

再構成可能なプロセッサにおける高速化への一手法

白本 長遊†

浜辺 直輝†

柴田 浩†

†大阪工業大学 情報科学部

概要: パイプライン方式のCPUに再構成可能デバイスを搭載する場合のアーキテクチャ方式を提案する。本稿では、サブルーチンをハードウェアに置換えることを目的として、サブルーチン呼出によって動的に再構成されたハードウェアに処理を移行する手法を考えた。応用例として、パリティ計算とバイトサーチのシミュレーションを行った。

1. はじめに

プロセッサ高速化の手段として、(1)高クロック化、(2)並列化(パイプライン、スーパースカラ)、(3)専用ハードの利用などがある。(1)(2)の手段には汎用性があるが、(3)の方法で高速化したプロセッサは特定の用途でしかその高速性を発揮できない。これを解決できるものとして現在注目されているのがダイナミック・リコンフィギャブル・デバイスである。

ダイナミック・リコンフィギャブル・デバイスは、FPGAなどの再構成可能なデバイスの進化したもので、実行時にその構成を変更することができる。また、複数の構成情報を記憶しそれらを瞬時に切替えられるものも存在する[1]。

本稿ではこれらのデバイスの存在をふまえて、CPUに動的再構成可能なデバイスを内蔵し、その上に構成されたハードウェアをソフトウェアからサブルーチンとして実行できる汎用なハード付CPUアーキテクチャを提案する。

2. アーキテクチャ

2.1. システムの構成

全体の構成を図1に示す。

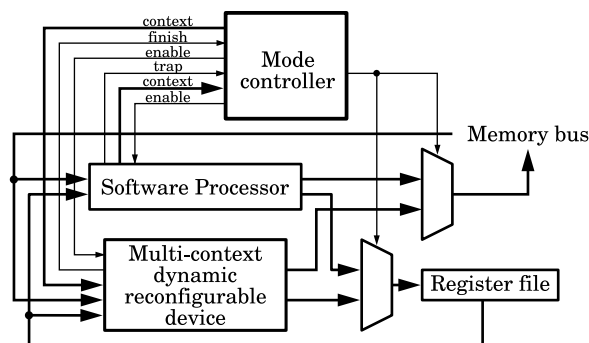


図1: 全体の構成

A method to improve the performance of a reconfigurable processor.

† Faculty of Information Science, Osaka Institute of technology

2.1.1. 部品の説明

各部品の機能を以下に示す。

[software processor]

ソフトウェアを実行する部品で、ほぼ通常のCPUに相当する。

[mode controller]

Software processorとリコンフィギャブルデバイスの切替を制御する。

[register file]

通常のレジスタファイルと同じであるが、software processorとリコンフィギャブルデバイスとのデータの受渡しにも利用される。

2.1.2. 接続

全体の構成は図1のように、レジスタファイルとメモリバスをソフトウェア処理装置と動的再構成可能なデバイスが共有する。どちらの値が使用されるかは、モードによって選択される。

2.1.3. software processorの仕様

Software processorは、fetch, decode, execute, writeの4段パイプラインになっている。分岐命令のディレイスロットは3命令、ロード命令のディレイスロットは2命令である。

2.1.4. メモリバスの仕様

メモリはインタリーブになっており32bitアクセスが可能。読込に2クロックかかる。

2.2. モード

本CPUには以下のソフトウェアモードとハードウェアモードの2つのモードがある。モードの切替はmode controllerが行う。

[ソフトウェアモード]

ソフトウェアを実行するモードで、メモリから命令を読み実行する。このとき、レジスタファイルとメモリバスへはsoftware processorの出力が送られる。

[ハードウェアモード]

リコンフィギャブルデバイスで処理するモードで、選択されたコンテキストが動作する。レジスタファイルとメモリバスへはリコンフィギャブルデバイス出力が送られる。

このとき、software processorは待機状態になっている。

2.3. モード切替

ソフトウェアモードとハードウェアモードの切替を説明する。

2.3.1. ソフトウェア→ハードウェア

ソフトウェアモードからハードウェアモードへの切替は、Jump(call)命令によって実行される。Jump命令の実行時に分岐先のアドレスがハードウェアとして登録されているか調べ(これは連想記憶メモリによって高速に実行することができる)登録されていないければ、通常のJumpを行う。

