

# ハードウェアによるランダム囲碁プレイヤーの実装と評価

三好 健文<sup>†1</sup> 加藤 英樹<sup>†1,†2</sup> 竹内 郁雄<sup>†1</sup>

専用ハードウェアによってモンテカルロ法による囲碁プレイヤーのシミュレーションエンジンを実現する手法を検討する。専用ハードウェア化することで、内在する並列性を活用した高い性能を実現することが期待できる。本論文では、回路遅延の削減とメモリアクセスを行う回数の削減にし、高速かつ効率の良い処理を実現する。FPGAを対象として合成した結果、提案する回路は現実的なハードウェアリソースで実現することができ、単一のモジュールでのシミュレーション速度は670playout/秒である。

## An Implementation of Hardware Random Go Player

TAKEFUMI MIYOSHI,<sup>†1</sup> HIDEKI KATO<sup>†1,†2</sup> and IKUO TAKEUCHI<sup>†1</sup>

### 1. はじめに

モンテカルロシミュレーションとUCTアルゴリズムを用いたコンピュータ囲碁対局プレイヤーは、局面の静的な解析が不要であることから、コンピュータ囲碁の大会において好成績をおさめている。モンテカルロ法では、シミュレーション回数が2倍になるとELOレーティングが50~100増加するとの報告がある<sup>1)</sup>、そのため、シミュレーション回数を増加し、強いプレイヤーを作るための各種の高速化手法が考えられている。例えば、2008年コンピュータオリンピアドの19路碁の優勝者、Many Faces of Goのシミュレーションの速度は、5000プレイアウト/秒である<sup>2)</sup>。

本研究では、19路碁のゲーム中に存在する高い並列性を活用して、専用ハードウェアによる高速なモンテカルロシミュレーションの実行を目標とする。専用ハードウェアを用いて強いゲームプレイヤーを実現する試みには、チェスのDeepBlueや、将棋のA級指し手リーグ1号などがある。その一方で、囲碁プレイヤーを専用ハードウェアで実現している例は多くない。

本論文では、高速な囲碁シミュレーションを専用ハードウェアにより実現することを目標とする。先行研究<sup>3)</sup>では、モンテカルロ囲碁シミュレータの専用ハードウェア化について、その実装手法を検討し回路の必要とするリソース量および信号遅延を評価している。しかし、その実装方法では、メモリを用いて連情報を保持することで処理の並列性が十分に引き出せておらず、また、

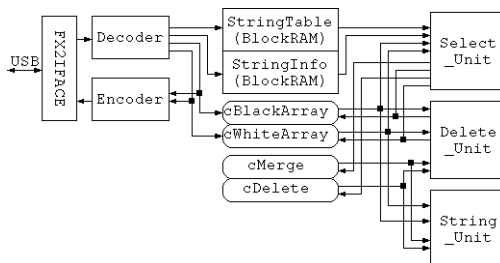


図1 専用ハードウェアの構成図

遅延の大きい回路を持つため動作周波数が低い。そのため、専用ハードウェア化により期待される性能の向上が果たされていない。本論文では、遅延の大きい回路を順序回路として分割することで動作速度の向上をはかる。またメモリへのアクセス回数を削減するためのモジュール構成を検討する。

### 2. 囲碁プレイヤーの実装

本節では、本論文で検討する囲碁プレイヤーの実装について述べる。本論文で検討する専用ハードウェアでは、ホストコンピュータから与えられたゲームの盤情報を用いてゲームを開始し、ゲームが終了するまでの手をランダムに選択することでシミュレーションする。ゲーム終了時の盤情報をホストコンピュータに転送し、勝敗の判定はホストコンピュータによって行う。

専用ハードウェア全体の構成図を図1に示す。ハードウェアユニット、Decoder、Select\_Unit、Delete\_Unit、String\_UnitおよびEncoderは、それぞれ、Request信号を外部から'H'にすることより動作を開始し、終了するまでBusy信号を'H'にする。全体の動作を管理す

<sup>†1</sup> 東京大学大学院情報理工学系研究科創造情報学専攻  
Dept. of Creative Informatics, The Univ. of Tokyo

<sup>†2</sup> 株式会社フィックスターズ  
Fixstars Corporation

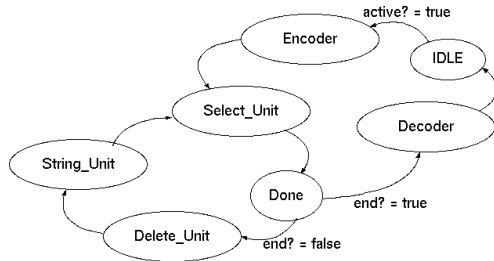


図 2 モジュールの動作を管理する状態遷移

る状態遷移を図 2 に示す。ホストコンピュータから与えられたデータを復号し内部データを生成 (Encoder) した後、各ユニットの動作をアクティブにしながら、手の選択 (Select\_Unit)、石の取り上げ (Delete\_Unit)、連の更新 (String\_Unit) によってゲームを進める。手が選択できなくなった場合終局状態とし、ゲーム盤情報を符号化し (Encoder) ホストコンピュータへ転送する。StringTable, StringInfo, cWhiteArray, cBlackArray などへの競合するデータ入力、この状態遷移に基づくマルチプレクサで排他的に取り扱う。

