

生産順序型並列キュープロセッサのための 効率的な命令発行のメカニズム

木内 和之、ABDERAZEK Ben、繁田 聡一、
曾和 将容、吉永 努

電気通信大学 大学院情報システム研究科

1 はじめに

プロセッサアーキテクチャにおいて、レジスタはデータの一時格納場所として用いられる。しかし、レジスタはランダムアクセスなのでデータごとにその位置を示す必要があった。これに対して、本研究室で研究されているキュープロセッサではランダムアクセスレジスタの代わりにFIFO(先入れ先出し)機構をもつキューレジスタを用いる。そうすることによって、データの並びに規則性が生じ、その位置を明示的に示す必要がなくなる。結果として、ハードウェアが簡単化され、消費電力も抑えられるなどの効果が期待される。

命令の実行後に算出されたデータをその順番どおりにキューレジスタに格納し、その並びに従ってデータを使用していくものを生産順序型キュープロセッサと呼ぶ。さらに、この生産順序型キュープロセッサで依存関係のない命令を実行する機構を備えたものを生産順序型並列キュープロセッサ(PQPpf)と呼ぶ[1][3]。

PQPpfは以下の6つのユニットから構成される。

- 1) フェッチユニット(FU)
- 2) デコードユニット(DU)
- 3) キュー計算ユニット(QCU)
- 4) バリアキューユニット(BQU)
- 5) 命令発行ユニット(IU)
- 6) 実行ユニット(EU)

このユニットのそれぞれがパイプラインのステージに対応している。

本稿では、このパイプラインステージのうち、命令発行ユニット(IU)における命令発行のメカニズムについて述べる。

2 命令発行のメカニズム

IUが担当する働きは、次の3つである。

- 1) 各命令にそれぞれに対応したプログラムカウンタ(PC)の値を付加する。
- 2) 直前のBQUで各命令にセットされたフラグに従って、命令間の依存関係を取り除く
- 3) 各命令に必要なデータ、実行ユニットのチェックをして、発行可能条件を満たした命令から発行する。

このとき、BQUで命令にセットされるフラグは、各命令間に存在する依存関係の種類とその有無を示すためのものである。依存関係にはロード・ストア命令間のデータ依存関係や分岐・ジャンプによる制御依存関係がある。IUは依存関係のある命令が並列に発行されないように制御する。一連の動作を表すフローチャートを図1に示す。

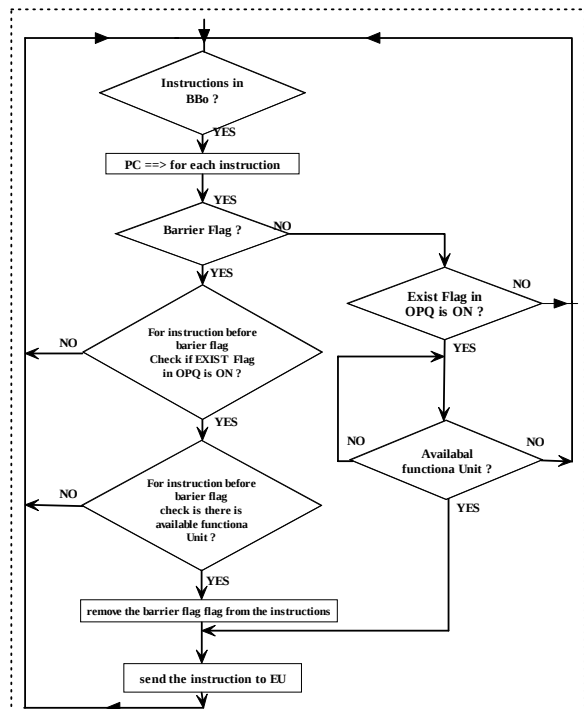


図1. IUのフローチャート

3 バリアバッファ

An Efficient Instruction Issue Mechanism for a
Produced-order Parallel Queue Processor
Kazuyuki Kiuchi, Ben Abderazek, Soichi Shigeta,
Masahiro Sowa, and Tsutomu Yoshinaga
The Graduate School of Information Systems,
University of Electro-Communications

