



XAPP201 (v1.2) 2000 年 2 月 25 日

# Virtex デバイスにおける各種の CAM デザインの概要

著者 : Jean-Louis Brelet

## 概要

基本的な LUT をシフトレジスタ (SRL16) や SelectRAM+™ メモリとして再プログラミングできる機能や、高速なキャリー ロジック チェーンを利用すると、Virtex™ デバイスで柔軟な CAM (Content Addressable Memory) をインプリメントできます。Spartan™ および XC4000X デバイスでも CAM をインプリメントできますが、このアプリケーション ノートでは Virtex、Virtex-E、Virtex-EM (拡張メモリ) デバイスを中心に解説します。CAM のデザインでは、アプリケーションに最良のインプリメンテーションを与える上で、Virtex デバイスの柔軟性が重要となります。このアプリケーション ノートでは CAM の概要を説明しており、アプリケーションノート XAPP202 『Content Addressable Memory (CAM) in ATM Applications』、アプリケーションノート XAPP203 『Virtex ファミリー FPGA による柔軟で高速な CAM のデザイン』、アプリケーションノート XAPP204 『Block SelectRAM+ を使用した高いパフォーマンス リード/ライト CAM』を参照しています。

## はじめに

CAM (Content Addressable Memory) は、特定の格納されている値のロケーションを迅速に検索できるように設計されている記憶アレイです。メモリ内のデータと入力と比較することによって、CAM はアレイ内に格納されている 1 つ以上の値と入力値が一致するかどうかを決定します。比較が同時に行われる場合、CAM は最大の効率で動作しているといえます。一致する値は、1 クロック サイクルで検出されます。

RAM と同じように、CAM はアレイにワードを格納します。CAM の書き込みモードは RAM と同様ですが、読み込みモードは異なります。RAM の場合は、特定のロケーションにあるワードがアドレスによって読み込まれますが、CAM の場合は、入力したデータが一致するかどうか調べられます。一致する場合は、アレイのアドレスが出力されます。

RAM のデータサイズは、アドレスラインの数によって制限されます。たとえば、10 ビットのバスでアドレスできるのは、1024 までのロケーションにある 8 ビットデータです。CAM では、ロケーションの読み込みにアドレスバスを使用しないため、このような制限がありません。1024 のロケーションにある 8 ビット値を検索するために必要となるのは、入力の 8 ビットだけです。CAM でデータが検出されると、一致信号がアクティブになり、データが一致したアドレスが出力されます。CAM では、データの検出 (読み込みモード) にアドレスラインを必要としないので、メモリサイズを容易に拡張できます。幅は、記憶エレメントとコンパレータのサイズによって決定されます。図 1 に、RAM と CAM の読み込みモードの比較を示します。

CAM の基本コアには、記憶ロケーション値と入力データを比較するコンパレータと、記憶ロケーションがあります。このアプリケーション ノートでは、速度と集積度のどちらか、あるいは両方について最適化した基本コアを設計する方法について説明します。

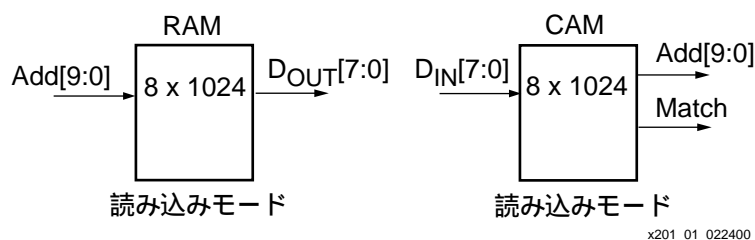


図 1: 単純な RAM と CAM の比較

## 代表的な CAM アプリケーション

CAM は、テレコム、ネットワーク、Ethernet、ATM スイッチ、各種のプロトコル アプリケーションで使用されています。特定のアプリケーションに適した CAM インプリメンテーションを決定するには、次の点を検討する必要があります。

- ワードサイズ (幅)
- ワード数 (深さ)
- 比較時間 (読み込み)
- 書き込み速度の重要性
- クロック周波数
- マスクと出力

Virtex デバイスを使用すると、最適な CAM を設計するための柔軟なアプローチを利用できます。特定のタイプの CAM がすべての CAM アプリケーションに適しているということはありませんが、柔軟なアプローチによって、デバイス内部の集積度と速度をバランスよく最適化できます。

