

Cコンパイル方式のHDL統合シミュレーション

著者	井上 恒司, 檀 良, 若林 哲
出版者	法政大学計算センター
雑誌名	法政大学計算センター研究報告
巻	8
ページ	55-58
発行年	1995-03-31
URL	http://doi.org/10.15002/00024697

C コンパイル方式の HDL 統合シミュレーション

井上 恒司 檀 良

法政大学工学部[†]

若林 哲

東芝マイクロエレクトロニクス[‡]

近年、デジタル回路の設計における回路記述のため、VHDL と VerilogHDL による混在した HDL 記述が使われる傾向にある。このような状況に対応するため、共通の記述として C 言語を用い、C 言語によって表現された回路記述が C コンパイラの入力データとして使用される。本稿で我々は、C 言語による回路の効果的な表現方法とその表現に適したシミュレーションの方法を報告する。

1. はじめに

デジタル回路の設計は、従来の回路図を用いた手法から、設計向き言語 (Hardware Description Language = HDL) を用いた手法が中心となりつつある。設計工程中で言語表現の回路は、論理合成ツール、シミュレーションツール、レイアウトツールなどの入出力として使用される。現在数多くのツールが設計現場で使用されているが、市販されているツールの入力となる標準の HDL は事実上 VHDL と VerilogHDL であると言える。そしてこれまでこの 2 つの HDL はそれぞれ独立して使用されて来て、この 2 つの HDL を併用して 1 つの設計を行う事はほとんど無かった。しかし VHDL と VerilogHDL がこの先も併存する事は確実で、このような事態は近い将来頻発する可能性がある。これに対し最近、シミュレーションバックプレーンなどが提案されている¹⁾²⁾。

我々はそのような回路のシミュレーションの 1 方法として、C 言語の環境を利用して HDL を併用して記述された回路のシミュレーションを可能にするシステムを構築している。本稿では、このシステムの構築過程を報告する。

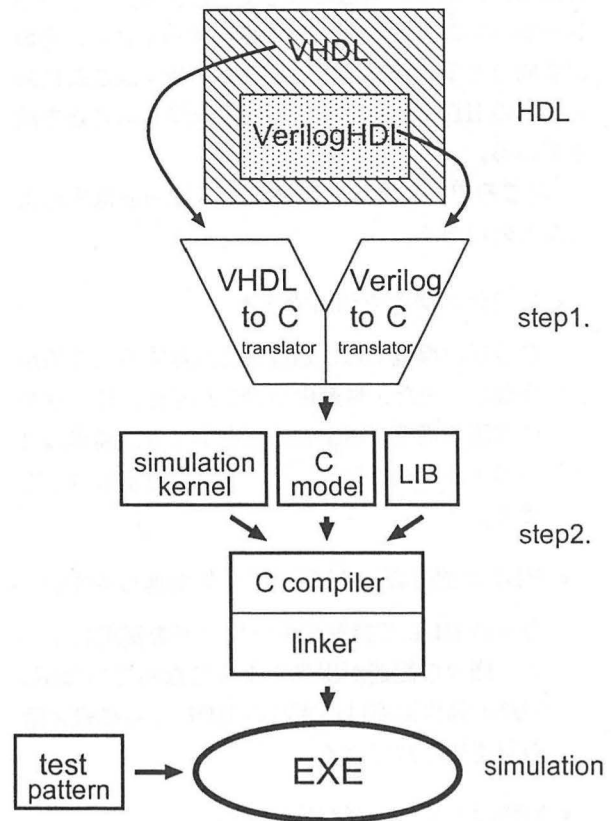


Fig. 1 全体の流れ

2. 対象・目的

VHDL と VerilogHDL の MIX レベル記述による回路のシミュレーションを目的としている。そのために HDL の表現に適した C 言語のデータ構造とシミュレーション方式を考慮している。

