XILINX®

XAPP691 (v1.0) 2004 年 2 月 2 日

パラメータ設定可能な LocalLink FIFO

概要

このアプリケーション ノートでは、両側に LocalLink インターフェイスを備えた First-In-First-Out メモリ キューであるパラメータ設定可能な LocalLink FIFO の設計について説明します。LocalLink インターフェイスによってプロトコルにとらわれない信号を定義でき、パケット用データの送信を行うことができます。また、フロー制御および任意の長さのデータ送信などの機能も可能になります。この LocalLink FIFO は 2 つの LocalLink インターフェイスで構成されます。一方は書き込みポートにあり、アップストリーム ユーザー アプリケーションと接続し、もう一方は読み出しポートにあり、ダウンストリーム ユーザー アプリケーションと接続します。LocalLink FIFO では、制御ロジックによって LocalLink 信号を読み取り、このロジックがすべての読み出しポインタおよび書き込みポインタの管理とフロー制御用の FIFO ステータス信号の生成を行います。LocalLink FIFO の読み出しポートおよび書き込みポートでは完全に同期し、独立したクロック ドメインを使用します。また、LocalLink FIFO では、読み出しポートおよび書き込みポート間のデータ幅を変換でき、FIFO 用のメモリは、ワード数にパラメータ設定可能な ブロック SelectRAM または分散 RAM を使用して設計できます。また、Frame Length 信号および Length Ready 信号をオプションとして出力することによって、各受信フレームのバイト数がわかり、ダウンストリーム ユーザー アプリケーションではデータを読み出す前にフレーム長がわかります。

はじめに

このアプリケーション ノートでは、LocalLink インターフェイス、データ幅およびワード数のパラメータ設定可能な FIFO、データ幅変換、フレーム長カウントなどの機能を非同期 FIFO へ統合する LocalLink FIFO の設計について説明します。図 1 に 2 つの LocalLink インターフェイスを示します。一方は書き込みポートにあり、アップストリーム ユーザー アプリケーションと接続し、もう一方は読み出しポートにあり、ダウンストリーム ユーザー アプリケーションと接続します。

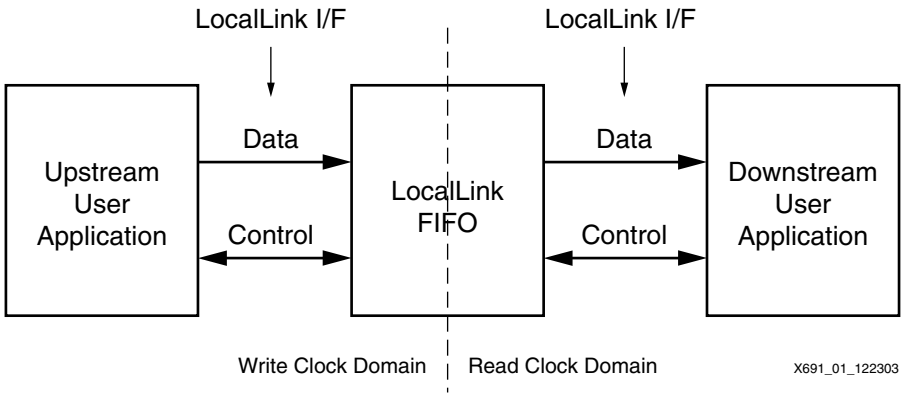

図 1 : LocalLink FIFO

LocalLink インターフェイスによってプロトコルにとらわれない信号を定義でき、パケット用データの送信を行うことができます。また、フロー制御および任意の長さのデータ送信などの機能も可能になります。ソース Ready 信号およびデスティネーション Ready 信号を使用することによって、LocalLink インターフェイス上でのフロー制御が明確になります。LocalLink FIFO は、XAPP258 「Virtex-II ブロッ

© 2003 Xilinx, Inc. All rights reserved. すべての Xilinx の商標、登録商標、特許、免責条項は、http://www.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。

ク RAMを使用した FIFO」および XAPP261「Virtex-II ブロック RAM メモリを使用したデータ幅変換 FIFO」に基づいて設計されています。このアプリケーション ノートをお読みになる前に、これら 2 つのアプリケーション ノートおよび LocalLink インターフェイス仕様をお読みください。

LocalLink FIFO は、Virtex-II および Virtex-II Pro シリーズ FPGA のブロック SelectRAM または分散 RAM を使用して設計できます。FIFO をブロック SelectRAM にインプリメントする場合は、分散 RAM にインプリメントする場合よりもメモリ領域が広くなります。

LocalLink FIFO の機能は次のとおりです。

- メモリをブロック SelectRAM または LUT ベースの分散 RAM にインプリメント
- 読み出しポートおよび書き込みポートに LocalLink インターフェイス
  - フロー制御およびパケット用のデータ送信などの LocalLink 機能
- 8 ビットから 128 ビットまでのデータ幅をサポート
- 分散 RAM の場合は最大 512、ブロック SelectRAM の場合は最大 32,767 までのメモリ ワード数をサポート
- 読み出し/書き込みポートでは完全に同期し、独立したクロック ドメインを使用
- パイプライン化が 2、3 段階であり、低いレイテンシを維持
- FIFO の使用率を示すための 4 ビット FIFO ステータス信号を含む
- オプションとして Length Count、Length Ready および Length Error 信号を出力
- 動作速度は約 200MHz

図 2 に、LocalLink FIFO インターフェイスのトップ レベル ブロック図を示します。LocalLink FIFO は主に Data FIFO、Control FIFO およびオプションとしての Length FIFO の 3 ブロックで構成されます。Data FIFO にはフレーム データが格納され、Control FIFO にはフレーム区切り文字 (SOF、EOF) およびデータの余りの部分を示す REM のような LocalLink 制御データが格納されます。これらの 2 つの FIFO は読み出しおよび書き込みアドレス カウンタを共有するため、同期で動作します。Data FIFO に格納されたすべてのデータ ワードは Control FIFO の制御データ ワードに関連します。オプションとしての Length FIFO は専用の読み出しおよび書き込みアドレス カウンタを使用します。この Length FIFO 内のすべてのデータ エントリは、フレーム全体の長さを示し、Data FIFO に格納されたデータ ワードを含む場合があります。Length FIFO をインプリメントする場合、LocalLink FIFO インターフェイスの Frame Length (**len_out**)、Length Ready (**len_rdy_out**)、および Length Error (**len_err_out**) 出力によって、各受信フレームのバイトの長さがわかります。これらの信号が出力されると、ダウンストリーム ユーザー アプリケーションでは、データを読み出す前にフレーム長がわかります。

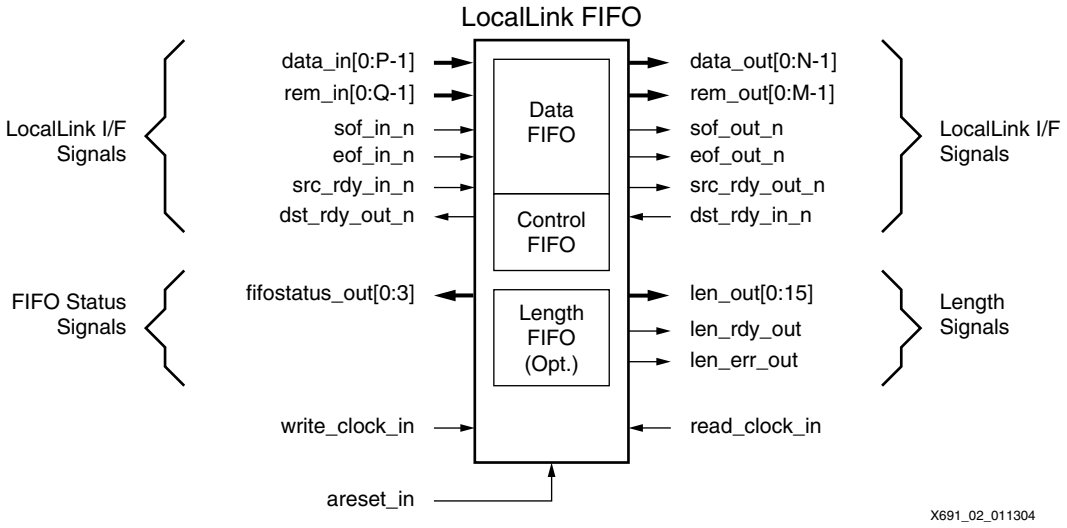

図 2： LocalLink FIFO インターフェイス

インターフェイス

2 ページの図 2 に LocalLink FIFO インターフェイス信号を示し、次の 表 1 に LocalLink FIFO インターフェイスのクロック信号およびリセット信号を示します。また、表 2 にアップストリーム アプリケーションおよびダウンストリーム アプリケーションへの LocalLink データ信号と制御信号の概要を示し、4 ページの表 3 および 4 ページの表 4 に FIFO のステータス信号を示します。5 ページの表 5 に示す Length Count、Length Ready、および Length Error 出力はオプションの信号として使用できます。

LocalLink FIFO インターフェイスは **data_in**、**data_out**、**rem_in** および **rem_out** などのベクタ信号をビッグ エンディアン形式で使用するため、**data_in** の最上位ビット (MSB) は最初に受信され、これが **data_out** の MSB となります。

表 1: LocalLink FIFO のクロック信号およびリセット信号

信号名	適用範囲	方向	説明
areset_in	アップストリーム /ダウンストリーム	入力	非同期リセット入力 (アクティブ High)
write_clock_in	アップストリーム	入力	アップストリーム ユーザー アプリケーションからの書き込みポート用クロック入力
read_clock_in	ダウンストリーム	入力	ダウンストリーム ユーザー アプリケーションからの読み出しポート用クロック入力

表 2: LocalLink FIFO の LocalLink インターフェイス信号

信号名	適用範囲	方向	説明	メモ
data_in[0:P-1]	アップストリーム	入力	アップストリーム ユーザー アプリケーションからのデータ入力で、P ビット幅です。	P = 8、16、32、64、128
rem_in[0:Q-1]		入力	アップストリーム ユーザー アプリケーションからの余り部分のデータ入力であり、**eof_in_n** がアサートされた場合の **data_in** の有効バイト数を示します。このデータのビット幅は Q であり、バイナリでエンコードされます。**data_in** が 32 ビット幅の場合、信号は **rem_in[0:1]** となります。値が 0 の場合、バイト **rem_in[0:7]** が有効であることを示し、3 の場合はすべてのバイトが有効であることを示します。	P = 8 の場合は Q = 1 となり、それ以外の場合は $Q = \log_2(P/8)$
sof_in_n		入力	**data_in** バス上でのフレーム送信の開始を示します。	アクティブ Low
eof_in_n		入力	**data_in** バス上でのフレーム送信の終了を示します。	アクティブ Low
src_rdy_in_n		入力	現在のサイクルで **data_in** が有効であることを示します。	アクティブ Low
dst_rdy_out_n		出力	現在のサイクルで **data_in** バスにあるデータを LocalLink FIFO で受信できる状態にあることを示します。	アクティブ Low

表 2: LocalLink FIFO の LocalLink インターフェイス信号 *(Continued)*

信号名	適用範囲	方向	説明	メモ
data_out[0:N-1]	ダウンストリーム	出力	ダウンストリーム ユーザー アプリケーションへのデータ出力で、N ビット幅です。	N = 8、16、32、64、128
rem_out[0:M-1]		出力	ダウンストリーム ユーザー アプリケーションへの余りの部分のデータ出力であり、**eof_out_n** がアサートされた場合の **data_out** の有効バイト数を示します。このデータのビット幅は M ビットであり、バイナリでエンコードされます。**data_out** が 32 ビット幅の場合、この信号は **rem_out[0:1]** となります。値が 0 の場合、バイト **rem_out[0:7]** が有効であることを示し、3 の場合はすべてのバイトが有効であることを示します。	N = 8 の場合は M = 1 となり、それ以外の場合は $M = \log_2(N/8)$
sof_out_n		出力	**data_out** バス上でのフレーム送信の開始を示します。	アクティブ Low
eof_out_n		出力	**data_out** バス上でのフレーム送信の終了を示します。	アクティブ Low
src_rdy_out_n		出力	現在のサイクルで **data_out** が有効であることを示します。	アクティブ Low
dst_rdy_in_n		入力	ダウンストリーム ユーザー アプリケーションが、現在のサイクルで **data_out** バスにあるデータを受信できる状態にあることを示します。	アクティブ Low

