



XAPP454 (v1.1.1) 2007 年 6 月 11 日

Spartan-3 FPGA の DDR2 SDRAM メモリ インターフェイス

著者 : Karthikeyan Palanisamy

概要

このアプリケーション ノートでは、Spartan™-3 デバイスを DDR2 SDRAM メモリである Micron DDR2 SDRAM デバイスとインターフェイスさせる際のインプリメンテーションについて説明します。ここでは、DDR2 SDRAM デバイスの機能について簡潔に解説した後、DDR2 SDRAM メモリ インターフェイス インプリメンテーションについて詳細に説明します。

DDR2 SDRAM デバイスの概要

DDR2 SDRAM デバイスは、次世代 DDR SDRAM デバイスです。DDR2 SDRAM メモリ インターフェイスはソース同期であり、DDR SDRAM メモリと同様にダブル データ レートをサポートします。また、DDR2 SDRAM デバイスでは SSTL 1.8V I/O 規格を使用します。

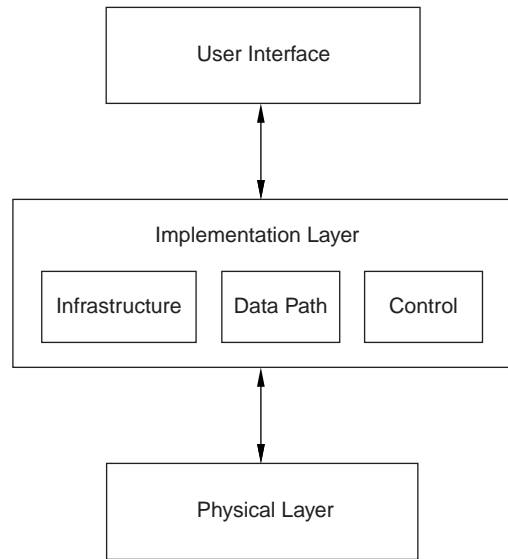
DDR2 SDRAM デバイスは、DDR SDRAM アーキテクチャを使用して高速処理を実現します。このメモリは、コントローラからの差動クロックを使用して動作します。(ウェブ上のリファレンス デザインは差動ストロブをサポートしていませんが、今後、サポートが追加される予定です。) コマンドは、クロックの立ち上がりエッジごとにラッチされます。双方向データ ストロブ (DQS) は、レシーバでのデータ取り込み用にデータと共に送信されます。DQS は、読み出し中は DDR2 SDRAM デバイスによって送信され、書き込み中はコントローラによって送信されるストロブです。また、DQS は、読み出しの場合はデータと共にエッジに同期し、書き込みの場合は中央にアラインされます。

DDR2 SDRAM デバイスへの読み出しアクセスおよび書き込みアクセスは、バースト方式です。アクセスは、アクティブ コマンドがレジスタに入ると開始し、読み出しコマンドおよび書き込みコマンドがそれに続きます。アクティブ コマンドと同期してレジスタで受信されたアドレス ビットは、アクセスするバンクおよび行の選択に使用されます。一方、読み出しまたは書き込みコマンドと共にレジスタに入力されたアドレス ビットは、バースト アクセスのバンクおよび列の開始位置の選択に使用されます。

インターフェイス モデル

DDR2 SDRAM メモリ インターフェイスは、デザインを簡潔化し、かつモジュール形式にするため階層になっています。図 1 に、メモリ インターフェイスの階層を示します。インターフェイスは、アプリケーション レイヤ、インプリメンテーション レイヤ、および物理レイヤの 3 つで構成されています。

© 2004-2007 Xilinx, Inc. All rights reserved. すべての Xilinx の商標、登録商標、特許、免責事項は、<http://japan.xilinx.com/legal.htm> にリストされています。他のすべての商標および登録商標は、それぞれの所有者が所有しています。すべての仕様は通知なしに変更される可能性があります。
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、および市場性や特定の目的に対する適合性についての黙示的な保証も含まれます。

