

このホワイトペーパーでは、アルテラの FPGA システム設計におけるネットワーク・オン・チップ (NoC) アーキテクチャの利点について解説します。NoC アーキテクチャでは、システム・オン・チップ・デザインのサブシステム間の通信に対して、ネットワーク技法とネットワーク技術の応用が可能です。NoC インタコネクタ・アーキテクチャは、独立した層設計や最適化の実現に向けたサポートなど、従来の非 NoC インタコネクタにはない多くの利点を備えています。Quartus®II 開発ソフトウェアに搭載されるアルテラの Qsys システム統合ツールは、アプリケーションの要件に基づいて、FPGA 向けに最適化された柔軟な NoC 実装を自動生成します。また、Qsys インタコネクタは、同程度のレイテンシおよびリソース特性における動作周波数が高くなり、また従来のインタコネクタと比較して f_{MAX} が最大 2 倍に向上します。

はじめに

FPGA デバイスの集積度が 100 万ロジック・エレメント (LE) 超まで増加する一方、設計チームは、増大する性能要件に伴う、大規模でかつ複雑なシステムを、より短期間で完成させる必要があります。設計者が、システム・レベル設計ツールを使用することにより、最小限の労力で、高性能なシステムを迅速に設計することが可能となります。

Qsys では、NoC アーキテクチャを使用してシステム・トランザクションを実装します。Qsys インタコネクタには、FPGA の高性能動作をサポートする機能が含まれており、与えられたアプリケーションを実装するのに、必要最小限のリソースのみ使用する柔軟性のあるネットワーク・インタコネクタや、サポートされているシステムによって変換可能なパケット形式、および命令ネットワークと応答ネットワークを分離して同時並行性を高め、リソース使用量を低減するネットワーク・トポロジーなどが挙げられます。

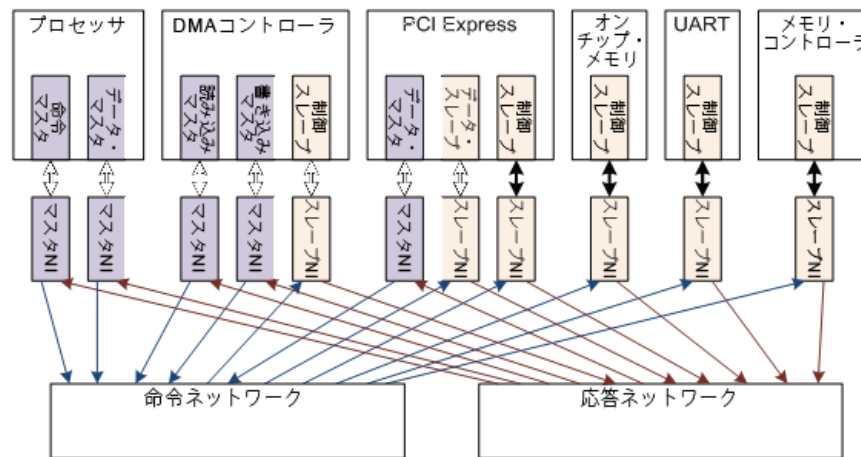
このホワイトペーパーでは Qsys ネットワークの実装と利点について解説し、従来のインタコネクタ・システムと Qsys インタコネクタ・システムの性能結果を比較します。その結果から、NoC 実装では、レイテンシ特性が同程度の場合に周波数性能が高くなり、パイプライン化オプションを有効にすると周波数を最大 2 倍まで高めることができる、ということをご理解いただけます。

NoC インタコネクタの理解

NoC インタコネクタは、エンティティ間の通信に関する問題を、例えば、システムのノード間でのトランザクションの伝送方法や、トランザクションをパケットにカプセル化して伝送する方法のような、小さな問題に切り分けます。NoC インタコネクタは、単純ながら効果的な方法を持つ点で従来のインタコネクタと異なります。つまり、NoC アプローチでは、インタコネクタをシステムのモノリシック・コンポーネントとしてではなく、それぞれの層がインタコネクタの別々の機能を実装するプロトコル・スタックとして扱います。TCP オーバー IP オーバーイーサネットなどの従来のプロトコル・スタックの長所は、各層の情報がその下の層によってカプセル化されることです。Qsys NoC 実装の長所も、同様にプロトコル・スタックの各層の情報をカプセル化することからもたらされます。

NoC システムの基本的なトポロジーを図 1 「NoC システムの基本的なトポロジー」に示します。ネットワーク（マスタまたはスレーブ）内の各エンドポイント・インタフェースは、ネットワーク・インタフェース（NI）コンポーネントに接続されます。ネットワーク・インタフェースは、トランザクション層プロトコルを使用してトランザクションや応答を取り込み、対応する形式の packets としてネットワークに送信します。パケット・ネットワークは、パケットを対応するパケット・エンドポイントに送信し、パケットを受け取ったパケット・エンドポイントは、そのパケットを他のネットワーク・インタフェースに渡します。すると、他のネットワーク・インタフェースはパケットの終了処理を行い、トランザクション層プロトコルを使用して命令や応答をマスタまたはスレーブに送信します。