表 3: LocalLink FIFO の FIFO ステータス信号

信号名	方向	説明	メモ
fifostatus_out[0:3]	出力	FIFO が 1/2 または 3/4 使用されているなどの情報を示します。この信号のビット定義については、表 4 を参照してください。	**write_clock_in** に同期

表 4: FIFO ステータス信号のビット定義

ビット 0	ビット 1	ビット 2	ビット 3	説明
1	1	1	1	LocalLink FIFO の 15/16 を使用
1	1	1	0	LocalLink FIFO の 7/8 を使用
1	1	0	1	LocalLink FIFO の 13/16 を使用
1	1	0	0	LocalLink FIFO の 3/4 を使用
1	0	1	1	LocalLink FIFO の 11/16 を使用
1	0	1	0	LocalLink FIFO の 5/8 を使用
1	0	0	1	LocalLink FIFO の 9/16 を使用

表 4: FIFO ステータス信号のビット定義 *(Continued)*

ビット 0	ビット 1	ビット 2	ビット 3	説明
1	0	0	0	LocalLink FIFO の 1/2 を使用
0	1	1	1	LocalLink FIFO の 7/16 を使用
0	1	1	0	LocalLink FIFO の 3/8 を使用
0	1	0	1	LocalLink FIFO の 5/16 を使用
0	1	0	0	LocalLink FIFO の 1/4 を使用
0	0	1	1	LocalLink FIFO の 3/16 を使用
0	0	1	0	LocalLink FIFO の 1/8 を使用
0	0	0	1	LocalLink FIFO の 1/16 を使用
0	0	0	0	LocalLink FIFO の 1/16 未満を使用

表 5: LocalLink FIFO Length 信号

信号名	方向	説明
len_out[0:15]	出力	フレーム長をバイトで示します。受信フレームの長さが LocalLink FIFO がサポートする最大フレーム長を超える場合、この出力バスは 65,535 を維持します。詳細については、12 ページの (c) 最大フレーム長を参照してください。
len_rdy_out	出力	**len_out** バスで新しい Length Out 信号が有効であることを示すアクティブ High パルス信号です。
len_err_out	出力	ブロック SelectRAM インプリメンテーションで Length FIFO がオーバーフローであることを示すアクティブ High 信号です。分散 RAM インプリメンテーションでは、つねにディアサートされます。

メモ：

1. 出力は **read_clock_in** に同期です。
2. Length FIFO がインプリメントされない場合、これらの出力はすべてゼロに接続されます。
3. 分散 RAM を使用してインプリメントした場合、出力は無効になり、読み出しクロック周波数は書き込みクロック周波数より高くなります。詳細については、Length FIFO のセクションを参照してください。

ハードウェア インプリメンテーション

メモリ タイプ

LocalLink FIFO は、Virtex-II およびVirtex-II Pro シリーズ FPGA のブロック SelectRAM または分散 RAM のいずれかを使用し、Data FIFO、Control FIFO、および Length FIFO をインプリメントします。この場合に使用するメモリはどちらか一方のみです。たとえば、Data FIFO にブロック SelectRAM を使用した場合、Control FIFO にもこのメモリを使用することを推奨します。Data FIFO にブロック SelectRAM を使用することによって、十分なメモリ領域を確保できるため、Control FIFO のインプリメンテーションに分散 RAM を使用することは実用的ではありません。一般的に、分散 RAM Control FIFO をブロック SelectRAM Data FIFO と共に使用するためには、複数の分散 RAM を並行に接続する必要があります。この場合、分散 RAM 出力に多入力マルチプレクサが必要になり、パフォーマンスが低下します。たとえば、8 ビット データを格納する場合のブロック SelectRAM のワード数は 2,048 となり、32X8 RAM64X1D をカスケード接続しなければなりません。メモリのワード数が増加するに従い、処理速度が大きく低下するため、両方のメモリ タイプを使用してインプリメンテーションを行うことは有効な設計方法ではありません。

タイミング考察

LocalLink FIFO の読み出しポートおよび書き込みポートは、別々の非同期クロック ドメインで動作できますが、ユーザー アプリケーション ロジックでは同期の問題に適切に対応する必要があります。2 ページの図 2 に示すように、インターフェイス信号はクロック ドメインによって、書き込みは左側に、読み出しは右側に分けられます。すべてのインターフェイス信号は、**read_clock_in** あるいは **write_clock_in** のいずれかに同期であり、クロックの立ち上がりエッジでサンプルされます。ただし、**areset_in** は LocalLink FIFO 全体の非同期リセット信号です。

ブロック SelectRAM インプリメンテーション

Data FIFO は、Virtex-II または Virtex-II Pro シリーズ FPGA のデュアル ポート ブロック SelectRAM プリミティブを使用して設計できます。1 つのプリミティブ (RAMB16_Sm_Sn) で、最大 32 ビット幅 (パリティ ビットは含まない) までの読み出しポートおよび書き込みポートと 18 Kb のメモリを作成できます。さらに、これらのプリミティブを並行に接続することによって、最大 128 ビットのデータ入出力で、さまざまなアドレス幅およびメモリ ワード数を持つマクロを作成できます。これらのマクロを使用した LocalLink FIFO デザインは、読み出しポートまたは書き込みポートで、8、16、32、64 および 128 ビットのデータ幅をサポートします。 また、データ幅の変換はブロック SelectRAM マクロ内部で行われます。

BRAM_MACRO_NUM パラメータは、FIFO にインプリメントされるマクロ数を指定する場合に使用され、FIFO のワード数に影響を与えます。表 6 に異なるデータ幅および BRAM_MACRO_NUM 設定で Data FIFO をインプリメントする場合に必要となるブロック SelectRAM メモリ数を示します。BRAM_MACRO_NUM パラメータの詳細については、13 ページの表 8 を参照してください。

表 6: LocalLink FIFO での Data FIFO 容量およびブロック SelectRAM 数

データ幅*	ブロック SelectRAM 数	Data FIFO 容量
8 ビット	1 x BRAM_MACRO_NUM	2 x BRAM_MACRO_NUM KB
16 ビット		
32 ビット		
64 ビット	2 x BRAM_MACRO_NUM	4 x BRAM_MACRO_NUM KB
128 ビット	4 x BRAM_MACRO_NUM	8 x BRAM_MACRO_NUM KB

* 読み出しポートと書き込みポート間の最大データ幅を示します。

Control FIFO は、各ブロック SelectRAM で使用可能なパリティ メモリを使用して作成できます。7 ページの図 3 に示すタイプ I インプリメンテーションの場合、LocalLink 制御信号がパリティ ポート (DIPA および DOPB) に入るため、Control FIFO と Data FIFO は同一のブロック SelectRAM メモリを共有することが可能です。一方、7 ページの図 4 に示すタイプ II インプリメンテーションでは、LocalLink 制御信号がパリティ ポート幅を超えているため、別のブロック SelectRAM メモリが残りの制御データを格納するために使用されます。8 ページの図 5 に示すタイプ III インプリメンテーションでは、各 Data FIFO および Control FIFO に専用のブロック SelectRAM を使用します。LocalLink FIFO は、読み出しおよび書き込みデータ幅の組み合わせによって、3 つのうちのいずれかの形で設計されます。

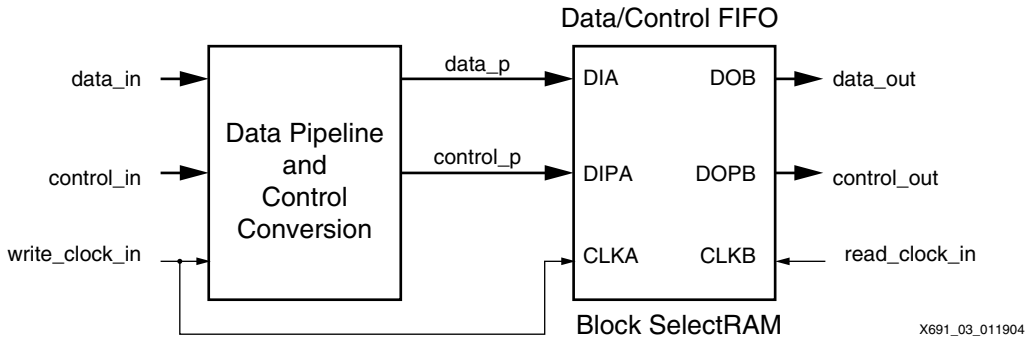

図 3 : Data FIFO および Control FIFO 用ブロック SelectRAM インプリメンテーション タイプ I

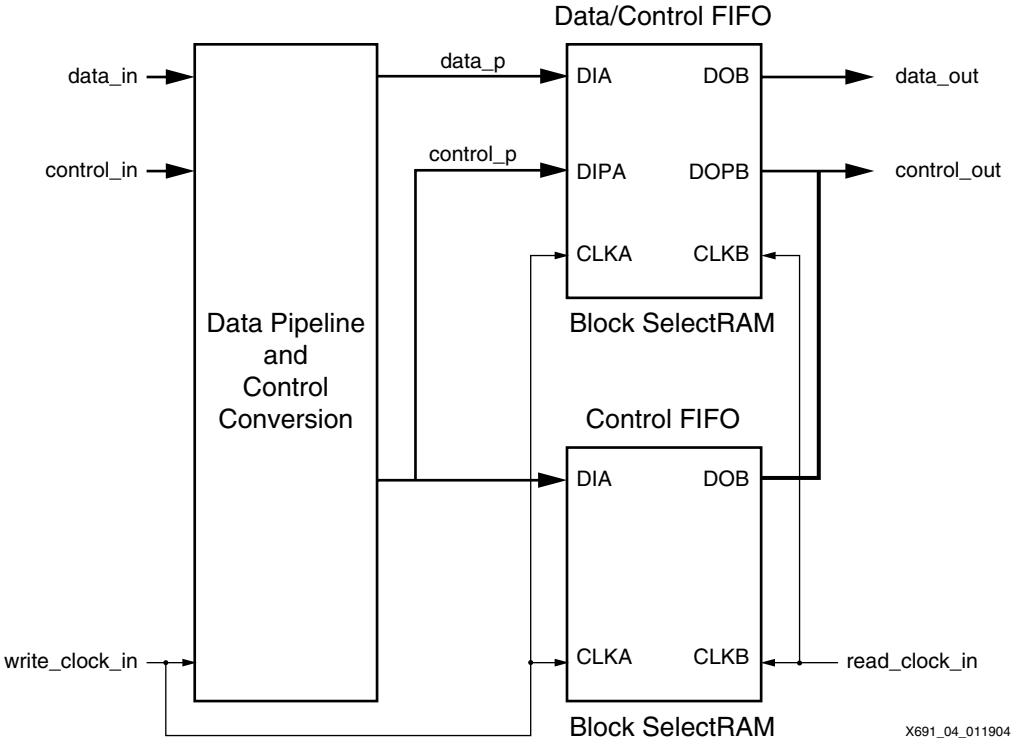

図 4 : Data FIFO および Control FIFO 用ブロック SelectRAM インプリメンテーション タイプ II

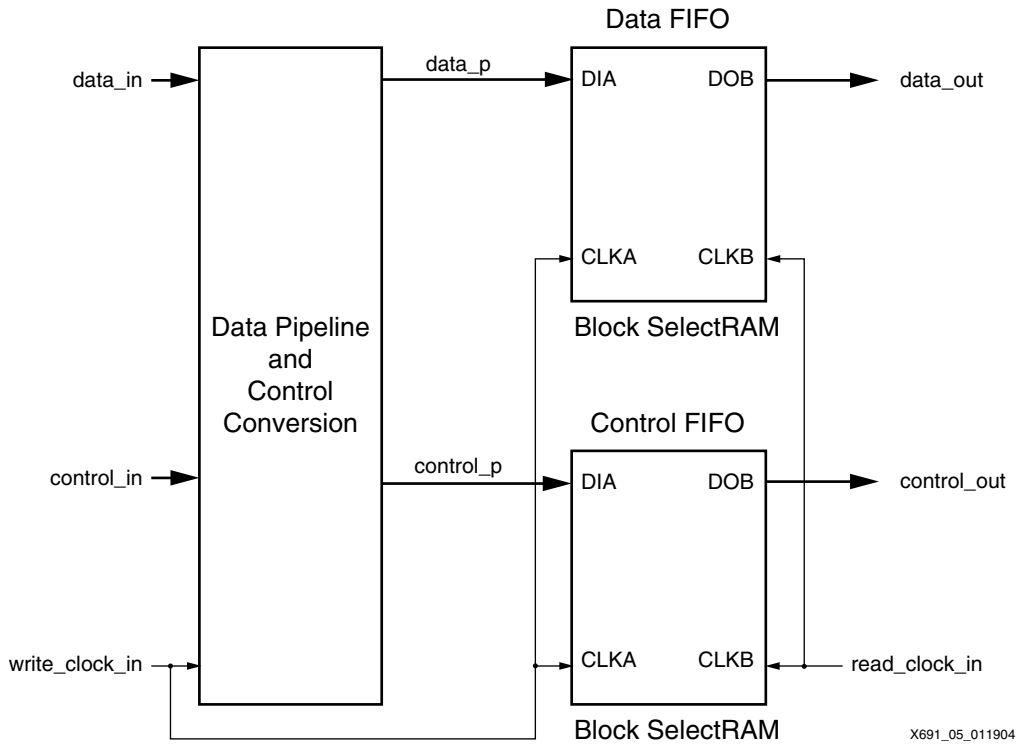

図 5： Data FIFO および Control FIFO 用ブロック SelectRAM インプリメンテーション タイプ Ⅲ

分散 RAM インプリメンテーション

ブロック SelectRAM を使用する代わりに、デュアル ポートで LUT ベースの分散 RAM を使用して、Data FIFO を作成できます。複数の RAM16X1D、RAM32X1D および RAM64X1D プリミティブをカスケード接続して使用し、必要な幅およびワード数のマクロを作成します。ブロック SelectRAM を使用する場合と異なり、デュアル ポート分散 RAM の場合はデータ幅の変換を行えないため、読み出しポートまたは書き込みポートのいずれかでデータ幅変換ロジックを使用する必要があります。