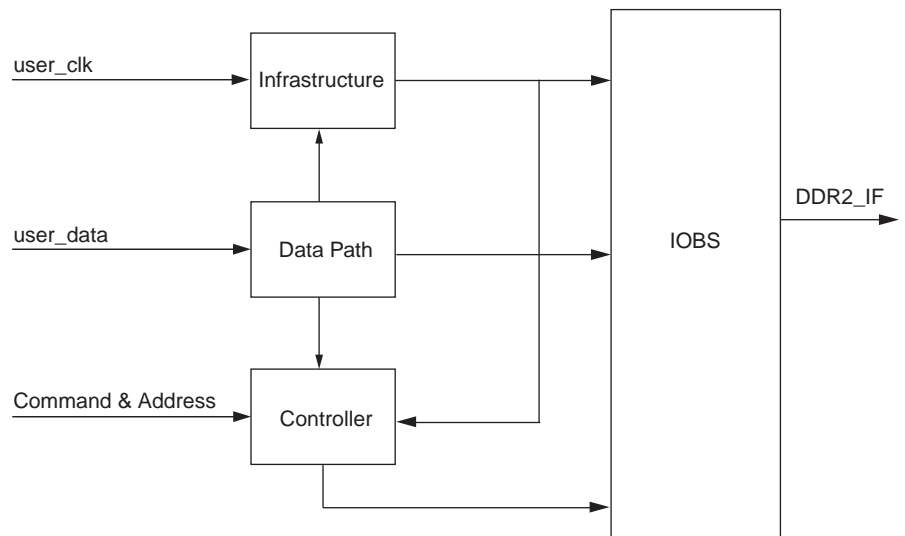


xapp549_02_113004

図 1: インターフェイス階層モデル

DDR2 SDRAM コントローラ モジュール

図 2 に、Spartan-3 DDR2 SDRAM メモリ インターフェイスのブロック図を示します。ここに示す 4 ブロックはすべて、ddr2_top モジュールのサブ ブロックです。次に、各ブロックの機能について説明します。



xapp549_03_113004

図 2: DDR2 SDRAM メモリ インターフェイス モジュール

コントローラ

コントローラは、4 バースト長と 3 および 4 の CAS レイテンシをサポートします。コントローラは、Load Mode コマンドの実行中に EMR(2) および EMR(3) レジスタを初期化し、差動データ ストローブを生成します。

コントローラは、ユーザー コマンドを受信し、デコードした後で、DDR2 SDRAM メモリへの読み出し、書き込み、およびリフレッシュ コマンドを生成します。さらに、その他のモジュール用の信号も生成します。

データ パス

データ パス モジュールは、メモリとのデータ送受信の役割を果たします。主な機能は次のとおりです。

- メモリへのデータ書き込み
- メモリからのデータ読み出し
- メモリ クロック ドメインから FPGA クロック ドメインへの読み出しデータの転送

書き込みデータおよび読み出しデータのキャプチャ手法については、XAPP768c『Spartan-3 デバイスと 166 MHz または 333 Mb/s DDR SDRAM メモリのインターフェイス』を参照してください。書き込みデータおよびストローブは FPGA からクロック出力されます。ストローブは、データの書き込み中にそのデータの中央にアラインされます。DDR2 SDRAM メモリの場合、ストローブはフリー ランニングではありません。ストローブを中央にアラインさせるには、メモリに入力される基準クロックを 90° および 270° シフトさせたクロックで書き込みデータを出力します。データ ストローブは、メモリに入力される基準クロックから生成されます。

メモリ読み出しデータは、ソース同期クロックに対してエッジに同期されます。DDR2 SDRAM クロックは、フリー ランニング ストローブではありません。データはそのストローブを使用して受信され、FPGA クロック ドメインに転送されます。データの入力側では、ストローブの入力側と類似したリソースが使用されます。これにより、ストローブ遅延回路内で遅延が追加されるまで、データおよびストローブ信号の遅延は確実に一致します。

インフラストラクチャ

インフラストラクチャ モジュールは、FPGA のクロックおよびリセット信号を生成します。クロックおよびその反転バージョンは、デジタル クロック マネジャー (DCM) を使用して生成されます。また、このモジュールには、遅延キャリブレーション回路がインプリメントされています。

遅延キャリブレーション回路は、読み出しデータに対してストローブ ラインを遅延させるための遅延エレメント数の選択に使用されます。遅延キャリブレーション回路は、すべての点でストローブ遅延回路と一致する回路の遅延を算出します。キャリブレーションでは、すべてのコンポーネントおよび配線遅延を含む、あらゆる遅延要素が考慮され、このキャリブレーション回路が任意の時間に対応する遅延エレメント数を選択します。キャリブレーションの実行後、遅延回路のセレクト ラインがアサートされず。遅延キャリブレーションについては、XAPP768c を参照してください。

IOBS

FPGA のすべての入力信号および出力信号は、IOBS モジュールにインプリメントされます。また、すべてのアドレスおよび制御信号はレジスタを介し、IOBS モジュールに入力またはそのモジュールから出力されます。

ユーザー インターフェイス信号 表 1 に、ユーザー インターフェイス信号を示します。ここに示す信号方向はすべて、DDR2 SDRAM コントローラに対するものです。

