのボード・トレース遅延およびボード・レイアウトにマッチング制約を課す必要があります。このような制約を克服するには、3つの独立した再同期化 / フィードバック / ポストアンブル・クロックを使用します。[図 4](#) に示すとおり、3 個の enhanced PLL から 11 個の PLL タップが選択されます。

I/O バンク 3 および 4 の DLL 基準クロックは、enhanced PLL5 またはデバイスの適切なサイドにある専用クロック入力ピン CLK[4-7] または CLK[12-15] によってのみドライブ可能です。I/O バンク 7 および 8 の DLL 基準クロックは、enhanced PLL6 またはデバイスの適切なサイドにある専用クロック入力ピンによってのみドライブ可能です。例えば、PLL 入力基準クロック周波数が 100 MHz で、所要出力周波数が 266.67 MHz の場合、乗算係数は 8/3 になります。この場合、266 MHz の周波数を持つ必要がある DLL 基準クロックは、どの専用クロック入力ピンでもドライブできないので、enhanced PLL5 または enhanced PLL6 でドライブしなければなりません。

フィードバック・クロックは信号がチップから出力されてからチップに戻るため、一般にジッタが大きくなるため、DLL 基準をドライブするのにフィードバック・クロックは使用しないでください。システム PLL はトップのコントローラに対しては PLL5、ボトムのコントローラに対しては PLL6 でなければなりません。

図 4 に、3 コントローラ・デザインにおける PLL リソースの用途を示します。この図はどの PLL タップがグローバル / リージョナル・リソースとして使用されるかも示しています。

図 4. PLL リソースの割り当て

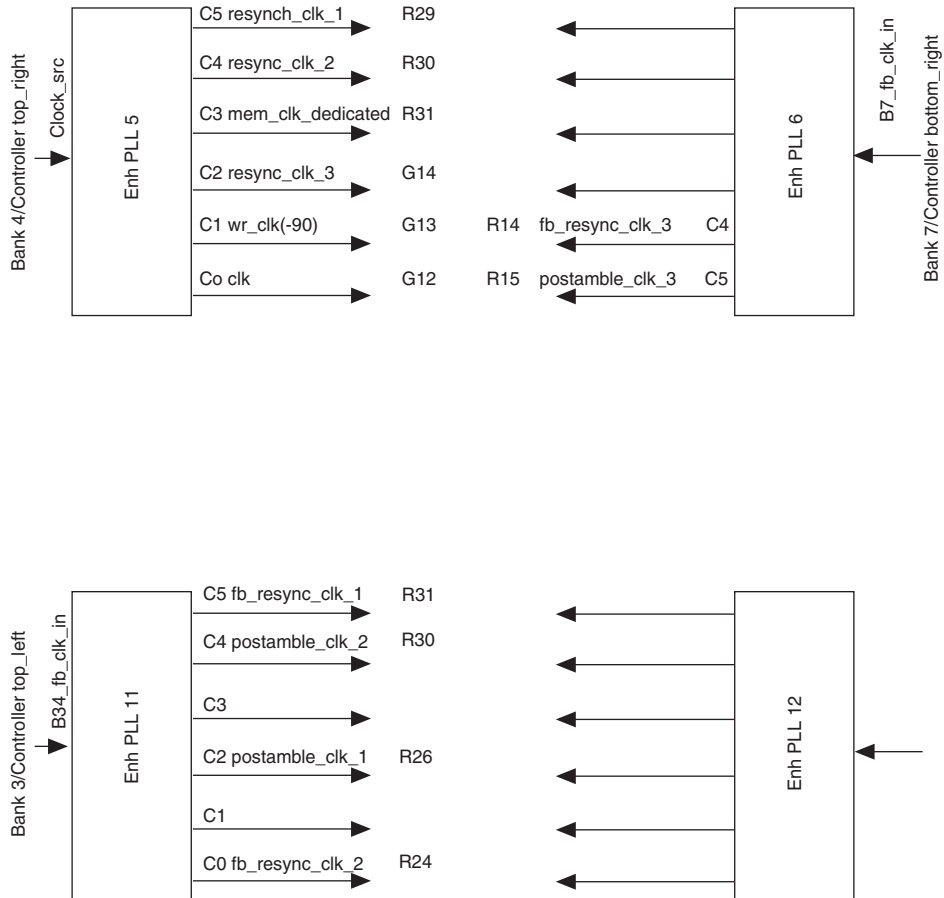


図 4 の注:

- (1) `mem_clk-dedicated` は、FPGA デザインではオプションですが、HardCopy® II アプリケーションで `clk_to_sdram` 出力をドライブする必要があります。

図 4 に示すとおり、3 個の enhanced PLL を使用する場合は、グローバル配線リソースが最少数で済みます。表 2 は、3 コントローラ・デザインで fast PLL またはより少ない enhanced PLL を使用する前に、検討する必要があるトレードオフについて説明しています。

