



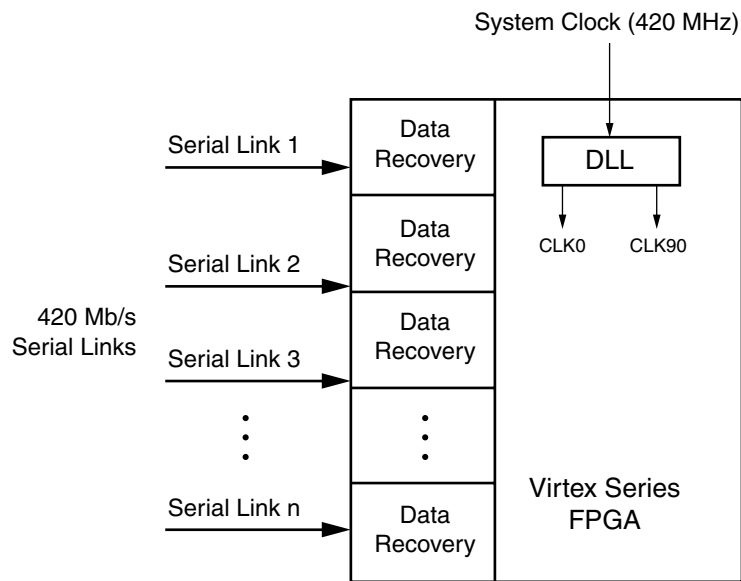
データの復元

著者: Nick Sawyer

XAPP224 (v2.2) 2002 年 8 月 7 日

概要

データの復元を行うと、受信側デバイスは受信したデータ ストリームからクロック データを抽出できます。通常、受信側デバイスは受信したクロック/データ ストリームから抽出したデータを、独立したクロック ドメインに移行します。受信側デバイスのクロックをデータ送信に使用する場合もあります。このアプリケーションノートで説明する回路は、Virtex™-E-7 デバイスの場合は最大 160Mb/s、Virtex-II-5 デバイスまたは Virtex-II Pro™ -6 デバイスの場合は最大 420Mb/s のデータ レートにおける部分的なソリューションを提供します。このソリューションは、実際にはクロックを復元しないので不完全ですが、受信したデータはすべて抽出されます。速度は、新しいクロックと位相が 90° シフトした別のクロックの両方を生成できるモードで DLL (データ ロック ループ) が処理できる最大周波数によって制限されます。代表的なアプリケーションを、図 1 に示します。



X224_01_010202

図 1: 代表的なデータ復元アプリケーション

はじめに

このアプリケーション ノートで説明する回路では、デコードするデータ ストリームの標準周波数と同じ周波数で動作するクロック (ローカル オシレータ) を使用します。一般に、ローカル オシレータの速度は、受信したクロック/データ ストリームの速度とわずかに異なります。たとえば、通常のリンクは 400MHz 付近で動作します。クロック レートに対する実際のパフォーマンスについては、回路の動作に関するセクションで説明します。

受信したクロック/データ ラインのエンコードでは、ラインが High の場合は「1」、Low の場合は「0」という以上のことは行わないものとします。その他のデータ エンコーディングの可能性については、このアプリケーション ノートの今後のバージョンで説明します。最低限の遷移条件については、次のセクションで説明します。

© 2002 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://www.xilinx.co.jp/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。

Virtex-E または Virtex-II を使用する低速なリンクの場合、受信したシステム クロックは DLL コンポーネントに供給され、同期回路へのクロック (CLK) 供給と DLL のフィードバックに DLL CLK0 が使用されます。位相が 90° 遅れていてオリジナルのクロックと同期しているもう 1 つの入力クロックも使用できます。Virtex-II を使用する高速なリンクの場合は、2 つの DLL (DCM) モジュールが必要になります。この場合、一方の DLL は CLK を、もう一方の DLL は Virtex-II デバイスで可能な固定位相シフト機能を使用して CLK90 を提供します。これらの波形を、**図 2** のタイミング図に示します。この図には、次のセクションで使用する 4 種類の受信データ例も示されています。

