



XAPP701 (v1.3) 2005 年 9 月 13 日

ダイレクト クロッキングを使用したメモリ インターフェイスのデータ キャプチャ

概要

このアプリケーション ノートでは、Virtex-4 デバイスにおけるダイレクト クロッキングを使用したメモリ インターフェイスのデータ キャプチャについて説明します。ダイレクト クロッキングには、Virtex-4 特有のアーキテクチャ機能 (各 I/O ブロック (IOB) に含まれる 64 タップの絶対遅延ラインなど) が使用されます。

はじめに

メモリ インターフェイスのほとんどはソース同期インターフェイスで、外部メモリ デバイスから送信されるクロック/ストロブとデータのエッジが揃っています。送信データを Virtex-4 デバイスに取り込むには、このクロック/ストロブまたはデータのどちらかを遅らせる必要があります。ダイレクト クロッキング手法では、データの方を遅らせて、内部 FPGA クロックのエッジにデータの中央を揃えることができます。この場合、送信されたデータは内部 FPGA クロックで取り込みます。外部メモリから送信されるクロック/ストロブは、対応するデータビットの遅延値を決定するために使用されます。このため、ストロブに対するデータビット数に制限はありません。ストロブをデータビットに分配する必要がないので、これ以上のクロック リソースは必要ありません。

クロック/ストロブとデータビットで使用される Virtex-4 リソースは、64 タップの絶対遅延ラインです。この遅延ラインは、IDELAY および IDELAYCTRL プリミティブを使用してインプリメントできます。クロック/ストロブとデータビットのどちらもこのラインを使用して配線されます。ストロブはデータの取り込みには使用されませんが、データの中央を内部 FPGA クロックのエッジに揃えるのに必要なタップ数を決定するために使用されます。ダイレクト クロッキング手法を使用したデザインおよび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。

ストロブ エッジ の検出

クロック/ストロブに対するデータビットの遅延値は、内部 FPGA クロックの立ち上がりエッジとクロック/ストロブ パルス中央の間の位相差になります。クロック/ストロブとデータのエッジは揃っていると想定されます。この位相差を決定するため、クロック/ストロブは IOB の 64 タップの絶対遅延ラインから入力され、内部 FPGA クロックを使用してインクリメンタル タップ出力でサンプリングされます。

クロック/ストロブ パルスの中央を決定するには、最低でも 2 つのエッジまたはトランザクションを検出する必要があります。2 つ目の遷移を検出するのに必要なタップ数 (セカンド エッジ タップ) から最初の遷移を検出するのに必要なタップ数 (ファースト エッジ タップ) を減算した値が、クロック/ストロブのパルス幅です。この差の半分がパルスの中央 (パルス センタ タップ) になります。内部 FPGA クロックの立ち上がりエッジからクロック/ストロブ パルスの中央までに必要なタップ数は、ファースト エッジ タップとパルス センタ タップの合計になります。

© 2004-2005 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」、ザイリンクスのロゴ、およびザイリン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。

表 1 に、さまざまなタイプのタップを示します。

表 1: タップのタイプ

タップ	説明
ファースト エッジ タップ	最初のクロック/ストローブのトランザクションを検出するのに必要なタップ数
セカンド エッジ タップ	2 つ目のクロック/ストローブのトランザクションを検出するのに必要なタップ数
セカンド エッジ タップ - ファースト エッジ タップ	クロック/ストローブのパルス幅
パルス センタ タップ	クロック/ストローブのパルス幅を 2 で割った数
ファースト エッジ タップ + パルス センタ タップ	内部 FPGA クロックのエッジにデータの中央に揃えるために必要なタップ数 (データ遅延タップ)

図 1 は、データ遅延タップ値でデータを遅らせ、内部 FPGA クロック エッジにデータの中央を揃えるケースを 2 つ示しています。ケース 1 では、クロック/ストローブの立ち下がりエッジが最初のエッジとして検出され、データの中央は内部 FPGA クロックの立ち上がりエッジに揃えられます。ケース 2 では、

