Virtex-II ブロック RAM を使用した FIFO

XAPP258 (v1.4) 2005 年 1 月 7 日

概要

Virtex™-II FPGA シリーズには、FIFO アプリケーションに使用するため、18Kbit True Dual-Port™ 同期 RAM という専用のオンチップ ブロックが含まれています。このアプリケーション ノートでは、共通クロック （同期クロック）および独立クロック （非同期クロック）の 511x36 FIFO の生成方法について説明します。この FIFO の幅とワード数は、Verilog または VHDL コードで変更できます。

はじめに

Virtex-II デバイスには、18K の完全同期デュアル ポート RAM が含まれています。これらのブロックは FIFO アプリケーションに適しており、各 RAM のポートに対して 16Kx1、8Kx2、4Kx4、2Kx9、1Kx18、または 512x36 など異なる設定をすることができます。

このアプリケーション ノートは、511x36 FIFO について説明しています。制御ロジックを修正することによって、各ポートの構造を変更することも可能です。FIFO のサイズは、512x36 ではなく 511x36 です。これは Empty/Full を正確に示すために FIFO からアドレスが 1 つ削除されるためです。このアプリケーション ノートでは共通の読み出し/書き込みクロックを使用する 511x36 FIFO について説明し、次に読み出しクロックと書き込みクロックが独立している複雑なデザインで必要となる変更について説明します。括弧内の信号名は、Verilog および VHDL コードで使用される記号です。

共通クロックを使用した同期 FIFO

図 1 に、同期 FIFO のブロック図を示します。読み出し/書き込みクロックの両方が同じソースから出力される場合、FIFO の動作および調停は単純であり、Empty/Full フラグは容易に生成できます。読み出し (read_addr) /書き込み (write_addr) アドレス カウンタの両方にバイナリ カウンタを使用します。図 2 に、511x36 同期 FIFO のタイミング図を示します。表 1 には、同期 FIFO のポート定義を示します。

