

この資料は英語版を翻訳したもので、内容に相違が生じる場合には原文を優先します。こちらの日本語版は参考用としてご利用ください。設計の際には、最新の英語版で内容をご確認ください。

NI151005-8.0.0

コアの概要

Avalon[®] インタフェース対応の SDRAM コントローラは、オフチップ SDRAM への Avalon Memory-Mapped (Avalon-MM) インタフェースを提供します。SDRAM コントローラにより、SDRAM チップに容易に接続できるカスタム・システムをアルテラ FPGA に構築することができます。SDRAM コントローラは、PC100 規格で規定される標準 SDRAM をサポートしています。

SDRAM は大規模な不揮発性メモリを必要とするコスト重視のアプリケーションに広く使用されています。SDRAM は比較的安価ですが、リフレッシュ動作、オープン・ロウ管理、その他の遅延およびコマンド・シーケンスを実行するコントロール・ロジックが必要です。SDRAM コントローラは、1 つまたは複数の SDRAM チップに接続され、すべての SDRAM プロトコル要件に対応します。FPGA の内部では、コアはリニア・メモリ（つまりフラット・アドレス空間）として扱うことができる Avalon-MM スレーブ・ポートを Avalon-MM マスタ・ペリフェラルに提供します。

コアは各種データ幅（8、16、32、または 64 ビット）、各種メモリ・サイズ、および複数のチップ・セレクトで SDRAM サブシステムにアクセスできます。Avalon-MM インタフェースはレイテンシを認識できるため、リード転送のパイプライン化が可能です。コアはオプションによりアドレス・バスとデータ・バスを他のオフチップ Avalon-MM トライステート・デバイスと共有できます。この機能は I/O ピンが制限されたシステムで、SDRAM の他に複数のメモリ・チップに接続する必要がある場合に有用です。

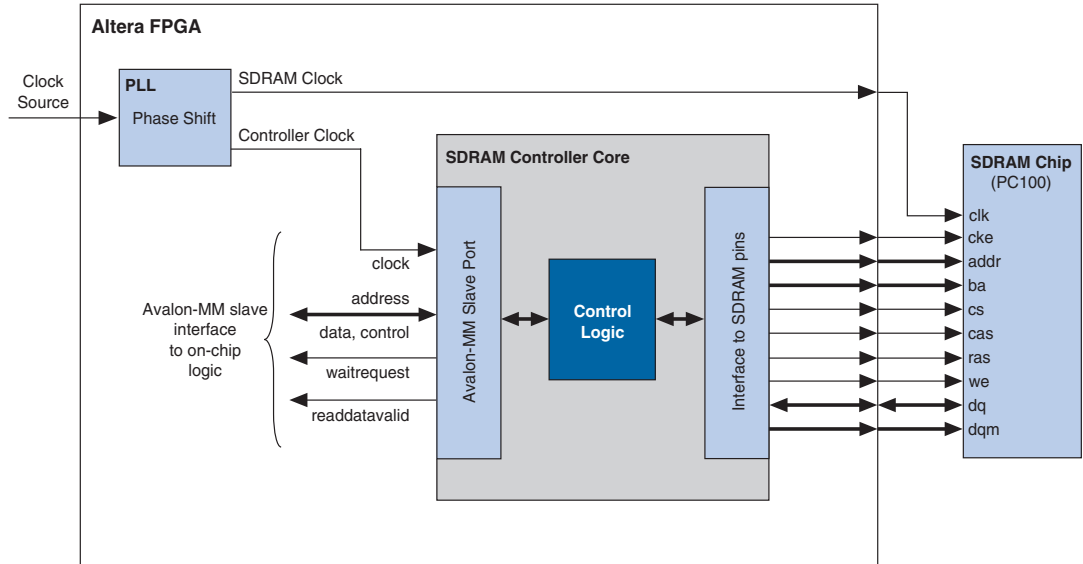
Avalon インタフェース対応の SDRAM コントローラ・コアは SOPC Builder に対応しており、SOPC Builder で生成されたどのシステムにも容易に統合できます。この章は、以下の項で構成されています。

- 1-2 ページの「機能の説明」
- 1-6 ページの「デバイスおよびツールのサポート」
- 1-7 ページの「SOPC Builder でのコアのインスタンス化」
- 1-10 ページの「ハードウェア・シミュレーションの考慮事項」
- 1-14 ページの「ソフトウェア・プログラミング・モデル」
- 1-14 ページの「クロック、PLL、およびタイミングの検討事項」

機能の説明

図 1-1 に、外部 SDRAM チップに接続されたコントローラ・コアのブロック図を示します。

図 1-1. Avalon インタフェース対応 SDRAM コントローラのブロック図



以下の項では、SDRAM コントローラ・コアのコンポーネントについて詳細に説明します。すべてのオプションは、システム生成時に指定され、実行時には変更できません。

Avalon-MM インタフェース

Avalon-MM スレーブ・ポートは、SDRAM コントローラ・コアでユーザーが認識できる部分です。スレーブ・ポートは、SDRAM チップと同サイズの連続したフラットなメモリ空間を提供します。スレーブ・ポートのアクセス時には、PC100 SDRAM プロトコルの詳細は完全にトランスペアレントになります。Avalon-MM インタフェースは、シンプルなメモリ・インタフェースとして動作します。メモリ・マップド・コンフィギュレーション・レジスタは存在しません。

Avalon-MM スレーブ・ポートは、リード転送およびライト転送用にペリフェラルで制御されるウェイト・ステートをサポートしています。スレーブ・ポートは、有効なデータの供給が可能になるまでの間、転送を停止します。また、スレーブ・ポートは可変レイテンシを使用したリード転送もサポートしているため、パイプライン化された広帯域幅のリー

ド転送が可能です。マスタ・ペリフェラルがスレーブ・ポートからシーケンシャル・アドレスを読み出すと、最初のデータはレイテンシの初期期間後に返されます。以降の読み出しでは、クロック・サイクルごとに新しいデータを生成できます。ただし、SDRAM コントローラは周期的に休止して SDRAM をリフレッシュする必要があるため、データは必ずクロック・サイクルごとに返されるとは限りません。