## CAM の概要

各入力にオプションのインバータが付いている AND ゲートは、コンパレータです。したがって、4 入力の LUT は 4 ビットのコンパレータになります。LUT と専用キャリチェーンは、ワイド AND ゲートのデザインに使用できます。たとえば、各入力にオプションのインバータがある 8 入力 AND ゲートは、8 ビット データとプリエンコードされたデータのコンパレータとして使用できます。高速なキャリチェーンの最終的な出力は、比較の結果 (ワイド AND ゲートの出力) になります。図 2 に示すように、16 個の AND ゲートによるアレイは 16 ワード (深さ) × 8 ビット (幅) の CAM を表します。高性能の配線機能を利用すると、データバスがデータを各 AND ゲートに分散するので、1 クロック サイクルで同時に比較を行うことができます。リプログラマブルな LUT を使用して各 4 入力 AND ゲートをインプリメントすると、AND ゲートは基本的な 4 ビットの記憶エレメント、つまり CAM になります。この概念を Virtex 4K ビット ブロック SelectRAM+ メモリの真のデュアル読み込み/書き込みポート™ 機能に適用すると、各ブロックで 8 ビット × 16 ワードをプリデコードすることもできます。LUT のソリューションと同じように、ブロック SelectRAM+ メモリをカスケードすると、より大きな CAM をインプリメントできます。

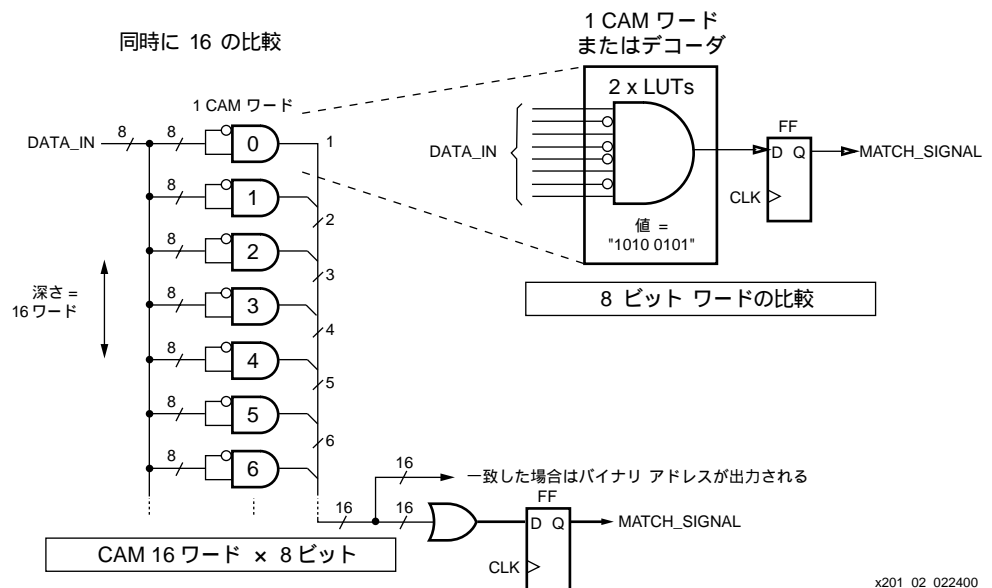


図 2: 基本的な読み込み専用の 8 ビット ワード CAM

## Virtex デバイスでの CAM デザイン

このアプリケーション ノートでは、3 つの CAM デザインを簡単に比較します。これらのデザインの詳細については、別のアプリケーション ノートで HDL 参考デザインとともに記載されています。これらのデザインは、高速キャリー チェーン、分散 RAM、ビルトイン シフトレジスタ (SRL16E)、ブロック RAM など、Virtex 特有のデバイス機能に基づいています。

基本的に、Virtex デバイスで CAM をインプリメントするには、次の 3 つの方法があります。

- 読み込みが 1 サイクルで、書き込みアクセスが遅い (16 クロック サイクル) デザイン - XAPP203
- 幅と深さが大きくて、読み込みアクセスが遅い (16 クロック サイクル) デザイン - XAPP202
- 読み込みと書き込みが 1 クロック サイクルで、ブロック SelectRAM+ メモリごとのデータ幅が 8 ビットに制限されるデザイン - XAPP204