I UとB Q Uバリアバッファ(B B)が存在する。これはB Q Uで処理された命令の一時格納に用いられ、B B i (バリアバッファ-i n)とB B o (バリアバッファ-o u t)の二段構成となっている。最初に命令が格納される場所がB B i である。B B i に命令が格納されたときに、B B o が空の状態であればB B i の命令はB B o へと移動する。逆にB B o が空でない場合、命令はB B i に格納されたままである。しかし、そのままでは次のサイクルでB Q Uから送られてくる命令によって上書きされてしまうので、B B i に命令が格納されている間は、上流のユニットにストール信号を送り、動作を一時停止させることで命令の上書きを防ぐ。また、P Q P p f は基本的に 4 命令同時処理であるので、B B に格納されるときも 4 命令同時である。

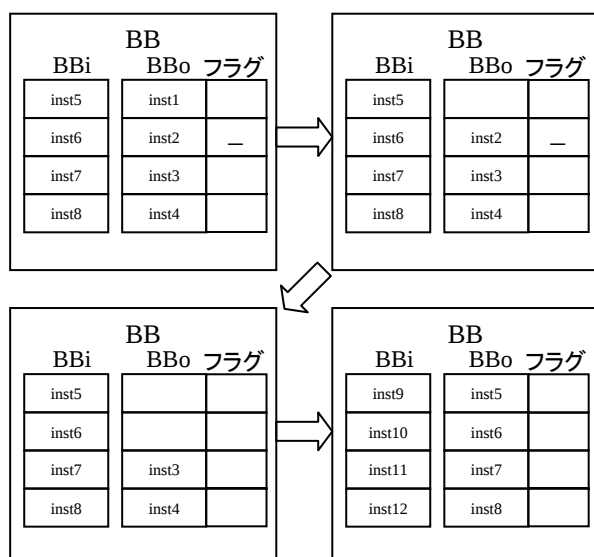


図2. B Bにおける命令の遷移の様子

4 命令発行ユニットの動作

命令発行ユニット(I U)は図1に示したフローチャートに従って動作する。B B o に命令が格納されているとき、I Uはまず、各命令に対応したプログラムカウンタ(P C)の値を付加する。次に、格納されている命令の中にB Q Uがフラグを立てた命令があるかどうかをチェックする。このとき、フラグを持つ命令がなければ、4 命令の全てについて、それぞれが必要とするデータがキューレジスタ上に存在するか否かをチェックする。キューレジスタのそれぞれの番地に対応したフラグが存在するので、それらをチェックしてデータの有無を確かめる。次に、加算命令ならA L Uというように命令に対応して

必要となる演算ユニットの使用状況を確認する。データと演算ユニットが共に利用可能であれば命令はE Uへ発行される。もし、必要とするデータが存在しない、もしくは演算ユニットが利用できない場合は、それらが利用可能な状況になるまで確認動作を繰り返す[2]。

B B o 内の命令にB Q Uがフラグをたてた命令が存在したときは、そのフラグが立った命令より小さいP Cの値となる命令に対して、発行可能条件のチェックを行う。図2に示す例では、まず inst1 が発行され、次にバリアを設定すべきことを示すフラグが立っている inst2 が発行され、最後に inst3 と inst4 が同時に発行されている。

5 動作検証

I Uを Verilog-HDL で記述し、synopsis Verilog-XL simulation tool を用いてR T Lレベルでのシミュレーションを行った。その結果、シミュレーションは成功し、本稿で述べたメカニズムに沿った動作の確認ができた。

6 まとめ

本稿では、P Q P p fの一構成要素であるI Uの動作メカニズムを解説、ハードウェアの設計を行った。シミュレーションを行った結果、動作確認もとれた。引き続きP Q P p fの実装を行っていく予定である。

7 参考文献

- [1]曾和 将容:”並列キュープロセッサ原理”(2003)
- [2]B. A. Abderazek, M. Sarem, M. Sowa, “Dynamic Fast Issue (DFI) Mechanism for Dynamic scheduled Processors”, IEICE Vol.E83-A No.12 pp.2417-2425 (2000)
- [3]B. A. Abderazek, 繁田 聡一, K. Nikolova, 吉永 努, 曾和 将容:”並列キュープロセッサの基本設計”, 電子情報通信学会, 技術研究報告(CPSY), Vol.102 No. 478 pp.55-60(2002)