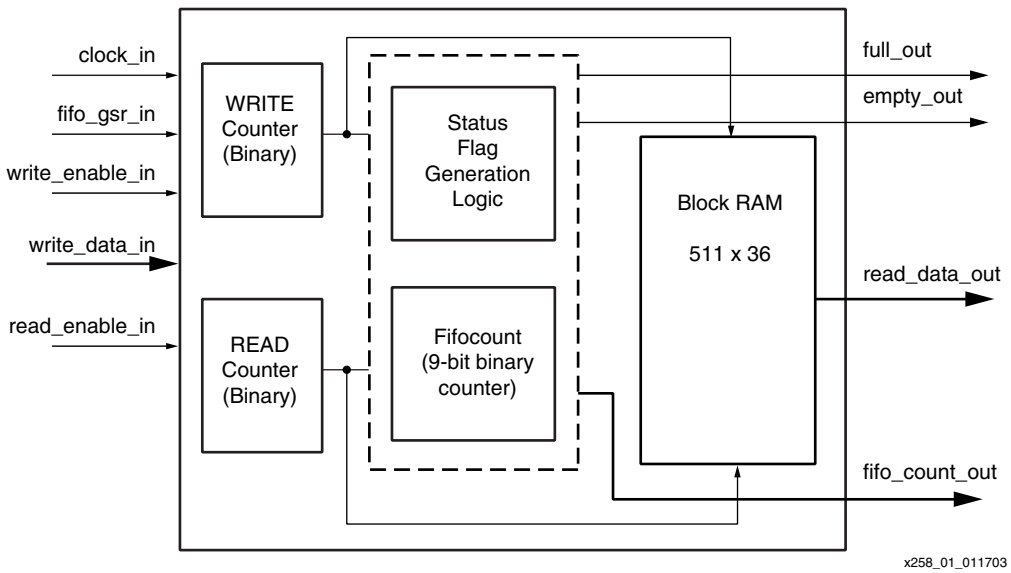


図 1： 511x36 同期 FIFO

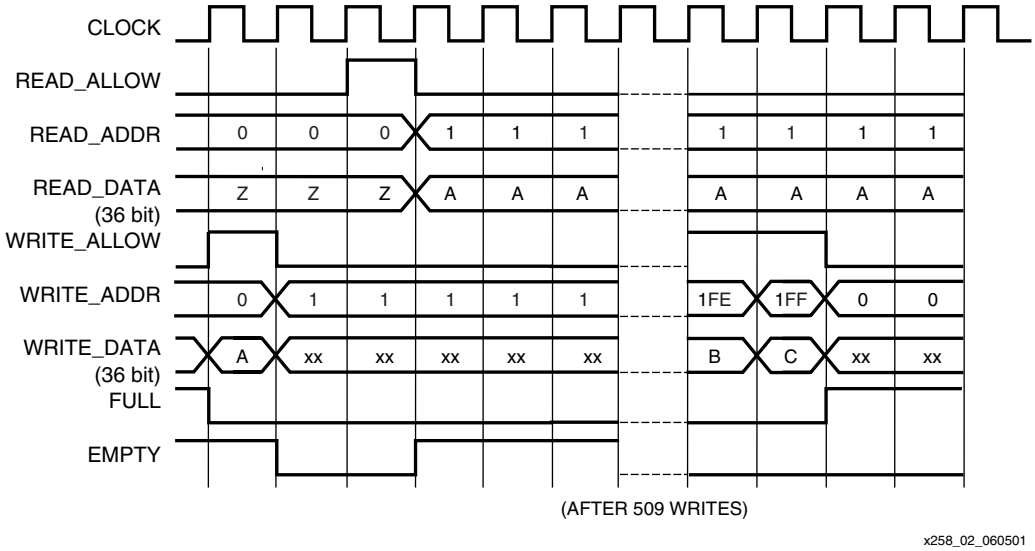


図 2： 511x36 同期 FIFO

表 1：ポートの定義

信号名	ポートの方向	ポート幅
clock_in	input	1
fifo_gsr_in	input	1
write_enable_in	input	1
write_data_in	input	36
read_enable_in	input	1
read_data_out	output	36
full_out	output	1
empty_out	output	1
fifocount_out	output	4

同期 FIFO の動作

読み出し動作の実行は、クロック エッジが立ち上がる前に読み出しイネーブル信号 (read_enable) が High になると、次のクロック サイクルで読み出しデータ (read_data) が出力されます。バースト読み出しは、読み出しイネーブルを数クロック サイクル間 High を継続するだけです。読み出しが開始され Empty がアクティブになると、最後のワードが読み出されて次の読み出しデータ信号は無効になります。

書き込み動作の実行は、まず入力に書き込みデータ (write_data) が必要です。クロック エッジが立ち上がる前に書き込みイネーブル信号 (write_enable) が High である必要があります。Full フラグが設定されない限り、書き込み動作は実行されます。バースト書き込みを実行するには、書き込みイネーブルを High に保持し、毎サイクル新しい書き込みデータ信号を入力する必要があります。

FIFO カウント (fifocount) という優れた機能が追加され、表 3 で示すように FIFO の占有状況を 1/2 または 3/4 などで判断します。これは、現在 FIFO に格納されているワード数のバイナリ カウントです。書き込みが実行されるとインクリメントし、読み出しではデクリメントします。両方の動作が同じクロック サイクル内で実行された場合、FIFO のステータスは変化しません。このアプリケーション ノートでは、上位 4 ビットのみ I/O に伝送されますが、容易に変更できます。

Empty フラグは、FIFO カウントが 0 になった場合、または FIFO カウントが 1 のときに読み出しが実行されている場合に設定されます。この早期デコード機能により、最後の読み出しの直後に Empty が設定されるようになっています。書き込み動作が開始されると Empty フラグは解除されます（ただし、同時に読み出しが実行されていないことが条件）。同様に Full フラグは、FIFO カウントが 511 になった場合、または FIFO カウントが 510 で書き込みのみが実行されている場合に設定されます。読み出し動作が開始されると Full フラグは解除されます（ただし、同時に書き込みが実行されていないことが条件）。読み出し/書き込みの両方が同じクロック サイクルで同時に実行される場合、ステータス フラグは変化しません。グローバル リセット (fifo_gsr) が適用された場合は、外部ロジックが FIFO と競合するのを回避するため、これらの信号はともに High になります。