### 基本的な CAM デザイン

読み込み専用の CAM は、各ワードを比較するデコーダ (反転入力または非反転入力がある AND ゲート) と同じです。図 2 は、基本的な読み込み専用 CAM です。Virtex CLB でワイド OR ゲートまたはワイド AND ゲートをインプリメントするには、キャリー ロジックを使用します。

Virtex デバイスは SRAM ベースで、Virtex LUT は分散ブロック RAM またはシフトレジスタ (SRL16E) としてプログラムできるので、デコーダを再コンフィギュレーションできます。再コンフィギュレーションは、CAM の書き込み動作に相当します。1 個の Virtex LUT は 4 ビットを格納できるので、4 ビット コンパレータでもあります。図 3 に、Virtex スライスのコンパレータを示します。

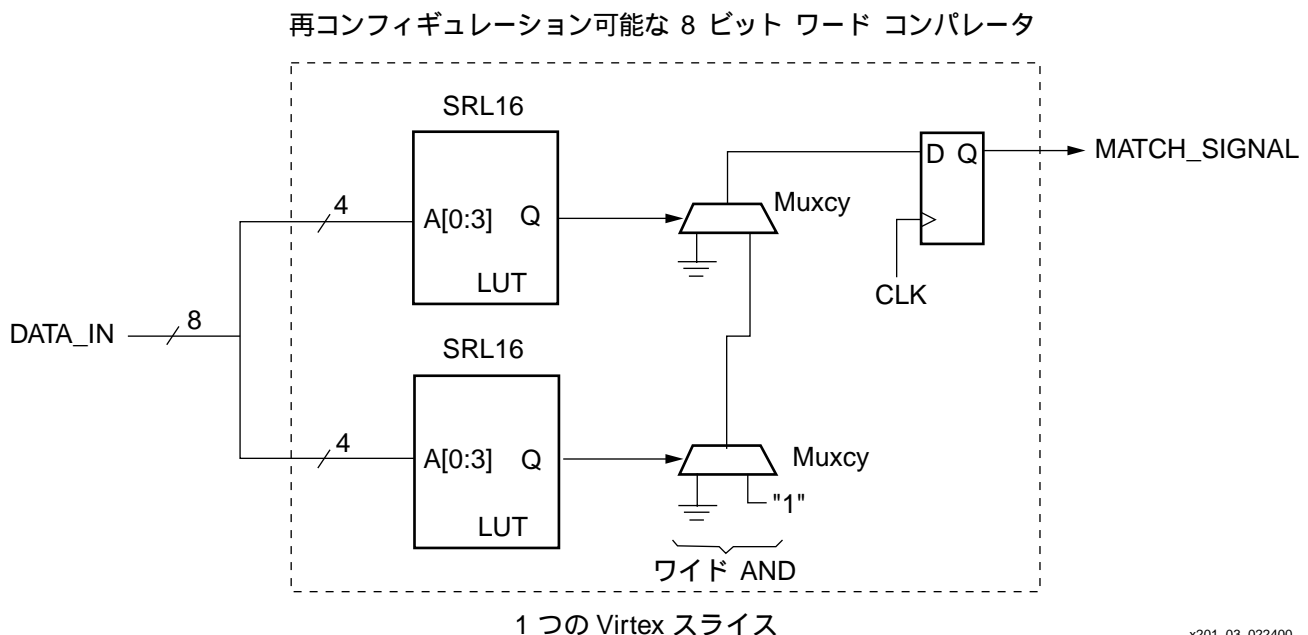


図 3: Virtex スライスによるコンパレータ

タイミングにわずかな遅延 (4 ビットごとに 1 つの MUXCY 遅延) が生じますが、ワード幅を拡張できます。Virtex デバイスでは同時に比較する数を変更できるので、深さも拡張できます。図 4 に、8 ビットワードが一致した CAM を示します。Virtex デザインでは、シフトレジスタ (SRL16E) としてコンフィギュレーションした LUT を使用して、各 CAM ワードの格納と比較を行います。RAM としてコンフィギュレーションした LUT は、Virtex、XC4000X、Spartan ファミリのいずれでも使用できます。

