



XAPP655 (v1.1) 2003 年 1 月 17 日

ミックス バージョン IP ルータ (MIR)

概要

このアプリケーション ノートでは、最大 4 個までのギガビット イーサネット ポートを持つミックス バージョン IP ルータ (MIR) のリファレンス デザインについて説明します。IPv4 および IPv6 のホストとギガビット イーサネット ネットワークに直に接続されたルータが混在した状態で、複数のギガビット イーサネット ネットワークが動作しており、さらに、それらのネットワークがルータを介してその先のノードに接続されているような場合に、MIR は有効です。Virtex-II Pro ファミリーを使用した MIR の特徴は、IPv4 から IPv6 への内部構造的な移行期、またインターネット上の外部的な移行期において、ルータ機能のスムーズな向上およびルータ パフォーマンスの維持が実現できることです。

このルータの機能には、IPv4 間または IPv6 間の通常のルーティング、IPv4 用に IPv6 パケットをカプセル化 (その逆も同様)、IPv4 から IPv6 へのパケット変換 (その逆も同様) があります。パケットごとの機能選択およびこれらの機能の実行は、ロジックとプロセッサの組み合わせで行われます。基本的には、頻度が高く制御の少ない機能を実行する場合はロジックで処理し、その他の機能はすべてプロセッサが行うようになっています。

このリファレンス デザインは、XC2VP7 デバイスのリソースを対象にしており、大量に送信されるパケットをバージョン 4 に変換するように設定されています。これらのパケットは、すべてロジックで処理され、変換されます。このとき、バッファ間でのパケットのコピーおよびパケット受信と後続処理への転送間の遅延はありません。このため、4 ポート間に独自の内部接続ネットワークが必要になります。また、ロジックでマルチスレッドを行うパケット受信とパラレルで実行する必要があります。バージョン 6 パケットまたはバージョン 4 の一部のパケットは、PowerPC プロセッサへ送られます。基本的に PowerPC プロセッサは、通常のマスタ-スレーブ関係ではなく、ロジックに対してつねにスレーブとして動作します。

このリファレンス デザインは、Virtex-II Pro デバイスのさまざまな機能を使用します。たとえば、RocketIO トランシーバ、FPGA (ロジックまたはメモリ)、プロセッサ ローカル バス (PLB) またはデバイス コントロールレジスタ (DCR) コア コネクト (CoreConnect) アーキテクチャを使用するエンベデッド PowerPC プロセッサがあります。

はじめに

この MIR リファレンス デザインは、設定可能なデザイン空間の広い MIR ルータについて説明しており、Virtex-II Pro Platform FPGA を使用したデザインを実例で示します。Virtex-II Pro Platform FPGA に関する詳細は、<http://www.xilinx.co.jp/products/platform/> を参照してください。

このリファレンス デザインの主な特徴を次に示します。

- ファンクション シミュレーションを使用した十分なテスト
- 合成可能
- 確認済みのタイミングおよびリソースを使用した配置配線
- 検証済みの主要な内部コンポーネント ブロックを組み込んだ実デバイスのテスト

このリファレンス デザインは、サービス IP ルータとしてそのまま使用することはできません。簡潔なデモ手段を提示するために、ネットワーク インターフェイスの技術が一部省略されています。また、こ

© 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。

のリファレンス デザインは、実際の外部ネットワーク インターフェイスと接続して検証されているものではありません。

2002 年^[1, 2] に開催された 2 つの国際会議では、MIR デザインに関する研究結果が発表されました。これらの資料には、MIR リファレンス デザインについての詳細情報が含まれているため大変有効です。

技術的な背景

インターネット プロトコル (IP) バージョン 4 およびバージョン 6

1981 年以来、IPv4 がインターネットの標準プロトコルでしたが、1994 年に新たに IPv6 が導入されはじめ、1998 年以降、推奨規格となっています。IPv6 へのプロトコル移行の要因は、機能の拡張および簡略化などさまざまですが、最大の要因はアドレス空間が 32 ビットから 128 ビットに拡大されることです。これまで幅広く IPv4 が利用されていたため、IPv6 へ完全に移行するには時間を要します。新プロトコルへの移行期間には、2 つのバージョンを混在させたルータおよびホストを使用することを推奨します。その最初のステップである IPv6 のテスト ベッドである「6bone」については、次のサイトを参照してください。 <http://www.6bone.net>

密接に関連する 3 つの RFC (Internet requests for comments) 文書があります。RFC2893^[3a] は、IPv6 ホストおよびルータの移行メカニズムについての文書です。IPv4/IPv6 両方のノードについておよび IPv4 トンネルに IPv6 パケットをカプセル化するメカニズムが記述されています。RFC 2473^[3b] には、IPv6 トンネルにパケットをカプセル化するメカニズムが記述されています。RFC 2765^[3c] は、IPv4 形式から IPv6 形式へ、またその逆の変換アルゴリズムについて記述されています。この文書には、アドレスの割り当ておよびルーティングについては記述されておりませんので、必要な場合は RFC2766^[3d] を参照してください。各々状況における複雑な問題に使用されるさまざまなパケット分割のメカニズムを適用します。IPv6 でのトンネリングは、初期の段階では重要ではありません。IPv4 形式のパケットが IPv6 環境に点在している場合、IPv6 への移行期の最終段階で必要な処理です。IPv6 専用のノードが IPv4 専用のノードと通信する必要がある場合のみ、最後の手段としてこの 2 つの IP バージョン変換が必要になります。

プログラマブル ロジックを使用するパケット処理について

次の出版文書には、プログラマブル ロジックを使用したパケット処理の例が含まれています。

- End-station protocol handling of specific stacks (e.g., TCP/IP by Fallside and Smith^[4]) and generalized stacks (e.g., protocol harness of Brebner^[5]);
- Simple packet routing (e.g., Lockwood^[6,7] et al) with low-width non-CAM lookup; and
- Simple packet filtering (e.g., Dirmar^[8] et al and Guccione^[9] et al) with high-width CAM lookup.

これは、既存のハードウェア インプリメンテーションにおいて、FPGA を直接 ASIC の代替として使用する場合の追加説明です。MIR により調べられる重要なことは、プログラマブル ロジック インプリメンテーションの特徴を生かせるパケット処理タイプをシステム全体の中で識別することです。

ネットワーク プロセッサ

MIR リファレンス デザインは、Virtex-II Pro がネットワーク プロセッサ市場において発展的で確かなソリューションであることを示します。Virtex-II Pro デバイスは、プログラム機能およびさまざまなアプリケーションでの使用に対応できるパフォーマンス属性を使用して、幅広い用途に対応します。Network Processing フォーラム^[10] では、ネットワーク プロセッサ エリアでの標準化について議論されています。このリファレンス デザインは、これらの標準に準拠したデザインではありません。標準化は、システム全体の中での特定のネットワーク処理をサポートするのに必要な、共通のアーキテクチャの特徴を示すだけでなく、ネットワーク プロセッサの対応するパラダイムにも言及しています。

Virtex-II Pro の属性について

このリファレンス デザインでは、主に 4 つの Virtex-II Pro 属性の組み合わせを使用します。その属性は、プログラマブル ロジック、PowerPC プロセッサ、RocketIO 高速リンク、およびブロック メモリです。このリファレンス デザインは、ロジックとプロセッサの相互作用、および FPGA を 1 つ使用し