Avalon-MM 転送タイプについて詳しくは、「[Avalon Interface Specifications](#)」を参照してください。

オフチップ SDRAM インタフェース

外部 SDRAM チップへのインタフェースは、PC100 規格で定義される信号を供給します。これらの信号は、アルテラ FPGA の I/O ピンを介して外部から SDRAM チップに接続する必要があります。

信号のタイミングおよび電気的特性

信号のタイミングとシーケンスは、コアのコンフィギュレーションによって決まります。ハードウェア設計者は、システムに選択した SDRAM チップに適合するようにコアをコンフィギュレーションします。詳細は、[1-7 ページの「SOPC Builder でのコアのインスタンス化」](#)を参照してください。FPGA ピンの電気的特性は、ターゲット・デバイス・ファミリと Quartus® II ソフトウェアで実行した割り当ての両方に依存します。一部の FPGA ファミリは、より広範囲の電気規格をサポートしているため、多様な SDRAM チップと通信が可能です。詳細は、ターゲット FPGA ファミリのハンドブックを参照してください。

クロックとデータ信号の同期化

SDRAM チップ（以下、“SDRAM クロック”）は、SDRAM コントローラ上の Avalon-MM インタフェース用クロック（以下、“コントローラ・クロック”）と同じ周波数でドライブする必要があります。すべての同期デザインと同様に、SDRAM ピンのアドレス、データ、およびコントロール信号は、クロック・エッジが到着したときに安定していなければなりません。[図 1-1](#) に示すとおり、オンチップ PLL（Phase-Locked Loop）を使用して、SDRAM コントローラ・コアと SDRAM チップ間のクロック・スキューを縮小することができます。低速クロックでは、PLL が不要な場合があります。高速クロックでは、ピンの信号が安定しているときにのみ SDRAM クロックがトグルするよう PLL が必要になります。PLL ブロックは、SDRAM コントローラ・コアの一部ではありません。PLL が必要な場合は、手動でインスタンス化する必要があります。SOPC

Builder コンポーネントである PLL コア・インタフェースをインスタンス化するか、または SOPC Builder システム・モジュールの外部で ALTPLL メガファンクションをインスタンス化することができます。

PLL を使用する場合は、PLL をチューニングしてクロックの位相シフトを導入し、同期信号が安定してから SDRAM クロック・エッジが到着するようにしなければなりません。詳細は、1-14 ページの「クロック、PLL、およびタイミングの検討事項」を参照してください。



ユーザーの SOPC Builder システムで PLL をインスタンス化する方法について詳しくは、Chapter 30, PLL Core を参照してください。Nios® II 開発ツールは、PLL と共に DSRAM コントローラ・コアを使用するハードウェア・デザイン例を提供しており、これはカスタム・デザインのリファレンスとして使用できます。Nios II 開発ツールは、www.altera.co.jp から無償でダウンロードできます。

サポートされていないクロック・イネーブル (CKE)

SDRAM コントローラはクロック・ディセーブル・モードをサポートしていません。SDRAM コントローラは、SDRAM の CKE 信号を永続的にアサートします。

その他の Avalon-MM トライステート・デバイスとのピンの共有

SOPC Builder システムに Avalon-MM トライステート・ブリッジが存在する場合、SDRAM コントローラ・コアは既存のトライステート・ブリッジとピンを共有できます。この場合、コアの `addr`、`dq` (データ) および `dqm` (バイト・イネーブル) ピンは、Avalon-MM トライステート・ブリッジに接続されているその他のデバイスと共有されます。この機能により I/O ピンが節約できます。これは複数の外部メモリ・チップ (フラッシュ、SRAM、および SDRAM など) を使用しているが、SDRAM チップ専用割り当てのピンが不足しているシステムに有効です。ピンの共有が性能に与える影響については、「性能に関する考慮点」を参照してください。



SDRAM アドレスは、トライステート・ブリッジの下位アドレス・ビットがメモリ・デバイスの下位アドレス・ビットに揃うように、ワードのサイズに関係なくすべてのアドレス・ビットを接続する必要があります。最小アクセス・サイズが 16 ビットのとときにメモリの A0 を、あるいは最小アクセス・サイズが 32 ビットのとときに、A0-A1 を非接続にすることはできません。

ボード・レイアウトおよびピン配置の考慮事項

ボード・レイアウトと FPGA ピン配置に関して決定するときは、SDRAM 信号間のスキューが最小になるよう試みてください。例えば、FPGA ピン配置を割り当てるときは、SDRAM クロック出力を含めて SDRAM 信号を物理的に互いに接近させてグループ化する必要があります。また、Quartus II ソフトウェアでは、**Fast Input Register** および **Fast Output Register** ロジック・オプションも使用できます。これらのロジック・オプションは、SDRAM 信号用レジスタを I/O セルに配置します。I/O セルのレジスタからドライブされる信号は、 t_{CO} 、 t_{SU} 、および t_H など、類似したタイミング特性を備えています。

性能に関する考慮点

最適な状態では、SDRAM コントローラ・コアの帯域幅は、1 クロック・サイクルあたり 1 ワード近くになります。ただし、SDRAM のリフレッシュに伴うオーバーヘッドのために、1 クロック・サイクルあたり 1 ワードを達成することは不可能です。以下で説明するように、コアの性能はその他の要因によっても影響を受けます。

オープン・ロウ管理

SDRAM チップは複数のメモリ・バンクとして編成され、各バンクは独立したオープン・ロウ・アドレス管理が可能です。SDRAM コントローラ・コアは、単一バンクに対するオープン・ロウ管理を活用しています。同じロウおよびバンク内で連続的に読み出したり書き込みを実行すると、動作レートは 1 クロックあたり 1 ワードに近づきます。異なるデステーション・バンクに頻繁にアクセスするアプリケーションでは、ロウ・クローズおよびロウ・オープンのために別の管理サイクルが必要です。