9 ページの図 6 に示すタイプ Ⅰ は、書き込みポートのデータ幅 (WR_DWIDTH パラメータ) が読み出しポートのデータ幅 (RD_DWIDTH パラメータ) より広い場合であり、データ幅変換のロジックは分散 RAM の読み出しポートに配置されます。9 ページの図 7 に示すタイプ Ⅱ は、書き込みポートのデータ幅が読み出しポートのデータ幅より狭い場合であり、データ幅変換用のロジックは分散 RAM の書き込みポートに配置されます。10 ページの図 8 に示すタイプ Ⅲ のように、読み出しポートと書き込みポートのデータ幅が一致している場合は、データ幅変換用のロジックは必要ありません。分散 RAM インプリメンテーションのパラメータに関する詳細は、13 ページの表 8 を参照してください。

Control FIFO には REM、SOF および EOF 制御データが格納されます。Control FIFO には、つねに Data FIFO と同一のアドレス ポインタを共有する専用の分散 RAM を使用し、Data FIFO でデータ幅の変換が必要な場合は、Control FIFO 用に類似したマッピング ロジックを配置します。

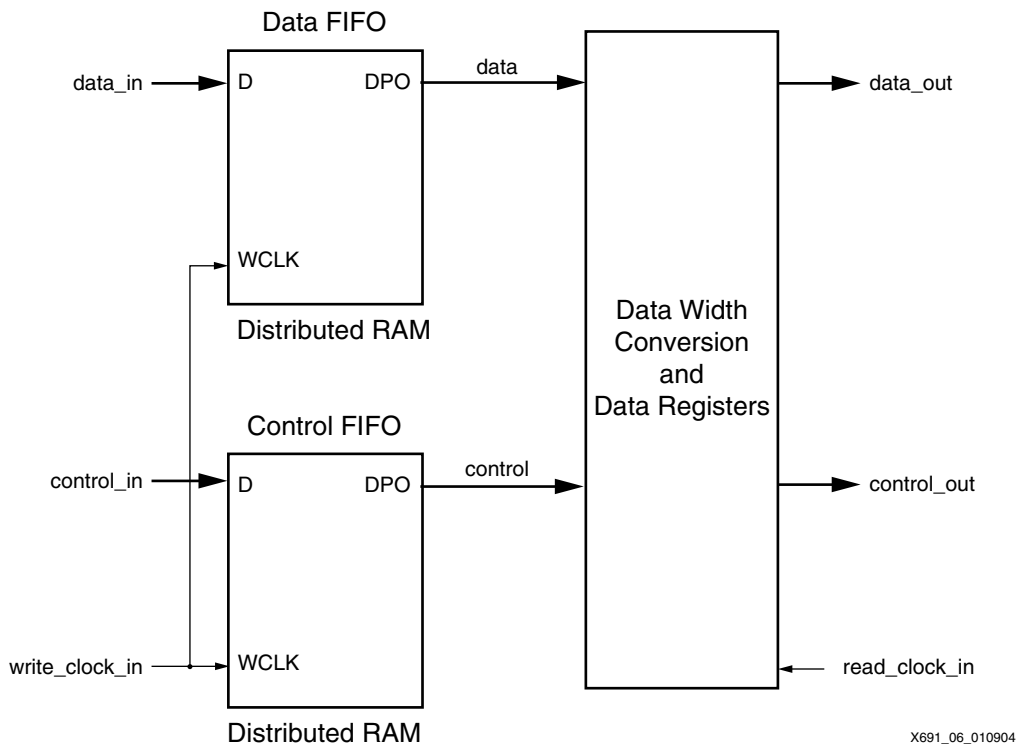

図 6： Data FIFO および Control FIFO (WR_DWIDTH > RD_DWIDTH) 用分散 RAM インプリメンテーション タイプ Ⅰ

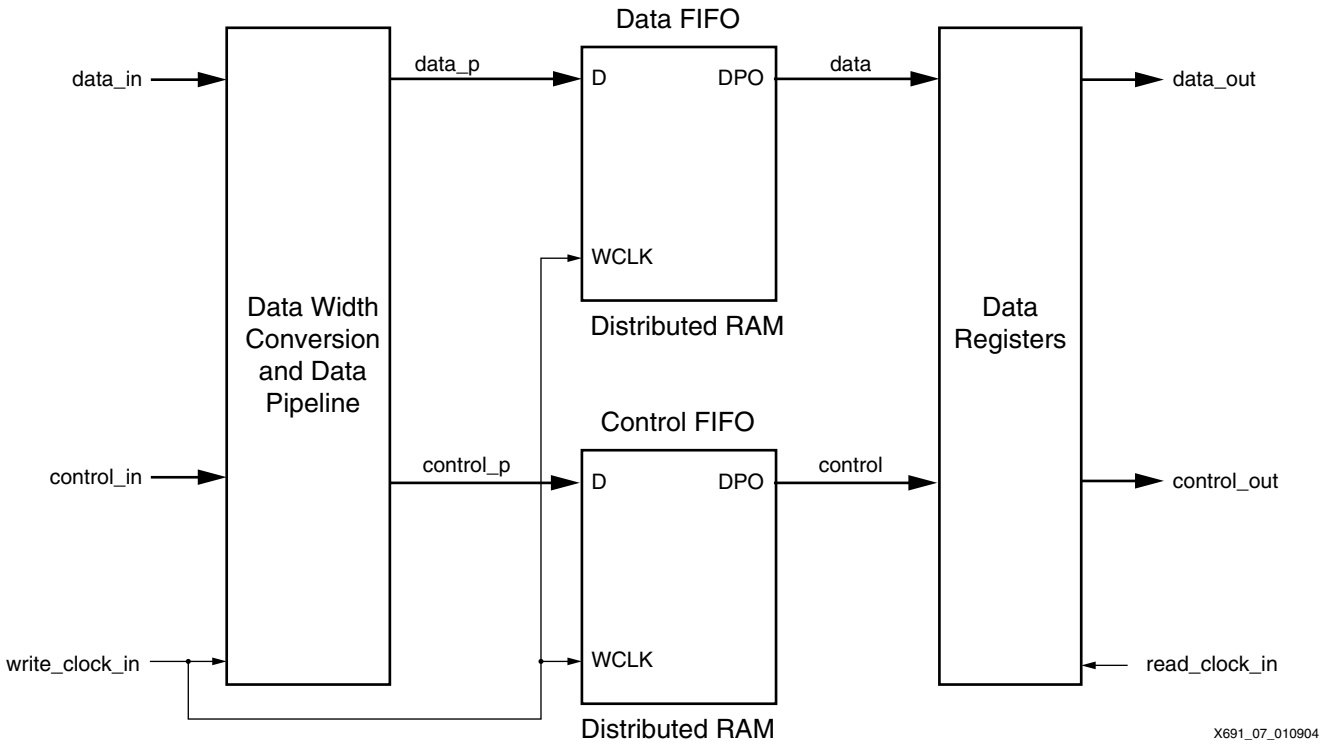

図 7： Data FIFO および Control FIFO (WR_DWIDTH < RD_DWIDTH) 用分散 RAM インプリメンテーション タイプ Ⅱ

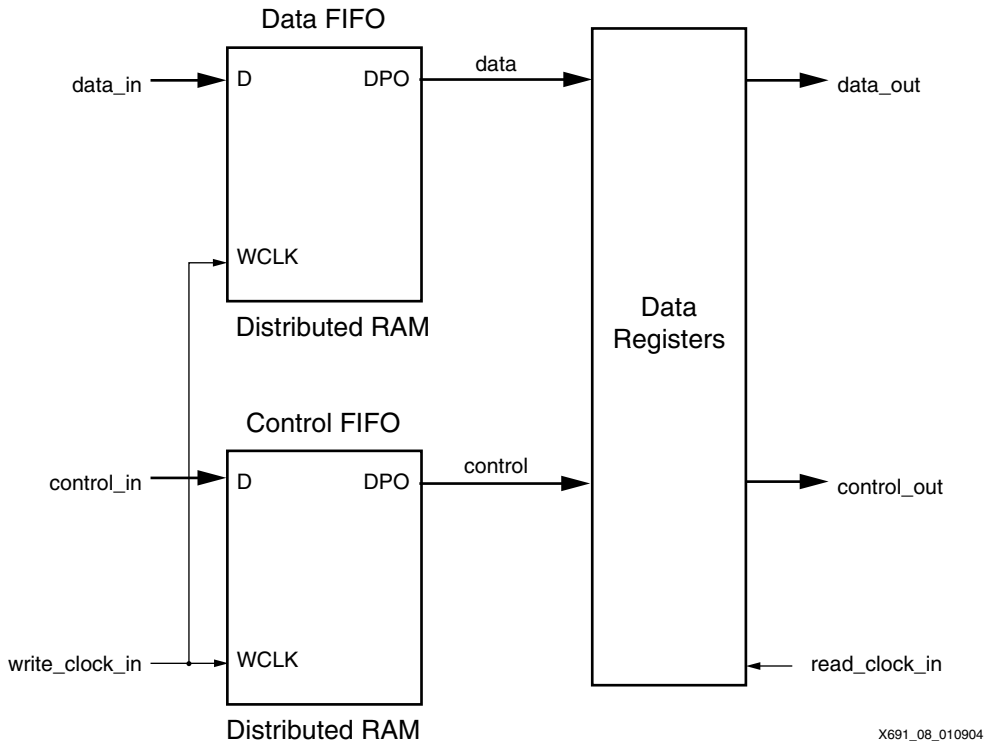

図 8 : Data FIFO および Control FIFO (WR_DWIDTH = RD_DWIDTH) 用分散 RAM インプリメンテーション タイプ Ⅲ

Length FIFO

Length FIFO は、各受信フレームのフレーム長をバイトで格納するためにオプションとして使用するブロックです。フレーム全体のバッファリングを行う前にフレーム長が必要となるダウンストリーム ユーザー アプリケーションでは、LocalLink FIFO の Length Ready (**len_rdy_out**) 信号をサンプルできます。**len_rdy_out** が High にアサートされた場合、フレーム全体が LocalLink FIFO に格納されたことを示し、フレーム長カウントは **len_out** バスでわかります。また、**len_rdy_out** 信号は各フレームに対して 1 サイクルのパルスです。

Length FIFO は、Data FIFO および Control FIFO に使用されたメモリによって、ブロック SelectRAM または分散 RAM でインプリメントできます。14 ビットに固定されたカウンタは、フレーム内で Data FIFO に書き込まれたデータ ワード数を加算します。また、フレーム長は、カウンタ値と各ワードのバイト数を乗算することによって求められます。フレーム全体が受信され、算出されたフレーム長が Length FIFO に入ると、Length Ready (**len_rdy_out**) 信号が LocalLink FIFO インターフェイスでアサートされます。この信号は、各受信フレームごとに 1 パルスです。ダウンストリーム ユーザー アプリケーションでは、この信号をサンプルし、インターフェイスで Frame Length 出力 (**len_out**) をラッチします。

