

アーキテクチャ性能評価のための動作記述の検討

2H-1

小川一 高井裕司 水野雅信 村岡道明
松下電器産業株式会社 半導体研究センター

1. はじめに

プロセッサ設計において、RTレベルより上流でのアーキテクチャの検証及び性能評価を行う試みはいくつか行われている[1,2]。この際、用いられるアーキテクチャの動作記述には、次の点で課題があった。

- 動作記述の拡張性、流用性
- 下流設計環境との整合性

そこで上記課題を解決するために、アーキテクチャレベルのモデルを明確化し、汎用的なハードウェア記述言語（Verilog-HDL[3]）を用いて、高速なシミュレーションと性能評価を可能とする動作記述の規約化を行った。

本稿では、Verilog-HDLによるアーキテクチャレベルでの動作記述（アーキテクチャ動作記述）及びそのモデルを提案し、これを用いた設計フローと通信プロセッサでの評価結果を報告する。

2. アーキテクチャ動作記述

2.1 アーキテクチャ動作記述の要件

アーキテクチャレベルでは、RTレベルよりも高速なシミュレーションと記述の容易性及び、命令レベルよりもパイプラインなどのアーキテクチャ動作を明確に表現することが必要である。

よって、提案するアーキテクチャ動作記述に求められる要件は次のものである。

- (1) アーキテクチャの動作の明確表現性
- (2) 性能評価に必要な最小限なリソース定義
- (3) シミュレーションの高速性
- (4) 記述容易性

2.2 動作記述のモデル

(1) リソースモデル

性能評価を行うために必要なリソースは、レジスタ、バス、メモリ、演算器、入出力ポートである。

(2) デコーダモデル

デコード部分は、デコードフェーズを記述する1つの always ブロックにおいて行い、デコード情報

はパイプラインで伝達させる。

(3) データ転送モデル

ハードウェアを反映して、バスを介したデータ転送記述を行う。

また、サイクルベースシミュレータによる高速検証が可能となるように次の記述形式を用いる。

- クロックエッジ動作記述
- 遅延を用いない記述

(4) パイプラインモデル

パイプライン動作の記述は、命令レジスタの情報やデコードした制御情報など、デコード後のステージまで保持しなければならない情報は、パイプラインレジスタとして格納し、パイプライン制御部として設けて記述する。

提案するアーキテクチャ動作記述の例を示す。

```

always@(posedge clk) begin // Decode
  casex(IR)
    'ADD:begin
      D_operation='add;
      ...
    end

always@(posedge clk) begin // Execution
  ABUS=Reg_A; BBUS=Reg_B;
  casex(EX_operation)
    'ADD:ALU(ABUS,BBUS,ALUOUT);
    ...
  end

always@(posedge clk) begin // Pipeline Control
  OF_IR<=DEC_IR; // 命令データの搬送
  EX_IR<=OF_IR; // DEC → OF → EX
end

```

リスト1.アーキテクチャ動作記述の例

2.3 動作記述の比較

図1に、我々の提案するアーキテクチャ動作記述と命令レベル、RTレベルの動作記述との違いを、加算命令を例にとって模式的に示す。

2.4 性能評価項目

現在、アーキテクチャレベルでの性能評価項目として必要と考えているのは、次の2つである。

- バス、演算器の使用頻度解析
- パイプラインハザードの検出

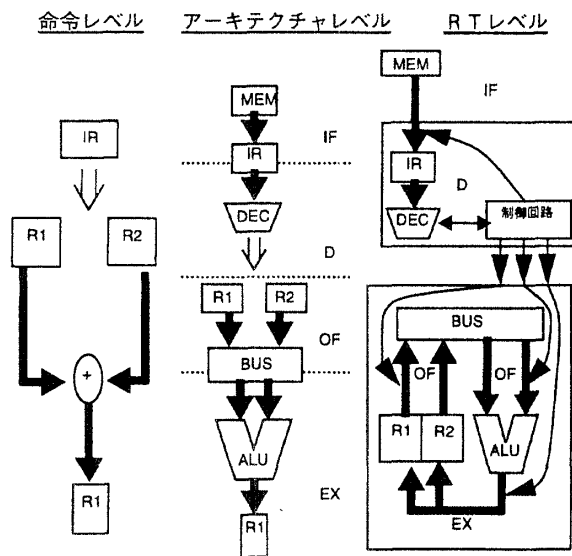


図1.各レベルのモデル化の違い (add R1 and R2)

