

## テクニカルノート

## 新方式データパスモジュールジェネレータ

鈴木 五郎<sup>†</sup> 堀内 重則<sup>†</sup>

チップ全体の配線容易性を考え、セル配置だけを行い配線はチップレベル一括配線に任せるデータパスモジュールジェネレータを開発した。グラフィックアクセラレータチップの設計において評価した結果、データパスモジュールジェネレータをまったく使用しなかった場合と比較してチップ全体の面積/総配線長とも最大で40%程度減少できることを確認した。またデータパスモジュール内のセル配置とセル間配線まで行う従来のシステムと比較を行った結果、チップ全体での総配線長を最大で5%程度短縮できる場合があることが明らかとなった。

## New Type Datapath Module Generator

GORO SUZUKI<sup>†</sup> and SHIGENORI HORIUCHI<sup>†</sup>

We have developed a new type datapath module generator which places cells automatically based on the Verilog-HDL net list. But this module generator doesn't do the routing intentionally in order to get high routability in the chip level. This was applied to real graphic accelerator chip design. The chip area and total wire length could be reduced by 40% comparing with no usage of any datapath module generator, and the total wire length could be reduced by 5% comparing with conventional module generator which does not only cell placement but also routing in the datapath module.

## 1. ま え が き

データパスの規則性を利用してレイアウトを行うデータパスモジュールジェネレータの有効性に関しては多数報告されている<sup>1),2)</sup>。しかし、そのほとんどはセル配置だけではなくセル間配線まで実行しており、チップ全体のレイアウトから見ると完成したデータパスモジュールは貫通信号（データパス内部にはまったく供給されていない信号）も許さない障害物として処理されてしまう。つまりデータパスモジュールの面積と総配線長を目的関数とした局所最適化だけを図り、チップ全体でのレイアウト大域最適化が不可能になっている。今回データパスモジュールジェネレータをより効果のあるものにするため、セル配置だけを行い、配線はチップレベル一括配線に任せてチップ全体で最適化を図るシステムを開発した<sup>3)</sup>。本モジュールジェネレータはフルカスタム、セルベース、およびゲートアレーを対象にしており、矩形の領域にセル配置を行っている。

## 2. チップ設計フロー

開発したデータパスモジュールジェネレータを用いたチップ設計フローは以下のとおりである。

- (P1) 論理合成ツールを使用し、データパスを記述した RT レベル Verilog-HDL 記述をゲートレベルつまり、ASIC で用意されているセルを用いた接続情報に変換する。
- (P2) 設計者はデータパスモジュールジェネレータに該当データパス部分の接続情報を入力し、データパス内のセルを自動配置する。ここではチップ全体は考慮せず、該当データパスモジュールに関してのみ処理を行う。
- (P3) 設計者は、フロアプランツールを用いてデータパスモジュールとその他のモジュールをチップ上に人手配置する。フロアプランツールはモジュールジェネレータで配置したセルの配置座標をチップ上の絶対座標に変換する。
- (P4) 配置/配線ツールを用い、データパスモジュール以外のモジュールに含まれるセルの自動配置と、データパスモジュールを含むチップ一括で配線処理を行う。

<sup>†</sup> 株式会社日立製作所大みか工場  
Omika Works, Hitachi Ltd.

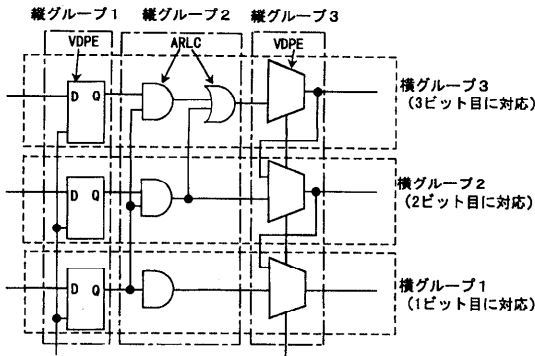


図1 データバス回路構造  
Fig. 1 Datapath structure.

### 3. モジュールジェネレータ処理手順

加減乗除などのデータバスは図1に示すように、基本的にはビットごとに繰り返される規則的な回路構造を持っている。しかしながら詳細を見ると、レジスタファイルのフリップフロップのように全ビットに必ず存在するセル（以下、VDPE: Vectorized Data Path Element と呼び、全データバスの60~70%を占めている）と、一部のビットにしか存在しないセル（以下、ARLC: Associated Random Logic Cells と呼ぶ）から構成される。以下では、各ビットに所属するセルグループを横グループと呼び、ビットを横断して存在する同一種類のVDPEグループとそれらに挟まれるARLCグループを縦グループと呼ぶことにする。

#### (S1) VDPE と ARLC 縦グループの認識

Verilog-HDL で記述された module 文の単位が縦グループ化の基本となるが、上記した基準でVDPE と ARLC の縦グループを分離する。

#### (S2) 横グループの認識

(S1) で認識したVDPE 縦グループ内のゲートつまりセルに関して、バス信号名や論理合成で付けられたインスタンス名を手がかりにして、どのセルがどのビットに所属するかを認識し、ビットに対応した横グループとする。またARLC 縦グループに関しては、VDPE のビットスライス構造から信号を辿り、どのARLC をどのビットに所属させるべきかを決定し、該当する横グループに加える。

