

## Verilog-HDLによる並列キュープロセッサのデザイン Design of a Parallel Queue Processor in Verilog-HDL

三好 崇之<sup>†</sup> Ben A. Abderazek<sup>†</sup> 繁田 聡一<sup>†</sup> 吉永 努<sup>†</sup> 曾和 将容<sup>†</sup>  
Takayuki Miyoshi Ben A. Abderazek Soichi Shigeta Tsutomu Yoshinaga Masahiro Sowa

### 1. はじめに

現在、我々は中間結果の格納用レジスタにFIFO (First In First Out) アクセス方式を用いるキュープロセッサ [1] の研究を行っている。キュープロセッサの特性として、構文木の幅優先探索によって生成されるキュープロセッサ用の命令列は、高い命令レベル並列性を含むことが期待できる。また、キューレジスタの使用法の違いによって、生産消費順序遵守型 (以下 生産消費型)、生産順序遵守型 (以下 生産型)、消費順序遵守型キュープロセッサという3種類のキュープロセッサを提案している [2]。この中で、生産型の並列キュープロセッサが命令数とステップ数の点で優れていることが示されている。本稿では、生産型並列キュープロセッサを Verilog-HDL を用いてデザインした。そして、シミュレーションによるプロセッサの動作確認を行った。

### 2. 生産型キュープロセッサ

#### 2.1 アーキテクチャ

生産型キュープロセッサアーキテクチャは、生産消費型キュープロセッサにおけるデータの読み出し (消費) 順序に柔軟性を持たせ、キューレジスタの先頭 (QH) 以外の読み出しを可能にしたもので、データの書き込み (生産) 順序のみを遵守するキュープロセッサである。キューレジスタの先頭以外のデータの読み出しを行うために、先頭から参照データまでの距離を命令のオペランドとして記述する。生産型において、参照先が (先頭 + 1) の時は生産消費型と同じであるので、生産型は生産消費型をサブセットに含む。また、生産型では過去に生産されたデータの参照を可能にするため、消費されたデータをキューレジスタに残している。

図1に例として、生産型キュープロセッサにおいて  $(a+b) * (c-a)$  を計算する処理の流れを示す。まず load a でキューレジスタの末尾 (QT) に a を書き込む。load b, load c も同様にデータをキューレジスタの末尾に書き込む。次に add 0, 0 では、キューレジスタの先頭から a と b を読み出して、その演算結果をキューレジスタの末尾に書き込む。sub 1, -2 では、キューレジスタの先頭と (先頭 - 2) の c と a を読み出して、演算結果をキューレジスタの末尾に書き込む。mul 0, 0 では、先頭から  $(a+b)$  と  $(c-a)$  を読み出して、演算結果を末尾に書き込む。すると、計算結果の  $(a+b) * (c-a)$  が得られる。

#### 2.2 命令セット

生産型キュープロセッサの命令セットは、メモリアクセス命令、演算命令、分岐命令、レジスタ操作・制御命令から構成されている。レジスタ操作・制御命令には、キューレジスタや汎用レジスタへの即値の代入やキューレジスタと汎用レジスタ間のデータ転送命令も含まれて

<sup>†</sup>電気通信大学 大学院情報システム学研究所

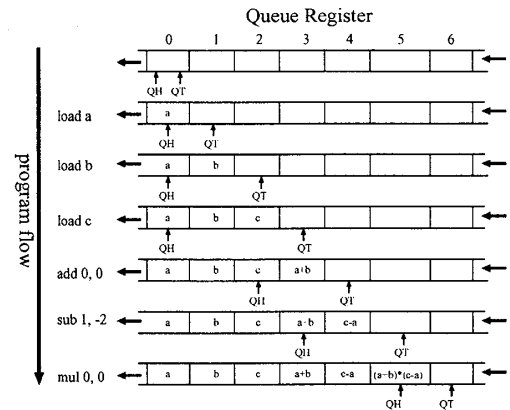


図1: 生産型キュープロセッサにおける計算の過程

いる。全ての命令は、オペコード部とオペランド部から成る2バイト固定長である。オペランド部には、命令のタイプ (生産消費型か生産型か) と、生産型の場合は参照先までのオフセットを指定する。インデックスレジスタやアドレスレジスタ相対でアドレス計算を行うメモリアクセス命令や分岐命令では、オフセット値は通常オペランド部を使用するが、拡張可能のようにデザインしてある。固定長命令セットを使用することで、可変長命令セットを使用する場合 [3] に比べ、プロセッサの内部構造を簡単化できる。

### 3. 生産型並列キュープロセッサ

キュープロセッサ用の命令列作成は、構文木の幅優先探索によって行う。生成された命令列は、前後の命令間にデータ依存関係があることが少なく、高い命令レベル並列性を含むことが期待できる。例えば、図1の load a ~ load c の3命令は互いにデータ依存関係がない。また、add 0, 0 と sub 1, -2 命令にも互いにデータ依存関係がない。そのため、これらの複数命令は並列に実行可能である。

そこで、生産型キュープロセッサにおいて、キューレジスタに対して複数の命令によるデータの同時書き込みと読み出しを可能にすることで、命令の並列処理を実現する。これが生産型並列キュープロセッサである。

#### 3.1 キュー計算ユニット

複数の命令によるキューレジスタへの同時書き込みと読み出しを可能にするためには、各命令が書き込みと読み出しを行うキューレジスタの位置情報を付加する必要がある。この処理を行うためにキュー計算ユニットをパイプラインに追加した。キュー計算ユニットは、各命令についてキューレジスタの位置情報を計算して付加するユニットで、並列キュープロセッサに特有のユニットで

