XILINX®

XAPP465 (v1.0) 04.10.03

Spartan-3 デバイスでのシフト レジスタ (SRL 16) としてのルックアップ テーブルの使用

概要

SRL16 は、16 ビット シフト レジスタとして使用されるルックアップ テーブルの代替モードです。このシフト レジスタ LUT (SRL) モードの使用により、パフォーマンスの向上と大規模なコスト削減ができます。SRL16 はソフトウェア ツールで自動的に推論できますが、効率的な使用でさらにコストを抑えたデザインが実現できます。

はじめに

Spartan-3 FPGA は、SLICEM スライスのルックアップ テーブル (LUT) をフリップフロップ リソースを使用しない 16 ビット シフト レジスタとして設定できます。シフト イン オペレーションはクロックと同期しており、出力幅はダイナミックに選択できます。独立した専用出力により、16 ビット シフト レジスタを何個でもカスケードできるため、必要なサイズのシフト レジスタを生成できます。各 CLB リソースでは、8 個の LUT のうち 4 個を 65 ビット シフト レジスタとして設定できます。

このアプリケーション ノートは、16 から 64 ビットまでのシフト レジスタのインプリメンテーションに必要な一般的な VHDL と Verilog のサブモジュール、およびリファレンス コードの例を紹介します。これらのサブモジュールは、16 ビット シフト レジスタ プリミティブ、および専用 MUXF5、MUXF6、MUXF7 マルチプレクサから構成されています。

これらのシフト レジスタにより、遅延やレイテンシの補正を必要とするアプリケーションの効果的なデザインを展開します。また、同期 FIFO および内容参照可能メモリ (CAM) デザインにおいてもシフト レジスタは有効です。フリップフロップ (SRL16 エレメントなど) を使用しない Spartan-3 シフト レジスタを素早く生成するには、CORE Generator RAM-based Shift Register モジュールを使用します。

シフト レジスタ アーキテクチャ

SRL16 構造についての説明は、基本的なシフト レジスタから始め FPGA 構造の周辺構築へと展開していきます。

LUT 構造

ルックアップ テーブルは、バイナリ セレクト ラインとして機能する 4 入力の 16:1 マルチプレクサです。ルックアップ テーブルにプログラムされた値が選択されたデータになります (図 1 を参照)。

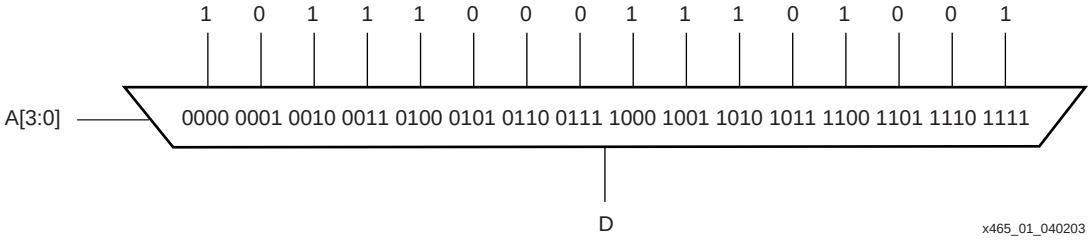

図 1: 16:1 マルチプレクサとして設計された LUT

SRL16 は、アドレス指定可能なシフト レジスタの代わりに固定した LUT 値が設定されます (図 2 を参照)。シフト レジスタの入力は LUT の同期 RAM 設定と同様に、データ入力、クロックおよびクロック イネーブル (図2 に表示されていません) があります。シフト レジスタの最後のフリップフロップには特

別な出力があります。ライブラリ プリミティブではこれを Q15、また FPGA Editor では MC15 と表示します。LUT 入力は、シフト レジスタの 16 記憶装置エレメントの中から非同期 (またはダイナミック) に 1 つ選択します。

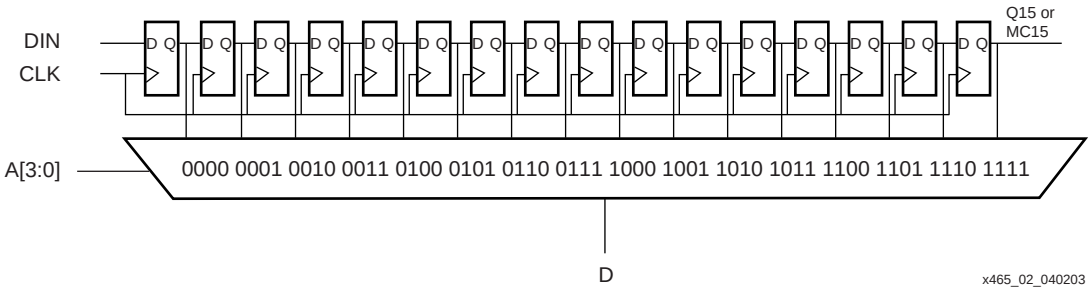

図 2: アドレス指定可能なシフト レジスタとして設定した LUT

ダイナミック レングス調整

アドレスでシフト レジスタの長さをダイナミックに変更できます。Q15 の代わりに D をシフト レジスタ出力として使用する場合は、アドレスに 7 (0111) を入力して出力を Q7 に設定し、8 ビット シフト レジスタをエミュレートします。アドレス ラインは mux を制御しているため、出力に非同期パスが与えられます。

ロジック セル構造

各 SRL16 LUT には関連したフリップフロップがあり、これらがすべてのロジック セルを構成しています。シフト レジスタのアドレス指定可能なビットは同期出力のため、フリップフロップに格納できます。また CLB の組み合わせ出力へ直接接続もできます。レジスタを使用する場合は、固定シフト レジスタ長を選択している固定アドレス ラインが必要です。フリップフロップのクロックから出力の遅延はシフト レジスタの遅延より短いため、パフォーマンス向上には最期から 2 番目のビットをアドレッシングし、シフト レジスタの最後にフリップフロップを使用します。フリップフロップの使用により、出力の非同期または同期セット/リセットも可能になります。

専用 SHIFTIN 信号がシフト レジスタ入力になり、最期のシフト レジスタの Q15/MC15 信号が SHIFTOUT 出力を駆動します。アドレス指定可能な D 出力は、すべての SRL プリミティブにありますが、SHIFTOUT を駆動する Q15/MC15 信号は、カスケード可能な SRLC16 プリミティブのみに存在します。

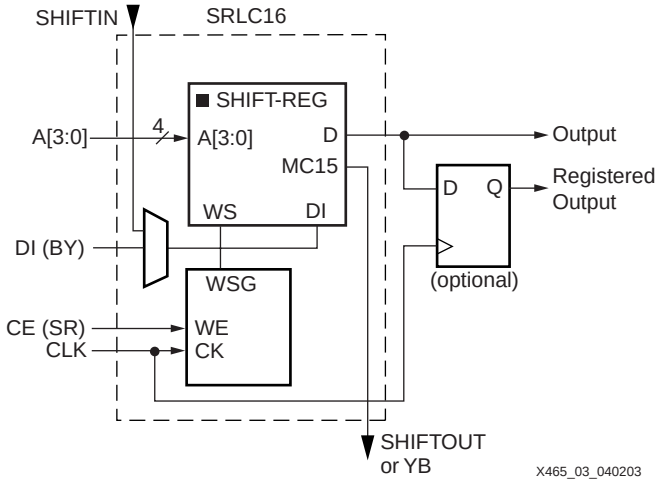

図 3: ロジック セル SRL 構造

スライス構造

スライス内の 2 つのロジック セルは、最大 32 ビットのシフト レジスタをカスケードするため、SHIFTOUT および SHIFTIN 信号で接続されます (図 4 を参照)。この信号は、前にあるシフト レジスタの Q15/MC15 と次のシフト レジスタの DI (または Q0 フリップフロップ) を接続します。