図 1. NoC システムの基本的なトポロジー



このシステムでは、プロセッサなどのコンポーネントがメモリ・コントローラなどのコンポーネントと通信します。これらの各コンポーネントはネットワーク・インタフェースのサービスを使用し、アルテラの Avalon® メモリマップド (Avalon MM) インタフェースや AXI (Advanced eXtensible Interface) などのトランザクション・インタフェース経由で相互に通信します。ネットワーク・インタフェースは、伝送サービスを提供する命令および応答ネットワークのサービスを使用して相互に通信し、トランザクション層サービスを提供します。トランスポート層の各コンポーネントは（命令および応答ネットワーク内の）トランスポート層プロトコルを認識しますが、各パケット内のトランザクションの詳細を認識する必要はありません。

NoC アーキテクチャの利点

プロトコル・スタックの層を分離するアプローチは、AHB (Advanced High Performance Bus) や CoreConnect などの従来のアプローチより以下の点で優れています。

- 層を独立して実装および最適化
- アプリケーションごとのカスタマイズを簡素化
- ネットワークのさまざまな部分に対して複数のトポロジーおよびオプションをサポート
- 機能開発の簡素化、インタフェースの相互運用性、および拡張性

層の実装および最適化

複雑なエンジニアリング上の課題に対しては、明確に定義された相互作用により、設計上の問題をより小さな問題に切り分けるアプローチが考えられます。NoC インタコネクットを使用すると、設計上の問題は、もはや「どうすれば複雑なシステムに柔軟なインタコネクットを適切に設計できるか」ではなく、「どうすればトランザクションを適切にパケットにマッピングできるか」や「どうすればパケットを適切に伝送できるか」という簡単な質問に変わります。また、層を分離することによって各層の実装を独立して最適化できるようにもなるため、他の層を再設計しなくても、ある層の性能を向上させることができます。例えば、トランザクション層においては何も変更せずに、さまざまなトランスポート層のトポロジーと実装を、同時に検討して実装することが可能です。

アプリケーションごとのカスタマイズの簡素化

命令と応答は、トランスポート層ではネットワークが伝達するパケットに過ぎず、ネットワーク層が行うことは、これらのパケットの伝送をサポートするのみでなければなりません。したがって、従来のインタコネクットと比較して、与えられたアプリケーションのインタコネクットのカスタマイズが簡素化されます。例えば、システムの一連のマスタと一連のスレーブの間にパイプライン化やクロック・クロッシングが必要であると設計者が判断すれば、パケットを安全に伝送できる限り、必要なコンポーネントを追加できます。クロック・クロッシングやパイプライン化を決定する上で、トランザクションの種類、応答の種類、バーストの種類の違いなどのトランザクション層が果たすべき機能を考慮する必要はありません。

複数のトポロジーおよびオプションの使用

NoC インタコネクットでは、ネットワークのさまざまな部分で各種最適化とトポロジーを行えます。例えば、プロセッサ、PCI Express® インタフェース、DMA コントローラ、メモリなどの高周波 / 高スループットの連続のコンポーネント、およびタイマ、UART、フラッシュ・メモリ・コントローラ、I²C インタフェースなどの低スループットの連続のペリフェラルを 1 つの設計に含めることができます。そのようなシステムはトランスポート層で分割できます。設計者は、高性能コンポーネントを広帯域の高周波パケット・ネットワークに配置する一方、ネットワーク間にパケット・ブリッジを配置するだけで、ペリフェラルを安価なメッシュ・ネットワークに配置できます。

機能開発の簡素化

インタコネクットは、新しい種類のトランザクションやバースト・モードなどの新機能をサポートするのに十分な汎用性を備えていなければなりません。インタコネクットを別々の層に分割すると、その機能をサポートする層を変更するだけで新機能を追加できます。例えば、ネットワーク・インタフェース・コンポーネントを変更するだけで、新しいバースト・モードをサポートできます。同様に、新しいネットワーク・トポロジーや伝送技術により性能を向上できる場合は、ネットワーク全体を再設計することなく元のネットワークを置き換えることが可能です。

インタフェースの相互運用性

AMBA® AXI、AHB、APB インタフェース、OCP インタフェース、Wishbone インタフェース、Avalon-MM インタフェースなど、さまざまなインタフェースを各種 IP (Intellectual Property) コアがサポートしています。インタフェースするトランザクションを、選択したパケット形式でカプセル化するネットワーク・インタフェースを実装するだけで、新しいインタフェースをサポート可能です。このアーキテクチャはブリッジ・コンポーネントを必要としないため、ロジックを節約してレイテンシを短縮できます。

拡張性

数百台のマスタとスレーブを持つシステムは珍しくなく、従来のインタコネクで必要な性能を実現するのは容易ではありません。数十台のマスタとスレーブに対応するように設計されているインタコネクを拡張して、今日のシステムが必要とする数百個のコンポーネントをサポートするには困難が伴います。NoC インタコネクでは、必要に応じて、ネットワーク全体のブリッジ、パイプライン・ステージ、およびクロック・クロッシング・ロジックを利用して、ネットワークをサブネットワークに比較的容易に分割することができます。したがって、マルチホップ・ネットワークが数千個のノードを容易にサポートでき、複数の FPGA にまたがる伝送ネットワークでさえも実現可能です。