クロック/ストローブの立ち上がりエッジが最初のエッジとして検出され、データの中央は内部 FPGA クロックの立ち下がりエッジに揃えられます。

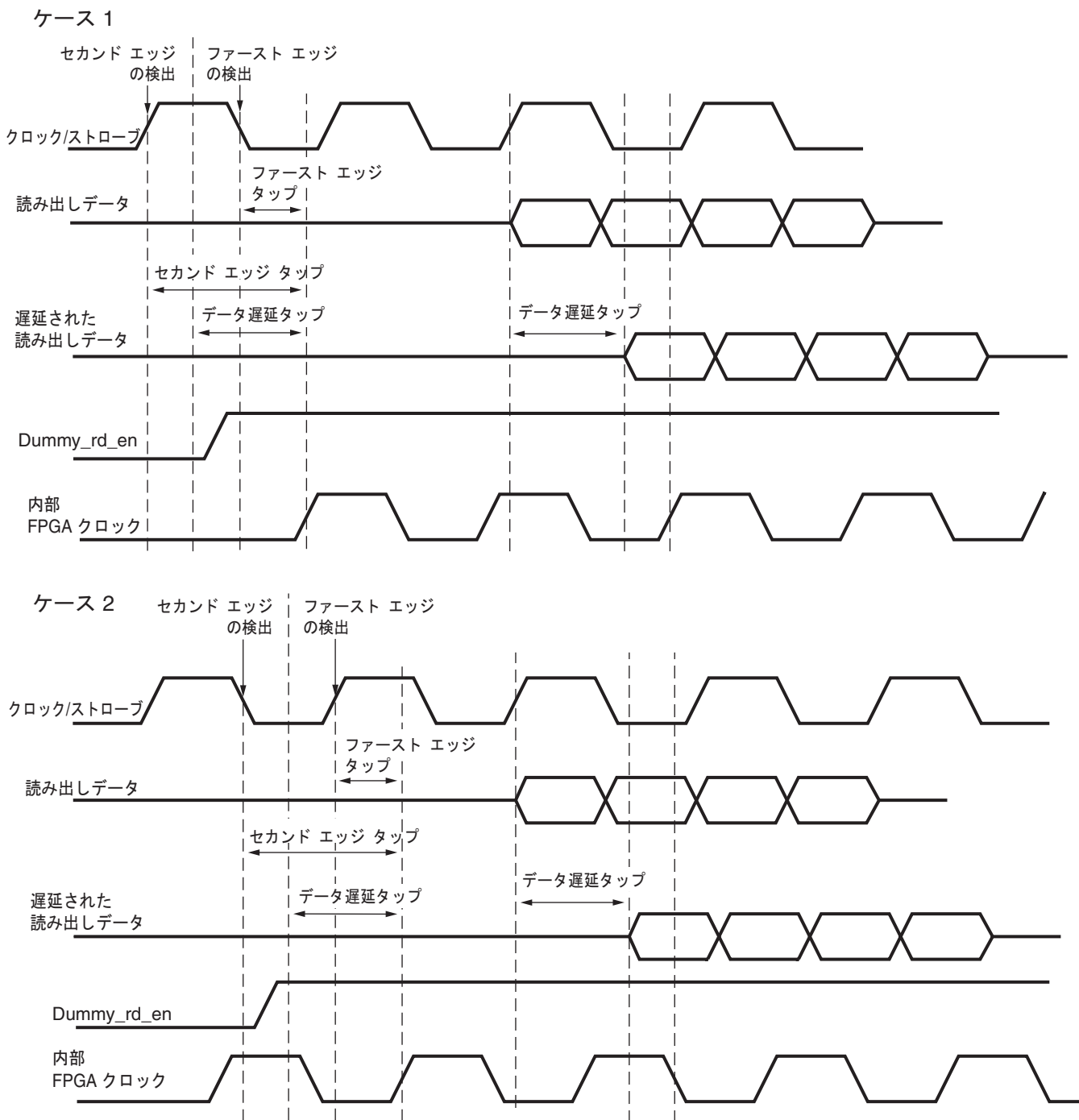


図 1: ケース 1 およびケース 2 - データの中央を内部 FPGA クロックに揃える

ストロブ エッジ 検出のインプリメ ンテーション

Virtex-4 デバイスの遅延値決定回路のインプリメンテーションは、IDELAY および IDELAY_CTR 回路を使用すると簡単です。図 2 は、遅延値決定回路をインプリメントするブロック図です。

