

Qsys Interconnect による オンチップ／オフチップ FPGA 通信網に関する研究

三浦 康之^{†1} 深瀬 尚久^{†1} 半田 寛信^{†2} 古市 晃久^{†1}

概要: 近年, 計算機の性能向上に FPGA を利用する手法が注目されており, 実装形態の一つである FPGA クラスタについても構成方法などが盛んに研究され, 実装されている. FPGA クラスタにおけるボード間の接続方法については, 決まった仕様が存在しておらず, 以前は各々の設計者が個別に作成していた. 近年, FPGA クラスタに使用する汎用的な結合網の研究がおこなわれている. 我々は Intel® FPGA の Qsys Interconnect の仕様に沿った制作することで高い汎用性と拡張性を持つ結合網の開発を目指している. 本稿では, ルータ回路の開発およびパケットの送信と受信に関する現在の検討状況について述べる.

キーワード: FPGA, Qsys Interconnect, Avalon-streaming, 相互結合網

The Research on On-chip/Inter-Chip FPGA Interconnection by Qsys Interconnect

Yasuyuki Miura^{†1} Naohisa Fukase^{†1} Hironobu Handa^{†2} Akihisa Furuichi^{†1}

Abstract: In recent years, methods using FPGAs to improve the performance of computers have attracted attention, and configuration methods such as FPGA clusters, which is one of implementation forms, have been extensively researched and implemented. The router circuit used for the FPGA cluster does not have the fixed production method so far, it was also the cause of the increase in research and development effort. In view of these circumstances in recent years, general purpose connection networks used for FPGA clusters are being studied. We aim to develop a connection network with high versatility and extensibility by making it according to the specification of Qsys Interconnect of Intel® FPGA. In this paper, we mention the development of router circuit, and packet transmission and reception, as the progress. At last, we discuss the vision of the future.

Keywords: FPGA, Qsys Interconnect, Avalon-streaming, Interconnection Network

1. はじめに

近年, 様々な分野でより高性能な計算機が求められている. これまで, このような計算機の実現にはプロセッサのコアを増加させる方法(マルチコア, メニーコアプロセッサ)などが取られてきた. しかし, これらの方法はコアの稼働率や消費電力の問題からこれ以上の性能向上が困難になりつつある. また, GPU を活用する方法なども取られている[30]-[34]が, こちらは消費電力と性能の両立が課題となっているほか, ホストコンピュータが必須となっているため必ずしも低コストとは言えないという問題もある. 我々は, 計算機の性能向上手段として, FPGA(Field-Programmable Gate Arrays)を用いた高性能計算[34]-[37]に着目しており, FPGA ボードを複数搭載した FPGA クラスタに着目している.

FPGA は, HDL(Hardware Description Language, ハードウェア記述言語)を用いて, 目的に応じて回路を構成することが可能なデバイスである. 対象問題に特化した演算ユニット, データパス, メモリシステムを実装可能で, かつ高い並列

性を実現できる. また, 比較的低い動作周波数で動作可能なため, 電力あたりの処理能力も高くできる可能性がある.

FPGA クラスタは単独の FPGA ボードを複数接続して使用する実装方法である. FPGA クラスタにおけるボード間の接続方法については, 決まった仕様が存在しておらず, 以前は各々の設計者が個別に作成していた. 近年, FPGA クラスタにおけるボード間の接続方法についての研究が盛んに行われており, いくつもの実装例が報告されている[1][2]. このような実装法が確立すれば, 多数の FPGA を接続したシステムを, 比較的簡単に開発することが可能になることが期待される.

本研究計画は, System on Chip (SoC)における既存の通信仕様である Qsys Interconnect[24]をベースとし, FPGA チップ内およびチップ間の統一的な通信仕様を提案するのみならず, クラスタ各所に点在するメモリに対する単一メモリ空間による統一的なメモリ・アクセスや, 実アプリケーションによる実証実験なども含めた計画としている[9]-[11].

本稿では, 本研究が目標とするところを整理し, 現在までの進捗を述べる. 2 節において, 本計画の狙いと実用例

^{†1} 湘南工科大学大学院電気情報工学専攻
Graduate School of Electronics and Information Technology, Shonan Institute
of Technology

^{†2} 株式会社トラストテック
Trust Tech Inc.

について議論し、3節以降において、現在の進捗を述べる。具体的には、3節がルータの開発、4節が入出力部の仕様について述べている。最後に、5節においてまとめとして、これまでの研究の進捗状況と、今後の予定について述べている。

2. システムの概要

2.1 FPGA クラスタ

本計画が目標とするシステムの概要を図1に示す。本計画では図1のように、複数のFPGAボードがホストコンピュータに搭載され、ボードに取り付けられた通信インターフェースにより接続されている構造のFPGAクラスタを想定している。