Qsys における NoC システム設計

Qsys は、アルテラの Quartus II 開発ソフトウェアに搭載される強力なシステム統合ツールです。Qsys によって FPGA システム設計が簡素化され、設計者は、オン・チップ・インタコネクやネットワークの幅広い知識がなくても高性能システムを容易に構築できます。Qsys には広範な IP ライブラリが含まれているため、設計者は従来の手動での統合手法を使用するよりはるかに短い時間でシステム・オン・チップ (SoC) を構築して実装可能です。従来の設計手法を使用した場合は、システムのコンポーネントを接続するための HDL モジュールを記述しました。Qsys を使用することで、GUI やスクリプト化されたシステム記述によってシステム・コンポーネントをインスタンス化したりパラメータ化したりできます。その後、ボタンを押すだけで、Qsys がコンポーネントとインタコネクを生成します。Qsys で構築したシステムの例を図 2「Qsys で表示されるシステム・コンポーネントの例」に示します。

図 2. Qsys で表示されるシステム・コンポーネントの例

System Contents		Address Map	Clock Settings	Project Settings	System Inspector	HDL Example	Generation
Use	Connections	Name	Description		Export		
<input checked="" type="checkbox"/>		[-] nios2_qsys_0	Nios II Processor				
		data_master	Avalon Memory Mapped Master		Click to export		
		instruction_master	Avalon Memory Mapped Master		Click to export		
<input checked="" type="checkbox"/>		[-] flash_controller	Generic Tristate Controller				
		uas	Avalon Memory Mapped Slave		Click to export		
		tcm	Tristate Conduit Master		Click to export		
<input checked="" type="checkbox"/>		[-] SSRAM_controller	Generic Tristate Controller				
		uas	Avalon Memory Mapped Slave		Click to export		
		tcm	Tristate Conduit Master		Click to export		
<input checked="" type="checkbox"/>		[-] tristate_conduit_pin_sharer_0	Tristate Conduit Pin Sharer				
		tcm	Tristate Conduit Master		Click to export		
		tcs0	Tristate Conduit Slave		Click to export		
		tcs1	Tristate Conduit Slave		Click to export		
<input checked="" type="checkbox"/>		[-] tristate_conduit_bridge_0	Tristate Conduit Bridge				
		tcs	Tristate Conduit Slave		Click to export		
	out	Conduit		tristate_conduit_out			

Qsys では、システム設計者が GUI を使用して必要な IP コンポーネントをシステムに追加し、各コンポーネントをパラメータ化し、システム・コンポーネント間のインタフェース・レベルでの接続を規定します。Qsys は、接続されているインタフェース内の個々の信号を自動的に接続します。Qsys はシステム実装を RTL として生成し、クロック・ドメイン・クロッシングや、インタフェース幅の適応、バーストの適応などのシステム・インタコネクต์の問題に対応します。

Qsys は、トランザクション（読み込みおよび書き込み）インタフェース、ストリーミング（パケットまたは非パケット）インタフェース、割り込み、リセットなど、多くの種類の異なるインタフェースをサポートしています。Qsys トランザクション・インタコネクต์は、FPGA 用に特別に設計された NoC 実装に基づいており、FPGA リソースの使用量を最小限に抑えると同時に、高周波数およびスループットの要件を持つ高性能システムをサポートします。

FPGA 向けに最適化された Qsys NoC インタコネクต์

Qsys NoC インタコネクต์は、柔軟な最小限の実装、適応作業を減らすように設計されたパラメータ化可能なパケット形式、低レイテンシのインタコネクต์、命令用と応答用に分離されたネットワークなど、とりわけ FPGA を使用するシステムに最適な機能を備えています。

柔軟かつ最小限の実装

Qsys インタコネクต์は、マルチギガビットのデータパスや複雑なバーストを持つ大規模な高性能システムだけを目的としているのではなく、わずか数個のコンポーネントからなる小規模システムも対象としています。Qsys は、そのように広範なシステムをサポートするために、与えられたアプリケーションの性能要件を満たすのに必要な最小限のインタコネクต์しか実装しません。

Qsys はまず、システムを複数のインタコネクต์・ドメインに分割します。システム・アルゴリズムを必要とする接続がない場合、それらのインタフェースは別々のインタコネクต์・ドメイン内に配置しておくこととなります。例えば、1 台のマスタが 2 台のスレーブと接続している場合、それらのスレーブは同じインタコネクต์・ドメイン内に配置します。Qsys は、ドメインごとにすべてのマスタとスレーブの幅を考慮し、ドメイン内のインタフェースのクロック・レートに基づいて、システム内の最高のスループット接続を目指して、スループット全体をサポートする最小限のネットワーク・データ幅を設定します。

Qsys はまた、アプリケーションが必要とするインタコネクต์・コンポーネントのみを追加します。例えば、システム内の 1 台のマスタが 1 台のスレーブとしか接続していない場合は、アドレス・デコーダ・コンポーネントが省略されます。システム内の 1 台のスレーブが 1 台のマスタとしか接続していない場合は、アービタ・コンポーネントが省略されます。アプリケーションがある種類のバースト適応を必要としない場合は、そのバースト適応のサポートが省略されます。

パラメータ化可能なパケット形式による適応作業の削減

Qsys は、インタコネクต์・リソースの使用量を最小限に抑えるだけでなく、ロジックの使用と適応を最小限に抑えるパケット形式を判断します。例えば、パケット内のアドレス・フィールドとバーストカウント・フィールドは、システムをサポートするのに必要な最小の幅です。パケット内のアドレス・フィールドやその他のフィールドは、適応コンポーネントがパケットの状態を維持する必要がなく、場合によってはアダプタをまとめて省略することもできるように、パケットのすべてのサイクルで有用かつ正確な値に変更されます。

