



XAPP723 (v1.4) 2007 年 10 月 17 日

Virtex-4 デバイスを使用した DDR2 コントローラ (267MHz 以上)

本資料は英語版 (v1.4) を翻訳したものです。英語の更新バージョンがリリースされている場合には、最新の英語版を必ずご参照ください。

概要

DDR2 SDRAM は DDR SDRAM をしのぐ新機能を備え、666Mb/s 以上のデータ レートを実現したデバイスです。データ レートの高速化に伴い、FPGA のコントローラおよび I/O にもより高い性能が求められます。メモリの動作速度に対応したコントローラを使用することにより、広帯域幅を達成できます。

はじめに

このアプリケーション ノートでは、Micron DDR2 SDRAM デバイスのインターフェイスとして使用する DDR2 コントローラを Virtex™-4 デバイスでインプリメントする方法を説明します。267MHz 以上のパフォーマンス レベルでは、このアプリケーション ノートで概説されているコントローラ デザイン および、アプリケーション ノート [XAPP721](#) 『ISERDES と OSERDES を使用した高性能 DDR2 SDRAM インターフェイスのデータ キャプチャ』に説明されているデータ キャプチャの手法を使用してください。

このアプリケーション ノートでは、まず、DDR2 SDRAM デバイス機能の概要を簡単に説明し、次に高速 DDR2 メモリへのインターフェイスとしてのコントローラの使用方法を詳細に説明します。また、コントローラへのバックエンド ユーザー インターフェイスについても解説します。

DDR2 SDRAM

概要

DDR2 SDRAM デバイスは、DDR SDRAM ファミリの次世代デバイスで、SSTL 1.8V I/O 規格を採用しています。次のセクションでは、このデバイスで使用可能な機能について、また、DDR SDRAM デバイスとの相違点について説明します。

DDR2 SDRAM デバイスは DDR アーキテクチャを使用することで高速動作を実現しています。メモリは、コントローラが供給する差動クロックを使用して動作します。コマンドは、クロックのすべての立ち上がりエッジで取得されます。双方向データ ストローブ (DQS) は、レシーバでのデータ キャプチャで使用するため、データと共に送信されます。DQS は読み出しでは DDR2 SDRAM デバイスによって、また書き込みではコントローラによって送信されます。読み出しではデータのエッジに揃えられ、書き込みではデータの中央に揃えられます。

DDR2 SDRAM デバイスへの読み出しおよび書き込みアクセスはバースト対応です。ACTIVE コマンドが取得されるとアクセスが開始し、その後、READ コマンドまたは WRITE コマンドが実行されます。アクセスするバンクおよび行は、ACTIVE コマンドと共に取得されたアドレス ビットを使用して選択され、バースト アクセスの開始列およびバンクは、READ コマンドまたは WRITE コマンドと共に取得されたアドレス ビットを使用して選択されます。

DDR2 コントローラのリファレンス デザインには、書き込みアドレス、書き込みデータ、および読み出しアドレスを生成するユーザー バックエンド インターフェイスが含まれています。この情報は 3 つのバックエンド FIFO に格納され、バックエンド モジュールとコントローラ モジュール間のアドレスおよびデータ同期に使用されます。コントローラは、アドレス FIFO のアドレスの有無に応じて、メモリのタイミング要件を考慮しながら、メモリに正しいコマンドを発行します。論理ブロックのインプリメンテーションの詳細は、次のセクションで説明します。

© 2005-2007 Xilinx, Inc. All Rights Reserved. XILINX, Xilinx ロゴ、およびその他本文に含まれる商標名は Xilinx の商標です。本文書に記載されている「Xilinx」、ザイリックスのロゴ、およびザイリックスが所有する製品名等は、米国 Xilinx Inc. の米国における登録商標です。その他に記載されている会社名および製品名等は、各社の商標または登録商標です。保証否認の通知: Xilinx ではデザイン、コード、その他の情報を「現状有姿の状態」で提供しています。この特徴、アプリケーションまたは規格の一実施例としてデザイン、コード、その他の情報を提供しておりますが、Xilinx はこの実施例が権利侵害のクレームを全く受けないということを表明するものではありません。お客様がご自分で実装される場合には、必要な権利の許諾を受ける責任があります。Xilinx は、実装の妥当性に関するいかなる保証を行なうものではありません。この保証否認の対象となる保証には、権利侵害のクレームを受けないことの保証または表明、および市場性に対する適合性についての黙示的な保証も含まれます。

コントローラの DDR2 SDRAM コマンド

表 1 に、コントローラにより発行されるコマンドを示します。各コマンドは、行アドレス セレクト ($\overline{\text{RAS}}$)、列アドレス セレクト ($\overline{\text{CAS}}$)、および書き込みイネーブル ($\overline{\text{WE}}$) の制御信号を使用したメモリで検知されます。クロック イネーブル(CKE) はデバイスのコンフィギュレーション後は High に、チップ セレクト ($\overline{\text{CS}}$) はデバイスの動作中は Low に保持されます。「モードレジスタの定義」では、コントローラでサポートされる DDR2 コマンドの機能を説明します。