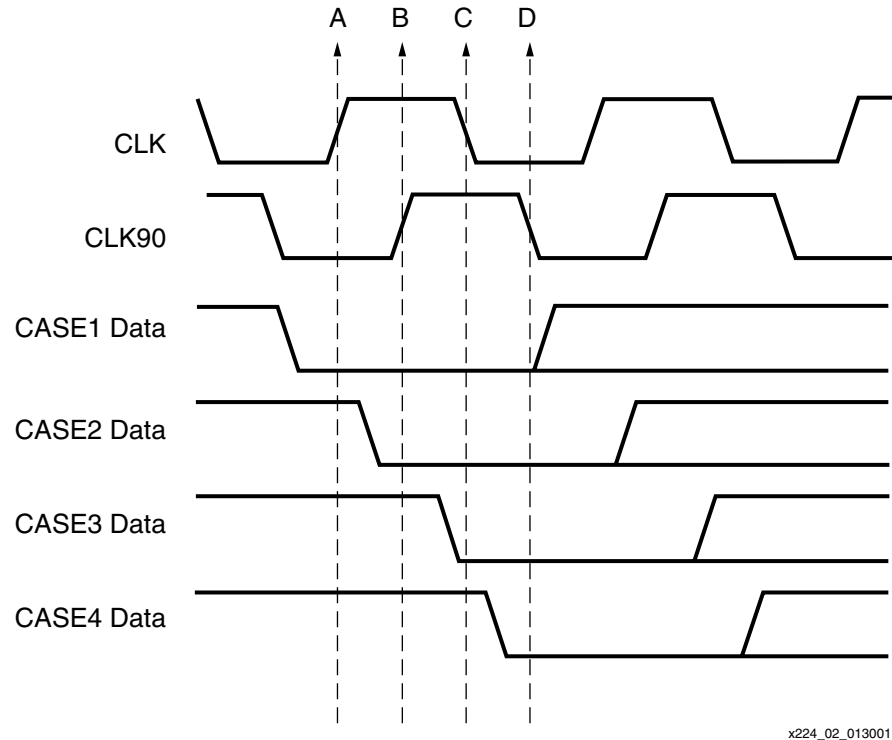
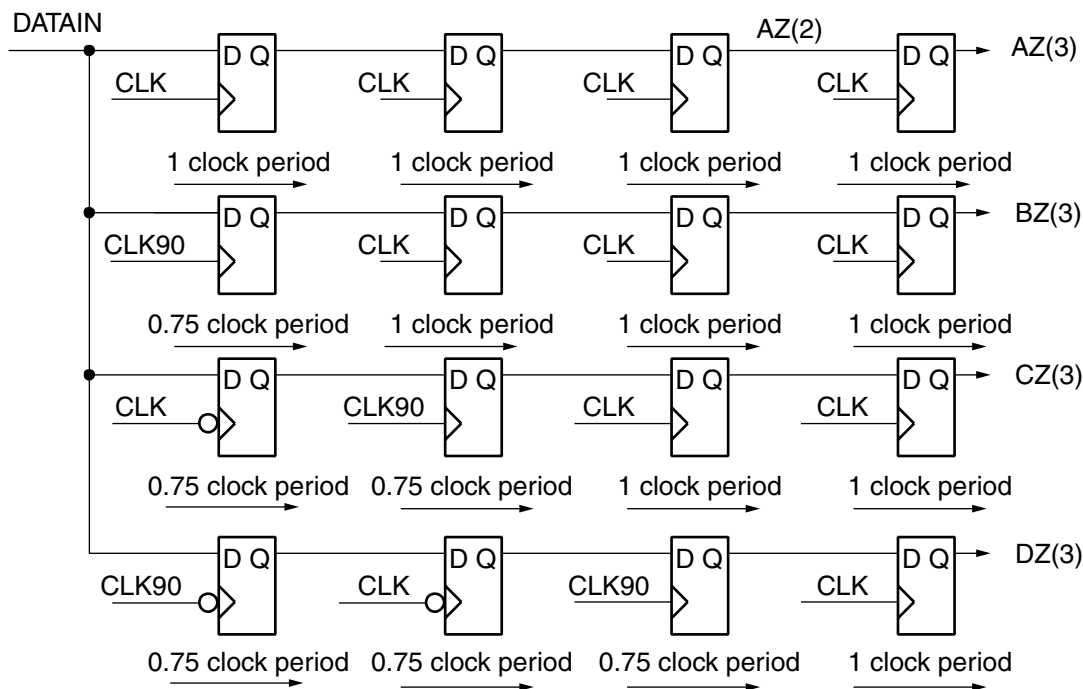