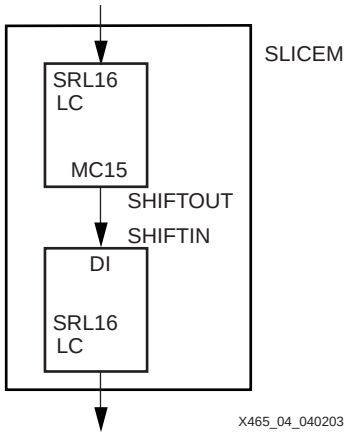

図 4: スライスのロジック セル間のシフト レジスタ接続

ダイナミック アドレッシング (または、ダイナミック レングス調整) を行う場合、各 SRL16 からの 2 つの独立したデータ出力は多重化します。F5MUX により、2 つの SRL16 ビットのうち片方が選択されます (図 5 を参照)。

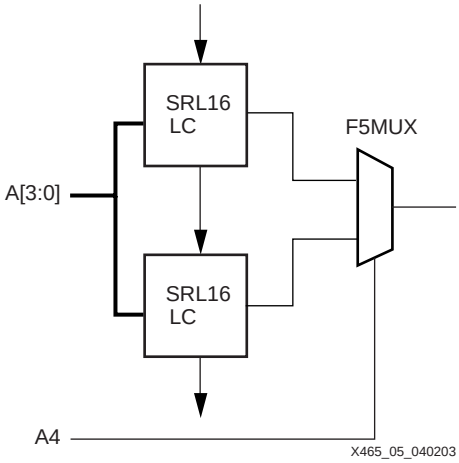

図 5: F5MUX を使用した複数 SRL16 コンポーネントのアドレッシング

CLB 構造

Spartan-3 の CLB には 4 つのスライスがあり、各スライスに 2 つのルックアップ テーブルがありますが、その中の 2 つのスライスのみ、ルックアップ テーブルを SRL16 コンポーネントおよび分散 RAM として使用できます。左側に位置する 2 つの SLICEM コンポーネントは、2 つの LUT を 16 ビット シフト レジスタとして設定できます。LUT 間のカスケードと同様に、SHIFTOUT から SHIFTIN への SLICEM コンポーネント間のカスケードもできます。ひとつの CLB の 左側に位置する 4 個の LUT をカスケードすると、最大 64 クロック周期の遅延を生成します (図 6 を参照)。

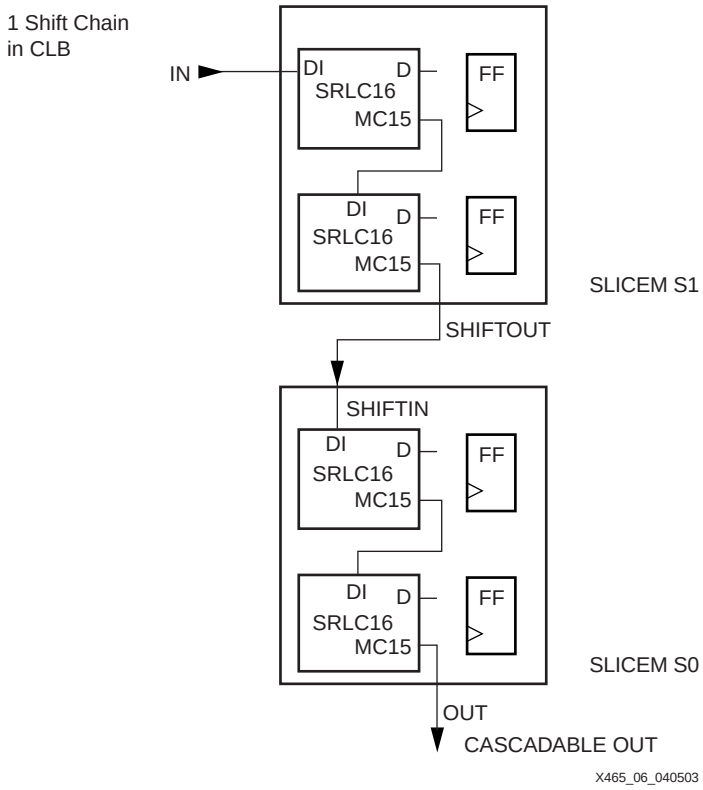

図 6: CLB のシフト レジスタ LUT のカスケード

SLICEM にある 2 つの LUT を接続するのと同様に、複数の SLICEM をアドレス指定するにはマルチプレクサを使用します。F6MUX を使用して CLB 内の SRL16 コンポーネントを 3 つまたは 4 つ選択し、最大 64 ビットまでのアドレス指定可能なシフト レジスタができます (図 7 を参照)。

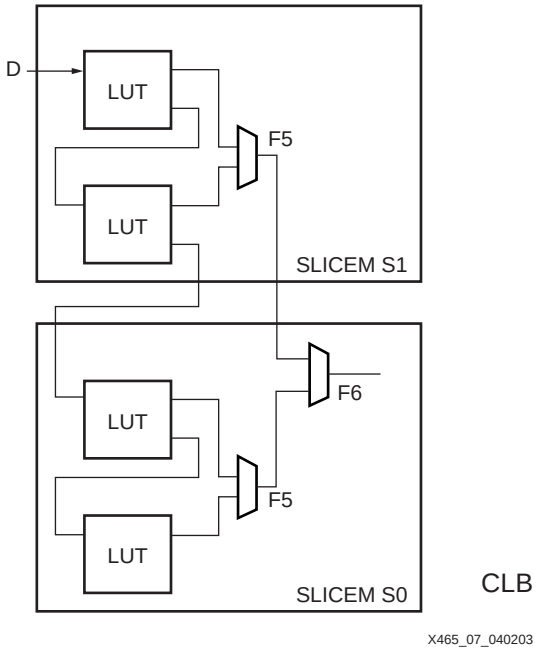

図 7: F6MUX を使用した 64 ビット シフト レジスタのアドレス指定

ライブラリ プリミティブ

ライブラリ プリミティブは 8 個あり、オプションでクロック イネーブル (CE)、反転クロック ($\overline{CLK}$)、およびカスケード可能な出力 (Q15) の組み合わせがあります。

合成およびシミュレーションで使用できるすべてのプリミティブを 表 1 に示します。

表 1: シフト レジスタ プリミティブ

プリミティブ	長さ	制御	アドレス入力	出力
SRL16	16 ビット	CLK	A3、A2、A1、A0	Q
SRL16E	16 ビット	CLK、CE	A3、A2、A1、A0	Q
SRL16_1	16 ビット	$\overline{CLK}$	A3、A2、A1、A0	Q
SRL16E_1	16 ビット	$\overline{CLK}$、CE	A3、A2、A1、A0	Q
SRLC16	16 ビット	CLK	A3、A2、A1、A0	Q、Q15
SRLC16E	16 ビット	CLK、CE	A3、A2、A1、A0	Q、Q15
SRLC16_1	16 ビット	$\overline{CLK}$	A3、A2、A1、A0	Q、Q15
SRLC16E_1	16 ビット	$\overline{CLK}$、CE	A3、A2、A1、A0	Q、Q15