PLL の可能性	注記
2 個の enhanced PLL — 両方ともトップ	デバイスのいずれかのサイドから 4 つのチップワイドのグローバル信号のみドライブ可能です。ただし、チップのボトムにあるコントローラに対するシステム・クロック、ライト・クロック、再同期化クロック、フィードバック再同期化クロック、およびポストアンブル・クロックはすべてグローバル・クロックを必要とします。これは使用可能な数よりも 1 つ多くなります。再同期化クロック、フィードバック再同期化クロック、またはポストアンブル・クロックの 1 つを別のインタフェースと共有することによって達成できます。ただし、ボード・レイアウトは制約されることになります。トップ DDR インタフェースに対するすべての再同期化クロック、フィードバック再同期化クロック、およびポストアンブル・クロックはデュアル・リージョンでなければなりません。
2 個の enhanced PLL — 1 つがトップで 1 つがボトム	システム PLL が 2 DDR インタフェースと共にトップに配置された場合、ボトム・インタフェースに対するシステム・クロック、ライト・クロック、および再同期化クロックは、グローバル・ネットワークにドライブしなければならない、トップで 1 つのグローバル・ネットワークがフリーになります。フィードバック PLL には、トップの DDR インタフェースに到達するには、4 つすべてのボトム・グローバル・ネットワークが必要です。システム PLL がシングル DDR インタフェースと共にボトムに配置された場合、システム・クロック、ライト・クロック、および 2 つの再同期化クロックは、チップのボトムからグローバルにしなければなりません。トップのフィードバック PLL はトップから 2 つの信号をグローバルにドライブすることのみ必要になります。したがって、2 DDR インタフェースとチップの同じサイドにフィードバック PLL を配置するのが望ましい構成です。
2 個の enhanced PLL と 1 個の fast PLL	フィードバック PLL の目的は、基本的に I/O バッファ入力および出力遅延の変動を補正するためのものです。フィードバック PLL に fast PLL を使用する場合は、トップやボトムではなくチップのサイドの I/O を使用します。これらのピンは特性が異なるため変動を追跡することはできません。fast PLL には 4 つの出力タップしかないので、3 DDR インタフェースをサポートするにはそれらのうち 2 つ以上が必要です。もう 1 つの問題は、fast PLL にはチップの反対側に信号をドライブする（すなわち、左から右または右から左に）ためのグローバル・ネットワークが必要なので、トップまたはボトムにのみ入る必要がある信号をドライブするには、グローバル信号を使用しなければならない可能性があります。メモリ・インタフェース数によっては、fast PLL をシステム PLL として使用することも可能です。

表 3 では、システム・クロック (clk)、ライト・クロック (wr\_clk)、および再同期化クロック (resync\_clk\_3) をグローバルにして Gclk で表記しています。これはシステム・クロックとライト・クロックでチップの両サイドにある 3 つのコントローラをすべてドライブする必要があり、また信号がチップのトップからチップのボトムまで縦断してボトムの右コントローラの再同期化レジスタをドライブしなければならないため、再同期化クロック (resync\_clk\_3) はグローバルとなっています。

信号	enhanced PLL #	リソース・タイプ	注記
clk	11	グローバル	このクロックは 3 つのエリアに配置された 3 つのコントローラをドライブします。
write_clk	11	グローバル	このクロックは 3 つのエリアに配置された 3 つのコントローラをドライブします。
dedicated_resynch_or_capture_clk_bottom_right	11	グローバル	この信号はエリアを横断しなければなりません。
mem_clk_dedicated	11	なし	専用クロック出力ピンをドライブするためにのみ必要です。必須ではありません。
dedicated_resynch_or_capture_clk_top_right	11	リージョナル	—
dedicated_resynch_or_capture_clk_top_left	11	リージョナル	—
feedback_resynch_clk_top_left	5	リージョナル	—
dedicated_postamble_clk_top_left	5	リージョナル	—
dedicated_postamble_clk_top_right	5	リージョナル	—
feedback_resynch_clk_top_right	5	リージョナル	—
feedback_resynch_clk_bottom_right	6	リージョナル	—
dedicated_postamble_clk_bottom_right	6	リージョナル	—


表 3 の注:

- (1) リージョナル・クロックおよびグローバル・クロックに対するこれらの設定は、各コントローラが 32 ビット以下の幅の DQ インタフェースを備えているときに動作します。DQ インタフェースが 32 ビットまたは 40 ビットを超える場合は、チップのトップまたはボトム・サイド全体に及びます。この場合、デュアル・リージョナル・クロックを使用しなければなりません。これはインタフェースが 2 つのエリアをまたがるためです。PLL 配線リソースはデザインごとに異なり、デザイン例で 1 つのケースを示します。

## DDR/DDR2 SDRAM メモリ・ コントローラの インスタンス化

この項では、このデザイン用のコントローラの生成方法を説明します。最小の繰り返しで適切なタイミングを達成するのに必要な設定を説明しているため、徹底的に検討することが重要です。

Quartus II プロジェクトを作成した後、MegaWizard® Plug-In Manager を使用して DDR/DDR2 SDRAM コントローラを生成します。

 同じプロジェクトで新しいコントローラを生成するたびに、MegaWizard Plug-In Manager を起動する必要があります。




MegaWizard Plug-In Manager を使用した SDRAM コントローラのパラメータ化および生成について詳しくは、「[DDR および DDR2 SDRAM コントローラ・コンパイラ・ユーザガイド](#)」を参照してください。

