



XAPP464 (v2.0) 2005 年 3 月 1 日

Spartan-3 シリーズ FPGA でルックアップ テーブルを分散 RAM として使用する方法

概要

Spartan™-3、Spartan™-3L、Spartan™-3E の CLB には、最大 64 ビットのシングルポート RAM または最大 32 ビットのデュアルポート RAM が含まれています。この RAM は、FPGA 全体に分散されるので、ブロック RAM と区別するため「分散 RAM」と呼ばれています。分散 RAM は高速で、関連ロジックの近くに配置でき、小さいデータバッファ、FIFO、またはレジスタファイルに最適です。このアプリケーションノートでは、分散 RAM の特徴と機能、CORE Generator™ システムや VHDL または Verilog インスタンスエーションを使用した各種オプションの指定方法について説明します。

はじめに

Spartan-3 FPGA には、エンベデッド 18Kb ブロック RAM に加え、CLB 内で分散 RAM を使用できるという特徴があります。CLB リソースの SLICEM のファンクションジェネレータ (LUT) に、オプションで 16 ワード X 1 ビットの同期 RAM をインプリメントできます。ただし、SLICEL スライスに LUT には、分散 RAM をインプリメントできません。

分散 RAM では、書き込みは同期、読み出しは非同期で行われます。ただし、アプリケーションに必要であれば、各 LUT に関連付けられたレジスタを使用して同期読み出しをインプリメントできます。16X1 ビット RAM は、カスケード接続してビット数およびワード数の多いメモリを作成でき、タイミング遅延も特殊ロジックリソースにより最小限に抑えることができます。

Spartan-3 の CLB は、最大 64 ワード X 1 ビットまでのさまざまな RAM プリミティブをサポートしています。SLICEM スライスに 2 つの LUT を組み合わせると、読み出し/書き込みポートを 1 つ、読み出し専用ポートを 1 つ持つデュアルポート 16X1 RAM を作成できます。この場合、1 つのポートで 2 つの 16X1 LUT RAM 両方に同時に書き込みを行います。もう 1 つのポートでの読み出しは独立して行われます。

分散 RAM は、FIFO や小さいレジスタファイルなど、比較的小さいエンベデッド RAM ブロックを使用するパフォーマンスの高いアプリケーションに必須です。CORE Generator は、Spartan-3 アーキテクチャ用に最適化された分散 RAM を自動的に生成します。また、分散 RAM を使用して非同期および同期 FIFO も生成します。

シングルポート RAM およびデュアルポート RAM

データフロー

分散 RAM には、次のタイプがあります。

- 同期書き込みおよび非同期読み出しのシングルポート RAM。分散 RAM に関連付けられたフリップフロップを使用した同期読み出しも可能です。
- 同期書き込みポート 1 つと非同期読み出しポート 2 つを持つデュアルポート RAM。シングルポート RAM と同様、同期読み出しも可能です。