表 1: DDR2 コマンド

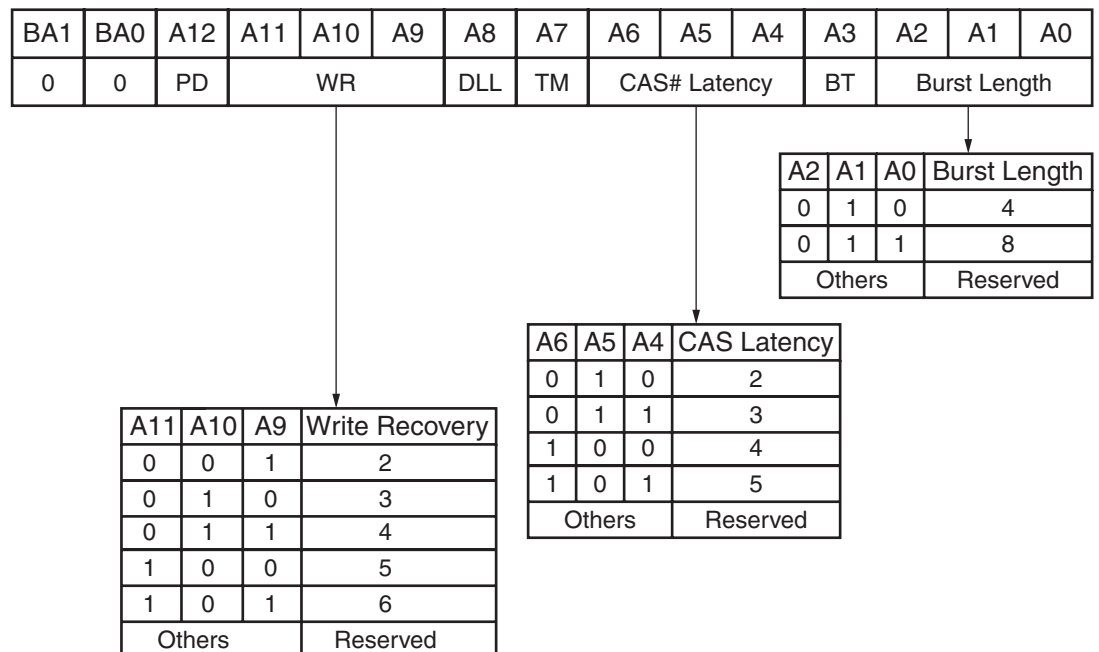
ステップ	機能	$\overline{\text{RAS}}$	$\overline{\text{CAS}}$	$\overline{\text{WE}}$
1	Load Mode	L	L	L
2	Auto Refresh	L	L	H
3	Precharge ⁽¹⁾	L	H	L
4	Bank Activate	L	H	H
5	Write	H	L	L
6	Read	H	L	H
7	No Operation/IDLE	H	H	H

メモ:

1. アドレス信号 A10 は、すべてのバンクをプリチャージするときは High に保持し、1つのバンクをプリチャージするときは Low に保持します。

モードレジスタの定義

モードレジスタは、DDR2 SDRAM の動作モードを指定します。バースト長、バーストタイプ、CAS レイテンシおよび動作モードを指定できます。図 1 に、このコントローラで使用するモードレジスタの機能を示します。



x723_01_091505

図 1: モードレジスタ

バンク アドレス BA1 および BA0 は、モードレジスタを選択します。表 2 には、バンク アドレス ビットのコンフィギュレーションを示します。

表 2: バンク アドレス ビット コンフィギュレーション

BA1	BA0	モードレジスタ
0	0	Mode Register (MR)
0	1	EMR1
1	0	EMR2
1	1	EMR3

拡張モード レジスタの定義

拡張モード レジスタ (表 3) では、モードレジスタで制御できない機能を設定できます。設定できる機能は、DLL イネーブル/ディスエーブル、出力駆動電流、オンチップ終端 (ODT)、Posted CAS Additive Latency (AL)、オフチップドライバインピーダンス調整 (OCD)、DQS イネーブル/ディスエーブル、RDQS/RDQS イネーブル/ディスエーブル、出力ディスエーブル/イネーブルです。オフチップドライバ調整 (OCD) はこのリファレンスデザインでは使用しません。

表 3: 拡張モード レジスタ

BA1	BA0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	Out	RDQS	DQS	OCD Program			R _{TT}	Posted CAS			R _{TT}	ODS	DLL

拡張モード レジスタ 2 (EMR2)

バンク アドレスは 10 (BA1 は High、BA0 は Low) に設定されています。アドレス ビットはすべて Low に設定されています。

拡張モード レジスタ 3 (EMR3)

バンク アドレス ビットは 11 (BA1 および BA0 は High) に設定されています。アドレス ビットはすべて Low に設定されています。

初期化シーケンス

コントローラ ステート マシンで使用される初期化シーケンスは、DDR2 SDRAM 仕様に従っています。インターフェイスには、メモリの電圧の要件を適用してください。次に初期化時に発行されるコマンドを示します。

1. 電力とクロックが安定した後、200µs 間 NOP または DESELECT コマンドが適用されます。
2. CKE がアサートされます。
3. 400ns 経過すると、PRECHARGE ALL コマンドが発行されます。
4. EMR (2) コマンドが発行されます。BA0 は Low に、BA1 は High に保持されます。
5. EMR (3) コマンドが発行されます。BA0 および BA1 はどちらも High に保持されます。
6. EMR コマンドが発行され、メモリ DLL がイネーブルになります。BA1 および A0 は Low に、BA0 は High に保持されます。
7. MODE REGISTER SET コマンドが発行され、DLL がリセットされます。DLL がロック状態になるには、200 クロック サイクルが必要です。
8. PRECHARGE ALL コマンドが発行されます。
9. AUTO REFRESH コマンドが 2 回発行されます。

10. MODE REGISTER SET コマンドが発行され、A8 が Low になり、デバイスの動作が初期化されます。
11. EMR コマンドが発行され、ビット E7、E8、および E9 が 1 に設定され、OCD default がイネーブルになります。
12. EMR コマンドが発行され、ビット E7、E8、および E9 が 0 に設定され、OCD exit がイネーブルになります。

初期化シーケンスが完了すると、コントローラより DDR2 SDRAM メモリに対してダミーの書き込みコマンドが発行され、続けてダミーの読み出しコマンドが発行されます。これにより、データパス モジュールで、Virtex-4 入力遅延ブロックに対して適切なタップ数が選択されます。タップ数が決定されると、dp_dly_slct_done 信号がコントローラにアサートされ、コントローラが IDLE 状態になります。

PRECHARGE コマンド

PRECHARGE コマンドは、特定のバンクの現在アクティブな行を非アクティブにします。コマンドの発行後、特定の時間 (t_{RP}) が経過すると、次の行をアクティブにできるようになります。入力 A10 は、1 つのバンクをプリチャージするか、すべてのバンクをプリチャージするかを指定します。