表 1: ユーザー インターフェイス信号

信号名	方向	説明
dip1	入力	DDR2 SDRAM のクロック イネーブル信号 (アクティブ Low)
rst_dqs_div_in	入力	DDR2 SDRAM メモリの読み出し中に dqs_div flop をイネーブル
reset_in	入力	システム リセット
user_input_data[(2n-1):0]	入力	DDR2 SDRAM の書き込みデータであり、n はメモリ インターフェイスの幅
user_input_address[addwidth:0]	入力	DDR2 SDRAM 列および行のアドレス
user_bank_address[bankaddwidth:0]	入力	DDR2 SDRAM バンク アドレス
user_config_reg1[14:0]	入力	DDR2 SDRAM コンフィギュレーション データ レジスタ 1
user_config_reg2[12:0]	入力	DDR2 SDRAM コンフィギュレーション データ レジスタ 2
user_command_reg[3:0]	入力	DDR2 SDRAM コントローラのユーザー コマンド レジスタ
burst_done	入力	バースト データ転送の Done 信号
rst_dqs_div_out	出力	外部から rst_dqs_div_in に接続する信号で、dqs_div flop をイネーブル
user_output_data[(2n-1):0]	出力	DDR2 SDRAM からの読み出しデータ
user_data_valid	出力	DDR2 SDRAM メモリからの読み出しデータが有効であることを示す (アクティブ Low)
user_cmd_ack	出力	user_command の受信応答信号
user_ODT_ack	出力	ODT コマンドの受信応答信号
init_val	出力	DDR2 SDRAM が初期化されたことを示す
ar_done	出力	DDR2 SDRAM に auto-refresh コマンドが送信されたことを示す
clk_int	入力	DDR2 SDRAM コントローラにより生成されたクロック
clk90_int	入力	DDR2 SDRAM コントローラにより生成された 90° 位相シフト クロック
sys_rst	入力	システム リセット入力で生成される信号
sys_rst90	入力	システム リセット入力を 90° 位相シフトさせたリセット信号
sys_rst180	入力	システム リセット入力を 180° 位相シフトさせたリセット信号
sys_rst270	入力	システム リセット入力を 270° 位相シフトさせたリセット信号

メモ :

1. すべての信号方向は、DDR2 SDRAM コントローラに対するものです。

信号について

user_input_data[(2n-1):0]

ユーザー インターフェイスから DDR2 SDRAM への書き込みデータです。このデータは、DDR2 SDRAM 書き込みコマンドで有効になり、n は DDR2 SDRAM メモリの幅を示します。また、DDR2 SDRAM コントローラは、物理レイヤ側でシングル データ レートをダブル データ レートに変換します。

user_input_address[addwidth:0]

DDR2 SDRAM 書き込みおよび読み出しの行列アドレスの合計です。アドレス幅の選択によって、user_input_address は列および行のアドレス ビットに分割されます。

user_bank_address[bankaddwidth:0]

DDR2 SDRAM のバンク アドレスであり、変数によって、バンク アドレスを選択できます。

user_config_reg1[14:0]

DDR2 SDRAM メモリ初期化のコンフィギュレーション データです。このレジスタの内容は、Load Mode コマンドの実行中にモード レジスタに取り込まれます。user_config_reg1 のフォーマットは次のとおりです。

14	13	11	10	9	7	6	4	3	2	0
PD	WR	TM	Res			Cas_latency	BT	Burst_length		

Burst_length[2:0]

コントローラでは、4 バースト長のみサポートされます。

BT

バースト タイプを選択します。コントローラでは連続したバーストのみがサポートされ、このビットは常にゼロに設定されます。

Cas_latency [6:4]

ビット [6:4] は CAS レイテンシを選択します。DDR2 SDRAM コントローラは、3 または 4 の CAS レイテンシをサポートします。

Res [9:7]

ビット [9:7] は、後のインプリメンテーションで使用されます。

TM

Load Mode レジスタの TM ビットに取り込まれます。

WR [13:11]

Load Mode レジスタの WR (Write Recovery) ビットに書き込まれます。

PD

Load Mode レジスタの PD (Power Down Mode) ビットに書き込まれます。

Load Mode レジスタについては、Micron DDR2 SDRAM データシートを参照してください。

user_config_reg2[12:0]

Extended Mode レジスタの DDR2 SDRAM コンフィギュレーション データであり、user_config_reg2 のフォーマットは次のとおりです。

12	11	10	9	7 6	4 3	2	1	0
OUT	RDQS	DQS	OCD	Posted CAS	RTT	ODS	Res	

Extended Mode レジスタについては、Micron DDR2 SDRAM データシートを参照してください。

user_command_reg[3:0]

ユーザー コマンド レジスタです。さまざまなコマンドがこのレジスタを通して、DDR2 SDRAM モジュールに受け渡されます。表 2 にサポートされているコマンドを示します。