本システムにおいてFPGA内に実装する回路の概略を図2に示す。図2にあるように本システムは、大きく分けて「ユーザーロジック」、「モジュール」、「ルータ」の3つから構成される。「ユーザーロジック」は、実際に行われる処理(画像処理など)を担当する回路である。そして、「モジュール」は「ユーザーロジック」とデータの移動、格納などを行う回路の集合である。本システムでは、この「モジュール」を一つの単位として扱い、一つのFPGAに一つまたは複数搭載する予定である。なお、個々のFPGAに搭載されるモジュールの数は、FPGA外部と接続されるメモリ等の数に依存することを想定している。そして、この「モジュール」間を接続し、通信の制御をおこなう回路が「ルータ」回路である。

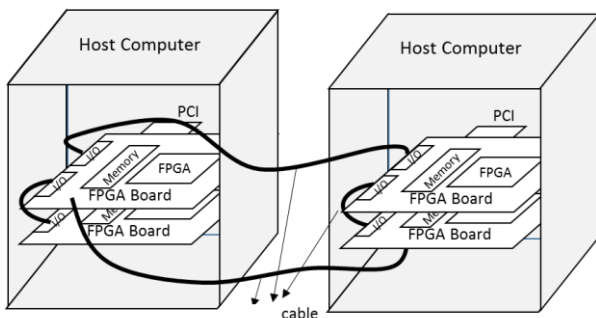


図1 FPGAクラスタのイメージ
 Figure 1 FPGA Cluster

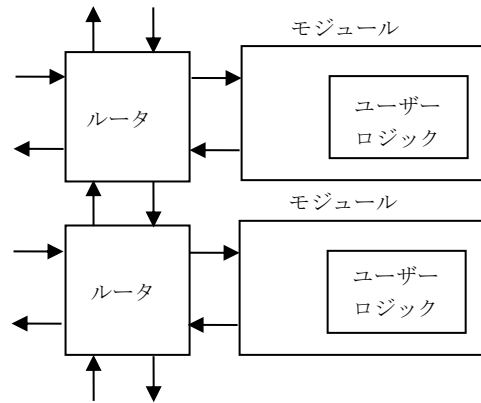


図2 実装回路の概略
 Figure 2 The Overview of the Circuit

2.2 開発環境

本研究において使用される環境は、以下の通りとなる。なお、本稿における実験は、テキスト等が充実して使用実績の多いDE0の結果を示しているが、より大規模で本番環境に近い環境を持つDE4ボードも合わせて使用している。

- **QuartusII Ver.13.1**

Altera社のFPGAを開発する統合開発ツールである。回路設計、論理合成、配置配線、タイミング検証、デバイスの製品構成の作成、プログラミング、デバッグ機能など様々な機能を持つ。

- **Qsys**

FPGAに実装する回路を、HDL記述を用いずに自動生成することができるツールである。FPGAで使用できるソフトコアプロセッサであるNios IIもこのツールで生成することができる。

- **Nios II SBT for Eclipse**

プログラムのエディット、ビルド、デバッグを含むすべてのソフトウェア開発作業を行うことができる。また、Niosを持つシステムにおいてプログラムを実行させることができる。

2.3 システム統合ツール Qsys の概要

FPGAボードを用いた開発のためのシステム統合ツールとして、Altera社のQsysが知られている。Qsysは、グラフィカルなユーザーインターフェイスを介して、CPUやRAMを含んだシステムをFPGA上に容易に構築できるツールである。Qsysにおけるコンポーネント間結合インターフェースとしては、Avalon-MM (Avalon-Memory Mapped interface) および Avalon-ST (Avalon Streaming interface)の二種類が存在する[24]。

- **Avalon- Memory Mapped Interface**

メモリ・アクセスと同様にアドレス線とデータ・バス

を基本構成とし、双方向にデータを扱う。複数のペリフェラルを接続したシステム・バスとしてマネージメント用途にも利用される。コンポーネントは、コントロール用アクセス・ポイントとして、Avalon-MM インターフェースを使用することもある。

・ Avalon- Streaming Interface

アドレスを持たない構成として、データ・ソースからデータ・シンクへの単方向のストリーミング・データを低レイテンシ(待機状態が少ない)で転送させる用途に利用される。ストリーミング・データを処理するコンポーネント(構成要素)は、一般に高い処理能力を持ったデータバスに Avalon-ST を使用する。

2.4 実用例

本研究では、複数の FPGA ボードを相互結合ネットワークで接続するとともに、FPGA 内部のモジュールを同一構造の相互結合ネットワークで結ぶことを想定している。以下に、本研究で想定している実装形態の実例をいくつか示す。

- 1) おもに科学技術シミュレーションを実行するためのシステムの一部として用いられる[3][4][8][36][37]。この場合は、一つの計算モジュールが問題の一部を解き、複数の FPGA に散在する複数の計算モジュールで並列計算を行う。一つの FPGA には、一つまたは少数の計算モジュールが存在することを想定する。本用途に関しては、文献[3]を参考に設計している。
- 2) 多数のモジュールが FPGA 上に搭載されるモデルで、NoC のシミュレーションなどに使用する。この場合、一枚の FPGA に多数の計算モジュールを接続し、主にオンボードによる様々なシミュレーションを行うことにより、相互結合網の研究[17]-[19]やルーティングアルゴリズムの研究[20]-[22]に役立つ。このようなシミュレータは、他にも開発された事例[5]がある。
- 3) 最適化アルゴリズム PSO の FPGA への実装事例が報告されている[6][7]。これを拡張することにより、FPGA クラスタ上に搭載することで、大規模計算が可能になると考えられる。
- 4) 画像処理への応用[29]を想定した、関連技術の開発を行っている。FPGA ボードに、カメラモジュールやディスプレイを接続した実験装置を構築し、防犯カメラ解析の支援を想定した実験を行うことを考えている。簡単のため、さしあたり現時点では、主に 1)の用途を想定して、一枚の FPGA ボードに一個の計算モジュールを搭載する想定で開発を進めている。