MegaWizard Plug-In Manager を使用してコアを生成する場合、以下の 4 ページが重要です。

- MegaWizard Plug-In Manager、ページ 2a
- IP Toolbench - ステップ 1: Parameterize - Controller ページ
- IP Toolbench - ステップ 1: Parameterize - Project Settings ページ
- IP Toolbench - ステップ 1: Parameterize - Manual Timings ページ

### MegaWizard Plug-In Manager ページ 2a

別個のディレクトリにコントローラを生成。例えば、デザインに 1 つはトップ、もう 1 つはボトムに 2 つのコントローラがある場合は、`<quartus_project_dir>/top_dds.v` および `<quartus_project_dir>/bottom_dds.v` に DDR2 コントローラ・メガファンクションを生成します。これは多くのファイルをすべて 1 つのディレクトリに持たせないようにするのに役立ちます。

 2 つのコントローラが別々のディレクトリに生成されると、PLL は両方のコントローラとも同じ名前 (`dds_pll_stratixii` および `dds_fb_pll_stratixii`) でインスタンス化されます。PLL インスタンスの名前を変更して、デザイン内で 2 つのコントローラに共通した名前にならないようにします。また、何らかの理由でコントローラを再生成すると、PLL ファイル (`dds_pll_stratixii.v` および `dds_fb_pll_stratixii.v`) は上書きされるので、名称変更プロセスを繰り返す必要があります。



コントローラを生成すると、MegaWizardPlug-In Manager はシステム PLL とフィードバック PLL の 2 個の PLL を生成します。これらの PLL は、ディレクトリ **top\_right**、**top\_left** および **bottom\_right** 内のファイル **multiple\_ddr2\_top.v** でインスタンス化されます。3 コントローラ・デザインの場合、これらの PLL はトップレベル・デザインでは使用されず、デザインの要求に合わせ、MegaWizard Plug-In Manager を使用して 3 つの独立した PLL が生成されます。これらの PLL はディレクトリ **pll\_1\_top**、**pll\_2\_top**、および **pll\_bottom** 内に生成されます。3 コントローラ・デザインのトップレベルのデザイン・ファイル (**multiple\_ddr2\_top.v**) を参照して、PLL インスタンスをそれらの出力接続と併せて確認してください。

## Controller ページ

- a. **DQS mode** をオンにします。
- b. **Use feedback clock** をオンにします。



このオプションがイネーブルされると、読み出し中に DQ ピンからのデータをキャプチャするレジスタは DOS を遅延した信号でクロックされます。そうでない場合、PLL で生成されたクロックがデータをキャプチャします (Stratix シリーズのみ)。DQS モードは非 DQS モードよりも高い性能を提供します。非 DQS モードでは、DQ ピンをサイド・バンクに配置できるため、より柔軟にピンを配置できます。トップおよびボトム・バンクだけが、DQS 信号を遅延させてリード・データをキャプチャするのに必要な回路をサポートします。



DQS および非 DQS モードについて詳しくは、「[AN 328: Stratix II デバイスによる DDR2 SDRAM インタフェース](#)」を参照してください。

- c. **Insert pipeline registers on address and command outputs** をオンにします。



このレジスタは 200 MHz の周波数での所要性能を達成するのに役立ちます。このオプションがイネーブルされているとき、ウィザードはメモリ・コントローラとコマンドおよびアドレス出力間にパイプライン・レジスタ・ステージを挿入します。このオプションをオンにすると、ローカル・インタフェースで **local\_read\_req** または **local\_write\_req** 信号がアサートされてからメモリ・インタフェースにアドレス / コマンドが現れるまでの間に、余分なレイテンシ・サイクルが追加されます。