#### (S3) 縦グループの配置順序の決定

信号の流れをベースとしてVDPE と ARLC 縦グループの配置順を決める。

#### (S4) セル配置暫定位置決定

(S2) で認識された同一横グループに属するセルに

関しては、同一配置暫定 Y 座標を与える。横グループ内のセルの中で、高さの最大値を該当横グループの縦幅とし、図1の例では下方に位置する横グループ1から横グループ3まで順に隙間なく積み上げ、各横グループ内セルの Y 座標を決定する。一方縦グループ内のセルの内横幅の最大値（同一縦グループの1つのビットに複数のセルが存在する場合は横幅の合計の最大値）を該当縦グループの横幅とし、左方に位置する縦グループ1から縦グループ3まで順に隙間なく詰め込み、各縦グループ内セルの X 座標を決定する。同一縦グループに属するセルに関しては、同一配置暫定 X 座標を与える。図1のARLC 縦グループ2のように、同一縦グループの1つのビットに複数のセルが存在する場合にはセルどうしが重ならないように信号の流れをベースとして X 座標を決定する。

#### (S5) セル最終配置座標の決定

縦グループ間に矩形ギャップを想定し、スタイナー木を基本として縦方向の配線数を見積もり（データバスモジュールを貫通する信号用に、ある程度の余裕を持たせている）、必要な幅のギャップを空けながら各セルの最終配置 X 座標を決定する。同様に横グループ間の配線ギャップを空けながら各セルの最終配置 Y 座標を決定する。

レジスタファイルが主であるデータバスモジュール（約1kゲート、0.35 $\mu$ m、AL3層配線）に対して本モジュールジェネレータを適用した例を図2に示す。レイアウト密度は10kトランジスタ/mm<sup>2</sup>である。

### 4. チップ設計への適用結果

データバスをゲート数換算で50%程度含む7種類のグラフィックアクセラレータ（トランジスタ全面敷き詰めゲートアレー、0.35 $\mu$ m、AL3層配線、50MHz）の設計において本モジュールジェネレータを適用した。図3に示した2例とも、チップレイアウト中規則的に見えるところがデータバスである。

事例1は図2のモジュールを含むバス制御用ゲートアレー（ベースゲート数250k、ユーザゲート数90k）である。この例では、データバスの内部を貫通する信号が100本ほど存在している。データバスの内部をブラックボックス化し、チップ配線においてデータバスモジュール内の貫通信号を許さない（つまり該当データバスモジュールの周りを迂回してしまう）従来のデータバスモジュールジェネレータ<sup>2)</sup>を使用した場合と比較を行った。その結果チップ全体の総配線長を

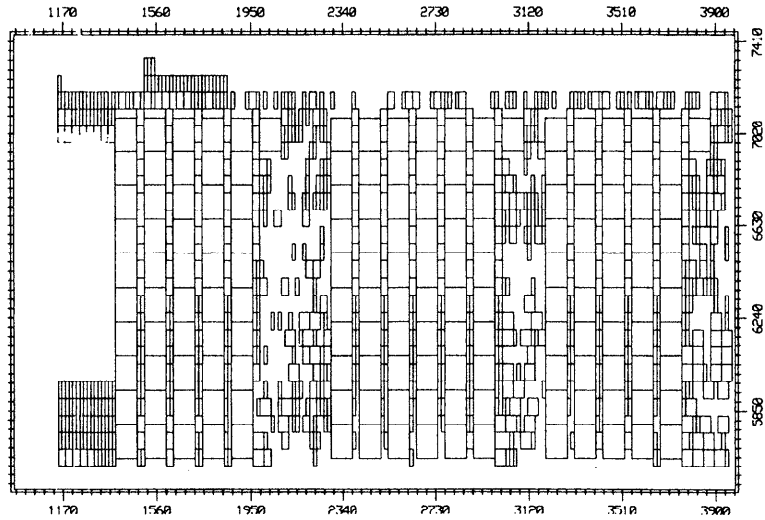
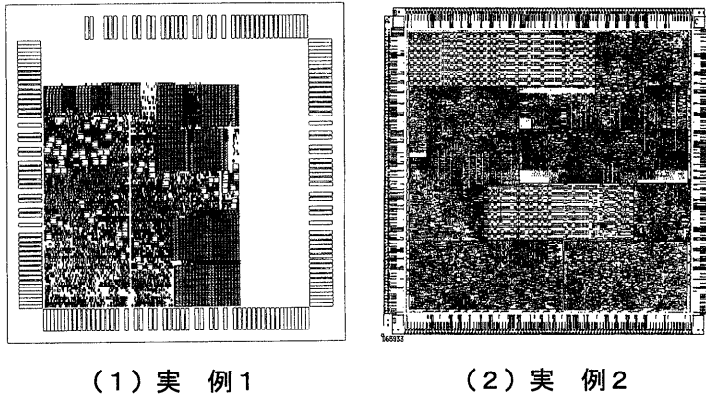


図 2 データバスモジュールジェネレータ出力例  
Fig. 2 Datapath module example.