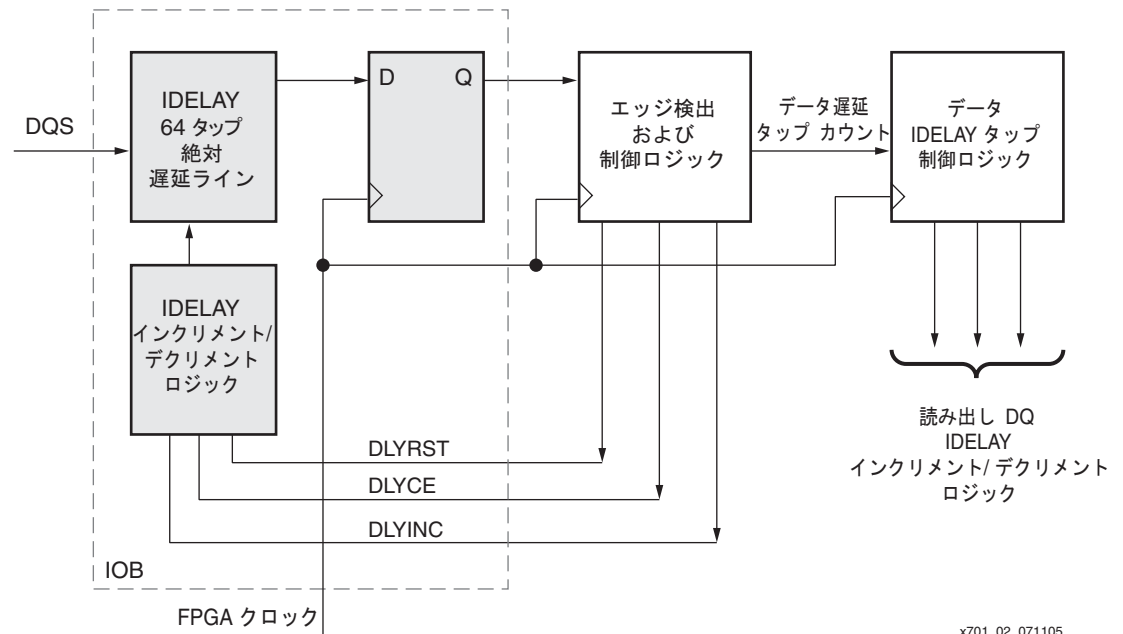


図 2：ストロブ エッジの検出

メモリのクロック/ストロブのエッジは、単純なアルゴリズムで検出できます。クロック/ストロブは、初期値 0 で IDELAY ブロックに入力され、最初のエッジが検出されるまで 1 タップずつ遅延が追加されます。最初のエッジが検出されると、必要となったタップ数が記録されます。その後クロック/ストロブには、2 つ目のエッジが検出されるまで続けて 1 タップずつ遅延が追加されます。2 つ目のエッジが検出されると、必要となったタップ数が記録されます。記録された 2 つの値を使用してパルス幅が計算されます。クロック/ストロブのパルス幅のタップ数が算出されると、それを 2 で割って中央地点が算出されます。中央地点と最初のエッジを検出するのに必要だったタップ数の合計値が、データを遅延させるのに必要なタップ数です。

IDELAY ブロックで使用可能なタップ数の合計は 64 です。このため、周波数が 200MHz 以下の場合、2 つのエッジを検出できません。64 タップの終わり迄で、検出されたエッジが 1 つだけだった場合、データを遅延させるのに必要なタップ数は、最初のエッジを検出するのに必要なタップ数に 16 タップ (約 80ps のタップ精度で約 1.25ns) を追加した値になります。これは、200MHz のクロック/ストロブの 1/4 サイクルが約 16 タップだからです。タイミング解析によると、この値を 110MHz までの周波数で使用できます。周波数が 110MHz 未満で 64 タップ終了してもエッジが検出されない場合は、データを遅延させるのに必要なタップ数は 32 (約 80ps のタップ精度で約 2.5ns) に設定されます。これは、内部 FPGA クロックのエッジをデータ有効ウィンドウ内に設定するには問題のない値です。

最初のエッジと 2 つ目のエッジを検出するには、小型のステート マシンが必要になります。このステート マシンが使用されるのは、データ遅延のタップ値を決定するためのダミー リードが動作している間のみです。複数の連続読み出しコマンドを含むダミー リードは、通常動作の前に外部メモリ デバイスに送信されます。ステート マシンでは、DLYRST、DLYCE、DLYINC などの IDELAY への入力が制御されます。

DLYRST - 遅延ラインのタップ数を IOBDELAY_VALUE 属性で設定された値にリセットする遅延ライン リセット信号。このデザインでは 0 に設定されています。