3. アーキテクチャ動作記述を用いた設計フロー

アーキテクチャ動作記述を用いた設計フローについて述べる (図2)。

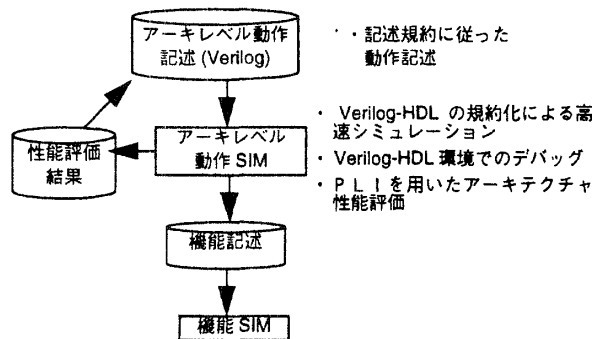


図2.アーキテクチャ動作記述を用いた設計フロー

アーキテクチャレベルにおいて、Verilog-HDLによる記述規約に従ったアーキテクチャ動作記述を行い、Verilog-HDLのシミュレーション、デバッグ環境でアーキテクチャの検証を行う。さらに性能評価では、PLIを用いて性能評価関数をプラグインする。性能評価結果に基づきアーキテクチャの最適化を行った後に、RTLレベル以降の設計を行う。

以上のアーキテクチャの設計フローは、アーキテクチャレベルにおいて効率良く性能評価を行うことにより、設計の早い段階でのアーキテクチャの最適化が可能なり、RTLレベル以降からのアーキテクチャ設計に関する手戻りを削減することができる。

4. 評価結果

今回は、約3万ゲート規模の通信用プロセッサに対し適用して、記述量、シミュレーション速度、及び可能な性能評価項目について評価を行った。

表1 (a).記述量、シミュレーション速度のRTLレベルとの比較

項目	アーキテクチャレベル	RTLレベル
記述量 (Verilog-HDL)	4,500行	25,000行
SIM速度	4.5s	74.2s

表1 b.可能な性能評価の命令レベルとの比較

性能評価項目	アーキテクチャレベル	命令レベル
命令動作解析	○	○
レジスタ使用頻度	○	○
バス、演算器使用頻度	○	×
パイプラインハザード検出	○	×

表1から、RTLレベルよりも少ない記述量、高速なシミュレーションで、バス、演算器の使用頻度、パイプラインハザードの検出が可能であり、我々の提案するアーキテクチャ動作記述が要件を満たすことがわかる。

5. おわりに

プロセッサ設計のアーキテクチャレベルにおけるアーキテクチャ動作記述の提案とその適用評価について報告し、その有効性について示した。今後の検討課題として、下記のものがある。

- アーキテクチャ動作記述での設計フローの実証
- より複雑なアーキテクチャへの拡張
- アーキテクチャ動作記述からRTL記述の自動生成手法

参考文献

- [1] 塩見彰睦, 片岡健二, 今井正治, 青山義弘, 佐藤淳, 引地信之, "ASIP 設計用ワークベンチ PEAS-III のアーキテクチャ入力系の試作", DA シンポジウム, pp129-134, 1996
- [2] 富山宏之, 赤星博輝, 安浦寛人, "アーキテクチャ評価用コンパイラの自動生成", 情処研報, 93-DA-96-19, pp143-150, 1993
- [3] E.Sternheim, R.Singh, and Y.Trivedi "Digital Design with Verilog HDL", Automata Publishing Company, 1990