表 2: ユーザー コマンド

user_command_reg[3:0]	ユーザー コマンドの説明
0000	無演算命令
0010	メモリ (DDR2 SDRAM) 初期化
0011	自動リフレッシュ
0100	書き込み
0101	ロード モード (Load Mode の場合のみ)
0110	読み出し
その他	予約済み

burst_done

データ送信の最後に、この信号を 2 クロック周期間イネーブルにする必要があります。DDR2 SDRAM コントローラは、1 列で書き込みバーストまたは読み出しバーストをサポートします。したがって、1 行のバウンダリで処理を終了し、次の列の処理を行うために 1 行のバウンダリで再初期化する必要があります。コントローラは、DDR2 SDRAM メモリにプリチャージ コマンドを入力することによって、書き込みバーストまたは読み出しバーストを終了します。

user_output_data[(2n-1):0]

DDR2 SDRAM メモリからの読み出しデータです。DDR2 SDRAM コントローラは、DDR2 SDRAM メモリから読み出された DDR SDRAM データを SDR データに変換します。このように変換される場合、バス幅は 2n であり、n は DDR2 SDRAM メモリのデータ幅を示します。

user_data_valid

この信号がアサートされると、user_output_data[(2n-1):0] 信号が有効になります。

user_cmd_ack

ユーザー読み出しコマンドまたは書き込みコマンドに対する応答信号であり、DDR2 SDRAM との読み出しまたは書き込み中に、DDR2 SDRAM コントローラによってアサートされます。この信号がデアサートされるまで、コントローラに新しいコマンドを入力できません。

init_val

DDR2 SDRAM の初期化完了後、DDR2 SDRAM コントローラによってアサートされます。

ar_done

自動リフレッシュ コマンドが DDR2 SDRAM に入力されると、DDR2 SDRAM コントローラによって 1 クロック サイクル間アサートされます。

メモ：出力クロックおよびリセット信号は、データの同期化に使用できます。

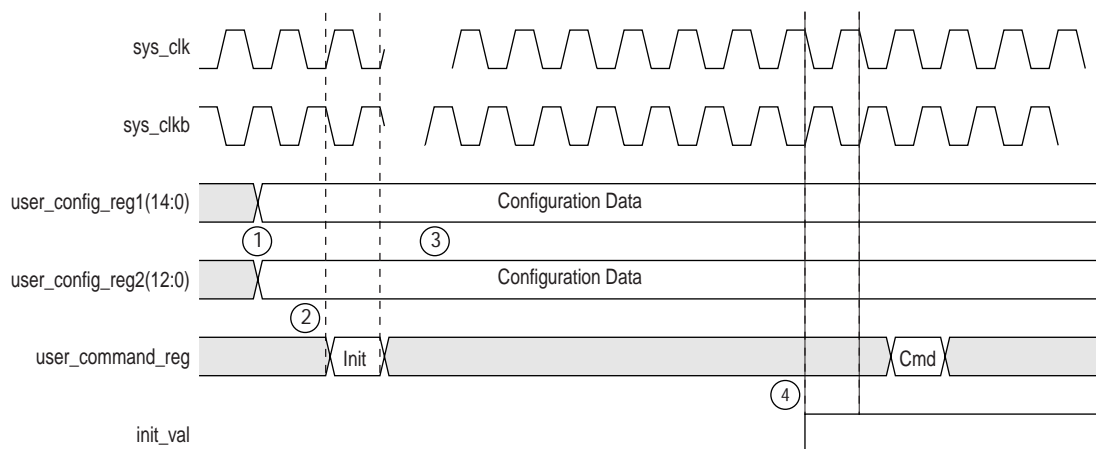
表 3 に、メモリ インターフェイス信号を示します。

表 3：メモリ インターフェイス信号

信号名	方向	説明
ddr_dq[(datawidth-1):0]	入出力	双方向 DDR2 SDRAM メモリ データ
ddr_dqs[(dqswidth-1):0]	入出力	双方向 DDR2 SDRAM メモリ データ ストローブ信号であり、ストローブ信号数はデータ幅およびデータ比に対するストローブによって異なる
ddr_cke	出力	DDR2 SDRAM メモリのクロック イネーブル信号
ddr_csb	出力	アクティブ Low のチップ セレクト信号
ddr_rasb	出力	アクティブ Low のメモリ列アドレス ストローブ
ddr_casb	出力	アクティブ Low のメモリ行アドレス ストローブ
ddr_web	出力	アクティブ Low のメモリ書き込みイネーブル信号
ddr_dm	出力	メモリ データ マスク信号
ddr_ba	出力	メモリ バンク アドレス
ddr_address	出力	メモリ アドレス (行列の両アドレス)
ddr2_clk*	出力	メモリ差動クロック信号
ddr_odt[4:0]	出力	メモリのダイ上の終端信号

