

## FPGA アレイを用いたスケーラブルな Reconfigurable HPC

飯島 浩晃<sup>†</sup> 佐藤 一輝<sup>†</sup> 関根 優年<sup>†</sup>

<sup>†</sup> 東京農工大学 工学府

〒184-8588 東京都小金井市中町 2-24-16

E-mail: †{ijijima,sato}@ sekine-lab.ei.tuat.ac.jp, ††sekinem@ cc.tuat.ac.jp

あらまし 大規模演算をスケーラブルに行える基盤として、多数の FPGA 同士を相互接続した FPGA アレイが提案されている。我々が提案する FPGA アレイは、大規模 FPGA を搭載し、外部 IO を大量に装備した小型の FPGA カードを、格子状に並べた構成を取る。演算回路を格納する FPGA は、対象問題にあわせて回路を構成することができるため、比較的安価に専用回路を用いた高性能な演算を手に入れることができる。FPGA アレイは PCI 型 FPGA カード (hwModule) に接続し、hw/sw 複合体に組み込むことで容易な制御を目指す。本稿では提案システムによって、差分法によるポアソン方程式の演算を、どれだけ高速に行うことができるかについて予備実験を行った。

キーワード ハードウェア、ソフトウェア、FPGA、HPC、シミュレーション、hw/sw 複合体

## Scalable RHPC(Reconfigurable HPC) by using FPGA array

Hiroaki IJIMA<sup>†</sup>, Kazuki SATO<sup>†</sup>, and Masatoshi SEKINE<sup>†</sup>

<sup>†</sup> The Faculty of Technology, Tokyo University of Agriculture and Technology

2-24-16 Naka-chou, Koganei-shi, Tokyo, 184-8588 Japan

E-mail: †{ijijima,sato}@ sekine-lab.ei.tuat.ac.jp, ††sekinem@ cc.tuat.ac.jp

**Abstract** FPGA array that interconnect among many FPGAs was reported as foundation for large-scale operations. FPGA array we propose in this paper consist of a grid structure by using small FPGA card, which equips with large-scale FPGA and large amount of external IOs. This FPGA can reconfigure the arithmetic circuit to fit the target problem, so get a dedicated circuit for high-performance computing at relatively cheap. FPGA array aims at easy to control by embedding in hw/sw complex through PCI type FPGA card. This paper takes the pilot study how fast proposed system can calculate Poisson's equation by the finite difference method.

**Key words** Hardware, Software, FPGA, HPC, Simulation, hw/sw complex

### 1. はじめに

近年において、大規模演算の環境が高度に整備されてきたのを受け、生物・化学・天文・工学などの広い分野において HPC(high-performance computing) による演算手法が利用されている。HPC は複数の計算機システムを結合し、一つのシステムとして演算環境を提供する。一つの計算機システムの性能が低くとも、システム全体としては高速に演算が可能となる。HPC には、汎用 PC によるグリッド構成のものや、GPGPU(General Purpose Graphics Processing Unit) を利用したグリッド構成など、IC をならべた構成のものもある。

また FPGA を用いて高性能な演算を行う研究も、みられるようになった [1], [2]。特に大規模演算向けに構成されているクラスタ型ものは、RHPC(Reconfigurable High performance Computing) や High-Performance Reconfigurable Comput-

ers(HPRCs) などと呼ばれている。FPGA を利用する利点は、対象問題にあわせて最適な演算器を利用することができることがあげられる。それに対して欠点は、PC とは違い演算環境が整っていない、CPU より動作周波数が低い、利用者が独自にシステムを構築する必要があるなどである。

その問題に対し、我々は hwModule と呼ばれる PCI 型 FPGA ボードを利用した、hw/sw 複合体を提案した。hw/sw 複合体の特長として、汎用性、並列分散処理や外部機器との接続などがあげられる。

hw/sw 複合体のアプリ例には、これまで hwModule のネットワーク対応に関する研究がなされてきた [3]。しかし、大規模並列化を要求する、数値シミュレーションなどのアプリケーションには、計算機間を非常に高速にデータ転送することを必要とするものもあり、必ずしも HPC としての要求性能を満たせないと考えている。そこで我々は FPGA アレイを hwModuleV2