図 1 に示すように、デュアルポートの分散 RAM には読み出し/書き込みポートが 1 つ、独立した読み出しポートが 1 つあります。

© 2003 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。

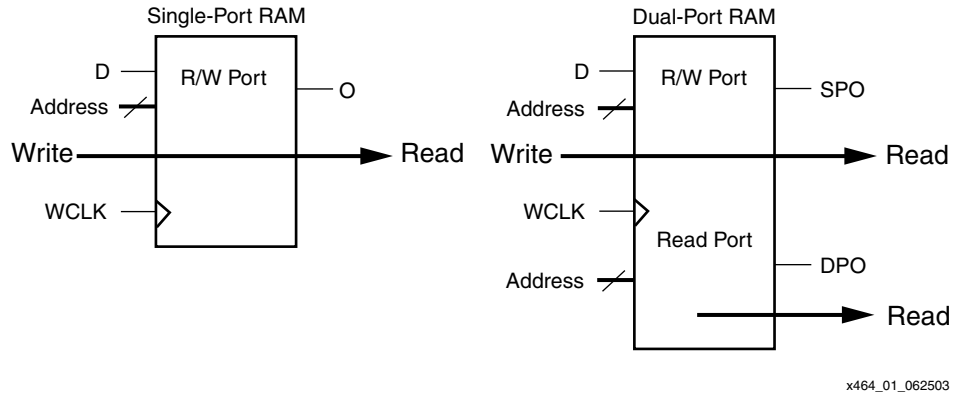


図 1： シングルポートおよびデュアルポートの分散 RAM

D 入力の書き込みと SPO 出力の読み出しは、もう 1 つの読み出し専用ポート DPO の書き込みと同時に独立して行うことができます。

書き込み操作

書き込みは、ライト イネーブル (WE) 入力によりクロック エッジに同期して行われます。WE は、デフォルトではアクティブ High になっていますが、分散 RAM 内で反転できます。WE が High の場合、クロック エッジで書き込みアドレスが取り込まれ、D 入力のデータが選択された RAM 位置に書き込まれます。

WE が Low の場合、データは RAM に書き込まれません。

読み出し操作

読み出しは、組み合わせ動作です。アドレスポート (シングルポートまたはデュアルポート モード) は非同期で、アクセス時間 (LUT ロジック遅延と同じ) 後に読み出しが実行されます。

書き込み中に読み出し

新しいデータの同期書き込みが実行されると、アドレス入力で指定されたメモリセルのデータが出力に送信されます。これは、Spartan-3 のブロック RAM で WRITE_MODE=WRITE_FIRST モードを設定した場合と似ています。図 2 は、書き込みのタイミングを示します。出力ポートにある前のデータが、クロック エッジ後に新しいデータに置き換わります。

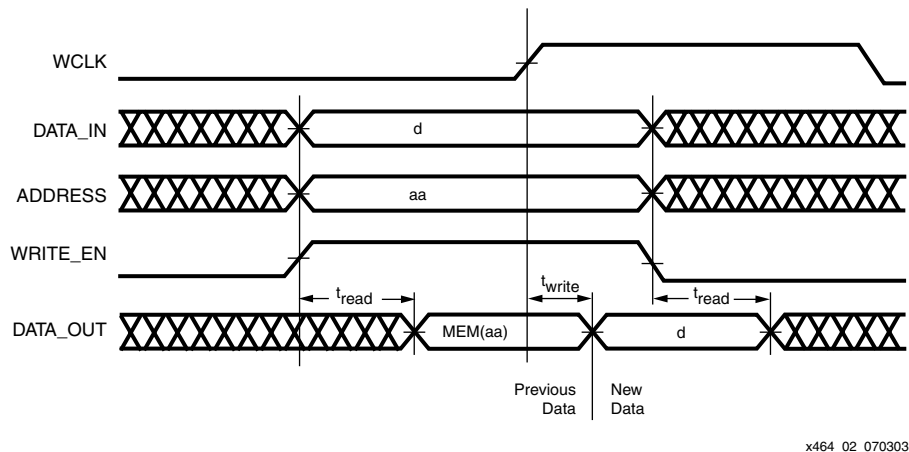


図 2： 書き込みタイミング図

特徴

- 書き込みは、クロック エッジに同期して行われます。
- 読み出しは、ロジック アクセス時間後に行われます。
- 出力は非同期で、LUT ロジック遅延のみに依存します。
- データ入力およびアドレス入力は、書き込みクロックに同期して取り込まれますが、setup-to-clock タイミング仕様に従います。ホールド タイム要件はありません。
- デュアル ポート RAM の場合、A[#:0] が書き込み/読み出し用のアドレス ポート、DPRA[#:0] が独立した読み出し専用のアドレス ポートになります。

ほかの FPGA
ファミリとの
互換性

Spartan-3 の分散 RAM は、Virtex™、Virtex™-E、Spartan™-II、Spartan™-IIE、Virtex™-II、および Virtex™-II Pro FPGA の分散 RAM と同じ動作をします。

表 1 は、各種 FPGA ファミリの CLB に埋め込まれた基本的なメモリ機能を示します。Virtex-II および Virtex-II Pro と同様、Spartan-3 の CLB には 8 つの LUT があり、128 ビットの ROM がインプリメントされています。Virtex、Virtex-E、Spartan-II、および Spartan-IIE と同様、Spartan-3 の CLB には 64 ビットの分散 RAM があります。Spartan-3 と Virtex-II および Virtex-II Pro の CLB のロジック機能は同じですが、Spartan-3 の CLB に埋め込まれた分散 RAM のビット幅は、Virtex-II および Virtex-II Pro の半分です。