DDR2 SDRAM メモリの初期化

読み出しコマンドおよび書き込みコマンドの入力前に、メモリ初期化コマンドを使用して、DDR2 SDRAM メモリを初期化する必要があります。DDR2 SDRAM の初期化が完了するまで、Mode Register および Extended Mode レジスタに書き込まれたデータは user_config_reg1 [14:0] および user_config_reg2 [12:0] になければいけません。DDR2 SDRAM が初期化されると、DDR2 SDRAM コントローラによって、init_val 信号がアサートされます。図 3 に、メモリ初期化コマンドのタイミング図を示します。



xapp549_09_120804

図 3：DDR2 SDRAM メモリの初期化

1. `command_reg [2:0]` に初期化コマンドを入力する前に 2 クロック サイクル必要であり、`user_config_reg1[14:0]` および `user_config_reg2[12:0]` に有効なコンフィギュレーション データを送信します。
2. `command_reg [2:0]` への初期化コマンドの入力によって、初期化シーケンスが開始します。
3. `user_config_reg1[14:0]` および `user_config_reg2[12:0]` 上のデータは、後続するメモリ処理によって変更されません。
4. コントローラが、`init_val` 信号をアサートし、コンフィギュレーションが完了したことを示します。

DDR2 SDRAM メモリへの書き込み

図 4 に、バースト長が 4 の場合の DDR2 SDRAM メモリへの書き込みタイミング図を示します。波形は、2 つの連続するバーストを示しています。メモリへの書き込みの前に、DDR2 SDRAM コントローラへの 1 書き込みコマンドが先行します。DDR2 SDRAM コントローラは、書き込みコマンドに応じて、`SYS_CLKb` の立ち上がりエッジで `user_cmd_ack` 信号をアサートします。ユーザー コマンドの応答信号を待ってから、次の処理を実行してください。

`user_cmd_ack` 信号がアサートされてから 2.5 クロック サイクル後に、メモリ バースト アドレスが `user_input_address[addwidth:0]` ラインに入ります。`user_input_address` は、`SYS_CLK` の立ち上がりエッジでアサートされる必要があります。メモリへの書き込みデータは、`clk90_int_val` でアサートされ、メモリ アドレスが `user_input_address` に入る前に、コントローラに入力されなければなりません。ここでのユーザー データ幅は、メモリ データ幅の 2 倍です。コントローラは、データをメモリに受け渡す前にダブル データ レートに変換します。

バースト長が 4 の場合、`user_input_data[(2n-1):0]` データの 2 つの部分は、各ユーザー アドレスで DDR2 SDRAM コントローラに入力されます。書き込みバーストを終了するには、`burst_done` 信号を `SYS_CLK` の立ち上がりエッジで、2 クロック間アサートします。`burst_done` 信号は、最後のメモリ アドレスで 2 クロック間アサートされる必要があります。その後の DDR2 SDRAM へのコマンド送信は、`user_cmd_ack` 信号のディアサート後にのみ行ってください。

