

松本 展徳吉 隆宏
N. Matsumoto T. Tokuyoshi

マイクロプロセッサ(MPU)などを集積したASIC(用途特定IC)の開発では、設計段階におけるハードウェア(HW)/ソフトウェア(SW)協調シミュレーションが求められている。HW/SW協調シミュレーションには、正確(サイクルアキュレート)かつ高速なプロセッサモデルが必要であり、そのためのモデリング方式として、当社は、パイプライン制御プリデコード方式を考案した。今回、この方式に基づいて、32ビットRISC(縮小命令セットコンピュータ)MPU TX39のためのサイクルアキュレートなシミュレーションモデルを開発した。このモデルをHW/SW協調シミュレーションツールとともに用いることにより、サイクルアキュレートなHW/SW協調シミュレーションが可能になる。

The design of ASICs with embedded processors is generating a need for hardware/software (HW/SW) co-simulation functionality, which requires cycle-accurate processor models as key elements for the simulation of embedded applications.

This paper presents a new approach, the pipeline-control pre-decode method, for cycle-accurate processor modeling. Using this approach, we have developed a cycle-accurate model for the Toshiba TX39 32-bit RISC microprocessor. Combination of the cycle-accurate model with a commercial co-simulation tool will provide a cycle-accurate HW/SW co-simulation environment.

1 まえがき

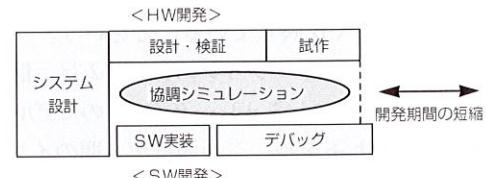
システムオンシリコンの時代を迎え、一つのASICのチップ上に、組込みプロセッサ、IP(Intellectual Property)，そして大規模なSWを集積させることが可能になった。従来、SW開発は、図1(a)のように、HW開発と独立に進められ、その過半はHWの試作後になされることが普通であった。しかし、SWの大規模化に伴い、SW開発がシステム開発のボトルネックとなってくる。これからは、図1(b)のように、SW開発をHW開発と並行して進めることが求められている。HW/SWの協調シミュレーションは、HWとSWの開発を並行して進めるために有効な手段である。

HW/SW協調シミュレーションに対し、以下のような具体的な効果が期待されている。

- (1) SWデバッグを、早期に開始できることにより、SWも含めたシステム全体の開発期間を短縮できる。
- (2) HW検証のために応用SWを利用できる。また、HWの検証ベクトルの作成に、使いやすいSW開発環境を利用できる。
- (3) システムの性能にクリティカルな部分を、HWの試作前の早い段階で評価・検証しておくことができる。
- (4) HWの問題がSWデバッグ時に発見されることがあるが、従来工程では、HWの試作が済んでおりHWの修正が大変なため、SWで無理に対処していた。このような問題をHWの修正で適正に対処することができる。
- (5) HWとSWの分担を適正化し、全体的に最適な設計



(a) 従来のシステム開発工程



(b) 協調シミュレーションを導入した場合のシステム開発工程

図1. HW/SW協調シミュレーションによる、開発工程の変化
HW/SW協調シミュレーションを導入することにより、HW開発とSW開発が平行に進められ、全体の開発期間が短縮される。
Impacts of HW/SW co-simulation on development process

を行うことができる。

ここでは、以上のように期待の大きいHW/SW協調シミュレーションでの構成技術全体を概観し、特筆すべき構成技術であるプロセッサのモデリング技術について述べる。

2 HW/SW協調シミュレーション

HW/SW協調シミュレーションを行うための方法は、表1

表1. HW/SW協調シミュレーション方法の分類
Classification of HW/SW co-simulation methods

方法の分類			処理性能 (IPS)	SWの デバッグ	HWの デバッグ	モデリング の大変さ
モデルの有無	機能	正確さ				
プロセッサモデルの必要な方法	フル機能モデル	ナノセコンドアキュレート	低	一	最も良好	大変
		サイクルアキュレート	~数k	良好	良好	大変
		命令レベル	10k~100k	良好	可能	中程度
プロセッサモデルの不要な方法	バス機能モデル	十分大	できない	良好	簡単	
	HWモデル	低	一	良好	タイミングのみ	
	ホスト(SW)-HWシミュレータ通信方法	十分大	良好	可能	なし	
	仮想HW	大	可能	できない	なし	
	エミュレーション	大	可能	可能	なし	

IPS : Instruction Per Second

に示すように多種多様である⁽¹⁾。これらは方法は、SWのシミュレーションのためのプロセッサモデルが必要か否かで大別される。プロセッサモデルが必要な場合は、機能面によりフル機能モデルとバス機能モデルに分類され、前者は、さらに正確さにより①ナノセコンドアキュレートモデル、②サイクルアキュレートモデル、および③命令レベルモデルに分類される。

