

## 3X-6

モジュールジェネレータにおける  
回路モデル生成手法

片井 ちひろ 津田 和彦 田中 宏 岡崎 芳

三菱電機株式会社

## 1. はじめに

高集積化、高性能化が進むLSIの設計効率向上を図るためのツールとして、ROM、RAM、PLAなど機能モジュールを自動生成するモジュールジェネレータが注目を集めている。

モジュールジェネレータ(以下、ジェネレータと略す)は、機能モジュールのレイアウトデータを生成するだけでなく、機能モジュールを含むチップ全体の検証やテスト生成に使用するCADツール用のデータ(以下、回路モデルと呼ぶ)を生成する必要がある。

CADツール用の回路モデルには、レイアウトから抽出したトランジスタレベルのネットリストがあるが、それでは目的とする検証などに不適当な場合がある。たとえば、ROMやRAMなどの論理シミュレーションを行なう場合、トランジスタレベルのネットリストを用いるのはシミュレーション時間の点からみて不適当である。したがって、ジェネレータは目的に応じた回路モデルが生成できなければならない。

我々は、ジェネレータにおいて目的とする検証などに適した回路モデルを生成するツールを開発した。本稿では、そこで用いた回路モデルの生成手法について述べ、開発した回路モデル生成ツールの概要を報告する。

## 2. ジェネレータにおける回路モデルの生成手法

一般に、CADツールは回路モデルとして素子間の接続情報を必要とするが、論理シミュレータの場合は機能や動作を記述した機能記述を扱えるものもある。したがって、ジェネレータは、素子間の接続関係を表現した回路モデルと、機能や動作を表現した回路モデルのいずれでも生成できる必要がある。

機能記述とは各論理シミュレータがもつ機能記述言語を用いて機能や動作を記述したデータである。また、一般に、各CADツールは素子間の接続関係を記述するための構造記述言語を用意している。したがって、言語で記述した回路モデルを生成する機能がジェネレータに備われば、素子間の接続関係を表現した回路モデルと、機能や動作を表現した回路モデルの両方を生成できる。

言語で記述した回路モデルをジェネレータが生成する場合、次の2点が問題となる。

## (1) パラメータに応じた回路モデルの生成

ジェネレータが生成する機能モジュールの回路モデルは、ユーザが指定するパラメータに応じ、レイアウトデータと対応のとれたものでなければならない。しかし、パラメータの値はユーザが指定するまで決定しないので、パラメータに応じた回路モデルをあらかじめ作成しておくことはできない。ここでいうパラメータとは、ビット数やワード数など回路規模を示す数値、ROMやPLAなどの論理、入出力ラッチの有無など回路構成を変更するオプション、などである。

## (2) 多様な回路モデルの生成

ジェネレータは論理シミュレータ、テスト生成ツールなど複数のCADツール用の回路モデルを生成する必要がある。しかし、各CADツールは回路モデルを記述する独自の言語をもっており、ジェネレータは各々の言語で記述した回路モデルを生成しなければならない。しかも、使用されるCADツールに追加や変更があった場合、新たな言語で記述した回路モデルが必要になる場合がある。ジェネレータはこのような言語の追加や変更にも容易に対応できなければならない。

これらの問題を解決するため、図1に示すフローで回路モデルの生成を行なう。ジェネレータで生成する機能モジュールの回路モデルは、各CADツールがもつ言語を用いて記述する。ただし、この記述中にはユーザがパラメータを指定するまで確定しない部分が存在する。この確定しない部分を含む回路モデル(以下、パラメータ化した回路モデルと呼ぶ)とパラメータから、記述編集処理系が確定した回路モデルを生成する。

パラメータ化した回路モデル中の確定しない部分を記述するためには、次に説明する“置換変数”と“記述編集文”を用いる。

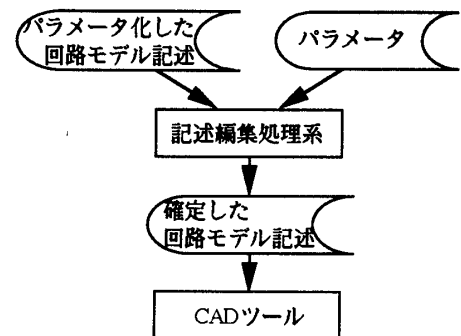


図1 回路モデル生成の概略フロー

Generating Method of Circuit Models for Module Generator

Chihiro KATAI, Kazuhiko TSUDA, Hiroshi TANAKA, Kaoru OKAZAKI

Mitsubishi Electric Corporation

“置換変数”

ビット数、ワード数、論理などパラメータそのものと置き換えられる変数。

“記述編集文”