独立クロックを使用した非同期 FIFO

図 3 に、511x36 非同期 FIFO のブロック図を示します。非同期 FIFO の読み出しポート信号および書き込みポート信号は、異なる読み出し/書き込みクロックを使用します。図 4 に、511x36 非同期 FIFO のタイミング図を示します。表 2 には、非同期 FIFO のポート定義を示します。

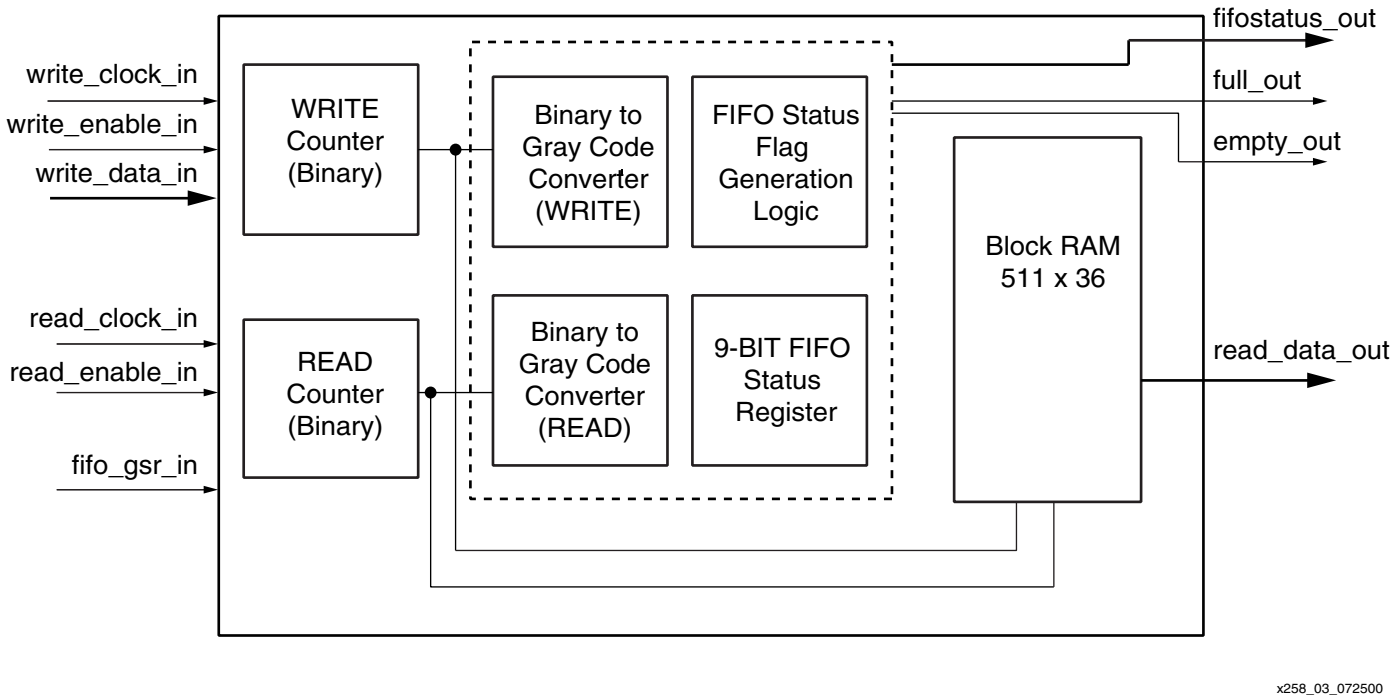
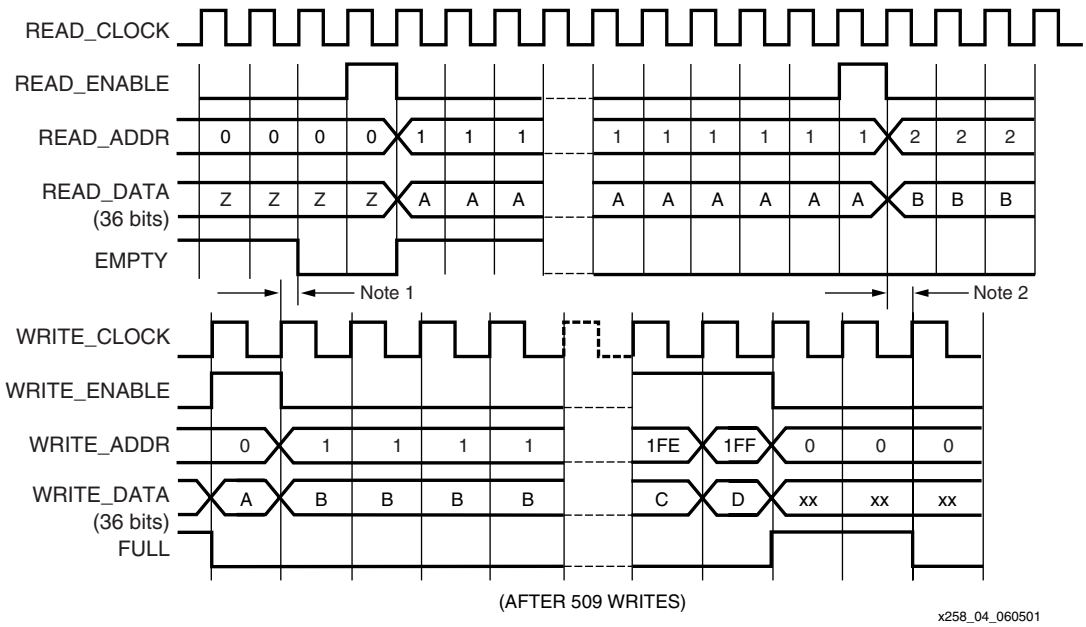