た場合の高速通信のトレードオフを検証します。このアプリケーション ノートでは、選択可能ないくつかの例を説明します。

Virtex-II Pro ファミリーは、さまざまな属性を持ち、PowerPC コアがなく 4 つの RocketIO インターフェイスを持つ XC2VP2 デバイスから、4 つの PowerPC コアと 24 の RocketIO インターフェイスを持つ XC2VP125 デバイスまで幅広く対応します。システム ゲート数およびブロック RAM のビット数は、デバイスにより異なります。MIR は、プロセッサを 1 つのみ使用するため、複数のプロセッサによる動作の競合は生じません。ターゲット デバイスは、XC2VP4 または XC2VP7 です。このリファレンス デザインは、最大で 4 つの RocketIO インターフェイスを含むため、XC2VP4 にも対応していますが、多くのロジック リソースを持つ XC2VP7 デバイスをターゲットにしています。

Virtex-II Pro 開発キット (PDK)

PDK には、PowerPC 405 コアおよび RocketIO トランシーバの SWIFT シミュレーション モデル、IBM CoreConnect バス モデル、コンパイラ/アセンブラ/リンカ ツール、および 3 つのサンプル デザインが付属しています。この開発キットでは、ハードウェアには Verilog、ソフトウェアには C 言語が使用されています。低レベルシステムの初期化には、少しだけアセンブラ コードが使用されています。

MIR の仕様

MIR ルータを使用すると、次の主要な 2 つの事項が実現します。

1. さまざまな機能に焦点をあてた、ハードウェア プログラム機能の使用
2. マスタ制御デバイスとは対照的に、使用頻度が低くタイミングに厳しくない機能を実行する場合、スレーブとしての従来プロセッサの使用

ミックスバージョン IPv4/IPv6 ルータを使用したこのデザインでは、上記の 2 つが実現できます。これらのプロトコルには非常に優れた機能があり、特定の時間に使用頻度の最も高い機能には、ロジックを使用し、使用頻度の低い機能に対しては、プロセッサを使用して対応できます。

このデザインは、パフォーマンスを維持しながら、前述したパケット処理の説明に続き、ここではネットワーク プロセッサの柔軟性を示します。このデザインは、インターネット プロトコル用にデザインされていますが、上位層プロトコルなど異なるプロトコルへ拡張するための機構が含まれています。

対象となる製品は、4 つのギガビット イーサネット ポートを持つミックス モード IP ルータです。IPv4 および IPv6 のホストとギガビット イーサネット ネットワークに、直に接続されたルータが混在した状態で複数のギガビット イーサネット ネットワークが動作しており、さらにそれらのネットワークがルータを介してその先のノードに接続されているような構成を対象としています。ギガビット イーサネットは、RocketIO トランシーバでサポートされている通信規格の一つです。

MIR ルータの機能を、次に示します。

- IPv4 間または IPv6 間の単純な通信
- IPv4 パケットの中に IPv6 パケットをカプセル化
- IPv6 パケットの中に IPv4 パケットをカプセル化
- IPv4 から IPv6 へのパケット変換、またはその逆

パケット毎の機能選択は、ソース アドレス、入力ポート、出力ポート、デスティネーション アドレスなどにより推論されます。パケットを送信する出力ポートに接続されているルータの種類により、推論される機能が決定されます。この機能の決定プロセスおよび実行は、ロジックとプロセッサの組み合わせで行われます。また、ロジックとプロセッサの使用比率は、ルータの特徴により異なります。使用頻度が高く、あまり制御を必要としない機能は、ロジックで実行され、その他の機能はプロセッサで実行されます。ルーティング テーブルなどのルータ管理は、プロセッサで行われます。

ネットワーク上の平均 IP パケット サイズは、従来どおり 1,600 ビットと想定します。このルータは、4 つのポートを最大限に活用して、1 秒間に平均約 125 万パケットのレートを維持する必要があります。最新の超高速 IP ルータ (参考資料 [11] を参照) を使用した場合、最低でもこの平均レートは十分クリアできます。

ネットワーク環境

ネットワーク環境の変化により、必要とされる動作や設定が制限されるため、ルータの動作環境は特に重要です。前述のとおり、MIR はさまざまなギガビット イーサネットに接続できるように設計されています。インターネット接続に関して、動作機構は 1 つのイーサネットに関し、1 つのルータを介してグローバル インターネットへ単一の接続をすることを想定します。MIR は、原理的には大きな変更を行わずにこのようなルータとして機能できますが、通常は外部へ接続するルータとして使用されません。

ネットワークのアドレス指定

動作機構は、クラス B の IPv4 アドレスを 1 つと、001/TLA/NLA の IPv6 アドレスを 1 つ割り当てます。IPv4 アドレス タイプが変化した場合、たとえば 1 つあるいは複数のクラス C アドレスが割り当てられた場合でも、このリファレンス デザインで要求される機能に大きな変更は必要ありません。このアドレス空間内で、イーサネットに接続されたデバイスへアドレスが割り当てられます。IPv4 アドレスには、イーサネットに対する 8 ビットのネットワーク識別子と、8 ビットのインターフェイス識別子が含まれています。また、この区分を主な機能を変更せずに、2 つの 8 ビット区画に分割、変更できます。IPv6 アドレスは、イーサネットに対する 16 ビットのネットワーク識別子と 64 ビットのインターフェイス識別子を含んでいます。アドレス空間の上限は、イーサネットの最大数、(この場合は、MIR ポートが 4 つあるため) および、イーサネットのステーション数の標準的な制限 (このネットワーク技術に対しては、一般的に 1024) により異なります。

派生アドレス テーブル フォーマット

MIR は、接続している各デバイスに関して、IPv4 のみ、IPv6 のみ、または IPv6/IPv4 の混在のどのタイプのデバイスであるかの情報を保存する必要があります。また、外部インターネット接続に対しても、この情報を格納する必要があります。このミックス モードの設定で、より多くの IP 接続を可能にすることが MIR の役目です。ただし、リファレンス デザインでは簡略化のため、ユニキャストのみをインプリメントしています。マルチキャストおよびブロードキャストに関してはサポートしていません。

オンチップ メモリのみ依存しているシステムの最大の問題は、アドレス ルックアップ テーブルの保管が必要なことです。前述した IPv4 および IPv6 アドレス空間について考慮すると、MIR には簡素化された機構が使用され、含まれる各デバイスに対して 64 ビットのみ使用します。したがって、1 つのブロック RAM に 256 デバイスのルーティング情報を格納することができます。64 ビットのフォーマットは、次のとおりです。

ビット数	目的
1	IPv4 プロトコルの使用を示す
1	IPv6 プロトコルの使用を示す
4	MIR ポート番号 (0, 1, ...) – 現在 2 ビットのみ使用
10	MIR ポートでのインターフェイス番号。IPv4 アドレスの下位 8 ビットを使用し、残りの 2 ビットは未使用
48	IEEE MAC-48 アドレス。IPv6 アドレスの下位 64 ビットの最初と最後の 24 ビットを結合したアドレス

最初の 2 ビットを使用して、ミックスバージョンのルータをガイドし、どのような機能が必要であるかを決定します。残りの 62 ビット (実際には 58 ビット使用される) から、Ethernet MAC アドレス、IPv4 アドレス (必要な場合)、および IPv6 アドレス (必要な場合) が検索されます。