これらの方法には、それに適した(有用な)場合がある。しかし、リアルタイム制約のある、多くの組込みプロセッサのアプリケーションの場合は、正確さと処理性能の両方を備え、HWとSWのデバッグ・検証に用いられるサイクルアキュレートなモデルがもっとも適している。ここで、サイクルアキュレートとは、動作タイミングの正確さについての一つの尺度であり、各クロックのエッジにおける動作状態の遷移を正しく反映していることをいう。

HW/SW協調シミュレーションのシステムは、図2に示すように、HWのためのシミュレータ(HWのモデル)、SWのためのプロセッサモデル、二つのモデル間のインタフェースを取りもつ協調シミュレーションカーネルからなる。

HWのモデルは通常HDL(Hardware Description Language)を用いて記述されるが、その場合の技術的課題は、シミュレーション処理性能の向上である。処理性能を向上するためには、①サイクルベースシミュレータなどの高速なHDLシミュレータ、あるいはHWのためのエミュレータを用いたり、②HWのモデルの起動を、極力少なく抑える方法(最適化)⁽²⁾が考えられている。また、処理性能を向上させるために、HDLではなく、プログラム言語のC言語^(注1)を利用する場合もある。

一方、プロセッサモデルは、プログラム言語で作られる場合が多いため、通常、処理性能はあまり問題とならない。プロセッサモデルにおける技術的課題は、動作タイミングをサイクルアキュレートの精度でシミュレートすることである。ここで注意すべきことは、SW単体のシミュレーションの場合には、モデルがサイクルアキュレートである必要がないことである。このため、これまで開発してきたモデルの多くは、サイクルアキュレートではなかった。最近になり、HW/SW協調シミュレーションに注目が集まるにつれて、サイクルアキュレートなモデルの必要性が認知されてきた。

3 モデリング技術

サイクルアキュレートなプロセッサモデルを作る方法として、パイプライン制御プリデコード方式を考案した。この方式は、プロセッサの機能と動作タイミングを分離して記述し、機能についてはプリデコード方式の命令セットシミュレータにより、動作タイミングはパイプラインレベルの制御シミュレータにより実現する。この方式では、動作タイミングが機能から明確に分離されているため、並列動作の記述能力がないプログラム言語を用いても、比較的容易に動作タイミングを記述できる。

3.1 プリデコードインタプリタ方式

命令セットシミュレータの方式は、表2に示すように次の四つの方に大別できる⁽³⁾。

インタプリタ方式は、対象プロセッサのプログラムコードを1命令ずつデコードし、実行していく方式である。一

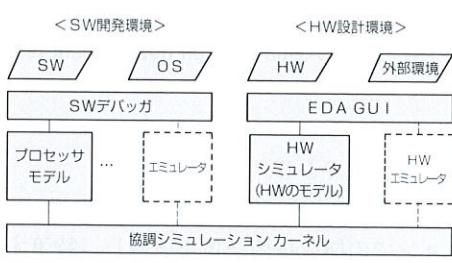


図2. HW/SW協調シミュレーションシステム HWシミュレータとプロセッサモデルが、協調シミュレーションカーネルを介して接続されている。

HW/SW co-simulation system

表2. 命令セットシミュレータの方式
Approaches for instruction-set simulator

方 式	処理性能	ポートビリティ
インタプリタ方式	低	良い
コンパイラ方式	スタティック	悪い
	ダイナミック	悪い
プリデコードインタプリタ方式	中程度	良い

(注1) 米国Bell研究所で開発されたプログラミング言語。

方、コンパイラ方式は、対象プロセッサのコードを、ホスト計算機のコードに翻訳して実行するものである。プリデコードインタプリタ方式では、シミュレーションすべき SW を、まず中間コードに変換（プリデコード）し、その中間コードをインタプリタで実行する。この方法は、インタプリタ方式とコンパイル方式の両方の長所をとったものであり、高速性と、シミュレーションのホスト計算機に対するポートビリティを兼ね備えている。

プリデコード方式自体は新しいものではないが、命令セットシミュレータとして最適な方式と考えられる。

3.2 パイプライン制御のシミュレーション方法

この方式では、動作のタイミングをシミュレーションするために、命令セットシミュレータのほかにパイプライン制御シミュレータをもつ。図 3 に、TX39 プロセッサの場合のパイプライン制御シミュレータの例を示す。

