

RT マクロを用いた複合データパス設計手法

今橋正彦, 中村 猛, 大井 康, 白石 豪, 平沢政夫

日本電気株式会社

本論文は信号処理プロセッサ等の特定用途向けプロセッサにおけるビット長の一定しないような複合データパスの設計手法について報告する。

本手法は RT マクロと呼ばれるパラメータ化されている RT レベルのライブラリを用いネットリストにおけるデータパス部と補助回路部の自動デザイン分割、RT マクロレベルのフロアプラン等により複合データパスを作成するものである。

本手法に基づく複合データパス設計システムを作成し実行結果について報告する。

Complex Datapath Design Method with RT Macro

Masahiko Imahashi, Takeshi Nakamura, Yasushi Ooi, Takeshi Shiraishi, Masao Hirasawa

NEC Corporation

1753, Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa, 211 Japan

We propose an integrated environment to design a large scale complex data path with various bit lengths of function units for a special-purpose processor such as a signal processor.

This system uses a parameterized RT level library called RT macro, and it is constructed by the integration of elemental tools incorporating parameterized schematic editor, an automatic design partitioning program, and a RT macro level floor planner.

1 はじめに

マイクロプロセッサ内部の演算処理を司るデータバス部のレイアウトはチップ全体の集積度やプロセッサの演算性能などに深く関係している。これはレイアウトの特殊性やデータバスの構造がアーキテクチャと密接に結びついているためである。現在、信号処理プロセッサ等特定用途向けプロセッサの高性能化、多様化の進む中でデータバス部も大規模になると共に複雑さを増している。

汎用プロセッサのデータバスのように比較的規則的なデータバスに対し本論文ではこのようなビット長の一定しない演算器を含むより複雑なデータバスを複合データバスと呼んでいる。

このような設計の複雑性において人手によるデータバス部のレイアウト作成は規模においても、構成の複雑度においても困難になりつつある。それに対しプロセッサをタイムリーに市場に送り出す必要性からレイアウトの設計期間の短縮はますます望まれている。

しかし、プロセッサのアーキテクチャを設計する段階で、データバス部の各ユニットの遅延情報や、レイアウトサイズを考慮しなければ、より高性能なプロセッサを作成することが困難であることは言うまでもない。

本論文では、多様な複合データバスのレイアウトを自動で短期間に合成する手法について延べる。この手法により、データバスの RT レベル回路図設計からレイアウトの生成までを、一括して CAD 化している。

またデータバス設計に関する各種データを RT マクロという形でライブラリ化しているため、設計資産の共有化が可能になり、ライブラリのチューニングも比較的容易になる。

2 本システムの狙い

従来の汎用マイクロプロセッサにおいて演算処理を司るデータバス部のビット長は例えば 32 ビットのようにほぼ一定しているのに対して、特定用途向けプロセッサにおいてはアーキテクチャの多様化により各演算器のビット長が必ずしも一定とならないデータバスを作成する必要性が生じて来た。

このような複雑性を持つデータバスを本論文では複合データバスと呼んでいるが、このような複合データバスを高性能かつ高集積に作成することが本設計手法のねらいである。

最近データバス自動合成の話題は論文 [1],[2] のように高レベル合成にシフトしている。しかし、レイアウト合成としては論文 [3],[4] のように汎用マイクロ

プロセッサのデータバスをそのメインターゲットとしているのが現状である。

汎用データバスにおいては各演算器のビット長が一定しているためデータバス内の自動配置は一次元自動配置問題に帰着できる。それに対し複合データバス系においては各演算器のビット長が一定していないため従来各セルの自動配置を行うことが困難であった [5]。

本設計手法では複雑なアーキテクチャにより各演算器のビット長が一定しないような複合データバスを効率的に設計するため RT マクロと呼ばれるデータバス用標準ライブラリを用意し、RT マクロに含まれるドライバ部と呼ばれる信号線のバッファを含む補助回路とデータバス部分の回路を自動デザイン分割を行い、データバス部分を RT マクロ単位で自動配置を行うことにより従来の問題点を解決した。