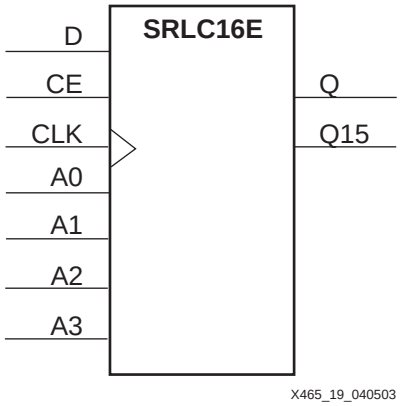

図 8: SRLC16E プリミティブ

表 2: VHDL および Verilog コードによる初期化

シフト レジスタは、合成およびシミュレーションにおいて VHDL または Verilog コードで初期化できます。合成を行う場合、16 ビット シフト レジスタ インスタンシエーションに INT 属性があるので、Alliance シリーズ ツールでコンパイルできるように、INT 属性を EDIF 出力ファイルにコピーします。VHDL コード シミュレーションは、`generic` パラメータを使用して属性を渡します。Verilog コード シミュレーションは、`defparam` パラメータを使用して属性を渡します。

S3_SRL16E シフト レジスタ インスタンシエーション コードの例 (VHDL および Verilog) を用いて、これらのテクニックを説明します (VHDL および Verilog テンプレートを参照)。`S3_SRL16E.vhd` および `.v` ファイルに関する説明はありません。

ポート信号

クロック — CLK

同期シフト インでは、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どちらか片方を使用します。データおよびクロック イネーブル入力ピンのセットアップ タイムは、選択したクロック エッジが基準になります。

データ入力 — D

データ入力は、シフト レジスタへ移動する新しいデータ (1 ビット) を与えます。

クロック イネーブル — CE (オプション)

クロック イネーブル ピンは、シフト機能に影響を与えます。非アクティブなクロック イネーブル ピンは、シフト レジスタへデータを移動しません。また、新しいデータの書き込みも行いません。クロック イネーブルをアクティブにした場合、data in (D) が最初のロケーションに書き込まれ、すべてのデータが 1 つのロケーションに移動します。このオプションを使用した場合、新しいデータは出力ピン （Q) およびカスケード可能な出力ピン (Q15) から出力されます。

アドレス — A3、A2、A1、A0

アドレス入力は、読み出されるビット (0 から 15 まで) を選択します。選択された n 番目のビットは、出力ピン（Q）から出力されます。カスケード可能な出力ピン (Q15) は、常にシフト レジスタの最後のビット (ビット 15) を出力するためアドレス入力の影響は受けません。

データ出力 — Q

データ出力 Q は、アドレス入力で選択されたデータ値 （1 ビット）を出力します。

データ出力 — Q15 (オプション)

データ出力 Q15 は、16 ビット シフト レジスタの最後のビットを出力します。各シフト インの動作が終了すると毎回新しいデータが出力されます。

制御ピンの反転

2 つの制御ピン（CLK、CE）には、反転オプションがあります。デフォルトでは、クロックは立ち上がりエッジで、CE はアクティブ high です。

GSR

グローバル セット/リセット (GSR) は、シフト レジスタに影響を与えません。

属性

内容の初期化 — INIT

INIT 属性は、シフト レジスタ内容の初期値を定義したものです。INIT 属性は 4 桁 (0000) の 16 進数でエンコードされたビット ベクタです。一番左の数字が最上位ビット (MSB) です。デバイス設定シーケンスで、シフト レジスタはデフォルトですべての数字が 0 に初期化されますが、その他の設定値を指定することもできます。

ロケーション制約

CLB 内部のスライス レイアウトは 図 9 に示します。各 CLB に 4 つのスライスがありますが、左下 2 つのスライスのみシフト レジスタとして使用できます。CLB で S0 および S1 に位置するこの 2 つのスライスは、SLICEM と表示されます。各座標は、XOYO および X0Y1 です。配置制約を行う場合、SRL プリミティブの LOC プロパティでこの座標を使用します。専用 CLB シフト チェーンは上下に渡っていますが、始めと終わりは 4 つの SLICEM LUT のどこを選んでもチェーンを作成できます。

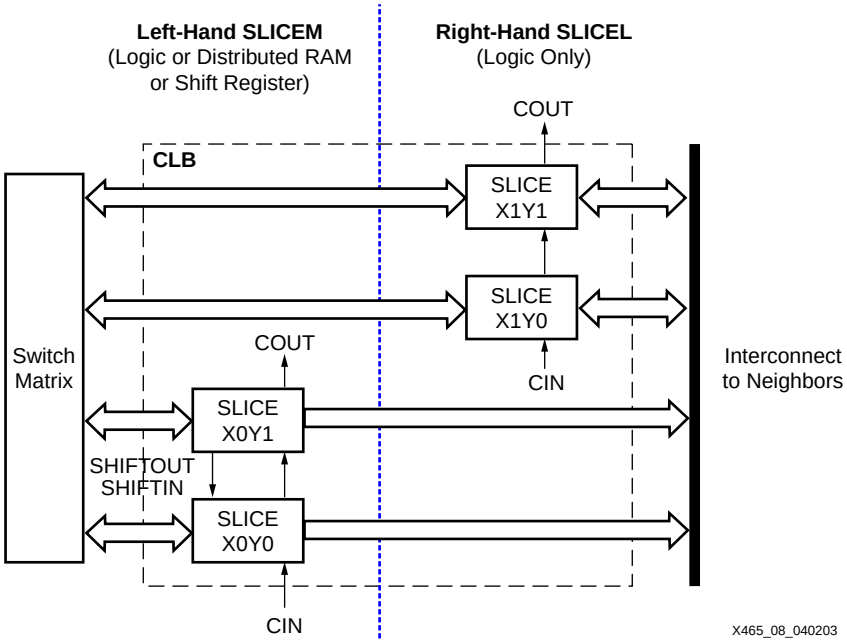

図 9: CLB 内部のスライス レイアウト

シフト レジスタの動作

データ フロー

各シフト レジスタ (SR16 プリミティブ) は、次をサポートします。

- 同期シフト イン
- アドレスがダイナミックに変更する場合の非同期 1 ビット出力
- アドレスが固定の場合の同期シフト アウト

その他、カスケード可能なシフト レジスタ (SRLC16) は、最期 (16 番目) のビットの同期シフト アウト出力をサポートします。この出力には、CLBリソース内で次の SRLC16 の入力への専用接続があります。2 つのプリミティブを 図 10 に示します。

