



## プロセッサにおけるメモリアクセス命令のスキップ手法

**目的: 不要なロード命令の実行をスキップし、プロセッサの命令実行の効率性を向上させるプロセッサ・アーキテクチャ**

プロセッサの命令実行の効率性を妨げる要因の1つとして、ロード命令の実行にかかる時間が大きいことが挙げられる。この研究の目的は、不要なロード命令の一部をスキップするようなプロセッサの構成を提案し、命令実行の効率性をあげようとするものである。

コンパイラが生成する実行プログラムには、そのプログラムの目的のためには「本質的に」不要なロード命令が存在、これを**レジスタ番号の読み替えにより、スキップしたい**

```

①ADD R1←Ra,Rb //汎用レジスタR1に中間結果格納
②ST R1→M[adr] //データを主記憶に退避
:
:
③LD R2←M[adr] //主記憶よりデータを復帰
④SUB Rc←R2,Rd //そのデータを使い、以降の計算
:
:
    
```

図1. 命令列の例

```

①ADD P3←Ra,Rb //汎用レジスタP1に中間結果格納
②ST P3→M[adr] //データを主記憶に退避
:
:
③LD P5←M[adr] //主記憶よりデータを復帰
④SUB Rc←P3,Rd //①の結果をP1から使い、計算
:
:
    
```

図2. 提案手法で読み替えた例

動的に読み替え

図1. の流れ:

- ①までの何らかの一連の実行結果がR1に格納されているとする
- ここで空いている汎用レジスタが枯渇、②主記憶にデータを退避、R1を空ける
- 後に、データが必要になったところで③ロード命令によりR2に復帰
- ④以降で、R2に格納されたデータを用いた命令が実行される

書き込み、読み出し順の関係から、実行順番は変更できない

**しかし、②③のメモリアクセス命令は、本来の目的のためには不要**

動的にレジスタ番号を読み替える

- ①で書き込み→④で読み出す関係となり、**①→④の順に実行可能**

**②③、特に遅いロード命令③をスキップして、④以降の実行が可能**

### 前提技術: Out-of-Order実行とレジスタリネーミング

現在のプロセッサの特徴:

- ・命令アーキテクチャのレジスタ数より多くのレジスタを搭載
- ・Out-of-Order (OoO) 実行で命令の順番を変えて効率良く実行
- ・**レジスタリネーミング**で、OoO実行の効率をあげている

・**レジスタリネーミング:**

命令指定の汎用レジスタ(論理レジスタ)をそのまま利用せず、実際に搭載されたレジスタ(物理レジスタ)に、動的にリネーミングする。OoO実行での、順番変更可能な箇所が増やせる。

①ADD R1←Ra,Rb	①ADD P3←Pa,Pb
②ST R1→M[adr]	②ST P3→M[adr]
:	:
:	:
③LD R2←M[adr]	③LD P5←M[adr]
④SUB Rc←R2,Rd	④SUB Pc←P5,Pd
:	:

※Pnは、物理レジスタ番号を示す

図3. 通常のレジスタリネーミングを施した様子  
R1→P3, R2→P5にリネーミング

※この命令列では、OoO実行には何ら寄与せず、①②③④の実行順は変えられない。

現行のリネーミングでは、③の書き込み先と④の読み出し元は同じ⇒③④の実行順序は変えられない。

### 実装方法: 現行のレジスタリネーミングを拡張

現行のリネーミング:

- ③: 書き込みレジスタR2に、空き物理レジスタ割り当て(P5)  
R2 ⇔ P5 が対応していると、対応表に記録
- ④: 対応表を参照し、読み出しレジスタR2をP5として書き換え

提案手法: 赤字部が追加部分

**物理レジスタ⇔メモリアドレス対応表 (PA表)を用意**

- ①: 書き込みレジスタR1に、空き物理レジスタ割り当て(P3)  
対応表に、R1⇔P3 と記録
- ②: 対応表より、読み出しレジスタとしてP3に書き換え  
**PA表に、P3⇔メモリアドレスの対応を記録**
- ③: 書き込みレジスタR2に、空き物理レジスタ割り当て(P5)  
**PA表を参照し、メモリアドレス⇔P3 がわかり、対応表には R2⇔P3 と記録**
- ④: **現行処理と同じく対応表を参照すれば、R2をP3として書き換えられる**

①ADD R1←Ra,Rb	①ADD P3←Pa,Pb
②ST R1→M[adr]	②ST P3→M[adr]
:	:
:	:
③LD R2←M[adr]	③LD P5←M[adr]
④SUB Rc←R2,Rd	④SUB Pc←P3,Pd
:	:

図4. 提案手法によるレジスタリネーミングを施した様子