

[4E-7] マルチコンテキスト FPGA におけるコンテキスト分割問題とその一解法

本多 亮†

北道 淳司††

船曳 信生†

東野 輝夫†

† 大阪大学 大学院基礎工学研究科

†† 大阪大学サイバーメディアセンター

1 まえがき

動的再構成可能 FPGA[3] (以下, DRFPGA) はアプリケーション動作中に FPGA 上の論理の再構成を行う機能を有する. DRFPGA の一種であるマルチコンテキスト FPGA[1][4] は, 回路素子毎に回路論理を記憶する SRAM を複数持ち, それらをマルチプレクサで切替ることで, FPGA 上の論理構成を高速に変更することができる. 従来型 FPGA に比べてマルチコンテキスト FPGA では, 回路入力からの分割, 配置, 配線の 3 工程に加えて, 回路入力を FPGA の容量を上限とする部分回路ごとに分割する問題を考慮する必要がある. 本研究では, この問題を解くアルゴリズムの提案を行う.

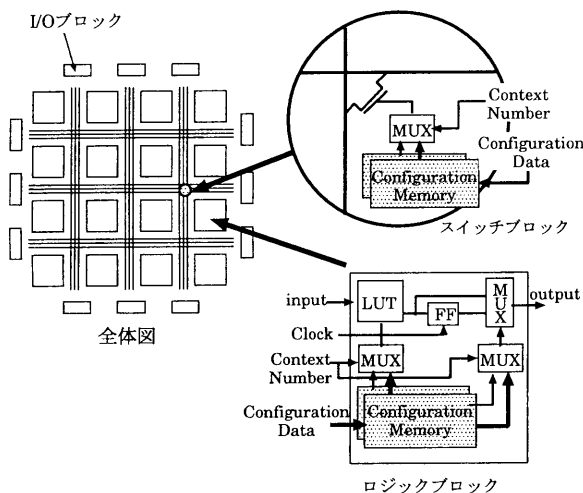


図 1: マルチコンテキスト FPGA の概略

2 コンテキスト分割問題

2.1 マルチコンテキスト FPGA

マルチコンテキスト FPGA[2] の概略を図 1 に示す. マルチコンテキスト FPGA は, ロジックブロック, スイッチブロックの各設定項目に対して設定用記憶 (Configuration Memory) を複数持ち, マルチプレクサ (MUX) によって設定用記憶を選択することにより, 回路の論理を決定する. 本論文では各 MUX には同じ制御信号 (Context Number) を与え, 全設定用記憶を瞬時に変更するものを対象とする. 同時に選択される設定用記憶の集合をコンテキストと呼ぶ. また, MUX

の制御信号で指定されない設定用記憶に対して動作と並行して外部から回路設定情報 (Configuration Data) を書き換えることができる (図 2).

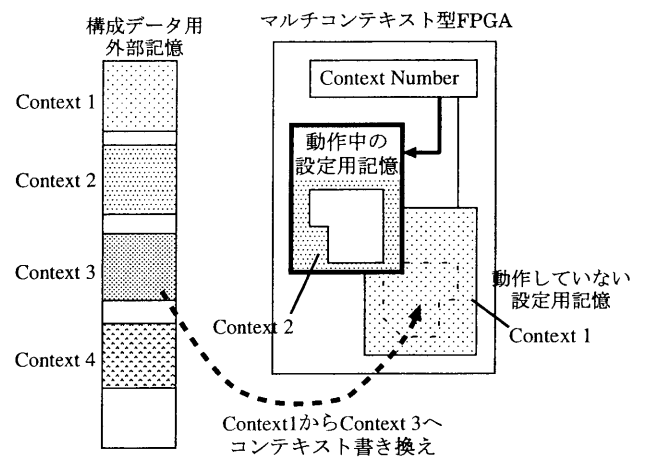


図 2: 設定情報の書き換え

2.2 コンテキスト分割問題の定式化

本論文では実装するアプリケーションをコンテキストに分割するコンテキスト分割問題に着目する.