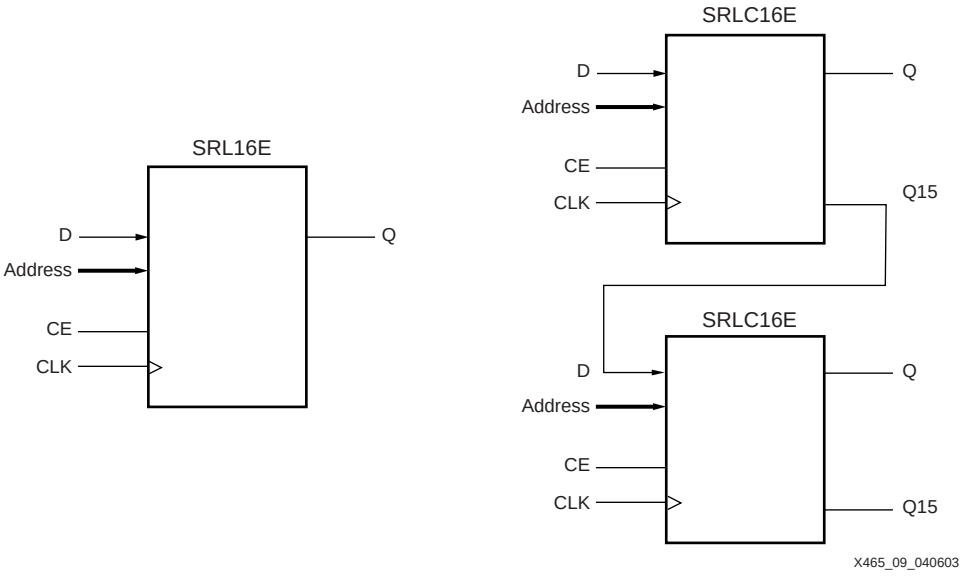

図 10: シフト レジスタとカスケード可能なシフト レジスタ

シフト オペレーション

シフト オペレーションは、CE がアクティブ High でシングル クロック エッジです。イネーブルが High のとき、入力 (D) はシフト レジスタの最初のビットにロードされ、各ビットが順に次のポジションへと移動します。カスケード可能なシフト レジスタ (SRLC16 など) では、最期のビットが Q15 から出力されます。

4 ビット アドレスで選択されたビットは、Q から出力されます。

ダイナミック リード オペレーション

Q 出力は、4 ビット アドレスで決定します。4 入力ピンに新しいアドレスが与えられるたびに、LUT へのアクセス時間を経て Q 出力に新しいビット ポジションの値が出力されます。このオペレーションは、クロックおよびクロック イネーブル信号とは非同期で単独の動作です。

シフトおよびダイナミック リード オペレーションを 図 11 に示します。

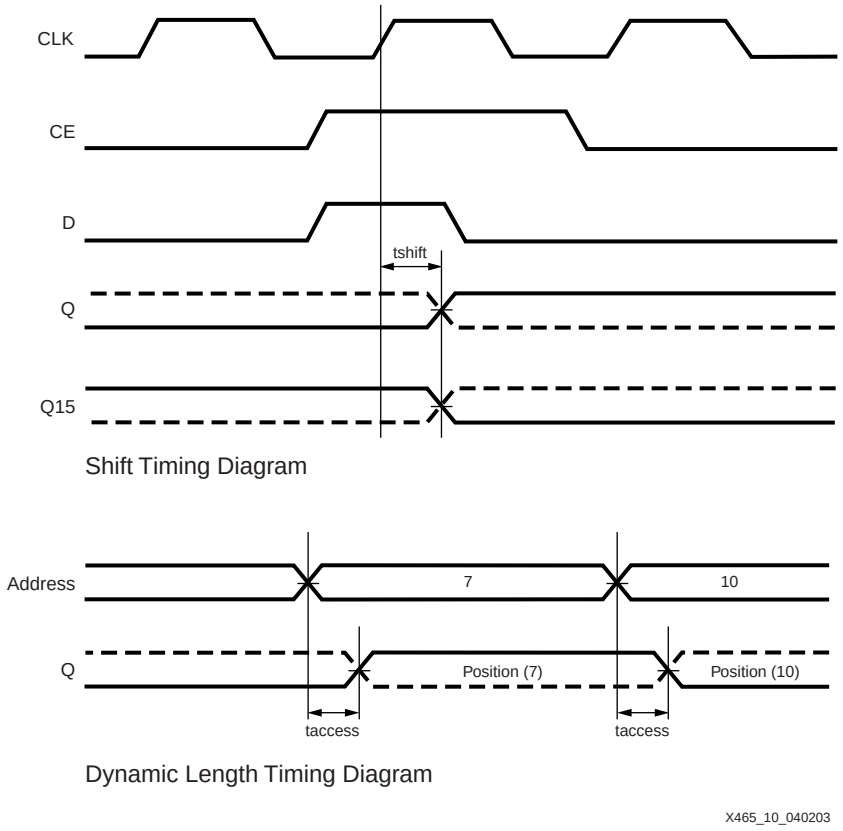

図 11: シフトおよびダイナミック レングス タイミング ダイアグラム

スタティック リード オペレーション

4 ビット アドレスが固定されている場合、Q 出力は常に同じビット ポジションを使用します。このモードは、1 つの LUT に 1 から 16 ビットまでどの長さのシフト レジスタでもインプリメントできます。シフト レジスタ長は (N+1) です。(N = 入力アドレス)

Q 出力は、各シフト オペレーションに同期して変化します。前のビットは次のポジションに移動し、Q から出力されます。

特性

- シフト オペレーション 1 回につき、クロック エッジが 1 つ必要です。
- ダイナミック レングス リード オペレーションは非同期です。(Q 出力)
- スタティック レングス リード オペレーションは同期です。(Q 出力)
- データ入力には、セットアップからクロック間のタイミング仕様があります。

- カスケード可能なコンフィギュレーションでは、Q15 出力に常に最期のビット値が出力されます。
- Q15 出力は、各シフト オペレーションの後に同期して変化します。

シフト レジスタ推論

一般的な HDL コードでシフト レジスタを記述した場合、合成ツールは SRL16 コンポーネントの使用を推論します。SRL16 には、同期または非同期セット / リセット入力のどちらもないため、同時にすべてのビットにアクセスできません。この機能は SRL16 を使用する上で妨げとなるため、フリップフロップの方にこの機能をインプリメントします。シフト レジスタが 16 ビット以上の場合、または Q15 のみ使用した場合は、カスケード可能なシフト レジスタ (SRLC16) が推論される可能性があります。SRL16 シフト レジスタはパラレル ロードできませんが、ロードしたいデータをシフト インすると同等の機能をインプリメントできます。ロード コマンドには一定のタイミングが必要です。

VHDL 推論コード

次は、SRL16 を推論する VHDL コードです。

```
architecture Behavioral of srl16 is

signal Q_INT: std_logic_vector(15 downto 0);

begin

process(C)
begin
  if (Cíevent and C=’1’) then
    Q_INT <= Q_INT(14 downto 0) & D;
  end if;
end process;

Q <= Q_INT(15);

end Behavioral;
```

反転クロック (SRL16_1) を推論させる場合は、`C='1'` を `C='0'` に変更します。クロック イネーブル (SRL16E) を推論させる場合は、最初の if-then 文の後に `if (CE='1') then` を挿入します。

Verilog 推論コード

次は、SRL16 を推論する Verilog コードです。

```
always @ (posedge C)
begin
  Q_INT <= {Q_INT[14:0],D};
end

always @(Q_INT)
begin
  Q <= Q_INT[15];
end
```