低レイテンシのインタコネク

設計者は、パケット・ベースのアプローチでは、サイクルごとにトランザクション全体の一部しか伝達されないと考えて、パケットをシリアル化と関連付けるのが普通です。多くの NoC 実装ではこのアプローチを使用します。そのような NoC 実装のネットワーク・レイテンシは約 12 ~ 15 クロック・サイクルであるため、例えば、マイクロコントローラとそのローカル・メモリ間のインタコネクなどには適していません。Qsys インタコネクのどのコンポーネントも、データパスを組み合わせることでレイテンシの問題を解決しようとしています。パケット形式の幅は、1 つのクロック・サイクルに完全なトランザクションを含むことができる広さを持っているため、インタコネク全体でレイテンシが 0 サイクルの書き込みとラウンド・トリップ・レイテンシが 1 サイクルの読み込みをサポートできます。このような広い幅の接続は、今日の FPGA であれば十分にサポート可能です。システム設計者は、パイプライン化オプションを変更することによって、レイテンシの増加と引き換えに周波数を高めることもできます。

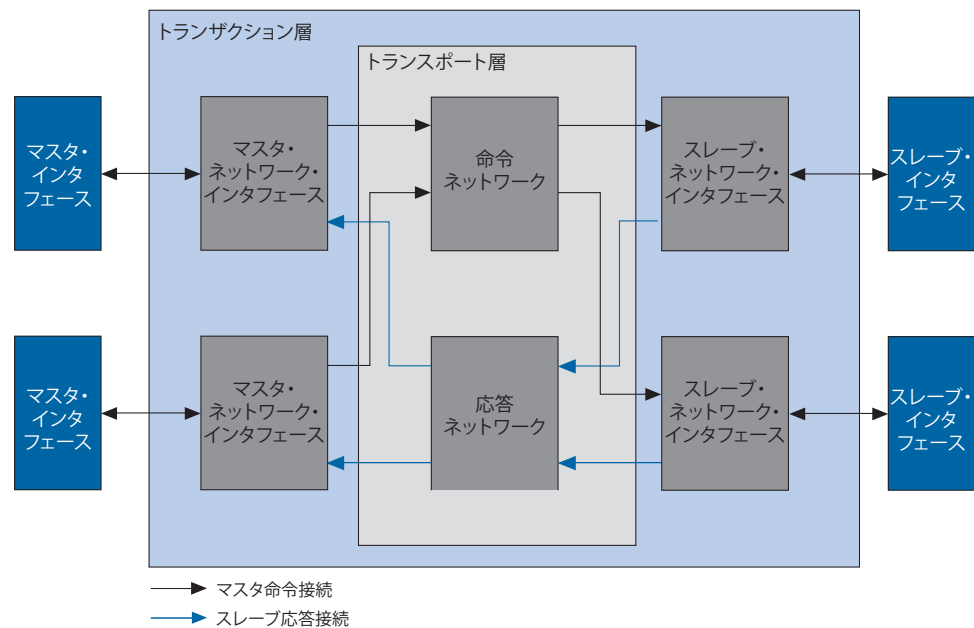
命令ネットワークと応答ネットワークの分離

Qsys は、トランザクション・ドメインごとに、命令トラフィックと応答トラフィックの両方をサポートする 1 つのネットワークではなく、それぞれに 1 つずつ、2 つの独立したパケット・ネットワークをインスタンス化します。そのため、ネットワーク・ノード間のリンクのように命令トラフィックと応答トラフィックがリソースを奪い合うことがなくなって、同時並行性が高まります。Qsys では、この 2 つのネットワークを独立して最適化することもできるため、2 つのネットワーク内のネットワーク・トポロジーとパケット形式を別々のものにすることも可能です。

最適化された命令および応答ネットワーク

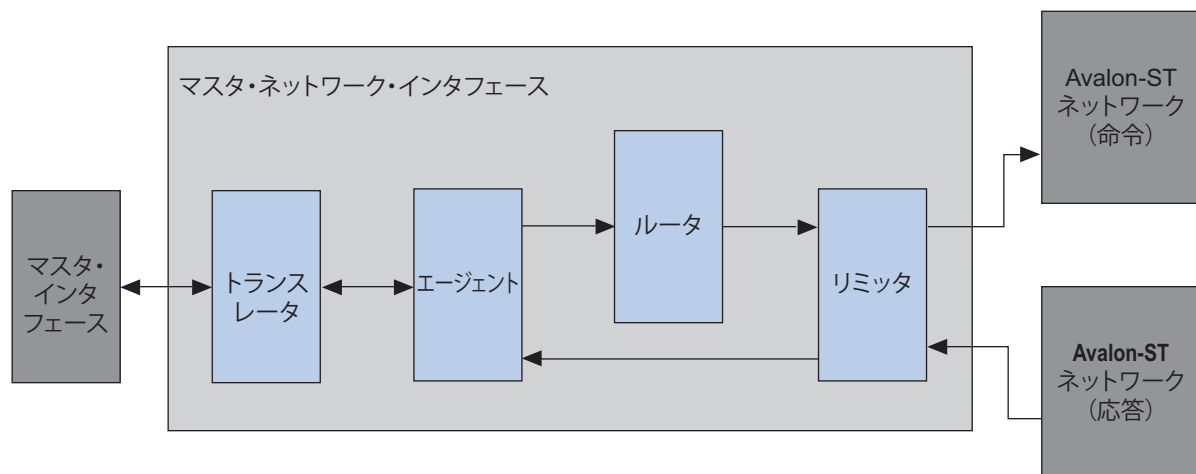
マスタが対象スレーブに発行するリード命令と、スレーブがマスタに返す応答を説明する以下の手順では、[図 3](#)「Qsys NoC インタコネクットのトポロジ」に示す NoC インタコネクットの命令および応答ネットワークの概要を示します。