経由で hw/sw 複合体から制御する方法を考案した。

FPGA アレイとは、多数の FPGA 同士を相互接続したものである。我々が提案する FPGA アレイは、hwModuleV2 から制御し、大規模演算をスケーラブルに行える基盤部品である。FPGA は対象問題にあわせて回路を構成することができるため、様々なアプリケーションに適用でき、安価で高性能な演算環境を手に入れることができる。アレイのサイズはスケーラブルであり、用途に応じた構成を取ることができ、hw/sw 複合体に組み込むことで、制御を容易にする。以上の利点を生かし、パーソナル・スパコンとして利用できるシステムとしての普及を目指す。本稿では、FPGA アレイを用いたアプリケーションの例として、ポアソン方程式の数値解析を取り上げる。提案システムによってポアソン方程式の演算をどれだけ高速に行うことができるかを報告する。

## 2. RHPC

単一の計算機では対処できない規模の大規模演算を行う場合、専用ハード化や大規模並列化による演算性能向上を目指すこととなる。

### 専用ハード化

専用ハード化は汎用演算器である CPU を使用せず、対象演算用に作製した専用ハードウェアを用いるものである。一般に相当の演算速度向上を期待できるが、ASIC などで実装する場合は、製造コスト、期間の点で劣る。近年において、回路を適時再構成可能な FPGA が普及しており、FPGA に専用ハードウェアを実装する方法は一般的である。

### 大規模並列化

並列化には、CPU をマルチコア化する小規模な並列化もあるが、大規模な並列化は、単独の PC では過大な演算量を複数の PC で分散させることにより、システム全体で高速な演算環境を構築する。

FPGA を複数個使用する手法は、ハード化と大規模並列化を同時に実装する方法といえる。対応アプリケーション専用の ASIC とは違い、FPGA は様々なアプリケーションに対応する。しかしながらこの手法は、問題点も存在する。

#### (1) FPGA 利用

FPGA を簡便に利用する基盤がない

#### (2) 制御

大量の FPGA の管理の問題

#### (3) ボード形状

FPGA を大量に配置できるハードウェア

(1), (2) の問題については、我々が提案した hw/sw 複合体が解決している。hw/sw 複合体とは、hwModule の FPGA を仮想回路 (hwNet) として利用し、hwObject と呼ばれる hwNet の詳細な制御を隠蔽するクラスを継承することにより、回路資源を容易に利用できるシステムである。

我々が利用している最新の PCI 型 FPGA ボードは hwModuleV2 (図 1) である。従来の hwModuleV1 [4] と類似の構成を持つ。特長としては、従来の 4-16 倍のゲート数が利用できる FPGA、 $16 \times 4 = 64$ [MB] のローカルメモリ、 $40 \times 3 = 120$ [本]



図 1 hwModuleV2 の外観

の GPIF (外部 IO) 線が搭載されている。また潤沢なボード内部のバス (LM-Bus, HN-Bus, FPC, FFC) が配線されている。hwNet として利用できる回路資源は、最大で  $XC3S4000 \times 3$  (12[MGate] 相当) であり、従来と比較して 20 倍の大きさの回路を、リコンフィギュレーションすることにより実装できる。hwModuleV2 をさらに増設することで、リコンフィギュラブルな HPC を目指すことは可能である。

しかし、hwModuleV2 は PCI ボードであり、PCI スロットの本数制約など単独の PC 上での拡張には物理的に限界がある。また PCI は共有バス故に輻輳が起きる可能性がある。HPC とまでいえる規模の演算環境を実現するのは、hwModuleV2 のみでは難しいといえる。