このシミュレータは、パイプラインの状態をシミュレートし、メモリへの書き込みなどの、外部におよぶ機能がどの時点で有効になるかを設定するものである。各命令の機能（演算など）自体については、いっさい関知しないものである。以下に、パイプラインの制御シミュレータの主要な特長を示す。

- (1) パイプライン制御シミュレータは、パイプラインの各段（図 3 の IF, ID, EX, MA, WB）、およびメモリの動作をシミュレートするモジュール（キャッシュのモジュールとメモリインターフェースモジュール）から構成される。スーパスカラプロセッサの場合は、命令発行ユニットのモジュールも追加する。これらの各モジュールは、並列に動作すべきものである。
- (2) 各モジュールの動作を関数で実現する。クロックごとに、パイプラインを逆にスキャンする順序で（図 3 では、WB → MA → EX → ID → IF の順序で）この関数

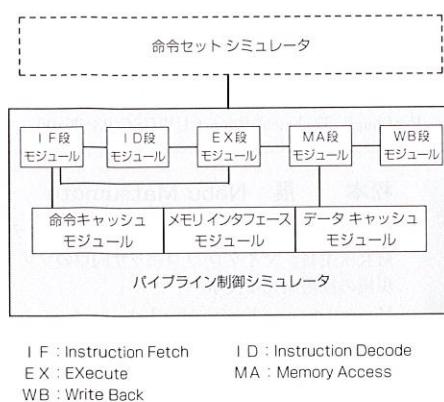


図 3. TX39 パイプライン制御シミュレータ パイプライン各段 (IF 段, ID 段, EX 段, MA 段、および WB 段) のモジュールとメモリ動作関連のモジュールから構成される。

TX39 pipeline-control simulator

を呼び出すことにより、パイプライン各段の並列動作を再現できる。

(3) モジュール動作として、データハザードおよび分岐ハザードにおける、ストール動作を記述する。

(4) キャッシュのヒット／ミスを判別するため、キャッシュのうちのアドレスだけを保持しておく。

以上のとおり、パイプライン制御シミュレータを実装し、命令セットシミュレータと組み合わせることにより、サイクルアキュレートなモデルの効率良い開発が可能になる。

4 TX39 プロセッサ シミュレーション モデル

3 章に記した、パイプライン制御プリデコード方式に基づき、32 ビットマイクロプロセッサ TX39 のプロセッサモデルを開発した。このモデルでは、機能と動作タイミングの両方をシミュレートするサイクルアキュレートなモードと、機能だけをシミュレートする高速な命令レベルモードの、二つのモードをサポートしている。一つのモデルで二つの使いかた（モード）が可能なことは、パイプライン制御プリデコード方式の長所の一つである。また、このモデルは、TCP/IP (Transmission Control Protocol/Internet Protocol) を介して SW デバッガと接続している。

ベンチマーク用いた場合の、このモデル単体の処理性能を表 3 に示す。処理時間の測定には、ホスト計算機として当社の AS7000-U1/140 (CPU は Ultra SPARC 143 MHz, メモリ 128 M バイト搭載) を、OS (Operating System) には Sun^(注2) OS 5.5.1 を用いた。表 3 によれば、サイクルアキュレートモードの処理時間は、命令レベルモードの 6~7 倍である。また、表 1 の処理性能を参考にすると、このモデルが充分高速であることがわかる。

このモデルを用いて測定された、いろいろなベンチマークの実行に要した実行サイクル数を図 4 に示す。ここで、実行サイクル数は、命令の実行段 (EX 段) におけるサイクル数を示す。図 4 で実行命令数が 600 以下の場合に、ヒル

表 3. TX39 モデルのシミュレーション性能
Simulation performance of TX39 model

ベンチマーク	実行命令数	命令レベルモード		サイクルアキュレートモード	
		処理時間 (s)	処理性能 (MIPS)	処理時間 (s)	処理性能 (MIPS)
ドライストーン	812,013	5.79	0.140	37.95	0.0214
ヒープソート	638,440	4.53	0.141	32.15	0.0199
8 クイーン	342,916	2.57	0.134	15.54	0.0221
エラトステネスのふるい	231,737	1.47	0.158	10.06	0.0230