図 3. Qsys NoC インタコネクットのトポロジ



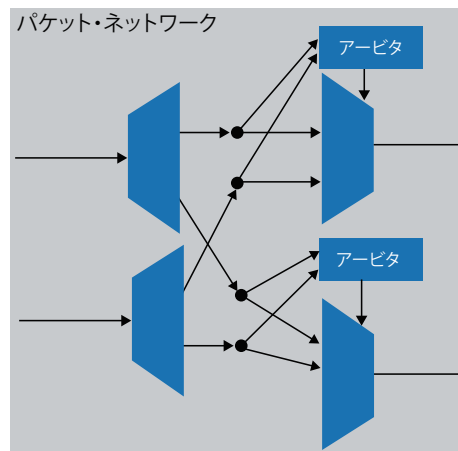
1. [図 4](#)「マスタ・ネットワーク・インタフェース」に示すように、マスタが命令を発行したときにトランザクションを最初に受け取るインタコネクット・コンポーネントはトランスレータです。トランスレータは、アクティブ High / アクティブ Low 信号オプションやオプションの読み込みパイプライン化など、トランザクション・プロトコル仕様のばらつきを多くを処理します。

図 4. マスタ・ネットワーク・インタフェース



2. 命令を次に受け取るブロックは、エージェントです。エージェントはトランザクションを命令パケットにカプセル化し、トランスポート層を使用してそのパケットを命令ネットワークに送信します。また、応答ネットワークからの応答パケットを受け付けて、マスタに転送します。
3. ルータは、パケット形式内のアドレス・フィールド、パケット送信先のスレーブ ID、および次のホップのルーティング情報を特定します。
4. リミッタは、別のマスタまで未処理トランザクションを監視し、アウト・オブ・オーダー・リードやリード応答の競合が発生しないよう、命令の発行を制御します。
5. 次に、コンポーネントがパケット・ネットワークに注入されます。Qsys NoC ネットワークは最大限の同時並行性を保証しており、図 5「最大限の同時並行性を持つパケット・ネットワーク」に示すように、2 台のマスタが同じスレーブにアクセスしようとしめない限り、すべてのマスタとスレーブがどのクロック・サイクルでも通信できます。

図 5. 最大限の同時並行性を持つパケット・ネットワーク

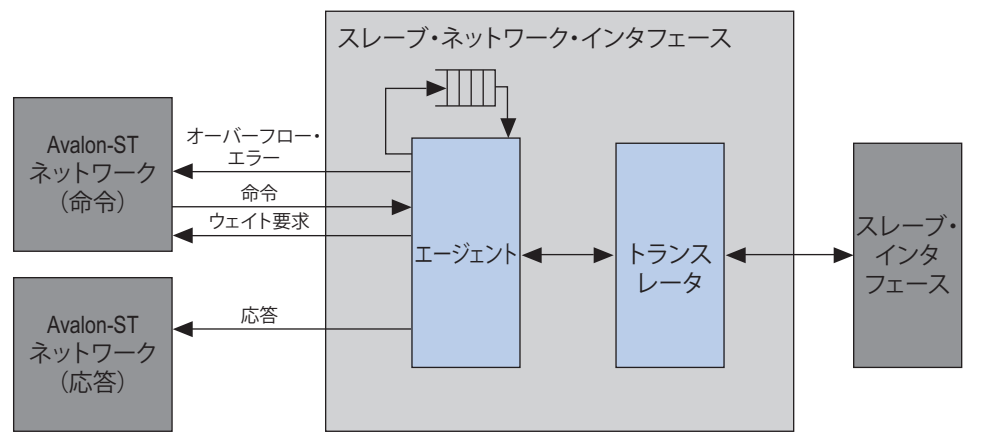


NoC アーキテクチャでは、パケット・ネットワークを、互換性のある他のどのネットワーク実装にも置き換えることができます。

6. トランスポート層ネットワーク内でパケットが初めて到着するコンポーネントは、デマルチプレクサです。デマルチプレクサは、次のスレーブにパケットを送信します。
7. パケットはスプリッタ・コンポーネント（図内にある黒い点）に到着します。スプリッタ・コンポーネントは、基本的にアービタの入力とマルチプレクサへの入力にパケットをコピーします。
8. Qsys がデフォルトで提供する、重み付けされたラウンド・ロビン・アービトレーションではなく、アプリケーション固有のアービトレーションを必要とするシステム設計者は、Qsys アービタを独自のアービタに置き換えることができます。これをサポートするために、Qsys アービタのフットプリントはパケット全体を受け付け、代替のアービタ実装が詳細なトランザクション情報を使用して、アービトレーション（データ依存アービトレーションなど）を決定できるようにします。

9. アービタが決定した結果がマルチプレクサに送信され、マルチプレクサは図 6「スレーブ・ネットワーク・インタフェース」に示すように、選択した packets をスレーブ・ネットワーク・インタフェースに転送します。