ビット数やワード数などに応じて記述が繰り返される部分を表現する繰り返し文、および、回路構成の選択により記述の追加や削除が行なわれる部分を表現する条件判断文。

記述編集処理系は、“置換変数”と“記述編集文”を用いてパラメータ化した回路モデルのうち、“置換変数”と“記述編集文”をパラメータで置換・編集し、その他の部分を元の書式のまま保持することにより、確定した回路モデルを生成する。

このようなフローで回路モデルを生成する際、“置換変数”と“記述編集文”については、パラメータ化した回路モデル中で正しく認識できるように、識別子をもたせる必要がある。この識別子はCADツールがもつ言語それぞれについて定義しなければならない。認識された“置換変数”と“記述編集文”に対する処理は、各言語の仕様に影響されないので、識別子を外部から定義することにより、記述編集処理系を共通にすることができる。

3. 回路モデル生成ツール

前章で述べた手法を用いた回路モデル生成ツールを開発した。このツールのプログラム構成を図2に示す。

パラメータ編集プログラムは、ユーザの指定したパラメータの正当性をチェックし、記述編集プログラムで取り扱う形式に変換する。

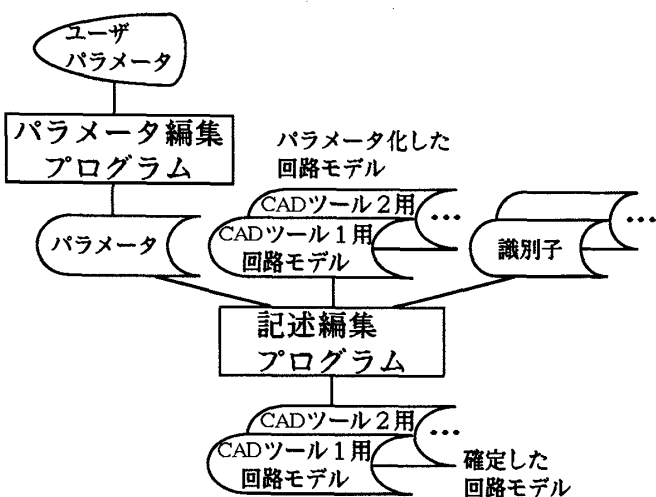


図2 回路モデル生成ツールの構成

記述編集プログラムは、CADツールごとにあらかじめ準備してあるパラメータ化した回路モデルをパラメータに応じて編集し、確定した回路モデルに変換する。

“置換変数”と“記述編集文”の認識は、あらかじめ識別子ファイルに記述しておいた識別子を用いて行なう。

図3に、このツールの適用例として、パラメータ化した記述、パラメータ、および編集後の確定した記述を示す。これは、乗算器について当社開発のミックスドレベル論理シミュレータ用の機能記述を生成した例である。この例では、“置換変数”と“記述編集文”の先頭に“\_”（アンダースコア）を記述し、その他の部分と区別できるようにしている。

```

DEFINE : _MPYNAME ;                MPYNAME = MPY1;
      : ;                          XNUM = 4;
      : ;                          YNUM = 4;
INPUT  : ?X[_XNUM-1:0],           IL = 1;
      : ?Y[_YNUM-1:0]             ;
      : _#if(_IL=1)#[,TI]#       ;
      : ;                          (b) パラメータ
EXECUTE :
      : _#if(_IL=0) #[ 'INPUT LATCH : N'
      : F_TI=0 ;
      : V_TI=1 ;
      : INITI=1 ;]#
      : _#else #[ 'INPUT LATCH : Y'
      : F_X=CHX(TI) ;
      : F_TI=F_X ;
      : IF (F_TI) THEN
      : V_TI=VAL(TI) ;
      : IF (V_TI) THEN
      : INITI=1 ;
      : ENDIF ;
      : ENDIF ;]#
      : ;
      : DEFINE : MPY1 ;
      : ;
      : INPUT : X3, X2, X1, X0,
      : Y3, Y2, Y1, Y0
      : , TI
      : ;
      : EXECUTE :
      : 'INPUT LATCH : Y'
      : F_X=CHX(TI) ;
      : F_TI=F_X ;
      : IF (F_TI) THEN
      : V_TI=VAL(TI) ;
      : IF (V_TI) THEN
      : INITI=1 ;
      : ENDIF ;
      : ENDIF ;
      : ;
      : (c) 編集後の確定した記述
    
```

(a) パラメータ化した記述

(c) 編集後の確定した記述

図3 回路モデルの生成例

4. おわりに

モジュールジェネレータにおける回路モデルの生成手法について述べた。この手法を用いた回路モデル生成ツールを、ROM、RAM、PLA、乗算器の各ジェネレータにおいて、当社開発の論理シミュレータおよびタイミング検証ツール用回路モデル生成に使用し、実用的であることを確認した。