

小堀 友義†, 丸山 勉‡, 星野 力‡  
 †筑波大学 工学研究科 ‡筑波大学 機能工学系

## 1. はじめに

セルラオートマトン法は様々な現象の振舞いをシミュレートするために用いられる。セルラオートマトンは原理的に高い並列性を有し、現在までに並列計算の分野で多数の研究がなされてきた。<sup>1)2)3)4)</sup>

本研究では、PC 一台と FPGA を 1chip 搭載している PCI ボードから構成される小規模なシステムを用いたセルラオートマトン法の高速計算システムを提案する。このシステムではユーザがセルラオートマトン専用言語を用いて、遷移則や格子の構成を記述する。記述されたソースはセルラオートマトンの高速計算手法を基にしたハードウェアライブラリを用い、FPGA 内に構成される回路として変換される。このセルラオートマトンの高速計算手法ではパイプライン化と並列化により、大規模な格子面における格子を効率よく計算することができる。

性能比較モデルとして、FHP-III モデルとライフゲームを用いた。結果として Pentium-III 700MHz に比べ FHP-III モデルにおいて 155 倍、ライフゲームにおいて 250 倍の速度向上が得られた。

## 2. システムの概要

本研究で提案する高速計算システムは

- セルラオートマトン専用言語を対象とした Verilog HDL へのコンパイラ
- Xilinx 社製 FPGA (Virtex XCV1000) を搭載した PCI ボード (Celoxica 社製 RC1000-pp)
- FPGA の制御とディスプレイへの表示を行う PC により構成されている。このようなシステムを用い、ユーザはまず、セルラオートマトン専用言語により実行するセルラオートマトンの構成および遷移則を記述、コンパイラを用いて高速計算回路に変換し、変換回路と一緒に生成された FPGA 制御および表示用の C 言語ソースを用いてセルラオートマトンを実行する。

本研究で使用した FPGA 搭載 PCI ボードの構成を図 1 に示す。この PCI ボードは 1M ゲート相当の規模を持つ FPGA (Xilinx 社製 Virtex XCV1000) を 1chip と外部 RAM

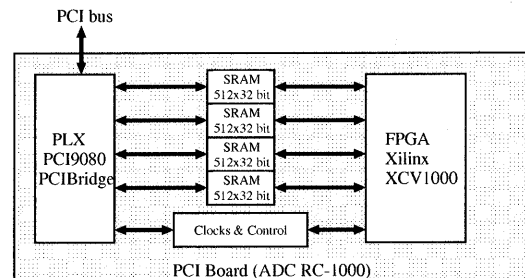


図 1 FPGA 搭載 PCI ボードの構成図

として合計 8MB の SRAM を搭載している。

格子面のデータの保持には外部 RAM が使用される。外部 RAM を読み出し用と書き込み用の 2 種類の RAM に分けることによって、各格子のデータを移し変えずにそのまま次の計算を行うことができる。PC とのデータ通信は DMA 転送により外部 RAM と PC とで行われる。格子面全体のデータ量が外部 RAM 容量を越える場合、多少性能は落ちるものの DMA 転送を繰り返し行い、格子データを計算が終了する度に FPGA に転送することで、格子面全体の計算を行うことができる。

## 3. 高速化手法とハードウェア

### 3.1 高速化手法

本システムの軸となる高速化手法は複数の計算回路をパイプライン状に配置し、複数世代先の格子の状態を計算することを基本としている。この高速化手法は以下の 3 種の大小関係により高速化手法が異なる。

- (1)  $FPGA$  の入出力データ幅  $\geq$  格子面の幅
- (2)  $FPGA$  の入出力データ幅  $>$  格子面の幅  $<=$   $FPGA$  の内部メモリ容量
- (3)  $FPGA$  の入出力データ幅  $<$   $FPGA$  の内部メモリ容量  $<$  格子面の幅

### 3.2 高速計算回路の詳細

FPGA 上に実装される高速計算回路は、主に以下に示す演算ユニットを格子状に配置したものにより構成されている。図 2 に格子の次の状態を計算する演算ユニットの構成を示す。演算ユニットは主に、Distributed RAM を用いて構成される格子データを保持するバッファと、4 つのバッファから出力されたデータの内計算時に必要なデータを選択し変換回路へデータを送るセレクタ、遷移則に従い各格子の次の世代の状態を計算する変換回路により構成されている。