この問題に対応するため、本稿では FPGA アレイによる RHPC を構築することとした。

## 3. FPGA アレイ

大規模 FPGA を搭載し、外部 IO を大量に装備した、小型の FPGA カードを開発し、それらを格子状に並べることで、FPGA アレイとする。FPGA カード同士の接続形態は自由に構成することができ、直線構成や平面構成、立方体構成などを組むことができる。したがって対象問題にあわせた構成を取ること、ユーザー要求に対して最適なパフォーマンスを実現できるようにする。また FPGA は対象問題にあわせて回路を構成することができるため、様々なアプリケーションを FPGA アレイにて解くことができる。

### 3.1 FPGA カード

FPGA カードは RHPC として、集積が可能な構成を取る必要がある。FPGA カードについて、大容量のゲート数を利用できる FPGA と、別の複数の FPGA カードと接続するための隣接接続端子 (隣接端子) による構成を基本とする (図 2)。

FPGA アレイは高速に相互接続する必要があるため、FPGA カード間の通信は高速に行えるようにする。FPGA カード数に比例して、システム全体の一方方向あたりのバス本数は増加する。したがってバス幅はスケーラブルであるといえる。

FPGA カード上にはデータ一時格納用の SDRAM を搭載する。シミュレーションデータなどを格納していく。ボードごとにメモリが搭載されているので、メモリ分散型のシステムといえる。

### 3.2 hwModuleVS

実際に FPGA カードを制作し、hwModuleVS と名付けた。

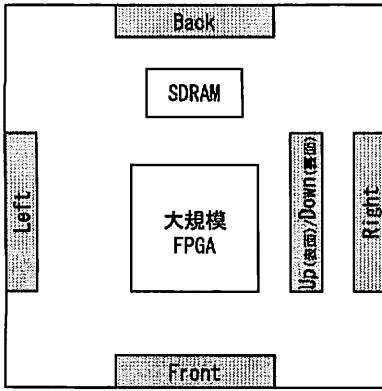


図2 FPGAカード概略図

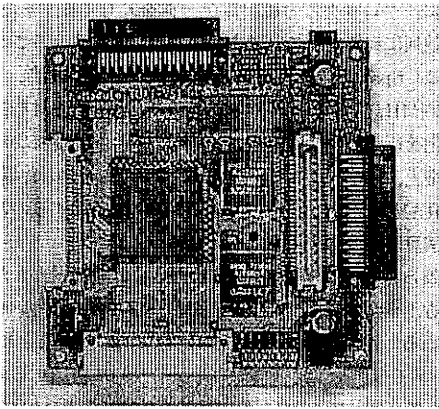


図3 hwModuleVS 外観

外観を図3に示す。サイズは95[mm]×95[mm]×52[mm] (縦×横×高)である。1[m]角の立方体でFPGAアレイを構成した場合、約2000個のhwModuleVSを積層することが可能である。hwModuleVSの仕様について、表1に示す。低価格FPGAであるSpartanシリーズの中でも、大容量の回路を格納できる、XC3S4000を採用した。FPGAのIOについても、隣接端子用に本数が多いFG676を採用した。電源は12[V]が必要で、汎用PCのATX電源から供給できる。

前後左右方向の接続についてはコネクタ直結とし、高速転送を可能とする(図4)。隣接接続端子の上下端子は、積層に適するよう、同一の場所にレイアウトされている。ただし、ボードの廃熱を考え、直結とはせずにケーブルを介して接続する。

#### 4. 差分法によるポアソン方程式の演算

ポアソン方程式を数値解析で解く方式は、データ並列型の大規模演算であり、提案システムの評価に使用する[5]。ポアソン方程式は、電磁気学や半導体工学で利用される、楕円型の偏微分方程式である。この方程式を差分法で解く場合、領域内の格子点数だけの連立方程式を解く必要がある。格子幅については任意でよいが、精度に反映されるため、いかに効率よく大型の