表 1: 分散メモリの特徴 (FPGA ファミリ別)

特徴	Spartan-3/ Spartan-3L/ Spartan-3E ファミリ	Virtex/Virtex-E、 Spartan-II/Spartan-IIE ファミリ	Virtex-II/ Virtex-II Pro ファミリ
CLB ごとの LUT 数	8	4	8
CLB ごとの ROM ビット幅	128	64	128
CLB ごとのシングルポート RAM ビット幅	64	64	128
CLB ごとのデュアルポート RAM ビット幅	32	32	64

表 2 は、各種 FPGA ファミリでサポートされるシングルポートおよびデュアルポートの RAM プリミティブをリストしています。ここでは、1 つの CLB にインプリメント可能なプリミティブのインスタンス数を RAM タイプ別に示しています。たとえば、Spartan-3 の CLB には、32X1 シングルポート RAM プリミティブを 2 つインプリメントできます。また、16X1 デュアルポート RAM プリミティブは 2 つインプリメントできますが、32X1 デュアルポート RAM プリミティブは 1 つもインプリメントできません。

表 2：1つのCLB にインプリメント可能なシングルポートおよびデュアルポートのRAMプリミティブ数 (ファミリ別)

ファミリ	シングルポート RAM				デュアルポート RAM		
	16X1	32X1	64X1	128X1	16X1	32X1	64X1
Spartan-3	4	2	1		2		
Spartan-II、Spartan-IIE Virtex、Virtex-E	4	2	1		2		
Virtex-II、Virtex-II Pro	8	4	2	1	4	2	1

ライブラリ
プリミティブ

Spartan-3 の分散 RAM をサポートするライブラリプリミティブには、ワード数が 16 ビットのものから 64 ビットのものまで 4 種類あります。すべてのプリミティブの幅は 1 ビットです。表 3 に示すように、プリミティブのうち 3 つがシングルポート RAM で、あとの 1 つがデュアルポート RAM です。

表 3：シングルポートおよびデュアルポートの分散 RAM

プリミティブ	RAM サイズ (ワード数 X ビット幅)	タイプ	アドレス入力
RAM16X1S	16 X 1	シングルポート	A3, A2, A1, A0
RAM32X1S	32 X 1	シングルポート	A4, A3, A2, A1, A0
RAM64X1S	64 X 1	シングルポート	A5, A4, A3, A2, A1, A0
RAM16X1D	16 X 1	デュアルポート	A3, A2, A1, A0