- d. **Clock address/command output registers on the negative edge** をオンにします。

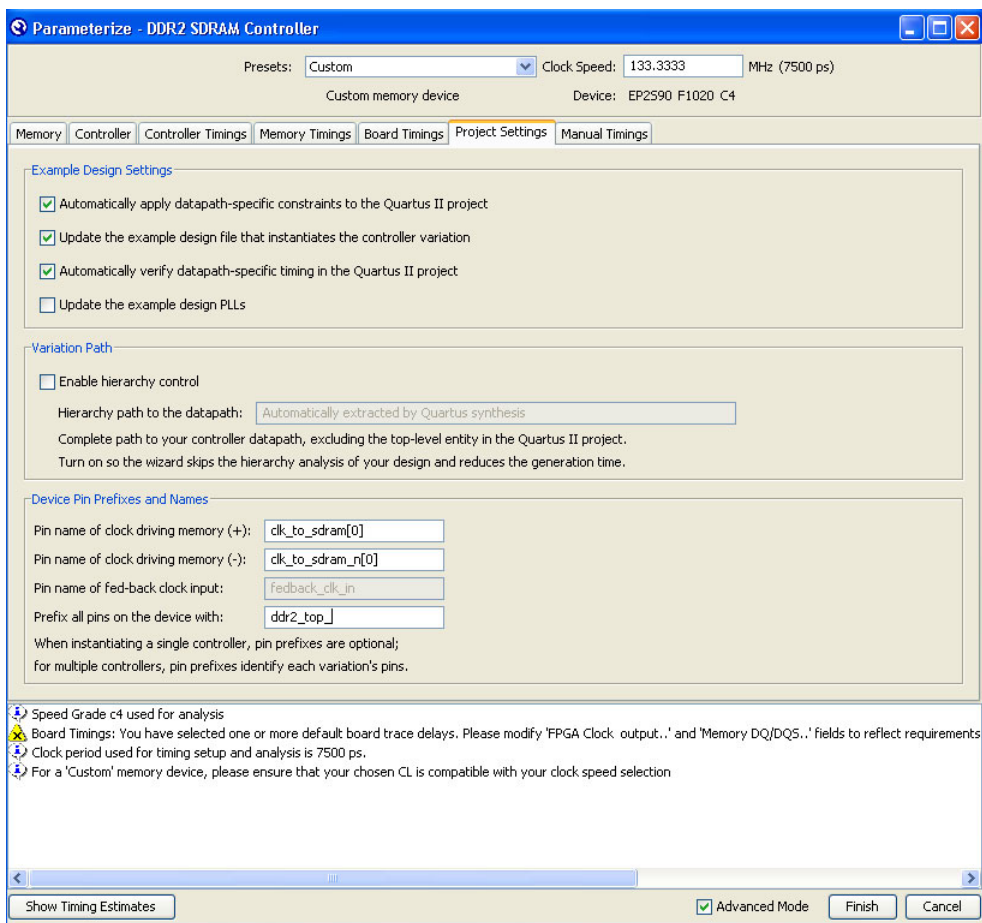


このオプションにより、アドレスおよびコマンド信号の出力タイミングを調整して、メモリ・デバイスのセットアップおよびホールド要件を満たすことができます。ただし、アドレス / コマンド・タイミングの独自のタイミング解析を実行しなければなりません。一般に、このオプションは 200 MHz 以上で動作する Stratix II デザインを除いて、オンにします。

## Project Settings ページ

対応するコントローラを生成している間に、デバイスのすべてのピンの先頭に適切な名前を付加します。これを行うことにより、図 5 に示すとおり、IP Toolbench はコントローラのトップレベルのピン名の前に、与えられた名前を付加します。

図 5. Project Settings ページ



2 コントローラ・デザインの場合：

- a. ddr2\_bottom\_
- b. ddr2\_top\_

PLL 位相を後で変更したり、PLL MegaWizard Plug-In Manager を使用して編集しなければならない場合は、**Update the example design PLLs** をオフにします。

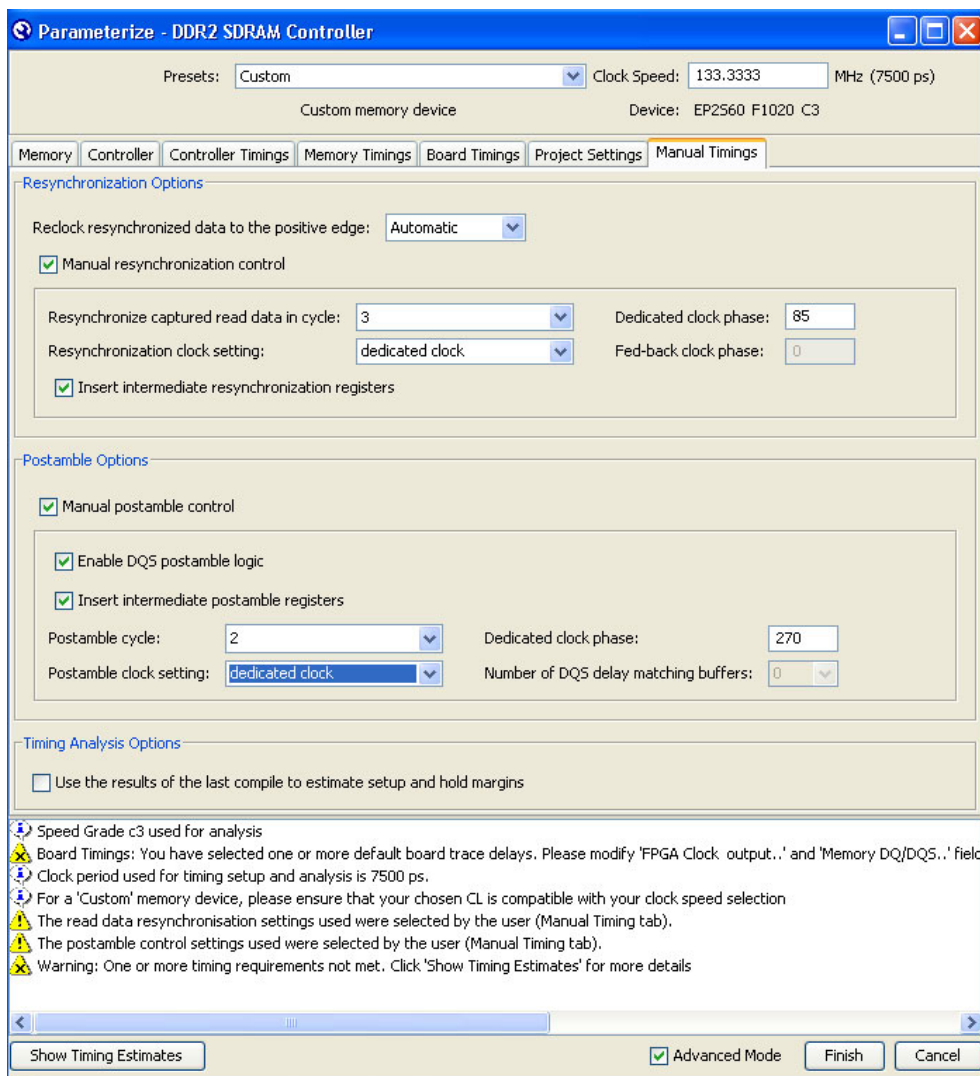


**Update the example design PLLs** オプションをオンにした場合、MegaWizard Plug-In Manager での PLL および位相の変更は、PLL 設定に自動的に反映され、PLL MegaWizard Plug-In Manager で位相を編集する必要はありません。