図3にモジュールの構造と実装例を示す。図3のように、ユーザーロジックやメモリコントローラ、NiosII プロセッサにより一つのモジュールが構成されている。この構成は関連研究[3]を参考にしており、入出力ポートにインターフ

ェースとして FIFO メモリを加えている。モジュール内の接続は、バスと直接つながっているコンポーネント同士は Avalon-MM、それ以外の 1 対 1 でコンポーネントがつながっている部分については Avalon-ST で接続されている。モジュール内の NiosII は、システムに与えられた処理自体でなく、おもに外部モジュールとの通信やデータの移動、メモリ・アクセスなどを制御に使用される。

Nios II の動作としては以下のようなものを想定している。

- ・ Avalon MM バスを通して NiosII が直接データを移動する。
- ・ NiosII が DMA コントローラを制御して、ブロック単位によるデータ移動を行う。
- ・ 出力ポートや入力ポートを通したデータの入出力を制御する。

これらの処理は Nios II SBT において、C 言語による記述で定義する予定である。

システム自体に与えられた演算処理等はユーザーロジックにおいてユーザーが作成したロジックに基づいて実行される。ユーザーロジックは、入力としてストリームを受け取り、出力としてストリームを返す。ユーザーロジックは、HDL による記述を行う他に、C 言語などの高水準言語で記述して、変換ツール等[5]を用いて HDL に翻訳を行うことも可能である。

図3の実装例は、前述したモジュールをひとつの FPGA 内に 1 つ配置した場合である。この例では、モジュール内のメモリコントローラは、ボード上の外部メモリと接続される。モジュールの入出力ポートである FIFO は、同様の FPGA 内に構築されたルータ回路と接続される。そして、このルータ回路をボード上に設置されている通信インターフェースと接続し、これを通して別のボード、モジュールと通信する。

図3のモジュールが、一つまたは複数個、FPGA 上に搭載される。ここでは簡単のため、図4のように FPGA 上に一つのモジュールが搭載されるモデルを想定して、開発を進めている。

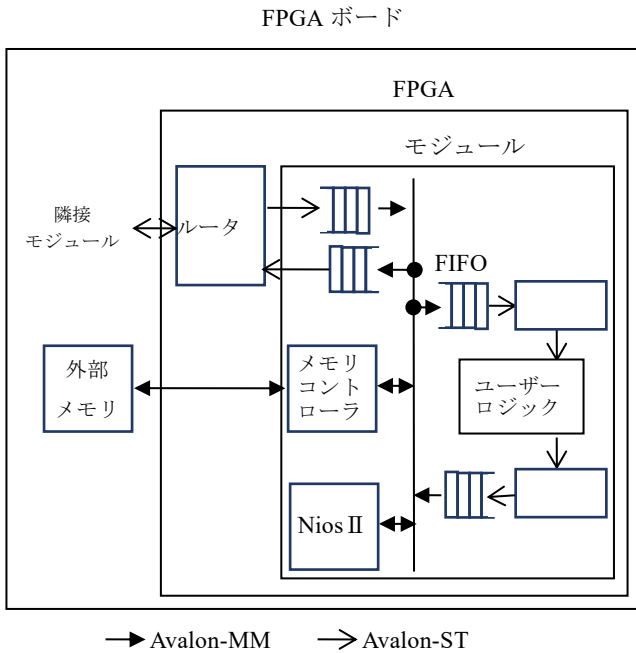


図 3 モジュールの構成
 Figure 3 Module Structure

また、これらの技術開発に合わせて、通信回路の高性能化に関する研究をあわせて進めてゆく。これまで我々は、ルータの設計に関するいくつかの研究実績を有している[10][14]-[16][23]ので、それらの技術の実用面の課題の検証も併せて行う予定としている。

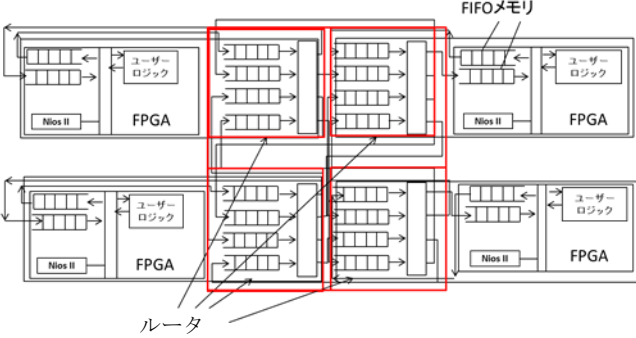


図 4 モジュール間の接続
 Figure 4 Module Interconnection

3. ルータ回路の開発

3.1 仕様

開発中のルータ回路を図 5 に示す。このルータ回路は、入力側に FIFO が取り付けられており、入力されたパケットは、一度ここに格納される。その後、パケット内のパケットヘッダ(通信制御用データ)を元に、制御回路が経路計算を行い、クロスバスイッチを設定する。最後にパケットは、クロスバスイッチを通過し、選択された出力に出力さ