AUTO REFRESH コマンド

DDR2 デバイスは、7.8 μ s おきのリフレッシュする必要があります。コントローラは、AUTO REFRESH コマンドを要求する回路を内蔵しており、リフレッシュ カウンタには、16 出力分周のシステム クロックが使用されます。auto_ref 信号は、アサートされると AUTO REFRESH コマンドを要求し、前の AUTO REFRESH コマンド発行後 7.8 μ s 間 High に保持されます。その時点で進行中のバーストが完了すると、コントローラより AUTO REFRESH コマンドが発行されます。AUTO REFRESH コマンドは、このコントローラのデザインでは最優先されます。

ACTIVE コマンド

DDR2 SDRAM メモリ内のバンクに READ または WRITE コマンドが発行される前に、ACTIVE コマンドでバンク内の 1 行をアクティブにする必要があります。行がアクティブになると、 t_{RCD} 仕様に従って、その行に対して READ または WRITE コマンドを発行できるようになります。また、DDR2 SDRAM デバイスは Posted CAS AL という新しい機能をサポートしています。実際の READ または WRITE コマンドの内部デバイスへの取得を AL クロック サイクルを使用して遅延させて、 t_{RCD} タイムの前に READ または WRITE コマンドを発行することができます。

コントローラは、競合を検知すると PRECHARGE コマンドを発行して現在アクティブな行を非アクティブにし、ACTIVE コマンドを発行して新しい行をアクティブにします。競合は、入力されたアドレスが現在アクティブな行と異なる行を参照している場合に発生します。

READ コマンド

READ コマンドは、アクティブな行に対してバースト読み出しアクセスを開始します。BA0 および BA1 の値でバンク アドレスが選択され、アドレス入力 $A_0 \sim A_i$ で開始列を選択します。読み出しバーストの完了後も、プリチャージされるまで、行はアクティブのままです。

図 2 に、追加レイテンシが 0 の場合の READ コマンドを示します。この場合の読み出しレイテンシは 3 で、CAS レイテンシと同じです。

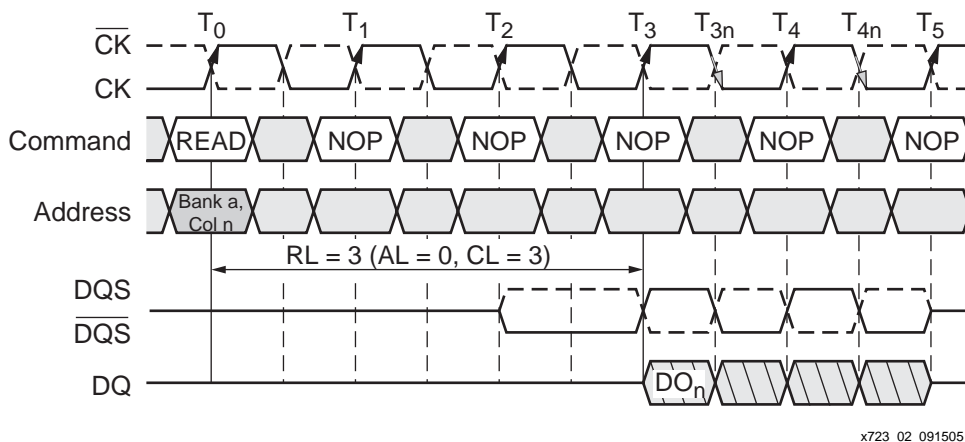


図 2： READ コマンド例

WRITE コマンド

WRITE コマンドは、アクティブな行に対してバースト書き込みアクセスを開始します。BA0 および BA1 はバンク アドレスを指定し、アドレス入力 $A_0 \sim A_i$ はアクティブ行の開始列の位置を指定します。DDR2 SDRAM で使用される書き込みレイテンシ (WL) の値は、読み出しレイテンシから 1 クロック サイクルを差し引いた値と同じです。

$$\text{書き込みレイテンシ} = \text{読み出しレイテンシ} - 1 = (\text{追加レイテンシ} + \text{CAS レイテンシ}) - 1$$

図 3 に、書き込みレイテンシが 2 の書き込みバーストの波形を示します。WRITE コマンドから DQS 信号の最初の立ち上がりエッジまでの時間は、書き込みレイテンシにより決まります。

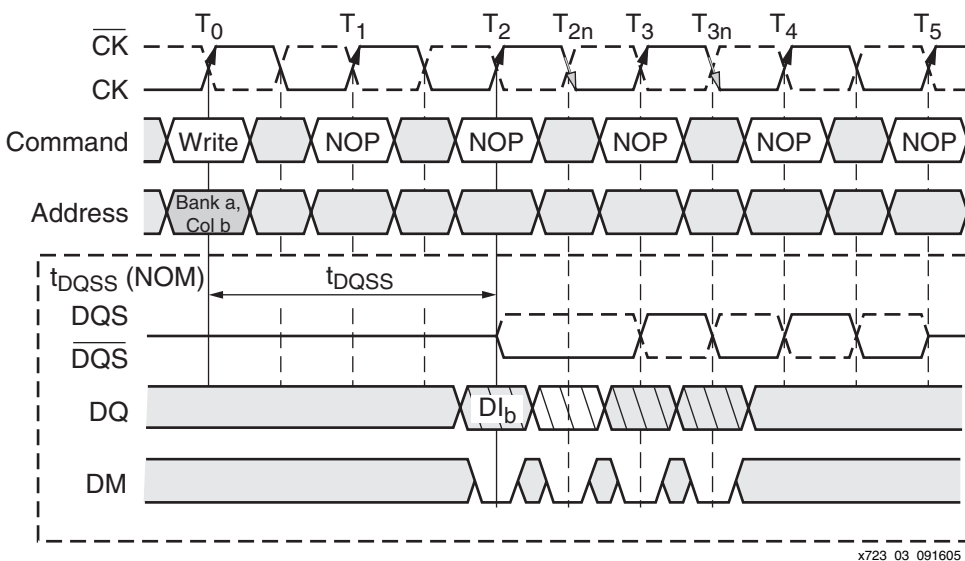


図 3： WRITE コマンド例

DDR2 SDRAM インターフェイス デザイン

DDR2 コントローラへのユーザー インターフェイスのクロック周波数 (図 4) とデータパスのクロック周波数はインターフェイスの半分であるため、周波数が 267MHz を超える場合のデザイン マージンが改善されています。コントローラが半分の周波数で動作しても、スループットやレイテンシには影響しません。DDR2 SDRAM デバイスでサポートされる最小のバースト長は 4 で、コマンドは 1 クロックおきに必要だけです。使用されるバースト長は次のとおりです。