図 2: タイミング図

図 3 に示すように、受信したデータは 4 つのフリップフロップに入力されます。これらのフリップフロップの 2 つは CLK、他の 2 つは CLK90 によってクロックが供給され、それぞれ立ち上がりエッジを読み取るものと立ち下がりエッジを読み取るものが 1 つずつあります。入力ピンからこれらのフリップフロップまでの遅延をほぼ等しくすることが重要です。これは、このネットに対してソフトウェアで MAXSKEW というパラメータ (500ps など) を指定することによって簡単に実現できます。遅延の値自体は無関係で、スキューのみ重要です。

1 番目のフリップフロップは、クロックの立ち上がりエッジでクロックが供給されます。これは、タイム ドメイン A に属します。2 番目のフリップフロップは CLK90 の立ち上がりエッジ (タイム ドメイン B)、3 番目のフリップフロップは CLK の立ち下がりエッジ (タイム ドメイン C)、4 番目のフリップフロップは CLK90 の立ち下がりエッジ (タイム ドメイン D) でクロックが供給されます。**図 2** のタイミング図図に示したように、これによってオリジナルのクロックから 90° ずつ位相がずれている 4 つのデータ サンプル ポイントが得られます。システム クロックが 420MHz の場合、このロジックは実質的に 1680 MHz で動作することになります。

この 4 つのサンプル ポイントには、メタステーブル状態を回避し、同じタイム ドメインに移行するため、もう一度クロックが供給されます。これは、4 倍クロック周波数のロジック パスを回避するため、3 段階で行われます。