Ethernet MAC アドレスは、下位 48 ビットにあります。

IPv4 アドレスは、固定 16 ビットのプレフィックス長を持つクラス B アドレス空間です。次の 8 ビットは、その機構内のネットワーク番号を示し、この 8 ビットを検索するには、各々がシングル ネットワークに接続している MIR ポートのナンバーが駆動する 4 入力のルックアップ テーブルを 1 つ使用します。最後の 8 ビットは、ネットワーク上のインターフェイス番号を示し、10 ビット インターフェイス番号のフィールドにあります。

IPv6 アドレスは、固定 48 ビットのプレフィックス長を持ち、その値は割り当てられた 001/TLA/NLA 値になります。次の 16 ビットは、その機構内のネットワーク番号を示しており、この 16 ビットを検索

するには、各々のシングル ネットワークに接続している MIR ポートのナンバーが駆動する 4 入力のルックアップ テーブルを 1 つ使用します。最後の 64 ビットは、ネットワーク上のインターフェイス番号を示し、EUI-64 の定義に基づき、固定の 16 ビット パターン (FFFEh) で MAC-48 アドレスを半分に分割して、(FFFEh) を間に挟み組み合わせられた形式である必要があります。

4 ビットのポート番号フィールドは、16 ポートまで指定でき、Virtex-II Pro ファミリのほとんどのデバイスに十分対応できます。10 ビットのインターフェイス番号フィールドは、イーサネット デバイスの最大数 (1024) に対応できます。

システム機能

IPv4 および IPv6 パケット処理には、次の機能の組み合わせが必要です。

IPv4 パケットの処理

- IPv4 アドレスの抽出および一致
- IPv4 ローカルアドレスの検索
- IPv4 から IPv6 へのパケット変換
- IPv4 over IPv6 トンネル検索
- IPv6 パケットへ IPv4 パケットをカプセル化
- 過大 IPv4 パケットの分割
- IPv4 から IPv4 へのカプセル化解除
- IPv4 から IPv6 へのカプセル化解除

IPv6 パケットの処理

- IPv6 アドレスの抽出および一致
- IPv6 ローカルアドレスの検索
- IPv6 から IPv4 へのパケット変換
- IPv4 over IPv6 トンネル検索
- IPv4 パケットへ IPv6 をカプセル化
- 過大 IPv6 パケットの分割
- IPv6 から IPv6 へのカプセル化解除
- IPv6 から IPv4 へのカプセル化解除

さらに、その後の処理や伝送のため、MIR 内部で IPv4/IPv6 パケットをハブやポートへ送る内部配線も重要な機能です。これらの動作を行うメカニズムについては、次のセクションで説明します。

コデザインおよびトレードオフ

サービス システム アーキテクチャ モデルは、コデザインおよびトレードオフの基本的な項目を示します。別のコンポーネントに対してサービスを実行するために、どのコンポーネントを使用すべきかを考慮した場合、重要な 4 つの MIR 機能があります。

- Coarse-grain サービス (パケット単位)
 - ロジックが、サーバーとしてプロセッサへパケットをオフロードする
 - プロセッサが、サーバーとしてロジックへパケットをオフロードする
- Fine-grain サービス (イン パケット機能)
 - ロジックが、プロセッサを使用して機能を実行する

- プロセッサが、ロジックを使用して機能を実行する

従来のシステムでは、その現象は明確でした。たとえば、2つの **Coarse-grain** サービスは、パケット受信（ロジック）およびパケット送信（プロセッサ）に相当します。一方、2番目の **Fine-grain** サービスは、アクセラレータで使用されるロジックに相当します。**Fine-grain** サービスの最初の機能は、プロセッサ以外が中心となるデザイン空間を示すため、最も注目されています。主としてロジックでデザインされた **MIR** システムは、**MIR** 環境からの入力要求に従って動作します。

MIR には、ロジックまたはプロセッサ インプリメンテーションに2つの機能が要求されます。ひとつは、IPv4 または IPv6（あるいはその他）として入力されるパケットを識別すること、もうひとつは、簡単なアドレス形式（フォーマットおよびローカリティ / ノンローカリティなど）の認識です。**Coarse-grain** サービスのプロセッサ インプリメンテーションを行わない場合、識別はロジック回路で行われます。

残りの3つのシステム機能を次に示します。

1. 検索機能 - ルーティング テーブルおよびトンネリング テーブルの検索
2. ルーティング機能 - 内部バッファおよびパケットの送信
3. 操作機能 - パケットのカプセル化およびカプセル化解除、分割、変換など

これらの3つの機能が **MIR** システムのコーデザイン空間を調べるために必要ですが、統合システムを構築する上で、実際にはルーティング機能および操作機能の組み合わせなど、機能間の相互作用が生じます。

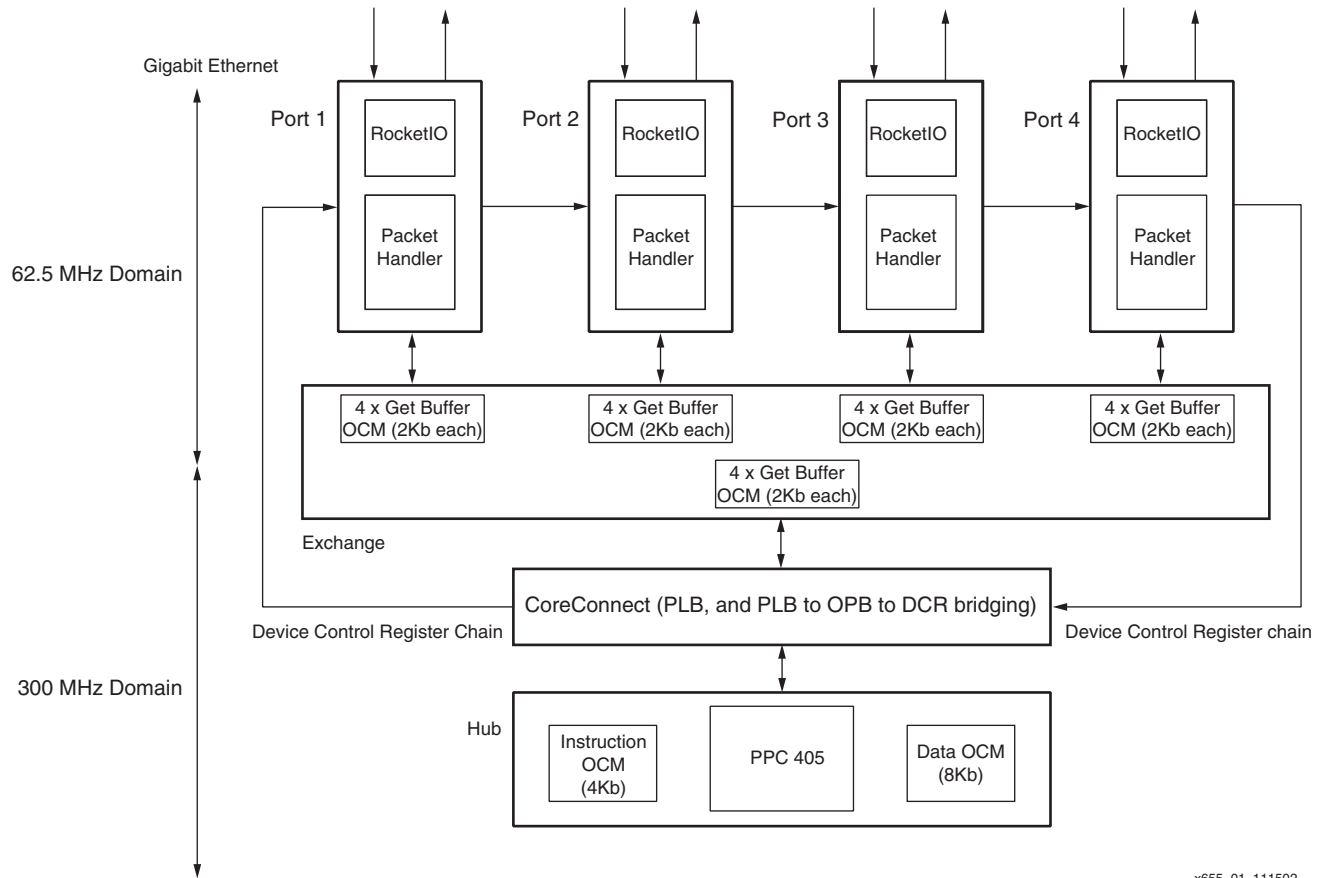
MIR リファレンス デザインを定義するにあたって、3つすべての機能の実現性を調査するという体系的な方法ではなく、ある特定の機能に焦点を当てた選択的な方法を使用しました。ロジック中心のシステムで、新しいタイプのデザイン オプションの使用に焦点を置いています。