- 4 の場合 : コントローラは 2 分の 1 の周波数のクロック サイクルごとにコマンドを発行
- 8 の場合 : コントローラは 2 分の 1 の周波数のクロック サイクルおきにコマンドを発行

ユーザー インターフェイスの FIFO はすべて非同期 FIFO なので、ユーザーのバックエンドはどの周波数でも動作します。I/O は指定した周波数でトグルします。

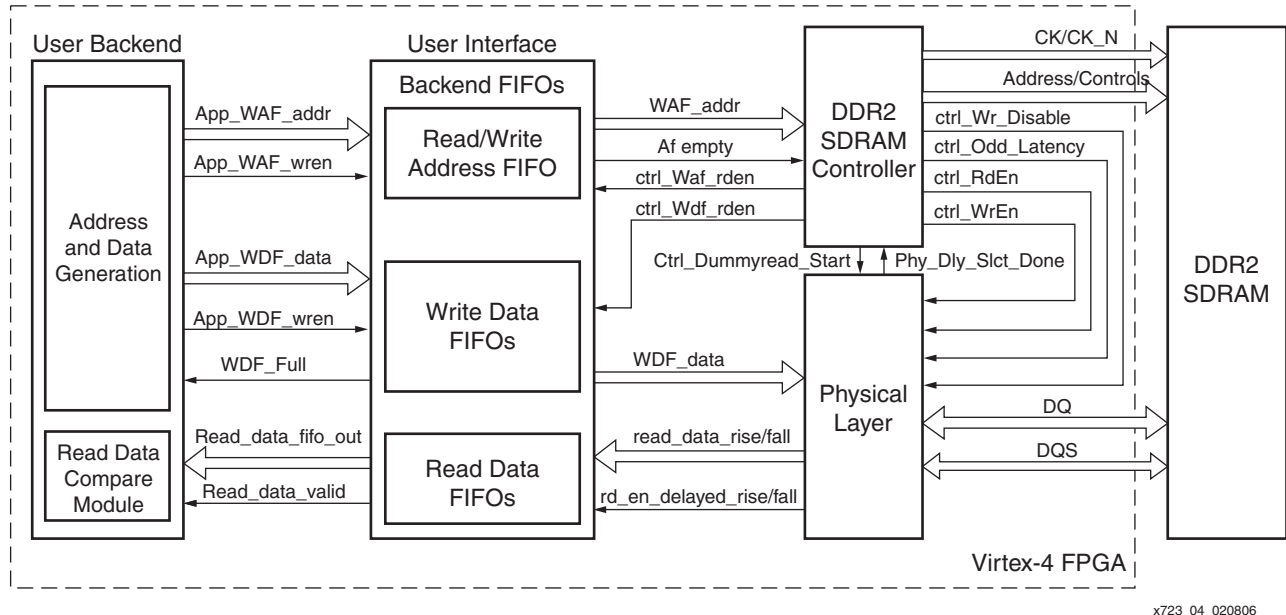


図 4 : DDR2 のインターフェイス ブロック図

ユーザー バックエンド

バックエンドから供給されるアドレス パターンおよびデータ パターンを使用して、DDR2 コントローラ デザインをいろいろな角度からテストできます。バックエンドには、バックエンド ステート マシン、読み出しデータ コンパレータ、およびデータ生成モジュールが含まれます。データ生成モジュールでは、メモリに書き込まれるさまざまなアドレスおよびデータ パターンが生成されます。アドレス位置はブロック RAM にあらかじめ保存されており、ここでは ROM として使用されます。保存されているアドレス値は、DDR2 SDRAM デバイスの異なる行およびバンクへのアクセスをテストするために選択されています。データ パターン ジェネレータには、データ パターンを生成するステート マシンが含まれています。バックエンド ステート マシンは、ユーザー バックエンドをエミュレートします。ステート マシンは、書き込みイネーブル信号または読み込みイネーブル信号を送信して、データ ジェネレータ モジュールでアクセスする FIFO を指定します。

ユーザー インターフェイス

バックエンド ユーザー インターフェイスには、次の 3 つの FIFO が含まれます。

- アドレス FIFO
- 書き込みデータ FIFO
- 読み出しデータ FIFO

最初の 2 つの FIFO はユーザー バックエンド モジュールによりアクセスされ、読み出しデータ FIFO は取り込まれた読み出しデータを保存するためにデータパス モジュールによりアクセスされます。

ユーザーからコントローラへのインターフェイス

表 4 に、ユーザー インターフェイスとコントローラ間で使用される信号を示します。

表 4: ユーザー インターフェイスとコントローラの間で使用される信号 (ポート)

