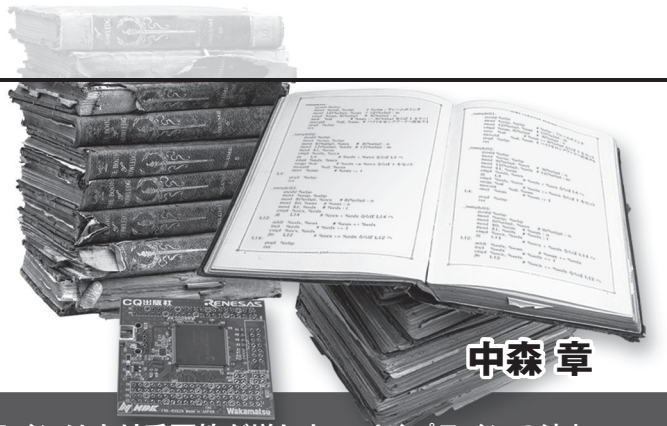
第2章 パイプライン化を中心とした

進化するCPUの高速化技法



中森 章

RISCアーキテクチャの登場以降，CPU設計においてパイプラインはより重要性が増した．パイプラインの流れを阻害するような要因を極力排し，スムーズに流れるようにすることが性能に直結する．
ここでは，パイプライン設計を中心にCPUの高速化技法を見ていく．

（編集部）

1. CPUの高速化技法

● 高速化技法：キャッシュ

これまで見てきたように，外部メモリのアクセス・タイムが命令フェッチ（IF）状態とメモリ・アクセス（Mem）状態の遷移時間を決定します[注1]．そこで，キャッシュという高速化技法が考えられます．

キャッシュとはCPUに内蔵する高速メモリです．これは，外部メモリの一部の内容のコピーを保持します．CPUはまずキャッシュをアクセスし，キャッシュに目的のアドレスの命令やデータがあれば，キャッシュから命令やデータを取り込みます．キャッシュは，通常1クロックでアクセス可能です．もし，目的のアドレスの内容がキャッシュになければ，外部メモリから目的の命令やデータがキャッシュに取り込まれます．このとき，目的とするアドレスの近くにあるデータをある程度まとまった単位でキャッシュに取り込みます．これにより，一度アクセスした近傍の命令やデータはキャッシュにある確率が高くなります．CPUが要求した命令やデータがキャッシュに存在することをヒット，キャッシュに存在しないことをミスといいます．また，外部メモリにアクセスを行ってキャッシュを更新する間，CPUは待たされているのが普通です．この様子を**図1**に示します．

● 高速化技法：パイプライン

キャッシュにヒットする場合，演算命令は，IF，ID，EX，WB状態の4クロック，ロード/ストア命令は，IF，ID，EX，Mem，WB状態の5クロックで実行できます．しかし，さらなる高速化を行う場合はパイプラインという技法があります．各状態においてほかの状態と内部資源が競合しないならば，各状態をオーバラップして実行させられます．つまり，1クロックごとに命令フェッチを開始させれば，あるクロック以降は1クロックごとに命令実行が終了することになります．しかし，今の場合，演算命令とロード/ストア命令を混在させて実行すると，データ格納（WB）状態が衝突することがあります（**図2**）．

内部資源が衝突しないようにするには，演算命令の状態遷移でEXとWBの中間にダミーのMem状態を挿入して5クロック実行にしてしまうことです．演算命令の実行クロック数は，4から5へと見かけ上増加しますが，命令が1クロックごとに終了することを考えると，最終的な性能低下はありません．これがパイプラインの威力です．

パイプラインでは各状態のことを段（ステージ）と呼び

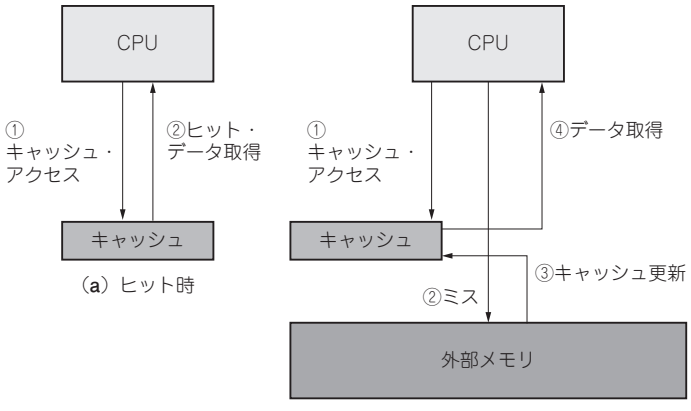


図1　キャッシュの概念

注1：メモリ・アクセス（Mem）状態とは，データ・フェッチ（DF）状態とデータ出力（DW）状態の総称としている．第1章の図10を参照のこと．

KEYWORD——キャッシュ，パイプライン，ストール，レジスタ・バイパス，フラグ・ハザード，遅延分岐