テーブル検索には、ロジックまたはプロセッサのどちらかが **fine-grain** スレーブとして動作します。その他の機能は、ロジックが検索オプションを調査します。

- プログラマブル ロジックでエンコードされた、ルックアップ テーブルを使用した固定アルゴリズム
- 内容参照可能 (CAM) メモリの使用
- ランダム アクセス メモリの使用 (検索前にハッシュ法を使用する場合もある)

従来のプロセッサを使用した検索方法は、テーブル検索の最良な手段であるということは周知の事実です。プロセッサをスレーブとして使用して、ロジックからリモート ファンクション コールを実行するためのオーバーヘッドを調査することにより、新たな検索方法が開発される可能性があります。

図 1 に、MIR システムのブロック図を示します。



x655_01_111502

図 1：MIR ブロック図

リファレンス デザインのインスタネーション

このリファレンス デザインは、Virtex-II Pro デバイスの能力を最大限に活用するための先進的なデザインであり、さまざまな新しいパケット処理機能を含みます。またミックスバージョンの IP 空間において、特定のポイントに焦点を当てています。

- 通過する IPv4 パケットおよび IPv6 トンネルへ向かう IPv4 パケットは、ロジックで処理されます。
- その他の IPv4 パケットおよび IPv6 パケットは、プロセッサで処理されます。

IPv6 が大半を占める環境で、IPv4 形式のパケットを配送するには、このような処理が最適なのですが、トレードオフが生じます。実際には、パケット処理の環境および詳細手順が簡略化される一方、すべての重要なシステム属性をデモすることができる最低限の設定が選択されました。内部的なパケットスイッチングおよびパケット解析/変更を行うための実験的なロジック関連機能についてのデモが課題になります。

モジュール詳細

すべてのシステム モジュールの一覧は、「付録 A」を参照してください。主なシステム モジュールを次に示します。

- トップ モジュール - システム全般
- ポート モジュール - 外部とのインターフェイスおよびロジックによるパケット処理
- 変換モジュール - ポートとハブ間のパケット変換およびバッファリング
- コア コネクト モジュール - コア コネクト (CoreConnect) アーキテクチャのインスタネエート

- ハブ モジュール - システムの PowerPC プロセッサ部分

テストについて

このリファレンス デザインには、シンプルな機能シミュレーション テスト ベッドがあり、システム 的な動作の観測と調査が研究できます。また、実際のハードウェアでテストする場合に最適な、診断出力ラインをオプションとして追加することもできます。

クロックドメイン

図 1 に、2 つのクロックドメインを示します。62.5 MHz のドメインは、パケット処理用のロジック回路に適用され、ギガビット イーサネット インターフェイスと同期します。300 MHz ドメインは、PowerPC システム デザインに適用されます。クロック調整には、各クロックドメインに対して、Virtex-II Pro の Digital Clock Module (DCM) を 1 つ使用します。

RocketIO トランシーバ

RocketIO インターフェイスは、ギガビット イーサネット用に設定されており、シリアライザ - デシリアライザおよびその他のシステムとの間は、32 ビット幅のインターフェイスです。イーサネット CRC の自動生成および自動チェックは、このインターフェイス内で行われます。この CRC チェックには、PowerPC プロセッサでソフトウェアを実行するオプションとして、トランシーバ内に内部パラレルまたはシリアルループバックを設定するモードもあります。

デモ用に、トランシーバへ接続されたロジック回路は完全なギガビット イーサネット PHY フレーム形成およびデフレーム機能をインプリメントしており、4 の倍数だけでなく、あらゆるバイト長のパケットに対して可能です。このロジック回路は、MIR システムの連続したテストで使用されます。ただし、このロジック回路は、標準ネットワークスイッチのギガビット イーサネットのすべての機能に関しては、サポートしていません。つまり、このロジック回路では、基本的なパケットのフレーミングおよびデフレーミング以外の機能、スタートアップ時のオートネゴシエーションおよびフロー制御のための送信中断などは実行できません。

上位層である Ethernet MAC 層の処理については、「[ポート回路におけるパケット処理](#)」で説明します。

CoreConnect アーキテクチャ

一般的な IBM CoreConnect アーキテクチャ (参考資料 12) は、PowerPC プロセッサをロジック回路にインターフェイスするために使用されます。このバスの機構は、システム内で PowerPC プロセッサの動作要件を整合し、ロジックですべての処理が行われる部分でのボトルネックを取り除きます。CoreConnect アーキテクチャでは、次の 2 つの機能を実行します。

1. PowerPC プロセッサの、内部交換ネットワーク内にあるパケット バッファへのアクセス
2. PowerPC プロセッサの、ポート内の制御レジスタおよびステータスレジスタへのアクセス

このアーキテクチャのバックボーンであるプロセッサ ローカルバス (PLB) は、64 ビット転送ができます。パケット バッファへアクセスする場合は、このバスが必要です。また、制御レジスタおよびステータスレジスタへアクセスする場合は、PLB とオンチップ ペリフェラルバス (OPB) を接続するブリッジ、および OPB と デバイス コントロールレジスタ (DCR) バスを接続するブリッジがあります。これら 3 つのバスの標準ブロックおよびブリッジは、Virtex-II Pro Platform FPGA で使用できます。

このシステムの CoreConnect モジュールは、PLB アービタ、PLB デバイス、OPB アービタ、および OPB デバイスをインスタンス化します。さらに、このモジュールは DCR チェーンを接続して、そのほかの制御レジスタおよびステータスレジスタへもアクセスできます。PLB マスタ デバイスには、CPU ICU ポートおよび DCU ポートの 2 つがあり、PLB スレーブ デバイスは、PLB-to-OPB ブリッジのみです。OPB マスタ デバイスは、PLB-to-OPB ブリッジのみであり、OPB スレーブ デバイスには、OPB アービタの制御レジスタおよび OPB-to-DCR ブリッジの 2 つがあります。DCR マスタ デバイスは、OPB-to-DCR ブリッジ (CPU ではない) のみであり、DCR スレーブ デバイスには、PLB アービタの制御レジスタ、PLB-to-OPB ブリッジ、ポート制御レジスタおよびステータスレジスタの MIR

