

アーキテクチャシミュレータの概要

7N-1

安里 彰 梅澤 靖

富士通研究所

1 はじめに

マイクロプロセサの設計フローは一般に図1のように捉えることができる。基本設計フェーズでは、回路のブロック図を描くなどして、プロセサの基本的な構造を決定する。次の詳細設計フェーズでは、ハードウェア記述言語を用いて回路を記述し、論理合成等によりネットリストを得る。そして実装フェーズではセルの配置やセル間の配線を行なう。設計目標を満たす実装ができればプロセサ設計は終了し製造に移ることになる。図中のフィードバックは、論理シミュレーションでバグが発見されたり、デザインルールを満たすような実装ができなかったり、実装はできたものの遅延時間に問題があるような場合に、設計の或る段階に戻ってやり直すことを意味する。しかし、ここで基本設計フェーズにまで戻るようでは、開発期間の大幅な遅れだけを見てもプロジェクトの失敗につながる可能性が高い。そうならないためにも、初期の段階から先を見通した設計を行なうことが非常に重要である。我々はこのような観点から基本設計フェーズを重要視し、このフェーズにおける作業をサポートするツールを開発した。

本システム(アーキテクチャシミュレータ)の開発に当たっては、

1. プロセサのモデルが容易に構築できること。
2. 各モデルについて性能およびサイズ/遅延時間の評価ができること。

を基本方針とした。なお開発にはC言語を用い、ターゲットとしてはスーパースカラタイプのマイクロプロセサを想定した。

アーキテクチャシミュレータは、性能評価システムと物量/遅延見積りシステムから構成される。以下ではそれぞれのシステムについて説明する。

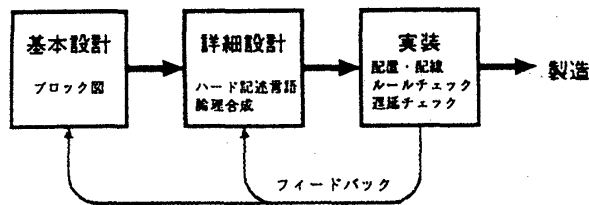


図1 設計フロー

2 性能評価システム

2.1 全体構成

図2は性能評価システムの構成図である。マシンモデルは実際のプロセサの如く振舞うプログラムで、内部にメモリキャッシュ・レジスタファイル等の記憶領域や命令実行制御機構を有し、クロックサイクル毎に内部状態を変化させていく。メモリイメージファイルはマシンモデル内の記憶領域の初期値を格納している。パラメタファイルはマシンモデルを構築するためのパラメタを記述したファイルである。シミュレーション制御プログラムは、メモリイメージファイルの内容をマシンモデルに設定した後、マシンモデルを起動し、更に適当なタイミングでマシンモデルから得られる内部情報を情報ファイルに出力する。

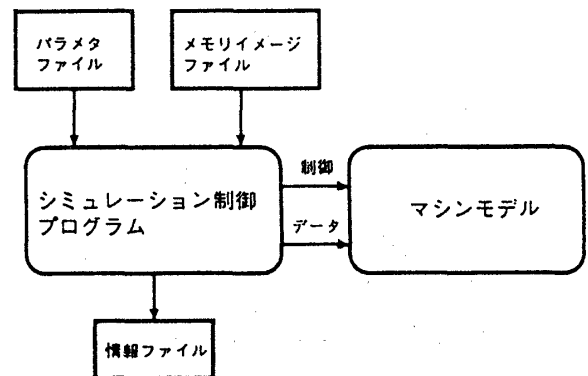


図2 性能評価システム

2.2 マシンモデル

マシンモデル中の記憶領域はC言語の大域変数として表され、その値はパラメタファイルによって与えられる。これらの変数値を変化させることで、キャッシュ容量、パイプライン段数、スーパースカラ度といったプロセサの基本構成を変えることができる。

また、命令フェッチ、デコード、コミット等の制御機構は、それぞれがCの関数で表現される。それぞれの関数間のインターフェースは入出力変数のレベルで定められている。例えばデコード関数は、opcode[decode.width]という入力変数に格納された1個以上(decode.width個)のマシン命令を受け取り、rs_in[rs.width]という出力変数が表す予約ステーションの(rs.width本の)入力信号を作成すると規定されている。これらの規定の範囲内で各関数を自由にコーディングして良いので、そうすることで分岐予測、レジスタリネーミング、

命令実行(in_order, out_of_order)、リカバリといった制御方式に複数の選択肢を用意することができ、記憶領域のサイズの変化と組み合わせて、多様なプロセッサモデルを構成することができる。

