

スタンダードセルLSI統合レイアウトシステム (6) ——パラメトリックセル自動生成——

6R-9

小山正弘
ソニー株式会社 半導体事業本部

1. はじめに

LSIの大規模化に伴いROM/RAM/PLA等を組み込む機会が増加している。しかし、従来のスタンダードセル設計方式では、あらかじめ固定化されたライブラリより所望のセルを選択する為冗長度が大きく、また希望のセルが無い場合、新規設計を行うことになり、LSI開発期間の長期化をきたす。

そこで、アドレス幅、データ幅等のパラメータを入力する事で、最適なサイズのROM/RAM/PLA等を生成できるパラメトリック・セル・コンパイラを開発した。

さらに、セル開発者が自由に最適なパラメトリック・セル・コンパイラを作成するパラメトリック・セル・コンパイラ開発用ツールとして、パラメトリック・セル・コンパイラ・コンパイラをあわせて開発した。

本機能は、単独でも使用可能であるが、スタンダードセル設計システムの一機能として組み込まれ、柔軟なスタンダードセル設計システムを構成している。

本文では、パラメトリック・セル・コンパイラ・コンパイラについて述べる。

2. 概要

スタンダードセル統合設計システムにおける、パラメトリック・セル・コンパイラの位置付けを図1に示す。

LSI設計時は、パラメトリック・セル・コンパイラに必要なパラメータを定義ファイルに定義する。本パラメータは、論理シミュレータと共に管理され、システムの一貫性を保証している。この定義ファイルを入力とし、パラメトリック・セル・コンパイラはシミュレーションの為のシミュレーション・モデル、及びレイアウトの為のフィジカル・パターンを生成し、カスタム・セル・ライブラリへ登録する。これらのライブラリを参照し、シミュレーション及びレイアウトが行われる。

一方、パラメトリック・セル・コンパイラ・コンパイラは、タイル形式を採用しており、コンパイラ・コンパイラ部とジェネレータ部からなる。

パラメトリック・セル・コンパイラ開発者は、後述するパラメトリック・セル定義言語を用い、タイル(プリミティブ)の相対的な配置構造を記述し、コンパイラ・コンパイラにより、パラメトリック・セル・コンパイラを作成する。

3. システム構成

本システムの構成を図2に示す。
プリミティブ・パターンから、ルール・ファイルに定義されたルール(デザイン・ルール、レイヤ対応表等)

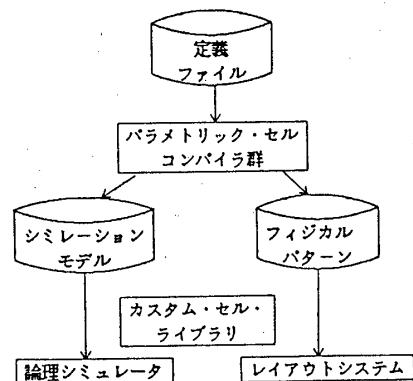


図1. パラメトリックセルコンパイラの位置付け

に従い、必要情報(形状、端子位置、端子レイヤ等)を抽出し、プリミティブ・ライブラリを作成する。

プリミティブの相対的な配置構造を、パラメトリック・セル定義言語(PCDL)を用いて記述する。この記述は、コンパイラ・コンパイラにてシンタックス・チェックが行われ、パラメトリック・セル・コンパイラが作成される。

定義ファイルには、どのパラメトリック・セル・コンパイラを使用するか、また個々のパラメトリック・セル・コンパイラに対応したパラメータ(例えば、ROMの場合、アドレス幅、データ幅、1/0パターン)が定義される。

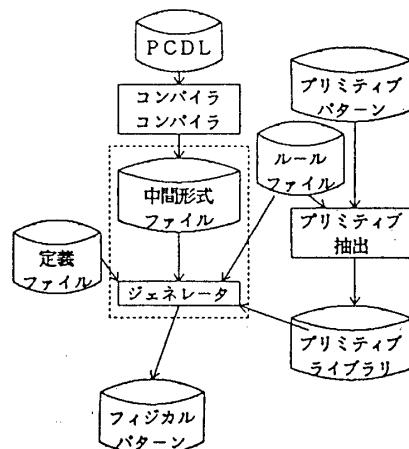


図2. システム構成

ジェネレータは、定義ファイル内のパラメータ、ルール定義ファイルおよびプリミティブ・ライブラリを参照し、コンパイラ・コンパイラによって作られた中間形成ファイルを解釈しながら、フィジカル・パターンを作成する。

4. パラメトリック・セル定義言語

パラメトリック・セル定義言語は、セル開発者が使用する言語であるため、シンプルな構成とした。

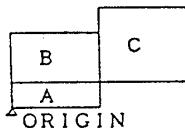
その基本形は、

「指示、対象、位置」に集約できる。

ここで、「位置」は、座標値だけでなく、すでに配置したプリミティブとの相対関係で指定できる。

たとえば、

「ORIGINにAを配置し、Aの上にBを配置しBの右にCを配置し...」という様に、配置したプリミティブをもとに相対配置可能なので、プリミティブの大きさを意識せずに記述できる。(図3)



```
PLACE A TO ORIGIN;
PLACE B TO UP A;
PLACE C TO RIGHT B;
```