図 3： 511x36 非同期 FIFO



メモ：

1. READ_CLOCK の立ち上がりエッジより前に、書き込みアドレスがセットアップ タイム要件を満たした場合、Empty が Low になります。この要件を満たさない場合、Empty は 1 サイクル後に Low になります。
2. WRITE_CLOCK の立ち上がりエッジより前に、読み出しアドレスがセットアップ タイム要件を満たした場合、Full が Low になります。この要件を満たさない場合、Full は 1 サイクル後に Low になります。

図 4： 511x36 非同期 FIFO

表 2 : ポートの定義

信号名	ポートの方向	ポート幅
write_clock_in	input	1
read_clock_in	input	1
fifo_gsr_in	input	1
write_enable_in	input	1
write_data_in	input	36
read_enable_in	input	1
read_data_out	output	36
full_out	output	1
empty_out	output	1
fifostatus_out	output	4

非同期 FIFO の動作

読み出しと書き込みクロックが異なる場合の FIFO を動作させるには、ステータス フラグを判断するために非同期アービトレーション ロジックを使用します。非同期クロックの FIFO の場合、Empty は読み出しクロックを使用し、Full は書き込みクロックを使用するため、Empty/Full 生成ロジックとフリップフロップでは正確な制御ができません。

この問題を解決して制御ロジックのスピードを最大にするには、複雑な追加ロジックを取り入れる方法があります。ブロック RAM のアドレス入力を駆動する、プライマリ 9 ビット読み出しおよび書き込みバイナリ アドレス カウンタがあります。このバイナリ アドレスがグレイ コードに変換されて、複数のアドレス ポインタ (read_addrgray、read_nextgray、read_lastgray、write_addrgray、write_nextgray) を作成するためにいくつかの段階にパイプライン化されます。これにより、Full および Empty フラグをすばやく生成できます。グレイ コードを使用することにより、レジスタ付き Full/Empty フラグが常に正確に動作します。つまり読み出し/書き込みクロックが非同期であるために懸念される曖昧なフラグ ステートを示すことはありません。ワースト ケースでは、Full/Empty フラグが 1 サイクル余計にアクティブになる場合がありますが、この場合でもエラーは発生しません。

