

## ルールベース形配線遅延時間算出プログラムLAVI

2M-8

月橋 祥治、  
(日本電気㈱)長谷川拓己、  
(日本電気㈱)加藤 忠道  
(日本電気ソフトウェア㈱)

## 1. はじめに

遅延解析の前段階として、配線遅延時間を算出するルールベース形プログラムLAVIを開発した。これを大型汎用計算機などの遅延解析に適用し、良好な結果が得られたので報告する。

LAVIは、遅延時間計算ルールをルールベース化することにより、配線のモデル化の手法やテクノロジーなどの違いに対応するためのプログラム開発を不要としている。このため、遅延時間計算ルールの変更に迅速かつ柔軟に対応できる。また、遅延時間計算ルールをコンパイルし、オブジェクト・プログラムの形でLAVI本体にリンクすることにより、高速な処理を実現している。さらに、諸条件、配線遅延時間計算式(以下、計算式)の入力をパソコンによる対話型遅延時間計算ルール入力ツールPCRULEで実現しており、遅延時間計算ルールを容易にルールベース化できる。LAVIを用いることにより、従来のように遅延時間計算ルールをプログラム本体に内蔵する場合に比べて、新たな遅延時間計算ルールへの対応に要する工数を約1/4に削減できた。

## 2. 配線遅延時間計算における問題点

遅延解析を行なう場合、それに先立って回路遅延時間と配線遅延時間を求める必要がある<sup>1)</sup>。回路遅延時間は、使用するテクノロジーにより値が一意に決まるのでライブラリ化することが可能である。一方、配線遅延時間は、任意のテクノロジーについてこれを正確に求めるためには、SPICEなどの詳細なレベルのシミュレーションを行なう必要があるが、これは処理速度などの面から困難である。このため、遅延解析時には、接続関係、配線結果に基づき、テクノロジー毎に近似の配線遅延時間の計算を行なっている。この計算を、遅延時間計算ルールをプログラム本体に内蔵する方式で行なう場合、新テクノロジーへの対応などで遅延時間計算ルールが追加、変更になる度にプログラムの作成、改造が必要になる。このため、ルール対応が遅れることにより設計装置の遅延検証の日程に影響を及ぼしかねない。また、プログラムの開発に多大な工数を費やし、プログラム・バグなどによる遅延解析の品質低下の要因にもなる。

## 3. LAVIの遅延時間計算ルールの特長

2章で述べた問題点を解決するため、LAVIでは、遅延時間計算ルールをプログラムとは別にルールベース化した。このため、ネ

ットのモデル化の手法やテクノロジーなどが変更になった場合にも、遅延時間計算ルールのみを作成すれば良く、従来のようなプログラム開発は不要となった。

LAVIの遅延時間計算ルールは、PCRULEを用いて、プログラム本体とは別に作成される。PCRULEは、接続関係、線長範囲などの条件や計算式などを、配線のモデルに近いイメージで図的にかつ対話的に入力し、ルールベース化する。したがって、遅延時間計算ルールを視覚的に捉えられ、ルール入力が容易にできる。PCRULEを用いることにより、新遅延時間計算ルールのプログラム対応が迅速に行なえ、工数の削減、早期の遅延検証が可能になった。

PCRULEは、入力した計算式をプログラミング言語のソース・コードに変換し、これをルールベースとして出力する。このソース・コードをコンパイルし、オブジェクト・プログラムの形でLAVI本体にリンクし、配線遅延時間算出を行なう。この手法により、インタプリタ方式によるルール・ベース形プログラムに比べて高速な処理を実現できた。

## 4. LAVIの構成

LAVIは、図1の構成をとる。論理回路の設計データは、ネット・モデル作成部により、LAVIの内部形式であるネット・モデルに変換される。一方、各配線遅延時間計算ルールは、PCRULEにより、計算式を選