れる。本稿では、このような機能を持ったルータ回路を可能な限り Qsys 内のコンポーネント[25]を使用して構成する。今回開発を終えたルータには、一般的な 3 サイクルルータ[12]から、仮想チャネル[13]の機能を除いたものとなっている。今後は、今回実装で含まれていない仮想チャネルの機能をいかにして実装するかが課題となっている。

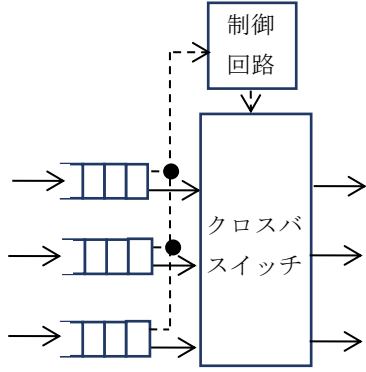


図 5 開発中のルータ回路
 Figure 5 Developing Router Circuit

● パケットの構造

パケットの構造を図 6 に示す[9]。パケットは、先頭フリット、中間フリット、最後尾フリットの 3 種類で構成されており、それぞれ Avalon-ST に用意された信号, sop (startofpacket), eop (endofpacket)を用いて区別する。これらはそれぞれ、Avalon-ST において通信データの始まりと終わりを示す信号である。Avalon-ST を使用するコンポーネントは、この信号によって、受信準備や受信完了の処理を行う。設計するルータではこれを利用して、リンクの使用、解放のための処理を行う。

図 6 にあるように、先頭フリットは、データ部に通信の制御に使用される情報であるパケットヘッダを含み sop を 1 に設定する。中間フリットは、データ部に送信データを持ち sop, eop 共に 0 に設定する。最後の最後尾フリットは、データ部には送信データを持ち、eop が 1 に設定される。

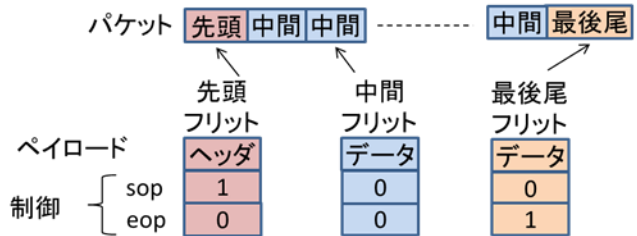


図 6 パケットの構造
 Figure 6 Packet Structure

● 入出力と FIFO

Qsys の規格に合わせるため、入出力には Avalon-ST の

「On-Chip FIFO Memory」 [26]をそのまま使用する。

● クロスバスイッチ

Qsys の標準コンポーネントには、クロスバスイッチに相当するものがないので、今回開発したルータでは、マルチプレクサとでマルチプレクサを組み合わせる同様の機能を実現する。マルチプレクサは、Qsys の標準コンポーネントに用意されている「Avalon-ST Multiplexer」 [27]を若干改良して使用している。Avalon-ST Multiplexer は調停機能を内蔵しているため、そのまま使用する予定であったが、特定のポート以外から入力したデータが送信されないという問題が起きたため、修正を行い、正常に動作することを確認した。

クロスバスイッチを再現するに当たり、Avalon-ST Multiplexer のみでは、以下の2つの問題が存在した。

- ・ 入力 FIFO から複数のマルチプレクサに同時に接続することができない
- ・ 制御回路からの制御信号による入力選択ができない

そこで、Qsys の標準コンポーネントに用意されている「Avalon-ST Demultiplexer」 [27]というデマルチプレクサを使用した接続を行っている。図7にマルチプレクサとデマルチプレクサを加えたルータを示す。デマルチプレクサは、Avalon-ST のオプション信号である channel 信号によって出力を指定できる。そのため、制御信号を channel 信号に割り当てることができれば任意の出力を選択することが可能となる。

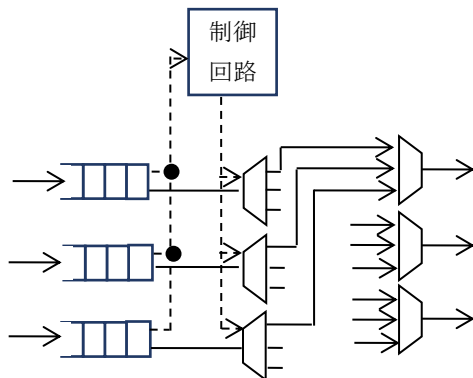


図7 デマルチプレクサを取り付けたルータ
 Figure 7 Router Circuit with De-Multiplexer

● 制御回路

制御回路は、FIFO の出力である Avalon-ST からパケットヘッダのみを取り出し、デマルチプレクサに入る Avalon-ST の channel 信号のみを書き変える必要がある。Qsys のシステム内において Avalon-ST 内の特定の情報のみを読み書きする方法は存在しないため、HDL(ハードウェア記述言語)によって自作する必要があるが生じた。また、制御回路内に経路計算回路を含む構成とした場合、ルータ回路の汎用性が大きく損なわれる可能性が考えられたため、経路計算を行