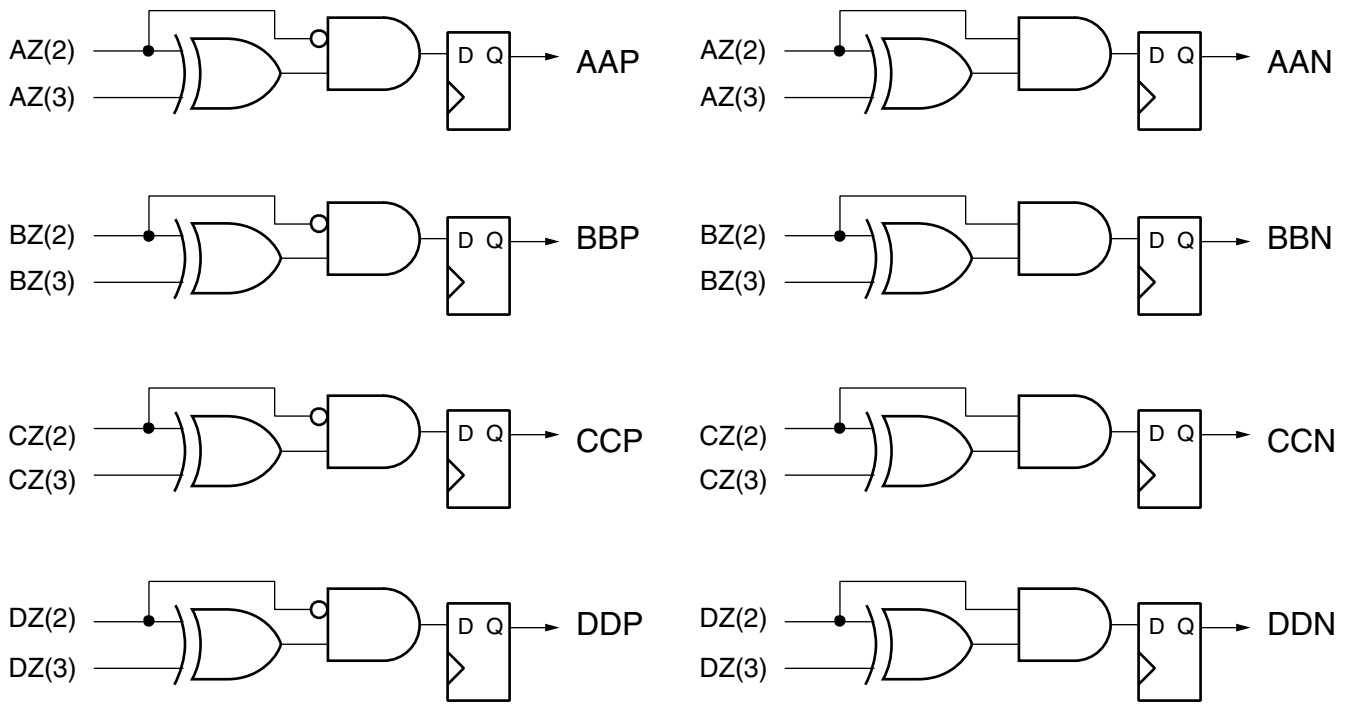
x224_03_121501

図 3: 入力ステージ

図 4 に示す初めの決定ステージでは、データラインの遷移が検出されます。

信号 AAP ~ DDP は正の遷移、AAN ~ DDN は負の遷移を表します。これで、決定プロセスでは 8 種類の信号を使用できるようになります。そして、データが遷移する場合はいずれか 1 つだけが High に遷移する 4 種類の信号をデコードできます。これには、次のような 4 種類の状態があります。

1. AAP = BBP = CCP = DDP = 1、または AAN = BBN = CCN = DDN = 1。初めにタイムドメイン A にデータが現れます。したがって、タイムドメイン C のデータが転送に使用されます。
2. AAP = 1 および BBP = CCP = DDP = 0、または AAN = 1 および BBN = CCN = DDN = 0。初めにタイムドメイン B にデータが現れます。したがって、タイムドメイン D のデータが転送に使用されます。
3. AAP = BBP = 1 および CCP = DDP = 0、または AAN = BBN = 1 および CCN = DDN = 0。初めにタイムドメイン C にデータが現れます。したがって、タイムドメイン A のデータが転送に使用されます。これは、周期の途中 (ノイズマージンが最高) でサンプリングされたデータです。
4. AAP = BBP = CCP = 1 および DDP = 0、または AAN = BBN = CCN = 1 および DDN = 0。初めにタイムドメイン D にデータが現れます。したがって、タイムドメイン B のデータが転送に使用されます。



x244_04_032002

図 4: 決定ステージ 1

データの選択は、該当するタイム ドメインからデータ ビットを選択する非常に単純なマルチプレクサによって行われます。たとえば、タイム ドメイン A のデータを使用する場合は、データ ビット AZ(3) が出力されます。