チェーン、ノンクリティカル割り込みコントローラ、クリティカル割り込みコントローラの 5 つがあります。現在のシステムでは、あまり使用されないコンポーネントがありますが、それらは将来的な機能拡大用です。たとえば、OPB アービタ (唯一のマスタ デバイス) および 2 つの割り込みコントローラは、あまり使用されません。

コア コネクタのアーキテクチャを使用することで、PowerPC プロセッサとロジックの接続が実現しました。パケット バッファへアクセスする場合、PowerPC プロセッサから直接、DSOCM (data-side on-chip memory) を使用することもできます。ただし、この場合は、バッファ部分での配線遅延およびデコード回路のロジック遅延により、PowerPC クロック ドメインへのタイミング問題が確実に生じます。制御レジスタおよびステータス レジスタへのアクセスの場合、DCR レジスタへ直接アクセスするには、*mfdcr* および *mtdcr* マシン コード命令を使用する方法もあります。ただし、コア コネクタを介す間接的なアクセス方法には、PowerPC プロセッサおよび DCR レジスタの間にマップされたメモリ インターフェイスがあります。直接メモリへアクセスすることにより、レジスタへのアクセス プログラムが、より単純で柔軟になります。

メイン システム コンポーネント および機能

次の 3 つのセクションでは、システムの主な 3 つの部分について説明します。

1. すべての処理段階においてバッファ間のパケット コピー回避機能を持った変換モジュール内での、パケットの移動およびバッファリングについて。
2. ポート モジュール内のロジック回路によるパケット処理、およびロジック内での制御フローのマルチスレッドの使用について
3. PowerPC プロセッサ内でのソフトウェア動作、および「CoreConnect アーキテクチャ」を使用してインプリメントした場合のロジック インターフェイス回路について

変換ネットワーク

変換ネットワークは、前セクションの「ルーティング機能」と関連しています。MIR アーキテクチャは、4 つのポートおよび 1 つのハブからなる 5 つの構成要素があります。内部ルーティングの基本的な要件は、ある構成要素から別の構成要素へパケット移動 (または、パケットの部分的な移動) などがあります。接続に関する問題は、コンピュータ ネットワーク、パラレル マルチプロセッサ コンピュータ、およびシングル デバイスなどのあらゆるコンピュータを対象に発生します。初期の多くの研究結果をガイドランスとして利用できます。

内部接続ネットワークで注目すべき点は、スタートポロジ回路がハブに集中する場合です。このトポロジを使用した接続での問題は、ハブでのシリアル化が影響し、ポートでパラレル化した場合のメリットが全く得られないことにあります。通常、集中化を避けるために、バス型またはリング型トポロジの使用、また、特異なものとしてはハイパーキューブ型やそれにより派生されるものを使用します。MIR リファレンス デザインには、スター型とは正反対のもので、すべてのポイントに対して専用の接続を持つ方法が使用されます。このリファレンス デザインは、合計 20 個 (5 個のソースから 4 つのディスティネーション) の単一方向の接続があります。この n^2 のリソース手法は、ポートが 4 つのみの場合は問題ありませんが、ポート数が多くなると高価になります。実際の手法は、システムの中にブラックボックスとして入っているため、オプションでリソースの使用を抑えた接続を行うことができます。

インプリメントされた正確な機構は、デュアル ポート バッファを 1 つ使用し、20 本の内部接続に対してブロック RAM を 1 つずつ配置し、また読み出しおよび書き出し用にそれぞれメモリ ポートを 1 つずつ使用します。したがって、各通信ポートおよびハブには、読み出し用に 4 つの バッファがあり、その他の 4 つのコンポーネントからの書き込みが可能なバッファが 1 つあります。各ブロック RAM では、最大幅 36 ビットをフルに使用しているため、内部接続ネットワークのデータ パスも 36 ビット幅になります。

フル状態のバッファに書き込みが実行される場合、書き込まれるワードは廃棄され、オーバーフロー フラグが立ちます。バッファに完全なパケットを格納するためのメカニズムが、主に次のセクションで説明されている理由に起因し、内臓されています。そのため、パケット送信中にバッファのオーバーフローが生じた場合でも、パケットのデータ片が残ることはありません。

各ポートおよびハブは、ラウンド ロビン方式で処理用または伝送用にパケットを選択します。この選択プロセスは、ポートのロジックおよびハブのプログラムによりインプリメントされます。

ハブでは、各プロセッサの特質に応じて、各バッファ (読み出し用に 4 個、書き込み用に 4 個) コンテンツへのインターフェイスはランダム アクセスです。CoreConnect アーキテクチャの 64 ビット PLB インターフェイスを介して 36 ビット ワードへアクセスします。各バッファ コンテンツへの直接アクセスに加え、プロセッサは DCR レジスタへもアクセスし、バッファからまたはバッファへのパケットの書き込み/読み出しの位置を示すためのポインタを取得します。すべてのポインタに対する読み出しアクセス、またハブに制御されているポインタへの書き込みアクセスがあります。

ポートのインターフェイスは、ロジック回路の性質が反映して、FIFO のような機能を果たします。つまり、36 ビット ワードが FIFO ベースのバッファへの書き込み、またはバッファからの読み出しが実行されますが、ポート ロジックによる明確なメモリ アドレスの指定はありません。FIFO のような機能と表現した理由については、次のセクションでも説明しますが、実際にインプリメントされた書き込みメカニズムが、本来シリアルでアドレスがないにもかかわらず、非常に適切な機能を果たすからです。一方、読み出しメカニズムは、完璧な FIFO 的機能を果たします。

ポート回路におけるパケット処理

パケット受信

操作機能

パケットを受信する場合、大部分の処理は「操作機能」が必要になります。ロジックによるインプリメンテーションおよびプログラムによるインプリメンテーションのトレードオフを検証するには、この機能グループは最も重要です。あらゆる最終段階のシステムにおいては、複数のインプリメンテーションメカニズムが混在します。これは、ネットワーク環境および負荷に依存して頻繁に要求される機能はロジック、頻度の低い機能はプログラムにインプリメントされます。MIR リファレンス デザインのアーキテクチャは、ロジックにインプリメントされた機能のみを使用して処理されるパケット タイプをサポートします。機能のインプリメント方法以外で MIR を使用する利点としては、バッファ間でパケットコピーを最小限に抑える (理想的には複製を除去する) ことが挙げられます。パケットが複製されるとルータおよびエンド システム間の効率が非常に悪くなってしまいます。

プログラムによるすべての必要な機能のインプリメンテーションについては、リファレンス デザインの PowerPC ソフトウェアの章で説明します。デモンストレーションをより確実なものにするため、サブセットを 1 つだけロジックでインプリメントしていますが、このリファレンス デザインでは、ロジックでインプリメントした機能をパケット処理フローに結合するために、確実な基礎構造を作ることに重点を置いています。この基礎構造には、パケット処理ロジックにおける次の 3 つの重要な先進的機能を含みます。

1. 内部パケットのブロードキャスト送信

この最初のコンポーネントでは、バッファ間のパケットの複製問題の回避処理を実行します。一見したところでは、最低 1 つのパケットの複製が必要に思われます。なぜなら、入力バッファでパケットを受信した後、後続の伝送または必要な操作機能とは無関係なプロセッサ処理のために、適切な出力バッファにパケットを配置する必要があるためです。この場合に適用されるソリューションは、最初に交換ネットワークでブロードキャスト方式を使用して、4 つすべての出力バッファにコンテンツを配置してパケット受信を行うことです。パケット ヘッドが受信され解析されると、すぐにブロードキャスト送信が停止され、パケットの後続ワードが指定したバッファにのみ書き込まれます。