入力データおよび出力データの幅は 1 ビットですが、複数の分散 RAM を並列接続すると、ビット数の多いメモリファンクションを簡単にインプリメントできます。

図 3 に、シングルポートおよびデュアルポートの分散 RAM プリミティブを示します。A[#:0] 信号および DPRA[#:0] 信号は、アドレスバスです。

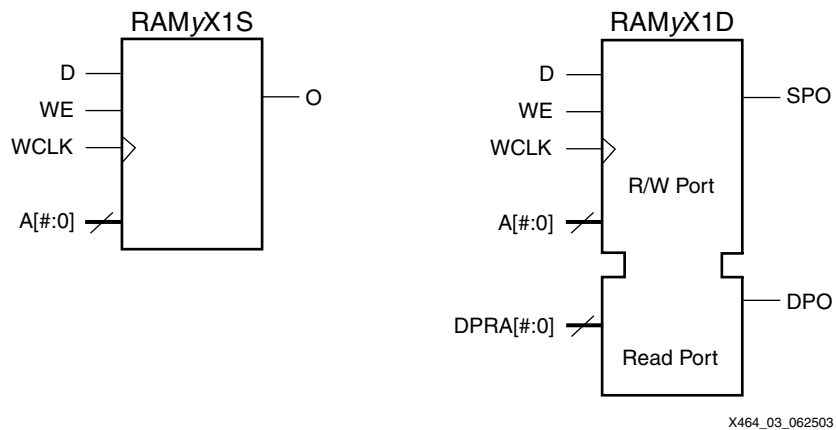


図 3：シングルポートおよびデュアルポートの分散 RAM プリミティブ

表 4 に示すように、2 ビットおよび 4 ビット RAM のライブラリプリミティブもあります。

表 4：2 ビットおよび 4 ビットのライブラリ プリミティブ

プリミティブ	RAM サイズ (ワード数 X ビット幅)	データ入力	アドレス入力	データ出力
RAM16X2S	16 X 2	D1, D0	A3, A2, A1, A0	O1, O0
RAM32X2S	32 X 2	D1, D0	A4, A3, A2, A1, A0	O1, O0
RAM16X4S	16 X 4	D3, D2, D1, D0	A3, A2, A1, A0	O3, O2, O1, O0

信号ポート

分散 RAM の各ポートは、同じメモリ セルを読み出している場合でも、動作はそれぞれ独立しています。

クロック - WCLK

クロックは、同期書き込みに使用します。データ入力ピンとアドレス入力ピンのセットアップ タイムは、WCLK ピンを基準としています。

イネーブル - WE

イネーブルピンは、ポートの書き込み機能を制御します。WE が非アクティブの場合、メモリ セルへの書き込みは実行されません。WE がアクティブの場合、クロック エッジに同期して、アドレス入力信号で指定したメモリ位置にデータ入力信号を書き込みます。

アドレス - A0、A1、A2、A3 (A4、A5)

アドレス入力は、読み出しまたは書き込みを行うメモリ セルを選択します。必要となるアドレス入力の数、ポート幅によって決まります。

メモ：VHDL または Verilog インスタンスーションでは、アドレス入力はバスではありません。

データ入力 - D

データ入力は、RAM に書き込まれる新しいデータ値となります。

データ出力 - O、SPO、DPO

シングルポート RAM のデータ出力 O またはデュアルポート RAM のデータ出力 SPO および DPO には、アドレス入力で指定したメモリセルの内容が送信されます。WE がアクティブになると、新しく書き込まれたデータがデータ出力 (O または SPO) に送信されます。

制御ピンの反転

2つの制御ピン (WCLK および WE) は、個別に反転できます。クロックを含むすべての制御信号は、ほかのロジック リソースを使用せずに、アクティブ High またはアクティブ Low にコンフィギュレーションできます。

グローバル セット/リセット - GSR

分散 RAM モジュールは、グローバルセット/リセット (GSR) 信号ではリセットされません。

グローバル ライト イネーブル - GWE

グローバル ライト イネーブル (GWE) 信号は、すべての書き込み可能なエレメントが有効になるようデバイス コンフィギュレーションの最後に自動的にアサートされます。GWE 信号を使用すると、初期化された分散 RAM の内容がコンフィギュレーション中に変更されることはありません。

GWE はグローバル信号で、デバイス全体に自動的に接続されるため、分散 RAM プリミティブには GWE 入力ピンはありません。

属性

内容初期化 - INIT

分散 RAM は、デフォルトではデバイスのコンフィギュレーション シーケンス中にすべて 0 に初期化されます。メモリの内容を 0 以外の値に初期化するには、INIT 属性を使用します。INIT 属性は、16 進数で表したビット ベクトルで、最上位ビットから最下位ビットの順に配列されています。一番右の 16 進数は、RAM の位置 (3, 2, 1, 0) を表します。表 5 に、INIT 属性の桁数をプリミティブ別に示します。

表 5 : INIT 属性の桁数

プリミティブ	テンプレート	INIT 属性の桁数
RAM16X1S	RAM_16S	4 桁
RAM32X1S	RAM_32S	8 桁
RAM64X1S	RAM_64S	16 桁
RAM16X1D	RAM_16D	4 桁

配置ロケーション - LOC

Spartan-3 の CLB には、図 4 に示すように、位置座標を持つスライスが 4 つ含まれています。分散 RAM は、SLICEM スライスにのみインプリメント可能です。SLICEM の M はメモリに関連したファンクションがサポートされることを、SLICEL の L はロジックのみがサポートされることを表します。