う回路を RC (Routing Control)回路としてルータ外部に設置することとし、それ以外の機能を OC (Out Ctrl)回路としてルータ内に配置する構成とした。

図8にこれらをすべて加えたルータ回路を示す。回路の動作は、以下ようになる。まず、入力された先頭フリットが FIFO に格納される。OC 回路は、FIFO から Avalon-ST の入力を受け取り、その中からパケットヘッダのみ取り出し、RC に送る。RC は、受けとったパケットヘッダからデマルチプレクサを制御する制御信号を生成する。RC によって計算された制御信号は、Avalon-ST の channel 信号に割り当てられて送信データと共にデマルチプレクサに出力される。デマルチプレクサにおいては、送られた制御信号に基づいて経路選択が行われ、マルチプレクサで調停処理が行われ、リンクがパケットに割り当てられる。中間パケットと最後尾パケットは、先頭パケット通過時の設定のままルータを通過する。そして、最後尾パケット通過時にリンクが解放される。

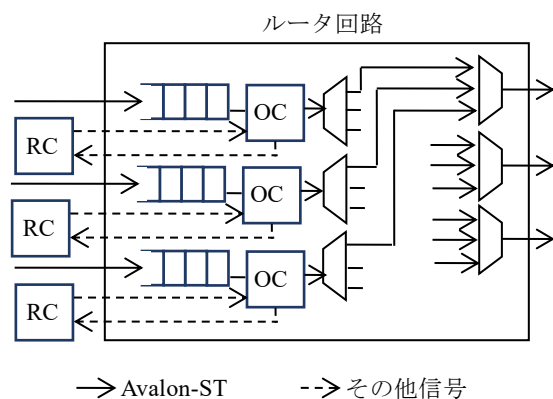


図8 開発されたルータ回路
 Figure 8 Completed Router Circuit

3.2 評価

評価および動作確認に使用する回路の構成を図9に示す。図9のルータは、図8で作成した3入力3出力のルータ回路の入出力それぞれに FIFO を取り付けたものである。この回路において入力 FIFO に、宛先(パケットヘッダ)と送信データを入力し、適切な出力 FIFO に送信データが届くかを確認する。このとき、入力側では一つの FIFO に3つのデータを連続で入力する場合と全ての入力 FIFO に1つずつデータを入力する場合の2パターンを試す。同様に、出力側でも一つの出力に3つのデータを送る場合と、全ての出力に1つずつ送る場合の2パターンを試す。また、これらの動作確認を、動作周波数を変えた状態でもおこない、正常に動作するかを確認する。その結果、動作周波数が50~150MHzであれば正常に動作することを確認した。

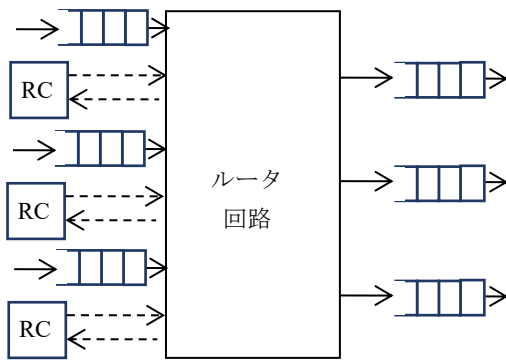


図 9 動作確認用の回路
Figure 9 Circuit for Experiment

表 1 に DE0 ボードに実装した場合の回路量を示す. この表では, 動作確認に使用した図 9 の回路をシステム全体, ルータ回路部分の回路量をルータ部分として表記している. この結果より, テスト用のボードである DE0 であっても 10% 弱の消費で実装できることが確認できる.

表 2 に DE4 ボードに実装した場合の回路量の見込みを示す. ここに示されている使用回路量は DE0 に実装したときのものを用いており, 実際に実装する場合とは若干異なる. この結果をみると DE4 では 1% にも満たない回路の消費で実装できることが確認できる. これらのことから, このルータ回路は実装するシステムに大きな影響を与えずに実装できると考えられる.

表 1 DE0 ボード(CycloneIII)における 3×3 ルータの回路量

Table 1 The Amount of 3×3 Router Circuits with DE0 (CycloneIII).

	システム全体	ルータ
Logic Elements	4,528/15,408(29.4%)	2,132/15,408 (13.8%)
Combinational Functions	2,960/15,408(19.2%)	896/15,408(5.8%)
Logic Registers	2,837/15,408(15.4%)	1,548/15,408 (10.0%)
Memory bits	145,056/516,096 (28.1%)	1,632/516,096 (0.3%)
最大動作周波数	151.15~168.89MHz	174.09~192.98MHz

表 2 DE4 ボード(StratixIV)における 3×3 ルータの回路量(見込み)

Table 2 The Amount of 3×3 Router Circuits with DE4 (StratixIV) (Predicted Value).