## Manual Timings ページ

- a. **Manual resynchronization control** をオンにします。
- b. **Manual postamble control** をオンにします。
- c. **Enable DQS postamble logic** をオンにします。
- d. 図 6 に示すように、Resynchronization Options セクションと Postamble Options セクションの両方に、適切な **Dedicated clock phase** 設定を入力します。

図 6. Manual Timings ページ



最初に、クロック位相に適切な数値を入力し、後でそれらの値を DTW タイミング解析スクリプトで変更することができます。



ページ設定について詳しくは、「[DDR/DDR2 SDRAM Controller Compiler User Guide](#)」を参照してください。

## RTL コードの変更

2 つまたは 3 つのコントローラを生成した後、これらのコントローラを統合してトップレベルのファイルを作成しなければなりません。図 1 および図 2 に、2 または 3 コントローラ・デザインのトップレベルのデザインを示します。この技術資料と共に参考のために、2 および 3 コントローラ・デザイン両方のトップレベル・ファイルが提供されていますが、設計者がこれらを複製してカスタマイズする必要があります。

トップレベル・ファイル (`multiple_ddr2_top.v`) に行われた主な変更は、以下のとおりです。

1. 3 つすべてのコントローラからの pnf 出力には、独立した信号として、以下の名前が付けられています。
  - a. pnf\_top\_right
  - b. pnf\_top\_left
  - c. pnf\_bottom\_right
2. pnf\_per\_byte[0:7] は、信号 pnf\_per\_byte\_top\_right | pnf\_per\_byte\_top\_left | pnf\_per\_byte\_bottom\_right の OR です。
3. test\_complete は、信号 test\_complete = test\_complete\_top\_right | test\_complete\_top\_left | test\_complete\_bottom\_right の OR です。
4. PLL および DLL 接続を図 1 および図 2 に示します。

## Quartus II のコンパイル

最大性能を達成するために、Quartus II ソフトウェアの表 4 に示す設定を使用しなければなりません。

表 4. Quartus II で最大性能を達成するための主な設定 (1 / 2)	
ページ名	設定
Analysis and Synthesis 設定	Optimization Technique セクションで、 <b>Speed</b> を選択します。
Fitter の設定	タイミング・ドリブн・コンパイル・セクションでは、 <b>Optimize hold timing</b> オプションをオフにします。
	<b>Optimize fast-corner timing</b> オプションをオフにします。
	Fitter Effort セクションで、 <b>Standard Fit (highest effort)</b> を選択します。

表 4. Quartus II で最大性能を達成するための主な設定 (2 / 2)

ページ名	設定
Timing Analysis 設定	Timing analysis processing セクションで、 <b>Use TimeQuest Timing Analyzer during compilation</b> または <b>Use Classic Timing Analyzer during compilation</b> を選択します。(1)

表 4 の注:

(1) アルテラでは、TimeQuest タイミング・アナライザの使用を推奨しています。



詳しくは、「[DDR Timing Wizard User Guide](#)」のデザイン・フロー・セクションのステップ 4 「Add other assignments for the design」を参照してください。

## I/O 規格の アサインメント

制約ファイル `auto_add_constraints.tcl` を提供して、I/O 規格をデザインのトップレベル・ピンに割り当てます。このファイルは、複数のコントローラを生成している間に自動的に生成され、デザインで使用されるコントローラの `.tcl` ファイルへのリファレンスが含まれます。3 コントローラ例の場合、`auto_add_constraints.tcl` ファイルは以下のファイルを参照します。

`top_right/add_constraints_for_ddr2_top_right.tcl`

`bottom_right/add_constraints_for_ddr2_bottom_right.tcl`

`top_left/add_constraints_for_ddr2_top_left.tcl`



デザイン例のコンパイルについて詳しくは、「[DDR and DDR2 SDRAM Controller User Guide](#)」を参照してください。

## I/O 位置の アサインメント

アドレス、データ、および `dqs` 信号などのトップレベルのピンを最も近い I/O バンクに割り当てます。例えば、`top_right` コントローラのトップレベルの信号を I/O バンク 3 に、`top_left` コントローラ信号を I/O バンク 4 に、そして `bottom_right` コントローラ信号を I/O バンク 7 に割り当てます。

`clock_source` を A16 に、`ddr2_top_right_feedback_clk_in` を A17 に、`ddr2_bottom_right_feedback_clk_in` を AM16 に割り当てます。