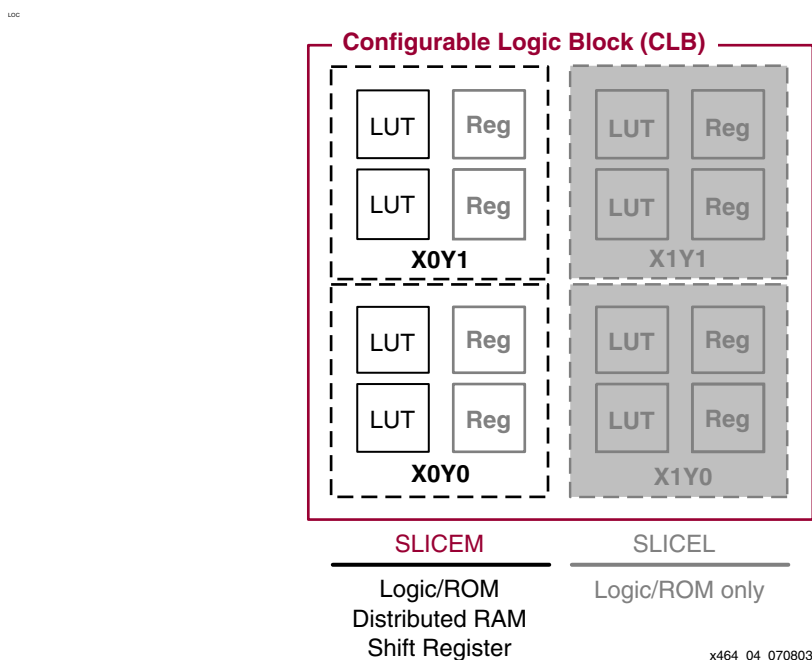


図 4 : Spartan-3 CLB 内の SLICEM スライス

LOC プロパティを分散 RAM インスタンスに割り当てると、ISE によりインスタンスが指定された位置に配置されます。図 4 は、Spartan-3 CLB のスライスの XY 座標を示しています。メモリは、SLICEM スライスのみでサポートされます。

分散 RAM の配置ロケーションの表記方法は、スライス位置の表記方法と同じなので、LOC プロパティを別のアレイにも簡単に転用できます。

たとえば、シングルポート RAM16X1S プリミティブは、SLICEM の LUT を使用します。スライス X0Y0 にインスタンス U_RAM16 を配置するには、次のような LOC 構文を使用します。

```
INST "U_RAM16" LOC = "SLICE_X0Y0";
```

図 5 に示すように、16X1 デュアルポート RAM16X1D プリミティブでは、SLICEM スライス内の 2 つの 16X1 LUT RAM を使用します。出力 SPO を持つ最初の 16X1 LUT RAM には、読み出し/書き込み共に同じアドレス A[3:0] で、読み出し/書き込みポートがインプリメントされます。2 つめの LUT RAM では、アドレス DPRA[3:0] で、独立した読み出し専用ポートがインプリメントされます。データは、アドレス A[3:0]、WE、および WCLK で制御され、両方の LUT RAM に同時に送られます。