反転クロック (SRL16_1) を推論させる場合は、`(posedge C)` を `(negedge C)` に変更します。クロック イネーブル (SRL16E) を推論させる場合は、begin 文の後に `if(CE)` を挿入します。

シフト レジスタ サブモジュール

16 ビット プリミティブのほかにも、32 ビットと 64 ビットのカスケード可能なシフト レジスタをインプリメントする 2 つのサブモジュールが VHDL および Verilog コードであります。このサブモジュールを 表 3 に示します。

表 3: シフト レジスタ サブモジュール

サブモジュール	長さ	制御	アドレス入力	出力
SRLC32E_SUBM	32 ビット	CLK、CE	A4、A3、A2、A1、A0	Q、Q31
SRLC64E_SUBM	64 ビット	CLK、CE	A5、A4、A3、A2、A1、A0	Q、Q63

サブモジュールは、SRLC16E プリミティブが基本となり、専用マルチプレクサ (MUXF5、MUXF6 など) があります。このインプリメンテーションは、大規模なシフト レジスタの場合でも、高速なスタティック レングス モードおよびダイナミック レングス モードができます。

表 3 のサブモジュールでインプリメントしたカスケード可能なシフト レジスタ (32 ビットおよび 64 ビット) を 図 12 に示します。

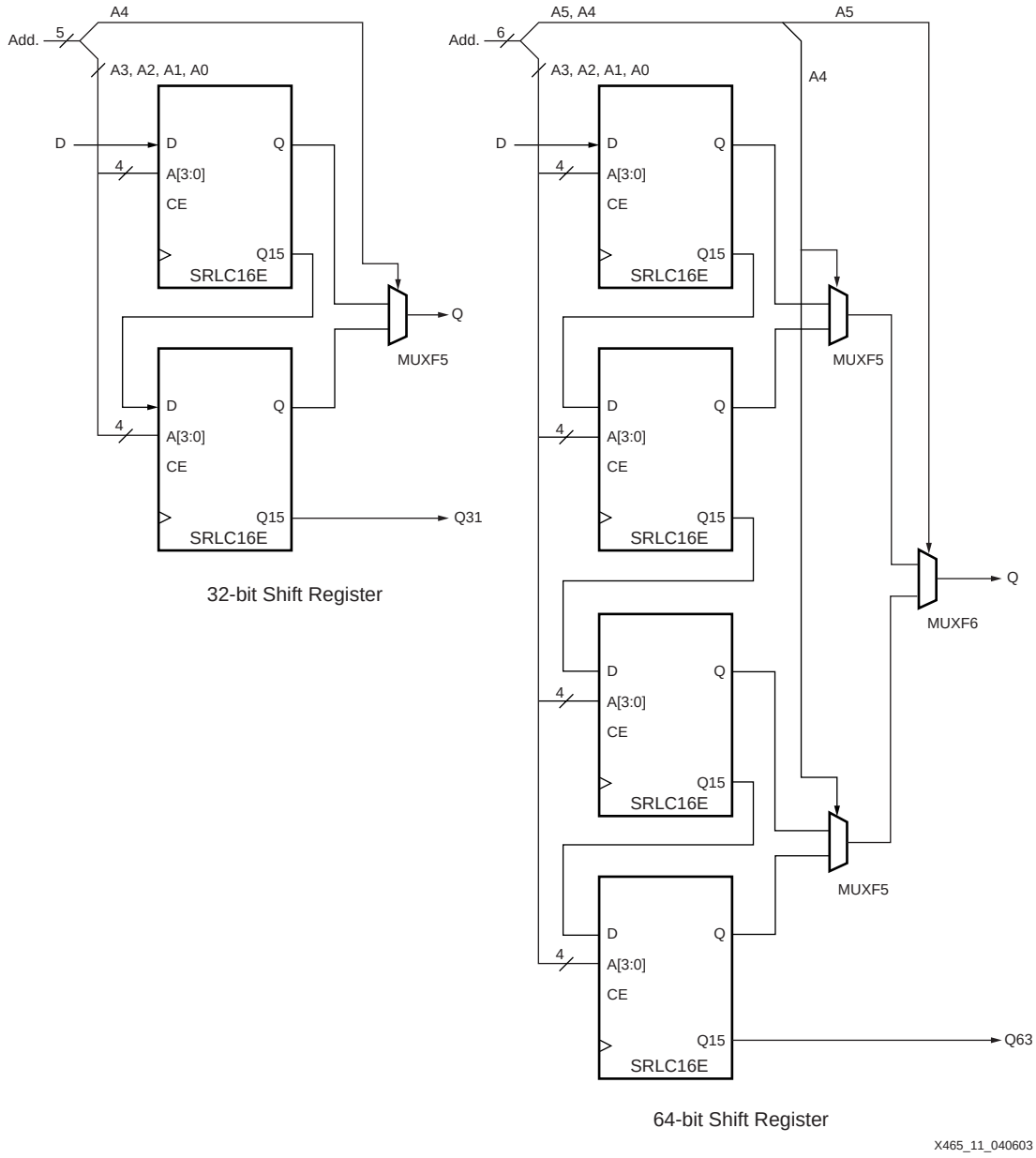

図 12: シフト レジスタ モジュール (32 ビットおよび 64 ビット)

1 つのサブモジュールにつき、すべてのクロック イネーブル (CE) とクロック (CLK) 入力は、1 つのグローバル クロック イネーブルと 1 つのクロック信号に接続しています。グローバルなスタティックまたはダイナミック レングス モードを必要としない場合は、マルチプレクサなしで SRLC16E プリミティブをカスケードできます。

完全同期シフト レジスタ

すべてのシフト レジスタ プリミティブおよびサブモジュールは、同じスライス内のレジスタを使用しません。完全同期リード/ライト シフト レジスタをインプリメントするときは、出力ピン Q をフリップ

フロップに接続する必要があります。図 13 で示すように、シフト レジスタとフリップフロップは同じクロックを共有しています。

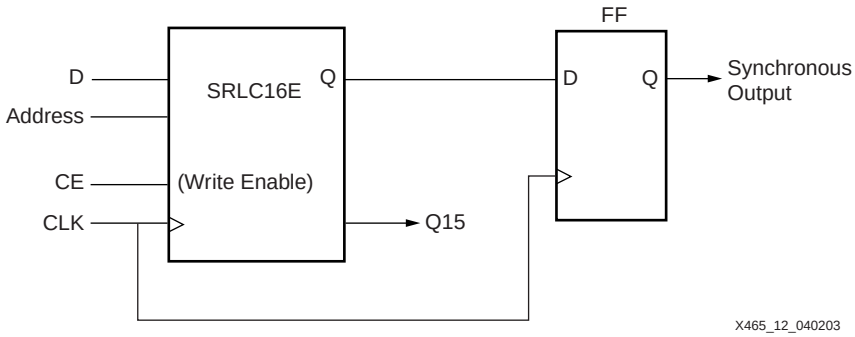

図 13: 完全同期シフト レジスタ

この設定では、タイミングが向上しデザインが簡略化します。フリップフロップは、シフト レジスタ チェーンの最後のレジスタになるので、スタティックまたはダイナミック アドレスは、要求する長さからフリップフロップの分を 1 ビット引いて考えます。必要な場合は、カスケードされた出力をフリップフロップにレジスタすることもできます。