3. システムの構成

構築しているシステムの全体の流れを 図 1 に示す。図の上から順にそれぞれ次のことを表している。

- 設計回路の HDL 記述
- HDL から C 言語への変換処理

[†]〒 184 東京都小金井市梶野町 3-7-2

[‡]〒 210 神奈川県川崎市

- C 言語記述の回路、カーネル、ライブラリ
- C 言語の環境でのコンパイル、リンク処理
- テストパターンを与えてシミュレーション

我々が考慮しているのは、図 1 の step1 と step2 の部分でそれぞれ次のような処理となる。

3.1. step1:HDL から C 言語への変換

VHDL と VerilogHDL では扱うことの出来るデータや記述方法も異なっている³⁾。これまでにされてきた HDL から C 言語への変換は、各々の HDL を単一で使用して記述した回路のシミュレーションのために、別々の構造になっていた。また、シミュレーションの制御も個々の特徴を持っていて、そのまま統合することは困難であった。我々はこれに対し両方の HDL に適した C 言語に変換することを試みている。

ところで、C 言語に変換するメリットは次のようなものがある。

- C 言語の環境を利用できる
C 言語の環境では、処理系に依存するところが少ない。また、最適化やコンパイル、リンクを C 言語の環境で済ませることができ、高速なコンパイル方式でシミュレーションをする事ができる。
- HDL の動作記述が C ライクで変換しやすい
各々の HDL で特有のキーワードを使用していて、様々な記述が出来るようになっているが、回路の動作記述はいずれの HDL でも高級言語の記述法に似ている。
- 回路のモデリングがしやすい
C 言語に変換された後のデータ構造や記述法を知れば、回路の HDL 記述が部分的になくてもいきなり C 言語で記述し使用することが出来る。

そのほかに、実際の回路と C 言語の関数が入出力の関係や階層構造の記述が似ているという理由がある。つまり回路モジュールではある出力信号の値は、組み合わせ回路では入力信号、順序回路では入力信号と回路の内部状態から決まる。一方 C 言語の関数では、関数からの戻り値は関数へ渡す引き数

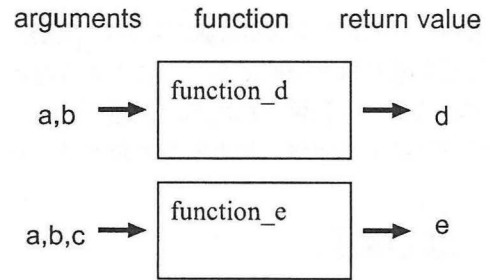
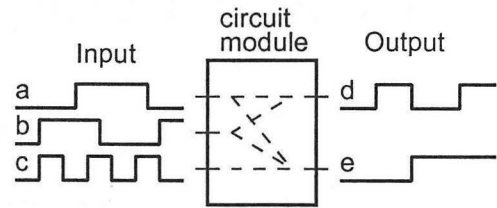


Fig. 2 回路モジュールとモデル化した関数の入出力

や関数内部の静的変数から計算することが出来る。この両者の違いである時間の概念は、関数内部の値をシミュレーションカーネルで制御するように考えている。この様な類似点を考え、回路中の 1 つの出力信号を 1 つの関数の戻り値として扱うように変換することを考えている。この関数では入力信号列から、結果である出力信号列を返すが、このとき入出力の関係だけでなく時間の考慮も行うようにする。これは、実際の回路では遅延時間が存在し、実用的なシミュレーションではそれを考慮することが必要不可欠であるからである。また、多入力多出力のモジュールも出力一本毎に分解することで、その出力に関係する入力のみを考えればよくなるというメリットも生じる。この様子を 図 2 に示す。

実際の回路ではすべてのモジュールが並列に、同時に動作している。我々が考案している C コンパイル方式のシミュレーションでは、関数として扱う各出力線を一度に 1 つずつ計算することになるので、コールされていない関数ではシミュレーション時間に応じた信号の値、回路の状態などを記憶しておく必要がある。

3.2. step2:シミュレーションの制御

図 1 の step1 で各信号線の値が関数と戻り値として計算する形式になっているので、その関数をコールする側をここでは考える。そのためシミュレーションカーネルがメインの関数として、実行時の制