また、集積度を向上するためには各演算器を高密度に敷き詰めることが重要であるが、本設計手法では RT マクロレベルのフロアプランを導入した。

本設計手法においてさらにタイミング保障のためにタイミングドリブン機能を導入し、ビット長の異なる演算器の自動配置も可能にしている [5]。

3 RT マクロ

汎用マイクロプロセッサにおいてデータバス回路はそのアーキテクチャ毎に設計するのが最適でありその意味では回路の流用性は乏しい。

それに対して複合データバス系において同様の演算器が随所に出現し、かつそのビット長も一定しないため RT マクロのようにパラメータ化されたライブラリを持つことが設計の効率化にとって非常に有効である。

RT マクロは図 1 に示すように ALU、加算器、マルチプレクサ、トライステートバッファ、レジスタなどの RT レベルのマクロである。

その内容は RT 回路図を描くときに用いるパラメータ化された RT マクロシンボルの下位階層に RT マクロ機能記述、RT マクロ回路図、データバス部セルレイアウト、ドライバ部セルレイアウト、セル内動作記述、遅延ライブラリである。

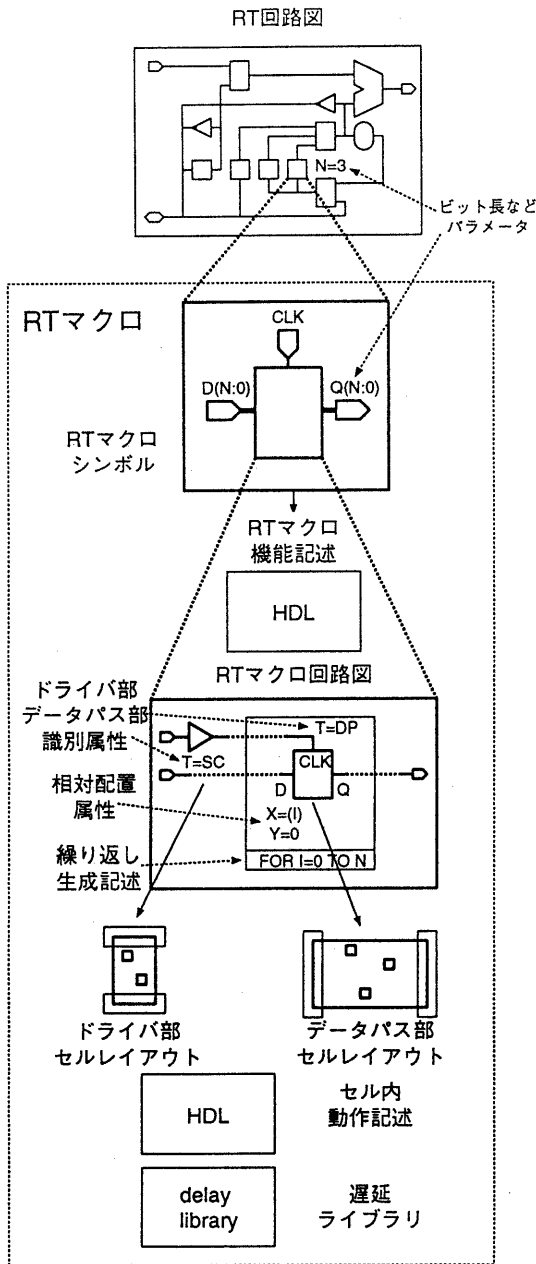


図1 RTマクロ

3.1 RTマクロシンボル

RTマクロシンボルはRT回路図を描くときに用いられる。RTマクロシンボルの端子は演算器のビット長などにより端子幅がパラメータ化されている。

3.2 RTマクロ機能記述

RTマクロ機能記述はRTレベルネットリストを用いて機能シミュレーションを行う場合に用いられる。RTマクロ機能記述は演算器のビット長などによりパラメータ化されている。本データは機能記述のためシミュレーションを比較的速い速度で行うことができる。

3.3 RTマクロ回路図

RTマクロ回路図は演算器のビット長などによりパラメータ化された回路図で、ドライバ部とデータバス部を識別するための属性 RTマクロセル生成のために用いられる相対配置属性などが付加されている。本論文のシステムで用いた回路図入力システムは繰り返し生成記述が用意されているため回路図のパラメータ化は比較的用意である。すなわち RTマクロ回路図は演算器のビット長などによりパラメータ化された回路ジェネレータとなっている。ドライバセルはデータバス制御線のバッファやデコーダを含むが演算器のビット数など RTマクロ内の負荷条件により駆動力の異なるセルを呼び出すように設定されている。