これらのピンを選択したのは、ドライブしている PLL と同じデバイスのサイドにある専用クロック入力ピンであるためです。

以下のガイドラインに従って、clk\_to\_sdram および sdram\_n ピンをすべて割り当てます。

1. メモリ・インタフェースと同じバンクにある専用クロック・ピン・ペア。例を以下に記します。
  - a. Bank3 = CLK14p/n および CLK15p/n
  - b. Bank4 = CLK12p/n および CLK13p/n
  - c. Bank7 = CLK6p/n および CLK7p/n
  - d. Bank8 = CLK4p/n および CLK5p/n
2. メモリ・インタフェースと同じデバイス・サイドの専用クロック・ピン・ペア。例を以下に記します。
  - a. トップ・サイド = CLK12/13/14/15p および n
  - b. PLL5\_OUT[2..0]p および n
  - c. PLL11\_OUT[2..0]p および n
  - d. ボトム・サイド = CLK4/5/6/7p および n
  - e. PLL6\_OUT[2..0]p および n
  - f. PLL12\_OUT[2..0]p および n

まず同じバンク内で、次にコントローラと同じサイドで、適切な差動ピン・ペアを探します。

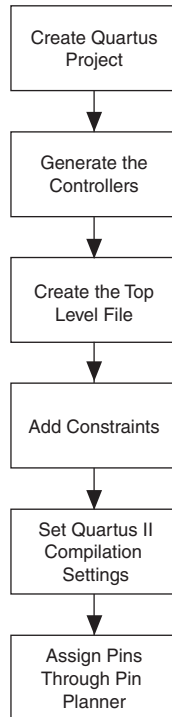
ピン reset\_n、test\_complete、pnf、および pnf\_per\_byte[7:0] の信号がクロージャ問題を発生しないように、これらのピンをサイド・バンク 5 および 6 に配置します。



タイミング違反が発生する可能性があるため、これらのピンは対応するコントローラから遠すぎない位置に配置してください。

図 7 に、複数のメモリ・インタフェース・コントローラを実装するために実施したステップを要約するチャートを示します。

図 7. 複数のメモリ・インタフェース・コントローラを実装するためのステップ



次のステップは DTW (DDR Timing Wizard) の実行です。



Quartus MegaWizard Plug-In Manager で生成されるタイミング解析スクリプト (`auto_verify_ddr_timing.tcl`) は、リージョナル・クロックが使用されているときには、複数のコントローラでは動作しません。Quartus II ソフトウェアは、以下のエラー・メッセージを生成します。

```
Error: Evaluation of Tcl script auto_verify_ddr_timing.tcl unsuccessful.
```

```
Error: Quartus II Shell was unsuccessful. 2 errors, 8 warnings
```

```
Error: Quartus II Full Compilation was unsuccessful. 2 errors, 774 warnings
```



このプロジェクトでは、多くの利点があるため DTW を使用してください。DTW について詳しくは、「[DDR Timing Wizard User Guide](#)」を参照してください。

## DTW スクリプトの 実行

DTW スクリプトを実行するには、Tools メニューの **Tcl Scripts** をクリックします。**dtw** を選択します。別々の名前を持つ各コントローラに対して、独立した \*.dwz ファイルを生成します。例えば、3 コントローラ・デザインに対しては、以下の 3 つのファイルが生成されます。

- **top\_right\_dds\_settings.dwz**
- **top\_left\_dds\_settings.dwz**
- **bottom\_right\_dds\_settings.dwz**



DTW タイミング解析スクリプトの使用法について詳しくは、「[DDR Timing Wizard User Guide](#)」を参照してください。

## デザインの コンパイル

DTW スクリプトはデザイン制約を SDC ファイルの形式で使用しており、この制約ファイルはデザインをコンパイルする前にデザイン・ファイルに追加する必要があります。



DTW タイミング解析スクリプトの使用法について詳しくは、「[DDR Timing Wizard User Guide](#)」を参照してください。

## DTW タイミング 解析

これでデザインはすべての制約を設定してコンパイルされました。タイミングが満たされたかどうかを確認する必要があります。



DTW の使用およびタイミング解析の実行方法について詳しくは、「[DDR Timing Wizard User Guide](#)」を参照してください。

コマンド・プロンプトで以下のステップを実行します。スクリプト **dtw\_timing\_analysis.tcl** ファイルは、Quartus II プロジェクト・ディレクトリにあります。

```
quartus_sh -t dtw_timing_analysis.tcl -dwz_file
top_right_dds_settings.dwz
```

```
quartus_sh -t dtw_timing_analysis.tcl -dwz_file
top_left_dds_settings.dwz
```

```
quartus_sh -t dtw_timing_analysis.tcl -dwz_file
bottom_right_dds_settings.dwz
```