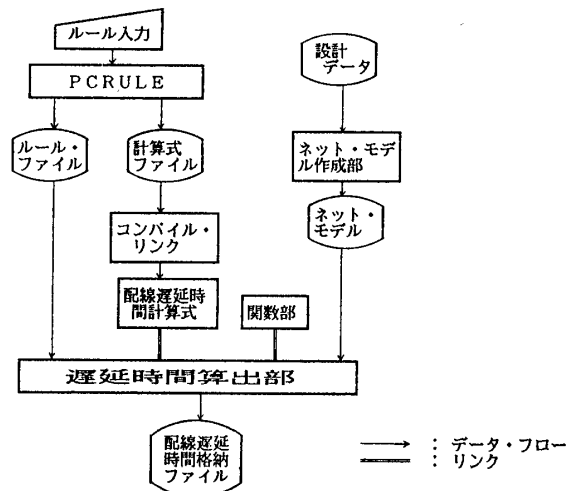


図1 LAVIプログラムの構成

択するための情報からなるルール・ファイルと配線モデル毎の計算式およびその関連情報からなる配線遅延時間計算式ソース・ファイル（以下、計算式ファイル）に変換される。ルール・ファイルには、接続関係、端子情報などがテーブルの形式で格納されており、計算式ファイルには、計算式などがプログラミング言語のソース・コードの形で格納されている。計算式ファイルのソース・コードは、言語コンパイラによりオブジェクト・プログラムに変換された後、関数部と共に遅延時間算出部にリンクされる。この結果生成された新たな遅延時間算出部が、ルール・ファイルとネット・モデルを入力し、両者の対応をとり配線遅延時間を算出し、算出結果を配線遅延時間格納ファイルに出力する。

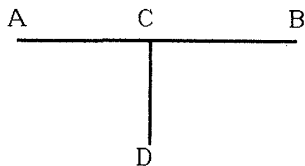
### 5. 遅延時間計算ルール入力

PCRULEは、対話形式でルールを入力し、計算式ファイル、ルール・ファイルを作成する。

計算式ファイルには、計算式がプログラミング言語のソース・コードの形式で格納される。計算式は、線長範囲、端子の種類などにより分類され、複数存在する場合もある。この場合には、配線遅延時間の計算時に、各ネットの情報に基づき、いずれか一つの計算式が選択される。このため、PCRULEでは計算式をif~then~else構造をもつ条件文の形で分類して、計算式ファイルに出力する。if以下には、計算式を分類するための条件文を、then以下には、その条件文が満たされた時に用いられる計算式をそれぞれ出力する。

ルール・ファイルは、複数の計算式の中から各ネットに合った計算式を選択するための情報からなる。実際には、配線モデルの端子の数、分岐点の数・位置、配線種類などをテーブルの形式で格納する。

図2は、PCRULEを用いたルール入力例である。線長範囲の項に指定した不等式の左辺は、指定区間の配線長を表している。図2の計算式は、図3に示すようなプログラミング言語のソース・コードに変換され、計算式ファイルに格納される。図3において、関数lengは、指定された2点間の配線長を返し、関数pointは、ルール入力時に指定された点の名前からプログラム中で扱うポイント値を返す関数である。また、関数resist、capは、



\*線長範囲はいくつですか？

$$\begin{aligned} [A-B] &\leq 10.0 \\ [C-D] &\leq 2.0 \end{aligned}$$

\*計算式を入力して下さい。

$$0.5 * (2 * LENG + RESIST) * CAP$$

図2 ルール入力例

それぞれ総端子抵抗値、総端子容量値を返す関数である。これらの関数は、LAVI本体の関数部において用意されている。

```
float calc()
{
  int point;
  float leng, resist, cap;
  float delay;
  if ( ( leng(point('A'), point('B'))
    <= 10.0 ) &&
    ( leng(point('C'), point('D'))
    <= 2.0 ) )
    delay = 0.5 * ( 2 * leng(point('A'),
    point('B')) + resist() ) * cap();
  else if
  {
  }
  else if
  {
  }
  return( (float)delay );
}
```

図3 計算式ファイルの例

### 6. 効果

同程度の遅延時間計算ルールに対して、従来のプログラムとLAVIとで新しいルールの対応に要する工数の比較を行なった。

従来のように遅延時間計算ルールをプログラム本体に内蔵する形式の場合、遅延時間計算ルールを手でコーディングするので、新遅延時間計算ルールへのプログラム対応にプログラムのテスト・評価を含めて2か月を要した。一方、LAVIでは、PCRULEを用いて図的かつ対話的に遅延時間計算ルールを作成し、ルールベース化することにより、2週間でルール対応ができた。したがって、工数が約1/4に削減され、早期の遅延検証が可能になった。また、PCRULEを用いたことにより、遅延時間計算ルールの誤りが原因である遅延解析上の障害は発生していない。このことより、LAVIの遅延時間計算ルールが高品質であることも証明されている。

### 7. おわりに

LAVIの遅延時間計算ルールのルールベース化の手法と、遅延検証における有用性について述べた。LAVIは、EWSからスーパー・コンピュータに至るまで幅広く用いられ、開発装置の遅延検証に多大な効果を挙げた。

### 参考文献

1) 長谷川他、"大規模回路向けタイミング検証システムHEART(1)高速化の手法", 情処第35回全国大会, 7F-6, 1987.