3.4 データバス部セル・ドライバ部セル

データバス部セルレイアウト、ドライバ部セルレイアウトはそれぞれデータバス部レイアウト生成、ドライバ部レイアウト生成で用いられる。データバスセルは演算器の1ビット分に相当する内容を持っている。ALUや加算器のような大規模な演算器の場合さらに1ビット分を分割してセル化する場合もある。ドライバ部はバッファやデコーダなどであり、データバスセルに比べて簡単なセルである。ドライバ部セルについては SPICE ネットリストからのセル自動生成を行っている。

4 設計フロー

設計フローを図3に示す。各項目を以下に示す。

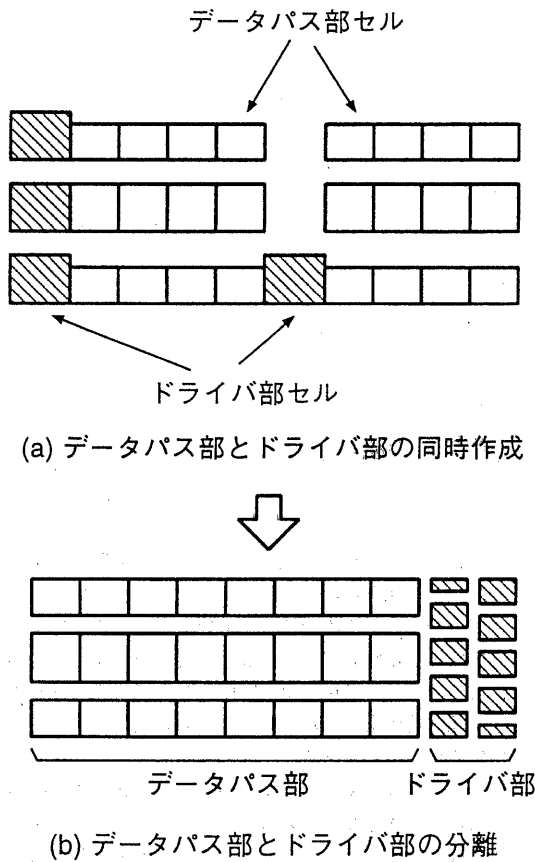


図2 データバス部とドライバ部の分離

図2にデータバス部とドライバ部の分離について示す。従来図2(a)のようにデータバス部とドライバ部を同時に作り込んでいたが[4]、複合データバスにおいてはドライバ部セルがデータバス部セルと混在し集積度が低下してしまう。高集積化のため本設計手法では図2(b)のようにデータバス部とドライバ部の分離を行っている。

3.5 セル内動作記述・遅延ライブラリ

セル内の動作記述と遅延ライブラリは遅延検証で用いられる。RTマクロ回路図でデータバス部セルとドライバ部セルが直接呼ばれているために、このレベルのデータはパラメータ化することなく用意される。セル内動作記述と遅延ライブラリを用いてダイナミックな遅延検証を、遅延ライブラリのみでスタティックな遅延検証を行うことができる。