スクリプトを実行した後、コンパイル・レポートを閉じてから再び開いて、タイミング解析結果を検証します。

DTW スクリプトは、図 8 に示すとおり、タイミングが満たされなかった場合に、次に何を行うかについての詳細を提供します。

図 8. DTW スクリプトのタイミングの詳細

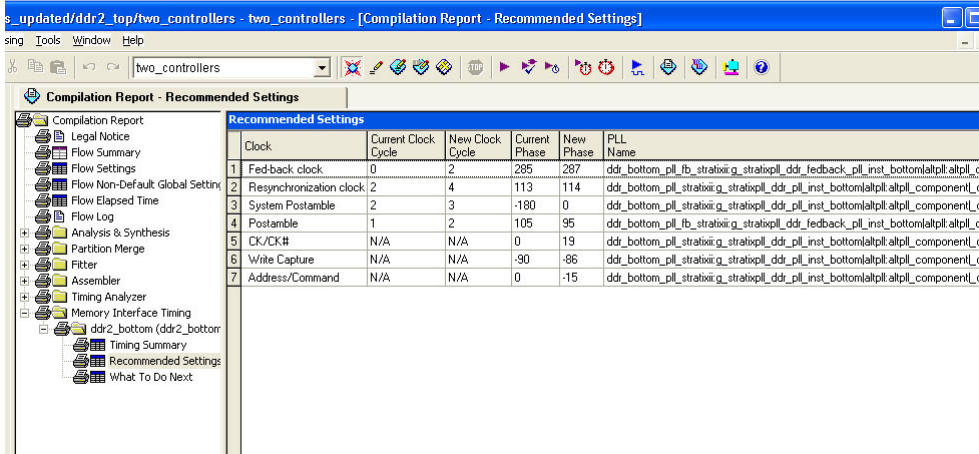
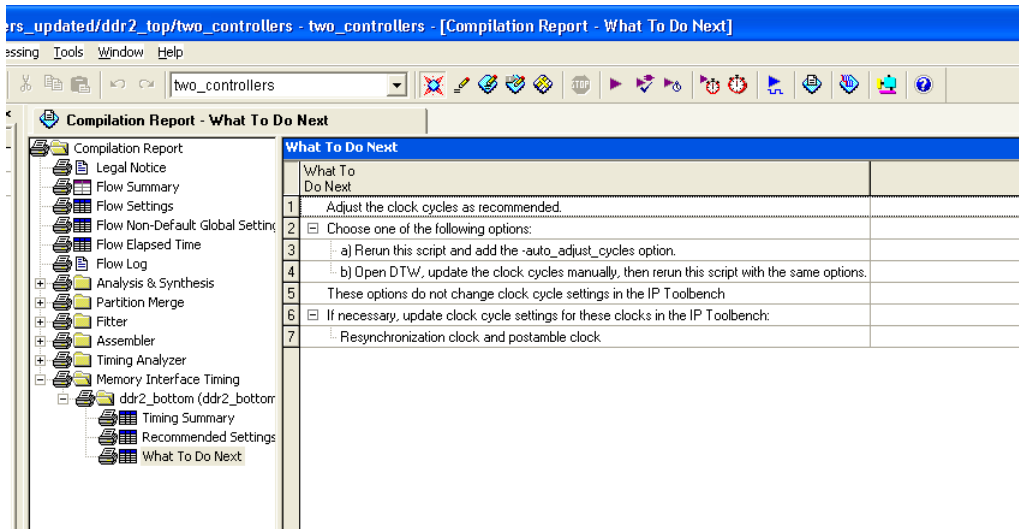


図 9 に示すとおり、一般にサイクル調整が必要です。

図 9. サイクル調整



サイクル調整を行うには、以下のステップを実行します。

```
quartus_sh -t dtw_timing_analysis.tcl -dwz_file
top_right_ddr_settings.dwz -extract_tcos ignore
-read_side tq -auto_adjust_cycles
```

このステップはサイクルを調整する必要がある場合に、対応するコントローラに対して行われます。DTW スクリプトの実行後、Quartus II ソフトウェアに次に何を行うかが示されます。

サイクル調整後、次のステップは3つすべてのクロックの位相を調整することです。これは IP Toolbench で、または対応する PLL の設定を編集して行うことができます。

一度、位相調整を行えば、次のステップは新しい PLL 位相でデザインをコンパイルすることです。

```
quartus_sh -t dtw_timing_analysis.tcl -dwz_file
top_right_ddr_settings.dwz -extract_tcos no -read_side
tq -after_iptb import_and_compile
```

PLL 位相調整が行われたすべてのコントローラに対して、前のステップを繰り返します。3 つのコントローラすべてに位相調整が行われた場合は、以下の例に示すように、コンパイルは一度だけ実行でき、タイミング解析は3つのコントローラに対して実行できます。

```
quartus_sh -t dtw_timing_analysis.tcl -dwz_file
top_right_ddr_settings.dwz -extract_tcos no -read_side
tq -after_iptb import_and_compile
```

```
quartus_sh -t dtw_timing_analysis.tcl -dwz_file
top_left_ddr_settings.dwz -extract_tcos no
```

```
quartus_sh -t dtw_timing_analysis.tcl -dwz_file
bottom_right_ddr_settings.dwz -extract_tcos no
```

この場合、デザインを3回コンパイルしなくても済みます。

## トラブルシューティング

DTWスクリプトで記述されるすべてのステップを実行します。デスティネーション・レジスタに関連する違反がある場合は、「**デスティネーション・レジスタに関連する違反**」に示すとおり、`inter_rdata[*]` という名前が表示されます。