2. ロジック回路におけるマルチスレッド

2 番目のコンポーネントでは、パケット受信と並列にパケット処理を行い、レイテンシを軽減します。マルチスレッド方式は、各ポートで実行されている処理に導入されています。さまざまな回路ブロックを同時に動作させるのは容易ですが、パケット アクセスにはアービトレーションが必要です。この場合、適用されるソリューションは、バッファ メモリ アクセス時間が処理を行うクロック サイクルよりも短いため、異なるクロック位相でスレッドを操作することです。1 つの例として、パケット受信を行うスレッドは、RocketIO レシーバと同じクロックを使用できますが、パケット操作を行うスレッドは、各々

の packets 受信ステップのクロック位相を使用します。バッファへの書き込みインターフェイスを備えることにより、メモリ クロック サイクルを変化させて、この (デュアルポート ブロック RAM の 1 つのポートに対する) デュアル アクセスが実行でき、またスレッドを操作して、通常の厳しい FIFO の規則を回避できます。

「MIR 動作の例」に、マルチスレッド方式を使用した 2 つの例を示します。これらの例には、後続の伝送に対してパケットのカプセル化を行った 2 つのケースがあります。

ロジック回路でのマルチスレッド方式の一般原理については、MIR システム デザインから導かれ、また論文^[2]にも詳細説明があります。各スレッドは、ロジック内の有限状態 マシン (FSM) に対応しています。FSM の最初の状態からの遷移をイネーブル/ディスエーブルにする 1 ビットの実行フラグを使用して、個々のスレッドは、その他のスレッドから有効または無効にされます。スレッド間の通信には、競合回避のために確実に同期したロジック内の共有レジスタを使用します。

3. バッファ処理の詳細

上記で述べたエミュレートされた複数ポート アクセス以外のバッファ関連ソリューションは、このコンポーネントにあります。可能性のあるすべてのバッファへ、パケットを最初にブロードキャスト送信すると、目的のバッファが設定された後、中止されたフラグメントをそのまま残してしまうため、パケットが完全に受信、操作、および送信されるとすぐに、バッファへパケットを収容するメカニズムが必要でした。この時点では、すぐに後続の伝送を行うことができます。36 ビット メモリ ワードの中の余分な 4 ビットを、パケット (32 ビット データ) の最初および最後のワードの中に使用します。最初のワードの中で、それらが実際にパケット データが始まる前に、オフセット (0 から 15 ワード) を示し、カプセル化 ヘッダーの有無が考慮されます。最後のワードでは、4 ビット マスクを生成し、32 ビット ワード パケットの最後のバイトの位置を示します。

検索機能

操作方式の機能に加えて、パケット受信にはアドレス検索機能が必要です。追加のデュアルポート メモリが、変換モジュールの中に含まれており、各々のポートで、ロジックが検索をするためのルックアップ テーブルへのデュアル アクセスと、PowerPC プロセッサのテーブルの初期化および更新をサポートします。PLB を使用する 64 ビット アクセス、および共有するパケット バッファに対して、類似のメカニズムを使用します。アドレス テーブルのフォーマットについては、「派生アドレス テーブル フォーマット」で説明しています。各テーブル エントリは 64 ビットであるため、メモリ アクセス メカニズムを一致させます。リファレンス デザインでは、2 つの ブロック RAM を使用して、最大 512 テーブル エントリができます。

これは、テスト デザインであるため、各ポートに追加された検索回路は、IPv4 アドレスに簡単な要約関数を使用してテーブルに索引をつけて、1 クロック サイクル内の検索をしています。より巧妙な検索アルゴリズムの使用については開発されておりません。テスト オプションとして、PowerPC プロセッサをスレーブとして使用し、検索を行う方法があります。クリティカルまたはノンクリティカルな割り込みコントローラをオプションで設定できる機能がこのリファレンス デザインに組み込まれており、この機能はデフォルトでは無効になっています。

パケット送信

ポートにおけるパケット送信は、解析や処理が不要なためパケット受信より単純です。簡単な有限状態 マシンを使用して、イーサネット物理層でのフレーミングおよび RocketIO インターフェイスへのパケット コンテンツのための連続したワードを出力します。このパケット コンテンツは、ポートの出力バッファの中から検索されます。状態 マシンはパケット長をバイトとして送信完了時に 32 ビット ワード境界に対して正しい物理層フレーミングを生成します。

PowerPC ソフトウェア

組み込まれたテストおよびサービス ソフトウェア

「付録 A」で説明している方法でパラメータを 1 つ使用すると、ソフトウェアを設定して、サービス モードまたはテスト モードでの設定ができます。サービス モードは通常の動作モードであり、PowerPC プ

ロセッサは入力される IP パケット処理をスレーブとして適切に実行します。テスト モードの場合、PowerPC プロセッサは、よりアクティブな役割を果たします。最初に PowerPC プロセッサは、ポートの入出力をリング状に接続して、ポートによる一連の送信用パケットを生成します。ルータ内にあるすべての内部バッファと配線がテストされます。

必要なルータ機能のプログラム インプリメンテーションについての説明は、IP パケットの分割に関するものを除いて、ほぼ終了しました。MIR リファレンス デザイン内のポートのロジック回路でインプリメントされたこれらの機能は、PowerPC プロセッサの補助は必要ないのでコメントアウトできます。

テスト モードの場合、システムのポート モード (2、3、または 4) によって、3 種類のテスト パケットのうちの 1 つが最初に生成されます。テスト パケットは、受信パケットの処理に含まれているアドレス検索メカニズムであるため、IPv4 デスティネーション アドレスが含まれています。アドレスを選択することにより、交換ネットワークを通る 20 本すべてのパスが、出力ポートからループバック接続を経由して入力される受信パケットにより、確実に使用されるようになります。PowerPC ハブで戻ってしまうパケットもあり、このパケットは他へ送信されません。また、2 ポートまたはそれ以上のポート間を永遠にループするパケットもあります。最も厳しいタイミング条件を与えるために、最小のイーサネット パケットを使用します。

命令およびデータ格納用メモリ

プロセッサのすべての命令およびデータは、ブロック RAM のオンチップ メモリに格納され、シングルチップ システムが実現します。プログラム コードは、4 K メモリ サイズに相当する 2 つのブロック RAM を使用して格納できます。このメモリを拡張させる場合は、システムを設定をして 8K に変更できます。原理上、プログラム コードは、16K の命令キャッシュの中に完全に格納できるため、実行中に異なるメモリへのアクセスを回避できますが、現在この命令キャッシュは無効にされています。8K メモリを倍にするオプションはありますが、プログラム データおよびスタックなどは、問題なく 8K メモリに格納できます。このプログラムのデータ キャッシュは、無効にされています。