ポート名	ポート幅	ポートの説明	メモ
Af_addr	36	ユーザー インターフェイスのアドレス FIFO の出力。次のアドレスをマップします。 <ul style="list-style-type: none"> • メモリ アドレス (CS、バンク、行、列) - [31:0] • 予約済み - [35] • ダイナミック コマンド要求 - [34:32] 	FIFO full ステータス フラグをモニタし、アドレス FIFO にアドレスを書き込みます。
Af_empty	1	ユーザー インターフェイスアドレス FIFO empty ステータス フラグ出力。この信号がディアサートされると、FIFO の出力にあるアドレスがコントローラにより処理されます。	FIFO16 の Empty フラグ
ctrl_Waf_RdEn	1	ユーザー インターフェイスのアドレス FIFO への読み出しイネーブル入力	この信号は、コントローラのステートが書き込み、読み出し、モードレジスタの読み込み、すべてをプリチャージ、自動リフレッシュ、あるいはダイナミック コマンド要求の結果アクティブとなった場合、1 クロック サイクル間アサートされます。
ctrl_Wdf_RdEn	1	ユーザー インターフェイスの書き込みデータ FIFO への読み出しイネーブル入力	コントローラは、最初の書き込みステートに遷移後にこの信号を 1 クロック サイクル間アサートします。バースト長が 8 の場合は、2 クロック サイクル間アサートします。 WRITE コマンド送信前に、書き込みデータ FIFO にバースト長分のデータが必要です。 たとえば、64 ビット データ バスおよびバースト長 4 の場合、 WRITE コマンド送信前に、各書き込みアドレスに対して、2 つの 128 ビット データ ワードが書き込みデータ FIFO に入力されている必要があります。

表 5 に、メモリ アドレス (Af_addr) の列アドレス、行アドレス、バンクアドレス、ディープ メモリ インターフェイスのチップ セレクト幅を示します。

表 5: Af_addr メモリ アドレス

アドレス	説明
列アドレス	[col_ap_width - 1:0]
行アドレス	[col_ap_width + row_address - 1:col_ap_width]
バンクアドレス	[col_ap_width + row_address + bank_address - 1:col_ap_width + row_address]
チップ セレクト	[col_ap_width + row_address + bank_address + chip_address - 1:col_ap_width + row_address + bank_address]

Af_addr のアドレス空間は不連続です。これはユーザー インターフェイス アドレス バスの Af_addr [10] ビットが制御ロジックで無視されるためです。DDR2 デバイスへのメモリ コントローラのインターフェイスが 9 ビットの場合は、Af_addr[9] も無視されます。列アドレス幅パラメータ col_ap_width には自動プリチャージ ビット (A10) および列アドレス パラメータが含まれます。列アドレス パラメータは、選択されたメモリ コンポーネントの列アドレス ビット数です。定義を次に示します。

- 9 ビットの列アドレスでは、col_ap_width は 11 です。下位 9 ビットが列アドレスです。ビット A9 は使用されず、ビット A10 は通常の読み出しおよび書き込みの間 Low に固定されます。このため、自動プリチャージはサポートされません。PRECHARGE コマンドの実行中は内部的に、A10 ビットの先頭に col_ap_width パラメータが追加されます。
- 10 ビットの列アドレスでは、col_ap_width は 11 です。
- 11 ビットの列アドレスでは、col_ap_width は 12 です。

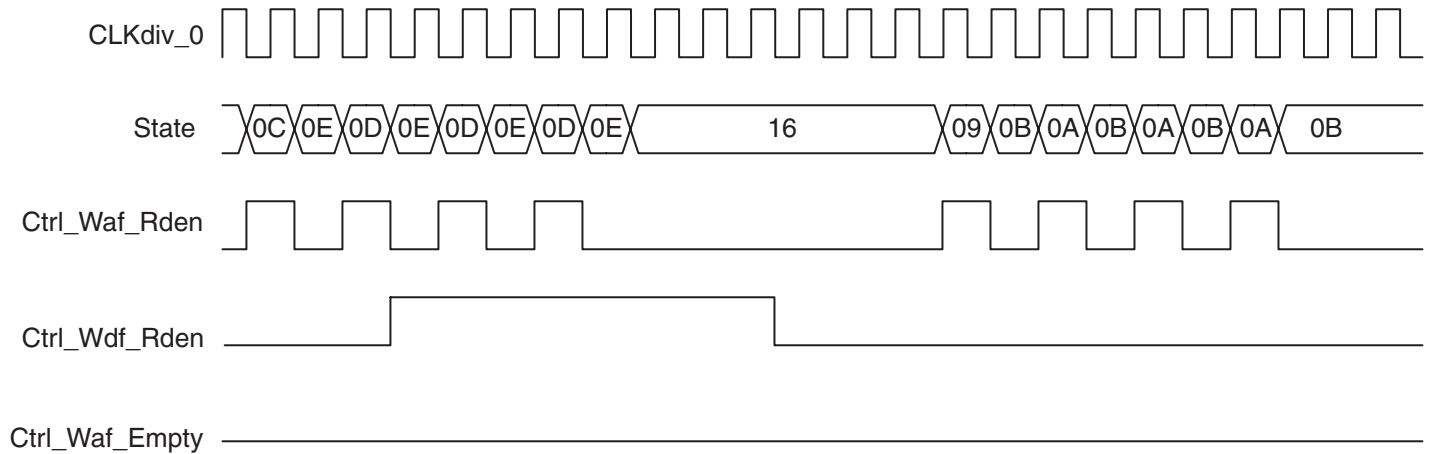
ダイナミック コマンド要求

表 6 に、オプションのコマンドを示します。ここに示されたコマンドは、コントローラの通常動作には不要です。アプリケーションで必要であれば、使用することもできます。

表 6: コマンドのオプション

コマンド	説明
000	LOAD MODE REGISTER
001	AUTO REFRESH
010	PRECHARGE ALL
011	ACTIVE
100	WRITE
101	READ
110	NOP
111	NOP

図 5 に、バースト長 8 の、4 連続書き込みとそれに続く 4 連続読み出しを示します。



X723_05_091905

図 5：バースト長 8 の連続書き込みとそれに続く連続読みだし

表 7 には、図 5 のステート信号値を示します。

表 7：図 5 のステート信号値

ステート信号値	説明
0C	最初の書き込み
0E	書き込み待ち
0D	バースト書き込み
16	書き込み読み出し
09	最初の書き込み
0B	読み出し待ち
0A	バースト読み出し

コントローラから物理レイヤへのインターフェイス

表 8 に、コントローラと物理レイヤの間で使用される信号を示します。

表 8: コントローラと物理レイヤの間で使用される信号 (ポート)