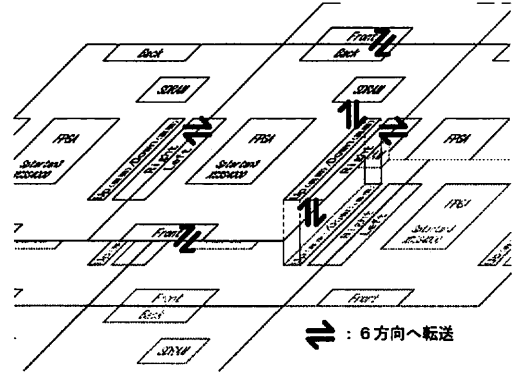


図4 hwModuleVSの相互接続

表1 hwModuleVSの仕様

配線	
hwModuleVS間	(data)32+(ctrl)17 = 49[本]
隣接接続端子	x(上下左右前後)6 = 294[本]
SIM-Data線	(data)32+(ctrl)24 = 56[本]
Configuration線	23[本]
搭載チップ	
FPGA	XC3S4000-4FG676
SDRAM	MT48LC2M32B2-6 x 2
(データ幅)	64[bit]
(容量)	16[MByte]

連立方程式を解くかが問題となる。解きたい問題の領域について、格子点間隔を狭めれば格子点が増加し、従って大規模な演算が必要となる。

ポアソン方程式を反復法で解く方式を、FPGAアレイに実装することで、ポアソン方程式で示される数値解析問題が高速で解くことができれば、FPGAアレイの有効性を実証することができる。

ポアソン方程式は、式(1)で表される。

$$\Delta\phi = \rho \quad (1)$$

一次元の場合の式について、偏微分方程式を差分法によって変形すると、以下の式に変形できる。

$$\begin{aligned} \frac{\nabla\phi_i - \nabla\phi_{i+1}}{h} &= \frac{1}{h} \left( \frac{\phi_i - \phi_{i-1}}{h} - \frac{\phi_{i+1} - \phi_i}{h} \right) \\ &= \frac{1}{h} \left( \frac{-\phi_{i-1} + 2\phi_i - \phi_{i+1}}{h} \right) \end{aligned} \quad (2)$$

$$\phi_i^{new} = \frac{h^2\rho + \phi_{i-1}^{old} + \phi_{i+1}^{old}}{2} \quad (3)$$

式(3)を反復法にて解く。

二次元の場合の式についても同様に変形すると、以下のようになる。

$$\begin{aligned} \frac{1}{h} (\nabla\phi_{i,j} - \nabla\phi_{i+1,j} + \nabla\phi_{i,j} - \nabla\phi_{i,j+1}) \\ &= \frac{-\phi_{i-1,j} + 2\phi_{i,j} - \phi_{i+1,j}}{h^2} \\ &+ \frac{-\phi_{i,j-1} + 2\phi_{i,j} - \phi_{i,j+1}}{h^2} \end{aligned} \quad (4)$$

$$\phi_{i,j,k}^{new} = \frac{\alpha}{4}$$

$$\alpha = h^2\rho + \phi_{i-1,j}^{old} + \phi_{i+1,j}^{old} + \phi_{i,j-1}^{old} + \phi_{i,j+1}^{old}$$

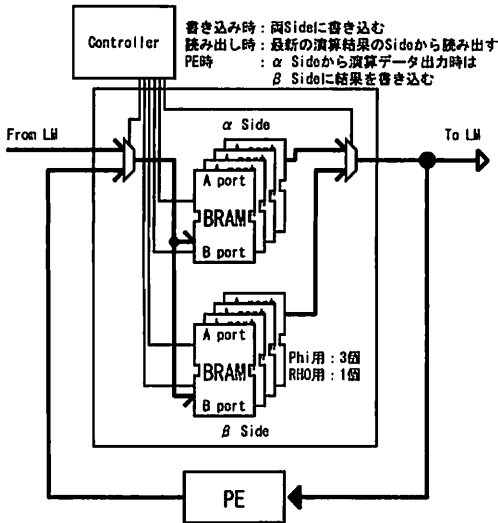


図5 ポアソン方程式演算回路

#### 4.1 プラズマ解析

ポアソン方程式を解く必要があるアプリケーションとして、プラズマの数値解析があげられる。プラズマとは、荷電粒子を含む気体であり、荷電粒子の振る舞いをシミュレートすることで、プラズマの様子を観察できる。 $\phi$ と $\rho$ は問題によって異なるが、今回はプラズマのシミュレーションを参考にするため、 $\phi$ は電位、 $\rho$ は電荷密度と誘電率の式となる。

