

論理シミュレータのための高速コンパイル方式

5N-2

庄司 稔 広瀬 文保
(株) 富士通研究所

1. はじめに

論理回路の大規模化とともに増大する論理シミュレーション実行時間の短縮のために、論理シミュレータの高速化が研究されている。しかし、論理シミュレーション実行のためにはその前処理としてコンパイルを実行する必要がある。これは、回路構造を記述したデータをシミュレータの理解可能なデータに変換するものである。論理シミュレータの高速化が専用マシンなどにより達成された現在、このコンパイラの高速化が要求されている。

本稿では、リンク時に各モジュールの回路データをそのまま再利用する高速コンパイル方式について述べる。

2. 従来方式

コンパイルの高速化の手段として、回路をモジュール毎に個別にコンパイルするアプローチがなされている。これは、回路の各機能モジュールについてコンパイルを実行して回路データを作成した後、その回路データをリンクして目的とする回路の回路データを作成するものである。この方式では、一部のモジュールの設計変更時にはそのモジュールのみ再コンパイルし、全体をリンクすればよい。このため、回路全体について改めてコンパイルし回路データを作成する場合と比較してコンパイルの実行時間を短縮することが可能となり、一部のモジュールの変更時の再コンパイル時間を短縮することも可能となる(図1)。しかし、このアプローチではリンクのための時間を短縮することが出来ない。

従来はリンク時に各モジュールの回路データよりゲート間の接続情報を読み取り回路全体のゲート間の接続を設定して目的とする回路データを作成していたため、リンクに非常に時間がかかるものとなっていた。このため、モジュール

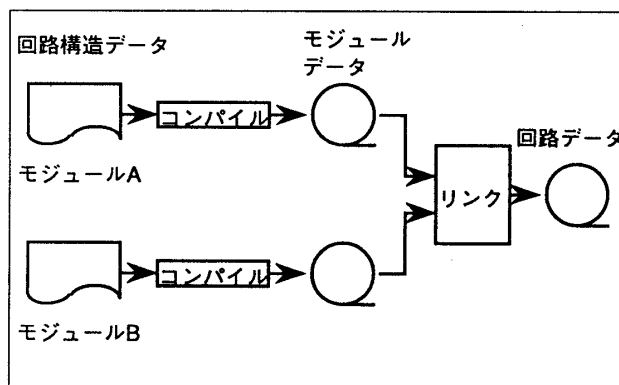


図1 分割コンパイル

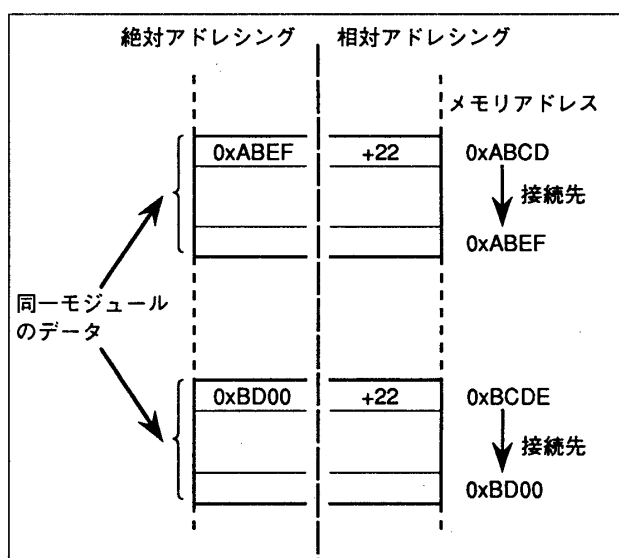


図2 相対アドレッシング

ルのコンパイル時間の短縮のみでは十分な高速化が得られなかった。

3. 高速リンク方式

本方式では、次のような機能を論理シミュレーション専用マシンに搭載し、各モジュールの回路データをリンク時にそのまま使用することでコンパイルの高速化を図る。

(1) 相対アドレッシング

回路データはゲートの種類とゲート間の接続情報とで構成されている。これらの情報は各ゲートに付けられた識別記号(数字など)

を指定することで参照される。一般にはこの識別のための数値は図2に示すようにその情報を記録しているメモリのアドレスを使用する。従って、モジュールのデータが相対アドレスによって参照されることで、このモジュールのデータはその割当先メモリのアドレスには左右されない、リロケータブルなものとなる。