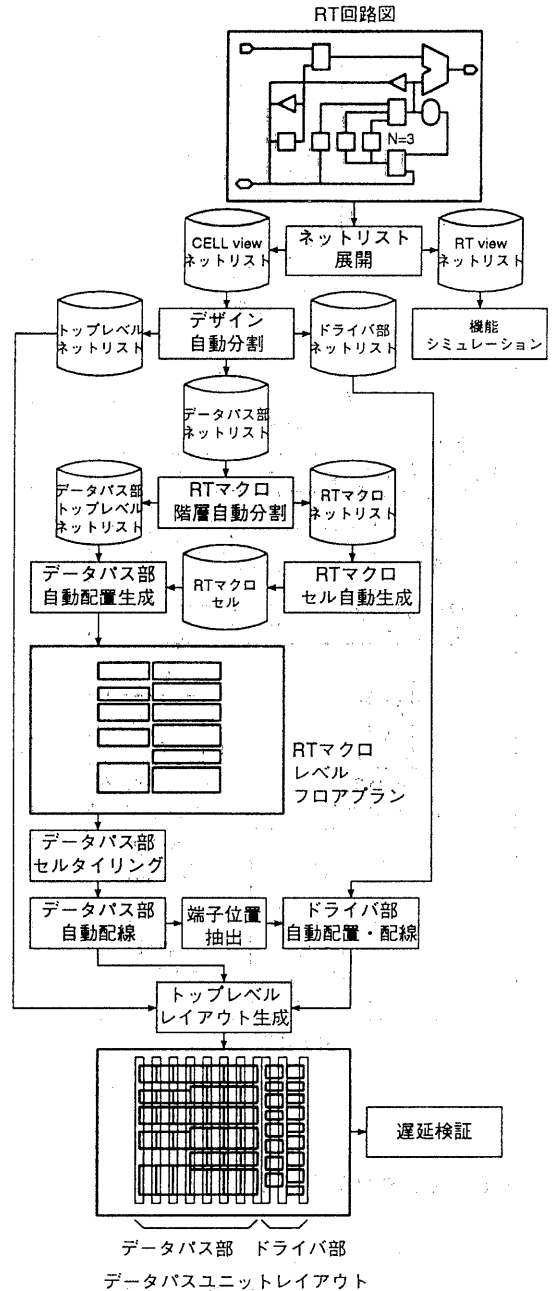


図3 設計フロー

4.1 ネットリスト展開

RT 回路図を展開してネットリストを作成する。機能シミュレーション用には RT レベルネットリストを、レイアウト作成用にはセルレベルのネットリストを作成する。

4.2 機能シミュレーション

RT レベルネットリストと RT マクロ機能記述により機能シミュレーションを行う。

4.3 デザイン自動分割

RT マクロ回路に図に付加されたドライバ部・データバス部識別属性によりセルレベルのネットリストの分割を行う。このデザイン自動分割によりセルレベルネットリストはトップレベルネットリスト、データバス部ネットリスト、ドライバ部ネットリストに分割される。

4.4 RT マクロ階層自動分割

データバス部において同一 RT マクロ内データバスセル群の相対配置を固定してマクロ化したものが RT マクロセルである。このフェーズでは RT マクロ階層をデータバス部ネットリストから自動分割する。この際出力されるデータバス部トップレベルネットリストは各 RT マクロ階層を呼び出すネットリストである。

4.5 RT マクロセル自動生成

図 4 に RT マクロセルについて示す。

図 4(a) のようにデータバス部セルの整列を行う場合、従来の自動配置方式では図 4(b)、図 4(c) のようにセルを同一行や同一列に整列することは困難であった。そのため図 4(d) のように同一 RT マクロに属するセル群を RT マクロセルとして併合する。

RT マクロセル内のデータバス部のセルの並べ方は RT マクロ回路図に付加されている相対配置属性を用いる。

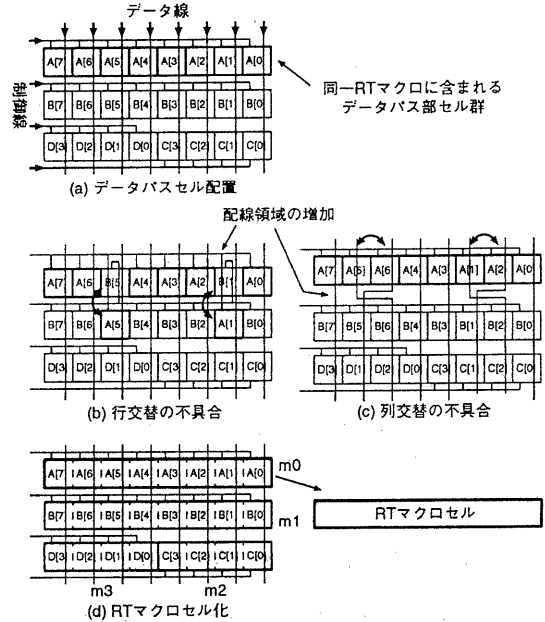


図 4 RT マクロセル

4.6 データバス部自動配置生成

前述の RT マクロセルとデータバス部トップレベルネットリストを用いて、データバス部自動配置生成を行う。この際ネット荷重などのタイミングドリブンの制約を付加することも可能である [5]。

4.7 RT マクロレベルフロアプラン

データバス部の自動配置結果を対話的に修正する。自動配置では主としてネットのカット数減少により配置を行うため、高集積化の目的に人手配置修正を加えられるようにしている。データバス回路によっては人手配置修正を加えなくてもよいものもあるが、複合データバスにおいては各演算器のビット数が一定でないため RT マクロレベルフロアプランは有用である。