各格子について次の世代の状態を求める場合、前列、後

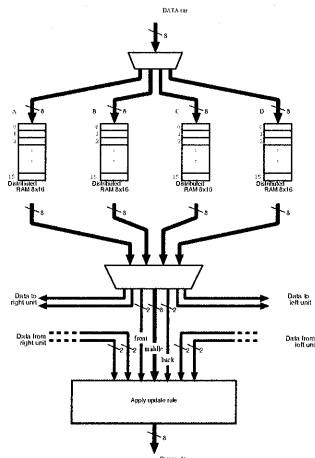


図2 演算ユニットの構成

列を含めた3列分の格子データ必要となる。そのため、演算ユニットにバッファを4つ並列に配置することで、パイプラインが詰まることなく1clock毎に次の格子の状態を計算し、左側の演算ユニットへ格子データを出力することができる。ユーザは演算ユニット1つ分の回路規模やFPGA内の構成可能な回路の規模等の情報により、任意に演算ユニットの配置を決定することができる。

#### 4. ソフトウェア

本章では、セルラオートマトン専用言語について述べる。この専用言語は2次元のセルラオートマトンのみに対応している。この言語は以下に示す7つの部分により構成されている。

- (1) 格子面の規模を指定
- (2) 格子の構造を表記
- (3) 近隣格子との位置関係を表記する際に必要なローカル変数の設定
- (4) 近隣格子と計算対象となる格子との位置関係および近隣格子から与えられるデータのbit位置の表記。近隣格子の位置は計算対象となる格子の位置を基準としたxy座標表現で表す。
- (5) 境界条件の表記を行う部分。この部分に表記されたものは前述した近隣格子と計算対象となる格子との位置関係を表記する部分に追加される。格子面の4辺はxBoundary0, xBoundaryX, yBoundary0, yBoundaryYによりそれぞれ表される。格子面上での障害物を表現する場合には、ifBoundaryを用いる。
- (6) 遷移則の記述を行う部分。ここでは論理演算子、if、switch等を用いて遷移則を記述する。
- (7) 格子の初期値を設定するのに必要な処理の記述をする。この部分はそのままFPGAを制御するC言語ソースに加えられる。

#### 5. 性能比較

本高速計算システムの性能比較として格子ガスオートマトン法の一つであるFHP-IIIモデルとライフゲームについて行った。

Pentium-III 700MHz上で実行したソフトウェアと性能比較をした結果、表1, 2に示す速度向上比を得ることができた。それぞれの高速計算回路の動作周波数は34.1 MHz, 35.6 MHzである。それぞれの高速計算回路における演算ユニットは、 $8 \times 16$  (FHP-III),  $32 \times 8$  (ライフゲーム)の構成をとる。

表1 FHP-IIIモデルにおける速度比較結果

	実行速度 (msec)	速度比
ソフトウェア	$376 \times 16$	1
高速計算システム	38.7	155

表2 ライフゲームにおける速度比較結果

	実行速度 (msec)	速度比
ソフトウェア	$315 \times 8$	1
高速計算システム	10.1	250

#### 6. おわりに

本論文では小規模システムを用いたセルラオートマトンのための高速計算システムについて述べた。このシステムを用いて格子ガスオートマトン法の一つであるFHP-IIIモデルとライフゲームについてPentiumIII700MHzと速度比較を行ったところ、それぞれ155倍、250倍の速度向上を得た。また、本システムで使用しているセルラオートマトン専用言語は今だ記述に対する制限が多い。今後、専用言語、コンパイラを改良していく必要がある。

#### 参考文献

- 1) Paul Shaw and Paul Cockshott and Peter Barrie. "Implementation of Lattice Gases Using FPGAs" Journal of VLSI Signal Processing, Vol 12, Number 1, PP.51-66 1996.
- 2) Bubak. M, moscinski. J and Slota. R. "FHP lattice gas on networked workstations" The 6th Joint EPS-APS International Conference on Physics Computing Lugano, Switzerland, pp. 427-430 1994.
- 3) Bubak. M, Gryniewicz. R, Moscinski. J and Palubiak. T. "FHP lattice gas on T800 transputer board" Proc. 1st Int. Conf. Parallel Processing and Applied Mathematics - PPAM'94, Czestochowa, pp.28-34 1994.
- 4) Norman. Margolus. "An FPGA architecture for DRAM-based systolic computations" Proc. FCCM '97, pp. 2-11, IEEE Computer Soc 1997.
- 5) J. Dana. Eckart. "Cellang: A Cellular Automata Programming Language" Radford University CPSC Department Technical Report, TR-92-002, 1992.