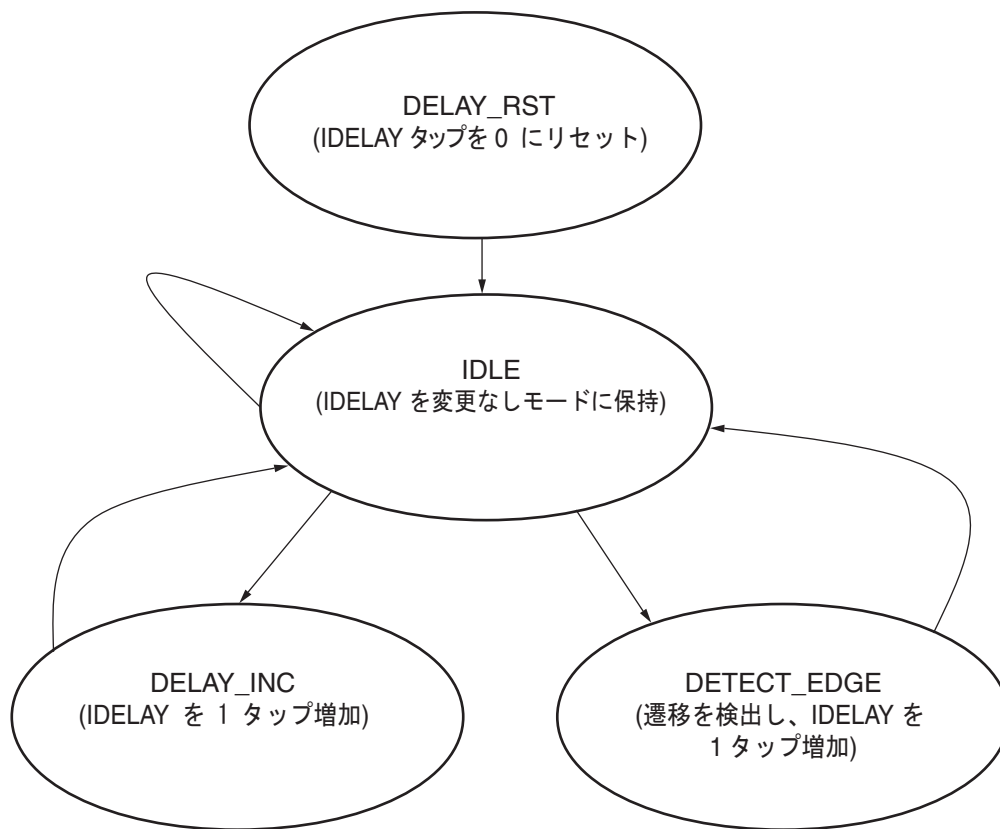
DLYCE - 遅延ライン インクリメント/デクリメント信号をアクティブにするタイミングを決定する遅延ライン イネーブル信号。

DLYINC - 遅延ブロックのタップ数を増加または削減するインクリメント/デクリメント信号。遅延ラインの動作は、表 2 を参照してください。

表 2：遅延ブロックの動作

動作	DLYRST	DLYCE	DLYINC
タップカウンタの設定値にリセット	1	X	X
タップカウンタの増加	0	1	1
タップカウンタの削減	0	1	0
変更なし	0	0	X

図 3 は、これらの遅延ブロックの入力を制御する状態 ダイアグラムです。この状態 マシンの状態は、DELAY_RST、IDLE、DELAY_INC、DETECT_EDGE の 4 つです。



x701_03_071105

図 3：ストローブ エッジを検出する状態 ダイアグラム

DELAY_RST

この状態は、ダミー リードの開始によってイネーブルされる状態 マシンの最初の状態です。この状態では、遅延ブロックは 0 タップにリセットされます。その後、IDLE ステートが複数回繰り返されます。

IDLE

この状態では、遅延ブロックは変更なしの状態に保持されます。IDLE 以外の各ステートの後、IDLE ステートが繰り返されます。これにより、タップ出力値が安定します。その後、IDLE ステートが繰り返されるか、DELAY_INC または DETECT_EDGE ステートに遷移します。