前述したように、ローカルクロックの速度は受信したクロック/データ ストリームとわずかに異なる可能性があります。このため、受信データが無効な場合 (ローカルクロックが速い) と、2 データ ビットが必要な場合 (ローカルクロックが遅い) があります。クロック/データは時間とともに変化するので、マルチプレクサを使用して 4 種類あるタイム ドメインのいずれかからデータを選択できます。ただし、これには 2 つの例外があります。タイム ドメインが A から D に移行する場合、出力されるデータは無効になります。これは、データ有効信号 (DV) が 0 になることによって示されます。タイム ドメインが D から A に移行する場合は、2 ビットのデータを生成する必要があります。これは、データ有効ビットが両方とも High になることによって示されます。これを、図 5 に示します。実際のシステムでは、クロックのジッターなどのため、このような現象が連続して不規則に発生します。ただし、データ有効ビットに従えば、データ出力は常に有効になります。たとえば、受信データ ストリームが 400Mb/s (2.500ns 周期)、ローカルクロックが 401MHz (2.494ns 周期) の場合、ローカルクロックの約 416 サイクルごとに受信データが無効 (DV = 0) になります。

逆に、ローカルクロックが 400MHz、受信データ ストリームが 401 Mb/s であるとします。この場合は、平均して 416 クロック サイクルごとに 2 ビットのデータが生成されます。2 つの独立したクリスタル オシレータを使用するなどしてほぼ周波数が等しい場合は、周波数の差が数 ppm になり、ローカルクロックに従って 0 ~ 2 ビットのデータが生成されるように回路が変化します。しかし、データは常に正しく抽出されます。

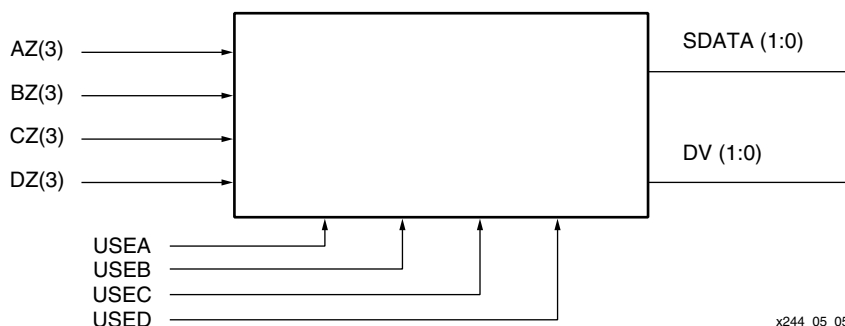


図 5: データ有効ロジック

x244_05_051602

受信データの使用

前述したように、受信クロック サイクルごとに 0～2 ビットのデータがデータ復元回路によって生成されます。図 5 では、これらのデータに SDATA(1:0) というラベルが付いており、その有効性は DV(1:0) によって示されます。受信した 1 番目のビットは SDATA(0) です。SDATA(2) が有効な場合は、これが 2 番目のビットになります。通常、これらのビットは 8、16、または 32 ビット幅でさらに処理するため FIFO に入力されます。8 ビットデータの処理が必要な場合、ロジックが少し複雑になります。通常は、受信したデータ ビットはパラレル化を行うためにシフトレジスタに渡され、8 ビットが入力されると 8 ビットワードが FIFO に入力されます。しかし、データ復元回路は 0 または 2 ビットのデータを生成する可能性があるため、受信ビットカウンタが 7 または 9 になるケースを考慮する必要があります。これに対処する一番良い方法は、9 ビットのシフトレジスタを使用することです。有効なデータビットがない場合、シフトレジスタとカウンタは変化しません。1 ビットが有効な場合は MSB にシフトインされ、カウンタの値が 1 増加します。2 ビットが有効な場合は、両方がシフトレジスタに入力され、カウンタの値が 2 増加します。カウンタが 8 になった場合、有効なバイトはシフトレジスタのビット位置 1～8 にあります。カウンタが 9 になった場合は、ビット位置 0～7 が有効なバイトになります。したがって、単純なマルチプレクサを使用して FIFO (または残りのシステム) に有効なデータを渡すことができます。

この回路のコード例は、後述するデザインファイルに含まれています。

回路が動作を始めたときは正しい位相の関係を確立しようとするため、受信した初めの数ビットが正確であるという保証はありません。これは初めの 8 ビットに適用され、残りのデータは有効になります。