(2) 間接アドレッシング (ポインタ)

各モジュールの入出力先の情報はそのモジュールデータのメモリを介して参照される必要がある(図3)。しかし、モジュールのデータの作成時にはそのモジュールの接続先や接続数(ファンアウト数)は不明であるため、必要なメモリ領域を確保することは不可能である。この問題を解決するために、間接的にメモリを参照する間接アドレッシング(ポインタ)機能を使用する。この機能により、モジュールのデータの作成時には1個のポインタを書き込むだけのメモリ領域を確保しておけばよいことになる。

4. 高速化率

図4は回路の部分変更時の再コンパイル時間を従来のリンク方式を使用した場合と、ここで述べた方式を使用した場合とで比較を行なった結果である。グラフは、本方式を使用した場合の70万ゲートの回路の再コンパイル時間を1として表している。従来方式では、この再コンパイルに要する時間のほとんどをリンクに要していた。一方、本方式では従来方式に比べ350万ゲートの場合で50分の1の時間で再コンパイルが実行されており、リンクが高速化されていることがわかった。

また、本方式は専用マシンのみならず、汎用計算機上で動作するソフトウェアのシミュレータにも適用可能である。この場合、相対アドレス、間接アドレスで示されたデータを絶対アドレスに変換するための計算をする必要があるが、このオーバーヘッドについては数%程度と見積られ、本方式により論理検証に要する時間はかなり短縮されることとなる。

5. まとめ

論理シミュレーション専用マシンに、相対アドレッシング、間接アドレッシング(ポインタ)機

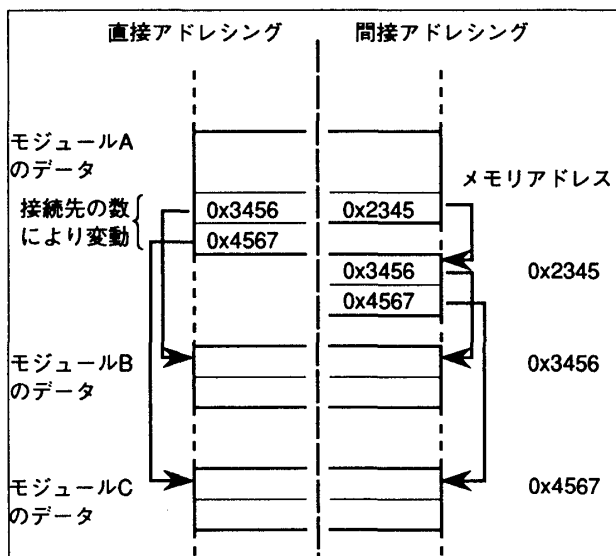


図3 間接アドレッシング

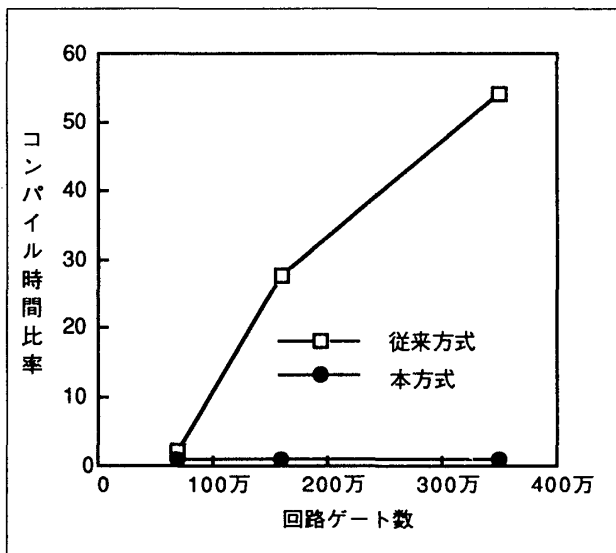


図4 コンパイル時間比率

能を搭載することで、回路データのコンパイルが高速化されることを示した。さらに、本方式は論理シミュレータ全般に適用可能であり、その場合でも論理検証に要する期間を短縮出来ることを示した。

参考文献

- [1] 三善、溝上、高木、名倉、長藤：部分更新方式による論理コンパイル時間の短縮、情報処理学会第40回全国大会、2M-1、(1990.3).
- [2] Lewis, D.M.: Hierarchical Compiled Event-Driven Logic Simulation, Proc. of ICCAD '89, pp. 498-501 (1989).