図 6. スレーブ・ネットワーク・インタフェース



10. スレーブ・ネットワーク・インタフェース内で、パケットはスレーブ・エージェント・コンポーネントに入ります。スレーブ・エージェント・コンポーネントは、パケットの終了処理を行ってパケットに含まれるトランザクションをスレーブ・トランスレータに転送します。同時に、スレーブ・エージェント・コンポーネントは、リードやノンポストド・ライトなどの応答を要求するトランザクションのために、トランザクション情報をスレーブ・エージェントの FIFO バッファにプッシュします。スレーブ・トランスレータはマスタ・トランスレータと同じ役割を担い、インタフェース仕様に考えられるあらゆるばらつきを考慮します。スレーブがビジーでそれ以上のトランザクションを受け付けることができない場合は、エージェントの入り口で命令にバックプレッシャーがかけられます。
11. スレーブがリード・トランザクションに応答すると、トランスレータがスレーブ・エージェントに応答を転送します。スレーブ・エージェントは、送信元のマスタ ID などのトランザクション情報をスレーブ・エージェントの FIFO バッファから取り出し、トランザクション応答にマージして応答パケットを作成します。読み込みデータの FIFO は、応答ネットワークが一時的に応答を受け付けられない場合に応答を格納するためのものです。
12. 次に、スレーブ・ルータがパケットを検査してマスタ ID を判断し、ローカル・ルーティング情報を割り当てます。
13. 応答は命令と同じですが、方向が逆になります。応答パケットはデマルチプレクサを経由してアービタに到着し、選択されると、マルチプレクサ経由でリミッタに戻るよう転送されます。すると、リミッタは応答を受け取ったことを記録し、応答をトランザクション応答の形でマスタ・エージェントに、そして最終的にマスタに送り返します。

Qsys は、上に挙げたコンポーネント以外に、必要に応じてバースト・アダプタや幅アダプタを追加します。これらはいずれも、一部のフィールド内のデータでパケットを検査して、対応する適応手順を決定するためのパケット・コンポーネントです。Qsys は、パイプライン化ステージを追加してタイミングを合わせるのに役立つこともでき、マスタとスレーブが別のクロック・ドメインにある場合は、ハンドシェイキングまたはデュアルクロック FIFO コンポーネントを自動的に追加します。

性能の例

以下の例では、マスタ 16 台 / スレーブ 16 台のシステムと、マスタ 4 台 / スレーブ 16 台のバースト適応および幅適応システムの2つの異なるシステムの性能を比較しています。この比較は、Qsys NoC インタコネクットの周波数、レイテンシ、およびリソース使用量が従来のインタコネクット実装と比べてどのようになるかを示しています。これらの例では、すべてのシステムをアルテラの Stratix®IV デバイス上に実装しており、C2 スピード・グレードを使用しています。Qsys NoC インタコネクット・システムの性能を、アルテラの前世代のシステム統合ツール SOPC Builder ツールで同じシステム向けに生成した、従来の Avalon-MM インタコネクットと比較しています。

マスタ 16 台 / スレーブ 16 台のシステム

マスタ 16 台 / スレーブ 16 台のシステムが、合計 256 個の接続と完全に接続されています。この単純なマスタおよびスレーブ IP コンポーネントは、インタコネクットの特性をテストするためだけのものです。つまり、このシステムは完全に均一なシステムを表すものであって、代表的な組み込みシステムではありません。表 1「マスタ 16 台 / スレーブ 16 台のシステム：性能の結果（従来のインタコネクットに対するパーセント）」、図 7「マスタ 16 台 / スレーブ 16 台のシステム：従来のインタコネクットと比較した NoC の周波数 (MHz)」、および図 8「マスタ 16 台 / スレーブ 16 台のシステム：従来のインタコネクットと比較した NoC のリソース使用量 (ALUT)」に、従来のインタコネクットの周波数とリソース使用量の結果および NoC 実装のさまざまなレイテンシ・オプションを示します。

表 1. マスタ 16 台 / スレーブ 16 台のシステム：性能の結果（従来のインタコネクットに対するパーセント）

インタコネクットの実装	f_{MAX} (MHz)	リソース使用量 (ALM)
従来のインタコネクット	131	12766
Qsys NoC、組み合わせ論理	161 (+23%)	13999 (+10%)
Qsys NoC、1 サイクルのネットワーク・レイテンシ	225 (+71%)	11260 (-12%)
Qsys NoC、2 サイクルのネットワーク・レイテンシ	243 (+85%)	12761 (+0%)
Qsys NoC、3 サイクルのネットワーク・レイテンシ	254 (+93%)	14206 (+11%)
Qsys NoC、4 サイクルのネットワーク・レイテンシ	314 (+138%)	26782 (+110%)

図 7. マスタ 16 台 / スレーブ 16 台のシステム：従来のインタコネクと比較した NoC の周波数 (MHz)