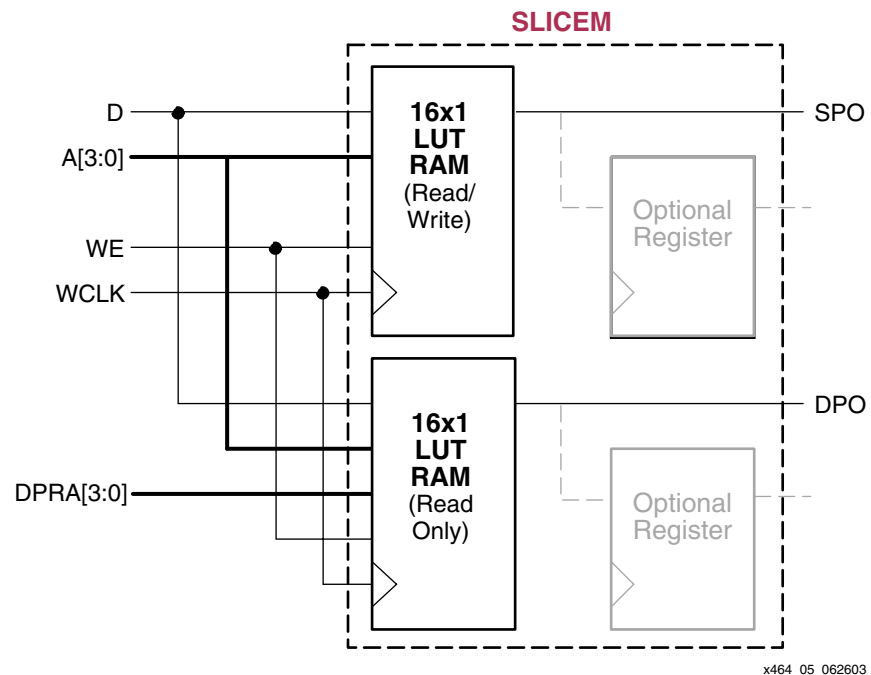
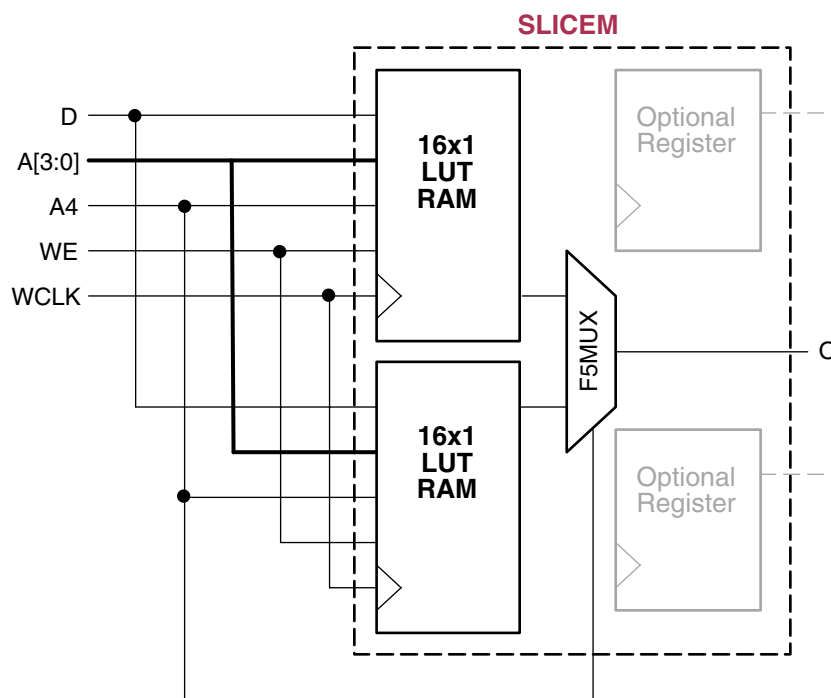


図 5: RAM16X1D の配置

図 6 に示すように、32X1 シングルポート RAM32X1S プリミティブはスライスを 1 つ使用します。RAM の 32 ビットは、SLICEM スライス内で 2 つの 16X1 LUT RAM に分割されます。A4 アドレス線は、スライス内の F5MUX マルチプレクサを介して、アクティブになっている LUT RAM を選択します。



x464_06_062603

図 6：RAM32X1S の配置

64X1 シングルポート RAM64X1S プリミティブは、SLICEM スライス を 2 つ使用します。読み出しパスは、CLB 内の F5MUX と F6MUX の両方のマルチプレクサを使用します。

分散 RAM の デザイン入力

アプリケーションで分散 RAM を指定するには、CORE Generator、VHDL、または Verilog などのデザイン入力ツールを使用できます。

CORE Generator

CORE Generator を使用すると、シングルポートおよびデュアルポート RAM、ROM、および SRL16 シフトレジスタのファンクションを持つ分散メモリ デザインを作成できます。

分散メモリ モジュールには、パラメータを指定できます。モジュールを作成するには、コンポーネント名、使用する制御入力、制御入力の極性を指定します。

オプションで、初期メモリ内容を指定できます。指定しない場合は、各メモリ位置が 0 に初期化されます。初期値を指定するには、メモリ初期化ファイルに、各メモリ位置に対して 1 行のバイナリデータを入力します。デフォルトのファイルは、CORE Generator により作成されます。または、図 7 のような係数ファイル (拡張子が .coe) を作成してください。このファイルでは、初期内容を 2、10、または 16 の基数で定義できるだけでなく、ほかの制御パラメータも定義できます。

```
memory_initialization_radix=16;
memory_initialization_vector= 80, 0F, 00, 0B, 00, 0C, ..., 81;
```