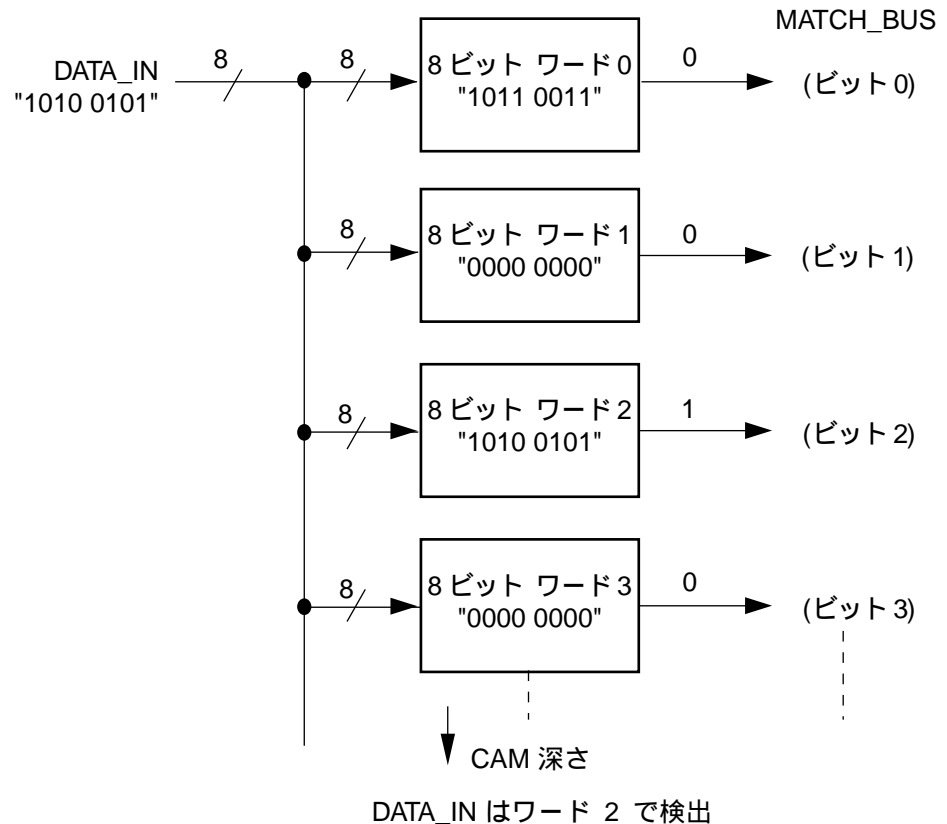


図 4: 一致した 8 ビット CAM ワード

## CAM のインプリメンテーション

### 定義

CAM の基本単位は、記憶エレメント (x ビット幅) と入力値コンパレータです。

読み込み効率は、各サイクルにおける LUT ごとのビット数によって測定されます。この値を最大にすると、パフォーマンスの面では最大の効率を得ることができます。最高速の CAM は、一致または不一致を 1 クロック サイクルで検出します。

"nb\_word" × "nb\_bit" は、幅が "nb\_bit" ビットで深さが "nb\_word" ワードの CAM を表します。

### デザイン例

表 1 に、サイズとパフォーマンスの仕様が異なる CAM のデザイン例を示します。サイズが大きい CAM の例は、Virtex アーキテクチャの幅広さを表しています。次に示す各参考デザインの説明を参照して、特定の CAM の条件に最適なソリューションを決定してください。Virtex デバイスにおけるスライスの使用率が示されているので、ユーザー デザインで CAM をインスタンス化するときインプリメンテーション結果を最適化できます。