PLB アクセスおよびデータ キャッシュ

36 ビット バッファへのアクセスおよび 64 ビット ルックアップ テーブルへのアクセスには、PLB において、64 ビット動作を使用します。64 ビット動作を可能にするために、プロセッサには標準の 32 ビット データ バスおよびデータ キャッシュがあります。このデータ キャッシュはプログラム データ メモリに対しては、無効にされていますが、パケット バッファ領域およびルックアップ テーブル領域には有効です。キャッシュ ラインのフェッチおよびフラッシュは、64 ビット動作を実行します。PLB は、DCR バスへの接続媒体として使用され、交換ネットワークでのパケットバッファ ポインタの読み出し/書き込み、また、ポートでの制御レジスタやステータス レジスタの読み出し/書き込みにも使用されます。

MIR 動作の例

パケット受信時のリファレンス デザインの MIR システム動作を説明するために、2 つの対照的なデザインを例に挙げます。この 2 つの例は、後続の転送のために、新しいパケットの中に受信パケットをカプセル化します。ただし、一方は、ロジック回路で完全に処理されており、もう一方は、PowerPC プロセッサのアシストが必要になります。

パケット処理ロジックには、必要に応じてアクティブになる 7 つのスレッドがあります。そのうちの 1 つは、32 ビット ワードのパケット受信に関連する主要スレッドです。受信したワードは、クロックの後縁で有効になるため、この主要スレッドは、31.25 MHz クロックの前縁を使用します。その他の 6 つのスレッドは、解析および処理の両方またはいずれかを行い、主要スレッドのクロックと同期するために、このクロックの後縁で動作します。

1. MAC/SNAP ヘッダ認識スレッド
2. IPv4 ヘッダ認識スレッド
3. IPv6 ヘッダ認識スレッド (リファレンス デザインにはありません)
4. IPv4 アドレス検索スレッド

5. IPv6 カプセル化スレッド
6. MAC ヘッダ機構スレッド

すべての場合において、各パケットが受信（物理層フレーミングワードの受信）されるとすぐに、MAC/SNAP ヘッダ認識スレッドが動作を開始します。また、すでにこの段階で、IPv6 パケットをカプセル化するスレッドが動作を開始します。受信するパケットは、つねに関連バッファ内の 10 ワードのオフセットに格納され、競合することなく、ヘッダを構成するためにバッファの全部にスペースを残してあります。早期に動作を開始することにより、最小のイーサネットパケットが受信されるのに要する時間よりも確実に早く完了できます。これらの次にアクティブになるスレッドの選択には、パケットに含まれる内容により異なります。次の 2 つの例を参照してください。

例 1 : IPv6 パケットに IPv4 パケットをカプセル化

上記に説明した基本動作の後に続くスレッド動作のシーケンスを、次に示します。

1. MAC/SNAP ヘッダ解析で IPv4 パケットであることが認識されると、IPv4 ヘッダ認識スレッドが動作開始
2. IPv4 ディスティネーションアドレスが取得されると、IPv4 アドレス検索スレッドが動作開始

この場合、すべてのスレッドが正常に完了します。操作の結果、オリジナルの IPv4 ヘッダの直前に、6 ワードの MAC/SNAP ヘッダと、それに続いて新たに 10 ワードの IPv6 ヘッダが構成されます。この場合、オリジナルの MAC/SNAP ヘッダは上書きされます。すべての処理は、パケットが完全に受信されるまでの時間内で行われるため、指定のポートからすぐに送信できる状態になります。

例 2 : IPv4 パケットに IPv6 パケットをカプセル化

上記に説明した基本動作の後に続くシングルスレッドの動作を、次に示します。

1. MAC/SNAP ヘッダ解析で IPv6 パケットであることが認識されると、IPv6 ヘッダ認識スレッドが動作を開始します（このスレッドは、実際には何も行いません）。スペキュレイティブな IPv6 カプセル化スレッドも、冗長的に完了まで継続されますが、影響はありません。パケット受信が完了すると、ハブの PowerPC ソフトウェアが IPv6 パケットを解析して、IPv4 パケットへカプセル化し、その後、特定の出力ポートからパケットを送信できるようになります。

リファレンス デザイン

このリファレンス デザインは、Verilog で記述されており、次のザイリンクス FTP サイトから入手できます。

<ftp://ftp.xilinx.com/pub/applications/xapp/xapp655.zip>

ファンクション シミュレーション

このリファレンス デザインは、ModelSim SE 5.5e を使用して十分にファンクション シミュレーションを行い、検証されたものです。このシミュレーションは、RocketIO インターフェイスがリング状に接続され、PowerPC プロセッサはテストモードで動作する必要があります。

合成 : 実際のリソースの必要条件およびタイミング

前述のとおり、このリファレンス デザインは、Virtex-II Pro ファミリの XC2VP7 デバイスを対象にしています。実機上のデバイス XC2VP7 でのテストのため、最小構成の 2 ポートバージョンを使用し、デザインフロー サイクル時間およびオンチップリソースの使用を最小限に抑えました。このバージョンは、XC2VP7 デバイスの有効スライスの約 70 パーセントを使用しており、配置配線を行う上では全く問題はありませんでした。また、プロセッサクロックドメインの動作スピードも、テスト目的のため、ターゲットスピード (300MHz) の半分の 150MHz です。このリファレンス デザインは、このように緩和されたタイミング制約を満たすことはできますが、配置を調整しない限りこれ以上のタイミング最適化を行うことはできません。このシステムでのプロセッサの役割は部分的であるため、タイミング問題はそれほど重要ではありません。ロジッククロックドメインは、62.5MHz のターゲットスピードで動作するため、容易にタイミング要件を満たすことができます。また、3 ポートバージョンを

使用した場合でも、ほとんどすべてのスライスを使用した状態ですが、配置配線に関しては問題なく XC2VP7 に適応します。この 3 ポートバージョンでは、詳細なタイミング検証を行っておりません。4 ポートバージョンでは、合成後のスライス使用率が 100% を少し超えた結果となりました。XC2VP7 には、これより少し小規模なバージョンの配置配線は可能ですが、この結果は、タイミング制約を与えない場合のものであります。

4 ポートバージョンに必要なロジックリソースの割合を次に示します。

- パケット処理 1 ポートにつき 10% (合計約 40%)
- 変換 35%
- CoreConnect 20%
- その他のロジック 5%

ステータス

このリファレンス デザインは、Virtex-II Pro ファミリのアドバンス デザインの例として提示しています。PowerPC ソフトウェアなどを含む内部アーキテクチャの採用は、完全かつ最適な方法ですが、実践的なルータとして利用するには、さらなる手順が必要になります。そのひとつは、ギガビット イーサネット プロトコル処理。もうひとつは、処理を行った実際のイーサネット スイッチに接続されている実機 Virtex-II Pro デバイス上でテストすることです。