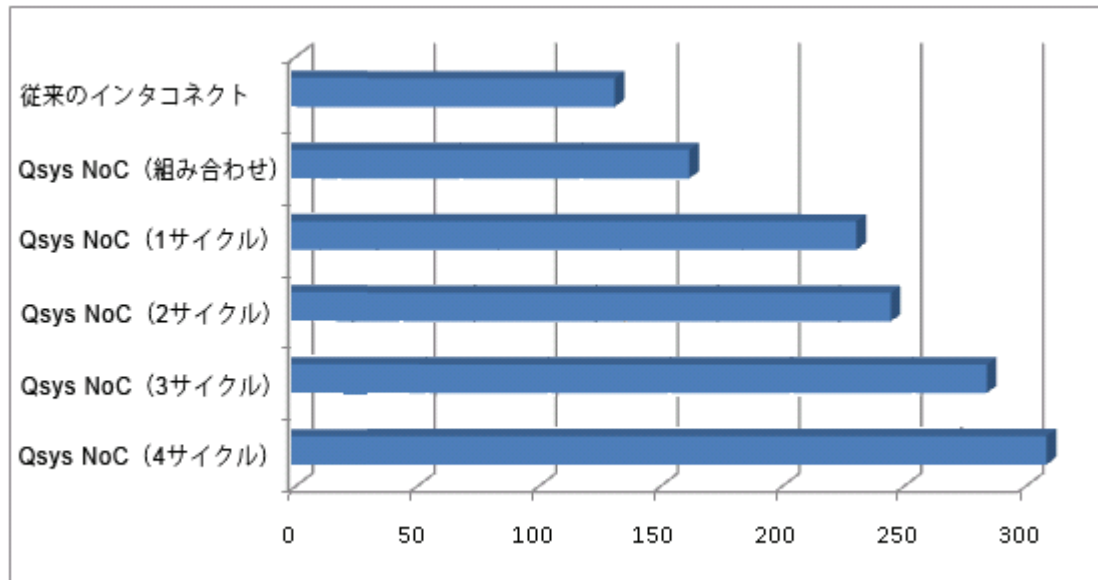
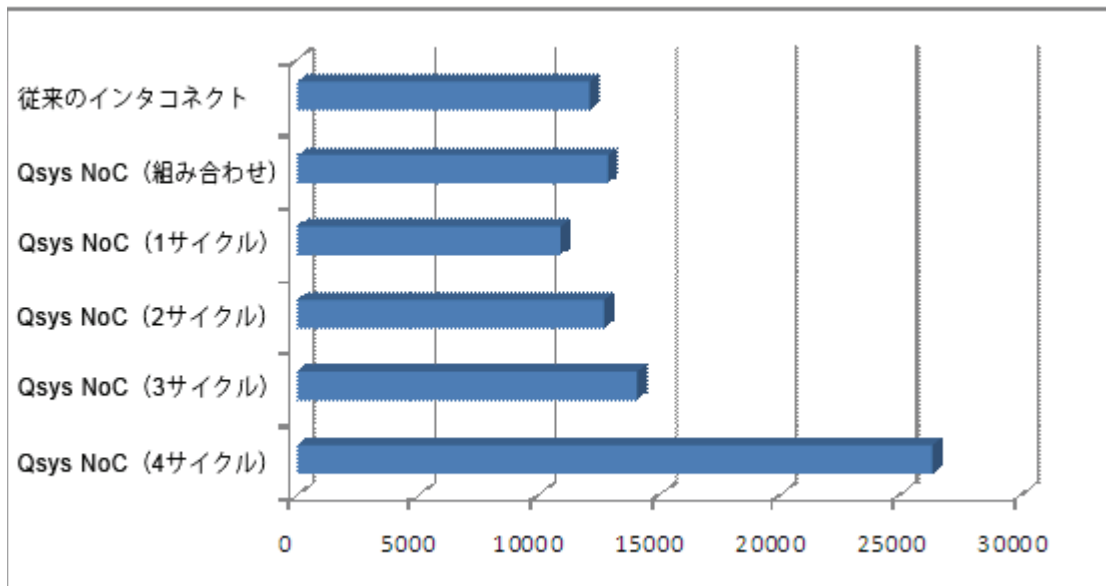


図 8. マスタ 16 台 / スレーブ 16 台のシステム：従来のインタコネクと比較した NoC のリソース使用量 (ALUT)



マスタ 4 台 / スレーブ 16 台のバースト適応および幅適応システム

マスタ 4 台 / スレーブ 16 台のバースト適応および幅適応システムは、幅が異なるマスタとスレーブおよびバースト・サポートの違いを含み、インタコネクタでのバースト適応を必要とする、代表的な不均一な特性を含んでいます。表 2「マスタ 4 台 / スレーブ 16 台のシステム：性能の結果（従来のインタコネクタに対するパーセント）」、図 9「マスタ 4 台 / スレーブ 16 台のシステム：従来のインタコネクタと比較した周波数 (MHz)」および図 10「マスタ 4 台 / スレーブ 16 台のシステム：従来のインタコネクタと比較したリソース使用量 (ALUT)」に、従来のインタコネクタの周波数とリソース使用量の結果および NoC 実装のさまざまなレイテンシ・オプションを示します。

表 2. マスタ 4 台 / スレーブ 16 台のシステム：性能の結果（従来のインタコネクタに対するパーセント）

インタコネクタの実装	f _{MAX} (MHz)	リソース使用量 (ALM)
従来のインタコネクタ	123	11658
Qsys NoC、組み合わせ論理	125 (+2%)	9655 (-17%)
Qsys NoC、1 サイクルのネットワーク・レイテンシ	150 (+22%)	9423 (-19%)
Qsys NoC、2 サイクルのネットワーク・レイテンシ	164 (+33%)	9847 (-16%)
Qsys NoC、3 サイクルのネットワーク・レイテンシ	154 (+25%)	13156 (+13%)
Qsys NoC、4 サイクルのネットワーク・レイテンシ	171 (+39%)	16925 (+45%)