ハードウェアが登録されていればそのコンテキスト番号を mode controllerへ送る。ここで、そのままハードウェアモードに切替えるとディレイスロットが実行されず、通常のjump命令と動作が一致しなくなる。このため通常のJump命令実行時に分岐先のアドレスの命令が実行ステージに入るのと同じタイミングで切替が実行されるようにする。

ハードウェアモードが終了した後正しいアドレスからソフトウェアの実行が再開される必要がある。このために、プログラムカウンタにリンクレジスタの値を入れる。これによって、call xxx,returnを最適化して jump xxxとしている場合にも正しく処理を再開することができる。図2参照。

2.3.2. ハードウェア→ソフトウェア

ハードウェアでの処理が終了すれば、mode controllerへ finish信号を送る。これによってソフトウェアモードに切替わり、software processorはプログラムカウンタが指す値から実行を再開する。図3参照。

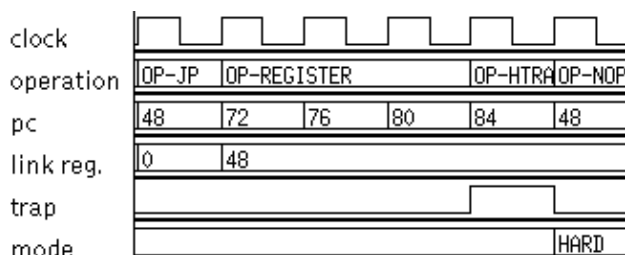


図2: ソフトウェア→ハードウェア

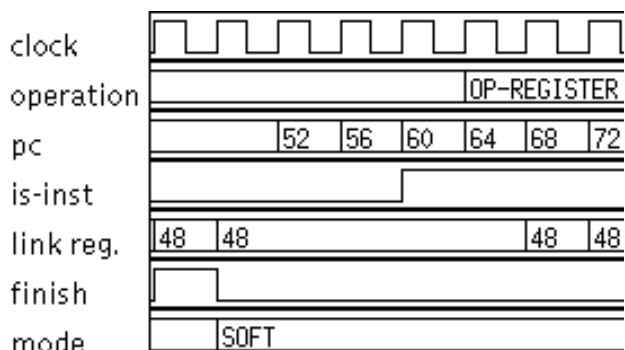


図3: ハードウェア→ソフトウェア

3. 使用例

3.1. パリティチェック

32ビットのパリティ計算をソフトウェアで実行すると17クロックかかる。対してハードウェアでは切替のオーバヘッドを含めて5クロックで実行できる。連続して行う場合には1ワード1クロックで実行できる。

3.2. バイトサーチ

バイトサーチを実行した。C言語で書くと以下のようになる。

```
char *byte_search(char ch, char *start, char *end) {  
    while (ch != *start && start != end) {  
        start++;  
        return start;  
    }  
}
```

このプログラムを機械語にしたところ、本CPUではループ一回につき8クロックとなった。できのよいCPUでも、スーパスカラでない場合少なくとも4クロックはかかると考えられる。対してハードウェアでは1クロックで実行できる。また、4バイトずつ同時に比較することも可能である。

4. まとめ

再構成可能なデバイスを搭載して、サブルーチンをハードウェアに置換えられるCPUアーキテクチャを提案した。このアーキテクチャによって自動的かつ少ないオーバヘッドでソフトウェアとハードウェアを切替えることができ、数倍程度の高速化が可能であることを確認した。そして次の効果が期待できる。

(1) 汎用性

任意の用途に対して、プログラムでHDLを用意すると高速な処理が可能なる。

(2) ハード規模の低減

多数のハードウェアを使い分ける場合、ROMやディスクに回路構成(ネットリスト)を準備すると、膨大なチップになるのを防ぐことができる。

今後の課題としてハードウェアでの実行時における外部割り込みへの対応が必要である。また、現在の方法ではハードウェアかソフトウェアのどちらか一方しか動作できず無駄がある。このためハードウェアとソフトウェアの並列処理が望まれる。

謝辞

本研究に際し、ご助言下さった小島正典教授に感謝いたします。

文 献

- [1] 長谷川 揚平, 山田 裕, 出口 勝昭, 安生 健一朗, 栗島 亨, 天野 映晴: リンコンフィギャラブルプロセッサ上でのブロック暗号 RC6の実装, 情報処理学会研究報告 vol.2004, No.5, pp 29-34