スタティック シフト レジスタ

カスケード可能な 16 ビット シフト レジスタは、専用マルチプレクサ (MUXF5、MUXF6 など) を使用しない、あらゆる長さのスタティック レングス モード レジスタをインプリメントできます。40 ビット シフト レジスタを 図 14 に示します。最期の SRLC16E プリミティブのみ、アドレス入力を 0111 に固定する方法、およびシフト レジスタの長さを 39 ビット (アドレスを 0110 に固定) に制限して、フリップフロップを最期のレジスタとして使用する方法があります。(SRLC16E プリミティブの場合、シフト レジスタの長さはアドレス入力 + 1 です。)

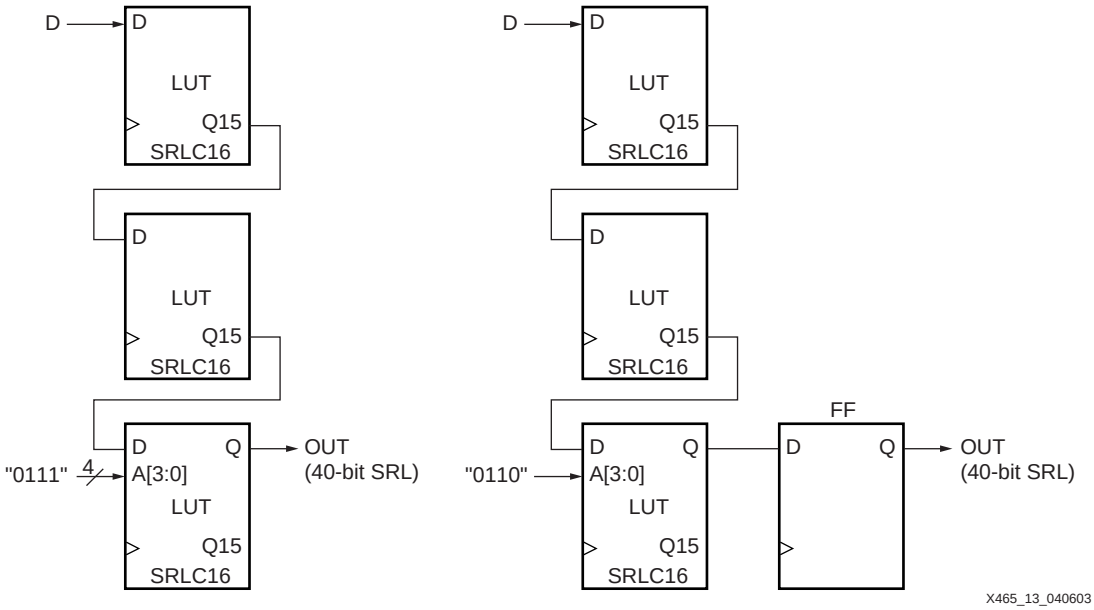

図 14: 40 ビットスタティック シフト レジスタ

VHDL および Verilog インスタンシエーション

すべてのプリミティブおよびモジュールの、VHDL および Verilog のインスタンシエーション テンプレートがあります。

VHDL の場合、各テンプレートに component declaration および architecture section があります。VHDL デザイン ファイルにテンプレートの各パートを挿入してください。architecture section の port map にはデザイン信号名が必要です。

ShiftRegister_C_x (x = 16、32、および 64) テンプレートはカスケード可能なモジュールで、対応する SRLCxE プリミティブ (16)、またはサブモジュール (32 および 64) をインスタンシエートします。

ShiftRegister_16 テンプレートを使用して、SRL16 プリミティブをインスタンシエートできます。

VHDL および Verilog テンプレート

テンプレート名の数字はビット数を表しています (例 : SHIFT_SELECT_16 は、16 ビット シフト レジスタ)。C はカスケード可能を意味します。

次は、プリミティブのテンプレートです。

- SHIFT_REGISTER_16
- SHIFT_REGISTER_16_C

次は、サブモジュールのテンプレートです。

- SHIFT_REGISTER_32_C (submodule: SRLC32E_SUBM)
- SHIFT_REGISTER_64_C (submodule: SRLC64E_SUBM)

対応するサブモジュールは、デザインと合成する必要があります。

例として、SHIFT_REGISTER_16_C モジュールのテンプレートを VHDL および Verilog コードで示します。

VHDL テンプレート

```
-- Module: SHIFT_REGISTER_C_16
-- Description: VHDL instantiation template
--  CASCADABLE 16-bit shift register with enable (SRLC16E)
-- Device: Spartan-3 Family
------------------------------------------------------------------------
-- Components Declarations:
--
component SRLC16E
-- pragma translate_off
  generic (
-- Shift Register initialization ("0" by default) for functional
simulation:
        INIT : bit_vector := X"0000"
  );
-- pragma translate_on
  port (
         D : in std_logic;
        CE   : in std_logic;
        CLK  : in std_logic;
        A0   : in std_logic;
        A1   : in std_logic;
        A2   : in std_logic;
        A3   : in std_logic;
        Q    : out std_logic;
        Q15  : out std_logic
  );
end component;
-- Architecture  Section:
--
-- Attributes for Shift Register initialization ("0" by default):
attribute INIT: string;
--
attribute INIT of U_SRLC16E: label is "0000";
--
-- ShiftRegister Instantiation
```

```
U_SRLC16E: SRLC16E
  port map (
  D      => , -- insert input signal
  CE     => , -- insert Clock Enable signal (optional)
  CLK    => , -- insert Clock signal
  A0     => , -- insert Address 0 signal
  A1     => , -- insert Address 1 signal
  A2     => , -- insert Address 2 signal
  A3     => , -- insert Address 3 signal
  Q      => , -- insert output signal
  Q15    =>   -- insert cascadable output signal
  );
```

Verilog テンプレート

```
// Module: SHIFT_REGISTER_16
// Description: Verilog instantiation template
// Cascadable 16-bit Shift Register with Clock Enable (SRLC16E)
// Device: Spartan-3 Family
//-----------------------------------------------------------------------
// Syntax for Synopsys FPGA Express
// synopsys translate_off

  defparam

//Shift Register initialization ("0" by default) for functional simulation:
  U_SRLC16E.INIT = 16'h0000;
// synopsys translate_on

//SelectShiftRegister-II Instantiation
   SRLC16E U_SRLC16E   ( .D(),
                         .A0(),
                         .A1(),
                         .A2(),
                         .A3(),
                         .CLK(),
                         .CE(),
                         .Q(),
                         .Q15()
          );

// synthesis attribute declarations
  /* synopsys attribute
  INIT "0000"
  */
```

CORE Generator システム

ザイリンクス Core Generator システムは、SRL16 を使用して高速で小型の FIFO スタイル シフト レジスタ、遅延ライン、およびタイム スキュー バッファを生成します。最大幅 256、ワード数 1024 の非常に効率的なシフトがある、RAM-based Shift Register モジュールを 図 15 に示します。スタティック レングス シフト レジスタおよびダイナミック レングス シフト レジスタを生成できます。また、モジュールの出力をレジスタするオプションもあります。出力をレジスタするオプションを使用する場合、次のような追加オプションがあります。Clock Enable、Asynchronous Set/Clear/Init、および Synchronous Set/Clear/Init。相対配置マクロ (RPM) または未配置ロジックとしてモジュールを作成するオプションもあります。