コンテキスト分割問題の入力は拡張有限状態機械 (EFSM) および FPGA のサイズである. EFSM は回路の演算結果を保持するデータレジスタ群と有限状態機械から構成される. 各状態では, 各レジスタ転送に用いる演算回路や, 複数の遷移が出射する場合は遷移条件判定回路が割り付けられており, それらはすべてロジックブロック単位の回路に分割されているとする. 出力は有限状態機械の各状態における演算回路をどのコンテキストで実現するか決定し, その結果とする. このときデータレジスタ群は各コンテキストに常駐するものとする. 制約条件は分割された各コンテキストにおいて, データレジスタと状態の演算回路の総和が FPGA のサイズの大きさを越えないことである. 目的条件はコンテキスト変更時のオーバーヘッドを最小化することである. コンテキストの切替えは瞬時に行えるが, アプリケーション実行中, 次に必要なコンテキストが設定用記憶群に保持されていない場合, 外部から必要な論理を書き込む必要があり, 書き込み終了まで動作を停止 (ストール) させる必要がある. このストール時間がオーバーヘッドである. 選択されていない設定用記憶に対して動作と並行して論理の書き込みを行えるので, 次の実行コンテキストを早い時点で判定できればオーバーヘッドを軽減できる. 1 コンテキスト分の論理書き込み時間を一定とすると, 次の実行コン

"A Proposal Algorithm for Context Partitioning Problem of Multi-Context FPGAs"
Ryo HONDA, Junji KITAMICHI, Nobuo FUNABIKI, Teruo HIGASHINO
Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka, 560-8531, Japan

テキスト判明からコンテキスト切替えまでの時間の最大化を目的条件とみなせる。(図3)

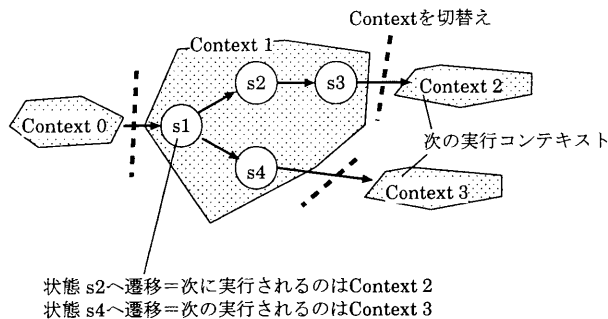


図 3: コンテキスト内での遷移の分岐

3 コンテキスト分割アルゴリズム

コンテキスト分割アルゴリズムに対して EFSM の各分岐における遷移確率が与えられているとする。提案アルゴリズムは二段階構成とし、第一段階で初期解を生成、第二段階で初期解の改良を行う。

第一段階では Greedy アルゴリズムを用い、入力された EFSM からコンテキスト分割の初期解を生成する。EFSM の初期状態から深さ優先探索を行い、分岐のある状態では、遷移確率の高いほうへ走査する。これは実行頻度の高い遷移経路上の状態は同じコンテキスト内に割り付けることが目的関数を良くすると考えたためである。走査された順に状態を FPGA の大きさの許す限り同じコンテキストに割り付け、FPGA の大きさを上回った時は、新たなコンテキストを用意し、そこに状態の割り付けを行う。この手順を有限状態機械の全状態に対して行い、初期解とする。