Length FIFO では専用のメモリ、アドレス ポインタ、および関連ロジックを使用します。Length FIFO にあるポインタ管理ロジックは、Data FIFO および Control FIFO に対して独立しています。Length FIFO の各アドレスには、Data FIFO および Control FIFO のそれぞれに対応したフレーム長が格納されます。このように 1 つのデータに対して複数の位置が関係しているため、Length FIFO は Data FIFO および Control FIFO とポインタ管理ロジック (特に FIFO のステータスを認識させるためのロジック) を共有できません。インプリメンテーションを単純化し、サイズを減らすために、Length FIFO では FIFO ポインタ管理ロジックをインプリメントしません。たとえば、フルまたは空のステータス フラッグを使用して、書き込みおよび読み出し処理を行わないようにするという方法は使用しません。

(a) Length FIFO Self-Clear メカニズム、(b) Length FIFO オーバーフロー問題、および (c) 最大フレーム長で説明するように、Length FIFO を使用する場合には、いくつかの注意点と制限があります。

(a) Length FIFO Self-Clear メカニズム

ブロック SelectRAM メモリを使用して Length FIFO をインプリメントした場合、Length Ready フラグを保存するためにパリティ ビットが使用されます。前述したように、Length FIFO では読み出しポインタまたは書き込みポインタを保護させるためにフルまたは空のフラグを使用しません。そのため、FIFO の読み出しポインタが移動してしまう場合、その前の書き込みでメモリに残っているデータが不適切な長さの出力をトリガする可能性があります。これを避けるために、Length FIFO には FIFO に残っているデータをクリアにするための Self-Clear メカニズムが必要になります。算出されたフレーム長がインターフェイスに出力されると、Length FIFO は現在の読み出し位置にゼロを書き込み、次の読み出し位置へ進みます。このような機能を使用するためには、読み出しおよび書き込みの各ポートから同一のメモリ位置へアクセスできる完全なデュアル ポート メモリが必要であり、ブロック SelectRAM がこれに適しています。図 9 に示すように、ブロック SelectRAM のポート B を使用して FIFO から Length 値を読み出し、次のクロック サイクルで同じ位置にゼロを書き込みます。

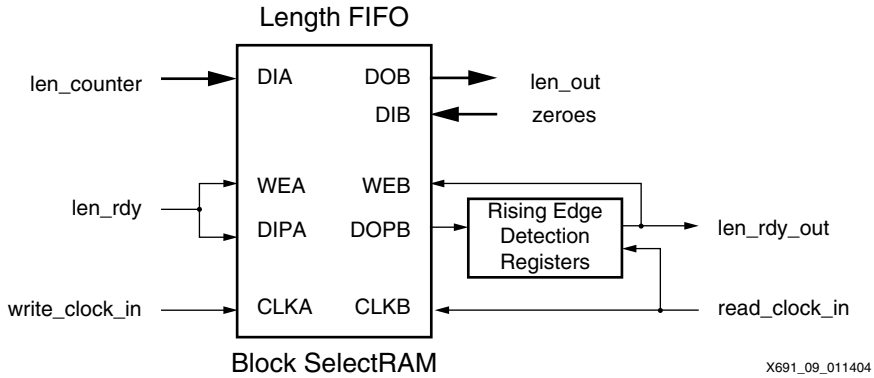

図 9： ブロック SelectRAM を使用した Length FIFO インプリメンテーション

ブロック SelectRAM と異なり、分散 RAM には 1 つの書き込みポートしかないため、Self-Clear ロジックをインプリメントすることはできません。したがって、分散 RAM の場合は、Length FIFO の長さに合わせたメモリを使用し、デザイン全体を簡潔化してください。Length FIFO によって Length 値が FIFO に書き込まれるごとに書き込みポインタが進み、次の有効な Length 値が書き込まれる前に、ゼロがメモリ位置に書き込まれます。ただし、このように機能するのは、書き込みクロック周波数が読み出しクロック周波数より高速の場合、または 2 つの周波数が一致している場合のみです。

書き込みクロック周波数より高い読み出しクロック周波数で分散 RAM をインプリメントした場合は、Length FIFO は有効な Length 値を出力しません。

(b) Length FIFO オーバーフロー問題

分散 RAM を使用した場合、Length FIFO のワード数はパラメータ設定が可能であり、つねに Data FIFO のワード数と等しくなります。Data FIFO がフルになるとポインタ管理ロジックによって、この Data FIFO への書き込みは停止します。前述したように、Length FIFO と Data FIFO には、1 つのアドレス位置に対して複数のアドレス位置が対応している関係があるため、Length FIFO がオーバーフローすることはありません。分散 RAM インプリメンテーションでは、LocalLink FIFO インターフェイス上の **len_err_out** 信号がつねにディアサートされます。

一方、ブロック SelectRAM を使用した場合、Length FIFO ではメカニズムを簡略化するために 1 つのブロック SelectRAM を使用することで、最大 1,024 個の Length カウンタ値を格納できます。Length FIFO のワード数はパラメータ設定できないため、FIFO のワード数が不足し、Data FIFO に格納された最大フレーム数の Length カウント値を格納できない場合があります。このため、Data FIFO のワード数が非常に大きく、受信フレームの平均的な長さが相対的に小さい場合には、Length FIFO のオーバーフローが発生しやすくなります。たとえば、受信フレーム長が平均 4 バイトであり、8 ビット幅の Data FIFO に 4,096 書き込みアドレス位置がある場合、Length FIFO でオーバーフローが起こる可能性があ

ります。Length FIFO のオーバーフローによって、LocalLink FIFO インターフェイス上の **len_err_out** 信号がアサートされ、Length 出力が無効であることを示します。このエラー状態を解除するためには、LocalLink FIFO をリセットする必要があります。

(c) 最大フレーム長

Length FIFO では 16,383 データ ワードまでをカウントする 14 ビット カウンタがつねに使用されます。また、データ ワード幅は WR_DWIDTH パラメータで設定します。LocalLink FIFO の Frame Length 出力は 16 ビット幅で一定であり、フレーム長を 65,535 バイトまでカウントできます。表 7 に Length FIFO でサポートされる最大フレーム長を示します。WR_DWIDTH が 8 の場合、Length FIFO は 16,385 バイトまでフレームをカウントでき、イーサネット アプリケーションのジャンボ フレームにも対応できます。受信フレーム長が LocalLink FIFO でサポートされている最大の長さを超える場合でも Frame Length 出力は 65,535 バイトになります。

表 7: Length FIFO がサポートする最大フレーム長

WR_DWIDTH (ビット)	最大フレーム長 (バイト)
8	16,385
16	32,770
32	65,535
64	65,535
128	65,535

LocalLink FIFO ステータス信号

LocalLink FIFO では FIFO ステータス信号 (**fifostatus_out**) を使用し、1/2 フル、1/4 フルなどの情報をアップストリーム アプリケーションに送ります。この信号は、書き込みクロック (**write_clock_in**) に同期であり、XAPP258 および XAPP261 に添付されてる FIFO デザインで使用されている FIFO ステータス信号と同一です。詳細については、これらのアプリケーション ノートを参照してください。

コンパイル パラメータ

表 8 に LocalLink FIFO のパラメータ値およびその説明を示します。

表 8: LocalLink FIFO パラメータ

パラメータ名	値	デフォルト値	説明
MEM_TYPE	整数 : 0 または 1	0	メモリタイプを選択します。 0 = ブロック SelectRAM 1 = 分散 RAM
DRAM_DEPTH	整数 : 16、32、64、128、256 または 512	16	分散 RAM インプリメンテーションでメモリ ワード数を設定します。 読み出しポートのデータ幅が書き込みポートのデータ幅以上の場合、このメモリ ワード数によって、読み出しポートでの最多アドレス位置を指定します。 読み出しポートのデータ幅が書き込みポートのデータ幅以下の場合、このメモリ ワード数によって、書き込みポートでの最多アドレス位置を指定します。たとえば、32 ビットの読み出しデータ ポートおよび 64 ビットの書き込みデータ ポートがあり、ワード数が 64 の LocalLink FIFO は 64X64 ビット メモリ ブロックを使用してインプリメントされます。
BRAM_MACRO_NUM	整数 : 1、2、4、8 または 16	1	ブロック SelectRAM マクロ数を指定し、ブロック SelectRAM インプリメンテーションでのメモリ ワード数を設定します。表 6 に示すように、データ幅およびこのパラメータによって実際の FIFO ワード数が決定されます。
WR_DWIDTH	整数 : 8、16、32、64 または 128	32	書き込みポートのデータ幅を選択します。
RD_DWIDTH	整数 : 8、16、32、64 または 128	32	読み出しポートのデータ幅を選択します。
WR_REM_WIDTH	整数 : 1、2、3 または 4	2	書き込みポートにおける余りの部分のデータ幅を選択します。 WR_DWIDTH = 8 の場合、このパラメータを 1 に設定する必要があります。それ以外の場合は $\log_2$(WR_DWDITH/8) を使用して、値が算出されます。
RD_REM_WIDTH	整数 : 1、2、3 または 4	2	読み出しポートにおける余りの部分のデータ幅を選択します。 RD_DWIDTH = 8 の場合、このパラメータを 1 に設定する必要があります。それ以外の場合は $\log_2$(RD_DWDITH/8) を使用して、値が算出されます。
USE_LENGTH	ブール : TRUE または FALSE	TRUE	TRUE の場合、オプションとしての Length FIFO をインプリメントします。FALSE の場合は、Length FIFO がインプリメンテーションから削除され、LocalLink FIFO にある Length 信号がゼロに接続されます。

波形

図 10 に書き込みデータ幅が 32 (WR_DWIDTH = 32) であり、読み出しデータ幅が 16 (RD_DWIDTH = 16) の LocalLink FIFO での基本的なフレーム送信を示します。**sof_in_n** および **src_rdy_in_n** をアサートし、アップストリーム ユーザー アプリケーションでのデータ送信を開始します。LocalLink FIFO にはフレームをバッファリングできる領域があるため、**dst_rdy_out_n** がアサートされます。この場合、ダウンストリーム アプリケーションでは、LocalLink FIFO からのデータ出力を受信するために **dst_rdy_in_n** をつねにアサートします。データ幅変換および余りの部分のデータ変換は、LocalLink FIFO 内部で行われることに注意してください。