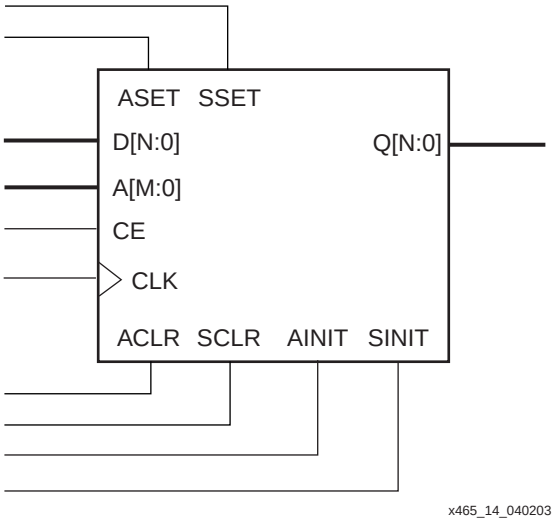

図 15: CORE Generator RAM-Based Shift Register モジュール

アプリケーション

遅延ライン

レジスタが豊富なザイリンクス FPGA アーキテクチャは、パイプライン ステージを追加できるため、高処理能力が得られます。要求する機能を維持するため、データ パスを均等にする必要があります。遅延の追加クロック サイクルが必要な場合は、SRL16 を使用します (図 16 を参照)。

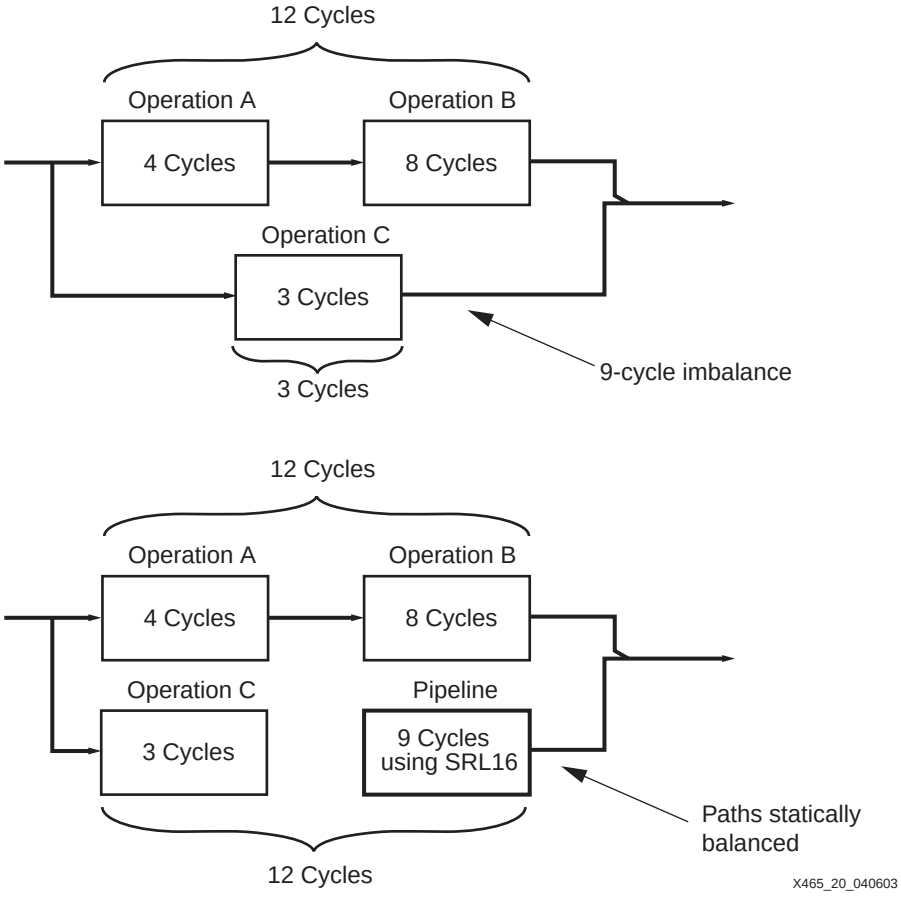

図 16: 遅延ラインとして SRL16 を使用

リニア フィードバック シフト レジスタ

リニア フィードバック シフト レジスタ (LFSR) のシーケンスは、2^n-1 （n = フリップフロップの数）の連続です。特定のビットを XOR または XNOR ゲートを通ってフィードバックすることでシーケンスができます。カウント シーケンスがあまり重要でないアプリケーション（例 : FIFO）に関しては、従来のバイナリ カウンタの代わりに LFSR を使用できます。また、擬似ランダム ナンバー ジェネレータとしても使用します。LFSR は、アルゴリズムの暗号化および解読に重要な構築ブロックです。

最長 LFSR は、シフト レジスタ内の特定の場所からビットをタップする必要があります。SRL16 にはさまざまなタップ方法があります。1 つ目は、Q15 と次の SRL16 をカスケードする一方で SRL16 で必要なビットをアドレッシングする方法です。2 つ目は、タップ ポイントにアクセスするためにフリップフロップを使用して SRL16 を拡張する方法です。たとえば、ビット 49 とビット 52 のフィードバックで 1 つの CLB に 52 ビット LFSR が インプリメンテーションされる様子を 図 17 に示します。3 つ目は、複数 SRL の LFSR を複製し、各 SRL に異なるビットをアドレス指定する方法です。4 つ目は、複数のビット ポジションを得るために 1 つの SRL クロック サイクルに複数のアドレスを生成する方法です。LFSR に必要な XNOR ゲートは、CLB の SLICEL にあらかじめ配置されています。詳細は、XAPP210 を参照してください。

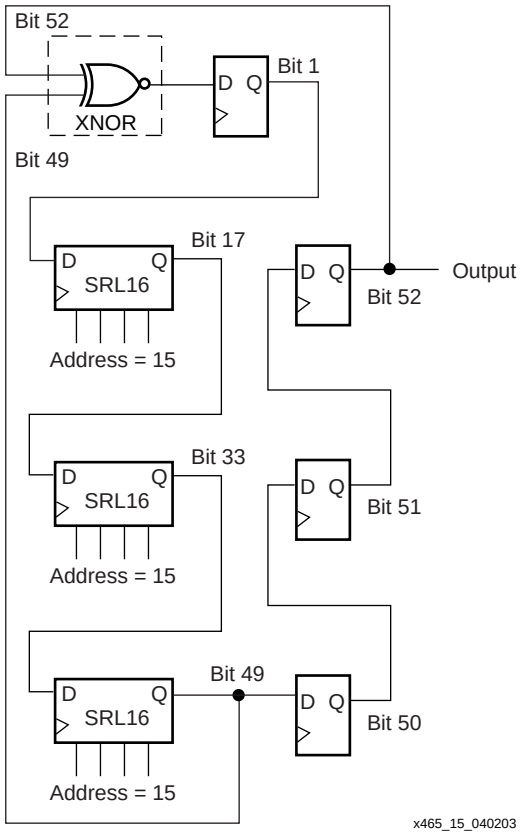

図 17: 1 つの CLB にある 52 ビット LFSR

Gold コード ジェネレータ

優れた相関プロパティを持つコード シーケンスを生成するには、CDMA システムで Gold コード ジェネレータを使用します (図 18 を参照)。要素コードを持つ 2 つの LFSR の結果を足したモジュロ 2 により小さな相関コードのセットを生成することを R.Gold 氏は提唱しています。たくさんのコード信号が混じったスペクトルの中で各コードのセットを識別できるようになります。Gold コード ジェネレータを図 18 に示します。LFSR を完成してフィードバックを行うのに必要なロジックは、CLB の SLICEL の中に配置されています。詳細は、XAPP217 を参照してください。

