

マルチタスク可能なパイプライン処理型動的再構成回路の 資源管理部の設計

小椋清孝[†], 黒田光紀[†], 山本哲哉[‡], 森下賢幸[†], 伊藤信之[†]

[†]岡山県立大学 情報工学部

[‡]岡山県立大学大学院 情報系工学研究科

はじめに

開発中のパイプライン処理型動的再構成回路 DRoMPA2.0 の演算資源を効率的に使用方法のひとつとして、複数の処理を同時に実行可能とするマルチタスク対応にすることを考え、現在マルチタスク管理部の設計を行っている。

本発表では、そのうちの各タスクへの資源(演算セル)割当を行う資源管理部の設計について報告をする。このような資源管理は、ホストプロセッサなどでソフトウェア的に行う方法も考えられるが、今回はハードウェアで処理することにより上位での管理の負担の削減を試みた。

動的再構成回路 DRoMPA2.0

メディア処理向け動的再構成回路アーキテクチャ DRoMPA2.0 (Dynamic Reconfiguration oriented Media Processing Architecture)は、加減算ユニット(ADDSUB unit)、乗算ユニット(MULT unit)、カウンタユニット(COUNTER unit)、レジスタユニット(REGISTER unit)等の複数種類(ヘテロ)の演算ユニットを用いてパイプライン処理回路を構成し、これを再構成しながら処理を行っていくタイプの動的再構成回路のアーキテクチャである。各演算ユニットは 32bit 精度であり、32bit データバスおよび制御信号系の 1bit データバスがそれぞれ隣接演算ユニットに接続された構成となっている。再構成のための構成データは、各演算ユニット数個ずつを 1 セットにした UnitGroup(UG)を単位として管理される。プロセッサとしては、この UG を数個と再構成時の中間データ保持用のメモリユニット数個、外部メモリを含めた I/O 部および管理回路で構成される。

これまでは、4 個の UG を搭載したプロセッサ

Design of Resource Management Circuit for Dynamically Reconfigurable Multitasking Circuit with Pipeline Processing

[†]Kiyotaka Komoku, Mitsuki Kuroda, Takayuki Morishita and Nobuyuki Itoh, Okayama Prefectural University, Faculty of Computer Science and System Engineering

[‡]Tetsuya Yamamoto, Okayama Prefectural University Graduate School of Computer Science and Systems Engineering

構成について FPGA に実装を行いアーキテクチャの検討を行ってきた。アプリケーションは音声や動画の符号化・復号化処理(MPEG 処理)であり、これを動的再構成で実行するものである。近年の FPGA の大規模化により、さらに多くの UG の搭載が可能となってきた。しかしながら、一つのアプリケーションを複数の再構成タスクに分解する際には、均等な大きさの処理に分割することが難しいため、多数の UG を使い切ることができずに UG の利用効率が低くなってしまいう可能性がある。そこで、マルチタスク化して同時に複数のアプリケーションを実行可能とすることで、多数の UG を効率よく利用することが可能になると考えた。

このようなマルチタスク/マルチスレッド型の動的再構成回路については以前から多くの研究が行なわれている。[1]ではこのような動的再構成回路のサーベイとともにいくつかの研究課題を提示している。その中にはスレッド(タスク)の選択や配置配線方法、構成データのローディング時間の隠ぺいなどといったものがある。

我々は、できるだけシンプルなアプローチでこれらの問題に取り組むこととした。各種の制御は基本的に HW で実装することを目標とした。

マルチタスク管理の方針

マルチタスク処理においては、実行中のアプリケーションの管理と演算資源(UG)の管理が必要となる。今回の設計においては、これらの二つの管理部分を完全に分離して個別の回路として設計した(図1)。これにより、UG 数が異なる構成でもアプリケーション管理部は再設計不要とすることができる。

アプリケーション管理部は、外部からのアプリケーション実行要求の際に各種情報をタスク管理テーブルに記録し、演算資源管理部にタスク実行のリクエストを送る役目を持つ。その際、空きの演算資源がないなどの理由でリクエストが断られる場合もあるため、そのタスクが実行中か待機中かといった情報も管理する。