プラズマ物理の分野では、PIC法 (Particle in Cell Method) が利用されている [6]。プラズマの一番基本的な方程式である、粒子の運動方程式、電場や磁場の方程式などを基本式として用いるため、物理に一番忠実なモデルとして知られている。同時に、負荷が大きいモデルでもある。

PIC法は、グリッドで分割された空間上に、電荷 $q$ 、質量 $m$ の荷電粒子が飛び回っているモデルを考える。演算の手順は、1. 粒子分配、2. 場の量の計算、3. 粒子加速、の3つに分かれる。本稿では、2. 場の量の計算を回路化して高速化する。

FPGAアレイでは、境界条件によらず、方程式の式変形をせずに直接に演算を行う方法を用いる。3. 粒子加速などはCPUを用いて行う予定ではあるが、将来的にはFPGAアレイ上に組み込むことで、さらに高速に演算できると考えている。

#### 5. ポアソン方程式演算回路

前項において、差分法によるポアソン方程式の数値解析を、演算回路にて行うことは述べた。本項では、ポアソン方程式演算回路 (ポアソン回路) の詳細を述べる。

ポアソン回路の構成について、図5に示すブロックからなる。

- ポアソン回路演算回路コア (PE)
- キャッシュ
- メモリ入出力部
- 回路制御部

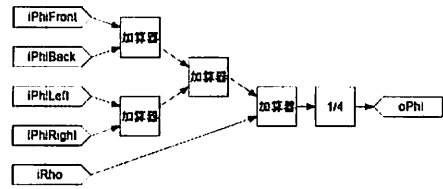


図6 ポアソン回路 PE 部

表2 ポアソン回路: PEの回路規模 (分母はXC3S4000の利用可能Slice数)、最大動作周波数

回路規模	1,729/27,648[slices]
最大動作周波数	140[MHz]

図6に示すPEについて、パイプライン構造とすることで、最大動作周波数を向上させる。方程式の計算が一回終了するのを待たずに一クロックごとに計算データを入力する。また、PE自体を並列に実装することでさらに高速に動作させることが可能となる。Spartan3シリーズのSpeedGrade4の場合、PEは最大140[MHz]で動作が可能である。ただし、現在は制御回路の最適化を行っていないので、66[MHz]で動作させている。回路の構造上、PEの追加は可能で、一つPEを追加した場合、2つのBRAMを追加する必要がある。PEを追加すると比例して、演算時間は減少するようになる。

場の値 $\phi$ は繰返し演算する必要があり、演算負荷がかかる要因である。また常に演算しているので、常に収束していると仮定している。演算結果となる場の値を一回の演算ごとにMM (メインメモリ) やLM (ローカルメモリ) に格納するのは低効率となる。従ってFPGAの内部メモリであるBRAM (Block RAM) を使用して場の計算の途中経過をFPGA内部に格納する。こうすることで、できるかぎりPEをパイプラインをストールさせないでポアソン回路を動作させることができる。

ポアソン回路の制御手順について示す。

- (1) BRAMにLMの演算データをコピーする
- (2) PEをスタートさせる
- (3) 適時、演算経過のデータをLM経由で読み出す

Spartan3に実装した場合、ポアソン回路のPE部分の回路規模と最大動作周波数について表2に示す。二次元のポアソン回路のPEあたり、4回加算処理を行うので、加算器は4個必要となる。hwModuleVS 1枚あたりのPE搭載可能数については、コントロール用の回路のマージンを考えると、10個のPEがXC3S4000に実装できると考えられる。hwModuleV2で演算するとき、演算中はLMを使用せず、FPGA内で演算を実行するため、2ポートBRAMに電場・電荷のデータを一時的に格納する。一回の演算 (1[clk]) あたり、2Dだと (3(電場)+1(電荷)) $\times$ 2(ライン)=8のBRAMラインが必要となる。512x32[bit]がBRAMの一単位なので、8BRAMあたり、最大20x20[node]の場について演算することができる。ただし現在の回路において、データの一時格納用BRAMは、データを読み取り・演算・結果を書き込むことは同時にできるが、さらに同時に隣接FPGAとデータ交換をすることはできない。隣接端