データ・ピンとアドレス・ピンの共有

コントローラが他のトライステート・デバイスとピンを共有する場合は、一般に平均アクセス時間は長くなり、帯域幅は減少します。他のデバイスにトライステート・ブリッジへのアクセスが許可されると、SDRAM はロウ・オープンおよびロウ・クローズのオーバーヘッド・サイクルを必要とします。さらに、SDRAM コントローラは、再びアクセスが許可される前に、数クロック・サイクル待機する必要があります。

帯域幅を最大にするために、SDRAM コントローラは同一ロウおよびバンク内で連続リード処理またはライト処理が続く限り、自動的にトライステート・ブリッジの制御を維持します。



この動作によって、Avalon-MM トライステート・ブリッジを共有する他のデバイスの平均アクセス時間が長くなる可能性があることに注意してください。

SDRAM コントローラは、連続処理にブレイクがある場合、またはリフレッシュ処理が要求される場合は、必ずオープン・ロウをクローズします。その結果は、以下ようになります。

- コントローラはトライステート・ブリッジを共有する他のデバイスへのアクセスを永久にブロックすることはできない。
- コントローラが SDRAM のロウ・オープン時間制限に違反しないことが保証される。

ハードウェア・デザインおよびターゲット FPGA

ターゲット FPGA は、ハードウェア・デザインの達成可能な最大クロック周波数に影響します。デバイス・ファミリによっては、他のデバイス・ファミリよりも高い f_{MAX} 性能を達成します。さらに、デバイス・ファミリ内でもスピード・グレードが高いデバイスがより高性能を達成します。SDRAM コントローラ・コアは、Stratix® シリーズ FPGA など、アルテラの高性能デバイス・ファミリで 100 MHz を達成できます。ただし、コアがすべてのアルテラ FPGA ファミリで、100 MHz の性能を達成できるとは限りません。

f_{MAX} 性能も SOPC Builder システム・デザインに依存します。SDRAM コントローラ・クロックは、システム・モジュール内のほかのロジックもドライブすることがあり、これが達成可能な最大周波数に影響を与える可能性があります。SDRAM コントローラ・コアが 100 MHz の f_{MAX} 性能を達成するには、同じクロックでドライブされるすべてのコンポーネントが 100 MHz のクロック・レート用に設計されており、かつ Quartus II ソフトウェアのタイミング解析で、ハードウェア・デザイン全体が 100 MHz で動作可能なことを検証する必要があります。

デバイスおよびツールのサポート

Avalon インタフェース・コア対応の SDRAM コントローラは、すべてのアルテラ FPGA ファミリをサポートしています。サポートされる I/O 規格は FPGA ファミリによって異なり、そのためコアが特定の SDRAM チップにインタフェースする機能が影響を受ける場合があります。サポートされる I/O タイプについて詳しくは、ターゲット FPGA ファミリのハンドブックを参照してください。

SOPC Builder でのコアの インスタンス化

設計者は、SOPC Builder で SDRAM コントローラ用の MegaWizard® インタフェースを使用して、ハードウェア機能とシミュレーション機能を指定します。SDRAM コントローラ MegaWizard には、**Memory Profile** と **Timing** の 2 つのタブがあります。この項では、各ページで使用可能なオプションについて説明します。

Presets リストには、いくつかの定義済み SDRAM コンフィギュレーションが提供されます。ターゲット・ボード上の SDRAM サブシステムがプリセット・コンフィギュレーションのいずれかに適合する場合、SDRAM コントローラ・コアは適切なプリセット値を選択することによって、容易にコンフィギュレーションできます。以下のプリセット・コンフィギュレーションが定義されています。

- Micron MT8LSDT1664HG モジュール
- SDR100 8 MByte × 16 チップ : 4 つ
- Micron MT48LC2M32B2-7 チップ : 1 つ
- Micron MT48LC4M32B2-7 チップ : 1 つ
- NEC D4564163-A80 チップ (64 MByte × 16) : 1 つ
- Alliance AS4LC1M16S1-10 チップ : 1 つ
- Alliance AS4LC2M8S0-10 チップ : 1 つ

プリセット・コンフィギュレーションを選択すると、**Memory Profile** および **Timing** タブの値は、特定のコンフィギュレーションに適合するよう自動的に変化します。タブのコンフィギュレーション設定を変更すると、**Preset** 値は **custom** に変更されます。

Memory Profile ページ

Memory Profile ページでは、アドレスおよびデータ・バス幅、チップ・セレクト信号、およびバンク数など、SDRAM サブシステムの構造を指定できます。表 1-1 に、**Memory Profile** ページで指定できる設定を示します。

設定		許容値	デフォルト値	説明
Data Width		8, 16, 32, 64	32	SDRAM データ・バス幅。この値は、dq バス（データ）および dqm バス（バイト・イネーブル）の幅を決定します。
Architecture Settings	Chip Selects	1, 2, 4, 8	1	SDRAM サブシステムでの独立したチップ・セレクト数。複数のチップ・セレクトを使用することによって、SDRAM コントローラは複数の SDRAM チップを 1 つのメモリ・サブシステムに結合できます。
	Banks	2, 4	4	SDRAM バンク数。この値は、SDRAM に接続する ba バス（バンク・アドレス）の幅を決定します。正しい値は、ターゲット SDRAM のデータシートに記載されています。
Address Width Settings	Row	11, 12, 13, 14	12	ロウ・アドレス・ビット数。この値は、addr バスの幅を決定します。ロウとカラムの値は、選択した SDRAM のジオメトリによって決まります。例えば、4096 (2^{12}) ロウ \times 512 カラムとして構成された SDRAM の場合、ロウの値は 12 です。
	Column	8 以上で Row より小さい値	8	カラム・アドレス・ビット数。例えば、4096 ロウ \times 512 (2^9) \times 512 カラムとして構成された SDRAM の場合、カラムの値は 9 です。
Share pins via tri-state bridge dq/dqm/addr I/O pins		チェックする (Yes)、 チェックしない (No)	No	No に設定すると、すべてのピンが SDRAM チップ専用になります。Yes に設定すると、addr、dq、および dqm ピンは、システムのトライステート・ブリッジと共有できます。この場合、プルダウン・メニューから適切なトライステート・ブリッジを選択してください。
Include a functional memory model in the system testbench		Yes, No	Yes	このオプションをオンにすると、SOPC Builder は SDRAM チップの機能シミュレーション・モデルを作成します。このデフォルトのメモリ・モデルにより、SDRAM コントローラを使用するシステムを作成および検証するプロセスが短縮されます。1-10 ページの「ハードウェア・シミュレーションの考慮事項」を参照してください。