図 7：バイト幅メモリの係数ファイル

CORE Generator では、選択されたオプションと必要となるデバイスリソースを示すレポートが出力されます。ワード数の多いメモリが生成されると外部マルチプレクサが必要になることがありますが、これらのリソースは必要なロジックスライス数としてレポートされます。また、シミュレーション用に VHDL または Verilog ビヘイビアモデルも生成されます。

CORE Generator の同期および非同期 FIFO モジュールでは、分散 RAM およびブロック RAM がサポートされます。

- CORE Generator : 分散メモリ モジュール
http://www.xilinx.co.jp/ipcenter/catalog/logicore/docs/dist_mem.pdf
- CORE Generator : 同期 FIFO モジュール
http://www.xilinx.co.jp/ipcenter/catalog/logicore/docs/sync_fifo.pdf
- CORE Generator : 非同期 FIFO モジュール
http://www.xilinx.co.jp/ipcenter/catalog/logicore/docs/async_fifo.pdf

VHDL および Verilog

VHDL および Verilog ベースのデザインでは、ブロック RAM を推論またはインスタンス化できません (使用する合成ツールによって異なる)。

ブロック RAM の推論

ザイリンクスの XST や Synplicity 社の Synplify などの VHDL および Verilog 合成ツールを使用すると、記述したハードウェアを基にしてブロック RAM を推論できます。ISE Project Navigator には、ブロック RAM を推論するためのテンプレートが含まれています。Project Navigator でテンプレートを使用するには、[Edit] → [Language Templates] をクリックして [VHDL] または [Verilog] を選択し、表示された階層ツリーから [Synthesis Templates] → [RAM] を展開して、分散 RAM のテンプレートを選択します。テンプレートを切り取ってアプリケーション用のソースコードに貼り付け、必要に応じて変更を加えます。

デザインの一部で分散 RAM が推論されていても、分散 RAM をインスタンス化することは可能です。

インスタンス化テンプレート

VHDL および Verilog をベースとしたデザインの場合、各種インスタンス化テンプレートを使用して開発時間を短縮できます。ISE Project Navigator で、[Edit] → [Language Templates] をクリックして [VHDL] または [Verilog] を選択し、表示された階層ツリーから [Component Instantiation] → [Distributed RAM] を選択します。テンプレートを切り取ってアプリケーション用のソースコードに貼り付け、必要に応じて変更を加えます。

シングルポートおよびデュアルポートのプリミティブ用の VHDL および Verilog テンプレートは、ダウンロードすることも可能です。RAM_xS テンプレート (x = 16、32、64) はシングルポートモジュールで、対応する RAMxX1S プリミティブをインスタンス化します。S は、シングルポート RAM を表します。RAM_16D テンプレートはデュアルポートモジュールで、対応する RAM16X1D プリミティブをインスタンス化します。D は、デュアルポート RAM を表します。

- 分散 RAM の VHDL テンプレート
ftp://ftp.xilinx.com/pub/applications/xapp/xapp464_vhdl.zip
- 分散 RAM の Verilog テンプレート
ftp://ftp.xilinx.com/pub/applications/xapp/xapp464_verilog.zip

次に示すのは、シングルポートのテンプレートです。

- RAM_16S
- RAM_32S
- RAM_64S

次に示すのは、デュアルポートのテンプレートです。

- RAM_16D

VHDL の場合、テンプレートにコンポーネント宣言とアーキテクチャ本体があります。テンプレートの両方のセクションを VHDL デザイン ファイルに追加してください。アーキテクチャ本体のポート マップは、デザイン信号名に含める必要があります。