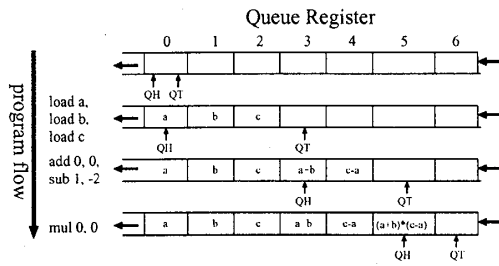


図 2: 生産型並列キュープロセッサにおける計算の過程

ある。キュー計算ユニットでは、命令がキューレジスタに書き込んだり読み出したりするデータ数をカウントしていくことで、各命令が書き込みと読み出しを行うべきキューレジスタの位置を計算する。実行ユニットで実際に演算処理を行う時には、この位置情報が示すキューレジスタにアクセスする。このように並列キュープロセッサは、各命令に対するキューレジスタの位置情報の計算と付加をプロセッサ内部で動的に行っている。

図 2 に、図 1 の例を並列処理した場合の流れを示す。キュー計算ユニットでは、load a 命令にキューレジスタの 0 番の位置に a を書き込むように情報を付ける。load b, load c 命令に対しても、それぞれキューレジスタの 1 番, 2 番に書き込むように情報を付ける。add 命令には、キューレジスタの 0 番と 1 番からデータを読み出し、3 番に書き込むように情報を付ける。sub, mul 命令に対しても、それぞれ 2 番と 0 番から読み出して 4 番へ書き込むように、3 番と 4 番から読み出して 5 番へ書き込むように情報を付ける。実行ユニットでは、3 つの load 命令を 1 ステップ目で並列処理し、add と sub 命令を 2 ステップ目に並列処理する。図 1 では 6 ステップかかっていた処理を 3 ステップで実行できる。

3.2 パイプライン構成

本研究の生産型並列キュープロセッサは、図 3 に示すような 7 段のパイプライン構造を持つ。フェッチユニットでは、サイクル毎に 4 命令フェッチする。デコードユニットでは、命令デコードを行い、各命令がキューレジスタに対して行うデータの書き込みと読み出し数の情報を付加する。この情報を元に、キュー計算ユニットは各命令が使用するキューレジスタの位置情報の計算を行う。バリア・キュー制御ユニットでは、分岐命令によるハザードの処理、汎用レジスタを使用する命令間のデータ従属関係のチェックを行う。そして必要に応じて、パイプラインのストールを行う。発行ユニットでは、キューレジスタにおけるデータ従属関係のチェックを行い、従属関

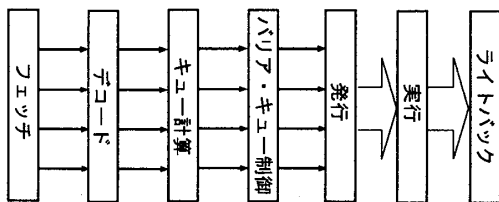


図 3: 生産型並列キュープロセッサのパイプライン構成

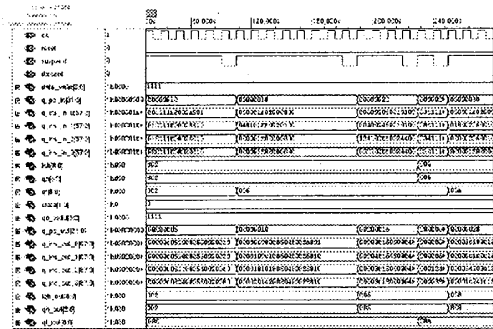


図 4: シミュレーション結果

係が解消された命令を発行する。実行ユニットは、ロード・ストアユニット、ALU、分岐ユニット、レジスタ操作・制御ユニットから構成されている。

レジスタは、32 ビット幅のキューレジスタを 256 本、汎用レジスタを 16 本、また、8 ビット幅の命令キャッシュ、データキャッシュを用意している。汎用レジスタはランダムアクセス可能なレジスタで、メモリアクセス命令や分岐命令がインデックスレジスタやアドレスレジスタとして使用する。

4. シミュレーション

生産型並列キュープロセッサを Verilog-HDL を用いて RTL (Register Transfer Level) 設計を行った。Verilog-HDL コンパイラには、Cadence 社の Verilog-XL を使用した。そして、設計したプロセッサで、メモリアクセス命令を含むプログラム、条件分岐・無条件分岐を含むループ演算プログラムなどを実行し、正しく動作することを確認した。図 4 に、シミュレーションにおける並列キュープロセッサの動作結果の一部を示す。

5. おわりに

生産型並列キュープロセッサのデザインを Verilog-HDL を用いて行った。シミュレーションによりプロセッサがデザインどおりに動作することを確認した。今後は、今回設計した生産型並列キュープロセッサをベースに、SMT (Simultaneous Multi-Threading) 実行機能を付加したマルチスレッド並列キュープロセッサへ発展させていく予定である。

参考文献

- [1] S. Okamoto, H. Suzuki, A. Maeda and M. Sowa : "Design of a Superscalar Processor Based on Queue Machine Computation Model", *Proc. of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pp.151-154 (1999).
- [2] H. Kutluk, B. A. Abderazek, 繁田聡一, 吉永努, 曾和将容: "並列キュー計算モデルの基本特性評価", 信学技報, (2004.7 発表予定).
- [3] 菊池遊, 吉永努, 曾和将容: "キュー計算モデルを用いた並列プロセッサの設計" 電子情報通信学会 情報・システムソサイエティ大会, p.42, (2001).