6N-5

FPGAを用いた将棋プロセッサの設計

堀 洋平, 丸山 勉†, 星野 力†

筑波大学大学院工学研究科, †筑波大学機能工学系

1. はじめに

将棋は人間の複雑な意思決定過程を表現可能な思考型ゲームであり，人工知能の立場から盛んに研究されている[2]．将棋プログラムは，局面をノード，指し手を枝とする木構造の探索問題である．将棋のゲーム木の探索空間は極めて広く，プログラム強化のためには高速な探索手法が不可欠である．

高速化の手法の一つとして，ASIC等のセミカスタムLSIを用いた専用プロセッサの使用が挙げられる．しかし，将棋プログラムは現在も研究が進められており，ASICのように製作した回路の変更ができないデバイスはアルゴリズムの修正に対応することができない．また，将棋のような大規模なアプリケーションを実装するためには複数のチップが必要となるが，チップ間のデータ転送時間やI/Oピンの不足等がボトルネックとなり，十分な性能を得られない可能性がある．

これらの問題を解決するため，筆者らはFPGAを用いた将棋プロセッサの開発を提案した[1]．FPGAは，ホストコンピュータの制御により回路構成の書換えが可能なLSIデバイスである．ゆえにFPGAを用いた回路は，アルゴリズムの修正に柔軟に対応することができる．また，モジュールを必要に応じて構成・削除することで回路規模を抑え，使用するチップ数を最小限にすることが可能である．本研究では，序盤・中盤・終盤等のゲームの進行状況および様々な戦略に応じてモジュールを構成・削除することで，小型かつ高性能なハードウェアプラットフォームを製作することを目指す．

現在，本将棋用プロセッサの前段階として，詰将棋の回路を作成している．詰将棋では，生成される指し手が王手またはそれを防ぐ手のみであることや，局面の評価関数が単純であることなどから，回路規模は本将棋の場合よりも小さくなる．しかし，データの更新，指し手の生成およびゲーム木の探索等の基本的なモジュールのアルゴリズムは共通であるため，詰将棋回路の設計を通して本将棋用プロセッサのアーキテクチャを模索することができる．

本論文ではまず，これまでに筆者らが提案した将棋プロセッサのアーキテクチャについて説明する．次に，これまでに完成している詰将棋のモジュールのアーキテクチャおよび性能について述べる．

A Shogi Processor with a Field Programmable Gate Array
Youhei Hori (University of Tsukuba)
Tsutomu Maruyama
Tsutomu Hoshino

2. 将棋プロセッサのアーキテクチャ

本研究では，将棋プログラムに必要とされる演算モジュールが，序盤・中盤・終盤等のゲームの進行状況および様々な戦略によって異なることに注目し，FPGAの書換えを利用しこれらのモジュールを動的に切替えることで，ハードウェアの規模を最小限に抑えることを試みる．序盤・中盤・終盤の各ステージにおける回路の構成は，図1の様になる．本章では，回路の各ステージの概要について説明する．

2.1 Opening Circuit

序盤における指し手は，主にライブラリ (データベース) を利用して生成される．ゆえに，現在の局面とライブラリとの間の高速なパターンマッチを行う回路が必要とされる．このライブラリは非常に大きくなるため一般的に外部のメモリ上に形成されていることが多いが，データ転送時間，I/Oピンの不足によるバンド幅の制限等の理由により，パターンマッチングのスピードは低下する．

FPGAを用いた場合，このライブラリはチップ内部に形成可能と考える．また，ライブラリをメモリに保存するのではなく，ロジック回路として形成することで，高速なパターンマッチングが実現可能となる．この回路は中盤以降はまったく必要なくなるので，チップの大部分を占有していたとしても，別の回路を上書きすることができる．

2.2 Middle Game Circuit

中盤は，ゲーム木を作成・探索することにより指し手を決定する．序盤の回路のOpening Libraryは，中盤の回路によって上書きされる．中盤には，局面の評価，候補手の生成を行うモジュール等が新たに必要となる．また，様々な戦法に合わせたライブラリを用意することで，通常の候補手生成回路では得られない好手が発見されると考える．将棋の戦法は数多くあり，ライブラリのサイズは極めて大きくなると思われる．しかしFPGAを用いた場合，これらの中の必要とされるライブラリのみをチップ上に構成することが可能である．

2.3 Endgame Circuit

終盤では駒の損得はほとんど考慮されず，最終的に相手の玉を詰ませることのみを考慮の対象とするため，中盤とはまったく異なる評価方法が必要となる．そこで，終盤において詰め将棋を解く回路を用意することで，将棋プログラムは大幅に強化される．また相手玉の囲いの崩し方は既に研究がされており，このデータベースを用意することで，通常の木探索では生成されない好手を発見することができると考える．