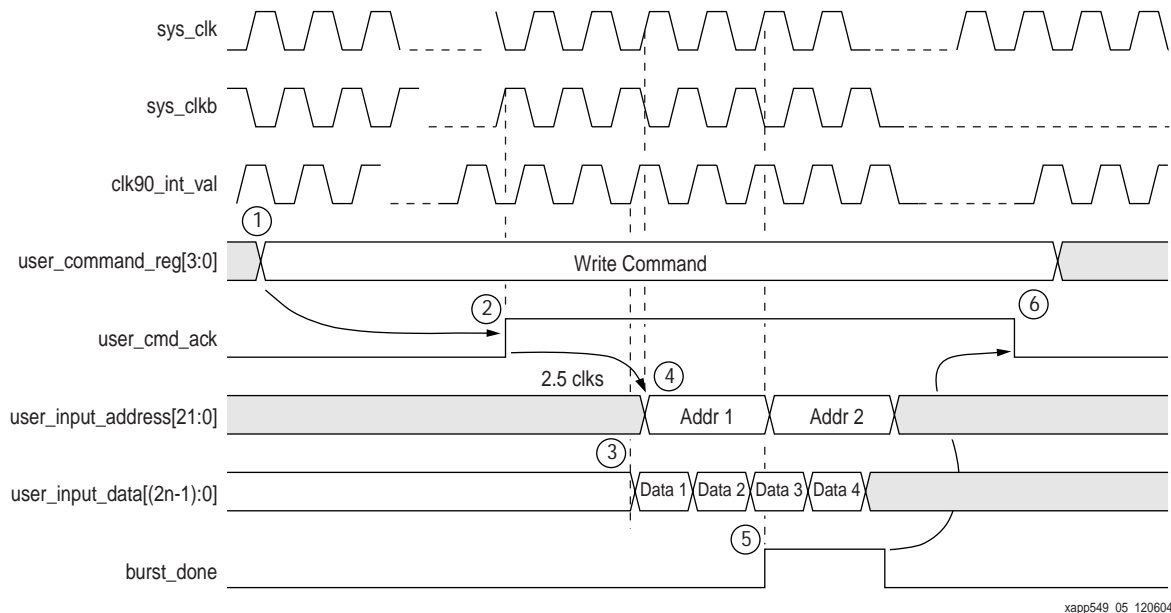


図 4：バースト長が 4 の場合の DDR2 SDRAM メモリへの書き込み

1. DDR2 SDRAM コントローラへ書き込みコマンドを与えることによって、メモリへの書き込みを開始します。書き込みコマンドは、`SYS_CLK` の立ち上がりエッジでアサートされなければなりません。
2. DDR2 SDRAM コントローラは、`SYS_CLKb` の立ち上がりエッジで `user_cmd_ack` 信号をアサートすることによって、書き込みコマンドに応答します。

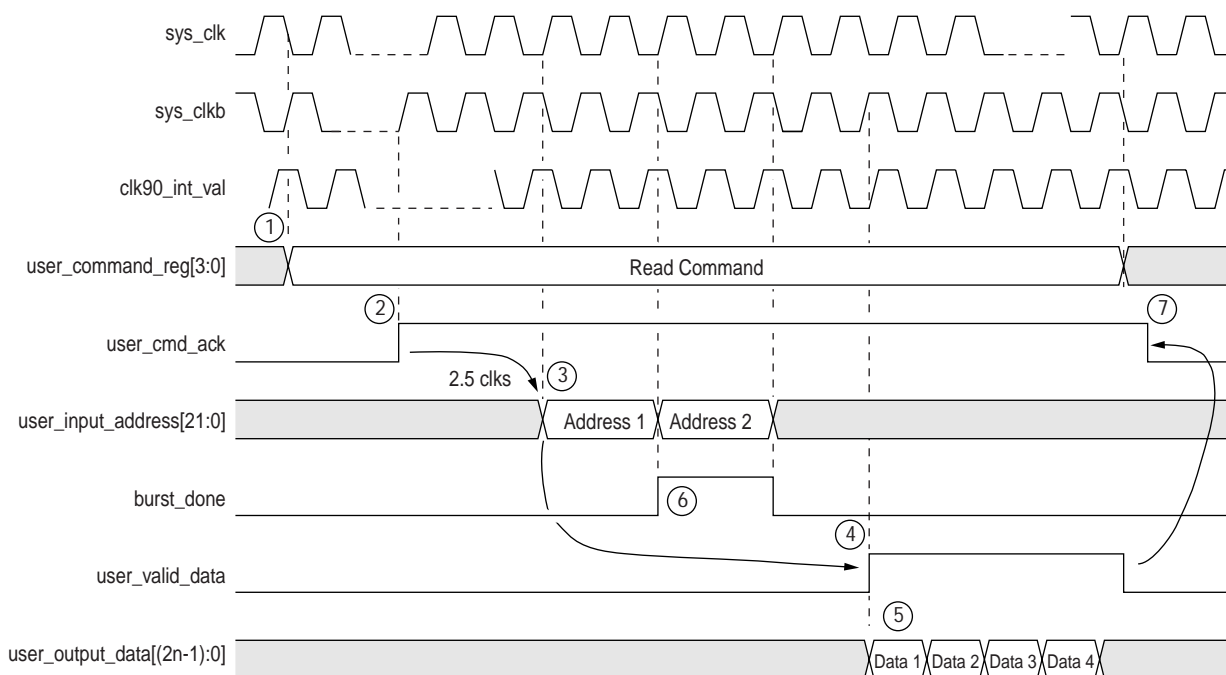
3. メモリアドレスを `user_input_address` に入れる前に、メモリへの書き込みデータを `user_input_data` に入れてください。入力データは、`clk90_int_val` signal でアサートされます。
4. `user_cmd_ack` 信号のアサートの 2.5 クロック サイクル後、`user_input address [21:0]` にメモリアドレスを入れてください。`user_input_address` 信号は、`SYS_CLK` の立ち上がりエッジでアサートされなければなりません。
5. 書き込みバーストを終了するため、最後の `last user_input_address` の 2 クロック間、`burst_done` 信号をアサートしてください。
6. その後の DDR2 SDRAM へのコマンド実行は、`user_cmd_ack` 信号のディアサート後にのみ行ってください。

DDR2 SDRAM メモリからの読み出し

図 5 に、バースト長が 4 の場合における、2 つの連続したバーストのメモリからの読み出しタイミング図を示します。メモリからの読み出しは、DDR2 SDRAM コントローラに読み出しコマンドを送信することによって開始します。