信号名	信号幅	信号の説明	メモ
ctrl_WrEn	1	コントローラから書き込みデータバスへの出力。この信号がアサートされると、書き込み DQS および書き込み DQ の生成が開始します。	この信号は、バースト長 4 の場合は 2 コントローラ クロック サイクル間、バースト長 8 の場合は 3 コントローラ クロック サイクル間アサートされます。 CAS レイテンシの値が 4 および 5 の場合は、WRITE コマンドの前で 1 コントローラ クロック サイクル間アサートされます。
ctrl_wr_disable	1	コントローラから書き込みデータバスへの出力。この信号がディアサートされると、書き込み DQS および書き込み DQ の生成が終了します。	この信号は、バースト長 4 の場合は 1 コントローラ クロック サイクル間、バースト長 8 の場合は 2 コントローラ クロック サイクル間アサートされます。 CAS レイテンシの値が 4 および 5 の場合は、WRITE コマンドの前で 1 コントローラ クロック サイクル間アサートされます。
ctrl_Odd_Latency	1	コントローラから書き込みデータバスへの出力。選択された CAS レイテンシが奇数のときアサートされます。正しい書き込みレイテンシの後、書き込み DQS および書き込み DQ の生成に必要です (書き込みレイテンシ = CAS レイテンシ - 1)。	
ctrl_Dummyread_Start	1	コントローラから読み出しデータバスへの出力。アサートされると、ストロープおよびデータのキャリブレーションが開始します。	データバスに有効な読み出しデータがある場合は、この信号が必ずアサートされます。dp_dly_slct_done 信号がアサートされると、ディアサートされます。
dp_dly_slct_done	1	読み出しデータバスからコントローラへの出力。ストロープおよびデータのキャリブレーションが終了したことを示します。	データおよびストロープのキャリブレーションが終了すると、アサートされます。通常の操作は、この信号のアサート後に開始されます。
ctrl_RdEn	1	コントローラから読み出しデータバスへの読み出しイネーブル信号の出力	この信号は、バースト長 4 の場合は 1 コントローラ クロック サイクル間、バースト長 8 の場合は 2 コントローラ クロック サイクル間アサートされます。 CAS レイテンシおよび追加レイテンシの値により、この信号と読み出しステートのタイミング関係が決定します。

図 6 にコントローラから物理レイヤへの制御信号のタイミング波形を示します。

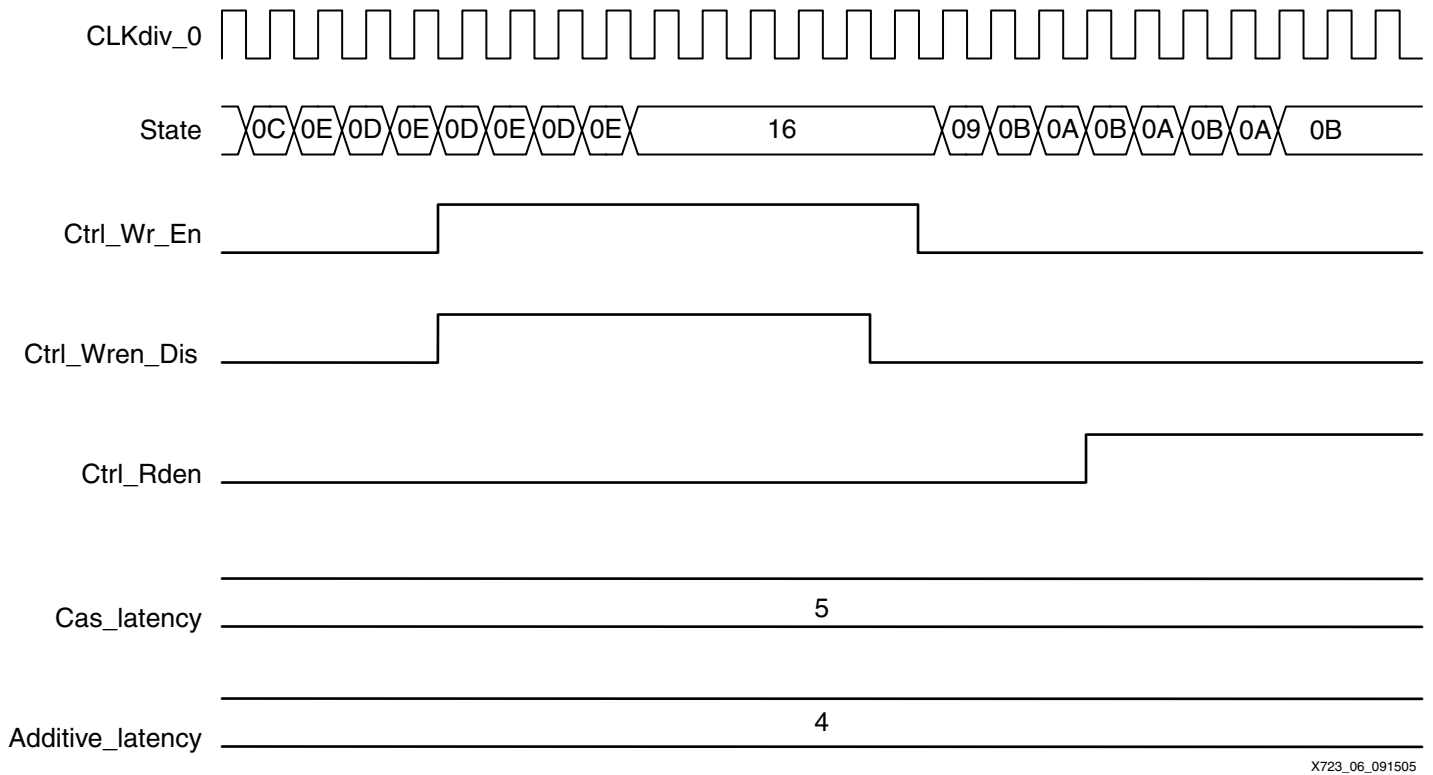


図 6：コントローラから物理レイヤへの制御信号のタイミング波形

コントローラの インプリメント

コントローラのクロックの周波数は、インターフェイスの半分です。このため、アドレス、バンクアドレス、およびコマンド信号 (RAS、CAS、および WE) は高速メモリ インターフェイス クロックの 2 クロック サイクル間アサートされます。制御信号 (CS、CKE、および ODT) は半分の周波数のクロックの DDR で、制御信号は、高速メモリ インターフェイス クロックの 1 クロック サイクルだけアサートされます。