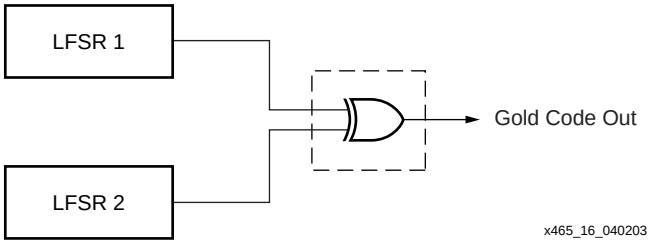

図 18: Gold コード ジェネレータ

FIFO

SRL16 コンポーネントから同期 FIFO を生成できます。他のリソースが不足している場合には、この方法が非常に有効です。1 つの CLB に対して最大 64 ビットまで可能です。より大規模な FIFO の場合は、最も効果的なリソースとしてブロック RAM を使用します。詳細は、XAPP256 を参照してください。

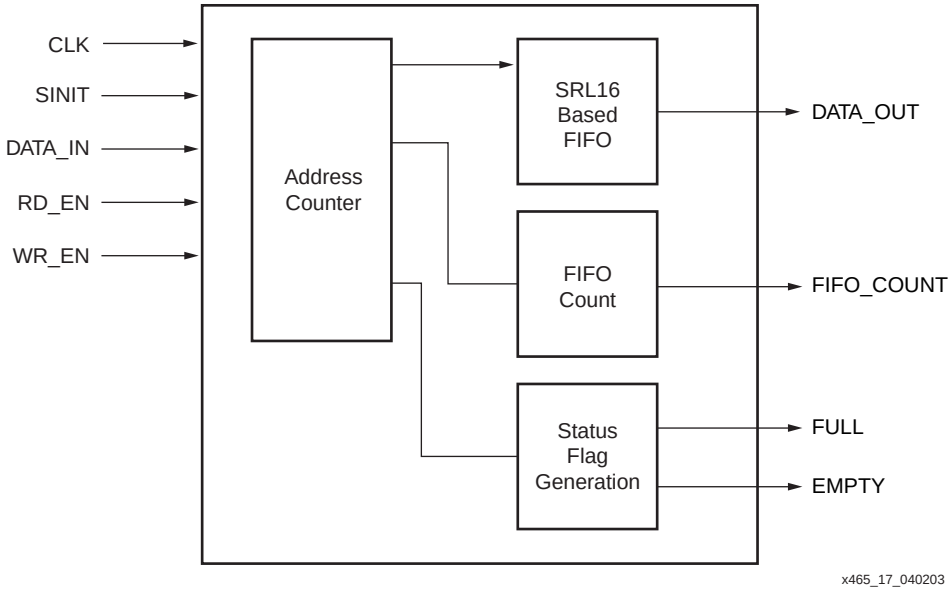

図 1: SRL16 シフト レジスタを使用した同期 FIFO

カウンタ

SRL16 の各出力をフィードバックすることにより、16 ステートのシーケンスができます。SRL16 をカスケードした場合、さらに長い不定カウント シーケンスができます。ターミナル カウントは、通常のキャリー チェーンを使用して生成できます（図 19 を参照）。

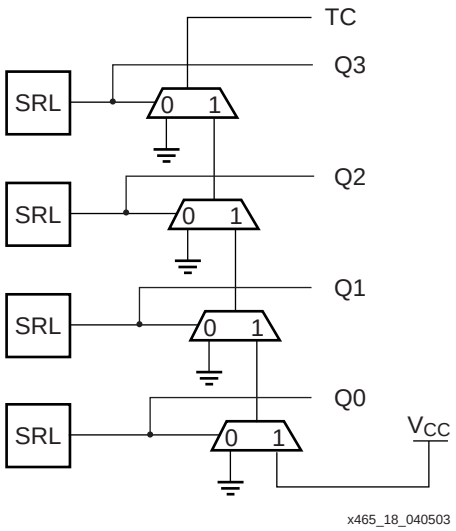

図 19: ターミナル カウント付きの SRL-Based カウンタ

関連資料およびリファレンス

- XAPP210 : Virtex デバイスのリニア フィードバック シフト レジスタ
  リニア フィードバック シフト レジスタは FPGA アーキテクチャで非常に有効なカウンタです。シフト レジスタの基本として SRL16 を使用した場合、1 つのスライスには 15 ビット カウンタ、2 つのスライスには 52 ビット カウンタが入ります。
- XAPP211 : SRL マクロを使用した PN ジェネレータ
  擬似ランダム雑音シーケンスを使用し、スペクトラム拡散の変調のために信号をコード化して幅広い伝送周波に信号を拡散します。PN ジェネレータは LFSR がベースであり、SRL16 コンポーネントで効率的に生成できます。
- XAPP217 : Virtex デバイスの Gold コード ジェネレータ
  PN シーケンスの特殊なタイプとして Gold コード ジェネレータがあります。これは、SRL16 ベースの LFSR で生成できます。
- XAPP220 : ワイアレス アプリケーションの機能ブロックとしての LFSR
  CDMA などのアプリケーションにおける LFSR (例 : Gold コード ジェネレータ) の使用に関する詳細について
- XAPP256 : Virtex-II シフト レジスタを使用した FIFO
  小規模な同期 FIFO を構築するには SRL16 が最適です。どんな幅の FIFO でも作成でき、1 ビットの解像度が得られます。カスケード可能な SRL16 シフト レジスタ (SRLC16) でフレキシブルな深度が 16 得られます。これらのテクニックは、ブロック RAM リソースが不足している場合の大規模 FIFO の生成に有益です。
- TechXclusive : The SRL16E : How Using this Exciting Mode Can Lead to Cost Saving of an Order of Magnitude
  SRL16 の機能および次の項目におけるアプリケーションについて説明します。パイプライン補正、擬似ランダム ノイズ ジェネレータ、シリアル フレーム同期装置、ランニング アベレージ、パルス生成とクロック分割、パターン生成、ステート マシン、動的アドレス指定可能シフト レジスタ、FIFO、および RS232 レシーバ。
- DS228 : RAM-Based Shift Register LogiCORE Module
  SRL16 を使用して、高速で小型な FIFO スタイルのシフト レジスタ、遅延ライン、およびタイムスキュー バッファを生成します。
- SRL16 Primitives in Libraries Guide
  SRL16 プリミティブとその変形の使用方法および機能について説明します。

おわりに

Sprtan 3 ルック アップ テーブルの SRL16 は、16 個のフリップフロップを使用しない空間効率のよいシフト レジスタです。小規模シフト レジスタが HDL コードで記述された場合、自動的にこの機能が推論されます。このアプリケーション ノートで説明する SRL16 の使用を積極的に取り入れることにより、その他のアプリケーションにおいても優れた効果が得られます。

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
04/10/03	1.0	初版リリース