	システム全体	ルータ
Logic Elements	4,528/228,000 (2.0%)	2,132/228,000 (0.9%)
Combinational Functions	2,960/228,000 (1.3%)	896/228,000 (0.4%)
Logic Registers	2,837/228,000 (1.2%)	1,548/228,000 (0.7%)
Memory bits	145,056/17,133K (0.8%)	1,632/17,133K (0.01%)

表 3 DE0 ボード(CycloneIII)における 5×5 ルータの回路量

Table 3 The Amount of 5×5 Router Circuits with DE0 (CycloneIII).

	システム全体	ルータ
Logic Elements	6,620/15,408(43.0%)	3,402/15,408 (22.1%)
Combinational Functions	4,783/15,408(31.0%)	2,005/15,408(13.0%)
Logic Registers	4,814/15,408(31.2%)	2,943/15,408 (19.1%)
Memory bits	146,144/516,096 (28.3%)	2,720/516,096 (0.5%)
最大動作周波数	144.72~160.95MHz	159.69~177.4MHz

表 4 DE4 ボード(StratixIV)における 5×5 ルータの回路量(見込み)

Table 4 The Amount of 3×3 Router Circuits with DE4 (StratixIV) (Predicted Value).

	システム全体	ルータ
Logic Elements	6,620/228,000 (2.9%)	3,402/228,000 (1.5%)
Combinational Functions	4,783/228,000 (2.1%)	2,005/228,000 (0.9%)
Logic Registers	4,814/228,000 (2.1%)	2,943/228,000 (1.3%)
Memory bits	146,144/17,133K (0.9%)	2,720/17,133K (0.02%)

4. パケットの送信と受信

現仕様では, パケットの生成および送受信は, NiosII プロセッサを介して行っている. 送信側モジュールにおいては, 1) ヘッダフリットの生成と送信 2) データ本体の送信 3) 最後尾フリットを示す制御信号と最後尾のデータを送信 の順に処理を行っている. 受信側モジュールにおいては,

ヘッダフリットとデータ本体を、続けて受信している。図 10、および図 11 に、送信側および受信側の C 言語ソースコードの例を示す。

```
int main(){
    int i;
    unsigned int tmp, tmp2[8];

    IOWR_32DIRECT( mem1_p+1, 0x0, 1);
    IOWR_32DIRECT( mem1_p, 0x0, 0x01222222);

    IOWR_32DIRECT( mem1_p+1, 0x0, 0x0);
    for(i=0; i<128*256*32*8; i++)
        IOWR_32DIRECT( mem1_p, 0x0, 0x0+i);

    IOWR_32DIRECT( mem1_p+1, 0x0, 2);
    IOWR_32DIRECT( mem1_p, 0x0, 0x02002222);
}
```

図 10 送信側のソースコードの例
 Figure 10 An Source Code Example of Transmission

```
int main(){
    int i,j;
    unsigned int tmp, tmp2[LEN];
    mem3_p = (unsigned int *) FIFO_1_OUT_BASE;

    for(j=0; j<128*256*32*8+2; j++)
        IOWR_32DIRECT( tmp2, i, *mem3_p);
}
```

図 11 受信側のソースコードの例
 Figure 11 An Source Code Example of Reception

動作確認を行うため、単一モジュール内で 1M バイトのデータ転送を行い、転送速度の測定を行った。測定方法として、以下のような方法を行った。

1. パケットヘッダとして、1ワード(4バイト)を送信する。
2. 以下の手順を 32768 回繰り返す。
 - 出力 FIFO に 8ワード(32バイト)のデータを送信する。
 - 入力 FIFO から 8ワードのデータを受信する。
3. 最後尾パケットとして、1ワード(4バイト)を送信する。
4. 最後に、2ワードのデータを受信する。

測定時の環境は以下のとおりである。

- 動作周波数：50MHz
- Avalon-MM における 1ワードのビット数：32ビット
- Avalon-ST における 1ワードのビット数：32ビット
- 出力 FIFO および入力 FIFO の容量：32ワード

測定の結果と課題は以下のとおりである。1パケット(1MB)につき 13.12 秒要したことがわかり、転送速度が 624.4 kbps ということが分かった。また、パケット全体の

転送を NiosII で行っているため、非常に低速である。そのため、DMA 転送[28]を使用する必要があることがわかった。

5. まとめと今後の計画

本研究計画は、System on Chip (SoC)における既存の通信仕様である Qsys-Interconnect をベースとした、FPGA チップ内およびチップ間の統一的な通信仕様の提案、および関連技術に関する研究についての現在までの進捗状況について紹介した。計画では、FPGA チップ内およびチップ間の統一的な通信仕様の提案を目的としている。ルータの設計に関しては、Avalon-ST の仕様に即し、可変長パケットのボード間送受信が可能になる方法を提案し、実装を行ったものの、仮想チャネルの実装については今後の課題となっている。入出力処理に関しては NiosII による可変長パケットの生成と送受信が可能となっているものの、現状の実験環境では通信速度が十分ではないため、DMA コントローラによる制御が必要となることが分かっている。