図 9. マスタ 4 台 / スレーブ 16 台のシステム：従来のインタコネクタと比較した周波数 (MHz)

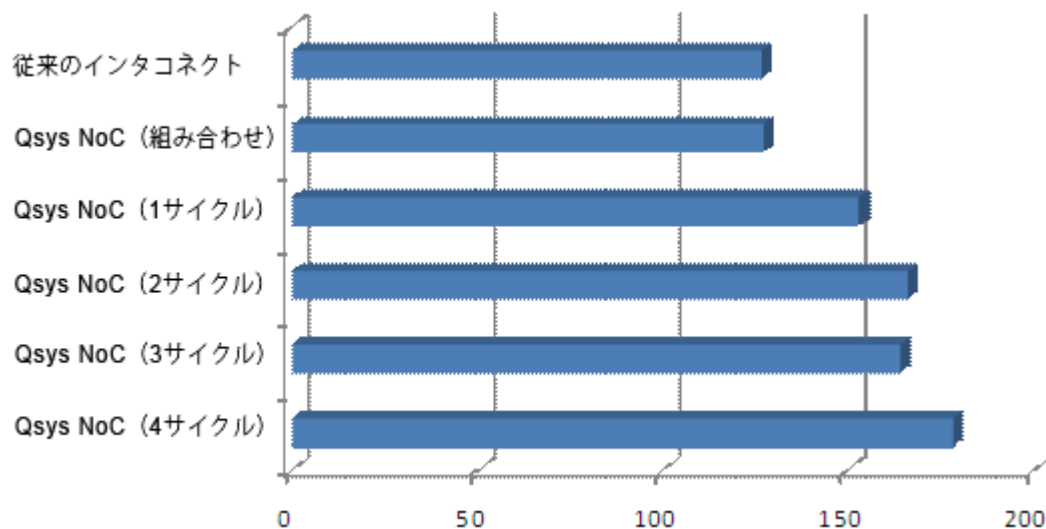
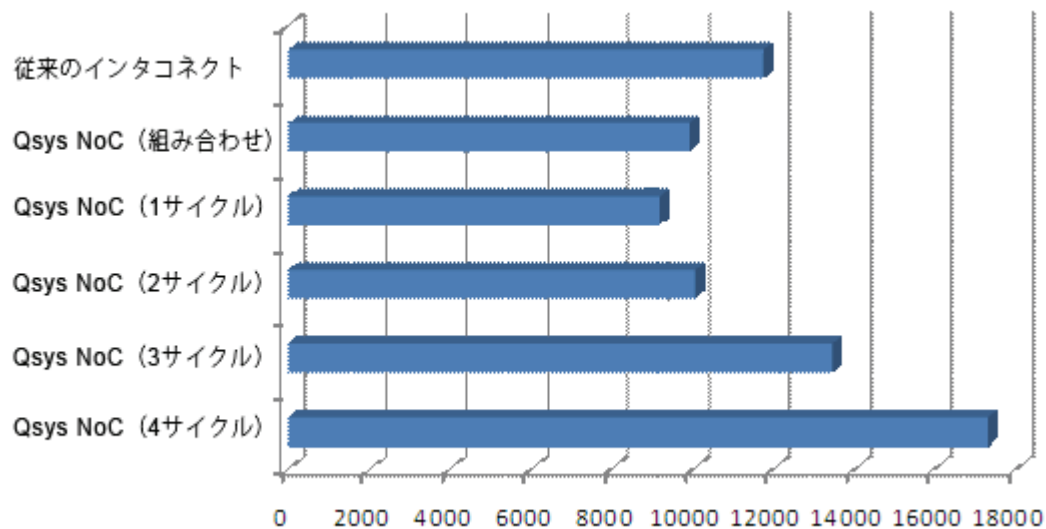


図 10. マスタ 4 台 / スレーブ 16 台のシステム：従来のインタコネクと比較したリソース使用量 (ALUT)



まとめ

NoC インタコネク・アーキテクチャは、トランザクションおよびトランスポート・プロトコル層の独立した設計や最適化の実現など、従来の非 NoC インタコネクにはない多くの大きな利点をもたらします。Qsys システム統合ツールは、アプリケーションの要件に基づいて、FPGA 向けに最適化された極めて柔軟な NoC 実装を生成します。Qsys NoC インタコネクは、同じレイテンシおよびリソース特性での動作周波数が高くなっており、従来のインタコネクと比較して f_{MAX} が最大 2 倍に向上します。

詳細情報について

- アルテラ Web サイトの Qsys ソフトウェア・サポート・ページ：
<http://www.altera.co.jp/support/software/system/qsys/sof-qsys-index.html>
- 『Quartus II ハンドブック Volume 1』の『System Design with Qsys』の章
http://www.altera.co.jp/literature/hb/qts/qsys_section.pdf
- AN632: SOPC Builder to Qsys Migration Guidelines
<http://www.altera.co.jp/literature/an/an632.pdf>
- 『Qsys System Design Tutorial』
http://www.altera.co.jp/literature/tt/tt_qsys_intro.pdf

謝辞

- Kent Orthner, Sr. Manager, Software & IP, Altera Corporation

文書改訂履歴

表 3 に、本書の改訂履歴を示します。

表 3. 文書改訂履歴

日付	バージョン	変更内容
2011 年 4 月	1.1	性能とリソース使用法の情報を更新
2011 年 1 月	1.0	初版