読み出しコマンドのフローは、書き込みコマンドの場合と類似しています。読み出しコマンドは、`SYS_CLK` の立ち上がりエッジでアサートされます。DDR2 SDRAM コントローラは、読み出しコマンドに応じて、`SYS_CLKb` の立ち上がりエッジで `user_cmd_ack` 信号をアサートします。`user_cmd_ack` の 2.5 クロック サイクル後、メモリ バースト読み出しアドレスが `user_input_address[addwidth:0]` に入り、`user_input_address` 信号が、`SYS_CLK` の立ち上がりエッジでアサートされます。

DDR2 SDRAM メモリからのデータは、`clk90_int_val` でアサートされる `user_output_data` で読み出すことができます。`user_output_data` 上のデータは、`user_data_valid` 信号がアサートされている場合のみ有効です。DDR SDRAM データは SDR データに変換されるため、このバス幅は $2n$ であり、 n は DDR2 SDRAM メモリのデータ幅を示します。読み出しバースト長が 4 の場合、DDR2 SDRAM コントローラは、各ユーザー アドレスで 2 データのみ出力し、それぞれが DDR2 SDRAM メモリの $2n$ 幅です。読み出しバーストを終了するには、`burst_done` 信号を `SYS_CLK` の立ち上がりエッジで、2 クロック サイクル間アサートします。`burst_done` 信号は、最後のメモリ アドレス後にアサートされる必要があります。その後の DDR2 SDRAM へのコマンド送信は、`user_cmd_ack` 信号のディアサート後にのみ行ってください。



XAPP549_07_120604

図 5: バースト長が 4 の場合の DDR2 SDRAM メモリからの読み出し

読み出しコマンド フローは、書き込みコマンドの場合と類似しています。

1. 読み出しコマンドを入力すると、このコマンドが SYS_CLK の立ち上がりエッジでアサートされます。
2. DDR2 SDRAM コントローラは、読み出しコマンドに応じて、SYS_CLKb の立ち上がりエッジで user_cmd_ack 信号をアサートします。
3. user_cmd_ack 信号のアサートの 2.5 クロック サイクル後、user_input_address [21:0] にメモリ アドレスを入れてください。次に、user_input_address 信号が、SYS_CLK の立ち上がりエッジでアサートされます。
4. user_output_data 上のデータは、user_data_valid 信号がアサートされた場合のみ有効です。
5. DDR2 SDRAM メモリからのデータ読み出しは、user_output_data 上で取得できます。この user_output_data は clk90_int_val でアサートされます。DDR SDRAM データは SDR データに変換されるため、このバスの幅は 2n であり、n は DDR2 SDRAM メモリのデータ幅を示します。読み出しバースト長が 4 の場合、DDR2 SDRAM コントローラは、各データアドレスで 2 データワードのみ出力します。
6. 読み出しバーストを終了するには、burst_done 信号を SYS_CLK の立ち上がりエッジで、2 クロック サイクル間アサートします。burst_done 信号は、最後のメモリ アドレスでアサートしなければなりません。
7. その後の DDR2 SDRAM コントローラへのコマンド送信は、user_cmd_ack 信号のデアサート後にのみ行ってください。

DDR2 SDRAM メモリの Auto_Refresh

DDR2 SDRAM コントローラは、それ自体にメモリ リフレッシュ機能がないため、定期的に auto_refresh コマンドを使用する必要があります。この auto_refresh コマンドは SYS_CLK でアサートされます。auto_refresh コマンドが完了すると、DDR2 SDRAM コントローラによって、SYS_CLKb で ar_done 信号がアサートされます。

物理レイヤおよび 遅延キャリブレーション

DDR2 SDRAM の物理レイヤは、アプリケーション ノート XAPP768c に記述されている DDR SDRAM の物理レイヤと類似しています。また、DDR2 SDRAM インターフェイスに XAPP768c で解説している遅延キャリブレーション テクニックを使用できます。

タイミングの算出

書き込みタイミング

表 4: 書き込みデータ

パラメータ	値 (ps)	クロックに対して 前にあるエッジの 不確定要素	クロックに対して 後にあるエッジの 不確定要素	説明
Tclock	6000			クロック周期
Tclock_phase	3000			クロック位相
Tdcd	250			クロックからメモリのデューティ サイクルのずれ
Tdata_period	2750			総データ周期であり、Tclock_phase-Tdcd
Tclock_skew	50	50	50	右または左側が使用されているため、ビットは互いに接近し、スキューは小さい
Tpackage_skew	90	90	90	パッケージ ピンおよびボード配置によるスキュー (配置を密接にすることによって削減できる)
Tsetup	350	350	0	メモリに関するデータシートに記載されているセットアップ タイム