Memory Profile ページで入力した設定に基づいて、ウィザードは SDRAM サブシステムの予想メモリ容量をメガバイト、メガビット、およびアドレス指定可能ワード数の単位で表示します。これらの予想値を選択した SDRAM の実際のサイズと比較して、設定が正しいことを検証します。

Timing ページ

Timing ページでは、使用する SDRAM チップのタイミング仕様を入力できます。正しい値は、ターゲット SDRAM のメーカーのデータシートで提供されています。表 1-2 に、**Timing** ページで指定できる設定を示します。

設定	許容値	デフォルト値	説明
CAS latency	1, 2, 3	3	リード・コマンドからデータ出力までのレイテンシ（クロック・サイクル単位）。
Initialization refresh cycles	1 - 8	2	この値は、SDRAM コントローラがリセット後の初期化シーケンスの一部として実行するリフレッシュ・サイクル数を指定します。
Issue one refresh command every	—	15.625 μ s	この値は、SDRAM コントローラが SDRAM をリフレッシュする頻度を指定します。標準的な SDRAM は、64 ms ごとに 4,096 回のリフレッシュ・コマンドを必要とし、この条件は $64 \text{ ms} / 4,096 = 15.625 \mu\text{s}$ ごとに 1 回のリフレッシュ・コマンドを発行すると満たされます。
Delay after power up, before initialization	—	100 μ s	クロックおよび電源安定化から SDRAM 初期化までの遅延。
Duration of refresh command (t _{rfc})	—	70 ns	自動リフレッシュ周期。
Duration of precharge command (t _{rp})	—	20 ns	プリチャージ・コマンドの持続時間。
ACTIVE to READ or WRITE delay (t _{rcd})	—	20 ns	ACTIVE から READ または WRITE までの遅延。
Access time (t _{ac})	—	17 ns	クロック・エッジからのアクセス時間。この値は CAS レイテンシに依存することがあります。
Write recovery time (t _{wr} , No auto precharge)	—	14 ns	明示的なプリチャージ・コマンドが発行された場合はライト・リカバリ。この SDRAM コントローラは常に明示的なプリチャージ・コマンドを発行します。

ハードウェア・シミュレーションの考慮事項

ユーザーが厳密なタイミング値を入力した場合でも、各パラメータに対して達成される実際のタイミングは、Avalon クロック周期の整数倍になります。**Issue one refresh command every** パラメータの場合、実際のタイミングは目標値を超えない最大クロック・サイクル数になります。その他のパラメータはすべて、実際のタイミング目標値と同じかそれを上回る値を提供する最小クロック・チック数になります。

この項では、SDRAM を搭載したシステムをシミュレーションする際の考慮事項について説明します。シミュレーションに必要な主要コンポーネントは、以下の3つです。

- SDRAM コントローラのシミュレーション・モデル
- メモリ・モデルとも呼ばれる SDRAM チップのシミュレーション・モデル
- メモリ・モデルを SDRAM コントローラ・ピンに配線するシミュレーション・テストベンチ

これらのコンポーネントの一部またはすべてが、システム生成時に SOPC Builder によって生成されます。

SRAM コントローラ・シミュレーション・モデル

SOPC Builder で生成された SDRAM コントローラ・デザイン・ファイルは、合成とシミュレーションの両方に適しています。一部のシミュレーション機能は、HDL コードの特定のセクションを合成ツールに対して隠蔽する“変換オン / オフ”合成ディレクティブを使用して HDL で実装されます。

シミュレーション機能は、主に ModelSim シミュレータを使用して Nios および Nios II プロセッサ・システムを簡単にシミュレーションするために実装されています。The SDRAM controller simulation model is not ModelSim specific. ただし、モデルを他のシミュレータで動作させるには、多少の変更が必要になることがあります。



シミュレーション・ディレクティブを変更して、カスタム・シミュレーション・フローを作成する場合は、SOPC Builder がシステム生成時に既存のファイルを上書きすることに注意が必要です。変更内容が上書きされないように注意してください。



Nios II エンベデッド・プロセッサ・システムにおける SDRAM コントローラのシミュレーションのデモについては、「AN 351: Nios II エンベデッド・プロセッサ・デザインのシミュレーション」を参照してください。

SDRAM メモリ・モデル

この項では、SDRAM チップのメモリ・モデルのシミュレーションの2つのオプションについて説明します。

汎用メモリ・モデルの使用

システム生成時に **Include a functional memory model the system testbench** オプションをイネーブルにすると、SOPC Builder は SDRAM メモリの HDL シミュレーション・モデルを生成します。自動生成されたシステム・テストベンチでは、SOPC Builder はこのメモリ・モデルを自動的に SDRAM コントローラ・ピンに配線します。

自動メモリ・モデルとテストベンチにより、SDRAM コントローラを使用するシステムを作成および検証するプロセスが短縮されます。ただし、このメモリ・モデルは汎用機能モデルであり、実際の SDRAM チップの真のタイミングや機能は反映していません。汎用モデルは常に、単一のモノリシック・メモリ・ブロックとして構成されます。例えば、2 つの SDRAM チップを結合するシステムでも、汎用メモリ・モデルは単一のエンティティとして実装されます。

SDRAM メーカーのメモリ・モデルの使用

Include a functional memory model the system testbench オプションがイネーブルにされていない場合は、設計者が SDRAM メーカーからメモリ・モデルを入手し、システム・テストベンチの SDRAM コントローラ・ピンに手動で配線する必要があります。