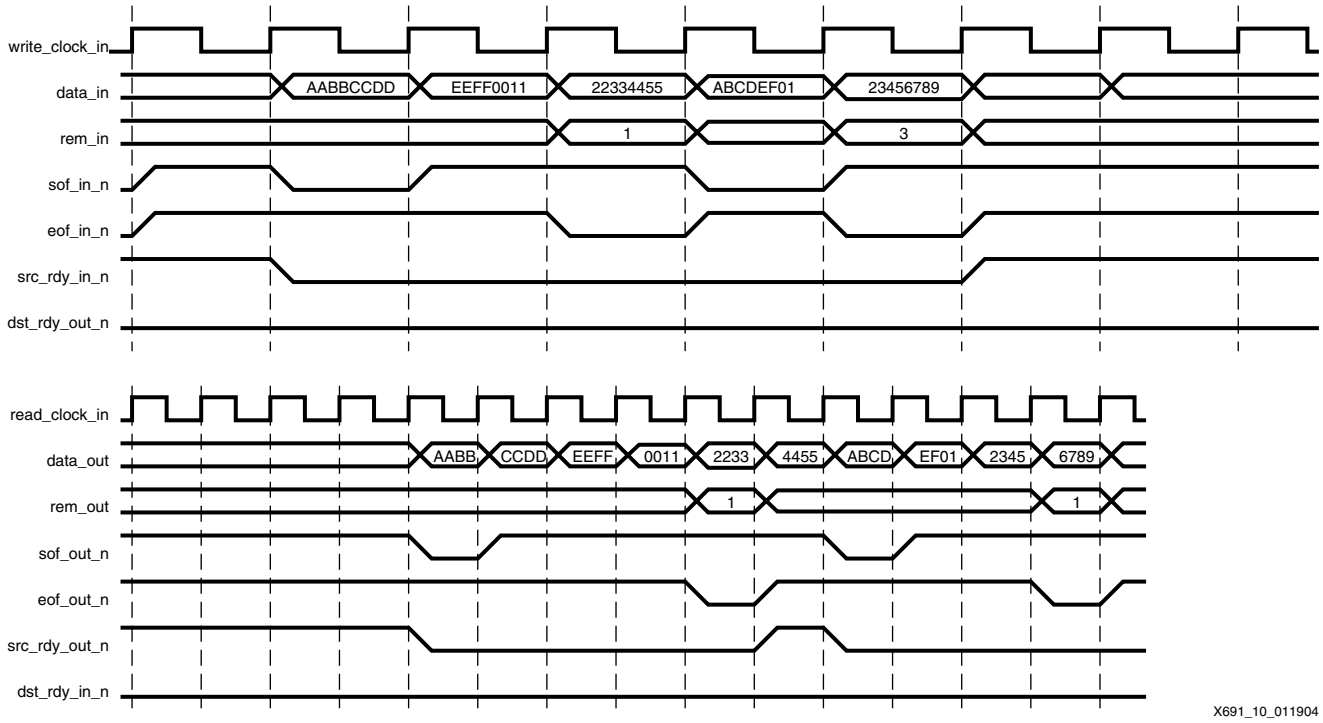

図 10： LocalLink FIFO での基本的なフレーム送信

図 11 にアップストリーム ユーザー アプリケーションと LocalLink FIFO 間にフロー制御がある場合の基本的なフレーム送信を示します。LocalLink FIFO がフルであり、アップストリーム ユーザー アプリケーションからの新しいデータを受信できる状態にない場合、**dst_rdy_out_n** がディアサートに維持されます。アップストリーム ユーザー アプリケーションでは、この **dst_rdy_out_n** が LocalLink FIFO でアサートされるまで、次のデータ バイトを保持します。つまり、LocalLink FIFO がデータ受信できる以前の状態のときのデータが失われることはありません。**data_out** バスは 1 バイト幅であり、余りのデータ値を必要としないため、図 11 に **rem_out** 信号は記載されていません。

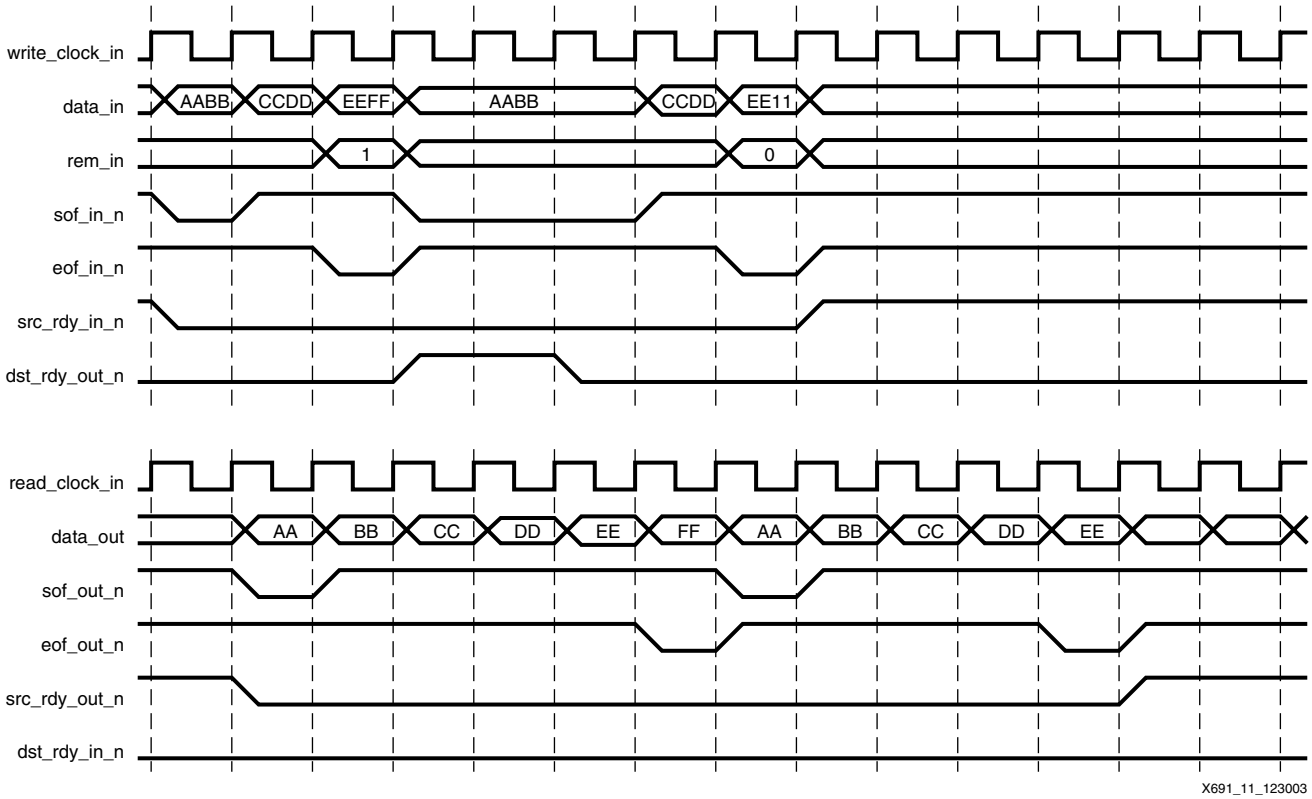

図 11： アップストリーム インターフェイスでフロー制御がある場合の LocalLink FIFO でのフレーム送信

図 12 にダウンストリーム ユーザー アプリケーションと LocalLink FIFO 間にフロー制御がある場合の基本的なフレーム送信を示します。ダウンストリーム ユーザー アプリケーションが LocalLink FIFO からの新しいデータを受信できる状態にない場合、**dst_rdy_in_n** がディアサートに維持されます。LocalLink FIFO は、ダウンストリーム ユーザー アプリケーションで **dst_rdy_in_n** がアサートされるまで次のデータ バイトを保持します。つまり、ダウンストリーム アプリケーションがデータ受信できる以前の状態のときのデータが失われることはありません。

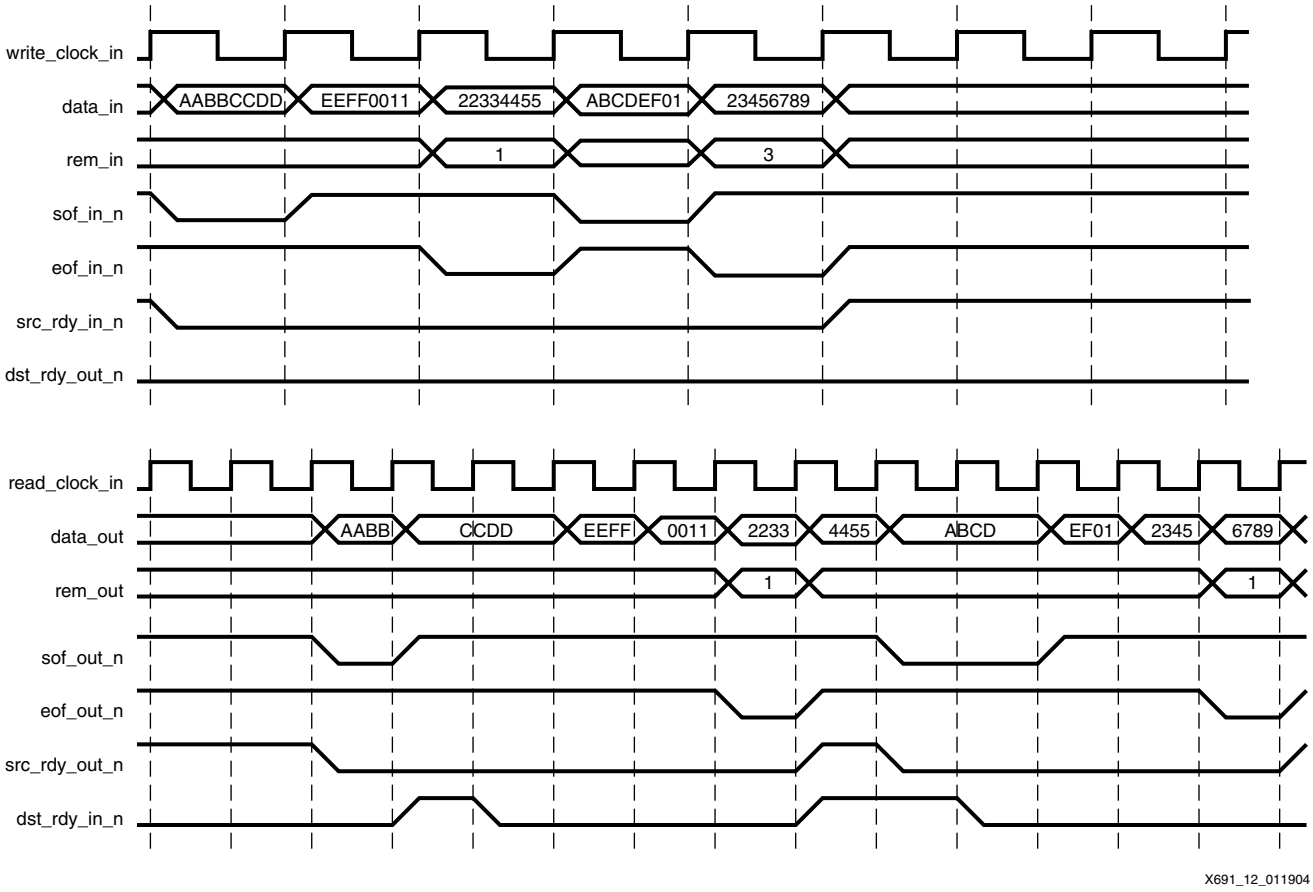

図 12： ダウンストリーム インターフェイスでフロー制御がある場合の LocalLink FIFO でのフレーム送信

図 13 に LocalLink FIFO での Length 出力に基づくフレーム送信を示します。ダウンストリーム ユーザー アプリケーションでは、**len_rdy_out** 出力をサンプルして、LocalLink FIFO からの **len_out** バス値をラッチし、フレーム データ読み出し前に LocalLink FIFO に格納された各フレームのバイト長を獲得します。ダウンストリーム ユーザー アプリケーションで先に受信されたフレームが処理されている場合、すべての Length 出力を格納できます。Length 値の出力をダウンストリーム ユーザー アプリケーションで維持することはできません。