図3 記述例

上記基本形に加え

- ・配置先座標の取り出し
- ・フローを制御するコントロール文
(配置の繰り返し、DO文、IF文など)
- ・位置演算の為の位置変数
(x、yの2値を持つ)
- ・非固定名称
(ピン名、プリミティブ名の生成機能、
ABC#2(i)、iの値により名前決定、
i=5の時、ABC05)
- ・豊富な組み込み関数
(プリミティブ情報の取り出し、演算関数等)
などの特徴を持っている。

5. 適用例

現在、本システムを使用し作成されたROM/RAM/PLA/シフトレジスタ用パラメトリック・セル・コンパイラが、スタンダードセルLSI統合設計システムに組み込まれている。

図4に、ROMの記述の抜粋、図5にROMのフィジカル・パターンの一例を示す。

記述量は、350~600ラインであり、各々1週間以内でパラメトリック・セルコンパイラが作成できた。

処理時間は、μVAX IIでROMの場合、コンパイラ・コンパイラ部2.4秒、ジェネレータ部1分6秒(図5のROM)であった。

6. まとめ

本システムの開発により、セル・コンパイラの作成をセル設計者自身が行え、セル・コンパイラをスタンダードセル統合設計システムへ容易に組み込める様になった。

```
cellcomp ROM00 { TYPE=CELL RULE=LARGE };
Parameter
  WORD_NUM      integer range (32:4096),
  EIT_NUM       integer range (1:2),
  NET_BIT       integer range (4:32) bound (4),
  BIT_PATTERN   bit (BIT_NUM=WORD_NUM);

--- PLACE BIT_PATTERNS
MY_INT1 = 0;
NET_WORD1 = NET_WORD - 1;
do LOOP_V from 1 to EIT_NUM by 1
begin
  MY_POS_EIT=MY_POS_ESTART;
  do LOOP_EIT_H from 0 to NET_WORD1 by 1
  begin
    MY_POS_EIT = MY_POS_EIT + (cell_xsize( REG00 ), 0 );
    MY_POS_BIT = MY_POS_EIT;
    MY_INT2 = mod(LOOP_EIT_H, 2);
    if LOOP_EIT_H < mod(LOOP_EIT_H, 2) then
      begin
        MY_INT1+=MY_INT2-1;
        if BIT_PATTERN(MY_INT1) then
          MY_INT4+=1;
        else
          MY_INT4=0;
        if MY_INT2 eq 0 then
          place REG100(MY_INT4) to up MY_POS_BIT
          set_pos up left MY_POS_BIT;
        else
          place REG100(MY_INT4) to up MY_POS_BIT z_mirror
          set_pos up left MY_POS_BIT;
        if LOOP_EIT_H eq 16 then
          MY_POS_EIT=MY_POS_EIT+10.CELL_VSIZE(REG00));
      end;
    end;
    MY_POS_ESTART=MY_POS_ESTART
      + (0, cell_ysize( RF*200( NET_BIT ) ) );
    if NET_BIT neq 32 then
      MY_INT4+=6/NET_BIT;
    else
      MY_INT4+=1;
    MY_INT5=mod(LOOP_V, MY_INT4);
    if MY_INT5 eq 0 then
      MY_POS_ESTART=MY_POS_ESTART+(0, CELL_VSIZE(REG00));
  end;
end;
end_cellcomp;
```

図4. ROMの記述(抜粋)

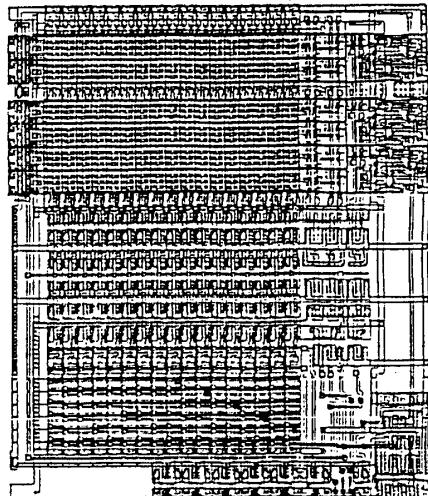


図5. ROMのフィジカルパターン

これにより、LSI設計時、希望パラメータを入力する事で、パラメトリック・セル・コンパイラが生成する最適なサイズのセルが得られる事になり、柔軟なスタンダードセル方式設計システムとなっている。

今後、パラメトリック・セル定義言語の拡張、およびプリミティブ間の自動配線機能等を組み込んだ、より柔軟で強力なパラメトリック・セル・コンパイラ作成システムを計画している。

[参考文献]

- (1) H-F.S.Law et al. "An Intelligent Composition Tool for Regular and Semi-Regular VLSI Structures" Proc. ICCAD No.6B Nov.1985
- (2) J.M.Mata, "ALLENDE:A Procedural Language for The Hierarchical Specification of VLSI Layout", Proc. 22nd D.A.Conf., pp.183-189, Jun.1985.