コントローラ ステート マシンは、メモリのタイミング要件を決定しながら適切な順序でコマンドを発行します。

次のセクションでコントローラ ステート マシンの各段階について、詳しく説明します。

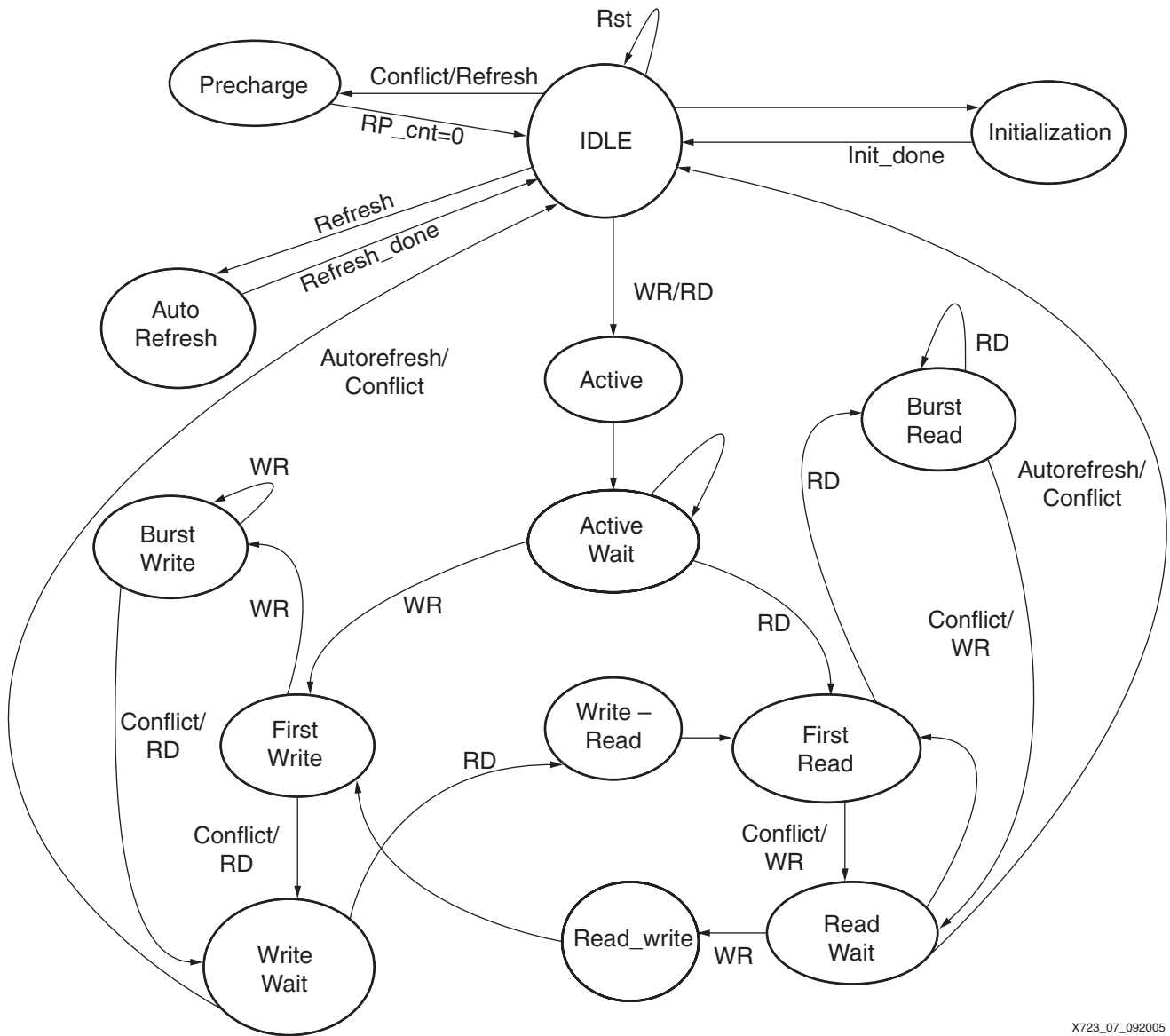


図 7: DDR2 コントローラ ステート マシン

図 7 に、DDR2 コントローラ ステート マシンを示します。コントローラからメモリにコマンドを発行する前の動作は次のとおりです。

1. アドレス FIFO は FWFT (First-Word-Fall-Through) モードです。このモードでは、FIFO に書き込まれた最初のアドレスが FIFO の出力となります。アドレスはコントローラによりデコードされます。