メタステーブル状態

クロックのサンプルポイント間でデータの状態が正しく変化しない可能性があるため、メタステーブル状態になる場合があります。データが遷移してサンプルポイント (図 2 のクロック サンプル A など) に近づくと、クロック CLK のセットアップタイムになり、何種類かの状況が発生することになります。

フリップフロップが十分に高速で遷移を認識できる場合は $AAx = BBx = CCx = DDx = 1$ (ケース 1) になり、前述のような動作になります。フリップフロップが遷移を認識できない場合は $AAx = 1$ and $BBx = CCx = DDx = 0$ (ケース 2) になり、この場合も正しく動作します。

さらに、フリップフロップが一時的にメタステーブル状態になる場合もあります。これが発生した場合でも、2 番目の同期フリップフロップは 1 または 0 を認識してその状態を記憶するので、前述したものと同一状況になります。メタステーブル状態にある期間が入力クロックの周期と「正確に」等しい場合を除いて、メタステーブル状態になる可能性があるフリップフロップの負荷が 1 つであれば問題はありませぬ。ただし、現在の非常に高速なシリコンではこのような状況になることはほとんどありません。また、そうなったとしても、データパスにはさらに 3 つのフリップフロップがあります。したがって、メタステーブル状態によって回路の動作が不安定になる可能性はまずありません。

ロックの必要条件

この回路では、データの遷移が発生する頻度が低いと、データの完全性を維持できなくなります。このため、ロックの必要条件があります。クロック周期の 1/4 以内の時間に、少なくとも 1 回はデータが遷移する必要があります。図 1 の例では、400Mb/s でデータを受信するので、1 周期は約 2.500ns ということになります。したがって、1/4 周期は 0.625ns になります。ローカル オシレータは周波数が 401MHz なので、周期は 2.494ns となり、受信データよりも (2.500 - 2.494)ns = 0.006ns 速いことになります。1/4 周期 (0.625ns) を 0.006ns で割ると、約 100 になります。したがって、回路が正しく動作するには、100 クロック サイクルごとに 1 回以上は負の遷移が必要になります。

ローカル オシレータの周波数を 400.5MHz にすると、クロック サイクルの必要条件は 200 になります。受信データが 8b/10b などの方法でコーディングされている場合は、十分な回数の遷移が発生するので、問題はありませぬ。受信データがコーディングなどをしていない生のビットストリームである場合は、十分な回数の遷移が発生しない可能性があるため、注意が必要です。このデザインは、送信と受信に別々のオシレータを使用する LVDS デモ ボード上で 23 ビットの prbs (疑似ランダム バイナリ シーケンス) データ パターンを使用した場合、420Mb/s で正しく動作しています。

シミュレーション

この回路では、2 つの非同期クロック ドメインを使用します。シミュレーションを行うと、セットアップタイムの違反と「X」伝搬を調べることができます。この「X」伝搬は、MTI で「vsim+no_notifier」コマンドを使用すると除去できます。「X」伝搬は、メタステーブル状態のセクションで説明した理由から、安全に除去できます。

リファレンス デザイン ファイル

リファレンス デザイン回路は、HDL でインプリメントされています。この回路は、完全に合成可能です。リファレンス デザイン ファイル ([xapp224.zip](#)) には、すべてのタイミング制約情報を含んでいる top.ucf ファイルが含まれています。非常に高速なパスがあるので、このファイルを使用してください。

おわりに

Virtex-E デバイスの場合は最大 160Mb/s、Virtex-II デバイスの場合は最大 420 Mb/s の速度でシリアルリンクからデータを抽出できます。

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
09/18/00	1.0	初期リリース
01/10/01	1.1	Virtex-II シリーズの FPGA 用にアップデート。
01/31/01	1.2	図 1、2、3 を修正。「シミュレーション」のセクションを追加。
01/30/02	2.0	回路を再デザインして速度が異なるローカル オシレータに対応。
03/28/02	2.1	正の遷移と負の遷移の両方が含まれるように図 4 をアップデート。
08/07/02	2.2	図 5 を修正。