囲碁のゲーム盤上の情報は、19 路碁のゲーム盤上の 361 個の交点における { 黒石, 白石 } の 2 状態を、361bit 幅のレジスタ cBlackArray と cWhiteArray を用いることで、ワンホットエンコーディングで表現する。また、ゲームの進行に伴って取り上げるべき石や連を統合すべき石は、打つ石を選択する Select\_Unit で判定し、cDelete および cMerge のレジスタで保持する。これらのレジスタの値は、Delete\_Unit および String\_Unit から参照される。

検討する回路では、連を、(1) 各連に属す石の座標列のテーブル StringInfo と (2) 各座標の石が属する連識別子のテーブル StringTable で保持する。連の識別子は、 $N$  路の盤に対して  $N^2$  用意すれば十分である。すなわち 19 路の場合に、識別子の個数は 361 個であり、これを数値として表現するために必要なビット数は 9bit である。従って、StringInfo は、 $361 \times 361$  bit のテーブルであり、また、StringTable は、 $361 \times 9$  bit のテーブルとなる。これらのテーブルをレジスタで取り扱うことはリソース量の上で無理があるためにメモリを用いて保持する。

検討した回路では、各種ユニットの動作に必要なサイクルは 500 サイクル程度となった。これは、連に関する情報を保持するためにメモリを用いているため、全ての連の情報の更新に必要な 361 サイクルを含む。

### 3. 評 価

Xilinx Vertex-5 ファミリの XC5VLX50<sup>4)</sup> に実装することを仮定し、必要となるハードウェアリソース量や信号遅延について評価する。実装には VHDL を

項目	使用数	最大数	使用率
スライスレジスタ数	11,146	28,800	38%
スライス LUT 数	18,186	28,800	63%
スライス数 f	6,426	7,200	89%
メモリ使用数	306	1,728	18%

用い、合成および配置配線は、Xilinx ISE Webpack 10.1<sup>5)</sup> を用いて行った。

合成および配置配線の結果、検討した回路に必要なリソース量を表 1 に示す。ロジック回路は FPGA のリソース量の最大値に逼迫しているのに対し、メモリ使用量には、まだ余裕があることが分かる。そのため、例えば、メモリを複数使用することでメモリバンド幅を広げることで、さらに、4 から 5 倍の実行効率の向上を検討することが可能である。

また、信号遅延は、9.854n 秒であった。これは、1 ゲームに必要な手数が 300 手であると考えると検討した回路のシミュレーションの速さが、約 680 プレイアウト/秒程度であることを示す。これは、文献<sup>3)</sup> で検討した回路のシミュレーションの速さの 17 倍に相当する。

### 4. ま と め

本論文では、専用ハードウェアにモンテカルロ法に基づく囲碁プレーヤの試行部分の実現方法を検討した。検討した回路は、FPGA 上で現実的に実装することができる程度のハードウェアリソース量であり、また、その信号遅延は 9.854n 秒であった。これは、約 680 プレイアウト/秒に相当する。

今後の課題としては、メモリバンド幅を広げることによるシミュレーション速度の向上を検討する。また、スタンドアロンでの動作を実現するために、考慮していないルールの追加および勝敗判定モジュールを実装することが必要となる。さらに、手の絞り込みや UCT 探索と組み合わせを行うことで強い囲碁プレーヤの実現を目指す。

### 参 考 文 献

- 1) Dailey D. scalability studies with uct. computer-go mailing list, 2006.
- 2) 加藤英樹. [cgf.01197] 13th international computer games championship (7 日目), 10 2008.
- 3) 三好健文, 加藤英樹, 竹内郁雄. ハードウェアによるランダム囲碁プレーヤの実装と評価. ゲームプログラミングワークショップ 2008(GPW2008).
- 4) Virtex-5 マルチプラットフォーム FPGA. [http://japan.xilinx.com/products/silicon\\_solutions\\_fpgas/virtex/virtex5/index.htm](http://japan.xilinx.com/products/silicon_solutions_fpgas/virtex/virtex5/index.htm).
- 5) ISE WebPack ソフトウェア. <http://japan.xilinx.com/ise/logic.design.prod/webpack.htm>.