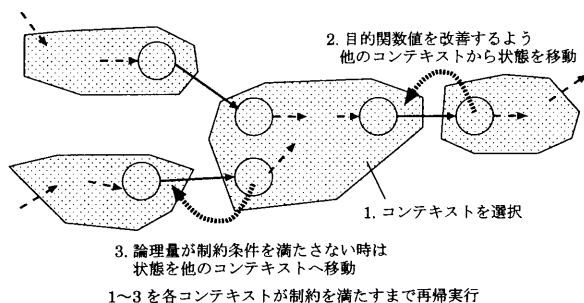


図 4: 第二段階での再帰実行の流れ

第二段階では、初期解の改良を行う。状態遷移の分岐からコンテキスト切替えまでの長さが最小のコンテキストを選択し、選択されたコンテキストへ他のコンテキストから状態を移動させる。状態の移動によりコンテキスト内に割り付けられている論理量が変化する。FPGA の大きさを上回った場合は、そのコンテキストから他のコンテキストへ状態をさらに移動させる。状態を移動させたコンテキストにおいても再帰的に制約条件の判定、状態の移動を行う(図4)。試行が無ループに陥ってしまう可能性があるので一定回移動を行った状態集合には、以降その試行中には状態の移動

を行わないものとする。すべてのコンテキストで一定回の移動が行われた時、制約条件を満たさないコンテキストがある場合には新たなコンテキストを設け、そこに状態を移動させることとする。アルゴリズムの第二段階では上記の試行を一定回繰返し、その間に解の改善が見られなかった時に繰返しを終了し、分割結果と目的関数値を出力する。

4 実行結果

アルゴリズムの第2段階に2つのコンテキスト内の状態を入れかえ Simulated Annealing (SA) により受理するかどうか決定する方法を用いた手法および提案アルゴリズムに対しランダム例題を与え、目的関数値、アルゴリズム実行時間の比較評価を行う。

実験1 では例題回路の状態数を200に固定し、分岐確率の比率と各状態の分岐数を変化させたときの各出力値について実行した。分岐を持つ状態数を状態数全体の5%, 10%, 1状態における分岐数の上限を4, 8と変化させ、10試行の平均を求めた。提案アルゴリズムはSAを利用したアルゴリズムの約100分の1の実行時間で解を出力した。目的関数値は両者とも初期解より約20%程度改善された。状態遷移の分岐が増えるにつれ解の目的関数値が小さくなった。これは状態間の接続関係が密になり頻繁にコンテキストを切替えているためと考えられる。また状態遷移の分岐数の変化による実行時間の変化は特に見られなかった。

実験2 では例題回路の分岐を持つ状態を状態数全体の5%, 1状態における分岐数の上限を4と固定し、状態数を変化させた。状態100, 200, 500と10試行の平均を求めた。同規模の回路において両者を比較した場合、提案アルゴリズムはSAを利用したアルゴリズムの100分の1程度の実行時間で解を出力した。また、状態数が増加するにつれて両者とも同様の比率でアルゴリズム実行時間が増加した。目的関数値は両者とも初期解より約20%程度改善された。

提案アルゴリズムにおいてコンテキスト数を増やす処理は実験1, 実験2ともに行わなかった。

5 まとめ

マルチコンテキスト FPGA に対するコンテキスト分割アルゴリズムの提案を行った。提案アルゴリズムは Greedy アルゴリズムで得られた初期解を20%程度改善した解を高速(状態数200の例題に対し約3秒)に求められる。またSAを利用したアルゴリズムの約100分の1の実行時間で同程度の解を出力した。

参考文献

- [1] Chang, D. and Marek-Sadowska, M.: Partitioning Sequential Circuits on Dynamically Reconfigurable FPGAs, *IEEE Transactions on Computers*, Vol. 48, No. 6, pp. 565-578 (1999).
- [2] 糸将之, 北道淳司, 船曳信生: 動的再構成可能FPGAの設計とそれへの並列アルゴリズムの実装, 情報研報DA, Vol. 98, No. 43, pp. 1-8 (1998).
- [3] 末吉敏則: リコンフィギュラブルロジック, 電子情報通信学会誌, Vol. 81, No. 11, pp. 1100-1106 (1998).
- [4] 凌曉萍, 天野英晴: データ駆動型制御機構付きMPLDを用いた並列処理マシンWASMIIの仮想化, 情報処理学会論文誌, Vol. 35, No. 4, pp. 646-657 (1994).