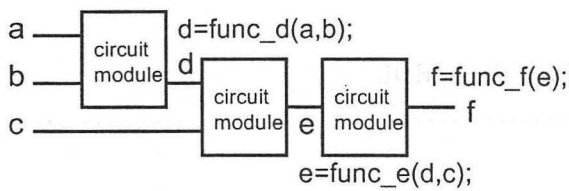
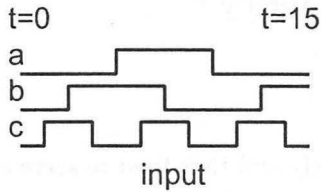


Fig. 3 回路モジュールと関数



```

for(t=0;t<=15;t++){
    d[t]=func_d(a,b);
    e[t]=func_e(d,c);
    f[t]=func_f(e);
}

while(1){
    if(flag_d)
        &d=func_d(&a,&b);
    if(flag_e)
        &e=func_e(&d,&c);
    if(flag_f)
        &f=func_f(&e);
}
    
```

(a) (b)

Fig. 4 関数のコール回数の比較

御をする。先述の通り変換した関数は、入力がかればそれに応じた出力を返すようになっているが、関数は一度に一つしかコールできないので、入力に近い方から計算していけばよいということは明らかである。ところがこの計算順序の入れ替え処理は回路規模が大きくなるにつれて処理時間が増大するし、また順序回路が入ることで一意に決めるのが困難な場合がある。これをせずに済ませるために、シミュレーションカーネルは関数に変換したモジュールの入力に変化があるかどうかをチェックし、入力値の変化があった場合のみその関数をコールするようにする。これにより必然的に入力側から計算するようになる。この様子を 図 3 に示す。この場合ソートしていれば関数の記述順序にコールし、入力信号 a,b,c から最初に d を計算し、以下 e,f の順に計算する。これに対し我々の方法では関数の記述順序に関わらず、関数のコールは d,e,f の順序で行われる。ところがこれでは、関数を呼ぶかどうかのチェックのために処理時間がかかることになるので、チェッ

クの数自体も減らすことが必要である。この点は 図 4 のように考える。(a) の場合は、すべてのモジュールが同時に動作している様子を忠実に表現しているが、関数のコール回数が多くなる。我々は、(b) のように一連の入力信号列から一連の出力信号を計算するようにしている。この方法だと、(b) は (a) の場合と比べて関数のコール回数は少なくすむことが分かる。順序回路の場合は 図 4(a) の場合と比べてコール回数はあまり減らない時もあるが、この場合も関数のソート処理は不要である。

シミュレーション結果のレポートの指示もここでするように考えているが、具体的なことは未定である。

4. まとめ

我々が構築している C 言語の環境を利用した HDL 統合シミュレーションにおける処理手順と各手順で考慮している事柄を簡単に説明した。今後これらの処理の実現に向けさらに細部まで厳格に仕様を決定し、同時に実際に作成していく。

参考文献

- 1) 新出 浩丈, 中道 宏治, 吉江 啓, “VHDL を全面的に採り入れた LSI/システム用 EDA を構築,” 日経エレクトロニクス, (no.618), pp.85-98, 1994
- 2) 中山 俊一, 小島 郁太郎, “HDL 本格導入が回路設計工程の変革迫る,” 日経エレクトロニクス, no.623, pp.75-95, 1994
- 3) 安浦 寛人, 神原 弘之, “ハードウェア記述言語の比較,” 情処研報 DA63-5, pp.31-38, 1992

キーワード

論理シミュレーション、ハードウェア記述言語、VHDL、VerilogHDL

.....

Summary

Logic Simulation using a unified HDL based on a C-compilation method

Kouji INOUE Ryo DANG

College of Engineering, Hosei University †

Satoshi WAKABAYASI

Toshiba microelectronics ‡

Recently, both VHDL and VerilogHDL are being used extensively and thus tend to serve as standard tools for the design of logic circuits. To cope with this situation, we propose in this paper a simulation system where C language is used as the common vehicle for circuit description and input data for the C compiler.

Key Words

logic simulation, hardware description language, VHDL, VerilogHDL

†3-7-2, Kajino-cho, Koganei-shi, Tokyo 184, Japan

‡Kawasaki-shi, Kanagawa 210, Japan