なお制御用関数のコーディングに当たっては、[1][2]をはじめとする諸文献を参考にした。また、文献[3]で筆者らが提案したパイプライン制御方式も、この枠組みで扱うことができる。

2.3 パラメタファイル

パラメタファイルは図3のように3つのパートから構成される。第一のパートはdecode_width等の変数の値を規定する部分である。第二パートは記憶領域の属性を規定する部分で、キャッシュなら容量、連想方式、ラインサイズ等、レジスタファイルなら語長、エントリ数、ポート数等の情報が記述されている。最後は制御用関数を指定する部分である。指定は関数の名前を用いて行なう。

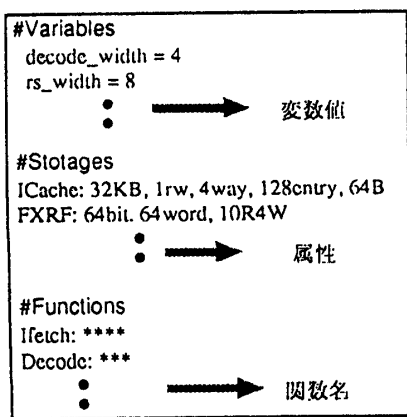


図3 パラメタファイル

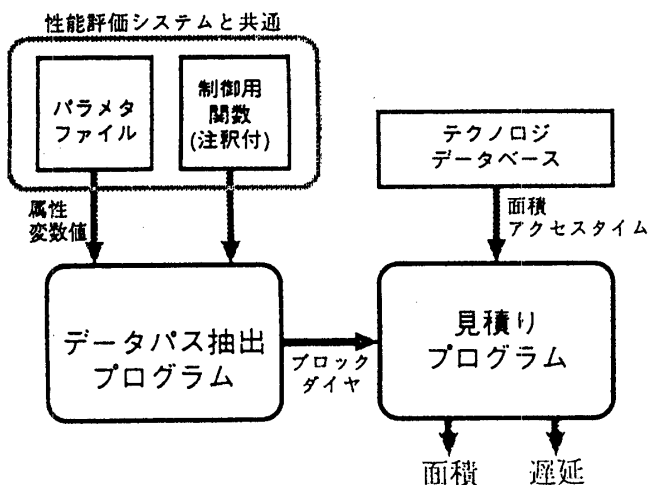


図4 面積/遅延見積りシステム

2.4 評価方法

ベンチマークプログラムのコードをマシンモデル内のメモリに設定した後、マシンモデルを起動し、終了までに要したサイクル数を測定して、各プロセッサモデルの性能を評価する。また、任意の時点の内部状態をトレースできるので、その情報を元にプロセッサの動作を解析して、更なる改良の指針を得ることができる。

3 面積/遅延見積りシステム

3.1 全体構成

図4にシステムの構成図を示す。パラメタファイルと制御用関数は性能評価システムのものと同じである。テクノロジーデータベースは設計に使用するテクノロジーに関する情報を保持している。これらのファイルの内容を元に、データベース抽出プログラムと見積りプログラムによって面積と遅延時間の見積りを行なう。

3.2 システム動作

データベース抽出プログラムは、制御用関数間の接続関係に関する知識を有し、その知識と制御用関数のソースコードの内容から、データベースを抽出してブロックダイヤを構築する。ただし、Cのプログラムソースからそれが表現する回路を完全に抽出するのは困難であるから、抽出を支援するための情報を注釈の形でソースに付加することにした。

見積りプログラムは、データベース抽出プログラムから受け取ったブロックダイヤの各構成要素をレイアウトした場合の面積および遅延時間を計算するプログラムである。計算にはテクノロジーデータベース中に存在する、レジスタファイルの面積やアクセス時間、セレクトラを構成するセルや配線領域の面積などの情報を利用する。テクノロジーデータベースに該当するモジュールがない場合は、属性の異なる他のモジュールのデータから計算して求めるようにした。

4 おわりに

プロセッサ設計の初期フェーズを支援するツールとして開発したアーキテクチャシミュレータの概要を述べた。本システム開発の主たる目的はプロセッサ設計期間の短縮であるが、性能評価システムを用いたシミュレーションにより、各種アーキテクチャモデルと性能に関するデータも得られている。これについては別の機会に報告する予定である。

参考文献

[1] M.Johnson: "Superscalar Microprocessor Design", Prentice-Hall, 1991.
 [2] D.A.Paterson, J.L.Hennessy: "Computer Architecture A Quantitative Approach", Morgan Kaufmann Publishers, 1990.
 [3] 安里他: "PIM/pにおけるパイプライン制御方式", 情報処理学会第43回全国大会, 1991.