表 3 テストに使用した PC のスペック

OS	Windows 2000 SP4
CPU	AMD AthlonXP2500+ 1.83GHz
Memory	DDR SDRAM 2700 2GB
Motherboard	ASUS A7N8X

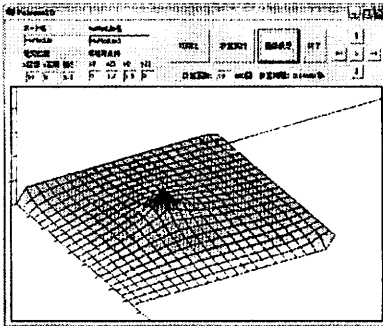


図 7 ポアソン方程式演算アプリ

子と通信する場合、方向ごとに BRAM に用意し、相互にデータをやりとりできるようにする。

## 6. 予備実験

### 6.1 ポアソン方程式演算回路の性能

hwModuleV2 上でポアソン方程式演算回路（ポアソン回路）を動作させ、演算回路単体の性能について述べる。ポアソン回路は hwNet（仮想回路）として実装される。本来 FPGA アレイでは、hwModuleV2 について hwModuleVS を制御するために使用する。hwNet は、hwModuleV2 と hwModuleVS の両方のボードで動作可能のため、ポアソン回路 hwModuleV2 でも動作させることができる。本項では 1FPGA あたりの演算性能について述べる。検証機のスペックを表 3 に示す。

ポアソン方程式の演算は、ソフトウェア上で記述した場合、for 文で Node の個数分について式 (4) を繰り返し演算行が必要がある。従って非常に負荷のかかる計算処理となり、 $20 \times 20$  の 2 次元の場では相当の演算時間が予想される。

図 7 は、今回作成したポアソン回路を動作させるためのアプリケーションである。場の境界値と、電荷の位置を指定して、場の計算を行うアプリケーションである。hwNet を使用した場合、演算回路はパイプライン処理によって最適化されており、高速に演算が可能である。PE を一つ搭載したポアソン回路について、場の計算の繰り返し回数によって、演算時間がどのように変わるかを計測した。

演算時間の結果を図 8 に示す。hwNet の動作クロックは 66[MHz] とした。結果からハードウェア (hwNet) を用いた演算時間が、ソフトウェアのみの場合と比較して、最大 8.1 倍高速に演算を行うことができることがわかった。hwModuleVS においても、同じ FPGA チップを使用するので、同等の性能を得ることができる。