次に、RAM_16S モジュールの VHDL および Verilog テンプレート例を示します。

VHDL テンプレート例

```
--
-- Module: RAM_16S
--
-- Description: VHDL instantiation template
-- Distributed RAM
-- Single Port 16 x 1
-- Can also be used for RAM16X1S_1
--
-- Device: Spartan-3 Family
--
-----
--
-- Components Declarations:
--
component RAM16X1S
-- pragma translate_off
generic (
-- RAM initialization ("0" by default) for functional simulation:
INIT : bit_vector := X"0000"
);
-- pragma translate_on
port (
    D      : in std_logic;
    WE     : in std_logic;
    WCLK   : in std_logic;
    A0     : in std_logic;
    A1     : in std_logic;
    A2     : in std_logic;
    A3     : in std_logic;
    O      : out std_logic
);
end component;
--
-----
--
-- Architecture section:
--
-- Attributes for RAM initialization ("0" by default):
attribute INIT: string;
--
attribute INIT of U_RAM16X1S: label is "0000";
--
-- Distributed RAM Instantiation
U_RAM16X1S: RAM16X1S
port map (
    D    => , -- insert Data input signal
    WE   => , -- insert Write Enable signal
    WCLK => , -- insert Write Clock signal
    A0   => , -- insert Address 0 signal
    A1   => , -- insert Address 1 signal
    A2   => , -- insert Address 2 signal
    A3   => , -- insert Address 3 signal
    O    =>  -- insert Data output signal
);
--
-----
```

Verilog テンプレート例

```
//
// Module: RAM_16S
//
// Description: Verilog instantiation template
// Distributed RAM
// Single Port 16 x 1
// Can also be used for RAM16X1S_1
//
// Device: Spartan-3 Family
//
//-----
//
// Syntax for Synopsys FPGA Express
// synopsys translate_off
defparam
//RAM initialization ("0" by default) for functional simulation:
U_RAM16X1S.INIT = 16'h0000;
// synopsys translate_on
//Distributed RAM Instantiation
RAM16X1S U_RAM16X1S (
    .D(),      // insert input signal
    .WE(),     // insert Write Enable signal
    .WCLK(),   // insert Write Clock signal
    .A0(),     // insert Address 0 signal
    .A1(),     // insert Address 1 signal
    .A2(),     // insert Address 2 signal
    .A3(),     // insert Address 3 signal
    .O()       // insert output signal
);
// synthesis attribute declarations
/* synopsys attribute
INIT "0000"
*/
```

ビット数の多い分散 RAM モジュール

表 6 に、n ビット幅のメモリをインプリメントする分散 RAM の VHDL および Verilog サブモジュールを示します。

表 6: VHDL および Verilog サブモジュール

サブモジュール	プリミティブ	サイズ	タイプ
XC3S_RAM16XN_S_SUBM	RAM16X1S	16 ワード X n ビット	シングルポート
XC3S_RAM32XN_S_SUBM	RAM32X1S	32 ワード X n ビット	シングルポート
XC3S_RAM64XN_S_SUBM	RAM64X1S	64 ワード X n ビット	シングルポート
XC3S_RAM16XN_D_SUBM	RAM16X1D	16 ワード X n ビット	デュアルポート

VHDL または Verilog コードでの初期化

分散 RAM は、合成およびシミュレーション用に VHDL または Verilog コードで初期化できます。合成の場合、分散 RAM インスタンスに属性を設定し、EDIF 出力ファイルにコピーして ISE ツールでコンパイルします。VHDL シミュレーションの場合は generic パラメータを使用して属性が渡され、Verilog シミュレーションでは defparam パラメータを使用して属性が渡されます。

関連資料および リファレンス

詳細については、次の資料を参照してください。

- Spartan-3 データシート (モジュール 2) 『Spartan-3 FPGA Family: Functional Description』の「Elements within a Slice」および「Function Generator」: CLB スライスおよび分散 RAM について説明しています。
<http://www.xilinx.co.jp/bvdocs/publications/ds099-2.pdf>
- 6.3i ライブラリ ガイドの分散 RAM プリミティブのセクション
<http://toolbox.xilinx.com/docsan/xilinx6j/books/docs/lib/lib.pdf>

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
2003 年 7 月 8 日	1.0	初期リリース
2005 年 3 月 1 日	2.0	Spartan-3L および Spartan-3E FPGA を追記。