

# 高速なシミュレーションのための 3W-02 マイクロプロセッサ評価用モデルの自動生成\*

山崎 志保 塩見 彰睦†

静岡大学大学院情報学研究所‡

## 1 はじめに

近年, VLSI の集積度が向上し, 大規模で複雑なアーキテクチャを持つ ASIP (Application Specific Instruction set Processor) の設計が可能となるとともに, より良いアーキテクチャの探索が困難となり, 設計工数が増大している。そのため, 設計の初期段階でアーキテクチャを短期間に評価することが求められている。

我々の研究グループが提案している, PEAS-III マイクロプロセッサ設計支援システム [1] では, 設計の詳細な部分が決定していない設計初期段階で, 高速にシミュレーションできる命令セットレベルのシミュレーションモデルを自動生成するシステムを提案している [2]。以下, このシステムを ISS (Instruction Set Simulation) 生成系と表記し, 生成されるシミュレーションモデルを ISS モデルと表記する。これにより, 従来の問題である人手設計でのシミュレーションモデルとシミュレータの記述工数を削減でき, 設計の初期段階でアーキテクチャを短期間に評価できる。

## 2 目的

PEAS-III システムの従来 ISS モデルでは, プロセッサ内のコンポーネント間のデータの受渡しに信号を用いており, デルタ遅延が生じるため, シミュレーションに時間がかかっていた。そこで, 文献 [3] では, デルタ遅延が生じない変数を信号の代わりに用いることにより, 従来 ISS モデルよりも高速にシミュレーション可能なモデルを提案した。

本研究では, 設計の詳細な部分が決定していない段階で, 高速にシミュレーション可能な ISS モデルを自動生成することにより, 設計初期段階におけるアーキテクチャの評価を短期間にできるようにすることを目標とし, アーキテクチャや命令セットなどの抽象度の高い設計情報を用いて, 文献 [3] で提案した ISS モデルを自動生成する手法を提案する。

## 3 自動生成方針

PEAS-III でのプロセッサ設計の流れは, CPU タイプやレジスタの数, ビット幅など基本的なアーキテクチャを決定し, リソース, 入出力ポート, 命令セットを定義し, 命令のビヘイビアレベルの動作を記述し, ISS モデルを自動生成する。その後, 命令の詳細な動作をクロックベースで記述し, 構造レベルのシミュレーション用モデルと論

理合成用モデルを自動生成する。

ISS 生成系は, 基本的なアーキテクチャ, 使用リソース, 入出力ポート, 命令セット, 命令のビヘイビアレベルの動作記述を基に, ビヘイビアレベルの VHDL 記述のシミュレーションモデルを自動生成する。

### 3.1 ISS モデル

生成される ISS モデルは, 実行に複数クロック・サイクルかかる命令を 1 クロック・サイクルで処理する。命令処理結果はキューに入力され, キューは適切なクロック・サイクル後にその命令処理結果を出力し, 適切なプロセッサの動作を実行することにより, クロックに同期したプロセッサの動作を実現する。

実行クロック・サイクルのタイミング調整により, プロセッサの動作を実現するため, タイミング調整を行うキューをプロセッサアーキテクチャに応じて生成し, 制御することで, 様々なタイプのプロセッサアーキテクチャを容易に実現できると考える。

### 3.2 キューの生成方針

プロセッサの動作のクロック・サイクル数の調整を行うキューの生成は, バイプライン段数, ステージにかかるクロック数, ステージで行われるプロセッサの動作, 演算にかかるクロック数を用いて行う。プロセッサの動作は以下のように分類する。現段階では, ハードウェアアーキテクチャであり, 各動作は 2 回以上は出現しないものとする。

1. 命令フェッチ
2. 命令デコード
3. レジスタからの読みだし (オペランドフェッチ)
4. 演算
5. アドレス生成
6. メモリからの読みだし (メモリロード)
7. メモリへの書き込み (メモリストア)
8. レジスタへの書き込み
9. PC 更新
10. PC への書き込み

実行までのクロック・サイクル数を調整する必要がある動作は, 一時記憶領域への書き込み動作, メモリアクセス動作, マルチサイクル演算であると考えられ, 分類した動作でいうと, 4. 演算, 6. メモリからの読みだし, 7. メモリへの書き込み, 8. レジスタへの書き込み, 10. PC への書き込みにあたる。これらの動作の実行タイミングを調整するキューを生成する。

マルチサイクル演算がない場合, 演算以外のタイミング調整を行う動作に, 命令を 1 クロック・サイクルで処理してから各動作を実行し始めるまでにかかるクロック

\*An Auto Generation of Micro Processor Simulation Model for High-speed Simulation

†Shiho Yamazaki, Akichika Shiomi

‡Shizuoka University Graduate School of Information

ク数だけデータを保持するキューを生成する。また、メモリからの読みだしが終了してからレジスタへの書き込み、メモリへの書き込みが始まるまでにかかるクロック数だけデータを保持するキューをそれぞれ生成する。タイミング調整を行う動作が同じステージにある場合は、共通のキューを用いる。