資源管理部は、アプリケーション管理部からのタスク実行リクエストに応じて、実行可能であれば演算資源である UG の割り当てを行い、構成データをそれらの UG にローディングする。また、タスクの実行が終了した場合には、アプリケーション管理部へタスク終了信号を送る役割も持つ。

その他、外部 I/O や中間データ保持用メモリブロックのタスクへの割当管理などが必要となる。これらについては現在、詳細を検討中である。本稿では、資源管理部の詳細について述べる。

資源管理部の設計

資源管理を行うに当たり、多数の UG をどのように管理するかが非常に重要となる。N×N のような 2 次元配置にしてしまうと、タスクの割り当ての際に、タスクをどの位置に割り当てるか、タスク自体をどのような形で割り当てるか(使用 UG 数が 2 以上の場合)等の冗長性が発生し、割り当てアルゴリズムの複雑化やタスク構成情報の多重化といった問題が発生する。

そこで、2×N とした形で一列を 2UG として横に並べた、一次元配置で管理することにした。これにより、割当位置決定は非常に容易となり、タスク自体の形も一本化されるので構成情報の多重化は不要となる。ただし、使用 UG 数が奇数個の場合、未使用 UG が発生することになる。今回の設計では 2×8 の 16UG を持つプロセッサ構成を管理対象とした。

資源管理部で管理する情報は、現在の UG 列使用状態と、各 UG 列での実行中のタスク番号である。使用状態は各 UG 列の使用/未使用を 1/0 の 1bit で、つまり全体で 8bit のレジスタで管理する。タスク番号は各列あたり 8bit である。タスクの実行リクエストは、リクエスト信号とともにタスク番号と必要 UG 数がアプリケーション管理部から送られてくる。リクエストがあった場合の実行可/不可判定は、必要 UG 数と UG 列使用状態レジスタの内容とから決定される。

各 UG 列間の接続について、隣接接続のみでは、空き UG 列が飛び飛びになって空き数は十分であるのに割り当てができないという状況が起こりうる。そこで、非連続 UG 列間接続を可能とするようにバスの選択回路を付加した。これにより離れた UG 列通しを隣接状態として扱えるようになった。資源管理回路としては、割当可能判定後にバス選択回路への選択信号を発生する機能を追加した。

タスク割り当て後、プロセッサ上の構成情報

メモリ(すべてのタスクの再構成用構成情報が収められている)から割り当てた各 UG に構成情報をローディングする。タスク番号から先頭メモリアドレスが決定され、一定時間ローディングが行われる。ローディング中は新しいタスクの割り当てが行われないよう、アプリケーション回路からのタスクリクエストは無視される。

実装結果

本回路を Xilinx XC5VLX110 を実装ターゲットとして設計を行った。設計ソフトウェアは Xilinx ISE Design Suite 14.6、シミュレーションは同 ISim を用いた。論理シミュレーションの結果、所望の動作を確認した。また、合成の結果、使用 LUT 数は 1511、使用 Slice Register 数は 795 となった。

まとめと今後の課題

本稿では動的再構成回路 DRoMPA2.0 のマルチタスク管理のための資源管理部の設計結果について述べた。本稿では詳細を述べなかったがアプリケーション管理部も設計を行っており、これらを統合して検証を行う予定である。また、外部 I/O やメモリブロックの割り当て方法の検討と回路化も進めることとしている。

参考文献

- [1] P. G. Zaykov, G. K. Kuzmanov and G. N. Gaydadjiev, "Reconfigurable multi-threading architectures: A survey", Proceedings of the 9th International Workshop on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS '09), pp. 263-274, 2009

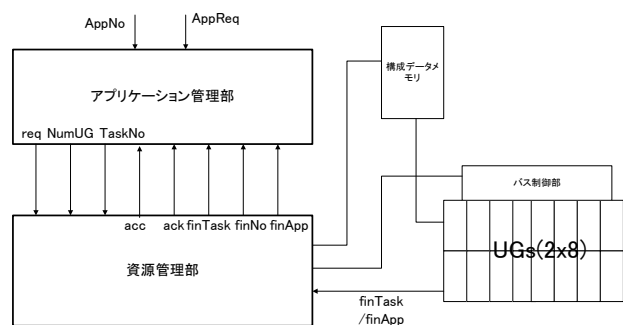


図1 構成概要図