コンフィギュレーションの例

以下の例では、SDRAM コントローラの出力を1つまたは複数の SDRAM チップに接続する方法を示しています。ct1 のラベルが付いたバスは、cas_n、ras_n、cke および we_n など、残りの信号を総合したものです。

図 1-2 に、32 ビット・データを持つ単一の 128 M ビット SDRAM チップを示します。アドレス、データ、およびコントロール信号は、コントローラからチップに直接配線されます。その結果、128 M ビット（16 M バイト）のメモリ空間が実現します。

図 1-2. 32 ビット・データを持つ単一の 128M ビット SDRAM チップ

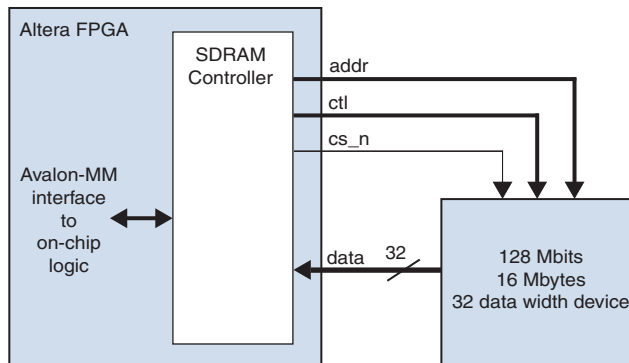


図 1-3 は、それぞれが 16 ビット・データを持つ 2 つの 64 M ビット SDRAM チップを示しています。アドレスおよびコントロール信号は、2 つのチップに並列に配線されます。チップ・セレクト (cs_n) が 2 つのチップで共有されることに注意してください。各チップは、32 ビット・データ・バスの半分を提供します。その結果、128 M ビット (16 M バイト) の 32 ビット・データ論理メモリが実現します。

図 1-3. それぞれが 16 ビット・データを持つ 2 つの 64 M ビット SDRAM チップ

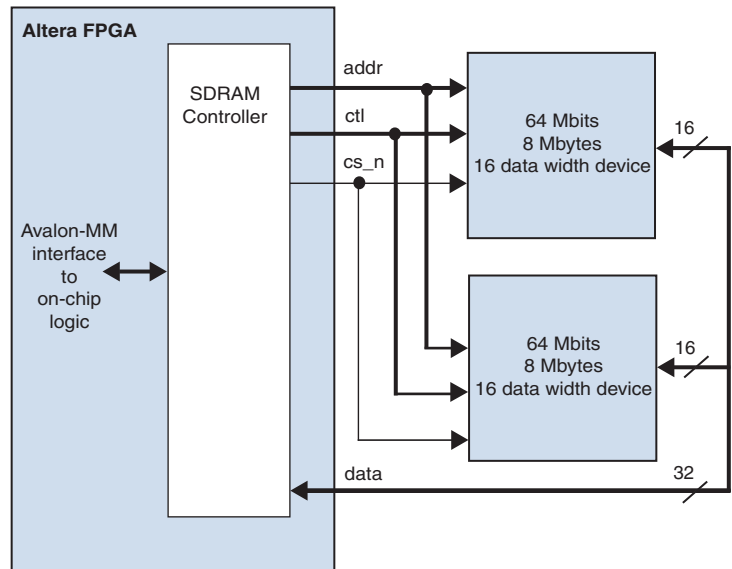
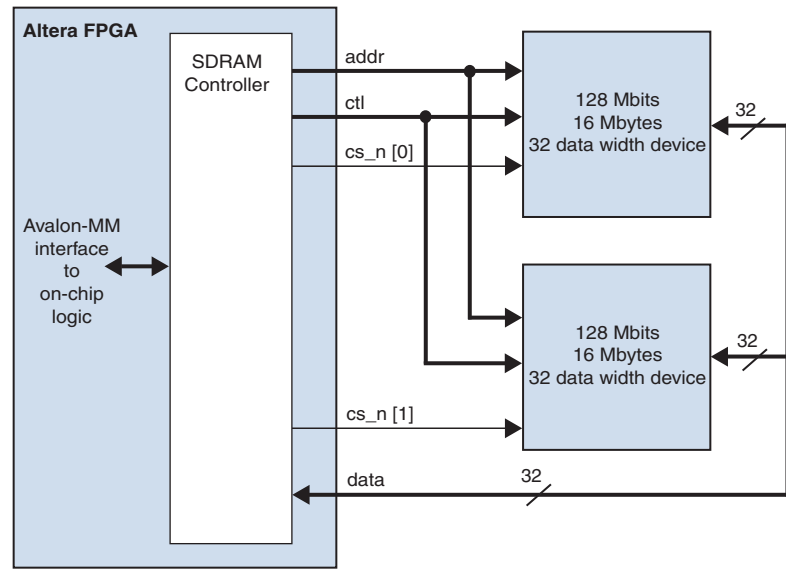


図 1-4 は、それぞれが 32 ビット・データを持つ 2 つの 128 M ビット SDRAM チップを示しています。アドレス、データ、およびコントロール信号は、2 つのチップに並列に配線されます。チップ・セレクト・バス (cs_n[1:0]) は、選択するチップを決定します。その結果、256 M ビットの 32 ビット幅論理メモリが実現します。

図 1-4. それぞれが 32 ビット・データを持つ 2 つの 128M ビット SDRAM チップ



ソフトウェア・プログラミング・モデル

SDRAM コントローラは、Avalon-MM インタフェースを通してアクセスされると、単独のメモリのように動作します。ソフトウェアでコンフィギュレーション可能な設定はなく、メモリ・マップト・レジスタもありません。プロセッサが SDRAM コントローラにアクセスするのに、ソフトウェア・ドライバ・ルーチンは不要です。