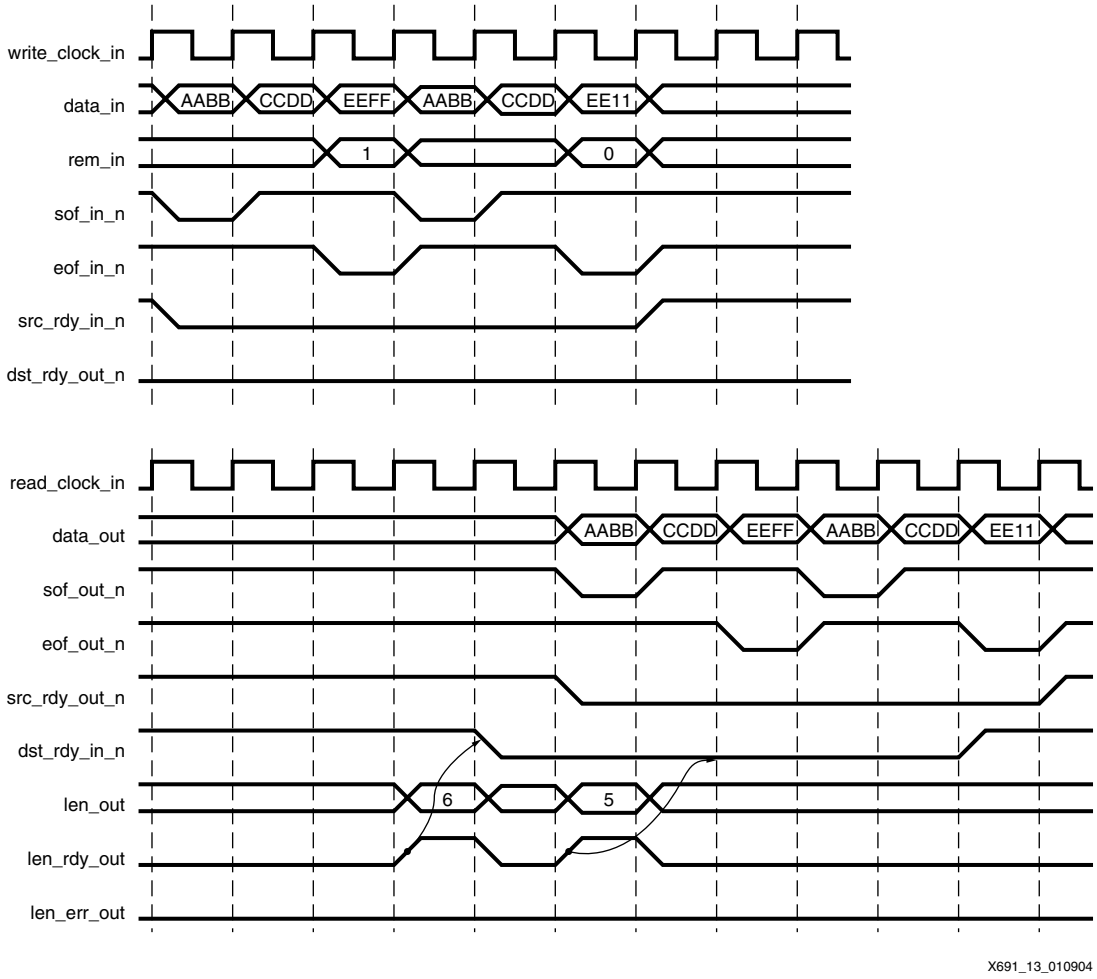

図 13： Length 出力に基づく LocalLink FIFO でのフレーム送信

検証

シミュレーション

図 14 に LocalLink FIFO を検証するためのシミュレーション テストベンチを示します。テストベンチには、出口となる LocalLink FIFO および入り口となる LocalLink FIFO の 2 つがインスタンシエートされています。テスト モジュールは、クロック ドメイン A で 2 つの FIFO の片側に接続されています。また、これらの FIFO はクロック ドメイン B を使用し、もう片側で互いに直接接続されています。テスト モジュールは、テスト ベクタ ファイルを読み込むことによって、フレーム データおよび制御データのベクタを出力し、出口となる FIFO にある LocalLink インターフェイスに信号を送ります。また、テスト モジュールは入口となる FIFO からのデータを読み出し、予想されているデータと比較して、テスト結果を作成します。LocalLink FIFO のシミュレーションには Mentor Graphics ModelSim を使用しました。

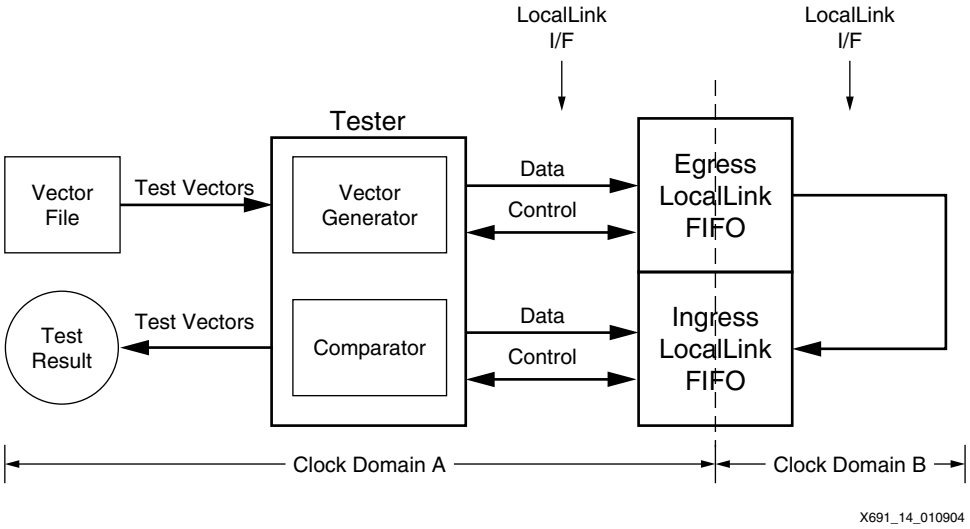

図 14： LocalLink FIFO シミュレーション テストベンチ

ハードウェア テスト

LocalLink FIFO ハードウェア テストは、ML320 プラットフォーム上で ChipScope Integrated Logic Analyzer (ILA) コア、およびそのソフトウェアを使用して行い、LocalLink FIFO のアップストリーム インターフェイスに接続されたパターン ジェネレータによって、インクリメンタルなフレーム長を持つフレーム データが生成されました。LocalLink FIFO のダウンストリーム インターフェイス信号は、ChipScope ILA コアに入り、ChipScope ソフトウェアを使用して検証されています。LocalLink FIFO はパラメータ設定が可能であるため、テストは一般的なデータ幅およびメモリ タイプでのみ行っています。ブロック SelectRAM および分散 RAM でのテストは、任意の読み出し幅と 8 ビット、32 ビット、および 128 ビットの書き込み幅で実施し、読み出しクロック周波数および書き込みクロック周波数は、50 MHz から 125 MHz です。

リソースの使用とパフォーマンス

LocalLink FIFO のリソース使用は、読み出しポートおよび書き込みポートにおけるデータ幅、使用するメモリ タイプ、および Length FIFO の有無の組み合わせによって異なります。次の表に、これらを組み合わせた場合のリソース使用とパフォーマンスを示します。

表 9: ブロック SelectRAM インプリメンテーション (Length FIFO なし) に使用したリソース (FF, LUT, BRAM)

書き込みデータ幅	読み出しデータ幅	BRAM_MACRO_NUM				
		1	2	4	8	16
8	8	(95, 59, 2)	(102, 74, 3)	(108, 88, 5)	(116, 114, 9)	(136, 178, 20)
8	16	(99, 63, 2)	(105, 86, 3)	(112, 111, 5)	(121, 155, 9)	(133, 263, 18)
8	32	(113, 70, 2)	(119, 112, 3)	(128, 154, 5)	(133, 233, 9)	(160, 406, 17)
8	64	(139, 82, 3)	(145, 148, 5)	(154, 214, 9)	(171, 356, 20)	(250, 644, 40)
8	128	(205, 108, 5)	(211, 238, 9)	(227, 367, 17)	(250, 634, 34)	(330, 1164, 68)
16	8	(89, 59, 2)	(95, 73, 3)	(101, 87, 5)	(109. 115. 10)	(130. 182. 20)
16	16	(100, 57, 2)	(106, 82, 3)	(113, 106, 5)	(121, 152, 9)	(135, 249, 17)
16	32	(112, 67, 2)	(118, 107, 3)	(125, 149, 5)	(133, 227, 9)	(156, 396, 17)
16	64	(140, 80, 3)	(146, 150, 5)	(155, 222, 9)	(158, 360, 17)	(226, 659, 34)
16	128	(204, 105, 5)	(210, 235, 9)	(225, 366, 17)	(247, 632, 34)	(326, 1162, 68)
32	8	(83, 56, 2)	(89, 71, 3)	(95, 85, 5)	(103, 113, 10)	(120, 180, 20)
32	16	(94, 55, 2)	(100, 79, 3)	(107, 104, 5)	(114, 149, 9)	(128, 249, 17)
32	32	(103, 59, 1)	(111, 103, 2)	(123, 145, 4)	(137, 226, 8)	(157, 396, 16)
32	64	(139, 77, 3)	(145, 145, 5)	(155, 215, 9)	(166, 349, 17)	(229, 641, 34)
32	128	(203, 119, 5)	(210, 276, 9)	(218, 410, 17)	(230, 706, 34)	(247, 1300, 68)
64	8	(82, 57, 3)	(88, 72, 5)	(96, 91, 10)	(125, 123, 20)	(149, 192, 40)
64	16	(93, 57, 3)	(99, 82, 5)	(106, 106, 9)	(113, 151, 17)	(147, 258, 34)
64	32	(104, 61, 2)	(112, 102, 4)	(124, 146, 8)	(130, 227, 16)	(171, 400, 32)
64	64	(132, 69, 2)	(140, 142, 4)	(155, 212, 8)	(167, 353, 16)	(223, 640, 32)
64	128	(198, 103, 4)	(207, 243, 8)	(233, 386, 16)	(247, 665, 32)	(357, 1232, 64)
128	8	(81, 59, 5)	(89, 75, 10)	(119, 95, 20)	(124, 124, 40)	(157, 193, 80)
128	16	(88, 58, 5)	(96, 83, 10)	(122, 111, 20)	(154, 158, 40)	(190, 261, 80)
128	32	(109, 63, 5)	(115, 104, 9)	(122, 145, 17)	(152, 228, 34)	(208, 400, 68)
128	64	(133, 71, 4)	(142, 141, 8)	(158, 215, 16)	(178, 354, 32)	(246, 646, 64)
128	128	(197, 91, 4)	(205, 228, 8)	(223, 365, 16)	(261, 635, 32)	(369, 1186, 64)

表 10: ブロック SelectRAM インプリメンテーション (Length FIFO あり) に使用したリソース (FF, LUT, BRAM)

書き込み データ幅	読み出し データ幅	BRAM_MACRO_NUM				
		1	2	4	8	16
8	8	(145, 111, 3)	(151, 125, 4)	(157, 139, 6)	(165, 163, 10)	(173, 231, 21)
8	16	(149, 114, 3)	(155, 137, 4)	(161, 162, 6)	(167, 207, 10)	(175, 315, 19)
8	32	(163, 120, 3)	(169, 162, 4)	(175, 204, 6)	(181, 286, 10)	(187, 458, 18)
8	64	(189, 133, 4)	(195, 199, 6)	(201, 265, 10)	(209, 408, 21)	(217, 691, 41)
8	128	(255, 159, 6)	(261, 289, 10)	(267, 418, 18)	(275, 684, 35)	(284, 1216, 69)
16	8	(137, 108, 3)	(143, 122, 4)	(149, 136, 6)	(157, 163, 11)	(165, 228, 21)
16	16	(148, 106, 3)	(154, 130, 4)	(160, 154, 6)	(166, 198, 10)	(172, 298, 18)
16	32	(160, 116, 3)	(166, 156, 4)	(172, 198, 6)	(178, 277, 10)	(184, 445, 18)
16	64	(188, 128, 4)	(194, 198, 6)	(200, 270, 10)	(206, 410, 18)	(215, 705, 35)
16	128	(252, 154, 6)	(258, 284, 10)	(264, 414, 18)	(272, 678, 35)	(281, 1211, 69)
32	8	(129, 105, 3)	(135, 120, 4)	(141, 134, 6)	(149, 162, 11)	(166, 230, 21)
32	16	(140, 104, 3)	(146, 128, 4)	(152, 153, 6)	(161, 197, 10)	(174, 299, 18)
32	32	(149, 107, 2)	(157, 149, 3)	(165, 191, 5)	(173, 271, 9)	(193, 444, 17)
32	64	(185, 125, 4)	(191, 193, 6)	(197, 264, 10)	(203, 398, 18)	(252, 689, 35)
32	128	(249, 169, 6)	(256, 324, 10)	(264, 460, 18)	(276, 755, 35)	(293, 1347, 69)
64	8	(126, 104, 4)	(132, 118, 6)	(140, 137, 11)	(150, 167, 21)	(187, 237, 41)
64	16	(137, 103, 4)	(143, 128, 6)	(149, 152, 10)	(157, 198, 18)	(191, 305, 35)
64	32	(148, 107, 3)	(156, 148, 5)	(165, 192, 9)	(175, 273, 17)	(213, 445, 33)
64	64	(176, 114, 3)	(184, 185, 5)	(192, 256, 9)	(200, 394, 17)	(240, 682, 33)
64	128	(242, 148, 5)	(250, 288, 9)	(268, 430, 17)	(285, 710, 33)	(401, 1278, 65)
128	8	(123, 102, 6)	(131, 118, 11)	(140, 138, 21)	(166, 167, 41)	(199, 235, 81)
128	16	(130, 101, 6)	(138, 125, 11)	(146, 152, 21)	(178, 198, 41)	(233, 304, 81)
128	32	(151, 105, 6)	(157, 146, 10)	(163, 187, 18)	(183, 267, 35)	(224, 440, 69)
128	64	(175, 113, 5)	(183, 183, 9)	(195, 257, 17)	(209, 395, 33)	(276, 686, 65)
128	128	(239, 132, 5)	(247, 269, 9)	(255, 405, 17)	(270, 672, 33)	(342, 1221, 65)