これらに加えて、マルチサイクル演算がある場合、命令を1クロック・サイクルで処理してからマルチサイクル演算が終了するまでにかかるクロック数だけデータを保持する演算動作のキューと、演算終了から演算以外のタイミング調整を行う動作を実行し始めるまでにかかるクロック数だけデータを保持するキューを生成する。

3.3 ISS モデルの動作

表1のようなパイプラインアーキテクチャでマルチサイクル演算命令がない場合、長さ2のPC書き込み用、長さ3のメモリアクセス用、長さ4のレジスタ書き込み用、長さ1のメモリ読みだし-レジスタ書き込み用の4種類のキューを生成する。このときのパイプラインの動作と生成されるVHDL記述の対応を図1に示す。

VHDL記述では、1クロック・サイクルで命令フェッチからレジスタへの書き込みまでの動作が逐次的に行われる。この例はパイプラインアーキテクチャであり、同じクロック・サイクルに複数の異なる命令の動作を並列実行するため、各キューには複数のデータが存在し、パイプラインレジスタの役目をする。

表1: パイプラインステージの構成

ステージ番号	プロセッサ動作	クロック数
ステージ1	命令フェッチ	1
ステージ2	命令デコード, PC更新	1
ステージ3	オペランドフェッチ, 演算, アドレス生成, PC書き込み	1
ステージ4	メモリ読みだし, メモリ書き込み	1
ステージ5	レジスタ書き込み	1

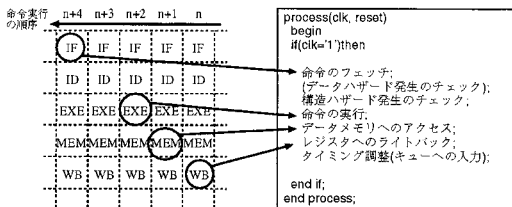


図1: パイプライン動作と制御部のVHDL記述の対応

4 実験と考察

第3章で述べた生成方針に基づき、ISS生成系を実装した。表1のような5段パイプラインアーキテクチャで、構造ハザード・データハザード発生時の命令のストール有り、データハザード発生時のレジスタからのフォワーディング有り、ハーバードアーキテクチャ、in-order実行であるMIPS R3000ベースのISSモデルを自動生成し、シミュレーション速度を評価した。

シミュレータにMentor Graphics社のModelSim EE 5.3dを使用し、シミュレーションの時間測定には、time

コマンドを使用した。シミュレーション速度の評価は、1秒当たりの命令実行数で行い、PEAS-IIIのHW生成系で生成した構造レベルのシミュレーション用モデルと従来のISSモデルを比較対象とした。

シミュレーション速度の測定結果を表2に示す。本研究のISSモデルのシミュレーション速度は構造レベルのシミュレーション用モデルと比較すると約4.5倍であった。また、従来のISSモデルと比較すると約2倍であり、より高速なシミュレーションを実現することができた。

表2: 測定結果

モデル	速度 [命令/秒]	速度比
シミュレーション用構造レベルモデル	763.78	1.00
従来のISSモデル	1783.97	2.33
本研究のISSモデル	3479.29	4.55

現段階では実装している命令数が少なく回路規模が比較的小さいため、構造レベルのモデルと従来のモデルのシミュレーション中の信号にかかるデルタ遅延が少ないので、本研究のモデルとの速度差への影響は少ないが、命令数を増やし回路規模の大きいモデルで評価すれば、構造レベルのモデルや従来のモデルと比較した本研究のモデルのシミュレーション速度の向上は、今回の測定結果以上になると考えられる。

キューへの入出力時の余分なデータの受渡しや、レジスタのデータを扱わない命令でもデータのフォワーディング検出を行ってしまっている点など、制御の冗長な部分を改善したり、頻繁に使用される命令が早く検出できるような命令デコード方法を採用すれば、さらにシミュレーション速度を高速化できると考える。

5 まとめ

プロセッサ設計の初期工程で得られる抽象度の高い設計情報から、高速にシミュレーションできる命令セットレベルのモデルを自動生成する手法を提案し、生成系を実装した。MIPS R3000ベースのISSモデルを生成し、シミュレーション速度を評価し、従来のISSモデルよりも約2倍高速にシミュレーションできるISSモデルを生成できることを示した。

今後は、より高速にシミュレーション可能なモデルを検討するとともに、生成可能なアーキテクチャを拡張していく予定である。

参考文献

- [1] 塩見彰睦, 今井正治, 片岡健二, 青山義弘, 佐藤淳, 引地信之, “ASIP設計用コデザインワークベンチPEAS-IIIの提案”, 情報処理学会 設計自動化, pp. 73-80, 情報処理学会, 1995.
- [2] 小崎展, 塩見彰睦, 今井正治, “PEAS-IIIにおけるシミュレーションモデルの自動生成”, 情報処理学会 設計自動化, pp. 69-74, 情報処理学会, 1995.
- [3] 山崎志保, 塩見彰睦, “パイプラインプロセッサのシミュレーションモデルの高速化”, 電気関係学会東海支部連合大会 講演論文集, pp. 320, 2000.