クロック、PLL、およびタイミングの検討事項

この項では、SDRAM コントローラ・コアからの信号と、SDRAM チップをドライブするクロックとの同期に関する問題について説明します。SDRAM トランザクション中、アドレス、データ、およびコントロール信号は、SDRAM ピンで時間ウィンドウの間有効になり、その間に SDRAM クロックがトグルして正しい値をキャプチャする必要があります。より低速なクロック周波数では、クロックは必然的に有効ウィンドウ内に収まります。より高いクロック周波数では、SDRAM クロックを補正して有効ウィンドウに揃える必要があります。

計算またはオシロスコープによる SDRAM ピン解析のいずれかによって、有効なウィンドウが発生するタイミングを決定します。その後、PLL を使用して、エッジが有効ウィンドウの中央で発生するように SDRAM クロックの位相を調整することができます。PLL の調整では、位相シフトがターゲット・ボードの特性に合うように、試行錯誤が必要な場合があります。



ターゲット・デバイスの PLL 回路について詳しくは、該当するデバイス・ファミリのハンドブックを参照してください。アルテラ FPGA のコンフィギュレーションについて詳しくは、「[ALTPLL Megafunction User Guide](#)」を参照してください。

SDRAM タイミングに影響する要因

ウィンドウの位置と接続時間は、いくつかの要因によって決まります。

- FPGA および SDRAM I/O ピンのタイミング・パラメータ — I/O タイミング・パラメータは、デバイス・ファミリとスピード・グレードによって変化します。
- FPGA のピン配置 — FPGA のロウ配線に接続された I/O ピンとコラム配線に接続された I/O ピンではタイミングが異なります。
- Quartus II コンパイルで使用されるロジック・オプション — **Fast Input Register** および **Fast Output Register** などのロジック・オプションがデザインのフィッティングに影響します。FPGA 内部のロジックとレジスタの位置は、I/O ピンへの信号の伝播遅延に影響します。
- SDRAM CAS レイテンシ

これらの結果、有効ウィンドウのタイミングは、FPGA および SDRAM デバイスの組み合わせごとに異なります。さらに、ウィンドウは Quartus II ソフトウェアのフィッティング結果とピン配置によっても異なります。

未調整 PLL の現象

PLL の調整が正しくないと検出するのは、困難な場合があります。SDRAM との間のデータ転送が全面的に失敗するとは限りません。例えば、SDRAM コントローラへの個々の転送は成功しても、バースト転送は失敗することがあります。プロセッサ・ベース・システムの場合、ソフトウェアが SDRAM の読み出しまたは書き込みは実行できるが、コードが SDRAM 内にあると実行できないときは、PLL が正しく調整されていないと考えられます。

有効な信号ウィンドウの推定

この項では、SDRAM データシートと Quartus II ソフトウェアコンパイル・レポートに記載されるタイミング・パラメータを使用して、有効な信号ウィンドウの位置と持続時間を推定する方法について説明します。ウィンドウを確認したら、SDRAM クロック・エッジがウィンドウのちょうど中央で発生するように PLL を調整します。

ウィンドウの計算は、2つのステップで行います。最初に SDRAM クロックがコントローラ・クロックからどれだけ遅れることができるかを推定し、次にどれだけ進むことができるかを推定します。最大の遅れ値および進み値を見つけたら、それらの中点を計算します。



これらの計算は推定値に過ぎません。以下の遅延は適切な PLL のチューニングにも影響を与える可能性があります。これらの計算では計上されていません。

- プリント基板の遅延に起因する信号スキュー—これらの計算ではスキュー・ゼロを想定しています。
- PLL クロック出力ノードから宛先までの遅延 — これらの計算では、PLL SDRAM クロック出力ノードからピンまでの遅延は、PLL コントローラ・クロック出力ノードから SDRAM コントローラのクロック入力までの遅延と同じと想定しています。これらのクロック遅延が大幅に異なる場合は、ウィンドウ計算でこの位相シフトを明らかにしなければなりません。

図 1-5 に SDRAM クロックがコントローラ・クロックから遅れることができる最大値の計算方法を示し、図 1-6 に進むことができる最大値の計算方法を示します。遅れはコントローラ・クロックを基準にして負のタイム・シフト、進みは正のタイム・シフトです。SDRAM クロックは、コントローラ・クロックから読み出しサイクルまたは書き込みサイクルの最大ラグ以内で遅れることができます。つまり、最大ラグ = 最小（読み出しラグ、書き込みラグ）です。同様に、SDRAM クロックは、リード・サイクルまたはライト・サイクルの最大リード以内で進むことができます。つまり、最大リード = 最小（読み出しリード、書き込みリード）です。