(注 2) Sun は、米国 Sun Microsystems 社の商標。

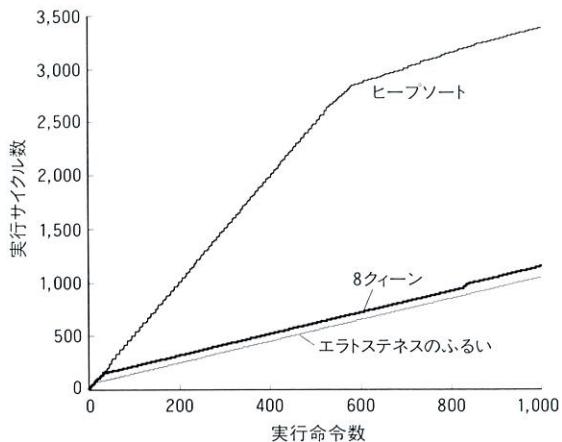


図4. TX39 モデルにより求められた実行サイクル数。ヒープソートでは、はじめにキャッシュミスが多発し、実行に多くのサイクル数がかかる。

Cycle count measured by TX39 model

ソートの実行が他より多くの実行サイクル数を要するのには、キャッシュのリフィル(refill)が多発しているからである。ここに示した実行サイクル数は、実機(TX39)の場合のサイクル数と完全に一致していることが確かめられている。

このモデルを米国 Mentor Graphics 社の協調検証ツール Seamless CVE^(注3)に組み込んだ場合の、画面例を図5に示す。図5の左側の大きなウインドウは SW のデバッガを、右側の大きなウインドウは HW シミュレータのユーザイン

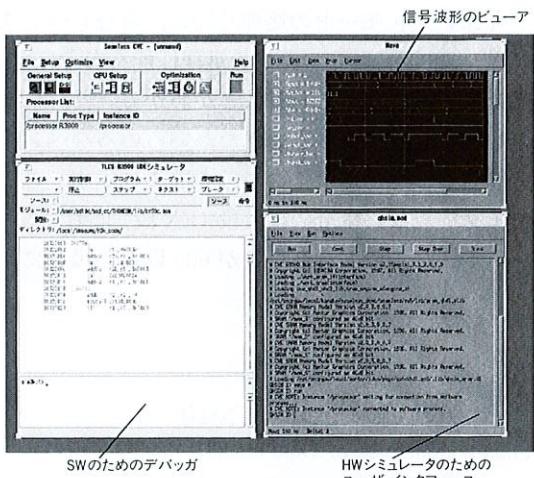


図5. HW/SW 協調シミュレーションのユーザインタフェースの例
HW と SW のデバッガにそれぞれ適したユーザ インタフェースを備えている。

Examples of HW/SW co-simulation user interface

(注3) Seamless CVE は、Mentor Graphics Corporation 社の商標。

タフェースを示している。図5のように、HW と SW のデバッガにそれぞれ適したユーザインタフェースを使用できることにより、デバッガ作業の効率向上が期待できる。

処理性能は、シミュレーションを行う HW の量に依存するため、一般に、モデル単体をシミュレートするのに比べ、シミュレーションスピードは低下する。したがって、現性能では、応用システム全体よりもデバイスドライバなどのシミュレーションやデバッガに適していると考えられる。

なお、当社では、Seamless CVE 用の TX39 の命令セットおよびサイクルアキュレートモデルを開発しており、今後、Seamless CVE による HW/SW 協調シミュレーション環境を顧客に提供していく所存である。

5 あとがき

HW/SW 協調シミュレーションに適したモデリング技術として、パイプライン制御プリデコード方式を紹介した。同方式に基づいて 32 ビットマイクロプロセッサ TX39 のシミュレーションモデルを開発し、サイクルアキュレート、かつ高速なシミュレーションが可能なことを実証した。また、このモデルと協調シミュレーションカーネルを用いることで、HW 試作前にデバイスドライバなどの SW のデバッガができ、システム全体の開発期間の短縮につながる。

今後、time-to-market(タイムリーな市場への対応)への要求がさらに厳しくなるにつれて、応用システム全体のシミュレーションの必要性が高まると考えられる。応用システム全体のシミュレーションのために、シミュレーションの性能向上にさらにつとめていくとともに、アナログ回路や、モータなどの機械系のシミュレーションを含む、より統合的な協調シミュレーション環境が必要になる。

文 献

- J.A. Rowson, "Hardware/Software Co-Simulation", Proc. 31st Design Automation Conference, pp.439-440, June (1994)
- "Seamless Co-Verification Environment User's and Reference Manual Software Version 1.0", Mentor Graphics Corporation, (1996)
- R.F. Cmelik, D. Keppel, "Shade: A Fast Instruction-Set Simulator for Execution Profiling", Technical Report UWCSC 93-06-06

松本 展 Nobu Matsumoto

マイクロエレクトロニクス技術研究所 システム LSI 技術研究所主務。マイクロプロセッサ向けのソフトウェア開発環境の技術開発に従事。

Microelectronics Engineering Lab.

徳吉 隆宏 Takahiro Tokuyoshi

マイクロエレクトロニクス技術研究所 システム LSI 技術研究所。マイクロプロセッサ向けのソフトウェア開発環境の技術開発に従事。

Microelectronics Engineering Lab.