

VHDLシミュレーションにおける  
バス入出力方向判定方式

6L-3

山元 渉 伊藤雅樹 高嶺美夫 清水嗣雄  
(株)日立製作所 中央研究所

1.はじめに

VHDLなどのハードウェア記述言語による高位記述が一般的になっているが、その一方で、高性能CPUなどの場合、性能にクリティカルなブロックについてはゲートレベルの論理設計も行われている。この不良追跡のためにはブロック単体のゲートレベル・シミュレーションが処理速度、扱い規模の点から現実的であるが、テストパタンの作成が困難である。理由は、回路全体の外部インタフェース仕様と比べて回路内部のブロック間インタフェース仕様はより複雑になること、ブロック単体の動作を理解しなければならないことが挙げられる。そこで、図1に示すように高位記述の全体シミュレーションから波形を抽出し、ブロック単体用パターンとして利用することが考えられる。

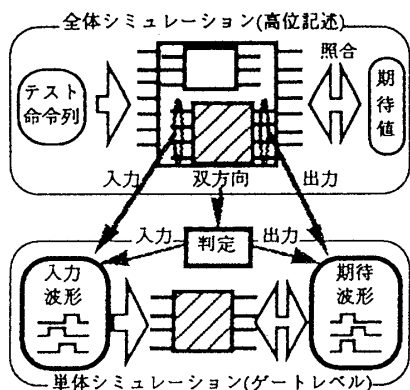


図1.単体シミュレーション用波形抽出

抽出波形を利用するには、これを入力であれば入力波形へ、出力であれば期待波形に分類しなければならない。ここで、バスに接続する入出力の両方に用いられる双方向ポートについては、シミュレーション実行時に初めて方向が判明するため、抽出の瞬間における入出力方向を判定する必要がある。

本報では、VHDLシミュレーションにおいて、バスに接続する双方向ポートに関する、データの入出力方向を判定する方法を提案する。

An Approach to Detection of Input/Output Direction on Bidirectional Port in VHDL Simulation

Wataru Yamamoto, Masaki Itoh, Yoshio Takamine, Tsuguo Shimizu

Central Research Laboratory, Hitachi, Ltd.

1-280, Higashi-koigakubo, Kokubunji-shi, Tokyo 185, Japan

2.バス入出力判定のモデル

図2に入出力判定問題のモデルを示す。複数のブロックが接続するバス、判定対象であるブロック、および双方向ポートが示されている。判定対象ブロックの内部には、バスに対する2つのソース、およびバスの値を入力とする論理が含まれている。

設計に不良がなければ、バスに接続するブロックのうちデータを出力するのは1つで、それ以外のブロックでは、内部に含まれるソースはZ(高抵抗値)を出力する。

したがって、バス値がZとなるバス・フローティングを除くと、入出力判定では、対象ブロック内部ソースの出力がZなら入力、バスと同値ならば出力とみなすことができる。

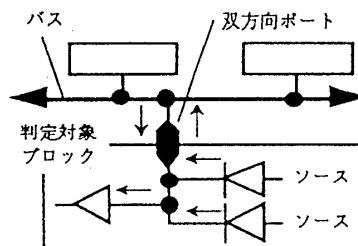


図2.入出力判定問題のモデル

3.提案手法

3.1 回路変更による仮想論理の生成

ここでは、入出力方向判定のためにシミュレーション対象回路を変更する手法を提案する。変更のポイントは下記の2つである。

- (1) 代替信号の設置：双方向ポートへのブロック内部での接続を全て新設した代替信号へ移す。
- (2) 仮想論理の挿入：代替信号と双方向ポートを仮想論理で接続する。仮想論理はシミュレーションでのみ設計回路に追加される論理である。