図 1-5. SDRAM クロック・ラグの最大値の計算

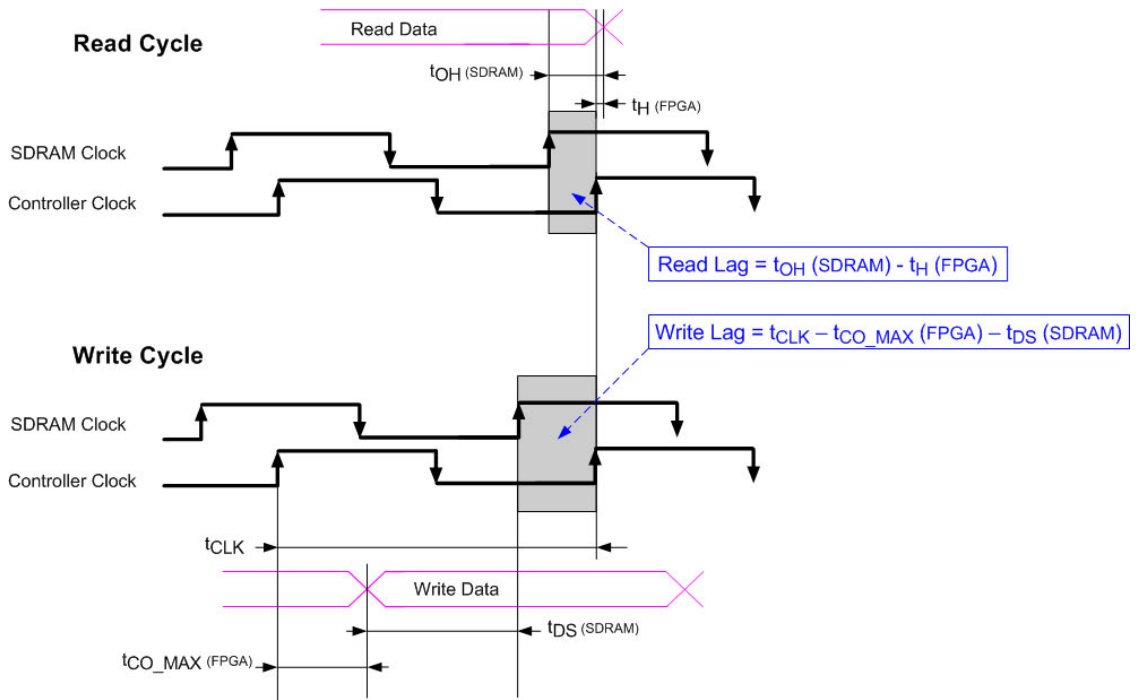
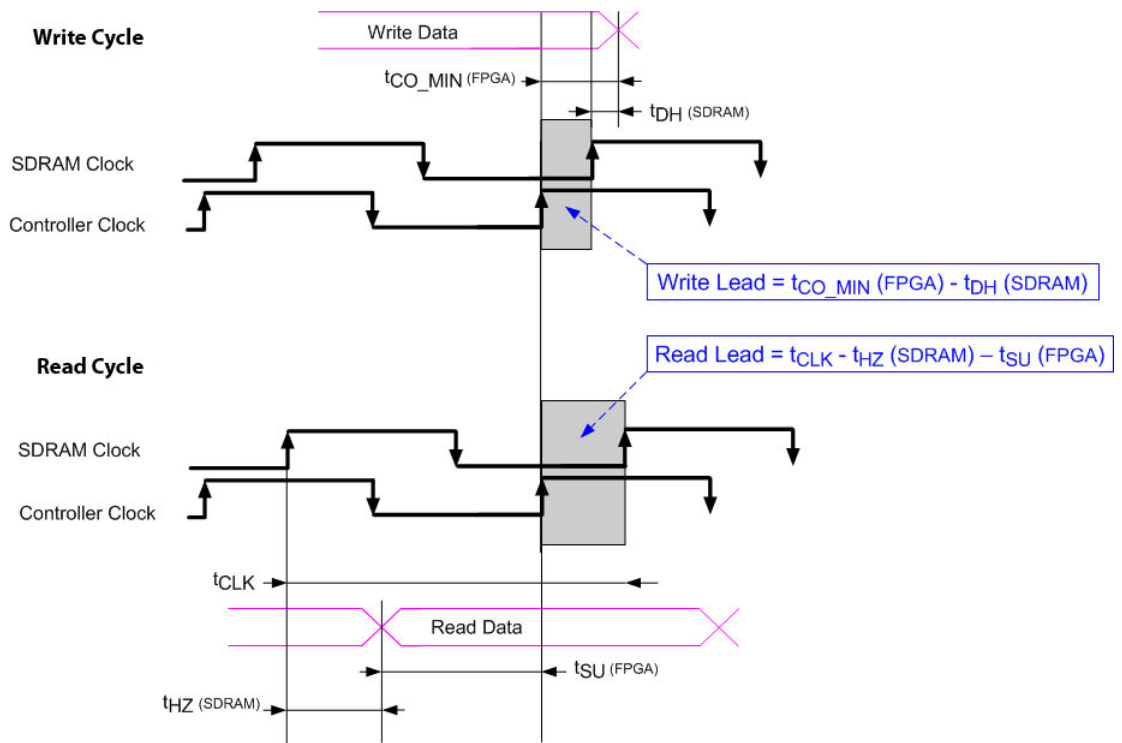


図 1-6. SDRAM クロック・リードの最大値の計算



計算例

この項では、Micron MT48LC4M32B2-7 SDRAM チップ、およびアルテラ Stratix II EP2S60F672C5 FPGA をターゲットにした FPGA デザインの単一ウィンドウの計算例を示します。この例では、3 サイクルの CAS レイテンシ (CL) と 50 MHz のクロック周波数を使用しています。FPGA のすべての SDRAM 信号は I/O セルに記憶され、Quartus II ソフトウェアの **Fast Input Register** および **Fast Output Register** ロジック・オプションでイネーブルされます。

表 1-3 に、MT48LC4M32B2 デバイス・データシートから抜粋した関連
タイミング・パラメータを示しています。

表 1-3. Micron MT48LC4M32B2 SDRAM デバイスのタイミング・パラメータ				
パラメータ		シンボル	-7 スピード・グレードでの値 (ns)	
			Min.	Max.
CLK からのアクセス・ タイム (ポジティブ・エッジ)	CL = 3	$t_{AC(3)}$		5.5
	CL = 2	$t_{AC(2)}$		8
	CL = 1	$t_{AC(1)}$		17
アドレス・ホールド時間		t_{AH}	1	
アドレス・セットアップ時間		t_{AS}	2	
CLK High レベル幅		t_{CH}	2.75	
CLK Low レベル幅		t_{CL}	2.75	
クロック・サイクル 時間	CL = 3	$t_{CK(3)}$	7	
	CL = 2	$t_{CK(2)}$	10	
	CL = 1	$t_{CK(1)}$	20	
CKE ホールド時間		t_{CKH}	1	
CKE セットアップ時間		t_{CKS}	2	
CS#, RAS#, CAS#, WE#, DQM ホールド時間		t_{CMH}	1	
CS#, RAS#, CAS#, WE#, DQM セットアップ 時間		t_{CMS}	2	
データ入力ホールド時間		t_{DH}	1	
データ入力セットアップ時間		t_{DS}	2	
データ出力ハイ・ インピーダンス時間	CL = 3	$t_{HZ(3)}$		5.5
	CL = 2	$t_{HZ(2)}$		8
	CL = 1	$t_{HZ(1)}$		17
データ出力ロー・インピーダンス時間		t_{LZ}	1	
データ出力ホールド時間		t_{OH}	2.5	