---

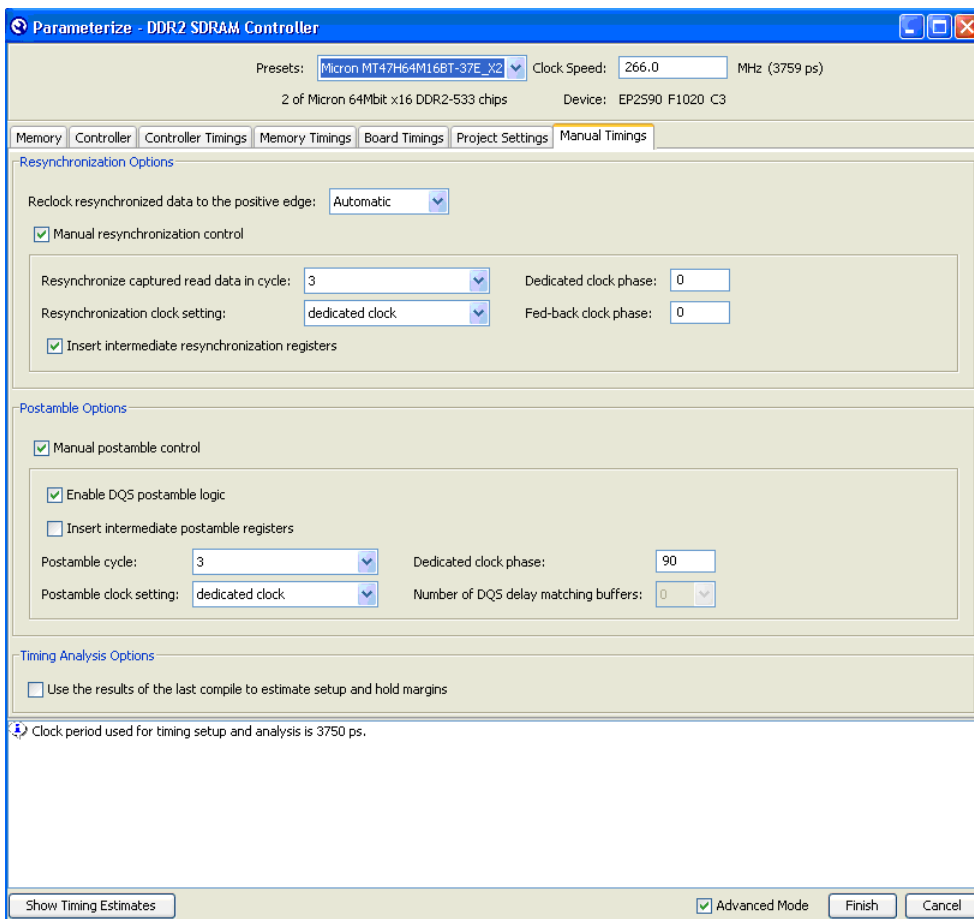
### 例 1-1. デスティネーション・レジスタに関連する違反

```
ddr2_bottom_right:ddr2_bottom_right_ddr_sdram|ddr2_bottom_right_auk_ddr_sdram:ddr2_bottom_right_auk_ddr_sdram_inst|ddr2_bottom_right_auk_ddr_datapath:ddr_io|ddr2_bottom_right_auk_ddr_dqs_group:\g_datapath:1:g_ddr_io|inter_rdata[5]
```

---

`inter_rdata[*]` 違反を受け取った場合は、Manual Page 設定の下にある **Insert intermediate synchronization registers** オプションを切り替えます。この設定がオンの場合はオフにし、オフの場合はオンにします。図 10 に、このオプションがオンになっているところを示します。

図 10. チェックなしでの中間再同期化レジスタの挿入



## まとめ

このアプリケーション・ノートに示すように、DDR/DDR2 SDRAM コントローラを使用して複数のコントローラ・デザインを作成する方法が複数あります。このアプリケーション・ノートでは、2 種類の複数コントローラ・デザインの性能要件を作成および適合させるのに必要なステップについて説明しています。ユーザ・デザインがこのアプリケーション・ノートで説明した内容と異なる場合は、ガイドラインに従い、必要に応じて適切な変更を行わなければなりません。複数メモリ・インタフェースの要件と制限を事前に知っておくと、より良いシステムを構築できます。

## 改訂履歴

表 5 に、本資料の改訂履歴を示します。

表 5. 改訂履歴		
日付 & ドキュメント・バージョン	変更内容	概要
2007 年 7 月 v2.0	「はじめに」の項を更新。「プロジェクトの設定」、「DDR SDRAM コントローラ MegaCore ファンクションの起動」、「トップレベル・デザイン・ファイルの編集」、「制約の追加」、「付録」の項（および関連する項）を削除。削除した項を「デザインの詳細」、「デザインの制約」、「DLL リソースの検討事項」、「PLL リソースの検討事項」、「DDR/DDR2 SDRAM メモリ・コントローラのインスタンス化」、「RTL コードの変更」、「Quartus II のコンパイル」、「I/O 規格のアサインメント」、「I/O 位置のアサインメント」、「DTW スクリプトの実行」、「デザインのコンパイル」、「DTW タイミング解析」、「トラブルシューティング」、「まとめ」に置き換え。すべての図を新しい図に置き換え。	—
2005 年 8 月 v1.0	初版	—



101 Innovation Drive  
San Jose, CA 95134  
[www.altera.com](http://www.altera.com)  
Technical Support:  
[www.altera.com/support/](http://www.altera.com/support/)  
Literature Services:  
[literature@altera.com](mailto:literature@altera.com)

Copyright © 2007 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.



L.S. EN ISO 9001