(1) 実例1

(2) 実例2

図 3 チップレイアウト例  
Fig. 3 Chip layout example.

5%程度短縮することが確認できた。セル配置だけを行い、配線はチップレベル一括配線に任せてチップ全体で最適化を図った効果が実証されたことになる。またすべての信号に付けられた信号伝播遅延制約が満足される状態で、配置/配線が可能となる限界までチップ面積を小さくした。データバスモジュールジェネレータをまったく使用しなかった場合、つまりデータバスと他論理とを区別せず一括で配置/配線(図3(1))の外枠がそのチップ面積に相当)した場合と比較した結果、使用時はチップ面積比で約-40%となり、面積削減効果が確認できた。また総配線長短縮率 =  $(L1 - L2)/L2 = -40\%$ である。

ここで、

L1:モジュールジェネレータ使用時のチップ全体の総配線長  
L2:未使用時のチップ全体の総配線長  
実例2はレンダリング処理用ゲートアレー(ベースゲート数1076k, ユーザゲート数505k)であり、セルの配置もチップ全体を考慮しながら最適化を図らないとモジュールジェネレータの効果が得られない例である。モジュールジェネレータを使わずにチップ一括配置/配線を行った場合、このチップサイズですべての信号に付けられた遅延制約が満足されているが、モジュールジェネレータを使った場合、一部の信号のタイミング制約が満足されていない状況が生じている。総配線長短縮率 = +13%であり、チップ全体の総配線

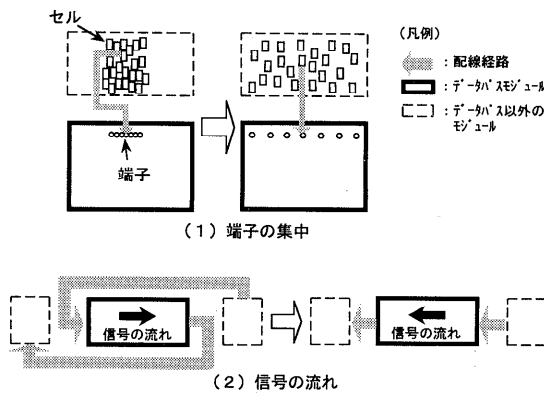


図 4 問題点分析

Fig. 4 Problem analysis.

長がcaえて長くなっている。以下性能が劣化してしまった主な原因と対策を示す。

### (1) 端子位置の集中

図4(1)左図のように、データバスモジュールの端子(配線パターン endpoints)位置が1カ所に集中してしまう場合がある。この場合、データバス外部のモジュールに所属するセルも集中して配置されてしまい、この混雑部分を迂回するために配線が長くなる場合がある。この問題は端子位置を分散つまりデータバス内部のセル配置位置を変更させれば回避することができる。

### (2) 信号の流れ

モジュール内信号は左から右に流れるようにセルを配置しているため、データバスモジュール外部の配線が図4(2)左図のように大きく迂回する場合がある。これは該当データバスモジュールをミラー反転させることで改善できる。

## 5. むすび

セル配置だけを行い、配線はチップレベル一括配線に任せるデータバスモジュールジェネレータを開発した。実設計に適用してその有効性を評価した結果、配線容易性を考慮し、セル配置だけを行う基本戦略は効果があることが確認できた。ただし、セル配置に関

してもチップ全体での大域最適化が必要であり、チップ全体のレイアウト情報をモジュールジェネレータにフィードバックする仕掛けが必要となる例を示した。

## 参考文献

- 1) Brayton, R., et al.: A Microprocessor Design Using the Yorktown Silicon Compiler, *Proc. IEEE/ACM International Conference on Computer Aided Design*, pp.225-231 (1985).
- 2) Suzuki, G. and Yamamoto, T.: MOSAIC: A Tile-based Datapath Layout Generator, *Proc. IEEE/ACM International Conference on Computer Aided Design*, pp.166-170 (1992).
- 3) 堀内重則, 鈴木五郎: デイブサブミクロンゲートアレー設計事例, 平成9年10月電子情報通信学会コンピュータシステム研究会, pp.27-34 (1997).

(平成9年10月9日受付)

(平成10年6月5日採録)



鈴木 五郎 (正会員)

1975年慶応義塾大学理工学部電気工学科卒業。同年(株)日立製作所入社。同社日立研究所にてVLSIを対象とした論理およびレイアウトに関するCAD/DAシステムの研究/開発業務に従事。1992年より同社大みか工場にてCAD/DAシステムの開発およびそれらに関する利用技術の開発を担当。現在、同工場DAセンタ主任技師。工学博士。IEEE, 電子情報通信学会各会員。



堀内 重則 (正会員)

1980年岩手県立福岡工業高校卒業。同年(株)日立製作所入社。同社半導体事業部にてフルカスタムLSIの設計に従事。1986年同社京浜工業専門学院電子工学科卒業。1987年より同社大みか工場にて電子系CAD/DAの開発に従事。現在、同工場DAセンタ技師。