表 4：書き込みデータ (続き)

パラメータ	値 (ps)	クロックに対して 前にあるエッジの 不確定要素	クロックに対して 後にあるエッジの 不確定要素	説明
Thold	350	0	350	メモリに関するデータシートに記載されているホールド タイム
Tphase_offset_error	140	140	140	同一 DCM からの差動クロック間のオフセット エラー
Tjitter	0	0	0	同一 DCM を使用してクロックおよびデータを生成して いるため、ジッタは考慮しない
Total uncertainties	980	630	630	ワースト ケースが同時に起こることはない
Window	1490	630	2120	総ウィンドウのワースト ケースは 1490ps

読み出しタイミング

表 5：読み出しデータ

パラメータ	値 (ps)	クロックに対して 前にあるエッジの 不確定要素	クロックに対して 後にあるエッジの 不確定要素	説明
Tclock	6000			クロック周期
Tclock_phase	3000			クロック位相
Tclock_duty_cycle_dist	300	0	0	クロックからメモリのデューティ サイクルのずれ
Tdata_period	2700			総データ周期であり、clock_phase-Tdcd
Tdqsq	350	350	0	メモリに関するデータシートに記載されている ストロープとデータのずれ
Tpackage_skew	90	90	90	パッケージ スキューのワースト ケース
Tds	452	452	0	Spartan-3 データシートに記載されているスピード グレードが -5 のセットアップ タイム
Tdh	-35	0	-35	Spartan-3 データシートに記載されているスピード グレードが -5 のホールド タイム
Tjitter	100	0	0	データおよびストロープは、同一クロックから生成 されるためジッタは考慮しない
Tlocal_clock_line	20	20	20	ローカル クロック ライン スキューのワースト ケース
Tpcb_layout_skew	50	50	50	ボード上のデータ ラインおよびストロープ間の スキュー
Tqhs	450	0	450	データシートに記載されている DQ のホールド スキュー ファクタ
Total uncertainties		962	575	ワースト ケースが同時に起こることはない
Window for DQS position for normal case	1163	962	2125	ウィンドウのワースト ケースは 1163ps

メモ：

1. Tdqsq および Tqhs は、Micron データシートの MT47H64M4FT-37E、Rev C、05/04 EN を参照しています。
2. Spartan-3 のタイミング値は、-5 デバイスのスピード ファイル、バージョン 1.33 を参照しています。

アドレスおよびコマンド タイミング

表 6: アドレスおよびコマンド データ

パラメータ	値 (ps)	クロックに対して 前にあるエッジの 不確定要素	クロックに対して 後にあるエッジの 不確定要素	説明
Tclock	6000			クロック周期
Tclock_skew	50	50	50	右または左側が使用されているため、ビットは互いに接近し、スキューは小さい
Tpackage_skew	90	90	65	同一バンクを使用し、パッケージ スキューを削減
Tsetup	500	500	0	メモリに関するデータシートに記載されている セットアップ タイム
Thold	500	0	500	メモリに関するデータシートに記載されている ホールド タイム
Tphase_offset_error	140	140	140	クロックの異なる位相間のオフセット
Tduty_cycle_distortion	0	0	0	デューティ サイクルのずれは適用されない
Tjitter	0	0	0	クロックおよびアドレスが同一クロックを使用して 生成されるため、両ジッタは同一であるため考慮 しない
Total uncertainties		780	755	
Command window	3025	2220	5245	ウィンドウのワースト ケースは 3025ps

リファレンス デザイン

Spartan-3 の DDR2 SDRAM メモリ コントローラのリファレンス デザインは、ザイリンクス CORE Generator™ に統合された MIG (Memory Interface Generator) ツールに統合されています。最新バージョンは、次のザイリンクス ウェブ サイトから IP アップデートをダウンロードしてください。

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

参考資料

XAPP768 『Spartan-3 デバイスと 166 MHz または 333 Mb/s DDR SDRAM メモリのインターフェイス』

Micron データシート MT47H16M16FG-37E

<http://www.micron.com/products/dram/ddr2sdram/partlist.aspx?density=256Mb>

まとめ

Spartan-3 FPGA では、高パフォーマンスの DDR2 SDRAM メモリ インターフェイスをインプリメントできます。ここで使用したデザインは、シミュレーション、理論合成 (Synplicity を使用)、およびザイリンクスの Project Navigator フローが実行されています。

改訂履歴

次の表に、このドキュメントの改訂履歴を示します。

日付	バージョン	改訂内容
2004/12/06	1.0	初版リリース
2007/06/07	1.1	MIG ツールに関する記述を追加
2007/06/11	1.1.1	商標を追加