

## 高速シミュレーションツール：リバース・コンパイラの開発

7P-6

鈴木慎一郎、竹内陽一郎

(株)東芝 情報処理・機器技術研究所

### 1. はじめに

新しく計算機を開発する場合（以下その計算機を「新機」と記す）、ハードウェア・シミュレータを用いて新機用プログラムの動作検証をすることがある。しかし、一般に、前記シミュレータでは、その実行に時間がかかり過ぎる。また、命令体系の見直しやコンパイラの検証等を行う場合、命令の動作レベルのシミュレーションが出きれば十分な場合が多い。そこで、新機の命令コードを前記シミュレータを用いずに高速に実行するためのツールとして、命令コードからその動作を実行（シミュレート）する高級言語ソース・プログラムを生成するツールすなわち「リバース・コンパイラ」を開発し、これを用いた新機コードの高速実行シミュレーションを試みた。本文では、その方法と特徴および効果について述べる。

### 2. リバース・コンパイラによる新機コード実行方式

図1に、新機用にコンパイルされた命令コードの実行シミュレーションの流れを示す。

まず、実行すべき新機用命令コード（アセンブリ・ソース）を用意する。命令コードは、直接記述したものか、または新機用命令コード（アセンブリ・ソース）を生成するクロス・コンパイラでコンパイルしたものである。この命令コードから、その中の命令の動作を各々行うC言語プログラム（以下、「Cソース」と記す）を生成、すなわち、コンパイルとは逆に命令コードからCソースを生成するものが「リバース・コンパイラ」である。これを用いて、新機の命令コードを次のようにして実行する計算機（以下、「ホスト」と記す）上で実行する。

Development of A High-speed Simulation Tool:  
Reverse Compiler

Shinichiro SUZUKI, Yoichiro TAKEUCHI  
TOSHIBA CORPORATION

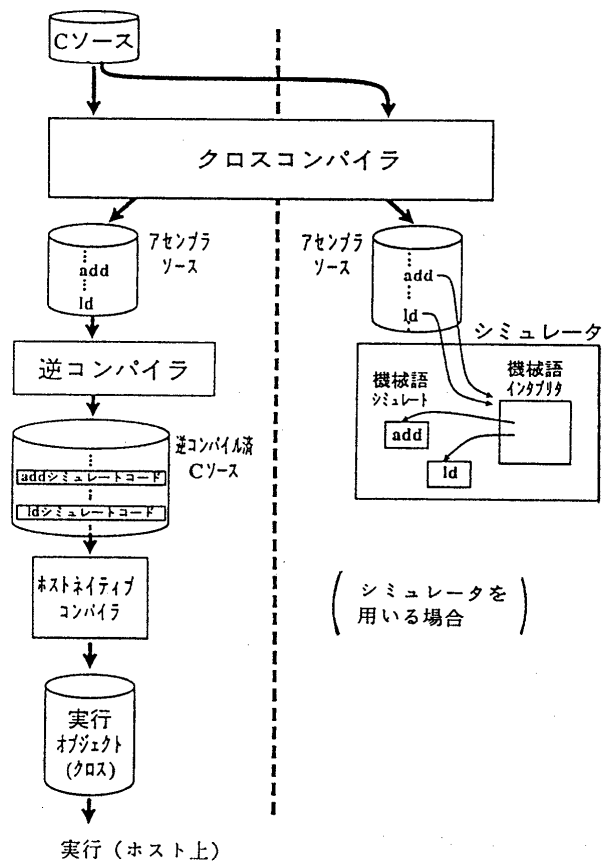


図1 リバース・コンパイラによるシミュレーション方式

新機の命令コードから、リバース・コンパイラを用いてCソースを生成する。このCソースを、ホストのコンパイラでコンパイル（及びリンク）し、ホスト用実行ロード・モジュールを作成する。これを実行することにより、ホスト上で新機用命令コードの実行シミュレーションができる。

### 3. 本方式の特徴

本シミュレーション方式による新機用命令コードの実行シミュレーションの特徴として、次の点があげられる。

- ①命令のフェッチおよびデコード処理が無く、その分高速。
- ②ソースオペランドの読出し、命令の実行、結果書込みの一連の処理を、ホスト用に最適化したオブジェクトで実行できる。

上記①については、ハードウェア・シミュレータを用いた場合の命令フェッチおよびデコード処理相当がリバース・コンパイラ内で済んでいることによる。②については、各々の命令の一連の動作をシミュレートするプログラムであるリバース・コンパイラの出力が高級言語ゆえ、ホスト上でそのコンパイル時に、ホストに見合った最適化を指定できることによる。

これらの特徴により、本実行シミュレーション方式では、ハードウェア・シミュレータを用いる場合に比べ、非常に高速に新しい計算機用にコンパイルされた命令コードの実行シミュレーションを行うことができる。

また、Cソースをホスト上でコンパイルするため、部分コンパイルが可能である。すなわち、プログラム中で検証したい部分のみ新機のコンパイラでコンパイルした命令コードのシミュレーション・コードで実行することも可能である。これにより、実行シミュレーションによるコンパイラの検証やアセンブリ・プログラムの検証等を更に効率的に行うことができる。

### 4. ホストのシステム・コール利用

実行しようとする命令コードがシステム・コールやライブラリ (unix-rte等、以下、単に「システム・コール」と記す) を呼び出している場合、それらも実行できなければ、その呼び出し時点までしか実行シミュレーションができない。しかし、入出力も含めて新機のシステム・コールをサポートするには、殆ど新機用のOSが動いていなければならない。そこで、システム・コールは、ホストのOSが提供するものを利用することとした。

新機のシステム・コールの代わりにホストのシステム

・コールを呼び出す場合、ホストのリンケージ規約 (レジスタ使用規約や引数渡しの方法等) に合わせて呼び出し、新機のリンケージ規約に合う様に返値を得なければならない。そのため、新機命令コードのシミュレーション・コードでは、リンケージ規約変換ルーチンを介してホストのシステム・コールを呼び出すこととした。この変換ルーチンはホストのアセンブラで記述し、そのホストのオブジェクトをまとめてリバース・コンパイラ専用ライブラリとし、ホストでのリンク時 (unix) に指定する。

こうしてシステム・コールをサポートすることにより、多くのホストのコマンドを新機のコンパイラでコンパイルし、新機の命令コードでの実行結果とホストのコマンドの実行結果を比較することにより、新機のコンパイラの検証を行うことも容易にできる。また、新機のアセンブリ・ソースの動作テストをする場合、システム・コールやライブラリを用いて、動作状況の出力などを行うテスト・プログラムを容易に作成できる。

### 5. 実行性能

Dhrystoneベンチマークを新機用コンパイラでコンパイルした命令コードの実行シミュレーションを、ハードウェア・シミュレータを用いた方式、リバース・コンパイラを用いる本方式の各々で実施した。その結果、本方式の方が、約250倍も高速に実行できるという結果を得た。

### 6. おわりに

今回リバース・コンパイラを開発し、これを用いた方式の機能および性能上の有効性を確認した。

パイプラインやキャッシュ、TLB、バスの動作等よりマイクロなシミュレーションには従来のハードウェア・シミュレータが必要と思われる。しかし、命令の動作レベルまたはプログラム単位の実行シミュレーションには本方式が非常に有効であり、そのシミュレーションによるコンパイラの検証には特に有効である。今後も本方式をコンパイラやアセンブリ記述プログラムの効率的な検証等に適用していきたい。