図3は図2の回路を変更した結果を示している。

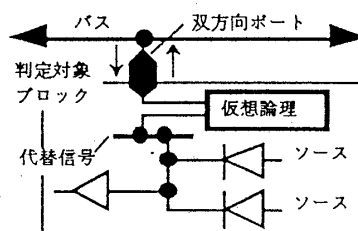


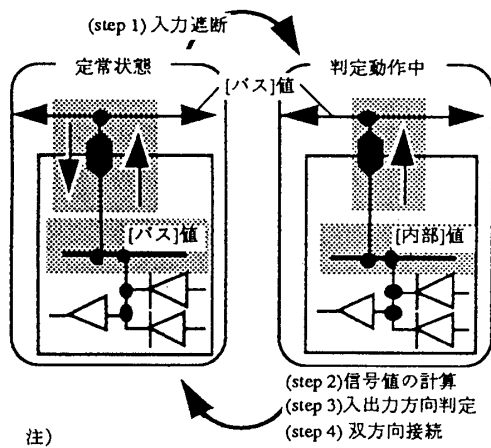
図3.回路変更（仮想論理，代替信号の追加）

### 3.2 仮想論理の動作

図3の回路がシミュレーションされることによって、双方向ポートの入出力方向が判定されることを以下で説明する。判定アルゴリズムでは、バス値の切り替わりの瞬間において、代替信号の値をブロック内部のソースの出力値に設定するシミュレーションサイクルを設け、その値がZであれば入力、それ以外ならば出力方向とする。

仮想論理の動作は下記4ステップからなり、図4が説明である。また図5にVHDLソースを示す。

- (step1) 代替信号へのバスからの入力を遮断する。
- (step2) バス、代替信号の値を再計算する。
- (step3) 上記の2値を比較し、入出力を判定する。
- (step4) 代替信号とバスを再び双方向に接続する。



注) [バス]値：全てのソースによって決定される値  
[内部]値：対象ブロック内部のソースのみの決定値

図4.仮想論理の動作、および作用

(step1)(step2)の結果、回路状態は図4左側の定常状態から右側の判定動作中へ移行する。定常状態では、ポートは双方向に信号を入出力し、その結果、代替信号の値はバスに接続する全ソースによって決定される値となる。判定動作中では、ポートは出力のみを行うため、代替信号の値がブロック内部のソースのみによって決定される値をとる。(step3)ではポートと代替信号の値を比較することで判定を行う。判定基準の詳細は図5(step3)の1~3の条件文のとおりである。(step4)では回路を定常状態に戻す。

本手法において動作の正当性を保証するため、下記2つの条件が満たされていなければならない。

- (1) ワイヤード論理を計算するリゾリューション関数 $rf()$ に結合則 $rf(a,b,c)=rf(rf(a,b),c)$ が成立する。
- (2) 設計論理で信号アトリビュート `transaction`, `quiet`, `active` を使用した結果、仮想論理の動作に干渉することがない。

```
JUDGE: process begin
  wait on io_port'transaction,
        alt_sig'transaction; } 動作開始条件
  alt_sig <= 'Z';                (step1)-1
  wait until alt_sig'quiet;      (step2)-1
  io_port <= alt_sig;            (step1)-2
  wait until io_port'quiet;      (step2)-2
  if io_port = 'Z' then          (step3)-1
    バス・フローティング;
  elsif alt_sig = 'Z' then 入力方向; (step3)-2
  else 出力方向;           (step3)-3
  end if;
  alt_sig <= io_port;           (step4)
  wait until alt_sig'quiet;
end process JUDGE;
注) io_port: 双方向ポート, alt_sig: 代替信号
```

図5.仮想論理のVHDLソース

### 3.3 実行オーバヘッド

KUE-CHIP[1]を対象にVHDLシミュレータでオーバヘッドを計測した。全体の14%のイベントが発生する双方向ポートを判定したところ、イベントは29%増加したが、波形抽出を含む全体の処理時間については7%程度の増加にとどまった。

オーバヘッドの要因として下記3点が挙げられる。

- (1) 代替信号のイベント：判定対象の双方向ポートに発生するイベント数の2倍に及ぶ。
- (2) リゾリューション計算処理の増加
- (3) 図5(step2),(step4)のwait文によるシミュレーションサイクル：3サイクル/イベント増加する。

今後'93版VHDLが利用可能となれば、新規追加の信号アトリビュート`DRIVING_VALUE`を利用し、上記のオーバヘッドを低減できる見込みである。

### 4.おわりに

提案手法は、VHDL記述を変更することのみでバス入出力方向判定を実現している。したがって、'87版VHDLをサポートする代表的なシミュレータにおいて利用可能である。また、記述変更は3.1節で述べたように設計論理の知識が不要であるため、構文解析ツールを利用して変更処理を自動化することも可能である。入出力の変化を表す値をとる信号を回路に作り込むなどの方法では、変更を行えるのは設計者に限られ、大きな労力が必要である。

また、仮想論理はブロックの出力値を個別に観測できることから、バスに接続する双方向ポートの2つ以上が出力の場合を検査することで、バス衝突の解析にも応用できる。

<参考文献>

[1] 神原,安浦:計算機教育用コンピュータの開発とその応用,情報処理,Vol.33,No.2,pp.118-127(1991).