今後は、仮想チャネルの実装を進めつつ、DMA アクセスの実装を完了させる。将来的には、実アプリケーションによる実証実験として、科学技術シミュレーションや NoC のシミュレーションなどを実施することを考えている。

謝辞 本研究は、JSPS 科研費 JP17K00087 の助成を受けたものである。関係各位に感謝する。

参考文献

- [1] 山内 脩吾, 武者 千嵯, 工藤 千宏, 天野 英晴, マルチ FPGA ボードによる Recurrent Network の高速化, 電子情報通信学会技術研究報告, RECONF2017-62, pp.1-6, 2018.01.
- [2] 田中 大智, Antoniette Mondigo, 佐野 健太郎, 山本 悟, 密結合 FPGA クラスタのための直接網の設計と評価, 電子情報通信学会技術報告, 2018.01.
- [3] 佐野 健太郎, ステンシル計算に基づく科学技術シミュレーションのハードウェアアクセラレーション, 電子情報通信学会誌, Vol.100, No.2, pp.82-86, 2017.02.
- [4] Kentaro Sano, Ryotaro Chiba, Tomoya Ueno, Hayato Suzuki, Ryo Ito and Satoru Yamamoto, "FPGA-based Custom Computing Architecture for Large-Scale Fluid Simulation with Building Cube Method," Computer Architecture News Special Issue for International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies, vol.42, no.4, pp.45-50, 2014.
- [5] T.V. Chu, S. Sato, and K. Kise, "Fast and Cycle-Accurate Emulation of Large-Scale Networks-on-Chip Using a Single FPGA," ACM Transactions on Reconfigurable Technology and Systems (TRETs), vol.10, no.4, pp.27:1-27:27, 2017.
- [6] 佐々木智志, 中野秀洋, 宮内新, 田口亮, "区分線系粒子群最適化法における解探索性能の解析," 進化計算学会論文誌 Vol. 8, No. 1, pp. 1-10, 2017.
- [7] Hidehiro Nakano and Arata Miyachi, Design of a Processor System for Particle Swarm Optimizers, 2015 International Symposium on Nonlinear Theory and its Applications NOLTA2015, pp.606-609, 2015.
- [8] 弘中, 三好他, 「FPGA を用いた計算処理の高速化技術」, 電