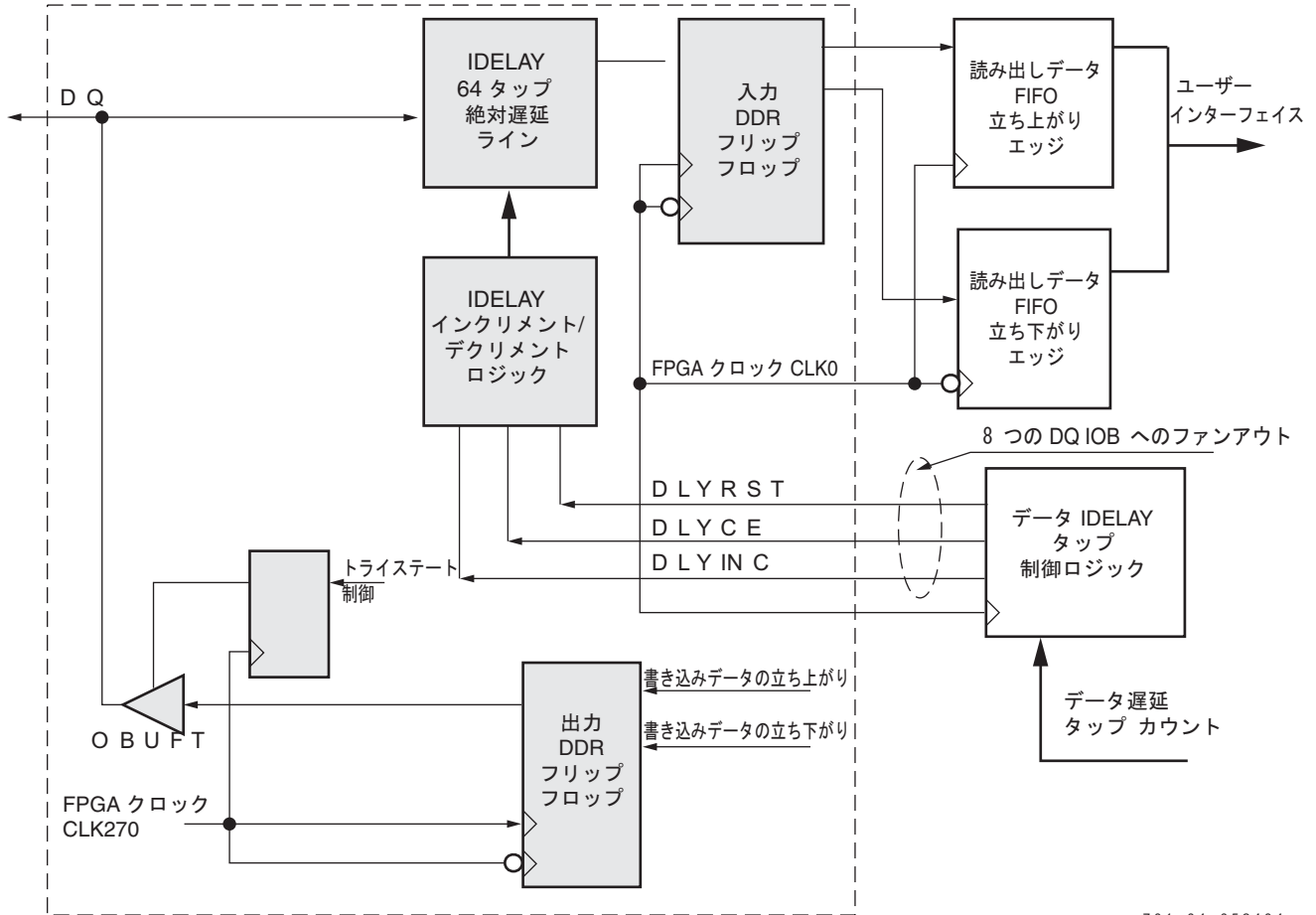
DELAY_INC

このステートは、遅延ブロックのタップを 1 つ増加させます。この後、IDLE ステートが複数回繰り返されます。

DETECT_EDGE

このステートでは、エッジまたは遷移を検出するため、遅延ブロックの出力がその前の値と比較され、遅延ブロックのタップが 1 つ増加されます。この後、IDLE ステートが複数回繰り返されます。

データを遅延させるタップ数が決まると、IDELAY 回路がイネーブルになり、この値までタップが増加されます。これは、必要なタップ数と同じクロック サイクル分 IDELAY 回路を増加することにより達成します。図 4 は、IDELAY 回路を含む読み出し/書き込みデータパスのブロック図を示しています。



x701_04_052404

図 4：読み出し/書き込みデータパス

データ キャプチャ と再キャプチャ

図 4 に示すように、遅延データは内部 FPGA クロックを使用して入力 DDR フリップフロップに取り込みます。これらのフリップフロップの出力は、2 つの FIFO (立ち上がりエッジのデータ用と立ち下がりエッジのデータ用) に格納されます。FIFO は、LUT RAM を使用してインプリメントされます。FIFO のライト イネーブル信号は、システム パラメータ用に正規化されたリード イネーブル信号から提供されます。また、リード イネーブル信号は、データ ビットと同じタップ遅延数の 64 タップ遅延ラインを通過します。

DDR2 SDRAM デバイスからは、読み出しデータと共にリード有効信号またはリード イネーブル信号が提供されないため、コントローラで CAS レンテンシおよびバースト長に基づいてリード イネーブル信号が生成されます。リード イネーブル信号は、リード プリアンブル中にアサートされ、ストロープの最後の立ち上がりエッジの後にデアサートされる必要があります。リード イネーブル信号は、システムへの依存性をなくすために正規化されます。この正規化プロセスは、PCB のループバックを伴ってインプリメントされます。FPGA から出力された READ_EN_OUT は PCB でループバックされ、READ_EN_IN という入力になります。このループバックのトレース遅延は、メモリ デバイスに転送されるクロックのトレース遅延 (CK/CK) とストロープ (DQS)/データ (DQ) の遅延を合計した値と同じである必要があります。CK、DQS、DQ のトレース遅延は、ほぼ同じにする必要があります。このループバック信号を使用して、読み出しデータを取り込む FIFO へのライト イネーブル信号が生成されます。複数のバンクを使用するインターフェイスの場合は、このイネーブル信号のファンアウトを制御するため、各バンクでループバックを使用することをお勧めします。