表 1: Virtex ファミリー参考デザインの比較

参考デザイン	深さ (ワード数)	幅 (ビット数)	CAM ブロック (ビット)	読み込み		書き込み	SRL16E または RAM16 × 1 の数	ブロック RAM の数	TBUF の数
				比較	エンコード				
XAPP204	32	8	256	4.5ns	11.5ns	2 × 11.5ns	16	2	
XAPP204	128	8	1K	5.5ns	15ns	2 × 15ns	64	8	32
XAPP204	256	8	2K	8.5ns	19ns	2 × 19ns	128	16	64
XAPP203	16	16	256	7.5ns	7.5ns	16 × 8.5ns	64		
XAPP203	32	16	512	8ns	8ns	16 × 10ns	128		
XAPP203	128	40	5K	12ns	12ns	16 × 14ns	1280		32
XAPP203	256	24	6K	12.5ns	12.8ns	16 × 15ns	1536		64
XAPP202	256	16	4K	16 × 12ns	12ns	12ns	256		
XAPP202	4096	16	64K	16 × 20ns	20ns	20ns	4096		

### SRL16 デザイン

SRL16 プリミティブによる高速比較 CAM (1 クロック サイクルで読み込み) - アプリケーション ノート XAPP203。代表的なアプリケーションは、プロトコル選択です。

- インプリメンテーション = LUT ごとに 4 ビット (SRL16E または分散 RAM + MUXCY)
- 速度 = 読み込みと書き込みが 115MHz、アクセス (比較) 時間は 133MHz 以上
- 読み込み = 1 クロック サイクル
- 書き込み = 16 クロック サイクル
- 効率 = 4 ビット / LUT / サイクル (読み込み)

アプリケーション例には、アドレス デコーダ/エンコーダやコントロール ロジックなどがあります。

- XCV50 インプリメンテーション = CAM32 × 16 (26% のスライス)、86 個の LUT、128 個のシフト レジスタ (または LUT)、43 個のスライス レジスタ
- XCV300 インプリメンテーション = CAM128 × 40 (49% のスライス)、313 個の LUT、1280 個のシフト レジスタ (または LUT)、141 個のスライス レジスタ
- XCV400 インプリメンテーション = CAM256 × 24 (42% のスライス)、585 個の LUT、1536 個のシフト レジスタ (または LUT)、270 個のスライス レジスタ

基本エレメントは、LUT です。エンコードした 4 ビットを格納するので、入力データを直接比較できます。ワードの各ビットの LUT 出力は、すべて Virtex キャリー ロジック チェーンのワイヤード AND を介します。LUT への書き込みには、さらにコントロール ロジックとカウンタを使用します。

このデザインは、アプリケーション ノート XAPP203『Virtex ファミリー FPGA による柔軟で高速な CAM のデザイン』で説明されているもので、HDL 参考デザインを利用できます。

注: ダイナミック マスクが必要な場合、インプリメンテーションは LUT ごとに 2 ビットになります。書き込み時にあらかじめ定義されたマスクでは、LUT ごとの 4 ビットが維持されます。

### ATM の CAM

最適化された CAM (16 クロック サイクルで読み込み) - アプリケーション ノート XAPP202。ATM アプリケーションでは大規模な CAM が必要で、比較には複数のクロック サイクルを要する場合があります。

- インプリメンテーション = LUT ごとに 10 ビット (分散 RAM + LUT + MUXCY)
- 速度 = 80MHz

- 読み込み = 16 クロック サイクル
- 書き込み = 1 クロック サイクル
- 効率 = 0.63 ビット /LUT/ サイクル (読み込み)

#### アプリケーション例

- XCV50 インプリメンテーション = CAM256 × 16 (33%のスライス)、224 個の LUT、256 個の RAM16 × 1 (または LUT)、46 個のスライス レジスタ
- XCV400 インプリメンテーション = CAM4096 × 16 (83%のスライス)、3298 個の LUT、4096 個の RAM16 × 1 (または LUT)、261 個のスライス レジスタ

SelectRAM+ メモリセル (RAM16 × 1s) は、データの格納に使用する基本エレメントです。書き込み動作は、従来の RAM への書き込み動作と同じです。コンパレータはキャリーチェーンの LUT とワイヤード AND 回路でビルトインされ、一致した信号を出力します。

このデザインで重要なエレメントは、Virtex 分散 RAM と高速なキャリーチェーンです。

このデザインは、アプリケーション ノート XAPP202 『Content Addressable Memory (CAM) in ATM Applications』で説明されているもので、HDL 参考デザインを利用できます。