表 11: ブロック SelectRAM インプリメンテーション (Length FIFO なし) の場合の XC2VP7 -6 パフォーマンス

書き込み データ幅	読み出し データ幅	BRAM_MACRO_NUM				
		1	2	4	8	16
8	8	240 MHz	236 MHz	197 MHz	202 MHz	191 MHz
8	16	179 MHz	190 MHz	176 MHz	168 MHz	142 MHz
8	32	262 MHz	231 MHz	223 MHz	185 MHz	183 MHz
8	64	253 MHz	251 MHz	219 MHz	198 MHz	168 MHz
8	128	247 MHz	210 MHz	203 MHz	169 MHz	169 MHz
16	8	251 MHz	214 MHz	170 MHz	177 MHz	174 MHz
16	16	205 MHz	204 MHz	204 MHz	210 MHz	200 MHz
16	32	256 MHz	254 MHz	198 MHz	182 MHz	194 MHz
16	64	272 MHz	231 MHz	207 MHz	179 MHz	169 MHz
16	128	247 MHz	209 MHz	181 MHz	168 MHz	168 MHz
32	8	242 MHz	211 MHz	251 MHz	191 MHz	183 MHz
32	16	175 MHz	171 MHz	173 MHz	173 MHz	151 MHz
32	32	255 MHz	272 MHz	245 MHz	209 MHz	182 MHz
32	64	226 MHz	208 MHz	188 MHz	170 MHz	170 MHz
32	128	215 MHz	179 MHz	172 MHz	164 MHz	164 MHz
64	8	272 MHz	238 MHz	176 MHz	187 MHz	169 MHz
64	16	187 MHz	188 MHz	181 MHz	176 MHz	153 MHz
64	32	192 MHz	238 MHz	173 MHz	188 MHz	173 MHz
64	64	262 MHz	286 MHz	228 MHz	188 MHz	170 MHz
64	128	242 MHz	214 MHz	192 MHz	170 MHz	170 MHz
128	8	245 MHz	180 MHz	175 MHz	168 MHz	168 MHz
128	16	207 MHz	172 MHz	171 MHz	150 MHz	150 MHz
128	32	246 MHz	203 MHz	200 MHz	172 MHz	172 MHz
128	64	190 MHz	250 MHz	210 MHz	168 MHz	168 MHz
128	128	256 MHz	240 MHz	193 MHz	170 MHz	170 MHz

表 12: ブロック SelectRAM インプリメンテーション (Length FIFO あり) の場合の XC2VP7 -6 パフォーマンス

書き込み データ幅	読み出し データ幅	BRAM_MACRO_NUM				
		1	2	4	8	16
8	8	172 MHz	170 MHz	171 MHz	169 MHz	169 MHz
8	16	168 MHz	170 MHz	171 MHz	167 MHz	143 MHz
8	32	173 MHz	169 MHz	170 MHz	169 MHz	169 MHz
8	64	169 MHz	174 MHz	170 MHz	167 MHz	168 MHz
8	128	168 MHz	173 MHz	174 MHz	170 MHz	170 MHz
16	8	168 MHz	169 MHz	168 MHz	169 MHz	168 MHz
16	16	170 MHz	169 MHz	177 MHz	169 MHz	173 MHz
16	32	169 MHz	173 MHz	169 MHz	169 MHz	167 MHz
16	64	168 MHz	171 MHz	171 MHz	169 MHz	168 MHz
16	128	171 MHz	175 MHz	170 MHz	172 MHz	172 MHz
32	8	215 MHz	179 MHz	181 MHz	187 MHz	178 MHz
32	16	181 MHz	190 MHz	173 MHz	168 MHz	152 MHz
32	32	182 MHz	196 MHz	188 MHz	187 MHz	177 MHz
32	64	225 MHz	184 MHz	205 MHz	175 MHz	167 MHz
32	128	192 MHz	182 MHz	169 MHz	161 MHz	161 MHz
64	8	194 MHz	184 MHz	205 MHz	190 MHz	168 MHz
64	16	195 MHz	174 MHz	177 MHz	170 MHz	151 MHz
64	32	183 MHz	180 MHz	171 MHz	173 MHz	168 MHz
64	64	226 MHz	205 MHz	195 MHz	177 MHz	169 MHz
64	128	245 MHz	215 MHz	170 MHz	168 MHz	168 MHz
128	8	189 MHz	221 MHz	175 MHz	169 MHz	169 MHz
128	16	217 MHz	169 MHz	174 MHz	166 MHz	166 MHz
128	32	243 MHz	235 MHz	168 MHz	177 MHz	177 MHz
128	64	169 MHz	176 MHz	182 MHz	172 MHz	172 MHz
128	128	200 MHz	185 MHz	177 MHz	169 MHz	169 MHz

表 13: 分散 RAM インプリメンテーション (Length FIFO なし) に使用したリソース (FF, LUT, BRAM)

書き込み データ幅	読み出し データ幅	ワード数					
		16	32	64	128	256	512
8	8	49, 52	70, 80	111, 132	175, 230	267, 452	514, 897
8	16	72, 81	106, 129	181, 216	293, 399	456, 809	920, 1646
8	32	112, 127	170, 210	311, 368	513, 709	824, 1483	1697, 3064
8	64	177, 202	281, 350	542, 630	902, 1236	1450, 2631	3028, 5487
8	128	42, 46	51, 60	63, 71	84, 97	113, 148	89, 242
16	8	51, 0	78, 128	161, 217	230, 395	191, 804	379, 1650
16	16	74, 76	101, 122	176, 209	287, 392	451, 808	915, 1650
16	32	110, 124	166, 207	307, 365	509, 706	820, 1479	1693, 3065
16	64	174, 197	275, 345	534, 625	896, 1230	1444, 2625	3025, 5486
16	128	319, 359	511, 647	1021, 1195	1739, 2404	2836, 5194	5988, 10939
32	8	62, 126	80, 207	175, 360	157, 690	257, 1467	477, 3069
32	16	74, 128	117, 211	267, 371	315, 694	272, 1475	792, 3094
32	32	119, 126	161, 200	301, 358	503, 698	815, 1487	1689, 3091
32	64	170, 194	271, 343	530, 622	892, 1227	1440, 2622	3021, 5487
32	128	313, 353	505, 641	1013, 1190	1732, 2401	2830, 5188	5982, 10932
64	8	75, 215	90, 359	150, 644	245, 1275	427, 2761	815, 5829
64	16	93, 217	117, 360	303, 645	261, 1273	441, 2757	859, 5834
64	32	126, 222	193, 373	483, 671	545, 1282	475, 2778	891, 5873
64	64	194, 207	266, 335	525, 614	887, 1222	1436, 2640	3019, 5538
64	128	309, 350	500, 638	1009, 1187	1728, 2398	2826, 5185	5978, 10929
128	8	81, 387	130, 661	233, 1206	411, 2439	770, 5322	1545, 11289
128	16	112, 392	145, 662	249, 1209	423, 2432	780, 5309	1555, 11284
128	32	153, 394	178, 661	482, 1209	457, 2424	814, 5304	1625, 11281
128	64	209, 382	332, 649	858, 1189	972, 2298	841, 5027	1607, 10692
128	128	366, 372	493, 632	1005, 1180	1720, 2389	2823, 5227	5979, 11035

表 14: 分散 RAM インプリメンテーション (Length FIFO あり) に使用したリソース (FF, LUT, BRAM)

書き込み データ幅	読み出し データ幅	ワード数					
		16	32	64	128	256	512
8	8	56, 61	65, 93	76, 147	90, 256	130, 496	170, 1003
8	16	80, 87	89, 139	100, 229	114, 419	172, 847	228, 1762
8	32	119, 131	128, 217	139, 375	157, 718	251, 1503	351, 3184
8	64	183, 205	192, 349	207, 623	233, 1227	391, 2621	571, 5605
8	128	323, 370	336, 648	359, 1183	391, 2371	698, 5131	1032, 11043
16	8	58, 91	67, 141	78, 231	110, 411	156, 849	303, 1765
16	16	73, 81	82, 131	93, 221	107, 411	169, 840	220, 1752
16	32	115, 128	124, 214	135, 372	153, 715	247, 1500	347, 3180
16	64	177, 199	186, 343	201, 617	227, 1221	385, 2615	565, 5601
16	128	313, 358	326, 632	349, 1165	381, 2354	688, 5114	1022, 11026
32	8	67, 135	90, 224	187, 385	177, 724	281, 1526	507, 3178
32	16	76, 136	106, 225	276, 390	320, 729	296, 1531	526, 3201
32	32	109, 121	132, 207	275, 371	352, 711	326, 1531	535, 3174
32	64	174, 196	211, 340	534, 618	597, 1220	244, 2679	907, 5598
32	128	311, 351	377, 625	1009, 1167	1108, 2350	1007, 5241	1677, 11023
64	8	83, 226	100, 379	162, 670	265, 1308	451, 2820	843, 5929
64	16	93, 226	127, 379	315, 671	281, 1307	465, 2816	887, 5973
64	32	112, 226	161, 384	420, 683	551, 1318	796, 2831	917, 5960
64	64	168, 189	204, 33	462, 617	891, 1213	543, 2673	900, 5594
64	128	307, 351	399, 631	1012, 1181	1104, 2348	1003, 5238	1673, 11021
128	8	88, 404	138, 681	245, 1231	425, 2468	789, 5376	1573, 11397
128	16	118, 402	155, 684	261, 1237	443, 2466	803, 5363	1586, 11391
128	32	145, 400	209, 685	494, 1238	477, 2458	838, 5363	1655, 11390
128	64	182, 380	266, 657	665, 1180	964, 2333	864, 5074	1611, 10768
128	128	300, 345	394, 623	800, 1156	1100, 2341	1001, 5232	1661, 11013

表 15: 分散 RAM インプリメンテーション (Length FIFO なし) の場合の XC2VP7 -6 パフォーマンス

書き込み データ幅	読み出し データ幅	ワード数					
		16	32	64	128	256	512
8	8	231 MHz	259 MHz	218 MHz	177 MHz	171 MHz	167 MHz
8	16	224 MHz	236 MHz	187 MHz	173 MHz	167 MHz	168 MHz
8	32	206 MHz	206 MHz	194 MHz	175 MHz	167 MHz	167 MHz
8	64	228 MHz	184 MHz	172 MHz	168 MHz	167 MHz	167 MHz
8	128	244 MHz	249 MHz	243 MHz	194 MHz	178 MHz	172 MHz
16	8	270 MHz	233 MHz	191 MHz	173 MHz	167 MHz	168 MHz
16	16	212 MHz	219 MHz	187 MHz	174 MHz	169 MHz	168 MHz
16	32	219 MHz	210 MHz	190 MHz	170 MHz	168 MHz	168 MHz
16	64	200 MHz	172 MHz	176 MHz	170 MHz	168 MHz	161 MHz
16	128	200 MHz	169 MHz	169 MHz	168 MHz	167 MHz	167 MHz
32	8	280 MHz	215 MHz	173 MHz	167 MHz	167 MHz	149 MHz
32	16	233 MHz	207 MHz	176 MHz	172 MHz	168 MHz	164 MHz
32	32	217 MHz	221 MHz	171 MHz	169 MHz	168 MHz	168 MHz
32	64	212 MHz	185 MHz	168 MHz	167 MHz	167 MHz	146 MHz
32	128	198 MHz	171 MHz	168 MHz	168 MHz	167 MHz	167 MHz
64	8	235 MHz	179 MHz	170 MHz	167 MHz	146 MHz	139 MHz
64	16	220 MHz	196 MHz	162 MHz	167 MHz	149 MHz	138 MHz
64	32	189 MHz	202 MHz	173 MHz	169 MHz	167 MHz	131 MHz
64	64	194 MHz	178 MHz	173 MHz	168 MHz	168 MHz	167 MHz
64	128	205 MHz	171 MHz	168 MHz	167 MHz	167 MHz	167 MHz
128	8	174 MHz	173 MHz	168 MHz	152 MHz	132 MHz	132 MHz
128	16	194 MHz	169 MHz	169 MHz	144 MHz	140 MHz	140 MHz
128	32	192 MHz	168 MHz	170 MHz	167 MHz	136 MHz	136 MHz
128	64	233 MHz	180 MHz	167 MHz	168 MHz	152 MHz	152 MHz
128	128	197 MHz	197 MHz	168 MHz	169 MHz	167 MHz	167 MHz