繰り返し回数が少ない場合に、ソフトウェアのほうが演算時間が速い件について、以下の理由による。回路を制御する時間は、

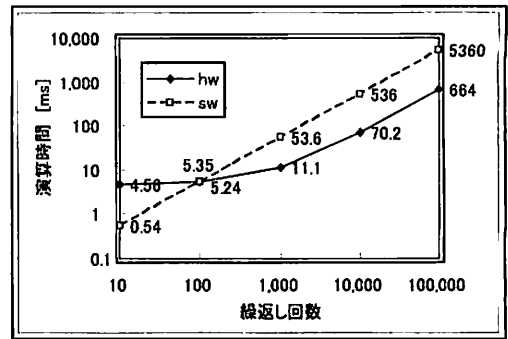


図 8 ポアソン方程式の演算結果

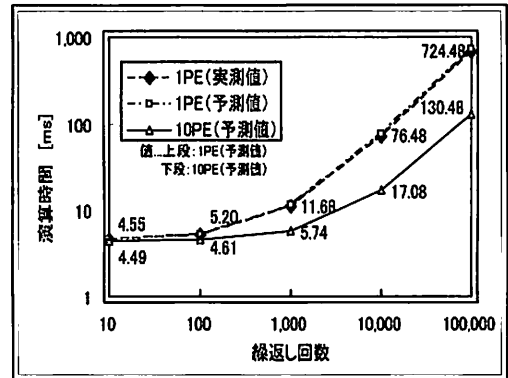


図 9 ポアソン方程式の演算予測値

表 4 ポアソン方程式回路を制御するために必要とする時間

演算データ書き込み時間 [ms]	
LMIに書き込み	1.02
BRAMに書き込み	1.34
合計	2.36
演算データ読み出し時間	
BRAMから読みだし	0.34
LMIから読みだし	1.78
合計	2.12
総計	4.48

繰り返し回数によらず一定である (表 4)。したがって繰り返し回数の少ない演算の場合、ほとんどが回路を制御する時間に割り当てられ、PE はアイドルのままとなり、結果ハードウェアによる計算速度が見いだせなくなっている。逆に繰り返し回数が多ければ、回路の制御時間は隠蔽され、専用ハードのメリットを生かした演算状態となる。

PE1 つを搭載した場合の演算回路の動作時間について、一回の繰り返しあたりの内訳は、 $20 \times 22$  (演算データの出力時間) +  $40$  (PE のレイテンシ) =  $480$  [clk] かかる。なお、今回は PE を 1 つのみの実装であったが、最大 10PE を搭載可能で、その場合の演算時間は  $(20/10) \times 22 + 40 = 84$  [clk] となり、5.71 倍高速化できる。1PE、10PE の場合の予測値 (PE のレイテンシ含む) について、図 9 に示す。10PE の予測値をみると、はっきりと速度向上が見取れる。

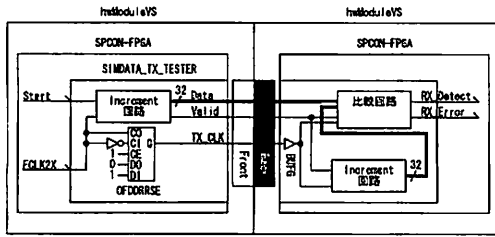


図 10 隣接接続端子をまたぐ hwModuleVS 間転送回路

## 6.2 hwModuleVS の評価

hwModuleVS は前後左右上下にデータを送れる構造をしているが、この部分の転送性能はアプリケーションの性能に影響する。たとえば、複数の hwModuleVS を使用して、ポアソン回路を動作させる場合、隣接する場のデータ（隣接値）を相互に転送する必要がある。このとき、hwModuleVS 間の転送速度が低速であると、PE をアイドルにしたままの時間が増大し、結果として演算性能に影響する。

本項では一秒あたりどれだけの転送速度を得ることができるかを検証する。検証に使用した回路を図 10 に示す。データは TX\_CLK と同期して転送した。結果 TX\_CLK を 133[MHz] にした場合でも、RX\_Error はアクティブにならなかった。したがって、hwModuleVS 間は、最大 532[MB/s] の転送レートを得ることができることがわかった。

前項で使用したポアソン回路は、 $20 \times 20$  の場の演算に  $480[\text{clk}]$  かかる。このとき、一辺あたり 20 個 (80[Byte]) ある隣接値をやりとりする時間は、演算中に隣接値を転送しない場合で  $11$  (一辺の転送時間)  $\times 2 = 22[\text{clk}]$  以内であると考えている。したがって隣接値を考慮した演算時間は  $480 + 22 = 502[\text{clk}]$  となる。なお隣接値を考慮しない場合、PE のレイテンシ (40[clk]) を考慮する必要はなく、440[clk] で一回の演算が終了する。その場合と比べて、14% 演算時間が増大するが、計算できる場の大きさは任意に増加させることができるようになる。