4.8 データバス部セルタイリング

上記データバス部自動配置生成後あるいはデータバス部対話的配置修正後に RT マクロセルを展開し元のデータバス部セルによって置き直す。RT マクロセルを各データバス部に分解するのは後述のデータバス部自動配線時に配線性により各セル間の間隔を必要とする場合があるためと、遅延検証用のセル内動作記述と遅延ライブラリがデータバス部セル毎に用意されており、RT マクロセル単位で用意されていないためである。

4.9 データバス部自動配線

データバス部セルタイリングの後データバス部自動配線を行う。自動配線時に配線性を見積もるがその際配線が混雑する場合セル間の間隔を調整した後配線を行う。

4.10 データバス部端子位置抽出

データバス部レイアウトとドライバ部レイアウト間の端子位置を一致させるため、データバス部のレイアウトの端子位置を抽出する。

データバス部端子位置抽出によりデータバス部とドライバ部間の配線領域を不要にすることができる。

4.11 ドライバ部自動配置・配線

ドライバ部ネットリストとデータバス部端子位置情報を用いドライバ部の自動配置・配線を行う。ドライバ部配置・配線は通常のスタンダードセルの配置・配線と同様である。

4.12 トップレベルレイアウト生成

データバス部レイアウトとドライバ部レイアウト間の端子位置をあわせ、トップレベルネットリストによりデータバスユニット全体の端子を付加する。

4.13 遅延検証

データバスユニット全体のネットリストとレイアウトおよびセル内動作記述、遅延ライブラリにより遅延検証を行う。遅延検証結果によりデータバス部セル、ドライバ部セルの駆動能力の変更(サイジング)やデータバス部配置の変更等のフィードバックにより最適なレイアウトを作成することができる。

5 レイアウト合成結果

レイアウト結果を図5に示す。実行結果の平均集積度は2.5層0.8 μm プロセスで平均5,000 Tr/mm^2 、最大9,335 tr/mm^2 であった。これはデータバス部とドライバ部の分離を行わない従来の設計手法[4]に比べて平均集積度が1.5倍程度増加した。

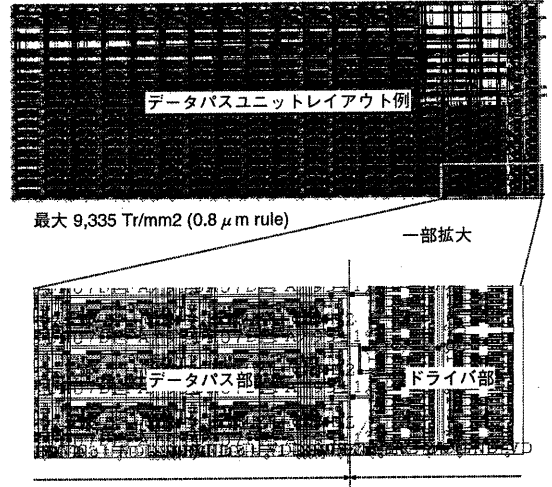


図5 レイアウト合成結果

6 結論

複合データバスの設計フローを確立することができた。本手法は傾向の異なるデータバス部とドライバ部を分離することによりレイアウトの最適化を行い集積度の向上を行うことができた。今後は、データバス部自動配置の最適化やタイミング保障機能の強化等を行っていきたい。

【文献】

- [1] G.Krishnamoorthy et.al., "Data Path Allocation using an Extended Binding Model," *Proc.29th DAC*, pp.279-284(1992).
- [2] C.Papachristou et.al., "An Approach for Redesigning in Data Path Synthesis," *Proc.30th DAC*, pp.419-423(1993).
- [3] 辻橋 他, "セルベース設計用データバスジェネレータ", 三菱電気技報, Vol.67, No.3, pp.21-24(1993).
- [4] 今橋 他, "レジスタトランスフェラブル回路図を用いたデータバスレイアウト設計手法," 電子情報学会技術研究報告, VLD91-131, pp.23-29(1992).
- [5] 中村 他, "パラメータ付きネットリストを用いたタイミングドリブンデータバスレイアウト設計手法," 情処学会第28回全国大会, Vol.6, pp.143-144(1993).