参考資料

1. Brebner G, "Single-chip Gigabit Mixed-version IP Router on a Virtex-II Pro device", Proc. 10th IEEE Symposium on Field-Programmable Custom Computing Machines, Napa, 22-24 April 2002, IEEE Computer Society Press 2002, pp. 35-44.
2. Brebner G, "Multithreading for logic-centric systems", Proc. 12th International Conference on Field-Programmable Logic and Applications, Montpellier, France, 2-4 September 2002, Springer LNCS 2438, pp.5-14.
3. Internet Requests for Comments (RFC) Accessible at <http://www.rfc-editor.org/rfc.html>
 - a. RFC 2893 "Transition Mechanisms for IPv6 Hosts and Routers"
 - b. RFC 2473 "Generic Packet Tunneling in IPv6 Specification"
 - c. RFC 2765 "Stateless IP/ICMP Translation Algorithm (SIIT)"
 - d. RFC 2766 "Network Address Translation - Protocol Translation (NAT-PT)"
 - e. RFC 2374 "An Aggregatable Global Unicast Address Format"
4. Fallside H and M Smith, "Internet Connected FPL", Proc. 10th International Conference on Field-Programmable Logic and Applications, Springer LNCS 1896, 2000, pp.48-57.
5. Brebner G, "Highly reconfigurable communication protocol multiplexing element for SCOPH", Reconfigurable Technology, Proceedings of SPIE, 4525, 2001, pp.99-106.
6. Braun F, J Lockwood and M Waldvogel, "Reconfigurable router modules using network protocol wrappers", Proc. 11th International Conference on Field-Programmable Logic and Applications, Springer LNCS 2147, 2001, pp.254-263.
7. Lockwood J, N Naufel, J Turner and D Taylor, "Reprogrammable network packet processing on the Field Programmable Extender (FPX)", Proc. 9th International Symposium on Field Programmable Gate Arrays, ACM Press 2001, pp.87-93.
8. Ditmar J, K Torkelsson and A Jantsch, "A dynamically reconfigurable FPGA-based content addressable memory for Internet protocol characterization", Proc. 10th International Conference on Field-Programmable Logic and Applications, Springer LNCS 1896, 2000, pp.19-28.
9. Guccione S, D Levi and D Downs, "A reconfigurable content addressable memory", Proc. 7th Reconfigurable Architecture Workshop, Springer LNCS 1800, 2000, pp.882-889.
10. <http://www.npforum.org>
11. <http://www.whnet.com/giga.html>
12. <http://www.chips.ibm.com/products/coreconnect/>

付録 A

MIR システム パラメータ

このリファレンス デザインは、オプションの構造に関連するコンポーネント、内部バス アドレス指定および外部ネットワーク アドレス指定に関連する、さまざまなシステム機能の簡単なパラメータ化について示します。パラメータおよびオプション機能の設定は、ハードウェア用のファイル (`global_params.v`)、およびソフトウェア用の (`global_params.h`)ファイルの中に含まれる 2 つのグローバルパラメータで制御されます。この 2 つのファイルは、「付録 C」のリストに示されています。この 2 つのファイルにあるパラメータは、ファイルの中にコメントで示されているように、一致している必要があります。また、他のファイルにおいても同様で、ファイル内のコメントで指示されています。

システムパラメータ要約を次に示します。

- アーキテクチャ上のパラメータ
 - ギガビット入力/出力ポートの数 (2、3、または 4)
 - サービス モードまたはテスト モード動作の選択
 - PowerPC プロセッサの命令格納用メモリ サイズを倍にするオプション (4Kb から 8 Kb)
 - PowerPC プロセッサのデータ格納用メモリ サイズを倍にするオプション (8Kb から 16Kb)
 - 割り込みコントローラ ロジックを追加
 - シミュレーション テスト用の、入力/出力ポートのリング状接続オプション
 - モジュールからの検証用出力信号の追加オプション
 - 実機ハードウェア テスト用の、AFX ボード検証用出力の追加オプション
- システム アドレス空間パラメータ : PLB、OPB、および DCR のアドレス指定
- ネットワーク アドレス指定パラメータ
 - MAC、IPv4、およびIPv6 アドレス
 - ローカル ネットワーク内の関連するルータ アドレスおよびトンネル アドレス

これらのパラメータについての詳細は、グローバルパラメータ ファイル内に示されています。

付録 B

MIR 回路モジュール

表 1: モジュールのまとめ (n-port コンフィギュレーションの場合、n = 2、3、または 4)

File	Midule(s)	Instantiated By
top.v	top	- (if real); testbench (if simulated)
exchange.v	exchange_module	top
	port_select_module	exchange_module (n times)
	hub_port_buffer_module	exchange_module (n times)
	port_hub_buffer_module	exchange_module (n times)
	port_port_buffer_module	exchange_module (n(n-1) times)
hub.v	hub_module	top
port.v	port_module	top (n times)
packet.v	packet_module	port_module
	DCR_module	packet_module
	transmitter_module	packet_module
	receiver_module	packet_module
ocm.v	isocm_module	hub_module (one or two times)
	dsocm_single_module	hub_module (one or two times)
	dsocm_dual_module	hub_port_buffer_module, port_hub_buffer_module
	gsocm_dual_module	port_port_buffer_module
	lutocm_dual_module	exchange_module (n times)
coreconnect.v	coreconnect_module	top
	plb_bus_logic	coreconnect_module
	opb_bus_logic	coreconnect_module
clocks.v	basic_clocks_module	top
	GIO_clocks_module	basic_clocks_module

表 1: モジュールのまとめ (n-port コンフィギュレーションの場合、n = 2、3、または 4)

File	Midule(s)	Instantiated By
Files from PDK	PPC405	hub_module
	GT_ethernet_4	port_module
	GT	GT_ethernet_4
	RAM16_S36	isocm_module (two times)
	RAM16_S9	dsocm_single_module (four times)
	RAM16_S36_S36	{ds,gs,lut}ocm_dual_module (x 1,1,2)
	IBUFG, DCM, BUFG	basic_clocks_module, GIO_clocks_module
braminit.v	– (BRAM initialization)	(included by) top (func.sim. only)

メモ:

13. coreconnect_module は、エンベデッド PPC 405 リファレンス デザインから派生し、使用するモジュールには、アービタ、plb2opb_bgo、plb_bram_cntlr、pb_arbiter、opb_dcr_brg_top、INTC (およびサブモジュール) があります。また、plb_bus_logic および opb_bus_logic モジュールには、簡単なサブモジュールが数個あります (これらは、この表にリストされていません)。オプションで、2 つの INTC モジュールを追加できます。

付録 C

MIR ソース ファイル

MIR リファレンス デザイン フォルダは、PDK にある make 機能をすべて使用します。このフォルダには、flow.cfg ファイルおよびサブフォルダ (par、sim、sw、syn、および sys) が含まれます。このリファレンス デザインに関連するソース ファイルを次に示します。PDK の make プロセスは、この他にも多数の規準となる名前を持つ派生ファイルを生成します。

回路ソース ファイル (「付録 B」のモジュールのまとめに対応する)

サブフォルダ sys/verilog 中のファイルを次に示します。

- clocks.v
- coreconnect.v
- exchange.v
- global_params.v (「付録 A」を参照)
- hub.v
- ocm.v
- packet.v
- port.v
- src.lst (PDK の make プロセスで使用される)
- top.v

プログラム ソース ファイルおよび関連ファイル

サブホルダー sw 中のファイルを次に示します。

- global_params.h (「付録 A」を参照)
- ppc.h
- kernel.S
- main.c

- mapfile
- code.bmm
- Makefile
- (その他、生成されたファイルには、kernel.o、main.o、code.elf、code.lst、code.map、bram_init.ucf、bram_init.v があります。)

シミュレーション ファイル

サブフォルダ sim/testbench/verilog 中のファイルを次に示します。

- testbench.v

サブフォルダ sim/func_sim 中のファイルを次に示します。

- wave.do

配置配線制約ファイル

サブフォルダ par 中のファイルを次に示します。

- top.ucf

改訂履歴

次の表は、このアプリケーション ノートの改訂履歴です。

日付	バージョン	改訂
2002/11/19	1.0	初版リリース
2003/01/17	1.1	「リファレンス デザイン」 へのリンクを追加