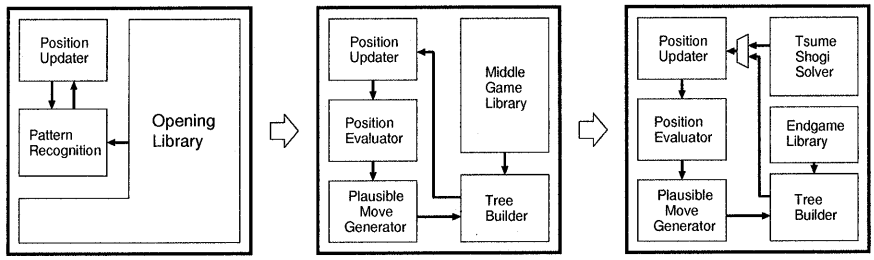

図 1　各ゲームステージの回路の概観

3. 詰将棋回路の設計

詰将棋において生成される指し手は，王手とそれを防ぐ手のみである．ゆえに指し手を生成する回路は本将棋よりもかなり小さくなり，評価関数も本将棋と比べて非常に簡素なものとなる．しかし，局面の更新・指し手の生成・ゲーム木の探索といった基本的なプログラムの流れは双方とも同様であり，詰将棋回路のアーキテクチャは本将棋の回路にも適用可能である．

本章では，詰将棋回路の概要について説明し，現在までに完成しているモジュールの性能について述べる．

3.1 回路の構成

本研究における詰将棋回路の構成を図 2に示す．Position Updaterによって更新された局面のデータは All Move Generator に入力され，すべての可能な指し手が生成される．これらの指し手の中から王手またはそれを防ぐ手のみを選択するため，本研究ではマスクを使用する．Check Mask Generator は王手のみを抽出するマスクを，Defense Mask Generator は王手を防ぐ手のみを抽出するマスクを生成する．2 つの Mask Generator は All Move Generator と並列に動作するため，効率の良い選択が可能である．

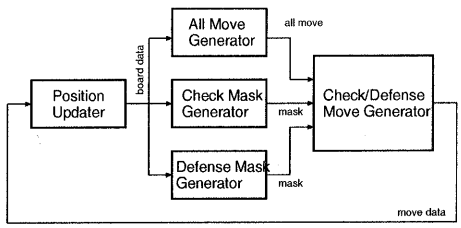

図 2　詰将棋回路の構成

3.2 性　　能

詰将棋回路の各モジュールの動作速度を，表 1に示す．動作速度は，回路の設計ツールである MAX+PLUS II の中のシミュレーション・ツールによって計算された．シミュレーションの結果，局面を与えてから指し手を生成するまでに 48clock かかったため，計算時間は 2.15[μsec] であった．また，筆者らがベンチマークのため作成したソフトウェアは，同様の結果を得るまでに 18.9[μsec] の時間を要した．この結果，詰将棋回路はソフトウェアと比較し約 8.8 倍の性能を得たことになる (表 2)．

表 1　各モジュールの動作速度

モジュール	動作周波数 [MHz]
Position Updater	80.06
All Move Generator	54.87
Check Mask Generator	90.07
Check/Defense Move Generator	22.32

表 2　詰将棋の計算時間の比較

	動作周波数 [MHz]	計算時間 [μsec]	性能比
Software	700	18.9	1
FPGA	22.32	2.15	8.8

しかし，表 1によると，Check/Defense Move Generator の動作速度が極端に遅いことがわかる．この速度は，より深いパイプラインを施すことにより改善が可能であることがわかっており，性能はさらに向上すると思われる．

4. おわりに

本研究では，将棋プログラムの高速化を実現するため，FPGA を利用したハードウェアプラットフォームを提案した．将棋プログラムに必要な演算モジュールが，ゲームの進行状況や戦略によって異なることに注目し，FPGA を用いて動的に回路を切替え，複数のモジュールを 1 チップ上で実現するアーキテクチャを提案した．

現在，本将棋の前段階として詰将棋の回路を設計している．詰将棋の回路は，ソフトウェアと比較し約 9 倍の性能が得られると見積もられている．アーキテクチャの改善により，この性能はさらに向上すると考える．

参考文献

1) Y.Hori, M.Seki, R.Grimbergen, T.Maruyama and T.Hoshino, “A Shogi Processor with a Field Programmable Gate Array”, *Computers and Games*, Hamamatsu, Japan, 2000.
2) 松原 仁, “将棋とコンピュータ”, 共立出版株式会社, 1994.