表 16: 分散 RAM インプリメンテーション (Length FIFO あり) の場合の XC2VP7 -6 パフォーマンス

書き込み データ幅	読み出し データ幅	ワード数					
		16	32	64	128	256	512
8	8	268 MHz	259 MHz	190 MHz	184 MHz	171 MHz	171 MHz
8	16	238 MHz	221 MHz	198 MHz	186 MHz	172 MHz	167 MHz
8	32	194 MHz	183 MHz	177 MHz	169 MHz	169 MHz	168 MHz
8	64	179 MHz	206 MHz	172 MHz	170 MHz	167 MHz	167 MHz
8	128	188 MHz	173 MHz	170 MHz	168 MHz	168 MHz	168 MHz
16	8	228 MHz	242 MHz	191 MHz	171 MHz	167 MHz	163 MHz
16	16	202 MHz	198 MHz	166 MHz	179 MHz	166 MHz	168 MHz
16	32	187 MHz	203 MHz	170 MHz	169 MHz	169 MHz	167 MHz
16	64	232 MHz	192 MHz	171 MHz	168 MHz	168 MHz	167 MHz
16	128	196 MHz	175 MHz	171 MHz	168 MHz	168 MHz	168 MHz
32	8	229 MHz	197 MHz	188 MHz	168 MHz	168 MHz	157 MHz
32	16	199 MHz	202 MHz	177 MHz	169 MHz	167 MHz	131 MHz
32	32	181 MHz	224 MHz	172 MHz	168 MHz	168 MHz	168 MHz
32	64	184 MHz	178 MHz	170 MHz	168 MHz	166 MHz	167 MHz
32	128	205 MHz	182 MHz	169 MHz	168 MHz	167 MHz	167 MHz
64	8	212 MHz	177 MHz	169 MHz	167 MHz	138 MHz	128 MHz
64	16	233 MHz	178 MHz	169 MHz	167 MHz	147 MHz	123 MHz
64	32	199 MHz	191 MHz	172 MHz	167 MHz	168 MHz	167 MHz
64	64	178 MHz	203 MHz	170 MHz	168 MHz	168 MHz	167 MHz
64	128	171 MHz	172 MHz	169 MHz	168 MHz	166 MHz	166 MHz
128	8	173 MHz	168 MHz	168 MHz	140 MHz	129 MHz	129 MHz
128	16	174 MHz	152 MHz	143 MHz	143 MHz	110 MHz	110 MHz
128	32	193 MHz	178 MHz	146 MHz	144 MHz	144 MHz	144 MHz
128	64	211 MHz	144 MHz	149 MHz	144 MHz	143 MHz	143 MHz
128	128	145 MHz	148 MHz	144 MHz	145 MHz	143 MHz	143 MHz

アプリケーション

LocalLink FIFO は、読み出しポートおよび書き込みポートのデータ幅にパラメータ設定が可能であり、最高 200 MHz で動作するため、10/100/1000 Mbps イーサネット、1G Fibre Channel と 2G Fibre Channel、Aurora、10 ギガビット イーサネット、および 10 ギガビット Fibre Channel のようにさまざまなアプリケーションに使用できます。表 17 および表 18 に LocalLink FIFO を使用した 1 ギガビット イーサネットおよび 10 ギガビット イーサネット アプリケーションを示します。

表 17: LocalLink FIFO を使用した 1 ギガビット イーサネット アプリケーション (ブロック SelectRAM)

データ幅(1) (ビット)	最大 FIFO 容量 (バイト)	クロック周波数 (MHz)		最大パケット数 (3)	最小パケット数 (4)	最大ジャンボ フレーム数(5)
		必要周波数	動作周波数(2)			
8	32,767	125	191	511	21	3.6
16	32,767	62.5	200	511	21	3.6
32	32,767	31.25	182	511	21	3.6
64	65,535	15.625	170	1023	43	7.3
128	131,071	7.8125	170	1023	86	14.6

メモ：

1. 読み出しおよび書き込みデータ幅が一致している場合
2. 表 11 に示したデータを使用
3. 最小イーサネット フレーム サイズ 64 バイト (CRC バイトを含む) を使用して測定
4. 最大イーサネット フレーム サイズ 1518 バイト (CRC バイトを含む) を使用して測定
5. 一般的なジャンボ フレーム サイズ 9000 バイトを使用して測定

表 18: LocalLink FIFO を使用した 10 ギガビット イーサネット アプリケーション (ブロック SelectRAM)

データ幅(1) (ビット)	最大 FIFO 容量 (バイト)	クロック周波数 (MHz)		最大パケット数 (3)	最小パケット数 (4)	最大ジャンボ フレーム数(5)
		必要周波数	動作周波数(2)			
64	65,535	156.25	170	1023	43	7.3
128	131,071	78.125	170	1023	86	14.6

メモ：

1. 読み出しおよび書き込みデータ幅が一致している場合
2. 表 11 に示したデータを使用
3. 最小イーサネット フレーム サイズ 64 バイト (CRC バイトを含む) を使用して測定
4. 最大イーサネット フレーム サイズ 1518 バイト (CRC バイトを含む) を使用して測定
5. 一般的なジャンボ フレーム サイズ 9000 バイトを使用して測定

リファレンス デザイン

LocalLink FIFO リファレンス デザインはザイリンクス ウェブ サイトから VHDL ファイルで入手できます (http://www.xilinx.co.jp/bvdocs/appnotes/xapp691.zip)。このリファレンス デザインには、シミュレーション テストベンチ、スクリプト、およびネットリストを作成してシミュレーションを実行するための PERL スクリプトが含まれます。

表 19 にリファレンス デザインの VHDL ソース ファイル名および説明を示し、29 ページの表 20 にシミュレーション テストベンチ ファイル名および説明を示します。また、30 ページの表 21 にスクリプトおよびプロジェクト ファイルを示します。

表 19: VHDL ソース ファイル

ファイル名	説明
ll_fifo.vhd	最上位モジュール。パラメータ設定によって、ll_fifo_BRAM または ll_fifo_DRAM をインスタンシエートします。
ll_fifo_BRAM.vhd	LocalLink 信号を処理し、BRAM_fifo をインスタンシエートする 2 番目の上位モジュール。1 つの LL_FIFO モジュールが含まれます。
ll_fifo_DRAM.vhd	LocalLink 信号を処理し、DRAM_fifo をインスタンシエートする 2 番目の上位モジュール。1 つの LL_FIFO モジュールが含まれます。
BRAM_fifo.vhd	ブロック SelectRAM 用の下位ソース コード
DRAM_fifo.vhd	分散 RAM 用の下位ソース コード
BRAM_fifo_pkg.vhd	BRAM_fifo 用のパッケージ
DRAM_fifo_pkg.vhd	DRAM_fifo 用のパッケージ
BRAM_macro.vhd	ブロック SelectRAM マクロを作成するためのソース コード
DRAM_macro.vhd	分散 RAM マクロを作成するためのソース コード
Virtex2p.vhd	Virtex-II Pro FPGA 用のパッケージ
fifo_utils.vhd	BRAM_fifo および DRAM_fifo モジュール用の制約を算出するため機能を含む FIFO ユーティリティ ファイル
RAM_64nX1.vhd	データ幅が 64 ビット以上の DRAM をインスタンシエートする場合に使用するファイル
BRAM_S8_S72.vhd	2 つの BRAM を使用し、8 から 72、または 72 から 8 へのデータ変換を行うブロックをインスタンシエートするための BRAM コンポーネント
BRAM_S8_S144.vhd	4 つの BRAM を使用し、8 から 144、または 144 から 8 へのデータ変換を行うブロックをインスタンシエートするための BRAM コンポーネント
BRAM_S18_S72.vhd	2 つの BRAM を使用し、18 から 72、または 72 から 18 へのデータ変換を行うブロックをインスタンシエートするための BRAM コンポーネント
BRAM_S16_S144.vhd	2 つの BRAM_S8_S72 を使用し、16 から 144、または 144 から 16 へのデータ変換を行うブロックをインスタンシエートするための BRAM コンポーネント
BRAM_S36_S144.vhd	2 つの BRAM_S18_S72 を使用し、36 から 144、または 144 から 36 へのデータ変換を行うブロックをインスタンシエートするための BRAM コンポーネント

表 19: VHDL ソース ファイル *(Continued)*

ファイル名	説明
BRAM_S36_S72.vhd	2 つの BRAM を使用し、36 から 72、または 72 から 36 へのデータ変換を行うブロックをインスタンシエートするための BRAM コンポーネント
BRAM_S72_S72.vhd	2 つの BRAM を使用し、72 から 72 へのデータ変換を行うブロックをインスタンシエートするための BRAM コンポーネント
BRAM_S72_S144.vhd	2 つの BRAM_S36_S72 を使用し、72 から 144、または 144 から 72 へのデータ変換を行うブロックをインスタンシエートするための BRAM コンポーネント
BRAM_S144_S144.vhd	2 つの BRAM_S72_S72 を使用し、144 から 144 へのデータ変換を行うブロックをインスタンシエートするための BRAM コンポーネント

表 20: シミュレーション テストベンチ ファイル

ファイル名	説明
ll_fifo.pl	LocalLink FIFO シミュレーションを実行するための PERL スクリプト
ll_fifo_tb_wave.do	LocalLink FIFO シミュレーションを実行するための波形フォーマット ファイル
FILEREAD_TESTER.v	ベクタ ファイルを読み出すモジュールのテスト用 Verilog コード
OUTPUT_TESTER.v	8 ビット データ幅以外のテスト ベクタを出力するモジュールのテスト用 Verilog コード
OUTPUT_TESTER_8_BIT.v	8 ビット データ幅のテスト ベクタを出力するモジュールのテスト用 Verilog コード
TESTER_pkg.vhd	TESTER モジュール用の VHDL パッケージ
UFC_CONVERTER.v	8 ビット データ幅以外のすべてのフロー制御を行う TESTER 用の Verilog コード
UFC_CONVERTER_8_bit.v	8 ビット データ幅のフロー制御を行う TESTER 用の Verilog コード
user_data_packets8.vec	8 ビット データ幅の場合のテスト ベクタ ファイル
user_data_packets16.vec	16 ビット データ幅の場合のテスト ベクタ ファイル
user_data_packets32.vec	32 ビット データ幅の場合のテスト ベクタ ファイル
user_data_packets64.vec	64 ビット データ幅の場合のテスト ベクタ ファイル
user_data_packets128.vec	128 ビット データ幅の場合のテスト ベクタ ファイル
ll_fifo_tb.vhd	LocalLink FIFO テストベンチのソース コード

表 21: スクリプトとプロジェクト ファイルのファイル名および説明

ファイル名	説明
ll_fifo.xst, ll_fifo.prj	XST 合成ツール用のプロジェクト ファイルおよびコマンド オプション
ll_fifo_run.pl	ネットリスト作成用の PERL スクリプト
config.csh	デザインの環境変数を設定するための Shell スクリプト ファイル
modelsim_unix.ini	UNIX を使用する場合の ModelSim 初期化ファイル
readme.txt	デザインのインプリメンテーションおよびシミュレーション方法を示した Readme ファイル

デザイン ツール。

- Xilinx ISE 6.1.03i
- ModelSim SE 5.6e

まとめ

パラメータの設定が可能な LocalLink FIFO は、LocalLink インターフェイスをサポートするユーザーアプリケーションに接続できます。この FIFO は、読み出しポートと書き込みポート間のデータ幅を変換し、これらのポート間の非同期動作をサポートします。ワード数にパラメータ設定可能な LocalLink FIFO は Virtex-II および Virtex-II Pro のブロック SelectRAM または分散 RAM を使用して設計でき、動作速度は、約 200 MHz です。

リファレンス

1. ザイリンクス社、XAPP258: Virtex-II ブロック RAM を使用した FIFO
2. ザイリンクス社、XAPP261: Virtex-II ブロック RAM メモリを使用したデータ幅変換 FIFO
3. ザイリンクス社、DS230: LocalLink インターフェイス
4. ザイリンクス社、LogiCORE データ シート、DS232: 非同期 FIFO v5.1

改訂履歴

次の表に、このアプリケーション ノートの改訂履歴を示します。

日付	バージョン	履歴
02/02/04	1.0	初期リリース