最初のデータワードは、内部 FPGA クロックの立ち上がりエッジまたは立ち下がりエッジのいずれかで取り込まれるため、立ち上がりエッジ FIFO のライト イネーブル用に別のロジックが必要になります。図 5 は、リード再キャプチャ用 FIFO のライト イネーブルロジックをインプリメントする回路を示しています。最初のデータが FIFO クロックの立ち上がりエッジで取り込まれる場合は、立ち上がりエッジ FIFO へのライト イネーブル信号が最初のフリップフロップの出力になります。取り込まれない場合は、2 つ目のフリップフロップの出力になります。図 6 は、再キャプチャ FIFO の読み出しデータの取り込みおよびライト イネーブルのタイミング図を示しています。

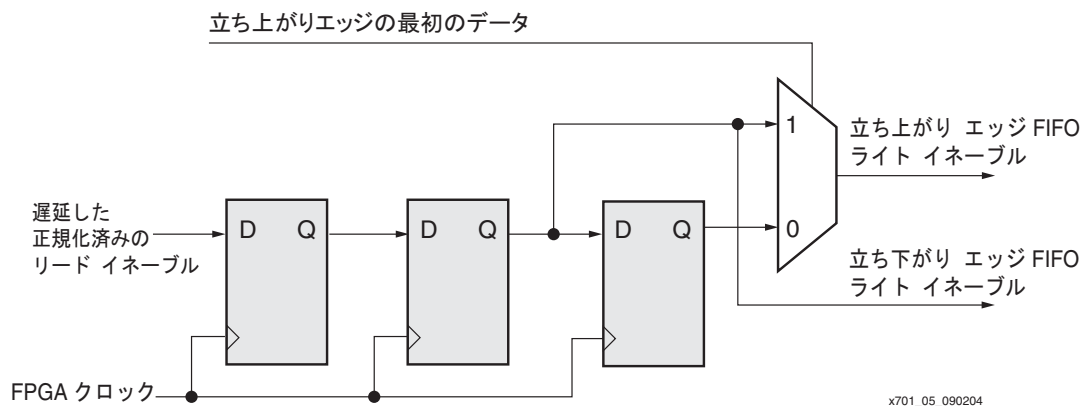
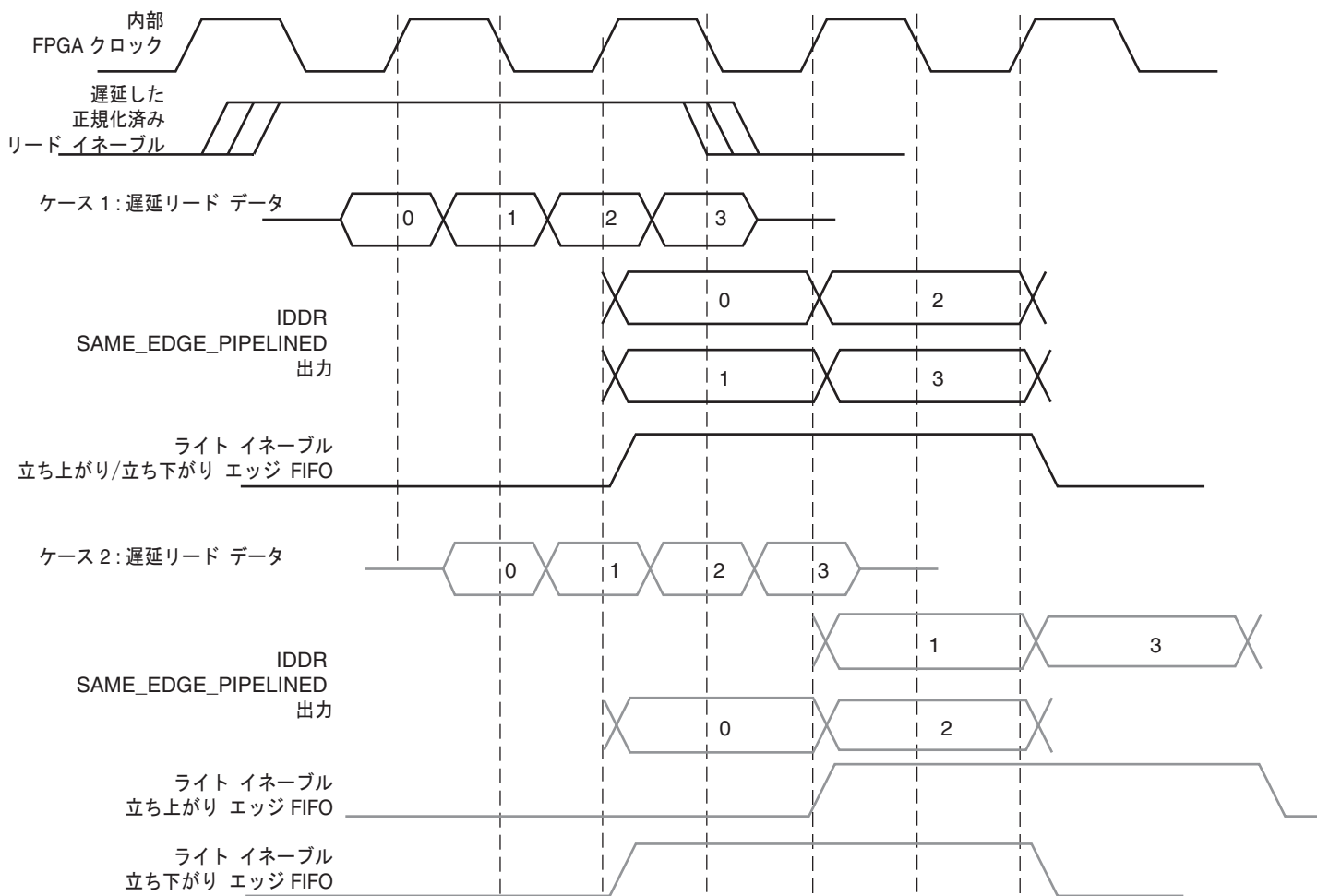