#### ブロック RAM を使用する CAM

高速なブロック SelectRAM+ CAM : CAM16 × 8 - アプリケーション ノート XAPP204。このソリューションは、読み込みと書き込みの両方を 1 または 2 クロック サイクルで行う必要があるアプリケーションに最適です。

- インプリメンテーション = ブロック RAM (各ブロック RAM+ メモリで CAM16 × 8)
- 速度 = 読み込みと書き込みが 90MHz、アクセス (比較) 時間は 200MHz 以上
- 読み込み = 1 クロック サイクル
- 書き込み = 1 消去サイクル後に 1 クロック サイクル
- 効率 = 128 ビット /RAM/ サイクル (読み込み)

#### アプリケーション例

- XCV50 インプリメンテーション = CAM32 × 8、32 個の LUT、16 個の RAM16 × 1 (または LUT)、2 個のブロック RAM、8 個のスライス レジスタ
- XCV50 = 1 列に CAM64 × 8 (ブロック RAM および隣接する CLB)、57 個の LUT、32 個の RAM16 × 1 (または LUT)、16 個の TBUF、4 個のブロック RAM、9 個のスライス レジスタ
- XCV300 インプリメンテーション = 1 列に CAM128 × 8 (ブロック RAM および隣接する CLB)、111 個の LUT、64 個の RAM16 × 1 (または LUT)、32 個の TBUF、8 個のブロック RAM、10 個のスライス レジスタ
- XCV1000 インプリメンテーション = 1 列に CAM256 × 8 (ブロック RAM および隣接する CLB)、226 個の LUT、128 個の RAM16 × 1 (または LUT)、64 個の TBUF、16 個のブロック RAM、11 個のスライス レジスタ

Virtex ライブラリには、CAM16 × 8 プリミティブと同じ値があります (16 ワード × 8 ビットの同期 CAM)。

ブロック SelectRAM+ プリミティブの RAMB4\_S1\_S16 も、読み込みと書き込みが 1 サイクルの CAM16 × 8 (幅が 8 ビット、深さが 16 ワード) を表します。

ブロック SelectRAM+ プリミティブは、各ポートを別々に特定のデータ幅に設定できるという独特な機能があるので、8 ビットワードの同時 16 出力デコーダとして使用できます。

このデザインはアプリケーション ノート XAPP204 『Block SelectRAM+ を使用した高いパフォーマンス リード/ライト CAM』で説明されているもので、HDL 参考デザインを利用できます。

## おわりに

Virtex デバイスの独特な機能により、幅広いシステム レベルソリューションが提供されます。これらの機能は、CAM デザインへのアプローチにおける柔軟性も提供します。CAM ソリューションは、他の RAM ソリューションに類似しています。小さい RAM は、分散 RAM 機能を使用してインプリメントできます。中規模サイズの RAM は、ブロック RAM 機能でインプリメントできます。CAM の条件は、次の方法で実現できます。

- SRL16E ベースのインプリメンテーションは、大きなワード幅と高いパフォーマンスを提供します。参考デザインは、アプリケーション ノート XAPP203 『Virtex ファミリ FPGA による柔軟で高速な CAM のデザイン』にあります。
- 分散 RAM ベースのインプリメンテーションは、ATM アプリケーションに最適な大きさのワード幅と深さを提供します。参考デザインは、アプリケーション ノート XAPP202 『Content Addressable Memory (CAM) in ATM Applications』にあります。
- ブロック RAM ベースのインプリメンテーションは、8 ビット幅で高いパフォーマンスの読み込み / 書き込みを提供します。参考デザインは、アプリケーション ノート XAPP204 『Block SelectRAM+ を使用した高いパフォーマンス リード/ライト CAM』にあります。

これらのアプリケーション ノートは、設計者が自分の CAM システムの条件に最適な Virtex デバイス ソリューションを決定する参考資料となります。

## 改訂履歴

次の表に、この文書の改訂履歴を示します。

日付	バージョン番号	改訂内容
1999 年 8 月 19 日	1.0	初期リリース
1999 年 9 月 23 日	1.1	Virtex-E の初期アップデート
2000 年 2 月 25 日	1.2	Virtex-EM ( 拡張メモリ ) デバイス用の再構成と更新