読み出しおよび書き込みの両方のグレイ コード ポインタが同一である場合、FIFO は Empty です。書き込みグレイ コード ポインタと次の読み出しグレイ コード ポインタが同一である場合、FIFO には511 (36 ビット) ワード全部が占有されている Full の状態です。FIFO が Almost Empty（ほぼ Empty) または Almost Full（ほぼ Full) であることを判断するためには、同じキャリー チェーン内で別の比較を行います。これにより、同じクロック エッジで最後の動作として Empty/Full を生成できます。(従来型の制御ロジックは、非同期信号を使用してフラグを設定しますが、スピードが非常に遅く、全体的なパフォーマンスは制限されます。)

FIFO ステータス信号は、上記の他にも 1/2 full および 1/4 full などさまざまな記号があります（表 3 を参照)。非同期 FIFO のステータスを生成するタスクは複雑であるため、より多くのロジックが必要になります。この信号がトリムされた場合、全体的なパフォーマンスは向上します。この FIFO ステータスの出力は、書き込み動作に対して 1 サイクル レイテンシであり、読み出し動作に対して 2 サイクル レイテンシです。

表 3 : FIFO カウントおよび FIFO ステータス信号の説明

ビット 3	ビット 2	ビット 1	ビット 0	FIFO ステータス/FIFO カウント
1	1	1	1	15/16 full
1	1	1	0	7/8 full
1	1	0	1	13/16 full
1	1	0	0	3/4 full
1	0	1	1	11/16 full
1	0	1	0	5/8 full
1	0	0	1	9/16 full
1	0	0	0	1/2 full
0	1	1	1	7/16 full
0	1	1	0	3/8 full
0	1	0	1	5/16 full
0	1	0	0	1/4 full
0	0	1	1	3/16 full
0	0	1	0	1/8 full
0	0	0	1	1/16 full
0	0	0	0	< 1/16 full

リファレンス デザイン

この独立クロック デザインは、ベンチ テスト済みです。各 FIFO デザインにシミュレーション テスト ベンチが提供されており、Model Tech Simulator を使用してシミュレーションを実行しました。ザイリ

ンクス FTP サイトより、VHDL および Verilog の両方のリファレンス デザインを入手できます。表 4 に、リファレンス デザインのファイル名と説明を示します。(ファイル : **xapp258.zip**)

表 4 : リファレンス デザインのファイル名および説明

ファイル名	説明
fifoctlr_cc_v2.v、vhd	511x36 FIFO 共通クロック
fifoctlr_ic_v2.v、vhd	511x36 FIFO 独立クロック
tb_x258v_cc.v	fifoctlr_cc_v2.v のテストベンチ
tb_x258v_ic1.v	fifoctlr_ic_v2.v のテストベンチ。書き込みより読み出しが高速
tb_x258v_ic2.v	fifoctlr_ic_v2.v のテストベンチ。読み出しより書き込みが高速

まとめ

Virtex-II ブロック RAM を使用して、同期 FIFO および非同期 FIFO の両方を生成できます。ブロック RAM の特徴である実質的なデュアル ポートを活用することにより、非同期 FIFO が可能になります。これらの FIFO の動作周波数は、約 200 MHz 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
2001/01/09	1.0	Virtex-II シリーズにおけるザイリンクス初版リリース。以前のデザインは、XAPP131 に記載。
2001/02/13	1.1	図 2 を更新
2001/06/05	1.2	図 2 および図 4 を更新
2003/01/17	1.3	テンプレートの更新および表 3 内の誤字修正
2005/01/07	1.4	リファレンス デザインのリンクを更新