図 5: リード再キャプチャ FIFO のライト イネーブルロジック



x701_06_062005

図 6：データ キャプチャと FIFO への転送

読み出し タイミング解析

このセクションでは、ダイレクト クロッキングを使用した読み出しタイミング解析について説明します。読み出しデータは、FPGA クロック ドメインに直接取り込まれるため、データ有効ウィンドウ解析にはアクセス時間 (T_{AC}) というメモリ パラメータが使用されます。次は、このタイミング解析に使用される各パラメータの概要です。

このタイミング解析に使用される外部メモリ パラメータは、次のとおりです。

- T_{AC} - メモリに転送されたクロックに対する読み出しデータ (DQ) のアクセス時間
- T_{MEM_DCD} - メモリ ベンダの指定するデューティ サイクルの歪み耐性

読み出しデータ (DQ) は、メモリのクロック/ストロブ (DQS) ではなく FPGA のクロックを使用して取り込まれるため、この解析には T_{AC} (クロックに対するデータのアクセス時間) が使用されます。 T_{DQSQ} および T_{QHS} のような DQS から DQ へのメモリ パラメータは、 T_{AC} の値が優先されるので使用されません。

このタイミング解析に使用される FPGA パラメータは、次のとおりです。

- $T_{GLOBAL_CLOCK_TREE-SKEW}$ - グローバル クロック ツリーのスキュー
- T_{JITTER} - DCM クロック出力のジッタ
- $T_{PACKAGE_SKEW}$ - 特定デバイス/パッケージのパッケージ スキュー
- T_{SETUP} - I/O フリップフロップのセットアップ タイム
- T_{HOLD} - I/O フリップフロップのホールド タイム

DQS に対するデータ ビットの遅延は、DQS のエッジが検出されると計算されます。DQS エッジは、グローバル クロックを使用して I/O フリップフロップの DQS が取り込まれると検出されます。このため、データの最終的な遅延値には、I/O フリップフロップのセットアップ タイムおよびホールド タイムが考慮されています。ワーストケースの解析には、I/O フリップフロップの固有セットアップ タイム (T_{IDOCK}) と固有ホールド タイム (T_{IOCKD}) が考慮されます。

データ ビットとそれに対するストロブ間のスキューには、PCB レイアウト スキューも考慮されます。

表 3 は、DDR-II インターフェイスの 267MHz での読み出しタイミング解析を示しています。パラメータの単位はすべてピコ秒です。 T_{DATA_PERIOD} はクロック周期の半分から T_{MEM_DCD} を減算した値になります。クロック前の不確定値を合計した地点 (770ps) が、データ有効ウィンドウの開始になります。 T_{DATA_PERIOD} からクロック後の不確定値の合計を引いた地点 (967ps) がデータ有効ウィンドウの終わりになります。この結果、267MHz でのデータ有効ウィンドウの幅は 197ps となります。このデータ有効ウィンドウには 75ps の精度で 2 つのタップがフィットするため、ウィンドウ幅は十分であると言えます。

表 3 : 267MHz の読み出しタイミング解析 (DDR-II インターフェイス)