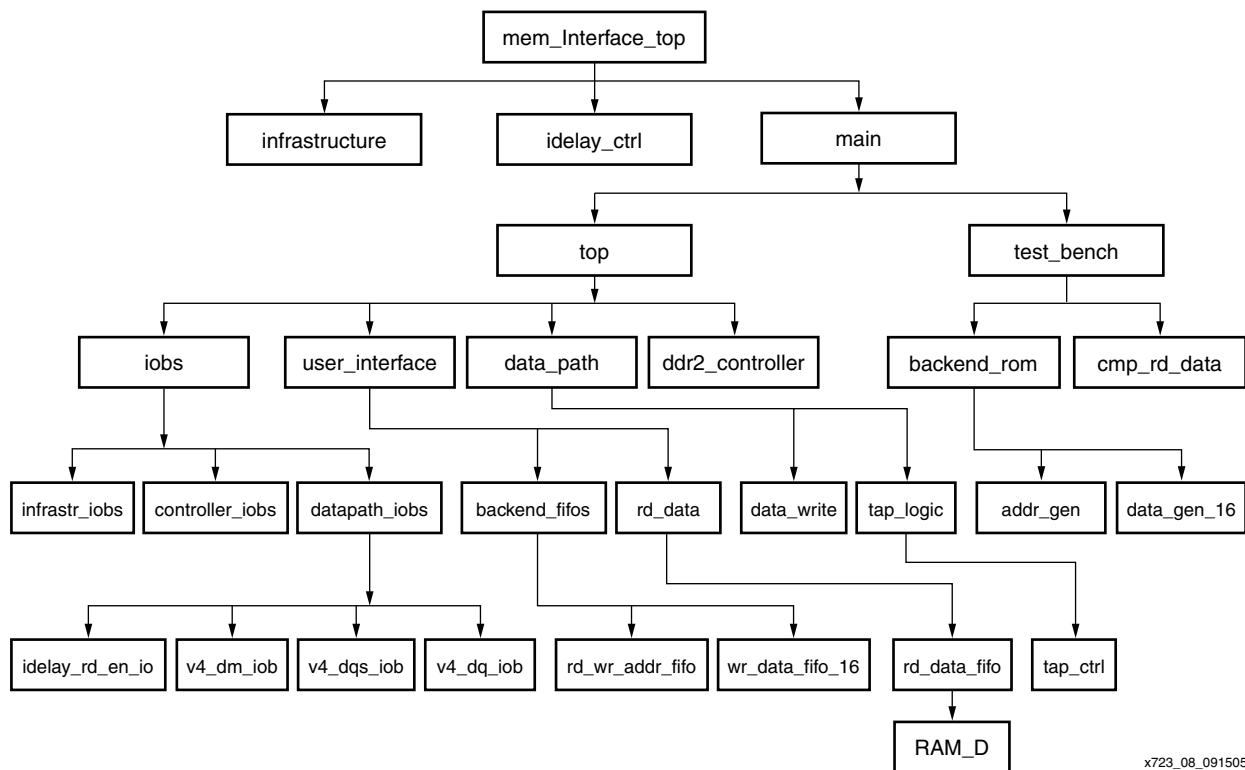
- すべてのバンクがプリチャージされている場合は対応するバンクの行がアクティブになり、既にアクティブな行がある場合は、その行とバンクのアドレスが新しい行とバンクのアドレスと比較されます。異なる場合は、現在アクティブな行がプリチャージされ、読み出し/書き込みステートに遷移する前に ACTIVE コマンドが発行されます。
- 書き込みステートのときにコントローラが READ コマンドを検出すると、write_to_read タイム後に READ コマンドが発行されます。同様に、読み出しステートのときにコマンド論理ブロックから WRITE コマンドが発行されると、read_to_write タイム後に WRITE コマンドが発行されます。読み出しまたは書き込みステートでは、コントローラは書き込みイネーブルをアドレス FIFO にアサートし、次のアドレスを取得します。
- コマンドは、DDR2 メモリに対して発行される前に、パイプライン化されてアドレス信号に同期化されます。

リファレンス デザイン

図 8 に、最上位モジュール mem_interface_top 以下のデザイン階層を示します。

DDR2 SDRAM インターフェイスのリファレンス デザインは MIG ツールと統合されており、MIG ツールはザイリンクス CORE Generator™ ソフトウェアと統合されています。最新バージョンのデザインは、次のサイトから ISE IP アップデートをダウンロードすることにより入手できます。

http://japan.xilinx.com/xlnx/xil_sw_updates_home.jsp



x723_08_091505

図 8： デザイン階層

リファレンス デザインのサマリ

表 9 に、72 ビット インターフェイスの最大周波数をスピード グレード別に示します。

表 9: 72 ビット インターフェイスの最大周波数

スピード グレード	最大周波数 (MHz)
-10	230
-11	267
-12	300

表 10 に、72 ビット インターフェイスのリファレンス デザインのサマリを示します。

表 10: 72 ビット インターフェイスのリファレンス デザインのサマリ

パラメータ	詳細/メモ
デバイスの使用量	スライス 6714 個 (コントローラ、合成可能なテストベンチ、ユーザー インターフェイス、物理層を含む)
	BUFG 6 個 (IDELAY ブロック用の 200MHz のリファレンス クロックに使用される BUFG 1 つを含む)
	BUFIO 9 個 (インターフェイス内のストロブ数と同じ)
	DCM 1 個
	PMCD 1 個
	ISERDES 72 個 (インターフェイス内のデータ ビット数と同じ)
	OSERDES 99 個 (データ ビット数、ストロブ数、およびデータ マスクビット数の合計と同じ)

まとめ

このアプリケーション ノートで説明されている DDR2 コントローラは、アプリケーション ノート [XAPP721](#) 『ISERDES と OSERDES を使用した高性能 DDR2 SDRAM インターフェイスのデータ キャプチャ』で説明されているデータ キャプチャの手法に加え、パフォーマンスの高いメモリ インターフェイスのすぐれたソリューションを提供します。

このデザインでは、FPGA 内のキャリブレーション ロジックを除くすべてのロジックがインターフェイスの半周波数で駆動され、クリティカル パスが削減されるため、高いマージンを達成できます。このデザインはハードウェアでも検証済みです。

改訂履歴

次の表に、この文書の改訂履歴を示します。

日付	バージョン	改訂
2005 年 12 月 15 日	1.0	初版リリース
2005 年 12 月 16 日	1.1	表 8 および 表 10 を改訂。
2006 年 02 月 02 日	1.2	図 4 を改訂。
2006 年 02 月 08 日	1.3	図 4 を改訂。

日付	バージョン	改訂
2007 年 10 月 17 日	1.4	<ul style="list-style-type: none">• 「はじめに」を改訂• 表 4 を改訂• 表 5 の後に説明を追加• 「デザイン階層」を「リファレンス デザイン」に変更し、内容を改訂• 「使用可能なリソース」を「リファレンス デザインのサマリ」に変更• 表 9 「72 ビット インターフェイスの最大周波数」を追加• 表 10 「使用可能なリソース」の名称を「72 ビット インターフェイスのリファレンス デザインのサマリ」に変更し、改訂• 「まとめ」を改訂