## 6.3 FPGA アレイの性能予測

本項では、1[TFLOPS] の演算性能が、何枚の hwModuleVS で実現できるかについて述べる。図 6 より、1 ノードあたりの場の計算には、4 回の浮動小数点数演算を行う。前項において、FPGA アレイ用のポアソン回路は、一回の繰返しに  $502[\text{clk}]$  かかるとした。このとき演算結果は  $20 \times 20 = 400$  個得ることができる。したがってポアソン回路の一回の繰返しで、1600 回の浮動小数点数演算を  $502[\text{clk}] (7,530[\text{ns}])$  で行うことができると予測する。FLOPS に換算すると、hwModuleVS 一枚あたり、212[MFLOPS] の演算が可能であるとした。このことより、1PE を実装するポアソン回路 (66[MHz] 駆動) にて 1[TFLOPS] を達成するには、4716 枚必要となる。

hwModuleVS に搭載されている FPGA の場合、10PE を実装するポアソン回路を作成することは可能と考えている。また、PE 自体は 133[MHz] 動作は可能であり、制御回路の最適化をすれば、現在の倍速で動作が可能となる。10PE の場合、一回の繰返しに 44 (演算データの出力時間) + 40 (PE のレイテン

シ) + 22 (隣接値の転送) =  $106[\text{clk}]$  かかる。ポアソン回路の一回の繰返しで、1600 回の浮動小数点数演算を  $106[\text{clk}] (795[\text{ns}])$  で行い、FLOPS に換算すると、hwModuleVS 一枚あたり、2.01[GFLOPS] の演算が可能である。このことより、10PE を実装するポアソン回路 (133[MHz] 駆動) にて 1[TFLOPS] を達成するには、497 枚必要となる。

## 7. むすび

本稿では、hwModuleVS を用いて構築する FPGA アレイの提案、およびその FPGA アレイの性能に関する予備実験を行った。我々が提案する FPGA アレイは、FPGA 同士を密に接続でき、対象問題の次元（一から三次元）似合わせた構成を取ることができる。予備実験では、ポアソン方程式回路を hwNet として実装し、ソフトウェアのみの場合と比較して、最大 8.1 倍の演算性能向上を確認することができた。また、hwModuleVS 間の転送速度について、最大 133[MHz] 駆動、データ幅 32[bit] で転送することができ、一秒あたりの換算で最大 532[MB/s] の転送レートを実現できることを確認した。今後は、本稿では予測値にとどめた FPGA アレイの性能を実証するべく、実際に hwModuleVS に回路を実装していくこととする。

## 8. 謝 辞

大学発ベンチャー創出推進「1811 動的な仮想回路による超高速 Hw/Sw 複合システムの研究開発」の資金による。

### 文 献

- [1] Herboldt, M. C., VanCourt, T., Gu, Y., Sukhwani, B., Conti, A., Model, J., and DiSabello, D., "Achieving High Performance with FPGA-Based Computing", Computer, Vol.40, no.3, pp.50-57, 2007/3.
- [2] Fukushima, T. and Makino, J. and Kawai, A., "GRAPE-6A: A single-card GRAPE-6 for parallel PC-GRAPE cluster system", Arxiv preprint astro-ph/0504407, 2005.
- [3] 飯島浩晃, 岸本浩, 関根優年, "hw/sw 複合体における Remote hwObject のハードウェア構成について", 電子情報通信学会技術研究報告 CAS 研究会, Vol.CAS2006-73, pp.19-24, 2007/1.
- [4] K.Kudo, Y.Myokan, W.C.Than, S.Akimoto, T.Kanamaru, M.Sekine, "Hardware Object Model and Its Application to the Image Processing", IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E87-A, no.3, pp.547-558, 2004/3.
- [5] TOMITA,Eisaku YAMAMOTO,Toshio IGUCHI,Yukihiro, "A Realization of Device Simulation Engine", IPSJ SIG Notes, 09196072, Information Processing Society of Japan (IPSJ), Vol.98, no.113, pp.69-74, 1998/12.
- [6] 内藤, 裕志 徳田, 伸二, "1. プラズマの粒子シミュレーションコードの並列化 (<小特集> 超並列計算機のシミュレーションとプラズマ)", プラズマ・核融合学会誌, 09187928, 社団法人プラズマ・核融合学会, Vol.72, no.8, pp.737-743, 1996/8.