不確定パラメータ	値 (ps)	クロック前の不確定値	クロック後の不確定値	説明
T_{CLOCK}	3750			クロック周期
T_{MEM_DCD}	188	0	0	クロック位相 (クロック周期の半分) からこのデューティ サイクルの歪み耐性が減算されて、 T_{DATA_PERIOD} が決定されます。
T_{DATA_PERIOD}	1687			データ周期は、クロック周期の半分から 10% のデューティ サイクルの歪みを減算した値です。
T_{AC}	±500	500	500	メモリ ベンダ指定のデータ出力のアクセス時間です。

表 3 : 267MHz の読み出しタイミング解析 (DDR-II インターフェイス)

不確定パラメータ	値 (ps)	クロック前の不確定値	クロック後の不確定値	説明
T _{PACKAGE_SKEW}	0	0	0	パッケージスキューを補正するために PCB トレース長が調整されているので、このスキューは考慮されません。
T _{SETUP} - 最小	100	100	0	DQS のエッジは、I/O フリップフロップにグローバル クロックを使用して DQS を取り込むと検出されます。このため、データの最終的な遅延値には、I/O フリップフロップのセットアップ タイムおよびホールド タイムが考慮されています。ワーストケースの解析の場合は、フリップフロップの固有セットアップ タイムが考慮されます。
T _{HOLD} - 最大	50	0	0	DQS のエッジは、I/O フリップフロップにグローバル クロックを使用して DQS を取り込むと検出されます。このため、データの最終的な遅延値には、I/O フリップフロップのセットアップ タイムおよびホールド タイムが考慮されています。ワーストケースの解析の場合は、フリップフロップの固有ホールド タイムが考慮されます。
T _{JITTER}	100	100	100	ストロブとデータのジッタの間接的な原因となるクロック ジッタ
T _{CLOCK_TREE_SKEW} - 最大	50	50	50	DQS と対応する DQ は近くに配置されているため、グローバル クロック ラインのスキューに考慮される値は小さくなります。
T _{PCB_LAYOUT_SKEW}	20	20	20	ボード上のデータラインと対応するストロブ間のスキュー
不確定値	387	770	720	
データ有効ウィンドウ	197	770	967	

図 7 は、計算されたデータ有効ウィンドウを示しています。

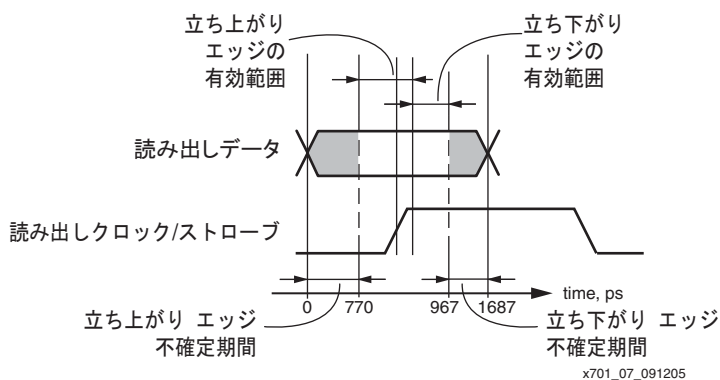


図 7 : データ有効ウィンドウ

リファレンス デザイン

ダイレクト クロッキングを使用したデータ キャプチャのリファレンス デザインは、Memory Interface Generator (MIG) ツールから使用できます。このツールは、ザイリンクスの CORE Generator に含まれています。最新のリファレンス デザインを入手するには、次のザイリンクスの Web サイトの IP アップデートをダウンロードしてください。

http://www.xilinx.co.jp/xlnx/xil_sw_updates_home.jsp

まとめ

Virtex-4 I/O アーキテクチャでは、ソース同期のメモリ インターフェイスを容易にインプリメントできます。このアプリケーション ノートおよびリファレンス デザインでは、次のアーキテクチャ機能が使用されます。

- IDELAY ブロック - 遅延素子を高いタップ精度で継続して調整します。
- FIFO16 プリミティブ - FIFO として使用されるブロック RAM です。ステータス フラグを生成するのに余分な CLB が必要ありません。
- 高速差動グローバル クロッキング リソース - 使用するとデューティ サイクルが改善します。差動クロッキングを実行すると、デザインに必要なグローバル クロック リソースの数が減少します。

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	改訂内容
2004 年 9 月 9 日	1.0	初期リリース
2004 年 11 月 1 日	1.1	「データ キャプチャと再キャプチャ」セクションの説明を変更。 図 6 を変更。Web のリファレンス デザインを更新。
2005 年 7 月 11 日	1.2	表 3 、 図 6 および 図 7 を変更。「リファレンス デザイン」を変更。 表 1 を追加。
2005 年 9 月 13 日	1.3	「読み出し タイミング解析」および「リファレンス デザイン」 セクションを更新。