表 1-4 に、Quartus II Compilation Report の Timing Analyzer セクションから引用した関連 FPGA タイミング情報を示します。値は、SDRAM に関連するすべての FPGA ピンでの最大値と最小値です。これらの信号用のレジスタが I/O セル内に配置されているため、FPGA の SDRAM ピン間のタイミング変動はわずか（100 ps 未満）です。

表 1-4. FPGA I/O タイミング・パラメータ		
パラメータ	シンボル	値 (ns)
クロック周期	t_{CLK}	20
最小 clock-to-output 時間	t_{CO_MIN}	2.399
最大 clock-to-output 時間	t_{CO_MAX}	2.477
最大ホールド時間（クロック後）	t_{H_MAX}	-5.607
最大セットアップ時間（クロック前）	t_{SU_MAX}	5.936



Quartus II ソフトウェアでデザインをコンパイルして、FPGA デザインの I/O タイミング情報を取得する必要があります。アルテラ・デバイス・ファミリのデータシートには、各デバイスの汎用 I/O タイミング情報が記載されていますが、Quartus II Compilation Report には特定のデザインについての最も正確なタイミング情報が示されています。



Compilation Report のタイミング値は、フィッティング、ピン配置、および他の Quartus II ロジック設定によって変わる可能性があります。Quartus II ソフトウェアでデザインを再コンパイルするときは、I/O タイミングが大きく変わっていないことを確認してください。

表 1-3 および表 1-4 の値を使用して、下記のとおり、図 1-5 および 1-6 からの計算を実行できます。

SDRAM クロックは、コントローラ・クロックよりも小さい読み出しラグまたは書き込みラグで進むことができます。

$$\begin{aligned} \text{読み出しラグ} &= t_{OH}(\text{SDRAM}) - t_{H_MAX}(\text{FPGA}) \\ &= 2.5 \text{ ns} - (-5.607 \text{ ns}) = 8.107 \text{ ns} \end{aligned}$$

または

$$\begin{aligned} \text{書き込みラグ} &= t_{\text{CLK}} - t_{\text{CO_MAX}}(\text{FPGA}) - t_{\text{DS}}(\text{SDRAM}) \\ &= 20 \text{ ns} - 2.477 \text{ ns} - 2 \text{ ns} = 15.523 \text{ ns} \end{aligned}$$

SDRAM クロックは、コントローラ・クロックよりも小さい読み出しリードまたは書き込みリードで進むことができます。

$$\begin{aligned} \text{読み出しリード} &= t_{\text{CO_MIN}}(\text{FPGA}) - t_{\text{DH}}(\text{SDRAM}) \\ &= 2.399 \text{ ns} - 1.0 \text{ ns} = 1.399 \text{ ns} \end{aligned}$$

または

$$\begin{aligned} \text{書き込みリード} &= t_{\text{CLK}} - t_{\text{HZ}(3)}(\text{SDRAM}) - t_{\text{SU_MAX}}(\text{FPGA}) \\ &= 20 \text{ ns} - 5.5 \text{ ns} - 5.936 \text{ ns} = 8.564 \text{ ns} \end{aligned}$$

したがって、この例の場合、SDRAM クロックの位相は、コントローラ・クロックを基準にして、 $-8.107 \text{ ns} \sim 1.399 \text{ ns}$ の範囲でシフトできます。このウィンドウの中央での位相シフトを選択すると、 $(-8.107 + 1.399)/2 = -3.35 \text{ ns}$ となります。

参考資料

この章では以下のドキュメントを参照しています。

- [「Avalon Interface Specifications」](#)
- [「PLL コア」の章 \(Quartus II ハンドブック Volume 5: エンベデッド・ペリフェラル\)](#)
- [「AN 351: Nios II エンベデッド・プロセッサ・デザインのシミュレーション」](#)
- [「ALTPLL Megafunction User Guide」](#)

改訂履歴

表 1-5 に、本資料の改訂履歴を示します。

表 1-5. 改訂履歴		
日付および ドキュメント・ バージョン	変更内容	概要
2008年5月 v8.0.0	前バージョンからの内容の変更はありません。	—
2007年10月 v7.2.0	前バージョンからの内容の変更はありません。	—
2007年5月 v7.1.0	<ul style="list-style-type: none"> ● トライステート・ブリッジによってピンを共有するための新しいメカニズムを反映して、Parameter Settings Memory Profile (パラメータ設定メモリ・プロフィール) ページの説明を更新。 ● 「概要」の項に目次を追加。 ● 「参考資料」の項を追加。 	—
2007年3月 v7.0.0	前バージョンからの内容の変更はありません。	—
2006年11月 v6.1.0	<ul style="list-style-type: none"> ● Avalon テクノロジーが変更されたため、Avalon 用語を更新。 ● 旧「Avalon スイッチ・ファブリック」を「システム・インタコネクト・ファブリック」に変更。 ● 旧「Avalon インタフェース」を「Avalon Memory-Mapped インタフェース」に変更。 	v6.1 のリリースで、アルテラは Avalon ストリーミング・インタフェースをリリースしたが、これにより既存の Avalon 用語を言い換える必要が生じた。
2006年5月 v6.0.0	章のタイトルは変更されましたが、前バージョンからの内容の変更はありません。	—
2005年12月 v5.1.1	<ul style="list-style-type: none"> ● 図 1-1 を更新。 ● 「オフチップ SDRAM インタフェース」および「ボード・レイアウトおよびピン配置の考慮事項」の項を更新。 ● 「クロック、PLL、およびタイミングの検討事項」の項を追加。 	—
2005年10月 v5.1.0	前バージョンからの内容の変更はありません。	—
2005年5月 v5.0.0	前バージョンからの内容の変更はありません。以前は Nios II プロセッサ・リファレンス・ハンドブックに収録。	—
2004年9月 v1.1	Nios II 1.01 のリリースにより更新。	—
2004年5月 v1.0	初版	—