- 子情報通信学会誌, Vol.100, No.2, pp.81-112, 2017.02.
- [9] 半田 寛信, 三浦 康之, FPGA クラスタにおけるストリーミングインターコネクトを用いたモジュール間通信のための検討, 電子情報通信学会技術研究報告, FIIS-17-463, pp.1-10, 2018.03.
- [10] 深瀬尚久, 半田寛信, 古市晃久, 三浦康之, Qsys コンポーネントを用いた FPGA クラスタ向けネットワークルータ回路の検討, 電子情報通信学会技術研究報告, FIIS-18-489, pp.1-7, 2018.10.
- [11] 古市晃久, 三浦康之, 半田寛信, 深瀬尚久, 複数の FPGA ボードを使用した FPGA 間データ通信に関する検討第 81 回情報処理学会全国大会, 2019.03.
- [12] W.J. Dally and B. Towles, Principles and Practice of Interconnection Networks, Morgan Kaufmann, 2004.
- [13] W.J. Dally, Virtual-Channel Flow Control, IEEE Trans. on Parallel and Distributed Systems, Vol.3, No.2, pp.194-205, 1992.
- [14] 深瀬尚久, 三浦康之, 渡辺重佳, 直接結合網のルータ回路におけるバッファのリンク間共有法の提案, 電気学会論文誌, Vol. 132(2012), No. 10, pp.1675-1688, 2012.10.
- [15] Naohisa Fukase, Yasuyuki Miura, M.M.Hafizur Rahman, and Shigeyoshi Watanabe, The Performance Evaluation of a 3D Torus Network Using Partial Link-Sharing Method in NoC Router Buffer, IEICE Transactions on Information & Systems, Vol.E100-D, No.10, pp.2478-2492, 2017.10.
- [16] Naohisa Fukase, Yasuyuki Miura, M.M.Hafizur Rahman, and Shigeyoshi Watanabe, The Performance Evaluation of Link-Sharing Method of Buffer in NoC Router, Proc. of 4th International Workshop on Advances in Networking and Computing, pp.567-571, 2013.12.
- [17] Yasuyuki Miura, Masahiro Kaneko, M.M.Hafizur Rahman and Shigeyoshi Watanabe, Adaptive Routing Algorithms and Implementation for TESH Network, Communications and Network (CN), Vol.5, No.1, pp.34-49, 2013.02.
- [18] Ala Ahmed Yahya Haga, M.M. Hafizur Rahmana, Rizal Mohd Nora, Tengku Mohd Tengku Sembokb, Yasuyuki Miura, Yasushi Inoguchi, Uniform Traffic Patterns using Virtual Cut-Through Flow Control on VMMN, Procedia Computer Science, Vol.59, pp.400-409, 2015.08.
- [19] M.M. Hafizur Rahman, Mohammed N. M. Ali, Adamu Abubakar Ibrahim, Dhiren K. Behera, Yasuyuki Miura and Yasushi Inoguchi, A New Static Cost-Effective Parameter for Interconnection Networks of Massively Parallel Computer Systems, Soft Computing in Data Analytics, pp 147-155, 2018.08.
- [20] Yasuyuki Miura, Kentaro Shimozono, Naohisa Fukase, Shigeyoshi Watanabe, and Kazuya Matoyama, An Adaptive Routing Algorithm of 2-D Torus Network Based on Turn Model: The Communication Performance, International Journal of Networking and Computing (IJNC), pp.223-238, 2015.01.
- [21] 中尾司ピエール, 三浦康之, 深瀬尚久, Turn モデルに基づく二次元トーラス・ネットワークの適応ルーティング・アルゴリズムにおける耐故障性評価, 電気学会論文誌, Vol. 139(2019), No. 4, 2019.04.
- [22] Yasuyuki Miura, Kentaro Shimozono, Kazuya Matoyama, and Shigeyoshi Watanabe, The Static and Dynamic Performance of an Adaptive Routing Algorithm of 2-D Torus Network Based on Turn Model, Proc. of the 2014 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'14), pp.114-120, 2014.07.
- [23] Yasuyuki Miura, Junpei Sugioka, Buffer Size Evaluation of Mixture Communication of the Wormhole and Single-Flit Routing, IEEE International Conference on Consumer Electronics – Taiwan (IEEE 2017 ICCE-TW), pp.11-12, 2017.06.
- [24] 「7.Qsys インタコネクト」, Quartus II ハンドブック・バージョン 11.0 Vol 1: デザインおよびシンセシス
- [25] アルティマ技術データベース, 「Qsys に用意されている汎用ペリフェラルの紹介」, ELSENA, Inc.
- [26] “14. On-Chip FIFO Memory Core”, Quartus II Handbook Version 9.1 Volume 5: Embedded Peripherals, Altera.
- [27] 「17. Avalon Streaming チャンネル・マルチプレクサおよびデマルチプレクサ」, Quartus II Handbook Version 9.1 Volume 5: Embedded Peripherals, Altera.
- [28] アルティマ技術データベース, 「DMA 転送の実現 (NiosII)」, Altima Corporation
- [29] 木村 誠聡, FPGA を用いた画像処理フィルタの映像処理への適用, 電子情報通信学会 基礎・境界ソサイエティ Fundamentals Review, Vol.10, No.2, pp. 104-112, 2016.
- [30] 比嘉慎哉, 置田真生, 萩原兼一, 伊野文彦, PU プログラムにおける静的参照関係を表すハイパーグラフの分割を用いた参照効率のよいデータ配置, 情報処理学会研究報告, Vol.2018-HPC-169, No.15, 2019.03.
- [31] 長坂侑亮, 額田彰, 小島諒介, 松岡聡, 小疎行列積計算の GPU 最適化, 情報処理学会研究報告, Vol.2018-HPC-169, No.19, 2019.03.
- [32] 山口拓真, 藤田航平, 市村強, 堀宗朗, ラリス ウィジャラトネ, GPU による 3 次元非構造格子有限要素解析の高速化, 情報処理学会研究報告, Vol.2018-HPC-169, No.7, 2019.03.
- [33] 辻大亮, 多田野寛人, 朴泰祐, 池田亮作, 佐藤拓人, 日下博幸, 都市気象 LES コードの並列 GPU 環境における高速化, 情報処理学会研究報告, Vol.2018-HPC-169, No.8, 2019.03.
- [34] 中道安祐未, 小林諒平, 藤田典久, 朴泰祐, GPU・FPGA 混載ノードにおけるヘテロ演算加速プログラム環境に関する研究, 情報処理学会研究報告, Vol.2018-HPC-169, No.10, 2019.03.
- [35] 渡部裕, 李珍泌, 佐野健太郎, 朴泰祐, 佐藤三久, FPGA へのオフロード最適化のための SPGen と OpenCL の統合の検討, 情報処理学会研究報告, Vol.2018-HPC-169, No.11, 2019.03.
- [36] 小松弘人, 川原佑宇紀, 島田拓夢, 安永守利, 歌川紀之, 黒田千歳, Zynq FPGA を用いた自己組織化マップのハードウェア化と打音検査システムへの適用, 電子情報通信学会技術研究報告, FIIS-18-487, pp.1-5, 2018.10.
- [37] 肥川宏臣, 拡張性が高いハードウェア自己組織化マップ, 電子情報通信学会技術研究報告, FIIS-18-491, pp.1-5, 2018.10.
- [38] Nios II Performance Benchmarks, <https://www.intel.com/content/www/us/en/programmable/documentation/iga1408573581143.html>